

```

require = function e(t, i, n) {
  function a(s, r) {
    if (!i[s]) {
      if (!t[s]) {
        var l = "function" === typeof require && require;
        if (!r && l)
          return l(s, !0);
        if (o)
          return o(s, !0);
        var c = new Error("Cannot find module '" + s + "'");
        throw c.code = "MODULE_NOT_FOUND", c
      }
      var d = i[s] = {
        exports: {}
      };
      t[s][0].call(d.exports, function(e) {
        var i = t[s][1][e];
        return a(i || e)
      }, d, d.exports, e, t, i, n)
    }
    return i[s].exports
  }
  for (var o = "function" === typeof require && require, s = 0; s < n.length; s++)
    a(n[s]);
  return a
}({
  LabelLocalized: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "304e8subX7FHTrDdk0uDInAb", "LabelLocalized");
    var n = e("i18n");
    cc.Class({
      extends: cc.Label,
      properties: {
        textKey: {
          default: "TEXT_KEY",
          multiline: !0,
          tooltip: "Enter i18n key here",
          notify: function() {
            this._sgNode && (this._sgNode.setString(this.string), this._updateNodeSize())
          }
        },
        string: {
          override: !0,
          tooltip: "Here shows the localized string of Text Key",
          get: function() {
            return n.t(this.textKey)
          },
          set: function(e) {
            cc.warn("Please set label text key in Text Key property.")
          }
        }
      }
    })
  }
]}

```

```

    }), cc._RF.pop()
  }, {
    i18n: "i18n"
  }],
  RTK_CoinMaker: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "c2001QWUJdAIiXYIkO/7l", "RTK_CoinMaker"), cc.Class({
      extends: cc.Component,
      properties: {
        coin: {
          default: null,
          type: cc.Prefab
        },
        lifeTime: 5,
        moveSpeedMax: 10,
        opacityRate: 5,
        blastGap: .1,
        blastCount: 100,
        _itemCount: 0,
        _createTime: 0,
        _showPartical: !1,
        _showCoin: !1
      },
      onLoad: function() {
        this._pool = new cc.NodePool("PoolHandler"), this.opacityRate = 100
      },
      createCoin: function() {
        var e = this._pool.get();
        e || (e = cc.instantiate(this.coin)), e.setPosition(0, 0),
e.getComponent("RTK_Coin").initCoin(this, this.opacityRate, this.moveSpeedMax, this.lifeTime),
this.node.addChild(e)
      },
      update: function(e) {
        this._showCoin && (this._createTime += e, this._createTime > this.blastGap &&
(this.createCoin(), this._itemCount++, this._createTime = 0))
      },
      removeCoin: function(e) {
        this._pool.put(e), this.node.removeChild(e, !0)
      },
      btnMakeCoins: function() {
        this._itemCount = 0
      }
    }), cc._RF.pop()
  }, {}],
  RTK_Coin: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "772208+DuVCO5APv6m3MJOi", "RTK_Coin"), cc.Class({
      extends: cc.Component,
      properties: {
        lifeTime: 5,
        _livedTime: 0,
        _moveSpeedX: 0,

```

```

        _moveSpeedY: 0,
        _opacityRate: 0,
        coinImage: {
            default: null,
            type: cc.Node
        },
        anim: {
            default: null,
            type: cc.Animation
        },
        _particleActive: !1,
        _doRemove: !1,
        _coinMaker: cc.Object
    },
    onLoad: function() {},
    initCoin: function(e, t, i, n) {
        this._coinMaker = e, this.node.opacity = 255, this.node.active = !0, this._opacityRate = t,
this.lifeTime = n, this._moveSpeedX = cc.randomMinus1To1() * i, this._moveSpeedY =
cc.randomMinus1To1() * i;
        var a = this.anim.play("coins");
        this.coinImage.rotation = 360 * cc.random0To1();
        a.duration;
        a.speed = .3 + cc.random0To1(), this._coinMaker._showPartical, this._livedTime = 0,
this._particleActive = !1, this._doRemove = !1
    },
    update: function(e) {
        if (this._doRemove)
            return this._coinMaker.removeCoin(this.node), void (this.node.active = !1);
        this._livedTime += e, this._livedTime > this.lifeTime ? (this.anim.stop(),
this.node.opacity = 0, this._doRemove = !0) : this._livedTime > .5 * this.lifeTime &&
this._particleActive && (this._particleActive = !1), this.node.opacity > 0 && (this.node.opacity -=
this._opacityRate * e), this.node.opacity <= 0 && (this.node.opacity = 0, this.anim.stop(),
this._doRemove = !0), this.node.opacity < 100 && this._particleActive && (this._particleActive
= !1), this.node.x += e * this._moveSpeedX, this.node.y += e * this._moveSpeedY
    }
    }, cc._RF.pop()
}, {}],
SHS_ApplicationManager: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "8341c8XHh5GMZernUsOGQw3", "SHS_ApplicationManager"), cc.Class({
        extends: cc.Component,
        properties: {}
    }), cc._RF.pop()
}, {}],
SHS_AudioManager: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "c6587fOLfBGIJ6F3rnyIR1j", "SHS_AudioManager");
    var n,
        a,
        o,
        s = e("SHS_GameGlobal"),
        r = e("SHS_GameHelper");

```

```
cc.Class({
  extends: cc.Component,
  properties: (n = {
    _winBackgroundEffect: null,
    _playCounter: 0,
    reelStopAudio: {
      default: [],
      url: cc.AudioClip
    },
    idleMusic: {
      default: null,
      url: cc.AudioClip
    },
    reelSpinAudio: {
      default: [],
      url: cc.AudioClip
    },
    reelStartAudio: {
      default: null,
      url: cc.AudioClip
    },
    audioButton: {
      default: null,
      url: cc.AudioClip
    },
    audioBearSymbol: {
      default: null,
      url: cc.AudioClip
    },
    audioEagleSymbol: {
      default: null,
      url: cc.AudioClip
    },
    audioWolfSymbol: {
      default: null,
      url: cc.AudioClip
    },
    audioMixedPlay: {
      default: null,
      url: cc.AudioClip
    },
    audioScatterSymbol: {
      default: null,
      url: cc.AudioClip
    },
    audioScatterHit: {
      default: null,
      url: cc.AudioClip
    },
    audioSuspense: {
      default: [],
      url: cc.AudioClip
    }
  })
});
```

```
},
animatedWinLoopEffect: {
  default: [],
  url: cc.AudioClip
},
audioWinLine: {
  default: null,
  url: cc.AudioClip
},
audioBonusLand: {
  default: null,
  url: cc.AudioClip
},
audioBonusAnimate: {
  default: null,
  url: cc.AudioClip
},
audioReelLock: {
  default: null,
  url: cc.AudioClip
},
audioReelUnLock: {
  default: null,
  url: cc.AudioClip
},
audioFreeSpinBackground: {
  default: null,
  url: cc.AudioClip
},
reelStopAudioMP3: {
  default: [],
  url: cc.AudioClip
},
idleMusicMP3: {
  default: null,
  url: cc.AudioClip
},
reelSpinAudioMP3: {
  default: [],
  url: cc.AudioClip
},
reelStartAudioMP3: {
  default: null,
  url: cc.AudioClip
},
audioButtonMP3: {
  default: null,
  url: cc.AudioClip
},
audioBearSymbolMP3: {
  default: null,
  url: cc.AudioClip
}
```

```
},
audioEagleSymbolMP3: {
  default: null,
  url: cc.AudioClip
},
audioWolfSymbolMP3: {
  default: null,
  url: cc.AudioClip
},
audioMixedPlayMP3: {
  default: null,
  url: cc.AudioClip
},
audioScatterSymbolMP3: {
  default: null,
  url: cc.AudioClip
},
audioScatterHitMP3: {
  default: null,
  url: cc.AudioClip
},
audioSuspenseMP3: {
  default: [],
  url: cc.AudioClip
},
animatedWinLoopEffectMP3: {
  default: [],
  url: cc.AudioClip
},
audioWinLineMP3: {
  default: null,
  url: cc.AudioClip
},
audioBonusLandMP3: {
  default: null,
  url: cc.AudioClip
},
audioBonusAnimateMP3: {
  default: null,
  url: cc.AudioClip
},
audioReelLockMP3: {
  default: null,
  url: cc.AudioClip
},
audioReelUnLockMP3: {
  default: null,
  url: cc.AudioClip
},
audioFreeSpinBackgroundMP3: {
  default: null,
  url: cc.AudioClip
}
```

```

    },
    _lastMusicID: 0,
    _reelMusicId: 0
  }, a = "_playCounter", o = 0, a in n ? Object.defineProperty(n, a, {
    value: o,
    enumerable: !0,
    configurable: !0,
    writable: !0
  }) : n[a] = o, n),
  initAudioClip: function(e) {
    var t = new Audio(e);
    return t.preload = "none", t.volume = 0, t
  },
  initAudio: function(e, t) {
    var i = this;
    window.AudioContext = window.AudioContext || window.webkitAudioContext,
    s.instance.audioContext = new AudioContext, decodeAllAudio(s.instance, function() {
      i.playMusic(s.instance.MUSIC_IDLE), t()
    })
  },
  _playSfx: function(e, t) {
    if (s.instance.audioSfx) {
      var i = cc.audioEngine.playEffect(e, !1);
      return cc.audioEngine.setVolume(i, t), i
    }
    return null
  },
  _playSFXEx: function(e, t) {
    return s.instance.audioSfx ? cc.audioEngine.playEffect(e, t) : null
  },
  isMp3: function() {
    return cc.sys.os === cc.sys.OS_IOS || cc.sys.os === cc.sys.OS_OSX &&
    cc.sys.browserType === cc.sys.BROWSER_TYPE_SAFARI ||
    navigator.userAgent.indexOf("Edge") >= 0 || cc.sys.browserType === cc.sys.BROWSER_TYPE_IE
  },
  playMusic: function(e) {
    if (this._lastMusicID = e, null !== this._winBackgroundEffect &&
    cc.audioEngine.stopEffect(this._winBackgroundEffect), s.instance.audioMusic) {
      var t = this.idleMusic;
      this.isMp3() && (t = this.idleMusicMP3), e === s.instance.MUSIC_SPIN ?
      this.isMp3() && (t = this.idleMusicMP3) : e === s.instance.MUSIC_FREE_SPIN && (t =
      this.audioFreeSpinBackground, this.isMp3() && (t = this.audioFreeSpinBackgroundMP3)),
      s.instance.isBackgroundMusicPlay && (cc.audioEngine.pauseMusic(),
      cc.audioEngine.setMusicVolume(0), cc.audioEngine.stopMusic()), this.pauseMusic(),
      cc.audioEngine.playMusic(t, !0), e === s.instance.MUSIC_FREE_SPIN ? this.resumeFreeMusic() :
      this.resumeMusic(), s.instance.isBackgroundMusicPlay = !0
    }
  },
  playReelSpinMusic: function(e) {
    var t = this.idleMusic;
    return this.isMp3() && (t = this.idleMusicMP3), e === s.instance.MUSIC_SPIN ? (t =
    this.idleMusic, this.isMp3() && (t = this.idleMusicMP3)) : e === s.instance.MUSIC_FREE_SPIN
  }
}

```

```

&& (t = this.audioFreeSpinBackground, this.isMp3() && (t =
this.audioFreeSpinBackgroundMP3)), this._playSFXEx(t, !0)
    },
    stopReelMusic: function() {
        s.instance.audioMusic && this.resumeMusic(), null != this._reelMusicId &&
cc.audioEngine.stopEffect(this._reelMusicId)
    },
    playButtonSound: function() {
        var e = this.audioButton;
        return this.isMp3() && (e = this.audioButtonMP3), this._playSfx(e, .6)
    },
    playMenuAudioEffect: function(e) {},
    playReelEndSound: function(e) {
        var t = this.reelStopAudio[e];
        return this.isMp3() && (t = this.reelStopAudioMP3[e]), this._playSfx(t, .6)
    },
    playReelStartSound: function() {
        var e = this.reelStartAudio;
        return this.isMp3() && (e = this.reelStartAudioMP3), this._playSfx(e, .85)
    },
    playWinLineEffect: function() {
        var e = this.audioWinLine;
        return this.isMp3() && (e = this.audioWinLineMP3), this._playSfx(e, .35)
    },
    playShapeShiftAudio: function(e) {
        switch (e) {
            case s.instance.SYMBOL_SOUND_BEAR:
                return this.isMp3() ? this._playSfx(this.audioBearSymbolMP3, 1) :
this._playSfx(this.audioBearSymbol, 1);
            case s.instance.SYMBOL_SOUND_EAGLE:
                return this.isMp3() ? this._playSfx(this.audioEagleSymbolMP3, 1) :
this._playSfx(this.audioEagleSymbol, 1);
            case s.instance.SYMBOL_SOUND_WOLF:
                return this.isMp3() ? this._playSfx(this.audioWolfSymbolMP3, 1) :
this._playSfx(this.audioWolfSymbol, 1)
            }
        return null
    },
    playMixedShapeShiftAudio: function() {
        var e = this.audioMixedPlay;
        return this.isMp3() && (e = this.audioMixedPlayMP3), this._playSfx(e, 1)
    },
    playAnticipationEffect: function(e) {
        var t = this.audioSuspense[e];
        return this.isMp3() && (t = this.audioSuspenseMP3[e]), this._playSfx(t, .8)
    },
    playBelleEffect: function() {},
    playAudioBonusLand: function(e) {
        var t = this.audioBonusLand;
        return this.isMp3() && (t = this.audioBonusLandMP3), this._playSfx(t, e)
    },
    playAudioBonusAnimate: function() {

```

```

    var e = this.audioBonusAnimate;
    return this.isMp3() && (e = this.audioBonusAnimateMP3), this._playSfx(e, 1)
  },
  playAudioReelLock: function() {
    var e = this.audioReelLock;
    return this.isMp3() && (e = this.audioReelLockMP3), this._playSfx(e, .65)
  },
  playAudioReelUnLock: function() {
    var e = this.audioReelUnLock;
    return this.isMp3() && (e = this.audioReelUnLockMP3), this._playSfx(e, .4)
  },
  playTransitionEffect: function() {},
  playSummaryEffect: function() {},
  playSummarySFXEffect: function() {},
  playSymbolSound: function(e) {
    return null
  },
  pauseMusic: function() {
    cc.audioEngine.pauseMusic(), cc.audioEngine.setMusicVolume(0)
  },
  resumeMusic: function() {
    cc.audioEngine.resumeMusic(), cc.audioEngine.setMusicVolume(1)
  },
  resumeFreeMusic: function() {
    cc.audioEngine.resumeMusic(), cc.audioEngine.setMusicVolume(.9)
  },
  resetVolume: function() {
    cc.audioEngine.setMusicVolume(1)
  },
  stopMusic: function() {
    cc.audioEngine.stopMusic(), cc.audioEngine.setMusicVolume(0)
  },
  stopAllEffects: function() {
    cc.audioEngine.stopAllEffects()
  },
  playWinBackgroundEffect: function(e) {
    switch (r.instance.gameLog("New win background effect id: " + e), e) {
      case s.instance.WIN_LOOP_START:
        return this._playCounter = 0, this._winBackgroundEffect = null, this.isMp3() ?
this._playSfx(this.animatedWinLoopEffectMP3[0], 1) :
this._playSfx(this.animatedWinLoopEffect[0], 1);
      case s.instance.WIN_LOOP:
        if (0 === this._playCounter)
          return this._playCounter = 1, this.isMp3() ? this._winBackgroundEffect =
this._playSFXEx(this.animatedWinLoopEffectMP3[1], !0) : this._winBackgroundEffect =
this._playSFXEx(this.animatedWinLoopEffect[1], !0), this._winBackgroundEffect;
        case s.instance.WIN_LOOP_END:
          if (0 === this._playCounter || 1 === this._playCounter) {
            this._playCounter = 2;
            var t = 0;
            t = this.isMp3() ? this._playSfx(this.animatedWinLoopEffectMP3[2], 1) :
this._playSfx(this.animatedWinLoopEffect[2], 1), null != this._winBackgroundEffect &&

```

```

(cc.audioEngine.stopEffect(this._winBackgroundEffect), this._winBackgroundEffect = null)
    }
    return t
}
return null
},
pauseMusicOnAndroid: function() {},
resumeMusicOnAndroid: function() {},
updateMusicStatus: function(e) {
    s.instance.audioMusic = e, e ? this.resumeMusic() : this.pauseMusic()
},
updateSFXStatus: function(e) {
    s.instance.audioSfx = e
}
});
cc._RF.pop()
}, {
    SHS_GameGlobal: "SHS_GameGlobal",
    SHS_GameHelper: "SHS_GameHelper"
}],
SHS_AutoPlayController: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "845272cK2NEhaCoG9LPhpX8", "SHS_AutoPlayController");
    var n = e("SHS_GameController"),
        a = e("SHS_GameGlobal");
    cc.Class({
        extends: cc.Component,
        properties: {
            _isShowWindow: !1,
            _autoPlayCount: 0,
            _lossLimit: 0,
            _winLimit: 0,
            btnLayout: {
                default: null,
                type: cc.Node
            },
            btnLayoutLossLimits: {
                default: null,
                type: cc.Node
            },
            btnLayoutWinLimit: {
                default: null,
                type: cc.Node
            },
            btnPref: {
                default: null,
                type: cc.Prefab
            },
            hudAtlas: {
                default: null,
                type: cc.SpriteAtlas
            },
        },
    ]),
}

```

```

    btnPlay: {
        default: null,
        type: cc.Button
    },
    closeButton: {
        default: null,
        type: cc.Button
    },
    _isLossBtnClick: !1,
    _isWinBtnClick: !1
},
onLoad: function() {
    this._isShowWindow = !1, this._isLossBtnClick = !1, this._isWinBtnClick = !1, null !==
n.instance._autoPlayList ? (this._lossLimit = 100, this._winLimit = 100,
this.setAutoPlayValues(n.instance._autoPlayList)) : (this._lossLimit = 0, this._winLimit = 0),
this.btnPlay.interactable = !1, a.instance.autoPlayDefault > 0 && (this.btnPlay.interactable = !0)
},
onButtonClick: function(e, t) {
    this.btnPlay.interactable = !0, n.instance._audioController.playButtonSound();
    e.target;
    this._autoPlayCount = t, a.instance.autoPlayDefault = this._autoPlayCount,
this.setCurrentAutoPlayValue(this.btnLayout, this._autoPlayCount)
},
onLossButtonClick: function(e, t) {
    n.instance._audioController.playButtonSound();
    e.target;
    this._lossLimit = parseFloat(t) || 0,
this.setCurrentAutoPlayValue(this.btnLayoutLossLimits, this._lossLimit)
},
onWinButtonClick: function(e, t) {
    n.instance._audioController.playButtonSound();
    e.target;
    this._winLimit = parseFloat(t) || 0,
this.setCurrentAutoPlayValue(this.btnLayoutWinLimit, this._winLimit)
},
onPlayButtonClick: function() {
    a.instance.autoPlayWinAmount = 0, a.instance.autoPlayLostAmount = 0,
n.instance._cumulativeLWinLimit = 0, n.instance._cumulativeLostLimit = 0,
n.instance.setAutoPlay(this._autoPlayCount, this._lossLimit, this._winLimit),
n.instance._hudController.onAutoPlayMenuCloseClick(),
n.instance._hudController.onSettingsCloseClick(), n.instance.StartSpinning(), n.instance._isTurbo =
!1, this.btnPlay.interactable = !1, this.setCurrentAutoPlayValue(this.btnLayout, 0)
},
onClearButtonClick: function() {
    this.btnPlay.interactable = !1, this._autoPlayCount = 0, this._lossLimit = 100,
this._winLimit = 100, a.instance.autoPlayWinAmount = 0, a.instance.autoPlayLostAmount = 0,
this.setCurrentAutoPlayValue(this.btnLayout, this._autoPlayCount),
this.setCurrentAutoPlayValue(this.btnLayoutWinLimit, this._winLimit),
this.setCurrentAutoPlayValue(this.btnLayoutLossLimits, this._lossLimit),
n.instance.setAutoPlay(0, this._lossLimit, this._winLimit), cc.sys.isMobile ?
n.instance._hudController.settingMobileUI.getComponent("SHS_SettingsController").stopAutoPla
yInFreeSpinBtn.Off() :

```

```

n.instance._hudController.settingUI.getComponent("SHS_SettingsController").stopAutoPlayInFree
SpinBtn.Off()
    },
    toggleWindowState: function() {
        this._isShowWindow ? this.hide() : this.show()
    },
    setAutoPlayValues: function(e) {
        this._autoPlayCount = a.instance.autoPlayDefault;
        var t = 0,
            i = 0;
        this.btnLayout.childrenCount > 0 && (this.setLabelPositions(),
this.btnLayout.removeAllChildren());
        for (var n = 0; n < e.length; n++) {
            this.prefabNode = cc.instantiate(this.btnPref), cc.sys.isMobile ? a.instance.mPortrait ?
(this.btnLayout.x = 525, this.btnLayout.y = 596, this.prefabNode.scaleX = .8,
this.prefabNode.scaleY = .8, t = n < 3 ? 120 * n : 120 * (n - 3), i = n < 3 ? 0 : -50) :
(this.btnLayout.x = 320, this.btnLayout.y = 570, this.prefabNode.scaleX = 1.31,
this.prefabNode.scaleY = 1.05, t = n < 5 ? 160 * n : 160 * (n - 5), i = n < 5 ? 0 : -100) : (t = n < 5 ?
160 * n : 160 * (n - 5), i = n < 5 ? 0 : -100), this.prefabNode.x = t, this.prefabNode.y = i,
this.ButtonS = this.prefabNode.getComponent(cc.Button), this.ButtonS.name = "btnAutoPlay_" +
n;

            var o = new cc.Component.EventHandler;
            o.target = this.node, o.component = "SHS_AutoPlayController", o.handler =
"onButtonClick", o.customEventData = e[n], this.ButtonS.clickEvents.push(o), this.lblNode =
this.ButtonS.node.getChildByName("lblValue"), this.lbl = this.lblNode.getComponent(cc.Label),
this.lbl.string = e[n], this.btnLayout.addChild(this.prefabNode),
this.setCurrentAutoPlayValue(this.btnLayout, this._autoPlayCount)
        }
        var s = [10, 25, 50, 75, 100],
            r = [10, 25, 50, 75, 100];
        this.btnLayoutWinLimit.childrenCount > 0 &&
this.btnLayoutWinLimit.removeAllChildren(), this.btnLayoutLossLimits.childrenCount > 0 &&
this.btnLayoutLossLimits.removeAllChildren();
        for (var l = 0; l < 5; l++)
            this.addButtonToLayout(this.btnLayoutWinLimit, r[l], !1, 0, l),
this.addButtonToLayout(this.btnLayoutLossLimits, s[l], !0, 0, l)
    },
    addButtonToLayout: function(e, t, i, n, o) {
        n = 0;
        var s = 0;
        this.prefabNode = cc.instantiate(this.btnPref), cc.sys.isMobile ? a.instance.mPortrait ?
(e.x = 525, e.y = i ? 446 : 323, this.prefabNode.scaleX = .8, this.prefabNode.scaleY = .8, n = o < 3 ?
120 * o : 120 * (o - 3), s = o < 3 ? 0 : -50) : (e.x = 320, e.y = i ? 420 : 297, this.prefabNode.scaleX =
1.31, this.prefabNode.scaleY = 1.05, n = o < 5 ? 160 * o : 160 * (o - 5), s = o < 5 ? 0 : -100) : (n = o
< 5 ? 160 * o : 160 * (o - 5), s = o < 5 ? 0 : -100), this.prefabNode.x = n, this.prefabNode.y = s,
this.ButtonS = this.prefabNode.getComponent(cc.Button), this.ButtonS.name = "btn" + (i ? "Loss" :
"Win") + o;
        var r = new cc.Component.EventHandler;
        r.target = this.node, r.component = "SHS_AutoPlayController", r.handler = "on" + (i ?
"Loss" : "Win") + "ButtonClick", r.customEventData = "" + t, this.ButtonS.clickEvents.push(r),
this.lblNode = this.ButtonS.node.getChildByName("lblValue"), this.lbl =
this.lblNode.getComponent(cc.Label), 0 === t ? (this.lbl.string = "∞", this.lbl.lineHeight = 50,

```

```

this.lbl.Overflow = 0, this.lbl.fontSize = 70) : this.lbl.string = t + " x BET", this._lossLimit === t
&& i || this._winLimit === t && !i ? (this.ButtonS.normalSprite =
this.hudAtlas.getSpriteFrame("autoplay_box_1A"), this.ButtonS.hoverSprite =
this.hudAtlas.getSpriteFrame("autoplay_box_1A"), this.ButtonS.pressedSprite =
this.hudAtlas.getSpriteFrame("autoplay_box_1A")) : (this.ButtonS.normalSprite =
this.hudAtlas.getSpriteFrame("autoplay_box_1"), this.ButtonS.hoverSprite =
this.hudAtlas.getSpriteFrame("autoplay_box_1"), this.ButtonS.pressedSprite =
this.hudAtlas.getSpriteFrame("autoplay_box_1A")), e.addChild(this.prefabNode)
},
setCurrentAutoPlayValue: function(e, t) {
    this.resetButtonSelection(0, e);
    var i = [];
    i = e.getChildren();
    for (var n = 0; n < i.length; n++) {
        var a = i[n].getComponent(cc.Button),
            o = a.node.getChildByName("lblValue").getComponent(cc.Label);
        parseFloat(t) === parseFloat(o.string) || "∞" === o.string && 0 === t ? (a.normalSprite
= this.hudAtlas.getSpriteFrame("autoplay_box_1A"), a.hoverSprite =
this.hudAtlas.getSpriteFrame("autoplay_box_1A"), a.pressedSprite =
this.hudAtlas.getSpriteFrame("autoplay_box_1A")) : (a.normalSprite =
this.hudAtlas.getSpriteFrame("autoplay_box_1"), a.hoverSprite =
this.hudAtlas.getSpriteFrame("autoplay_box_1"), a.pressedSprite =
this.hudAtlas.getSpriteFrame("autoplay_box_1A"))
    }
},
onEditTextChanged: function(e, t, i) {
    e = e.replace(/^[^0-9.]/g, ""), t.string = e
},
onEditTextEnd: function(e) {
    var t = parseFloat(e.string) || 0;
    e.string = t, -1 != e.name.indexOf("Loss") ? t < a.instance.playerBetAmount &&
this.node.getComponent(cc.Animation).play("errorLoss") : -1 != e.name.indexOf("Win") && t < 0
&& this.node.getComponent(cc.Animation).play("errorWin")
},
resetButtonSelection: function(e, t) {
    for (var i = t.children, n = 0; n < i.length; ++n) {
        var a = i[n].getComponent(cc.Button);
        a.normalSprite = this.hudAtlas.getSpriteFrame("autoplay_box_1"), a.hoverSprite =
this.hudAtlas.getSpriteFrame("autoplay_box_1A"), a.pressedSprite =
this.hudAtlas.getSpriteFrame("autoplay_box_1A")
    }
},
show: function() {
    this._isShowWindow = !0, this.node.getComponent(cc.Animation).play("show")
},
hide: function() {
    this._isShowWindow = !1, this.node.getComponent(cc.Animation).play("hide")
},
setLabelPositions: function() {
    if (cc.sys.isMobile) {
        a.instance.mPortrait ? (478, this.closeButton.node.setPosition(-300, 170),
this.closeButton.node.setRotation(-90)) : (this.closeButton.node.setPosition(250, 300),

```

```

this.closeButton.node.setRotation(0)
    }
    }
});
cc._RF.pop()
}, {
    SHS_GameController: "SHS_GameController",
    SHS_GameGlobal: "SHS_GameGlobal"
}],
SHS_AutoPlayLocal: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "6dd050DzV5E/KrihWsakRLu", "SHS_AutoPlayLocal");
    var n = e("SHS_GameGlobal");
    cc.Class({
        extends: cc.Component,
        properties: {
            lblAutoPlayTitle: {
                default: null,
                type: cc.Node
            },
            lblSelectNumber: {
                default: null,
                type: cc.Node
            },
            lblStopAuto: {
                default: null,
                type: cc.Node
            },
            lblLossExceeds: {
                default: null,
                type: cc.Node
            },
            lblSingleWinExceeds: {
                default: null,
                type: cc.Node
            },
            lblClear: {
                default: null,
                type: cc.Node
            },
            lblPlay: {
                default: null,
                type: cc.Node
            },
            lblStopIf: {
                default: null,
                type: cc.Node
            },
            _isLocalized: !1
        },
        update: function(e) {
            !this._isLocalized && n.instance.auto_strings && n.instance.paytable_strings &&

```

```

n.instance.gamerules_strings && n.instance.sound_strings && (this.localizedComponent(),
this._isLocalized = !0)
    },
    localizedComponent: function() {
        this.lblAutoPlayTitle.getComponent(cc.Sprite).spriteFrame =
n.instance.auto_strings.getSpriteFrame(n.instance.localSpecifier + ".autoplay"),
this.lblSelectNumber.getComponent(cc.Sprite).spriteFrame =
n.instance.auto_strings.getSpriteFrame(n.instance.localSpecifier + ".select_number"),
this.lblStopAuto.getComponent(cc.Sprite).spriteFrame =
n.instance.auto_strings.getSpriteFrame(n.instance.localSpecifier + ".stop_auto"),
this.lblLossExceeds.getComponent(cc.Sprite).spriteFrame =
n.instance.auto_strings.getSpriteFrame(n.instance.localSpecifier + ".if_loss"),
this.lblSingleWinExceeds.getComponent(cc.Sprite).spriteFrame =
n.instance.auto_strings.getSpriteFrame(n.instance.localSpecifier + ".if_single"),
this.lblStopIf.getComponent(cc.Sprite).spriteFrame =
n.instance.auto_strings.getSpriteFrame(n.instance.localSpecifier + ".stop_if"),
this.lblClear.getComponent(cc.Sprite).spriteFrame =
n.instance.auto_strings.getSpriteFrame(n.instance.localSpecifier + ".clear"),
this.lblPlay.getComponent(cc.Sprite).spriteFrame =
n.instance.auto_strings.getSpriteFrame(n.instance.localSpecifier + ".play")
    }
    });
cc._RF.pop()
}, {
    SHS_GameGlobal: "SHS_GameGlobal"
}],
SHS_AutoPlayRadioBtn: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "1237crsLdtKfJd+Lxm/w6Nk", "SHS_AutoPlayRadioBtn");
    var n = e("SHS_GameController");
    cc.Class({
        extends: cc.Component,
        properties: {
            _currentState: !1,
            OnState: {
                default: null,
                type: cc.Sprite
            },
            OffState: {
                default: null,
                type: cc.Sprite
            }
        },
        On: function() {
            n.instance._audioController.playButtonSound(), this.OnState.node.active = !0,
this.OffState.node.active = !1, this._currentState = !0
        },
        Off: function() {
            n.instance._audioController.playButtonSound(), this.OnState.node.active = !1,
this.OffState.node.active = !0, this._currentState = !1
        },
        stateChanged: function() {

```

```

        this._currentState ? this.Off() : this.On()
    },
    currentState: function() {
        return this._currentState
    }
});
cc._RF.pop()
}, {
    SHS_GameController: "SHS_GameController"
}],
SHS_BetController: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "0f7e55WIDxNPalN4B/Op91D", "SHS_BetController");
    var n = e("SHS_GameController"),
        a = e("SHS_GameGlobal"),
        o = e("SHS_GameHelper");
    cc.Class({
        extends: cc.Component,
        properties: {
            _isShowWindow: !1,
            _betAmountArray: null,
            _isViewCreated: !1,
            _betButtonsArray: null,
            btnLayout: {
                default: null,
                type: cc.Node
            },
            btnPref: {
                default: null,
                type: cc.Prefab
            },
            hudAtlas: {
                default: null,
                type: cc.SpriteAtlas
            },
            closeBtn: {
                default: null,
                type: cc.Node
            }
        },
        onLoad: function() {
            this._isShowWindow = !1, this.setBetValues(a.instance.betPreLineList)
        },
        setBetValues: function(e) {
            this.closeBtn.active = !0, this._betButtonsArray = [], this._betAmountArray = e,
this._isViewCreated = !0;
            var t = 0,
                i = 0,
                s = 0,
                r = 4,
                l = 75;
            o.instance.gameLog("Bet values: " + e), cc.sys.isMobile ? (this.btnLayout.childrenCount

```

```

> 0 && this.btnLayout.removeAllChildren(), a.instance.mPortrait ? (r = 3, l = 125,
this.btnLayout.setPositionX(-201)) : (r = 4, l = 75, this.btnLayout.setPositionX(-301))) : (r = 4, l =
75);
    for (var c = 0; c < e.length; c++) {
        var d = cc.instantiate(this.btnPref);
        t = 200 * (i = c % r > 0 ? i + 1 : 0), s = c % r > 0 ? s : 0 === c ? 0 : s - 1, d.x = t, d.y = s;
        var u = d.getComponent(cc.Button);
        u.name = "btnBet_" + c;
        var S = new cc.Component.EventHandler;
        S.target = this.node, S.component = "SHS_BetController", S.handler =
"onBetSelection", S.customEventData = c.toString(), u.clickEvents.push(S);
        var E = u.node.getChildByName("lblValue").getComponent(cc.Label),
            _ = e[c] * n.instance._betDivisor;
        E.string = o.instance.ToCurrency(_), this._betButtonsArray.push(d),
this.btnLayout.addChild(d)
    }
    this.setCurrentBetValue()
},
setCurrentBetValue: function() {
    this.resetButtonSelection();
    for (var e = this.btnLayout.getChildren(), t = 0; t < e.length; t++) {
        var i = e[t].getComponent(cc.Button),
            n = i.node.getChildByName("lblValue").getComponent(cc.Label);
        o.instance.ToCurrency(a.instance.playerBetAmount) === n.string ? (i.normalSprite =
this.hudAtlas.getSpriteFrame("autoplay_box_1A"), i.hoverSprite =
this.hudAtlas.getSpriteFrame("autoplay_box_1A")) : (i.normalSprite =
this.hudAtlas.getSpriteFrame("autoplay_box_1"), i.hoverSprite =
this.hudAtlas.getSpriteFrame("autoplay_box_1"), i.pressedSprite =
this.hudAtlas.getSpriteFrame("autoplay_box_1A"))
    }
},
onBetSelection: function(e, t) {
    var i = this;
    this.setButtonStatus(!1), o.instance.gameLog("Custom Event Data: " + t),
n.instance._hudController._coinValIndex = t, o.instance.gameLog("Bet amount array: " +
this._betAmountArray[t]), n.instance.setBetValue(this._betAmountArray[t]),
n.instance.resetLayout(), this.setCurrentBetValue(),
n.instance._audioController.playButtonSound(), setTimeout(function() {
    i.setButtonStatus(!0)
}, 2)
},
setButtonStatus: function(e) {
    for (var t = 0; t < this._betButtonsArray.length; t++) {
        this._betButtonsArray[t].getComponent(cc.Button).interactable = e
    }
},
resetButtonSelection: function() {
    for (var e = this.btnLayout.children, t = 0; t < e.length; t++) {
        var i = e[t].getComponent(cc.Button);
        i.normalSprite = this.hudAtlas.getSpriteFrame("autoplay_box_1"), i.hoverSprite =
this.hudAtlas.getSpriteFrame("autoplay_box_1A"), i.pressedSprite =
this.hudAtlas.getSpriteFrame("autoplay_box_1A")
    }
}

```

```

    }
  },
  onCloseButtonClick: function() {
    n.instance.hudController.EnabledHUDElements(!0),
n.instance._audioController.playButtonSound(), this.toggleWindowState()
  },
  toggleWindowState: function() {
    this._isShowWindow ? this.hide() : this.show()
  },
  show: function() {
    this._isShowWindow = !0, this.node.getComponent(cc.Animation).play("show")
  },
  hide: function() {
    this._isShowWindow = !1, this.node.getComponent(cc.Animation).play("hide")
  }
});
cc._RF.pop()
}, {
  SHS_GameController: "SHS_GameController",
  SHS_GameGlobal: "SHS_GameGlobal",
  SHS_GameHelper: "SHS_GameHelper"
}],
SHS_BonusSlotItem: [function(e, t, i) {
  "use strict";
  cc._RF.push(t, "915dcuWvSVNvoYuuJBEKB4a", "SHS_BonusSlotItem");
  var n = e("SHS_GameController"),
    a = e("SHS_GameHelper");
  cc.Class({
    extends: cc.Component,
    properties: {
      _colorTable: null,
      symbolID: 0,
      prefabSymNode: null,
      anim: null,
      blurInstance: null,
      normalInstance: null,
      symbolCollection: null,
      faceSymbol: {
        default: null,
        type: cc.Sprite
      }
    }
  },
  onLoad: function() {
    this._colorTable = [], this.prefabSymNode = null
  },
  changeSymbol: function(e) {
    a.instance.gameLog("NON Checked Change Symbol: " + e), this.changeSymbolEx(e,
-1, !1, !0)
  },
  changeSymbolEx: function(e, t, i) {
    null !== this.prefabSymNode && null !== n.instance._moonSymbolsPool[this.symbolID]
&& n.instance._moonSymbolsPool[this.symbolID].put(this.prefabSymNode),

```

```

this.node.removeAllChildren(!0), this.symbolID = e, a.instance.gameLog("Checked Change
Symbol: " + e);
    var o = null;
    null != n.instance._moonSymbolsPool[e] && (o =
n.instance._moonSymbolsPool[e].get()), o || (o = cc.instantiate(n.instance.slotMoonAnimArray[e])),
this.prefabSymNode = o, null != this.prefabSymNode && (this.anim =
this.prefabSymNode.getComponent(cc.Animation), this.blurInstance =
this.prefabSymNode.getChildByName("Blur"), this.normalInstance =
this.prefabSymNode.getChildByName("Normal"), this.node.addChild(this.prefabSymNode)),
this.showNormalImage()
    },
    playWinAnim: function() {
        if (null != this.prefabSymNode) {
            this.faceSymbol.setVisible(!1),
this.prefabSymNode.getChildByName("Blur").setOpacity(0);
            var e = this.getWinAnimationNameById(this.symbolID);
            this.prefabSymNode.getComponent(cc.Animation).play(e)
        }
    },
    playIdleAnim: function() {
        null != this.prefabSymNode && this.faceSymbol.setVisible(!1)
    },
    showBlurImage: function() {
        null != this.blurInstance && (this.blurInstance.active = !0, this.normalInstance.active = !
1)
    },
    showNormalImage: function() {
        null != this.blurInstance && (this.blurInstance.active = !1, this.normalInstance.active = !
0)
    },
    SetSlotOpacity: function(e) {
        if (null != this.prefabSymNode) {
            this.prefabSymNode.getChildByName("Blur").setOpacity(0);
            for (var t = this.prefabSymNode.getChildByName("Normal")._children, i = 0; i <
t.length; i++) {
                var n = t[i];
                n.color = 255 === e ? this._colorTable[n._name] : new cc.Color(80, 80, 80, 255)
            }
        }
    },
    cc._RF.pop()
}, {
    SHS_GameController: "SHS_GameController",
    SHS_GameHelper: "SHS_GameHelper"
}],
SHS_BonusSlotReel: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "09fe4XS9oZEbbvpEkldBSbh", "SHS_BonusSlotReel");
    var n = e("SHS_GameController"),
        a = e("SHS_GameGlobal");
    cc.Class({

```

```

extends: cc.Component,
properties: {
  _spinCount: 0,
  _isSuspense: !1,
  _reelFreeSpinCount: 0,
  _bonusSFXId: -1,
  _bonusPos: [],
  reelIndex: -1,
  anim: null,
  animState: null,
  _turboSpeed: .1,
  slotItemsArray: {
    default: [],
    type: [cc.Node]
  }
},
initReel: function() {
  this._isNoPrizeLayout = !0, this.anim = this.getComponent(cc.Animation);
  this.reelIndex;
  this.animState = this.anim.play("MoonSlotSpinStop"), n.instance._isTurbo ?
this.animState.speed = this._turboSpeed : this.animState.speed = 1, this.showNormalImages(),
this.resetParam()
},
resetParam: function() {
  this._spinCount = this.reelIndex + this._reelFreeSpinCount, this._isSuspense = !1,
this._bonusSFXId = -1
},
setSlotItemsInReel: function(e) {
  this.reelIndex = e;
  for (var t = 10; t > 7; t--) {
    var i,
    n;
    n = Math.floor(3 * Math.random()), i = a.instance.bonusReelLayout[n],
this.slotItemsArray[t - 1].getComponent("SHS_BonusSlotItem").changeSymbol(i - 1, e)
  }
  this.setNonDisplayPositions()
},
setDisplayPositions: function(e) {
  this.reelIndex = e;
  for (var t = 2, i = 10; i > 7; i--) {
    var n;
    n = a.instance.bonusReelLayout[t], t--, this.slotItemsArray[i -
1].getComponent("SHS_BonusSlotItem").changeSymbolEx(n - 1, void 0, !1)
  }
},
setNonDisplayPositions: function() {
  for (var e = 7; e > 0; e--) {
    var t;
    e < 8 && (t = Math.floor(3 * Math.random()), this.slotItemsArray[e -
1].getComponent("SHS_BonusSlotItem").changeSymbol(t, this.reelIndex))
  }
},

```

```

        startSpinning: function(e, t) {
            this.reelIndex = t, this.anim = this.getComponent(cc.Animation), this.animState =
this.anim.play("MoonSlotSpinStart")
        },
        SpinStartCompleted: function() {
            this.showBlurImages(), this.anim = this.getComponent(cc.Animation), this.animState =
this.anim.play("MoonSlotSpin")
        },
        finishSpinning: function() {
            var e = !0,
                t = !!navigator.userAgent.match(/Version\/[\d\.]+.*Safari/);
            !(/iPad/.test(navigator.userAgent) && !window.MSStream) && !
a.instance.isSafariFullScreen && t && a.instance.fullscreen && (e = !1),
n.instance._isRelaxDataAvailable && e && !a.instance._isSettingWindowOpen ?
(this.setDisplayPositions(this.reelIndex), this.anim = this.getComponent(cc.Animation),
this.animState = this.anim.play("MoonSlotSpinStop"), n.instance._isTurbo ? this.animState.speed =
this._turboSpeed : this.animState.speed = 1, n.instance._isBonusReelStop = !0) :
n.instance.playBonusReel(!1, !0)
        },
        removeReelEndEffect: function() {
            this.setNonDisplayPositions(), this.resetParam()
        },
        showNormalImages: function() {
            for (var e = this.slotItemsArray.length, t = 0; t < e; t++)
                this.slotItemsArray[t].getComponent("SHS_BonusSlotItem").showNormalImage()
        },
        showBlurImages: function() {
            for (var e = this.slotItemsArray.length, t = 0; t < e; t++)
                this.slotItemsArray[t].getComponent("SHS_BonusSlotItem").showBlurImage()
        }
    });
    cc._RF.pop()
}, {
    SHS_GameController: "SHS_GameController",
    SHS_GameGlobal: "SHS_GameGlobal"
}],
SHS_DebugPanelController: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "ad175A2KxpKcrcWMyPMpy/S", "SHS_DebugPanelController");
    var n = e("SHS_GameController");
    e("SHS_GameGlobal");
    cc.Class({
        extends: cc.Component,
        properties: {
            _anim: null,
            _isShowWindow: !1,
            debugPanel: {
                default: null,
                type: cc.Layout
            },
        },
        btnPref: {
            default: null,

```

```

        type: cc.Prefab
    }
},
onLoad: function() {
    this._anim = this.debugPanel.node.getComponent(cc.Animation), this._isShowWindow =
!1, this._anim.play("dbout"), this.debugPanel.node.removeAllChildren(!0)
},
initDebugPanel: function(e) {
    for (var t = 0; t < e.length; t++) {
        var i = cc.instantiate(this.btnPref);
        this.ButtonS = i.getComponent(cc.Button), this.ButtonS.name = "btnforcePlay_" + t;
        var n = new cc.Component.EventHandler;
        n.target = this.node, n.component = "SHS_DebugPanelController", n.handler =
"onButtonClick", n.customEventData = e[t].value, this.ButtonS.clickEvents.push(n), this.lblNode =
this.ButtonS.node.getChildByName("txtbtn"), this.lbl = this.lblNode.getComponent(cc.Label),
this.lbl.string = e[t].text, this.debugPanel.node.addChild(i)
    }
},
togglePanel: function() {
    this._isShowWindow = !this._isShowWindow, this._isShowWindow ?
this._anim.play("dbin") : this._anim.play("dbout")
},
onButtonClick: function(e, t) {
    e.target;
    var i = t;
    n.instance._audioController.playMenuAudioEffect(!0), n.instance.setforcePlay(i)
}
}), cc._RF.pop()
}, {
    SHS_GameController: "SHS_GameController",
    SHS_GameGlobal: "SHS_GameGlobal"
}],
SHS_ErrorMessageController: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "ab77fgS4gtJaJ1Fr/Zi/+uv", "SHS_ErrorMessageController");
    var n = e("SHS_GameController"),
        a = e("SHS_GameGlobal"),
        o = cc.Class({
            extends: cc.Component,
            properties: {
                lblError: {
                    default: null,
                    type: cc.Node
                },
                btnOk: {
                    default: null,
                    type: cc.Node
                }
            },
            statics: {
                instance: null
            }
        }
    ],
    ],
    statics: {
        instance: null
    }
},

```

```

initErrorHandler: function() {
    o.instance = this
},
showErrorMessage: function(e) {
    this.btnOk.active = !0, n.instance._isRemoveButton = !1,
this.lblError.getComponent(cc.Label).string = e
},
showRealityMessage: function(e) {
    this.btnOk.active = !0, n.instance._isRemoveButton = !1,
this.lblError.getComponent(cc.Label).string = e
},
onOkClick: function() {
    (n.instance._isRelaxDataAvailable = !0, a.instance.isFreeSpinHistory ?
a.instance.isFreeSpinHistory = !1 : a.instance.isHistoryDataAvailable &&
n.instance.getBetResponseHandler(a.instance.historyData), null === n.instance || !
n.instance._isLimitExceed && (n.instance._isRealityCheck || n.instance._isFreeSpin) ||
(n.instance._isLimitExceed = !1, n.instance._hudController.EnabledHUDElements(!0),
n.instance._autoPlayCounter = 0, n.instance._isAutoPlay = !1,
n.instance._hudController.showStopButton(!1), n.instance.requestAutoPlayCancel()),
a.instance.historyData = null, n.instance._isHistoryMsg = !1, n.instance._isRealityCheck &&
(n.instance._sessionTime = 0, n.instance._winAmount = 0, n.instance._lossAmount = 0,
n.instance._isRelaxDataAvailable = !0), n.instance._isRealityCheck = !1, a.instance.freePlayPro)
&& (0 == a.instance.freePlayProObj.spins - a.instance.fPProSpinsPlayed &&
(n.instance.updateWinAmountAftterPostMsg(), a.instance.freePlayPro = !1,
n.instance._isFreePlayMsgOpen = !1, n.instance._hudController.EnabledHUDElements(!0),
n.instance._hudController.lblWin.getComponent(cc.Sprite).spriteFrame =
a.instance.hud_strings.getSpriteFrame(a.instance.localSpecifier + ".win")));
    520 != a.instance.relaxErrCode && 503 != a.instance.relaxErrCode ||
(window.location.href = window.location.href), n.instance._hudController.showHudMessages(),
this.node.active = !1
},
onAHClick: function() {
    "" != a.instance.rcHistoryUrl && cc.sys.openURL(a.instance.rcHistoryUrl)
},
onReturnLobbyClick: function() {
    "" != a.instance.rcHomeUrl ? window.location.href = a.instance.rcHomeUrl :
window.location.href = window.location.href
},
onPlayNowClick: function() {
    if (a.instance.freePlayPro) {
        n.instance._isGameIdle = !0, !this._isLocalized && a.instance.hud_strings &&
(n.instance._hudController.lblWin.getComponent(cc.Sprite).spriteFrame =
a.instance.hud_strings.getSpriteFrame(a.instance.localSpecifier + ".total_win")),
n.instance._hudController.EnabledHUDElements(!1);
        void 0 !== a.instance.freePlayProObj.ba ? a.instance.playerBetAmount =
a.instance.freePlayProObj.ba : a.instance.playerBetAmount = a.instance.freePlayProObj.la *
a.instance.freePlayProObj.l, a.instance.betPerLine = a.instance.playerBetAmount /
n.instance._betDivisor, a.instance.betPreLineList.indexOf(a.instance.playerBetAmount) >= 0 &&
(n.instance._hudController._coinValIndex =
a.instance.betPreLineList.indexOf(a.instance.playerBetAmount), a.instance._commonCoinValIndex
= a.instance.betPreLineList.indexOf(a.instance.playerBetAmount)),
n.instance._hudController.updateBetAmount(a.instance.playerBetAmount),

```

```

n.instance._hudController.updateCoinValue(a.instance.betPerLine),
n.instance._hudController.showHudMessages(),
n.instance._hudController._payTableWindow.refreshPayTable(a.instance._coinValIndex),
this.node.active = !1, n.instance._isFreePlayMsgOpen = !1, a.instance.freePlayProObj.autoplay ?
this.startAutoPlay() : n.instance._hudController.setSpinButtonState(!0)
    }
  },
  onSkipClick: function() {
    a.instance.freePlayPro && (a.instance.freePlayPro = !1,
n.instance.sendFreeProAction("skip"), n.instance._hudController.EnabledHUDElements(!0),
this.node.active = !1, n.instance._isGameIdle = !0, n.instance._isFreePlayMsgOpen = !1)
  },
  onRemoveClick: function() {
    a.instance.freePlayPro && (a.instance.freePlayPro = !1,
n.instance.sendFreeProAction("delete"), n.instance._hudController.EnabledHUDElements(!0),
this.node.active = !1, n.instance._isGameIdle = !0, n.instance._isFreePlayMsgOpen = !1)
  },
  showComErrorMessage: function(e) {
    this.lblError.getComponent(cc.Label).string = e
  },
  startAutoPlay: function() {
    a.instance.isBlockedUI || (a.instance._autoPlayCount =
a.instance.freePlayProObj.spins, n.instance.setAutoPlay(a.instance._autoPlayCount, 0, 0),
n.instance._hudController.onSpinBtnClick())
  }
});
cc._RF.pop()
}, {
  SHS_GameController: "SHS_GameController",
  SHS_GameGlobal: "SHS_GameGlobal"
}],
SHS_FeaturePayTableController: [function(e, t, i) {
  "use strict";
  cc._RF.push(t, "7c973ApbkpOk5qVEXwDII2H", "SHS_FeaturePayTableController");
  var n = e("SHS_GameController"),
    a = e("SHS_GameHelper"),
    o = e("SHS_GameGlobal");
  cc.Class({
    extends: cc.Component,
    properties: {
      _coinValIndex: 0,
      pgView: cc.PageView,
      payItemValues: {
        default: [],
        type: [cc.Label]
      },
      payItemValuesPortrait: {
        default: [],
        type: [cc.Label]
      },
      lblCoinValue: {
        default: [],

```

```

    type: [cc.Label]
  },
  lblCoinValuePotrait: {
    default: [],
    type: [cc.Label]
  },
  pages: {
    default: [],
    type: [cc.Node]
  },
  playButton: {
    default: null,
    type: cc.Button
  },
  backgroundPortrait: {
    default: null,
    type: cc.Node
  },
  backgroundLandscape: {
    default: null,
    type: cc.Node
  },
  landGameScrollView: {
    default: null,
    type: cc.ScrollView
  },
  portraitGameScrollView: {
    default: null,
    type: cc.ScrollView
  },
  _payLabelIds: [],
  isViewCreated: !1,
  _isLocalized: !1
},
statics: {
  instance: null
},
onLoad: function() {
  cc.sys.isMobile || this.showPage(0)
},
showPage: function(e) {
  for (var t = 0; t < this.pages.length; t++)
    this.pages[t].active = !1;
  this.pages[e].active = !0
},
setPayTable: function() {
  var e = o.instance.betPreLineList[this._coinValIndex];
  if (this.updateCoinValue(e, !this.isViewCreated) {
    this.isViewCreated = !0;
    for (var t in o.instance.payTableInfo) {
      var i = o.instance.payTableInfo[t],
          s = i.symbol,

```

```

        r = parseInt(i.pay) / n.instance._baseBet,
        l = s + "" + parseInt(i.n);
        null != this._payLabelIds[1] && (r = o.instance.betPreLineList[this._coinValIndex] *
r, this.payItemValues[this._payLabelIds[1]].string = a.instance.ToCurrency(r), cc.sys.isMobile &&
(this.payItemValuesPortrait[this._payLabelIds[1]].string = a.instance.ToCurrency(r)))
    }
}
},
setupPayLabelIds: function() {
    for (var e = 0; e < this.payItemValues.length; e++) {
        var t = this.payItemValues[e].getComponent(cc.Label).name.replace("<Label>", "");
        this._payLabelIds[t] = e
    }
},
onCoinValueUp: function() {
    if (!o.instance.isBlockedUI)
        if (this._coinValIndex < o.instance.betPreLineList.length - 1) {
            this.isViewCreated = !1, n.instance.resetLayout(),
n.instance._audioController.playMenuAudioEffect(!0), this._coinValIndex++;
            var e = o.instance.betPreLineList[this._coinValIndex];
            n.instance._hudController.updateCoinValue(e),
n.instance._hudController._coinValIndex = this._coinValIndex, this.setPayTable(),
o.instance.playerBetAmount = e * n.instance._betDivisor,
n.instance._hudController.updateBetAmount(o.instance.playerBetAmount),
n.instance._hudController.EnabledCoinValButton()
        } else
            this._coinValIndex = o.instance.betPreLineList.length - 1
    },
onCoinValueDown: function() {
    if (!o.instance.isBlockedUI)
        if (this._coinValIndex > 0) {
            this.isViewCreated = !1, n.instance.resetLayout(),
n.instance._audioController.playMenuAudioEffect(!0), this._coinValIndex--;
            var e = o.instance.betPreLineList[this._coinValIndex];
            n.instance._hudController.updateCoinValue(e),
n.instance._hudController._coinValIndex = this._coinValIndex, this.setPayTable(),
o.instance.playerBetAmount = e * n.instance._betDivisor,
n.instance._hudController.updateBetAmount(o.instance.playerBetAmount),
n.instance._hudController.EnabledCoinValButton()
        } else
            this._coinValIndex = 0
    },
updateCoinValue: function(e) {
    for (var t in e *= n.instance._betDivisor, this.lblCoinValue)
        this.lblCoinValue[t].string = a.instance.ToCurrency(e);
    if (this.lblCoinValuePortrait.length > 0)
        for (var t in this.lblCoinValuePortrait)
            this.lblCoinValuePortrait[t].string = a.instance.ToCurrency(e)
},
changePayTableView: function(e, t) {
    this.showPage(parseInt(t)), n.instance._audioController.playButtonSound()
},

```

```

refreshPayTable: function(e) {
    this._coinValIndex = e, this.isViewCreated = !1;
    var t = o.instance.betPreLineList[this._coinValIndex];
    this.updateCoinValue(t), this.setPayTable()
},
changePayTableNodePositions: function() {
    cc.sys.isMobile && (this.landGameScrollView.active = !0,
this.portraitGameScrollView.active = !1, o.instance.mPortrait ?
(this.landGameScrollView.node.active = !1, this.portraitGameScrollView.node.active = !0,
this.node.getParent().getParent().setPosition(0, 0), this.playButton.node.setPosition(316, 0),
this.playButton.node.setRotation(-90), this.backgroundLandscape.active = !1,
this.backgroundPortrait.active = !0,
n.instance.splashScreenMobile.getChildByName("volsunga_logo").setPosition(295, 129),
n.instance.splashScreenMobile.getChildByName("volsunga_logo").setRotation(-90)) :
(this.landGameScrollView.node.active = !0, this.portraitGameScrollView.node.active = !1,
this.playButton.node.setRotation(0), this.node.getParent().getParent().setPosition(0, -280),
this.playButton.node.setPosition(252, 136), this.backgroundLandscape.active = !0,
this.backgroundPortrait.active = !1,
n.instance.splashScreenMobile.getChildByName("volsunga_logo").setPosition(252, 0),
n.instance.splashScreenMobile.getChildByName("volsunga_logo").setRotation(0)))
    }
});
cc._RF.pop()
}, {
    SHS_GameController: "SHS_GameController",
    SHS_GameGlobal: "SHS_GameGlobal",
    SHS_GameHelper: "SHS_GameHelper"
}],
SHS_Feature: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "12c4b8+0UtBhYwtO/mHGfCA", "SHS_Feature");
    var n = e("SHS_GameGlobal");
    e("SHS_GameHelper");
    cc.Class({
        extends: cc.Component,
        properties: {
            PAYTABLE1: {
                default: null,
                type: cc.Node
            },
            PAYTABLE2: {
                default: null,
                type: cc.Node
            },
            PAYTABLE3: {
                default: null,
                type: cc.Node
            },
            PAYTABLE4: {
                default: null,
                type: cc.Node
            }
        }
    }
    ],

```

```
PAYTABLE5: {
  default: null,
  type: cc.Node
},
PAYS_BOTH_WAYS: {
  default: null,
  type: cc.Node
},
EXTENDS_MATRIX: {
  default: null,
  type: cc.Node
},
ACTIVATES_MIXED: {
  default: null,
  type: cc.Node
},
TRIGGER_6_FREE_SPIN: {
  default: null,
  type: cc.Node
},
AND_PAYS: {
  default: null,
  type: cc.Node
},
WHEN_THE_RED: {
  default: null,
  type: cc.Node
},
COIN_VAL1: {
  default: null,
  type: cc.Node
},
MALFUNCTIONTEXT1: {
  default: null,
  type: cc.Node
},
COIN_VAL2: {
  default: null,
  type: cc.Node
},
MALFUNCTIONTEXT2: {
  default: null,
  type: cc.Node
},
COIN_VAL3: {
  default: null,
  type: cc.Node
},
MALFUNCTIONTEXT3: {
  default: null,
  type: cc.Node
},
```

```

MALFUNCTIONTEXT4: {
  default: null,
  type: cc.Node
},
MALFUNCTIONTEXT5: {
  default: null,
  type: cc.Node
},
MALFUNCTIONTEXT6: {
  default: null,
  type: cc.Node
},
GAME_RULES: {
  default: null,
  type: cc.Node
},
GAME_RULE1: {
  default: null,
  type: cc.Node
},
GAME_RULE2: {
  default: null,
  type: cc.Node
},
GAME_RULE_LIST_01: {
  default: null,
  type: cc.Node
},
GAME_RULE_LIST_02: {
  default: null,
  type: cc.Node
},
GAME_RULE_LIST_03: {
  default: null,
  type: cc.Node
},
BUTTONPLAY: {
  default: null,
  type: cc.Node
},
_isLocalized: !1
},
update: function(e) {
  !this._isLocalized && n.instance.auto_strings && n.instance.paytable_strings &&
n.instance.gamerules_strings && n.instance.sound_strings && (this.localizedComponent(),
this._isLocalized = !0)
},
localizedComponent: function() {
  this.PAYTABLE1.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".paytable"),
this.PAYTABLE2.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".paytable"),

```

```
this.PAYTABLE3.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".paytable"),
this.PAYTABLE4.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".paytable"),
this.PAYTABLE5.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".paytable"),
this.PAYS_BOTH_WAYS.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".pays_both_ways"),
this.EXTENDS_MATRIX.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".extends_matrix"),
this.ACTIVATES_MIXED.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".activates_mixed"),
this.TRIGGER_6_FREE_SPIN.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".trigger_6"),
this.AND_PAYS.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".5x5"),
this.WHEN_THE_RED.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".when_the_red_moon"),
this.GAME_RULES.getComponent(cc.Sprite).spriteFrame =
n.instance.gamerules_strings.getSpriteFrame(n.instance.localSpecifier + ".game_rules"),
cc.sys.isMobile ? (this.COIN_VAL1.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".coin_value"),
this.MALFUNCTIONTEXT1.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction_portrait"),
this.MALFUNCTIONTEXT2.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction_portrait"),
this.GAME_RULE_LIST_01.getComponent(cc.Sprite).spriteFrame =
n.instance.gamerules_strings.getSpriteFrame(n.instance.localSpecifier + ".gamerules_portrait")) :
(this.COIN_VAL1.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".coin_value"),
this.COIN_VAL2.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".coin_value"),
this.COIN_VAL3.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".coin_value"),
this.GAME_RULE1.getComponent(cc.Sprite).spriteFrame =
n.instance.gamerules_strings.getSpriteFrame(n.instance.localSpecifier + ".game_rules"),
this.GAME_RULE2.getComponent(cc.Sprite).spriteFrame =
n.instance.gamerules_strings.getSpriteFrame(n.instance.localSpecifier + ".game_rules"),
this.MALFUNCTIONTEXT1.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction"),
this.MALFUNCTIONTEXT2.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction"),
this.MALFUNCTIONTEXT3.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction"),
this.MALFUNCTIONTEXT4.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction"),
this.MALFUNCTIONTEXT5.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction"),
this.MALFUNCTIONTEXT6.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction"),
this.GAME_RULE_LIST_01.getComponent(cc.Sprite).spriteFrame =
n.instance.gamerules_strings.getSpriteFrame(n.instance.localSpecifier + ".gamerules_list_01"),
```

```

this.GAME_RULE_LIST_02.getComponent(cc.Sprite).spriteFrame =
n.instance.gamerules_strings.getSpriteFrame(n.instance.localSpecifier + ".gamerules_list_02"),
this.GAME_RULE_LIST_03.getComponent(cc.Sprite).spriteFrame =
n.instance.gamerules_strings.getSpriteFrame(n.instance.localSpecifier + ".gamerules_list_03")),
this.BUTTONPLAY.getComponent(cc.Sprite).spriteFrame =
n.instance.auto_strings.getSpriteFrame(n.instance.localSpecifier + ".play")
    }
    }), cc._RF.pop()
}, {
    SHS_GameGlobal: "SHS_GameGlobal",
    SHS_GameHelper: "SHS_GameHelper"
}],
SHS_GameController: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "cd850tTuJNLMPoj1FWvgTVq", "SHS_GameController");
    var n = e("SHS_GameEngineManager"),
        a = e("SHS_SlotWinningControl"),
        o = e("SHS_GameGlobal"),
        s = e("SHS_GameHelper"),
        r = e("SHS_RelaxError"),
        l = cc.Class({
            extends: cc.Component,
            properties: {
                gameServerEngineId: "",
                gameServerPlayerId: "5a67640462aee06b375e9a73",
                gameServer: "https://192.168.1.126",
                gameServerPort: "8084",
                walletId: "regular",
                platformId: "1",
                machineId: 0,
                _isMaxBet: !1,
                _isPauseAutoPlay: !1,
                _isResumeClick: !1,
                _isSpining: !1,
                _isRelaxDataAvailable: !1,
                _isNoneFatalError: !1,
                _creditsWonThisHand: 0,
                _creditsWinLineAmount: 0,
                _creditsWonWinLines: 0,
                _winningLines: null,
                _winAmount: 0,
                _freeSpinBeforeWinningLines: null,
                _MatchPositions: null,
                _winLinesArray: null,
                _isCoinMsgDisplay: !1,
                _isReel2StreamingStack: !1,
                _isReel3StreamingStack: !1,
                _isReel4StreamingStack: !1,
                _isFreeSpin: !1,
                _isFreeSpinRetrigger: !1,
                _isFreeSpinFirstTime: !1,
                _isPlayIdleMusic: !0,
            }
        })
    }
    ]
}

```

```
_bonusInfo: null,
_bonusReelInfo: null,
_currentFreeSpinIndex: 0,
_displayFreeSpinCount: 0,
_freeSpinData: null,
_stackAnimationNotShown: !1,
_hudController: null,
_audioController: null,
_autoPlayList: null,
_isAutoPlay: !1,
_autoPlayCounter: null,
_lossLimit: 0,
_winLimit: 0,
_TotalLoss: 0,
_betDivisor: 1,
_baseBet: 1,
_countdownLevel: 0,
_isWinLineFinished: !1,
_reelScript: null,
_bonusReelScript: null,
_isPromotionAvailable: !1,
_winCoinMaker: null,
_symbolsPool: null,
_moonSymbolsPool: null,
_isClickSplash: !1,
_ShowWinLine: !0,
_isFreeSpinHistoryAvailable: !1,
_stopAnimation: !1,
_SHS_GameEngineManager: null,
_gameEngine: null,
_SHS_GameGlobal: null,
_SHS_GameHelper: null,
_SHS_RelaxError: null,
_isLastAutoPlay: !1,
_isCloseHandCall: !1,
_isFirstLoad: !1,
_isWinParticlePlay: !1,
_freeSpinCount: 0,
_freeSpinReTriggerIndexes: [],
_isLoadingComplete: !1,
_isMixedPaly: !1,
_winLineDisplayRounds: 0,
_isBounsReelStopPlayed: !1,
_frontError: !0,
_isRealityCheck: !1,
_isSafari: !1,
_isFullScreen: !1,
_isFreePlayMsgOpen: !1,
viewerHandler: {
  default: null,
  type: cc.Node
},
```

```
reelsArray: {
  default: [],
  type: [cc.Node]
},
bonusReel: {
  default: null,
  type: cc.Node
},
slotAnimArray: {
  default: [],
  type: cc.Prefab
},
slotMoonAnimArray: {
  default: [],
  type: cc.Prefab
},
slotWinArrayPositions: {
  default: [],
  type: cc.Node
},
winSlotPrefab: {
  default: null,
  type: cc.Prefab
},
winningSlotPrefabs: {
  default: [],
  type: a
},
slotWinPrefab: {
  default: null,
  type: cc.Prefab
},
reelStopSlotItems: {
  default: [],
  type: cc.Node
},
suspensNodeArr: {
  default: [],
  type: cc.Node
},
hudController: {
  default: null,
  type: cc.Node
},
mobileLandscapeHudController: {
  default: null,
  type: cc.Node
},
mobilePortraitHudController: {
  default: null,
  type: cc.Node
},
```

```
audioController: {
  default: null,
  type: cc.Node
},
winMessagePreFab: {
  default: null,
  type: cc.Prefab
},
freeSpinMessagePreFab: {
  default: null,
  type: cc.Prefab
},
windowHolder: {
  default: null,
  type: cc.Node
},
winMessageHolder: {
  default: null,
  type: cc.Node
},
freeSpinModeNode: {
  default: null,
  type: cc.Node
},
freeSpinPotraitModeNode: {
  default: null,
  type: cc.Node
},
freeSpinFrameModeNode1: {
  default: null,
  type: cc.Node
},
freeSpinFrameModeNode2: {
  default: null,
  type: cc.Node
},
freeSpinFrameModeNode3: {
  default: null,
  type: cc.Node
},
blackPanel: {
  default: null,
  type: cc.Node
},
HUDButtonPanel: {
  default: null,
  type: cc.Node
},
debugPanel: {
  default: null,
  type: cc.Node
},
```

```
reelSlotMaskArr: {
  default: [],
  type: [cc.Node]
},
gameWinFrame: {
  default: null,
  type: cc.Node
},
slotMask: {
  default: null,
  type: cc.Node
},
moonReelNode: {
  default: null,
  type: cc.Node
},
baseBackground: {
  default: null,
  type: cc.Node
},
mLandscapeBackground: {
  default: null,
  type: cc.Node
},
mPortraitBackground: {
  default: null,
  type: cc.Node
},
slotFrame: {
  default: null,
  type: cc.Node
},
moonSlotFrame: {
  default: null,
  type: cc.Node
},
ShapeShiftSymbolWinFrame: {
  default: null,
  type: cc.Node
},
shiftSymbolPrefabs: {
  default: [],
  type: a
},
slotWinShiftSymbolsArrayPositions: {
  default: [],
  type: cc.Node
},
timeLabelLandscape: {
  default: null,
  type: cc.Label
},
```

```
timeLabelLandscapeMobile: {
  default: null,
  type: cc.Label
},
fgtimeLabelLandscape: {
  default: null,
  type: cc.Label
},
timeLabelPotrait: {
  default: null,
  type: cc.Label
},
timeLabelSettingUI: {
  default: null,
  type: cc.Label
},
settingsBalanceLabel: {
  default: null,
  type: cc.Node
},
settingsBalanceValue: {
  default: null,
  type: cc.Label
},
settingsBetLabel: {
  default: null,
  type: cc.Node
},
settingsBetValue: {
  default: null,
  type: cc.Label
},
_isTurbo: !1,
_isClicked: !1,
_isClickQuickStop: !1,
_isClickTurbo: !1,
_isRealMoney: !1,
_isBigReelChanged: !1,
_bigLogoStartIndex: 0,
_isCommError: !1,
gameFreePlayData: null,
_tapCount: 0,
_timerCheck: null,
_arrTextureFileNames: [],
_selectedScheme: "",
_resetShiftPositions: [],
_isBonusReelStop: !1,
_isStartReelStop: !1,
_prvSelectedReelSet: "",
_savedDataObj: null,
_cumulativeLostLimit: 0,
_cumulativeLWinLimit: 0,
```

```

        _isShiftSymbolPlayed: !1,
        _isSeverError: !1,
        _restoreVersion: 1,
        _isLimitExceed: !1,
        _isMPAudioPlayed: !1,
        _isFirstBonusLandPlayed: !1,
        _isSecondBonusLandPlayed: !1,
        _isGameIdle: !0,
        _sessionTime: 0,
        _realityWinAmount: 0,
        _lossAmount: 0,
        _isRotation: !1,
        _isIpad: !1,
        _timeEnd: 0,
        _timeActivate: 300,
        _timeStart: 0,
        _portraitBgScale: 1,
        _landscapeBgScale: 1,
        _isAutoPlayPaused: !1
    },
    statics: {
        instance: null
    },
    onLoad: function() {
        this._gameEngine = 2, this._SHS_GameGlobal = new o,
this._SHS_GameGlobal.initGameGlobal(), this._winLineDisplayRounds = 0, this._isGameIdle = !
0, this._isIpad = /iPad/.test(navigator.userAgent) && !window.MSStream, this._SHS_GameHelper
= new s, this._SHS_GameHelper.initGameHelper(), this.loadTTFFont(), this._restoreVersion = 1,
l.instance = this, this._SHS_GameEngineManager = new n,
this._SHS_GameEngineManager.assignParam(),
this._SHS_GameEngineManager.initHandler(this._gameEngine, l.instance),
this._SHS_GameEngineManager.OpenPlayRequest(), this.startTime(), this._isClickSplash = !1,
this._isCommError = !1, this._isSeverError = !1, this._isFirstLoad = !0, this._isMixedPaly = !1,
this._isMPAudioPlayed = !1, this._isLimitExceed = !1, this._isFirstBonusLandPlayed = !1,
this._isSecondBonusLandPlayed = !1, this._isRealityCheck = !1, this._frontError = !0,
this._isBounsReelStopPlayed = !1, this._sessionTime = 0, this._realityWinAmount = 0,
this._lossAmount = 0, this._isRotation = !1, this._isFullScreen = !1, this._isAutoPlayPaused = !1,
this._isFreePlayMsgOpen = !1, o.instance.reelSymbolMap = ["A", "B", "C", "D", "E", "F", "G",
"H", "J", "K", "L", "FG"], o.instance.bonusReelSymbolMap = ["RL", "EX", "MP", "BLANK"],
l.instance._autoPlayList = [5, 10, 25, 50, 100], o.instance.rowsToViewMap = [[1, 6, 11, 16, 21], [2,
7, 12, 17, 22], [3, 8, 13, 18, 23], [4, 9, 14, 19, 24], [5, 10, 15, 20, 25]], o.instance.reelToViewMap =
[[1, 6, 11, 16, 21], [2, 7, 12, 17, 22], [3, 8, 13, 18, 23], [4, 9, 14, 19, 24], [5, 10, 15, 20, 25]],
o.instance.selectedReelSet = 53135, o.instance.reelSetArray = {
    53135: 0,
    53335: 1,
    55355: 2,
    55555: 3
}, o.instance.reelPositionsArray = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
18, 19, 20, 21, 22, 23, 24, 25], o.instance.reelDisplayPositionsArray = [14, 13, 12, 11, 10],
o.instance.noPriceSymbolLayout = [["B", "H", "H", "B", "F"], ["mask", "FG", "A", "A", "mask"],
["mask", "mask", "D", "mask", "mask"], ["mask", "K", "D", "D", "mask"], ["A", "A", "A", "FG",
"J"]], this.initSymbolsPool(), this._cumulativeLostLimit = 0, this._cumulativeLWinLimit = 0,

```

```

this._isWinParticlePlay = !1, this._freeSpinCount = 0, this._freeSpinReTriggerIndexes = [];
    cc.sys.isMobile;
    this._audioController = this.audioController.getComponent("SHS_AudioManager"),
this._winCoinMaker =
this.winMessageHolder.getChildByName("CoinMaker").getComponent("SHS_CoinMaker"),
this.loadWinPositionsAndPrefabs();
    var e = !!navigator.userAgent.match(/Version\/[\d\.]+.*Safari/),
    t = /iPad|iPhone|iPod/.test(navigator.userAgent) && !window.MSStream;
    if (this._isSafari = e && t, cc.sys.isMobile) {
        this.mLandscapeBackground.getChildByName("SHS_BG").height;
        var i = this.mLandscapeBackground.getChildByName("SHS_BG").width;
        this._landscapeBgScale =
this.mLandscapeBackground.getChildByName("SHS_BG").scaleX / i * (2 * window.screen.width),
this._portraitBgScale = this.mPortraitBackground.getChildByName("SHS_BG").scaleX / i * (2 *
window.screen.width), this._landscapeBgScale < 1 && (this._landscapeBgScale = 1.2),
this._portraitBgScale < 1 && (this._portraitBgScale = 1.2);
        var a = this.checkOrientation();
        "landscape-primary" == a || "landscape-secondary" == a || -90 == a || 90 == a ?
(this._isIpad ? (this.mLandscapeBackground.getChildByName("SHS_BG").scaleX =
this._landscapeBgScale, this.mLandscapeBackground.getChildByName("SHS_BG").scaleY =
this._landscapeBgScale, this.freeSpinModeNode.getChildByName("SHS_BG").scaleX =
this._landscapeBgScale, this.freeSpinModeNode.getChildByName("SHS_BG").scaleY =
this._landscapeBgScale, this.node.parent.parent.getComponent(cc.Canvas).fitHeight = !1,
this.node.parent.parent.getComponent(cc.Canvas).fitWidth = !0, this._isSafari &&
(this.mLandscapeBackground.getChildByName("SHS_BG").setPositionY(-300),
this.freeSpinModeNode.getChildByName("SHS_BG").setPositionY(50),
this.mobileLandscapeHudController.setPositionY(-70))) :
(this.node.parent.parent.getComponent(cc.Canvas).fitHeight = !0,
this.node.parent.parent.getComponent(cc.Canvas).fitWidth = !1), this._hudController =
this.mobileLandscapeHudController.getComponent("SHS_PSPHudController"),
this.hudController.active = !1, this.mobileLandscapeHudController.active = !0,
this.mobilePortraitHudController.active = !1, this.baseBackground.active = !1,
this.mLandscapeBackground.active = !0, this.mPortraitBackground.active = !1,
this.setRotationType(!1, !0, !1),
l.instance._hudController.settingMobileUI.getChildByName("Paytable_Mobile_Portrait").active = !
1,
l.instance._hudController.settingMobileUI.getChildByName("Paytable_Mobile_Landscape").active
= !0) : "portrait-primary" != a && "portrait-secondary" != a && 0 != a || (this._isIpad ?
(this.mPortraitBackground.getChildByName("SHS_BG").scaleX = this._portraitBgScale,
this.mPortraitBackground.getChildByName("SHS_BG").scaleY = this._portraitBgScale,
this.node.parent.parent.getComponent(cc.Canvas).fitHeight = !1,
this.node.parent.parent.getComponent(cc.Canvas).fitWidth = !0) :
(this.node.parent.parent.getComponent(cc.Canvas).fitHeight = !0, this._isSafari &&
o.instance.fullscreen ? this.node.parent.parent.getComponent(cc.Canvas).fitWidth = !1 : cc.sys.os
== cc.sys.OS_ANDROID && document.webkitIsFullScreen ?
this.node.parent.parent.getComponent(cc.Canvas).fitWidth = !1 :
this.node.parent.parent.getComponent(cc.Canvas).fitWidth = !0), this._hudController =
this.mobilePortraitHudController.getComponent("SHS_PSPHudController"),
this.hudController.active = !1, this.mobileLandscapeHudController.active = !1,
this.mobilePortraitHudController.active = !0, this.baseBackground.active = !1,
this.mLandscapeBackground.active = !1, this.mPortraitBackground.active = !0,
this.setRotationType(!1, !1, !0),

```

```

l.instance._hudController.settingMobileUI.getChildByName("Paytable_Mobile_Portrait").active = !
0,
l.instance._hudController.settingMobileUI.getChildByName("Paytable_Mobile_Landscape").active
= !1, l.instance.slotFrame.scaleX = .75, l.instance.slotFrame.scaleY = .75, this._isSafari ?
(this.mobilePortraitHudController.scaleX = 1.15, this.mobilePortraitHudController.scaleY = 1.15,
this.mobilePortraitHudController.setPositionX(0)) : (this.mobilePortraitHudController.scaleX = 1,
this.mobilePortraitHudController.scaleY = 1)),
l.instance._hudController.settingMobileUI.getComponent("SHS_SettingsController").changeSettin
gUIElementsPositions()
    } else
        this._hudController = this.hudController.getComponent("SHS_PSPHudController"),
this.hudController.active = !0, this.mobileLandscapeHudController.active = !1,
this.mobilePortraitHudController.active = !1, this.baseBackground.active = !0,
this.mLandscapeBackground.active = !1, this.mPortraitBackground.active = !1,
this.setRotationType(!0, !1, !1), this.node.parent.parent.getComponent(cc.Canvas).fitWidth = !0,
this.node.parent.parent.getComponent(cc.Canvas).fitHeight = !0;
    o.instance.minSpinTime < 3e3 ? l.instance._hudController.showTurboButton(!0) :
l.instance._hudController.showTurboButton(!1), this._isTurbo = !1, this._isShiftSymbolPlayed = !
1, this._isLastAutoPlay = !1, this._isMaxBet = !1, this._isPauseAutoPlay = !1, this._isResumeClick
= !1, this._isSpining = !1, this._tapCount = 0, this._timerCheck = null, this._isBonusReelStop = !1,
this._isStartReelStop = !1, this.reelStopSlotItems = [], this._selectedScheme = "", this._reelScript =
[], this._prvSelectedReelSet = "", this._bonusReelScript = [], this._savedDataObj = null,
o.instance.bonusReelLayout = [1, 2, 3], l.instance.resetShiftPositions(), this._bonusReelScript =
this.bonusReel.getComponent("SHS_BonusSlotReel"), this.hideWinningSymbols(),
this.initLocal(), onbodyresize(), this._bigLogoStartIndex = 12, this._timeEnd = 0,
this._timeActivate = 300, this.setUIForNormalModeDifferentDevice(),
this._hudController.EnabledHUDElements(!1), o.instance._isSettingWindowOpen = !1,
l.instance._hudController.singleWinHolder.active = !1
    },
    onDestroy: function() {
        this._SHS_GameEngineManager.gameDestroy(), this._audioController.stopMusic(),
this._audioController.stopAllEffects()
    },
    initLocal: function() {
        o.instance.localSpecifier = o.instance.localSpecifier.toLowerCase(),
o.instance.localSpecifier = o.instance.localSpecifier.replace("_", "-"), "pt-br" !=
o.instance.localSpecifier && (o.instance.localSpecifier = "" == o.instance.localSpecifier ? "en" :
o.instance.localSpecifier.substring(0, 2)), o.instance.localSpecifier = -1 == ["cs", "da", "de", "el",
"es", "et", "fi", "fr", "hu", "it", "lt", "lv", "nl", "nl-be", "no", "pl", "pt", "pt-br", "ro", "ru",
"sv"].indexOf(o.instance.localSpecifier) ? "en" : o.instance.localSpecifier, cc.loader.loadRes("hud/"
+ o.instance.localSpecifier + "/" + o.instance.localSpecifier, cc.SpriteAtlas, function(e, t) {
    if (e)
        throw e;
    o.instance.hud_strings = t
    }), cc.loader.loadRes("paytable/" + o.instance.localSpecifier + "/" +
o.instance.localSpecifier + "_Paytable", cc.SpriteAtlas, function(e, t) {
    if (e)
        throw e;
    o.instance.paytable_strings = t
    }), cc.loader.loadRes("gameruleslist/" + o.instance.localSpecifier + "/" +
o.instance.localSpecifier + "_gameruleslist", cc.SpriteAtlas, function(e, t) {
    if (e)

```

```

        throw e;
        o.instance.gamerules_strings = t
    }), cc.loader.loadRes("auto/" + o.instance.localSpecifier + "/" +
o.instance.localSpecifier + "_autoplay", cc.SpriteAtlas, function(e, t) {
    if (e)
        throw e;
        o.instance.auto_strings = t
    }), cc.loader.loadRes("sound/" + o.instance.localSpecifier + "/" +
o.instance.localSpecifier + "_sound", cc.SpriteAtlas, function(e, t) {
    if (e)
        throw e;
        o.instance.sound_strings = t
    }), cc.loader.loadRes("WinMsg/" + o.instance.localSpecifier + "/" +
o.instance.localSpecifier + "_WinMsg", cc.SpriteAtlas, function(e, t) {
    if (e)
        throw e;
        o.instance.winmsg_strings = t
    }), o.instance._i18n = e("i18n"), o.instance._i18n.init(o.instance.localSpecifier)
},
loadTTFFont: function() {
    cc.loader.loadRes("font/Condensed.ttf", cc.Font, function(e, t) {
        if (e)
            throw e;
            o.instance.font_normal = t
    }), cc.loader.loadRes("font/CondensedBold.ttf", cc.Font, function(e, t) {
        if (e)
            throw e;
            o.instance.font_bold = t
    }), cc.loader.loadRes("font/CondensedW10Bold.ttf", cc.Font, function(e, t) {
        if (e)
            throw e;
            o.instance.font_w10_bold = t
    }), cc.loader.loadRes("font/CondensedW10Regular.ttf", cc.Font, function(e, t) {
        if (e)
            throw e;
            o.instance.font_w10_regular = t
    })
},
hideWinningSymbols: function() {
    var e;
    for (e = 0; e < this.winningSlotPrefabs.length; e++)
        this.winningSlotPrefabs[e] && this.winningSlotPrefabs[e].hideLogos();
    for (e = 0; e < this.shiftSymbolPrefabs.length; e++)
        this.shiftSymbolPrefabs[e] && this.shiftSymbolPrefabs[e].hideLogos();
    for (e = 0; e < this.reelStopSlotItems.length; e++)
        this.reelStopSlotItems[e] && this.reelStopSlotItems[e].setOpacity(255)
},
gameInitializeHandler: function(e) {
    this.node.stopAllActions(), s.instance.gameLog("Game initialize response: " +
JSON.stringify(e));
    var t = e.refresh || e.refresh,
        i = e.limits || e.limits;

```

```

o.instance.gameMode = "REAL" == e.gameMode ? 0 : 1, o.instance.selectedReelSet =
t.gameMode || 53135, l.instance.resetSlotMask(), this.formatReelLayout(t.symbols),
o.instance.noPriceReelLayout = o.instance.reelLayout, o.instance.bonusReelLayout = t.bonusReel ||
[1, 2, 3], o.instance.noPriceBonusReelLayout = t.bonusReel || null;
var n = e.savedSettings || null;
null != n && (l.instance.extractSavedData(JSON.parse(atob(n))), null != t && null !=
t.freespins && t.freespins.length > 0 && !o.instance.isFreeSpinHistory && "FreeSpinEnd" !=
t.restoreState && (o.instance.isHistoryDataAvailable = !0, o.instance.historyData = e)),
o.instance.betPreLineList = i.legalBets, o.instance.payTableInfo = e.config.paytable,
o.instance._scatterPayouts = e.config.scatterPayouts, l.instance._betDivisor = 1, l.instance._baseBet
= e.config.basebet, o.instance.playerBalance = e.stats.b;
var a = i.defaultBet;
o.instance.playerBetAmount = a * l.instance._betDivisor, l.instance._savedDataObj
&& l.instance._savedDataObj.autoPValParams &&
l.instance._savedDataObj.autoPValParams.autoPlayDefault > 0 ? o.instance.autoPlayDefault =
l.instance._savedDataObj.autoPValParams.autoPlayDefault : o.instance.autoPlayDefault = 0,
e.refresh && void 0 !== e.refresh.ba && e.refresh.ba > 0 && (o.instance.playerBetAmount =
e.refresh.ba, a = o.instance.playerBetAmount / l.instance._betDivisor);
for (var r = 0; r < o.instance.betPreLineList.length; r++)
a == o.instance.betPreLineList[r] && (l.instance._hudController._coinValIndex = r,
o.instance._commonCoinValIndex = r, r = o.instance.betPreLineList.length);
if (a === o.instance.betPreLineList[o.instance.betPreLineList.length - 1] &&
(l.instance._isMaxBet = !0), l.instance._hudController.updateBalance(o.instance.playerBalance),
l.instance._hudController.updateBetAmount(o.instance.playerBetAmount),
l.instance._creditsWonThisHand = 0, l.instance._creditsWonWinLines = 0, this._restoreVersion =
1, l.instance._hudController.updateWinAmount(0), this._isCloseHandCall = !1, this._isFirstLoad
= !0, this._freeSpinCount = 0, this._freeSpinReTriggerIndexes = [], this.refreshSlotSymbols(!1),
null != e.stats.freespins && "undefined" !== e.stats.freespins && (e.stats.freespins.spinsended ||
(o.instance.freePlayProObj = e.stats.freespins, o.instance.freePlayPro = !0,
o.instance.fPProSpinsPlayed = 0, l.instance._winAmount = e.stats.freespins.win)),
o.instance.isHistoryDataAvailable && !o.instance.isFreeSpinHistory) {
this._isGameIdle = !1, this.refreshSlotSymbols(!0),
this._hudController.EnabledHUDElements(!1);
var c = e.refresh.win || 0;
l.instance._hudController.updateBalance(o.instance.playerBalance - c)
}
if (o.instance.isFreeSpinHistory && e.refresh) {
if (l.instance._audioController.playMusic(o.instance.MUSIC_FREE_SPIN),
o.instance.isHistoryDataAvailable = !0, l.instance._freeSpinData = e.refresh.freespins,
o.instance.selectedReelSet =
l.instance._freeSpinData[l.instance._currentFreeSpinIndex].gameMode,
o.instance.bonusReelLayout = o.instance.reelLayout, e.refresh.bonus) {
this._selectedScheme = e.refresh.bonus.symbol.toUpperCase();
var d = o.instance.bonusReelSymbolMap.indexOf(this._selectedScheme);
o.instance.bonusReelLayout = [Math.floor(4 * Math.random()) + 1, d + 1,
Math.floor(4 * Math.random()) + 1]
}
l.instance._isFreeSpin = !0, l.instance.resetSlotMask(), o.instance.mPortrait ?
this.freeSpinPotraitModeNode.active = !0 : this.freeSpinModeNode.active = !0,
this.freeSpinFrameModeNode1.active = !0, this.freeSpinFrameModeNode2.active = !0,
this.freeSpinFrameModeNode3.active = !0, l.instance.checkFreeSpinTrigger(), this._isGameIdle = !
1, l.instance._isFreeSpin && !this._isLocalized && o.instance.hud_strings &&

```

```

(l.instance._hudController.lblWin.getComponent(cc.Sprite).spriteFrame =
o.instance.hud_strings.getSpriteFrame(o.instance.localSpecifier + ".total_win")),
this.playFreeSpin()
    } else
        o.instance.isFreeSpinHistory = !1, cc.sys.isMobile &&
l.instance.updateHudElementsInFirstLoad();
    if (o.instance.isFreeSpinHistory && e.refresh) {
        this._isGameIdle = !1;
        c = e.refresh.win || 0;
        l.instance._hudController.updateBalance(o.instance.playerBalance - c)
    }
    this.hideProgressBar(), l.instance._hudController.settingsButton.interactable = !1,
l.instance._hudController._payTableWindow.setupPayLabelIds(),
l.instance._hudController._payTableWindow.refreshPayTable(l.instance._hudController._coinValIn
dex), this._SHS_GameEngineManager.gameSetRealityCheck(0, o.instance.rcInterval,
o.instance.rcElapsed)
    },
    reelSetCompleted: function(e) {
        if (4 == e) {
            var t = this;
            setTimeout(function() {
                t.clickOnSplashScreen()
            }, 40)
        }
    },
    getBetResponseHandler: function(e) {
        s.instance.gameLog("Get Bet response: " + JSON.stringify(e));
        var t = null;
        t = o.instance.isHistoryDataAvailable ? e.refresh : e;
        var i = e.stats;
        if (this.formatReelLayout(t.symbols), o.instance.freeSpinBeforeReel =
o.instance.reelLayout, l.instance._freeSpinData = t.freespins || null, "fun" !=
o.instance.moneyMode) {
            var n = i.realityCheck || null;
            this.initializeRealityCheck(n)
        }
        o.instance.playerBalance = void 0 !== i.b ? i.b : 0, o.instance.playerFreeBalance = void
0 !== i.b ? i.b : 0, l.instance._bonusInfo = t.bonus || e.bonus, l.instance._bonusReelInfo = t.bonus ||
e.bonus, l.instance._winningLines = t.winnings || null, l.instance._winAmount = t.win || 0;
        t.ba;
        s.instance.gameLog("Winning lines: " + l.instance._winningLines), null != i.freespins
&& "undefined" !== i.freespins && (o.instance.freePlayProObj = i.freespins,
o.instance.freePlayPro = !0, o.instance.fProSpinsPlayed = 0, l.instance._creditsWinLineAmount =
l.instance._winAmount, l.instance._winAmount = i.freespins.win), o.instance.freePlayPro ? null ==
i.freespins || "undefined" == i.freespins || i.freespins.spinsended ||
l.instance._hudController.updateBalance(o.instance.playerBalance) :
l.instance._hudController.updateBalance(o.instance.playerBalance - l.instance._winAmount),
o.instance.selectedReelSet = t.gameMode, l.instance.checkFreeSpinTrigger(),
l.instance.preProcessHand(), o.instance.isHistoryDataAvailable || l.instance._isRealityCheck ||
(l.instance._isRelaxDataAvailable = !0, o.instance.minSpinTime < 3e3 ?
l.instance._hudController.showTurboButton(!0) : l.instance._hudController.showTurboButton(!1)),
l.instance._isClicked = !1, l.instance._isClickQuickStop = !1, o.instance.isHistoryDataAvailable = !

```

```

1, this._isSeverError = !1, this._isWinParticlePlay = !1, l.instance._isFreePlayMsgOpen = !1,
o.instance.relaxErrCode = 0
    },
    checkFreeSpinTrigger: function() {
        var e = l.instance._freeSpinData;
        if (this._freeSpinCount = 6, this._freeSpinReTriggerIndexes = [], null !== e)
            for (var t = 0; t < e.length; t++)
                for (var i = 0; i < e[t].winnings.length; i++)
                    "scatter" == e[t].winnings[i][0] && (this._freeSpinReTriggerIndexes[t] = t)
    },
    musicStatusChangeResponseHandler: function(e) {
        e.data ? l.instance._audioController.resumeMusic() :
l.instance._audioController.pauseMusic(), cc.sys.isMobile ?
l.instance._hudController.settingMobileUI.getComponent("SHS_SettingsController").updateSound
UI() :
l.instance._hudController.settingUI.getComponent("SHS_SettingsController").updateSoundUI()
    },
    sfxStatusChangeResponseHandler: function(e) {
        cc.sys.isMobile ?
l.instance._hudController.settingMobileUI.getComponent("SHS_SettingsController").updateSound
UI() :
l.instance._hudController.settingUI.getComponent("SHS_SettingsController").updateSoundUI()
    },
    optionWindowChangeResponseHandler: function(e) {
        l.instance._hudController.showOptionWindow(e)
    },
    payTableChangeResponseHandler: function(e) {
        l.instance._hudController.showPayTable(e)
    },
    autoPlayConfirmationResponseHandler: function(e) {
        null !== l.instance && l.instance._hudController.EnabledHUDElements(!0),
l.instance._isPromotionAvailable && (l.instance._hudController.maxBetButton.interactable = !1)
    },
    autoPlayCancellationResponseHandler: function(e) {
        l.instance._hudController.showStopButton(!1)
    },
    StartSpinning: function() {
        this._isGameIdle = !1, l.instance.resetShiftSymbolsForQuickClick(),
l.instance._isSpining = !0, this._isBounsReelStopPlayed = !1, this._isMPAAudioPlayed = !1,
this._isFirstBonusLandPlayed = !1, this._isSecondBonusLandPlayed = !1,
this._hudController.setSpinButtonState(!1), this._stopAnimation = !1, this._isShiftSymbolPlayed
= !1, this._isAutoPlayPaused = !1, l.instance._hudController.singleWinHolder.active = !1,
this._isFreeSpin || (this._isCloseHandCall = !1), this.hideWinningSymbols(),
this.reelStopSlotItems.splice(0, l.instance.reelStopSlotItems.length), this.node.stopAllActions(),
this._MatchPositions = null, this._winLinesArray = null, this._isBonusReelStop = !1,
this._isStartReelStop = !1, this._ShowWinLine = !0, this._isWinParticlePlay = !1,
this._creditsWonThisHand = 0, this._creditsWinLineAmount = 0, o.instance.freePlayPro ||
(this._hudController.winAnimationBox.getComponent(cc.Animation).play("Win2Balance_Dissapp
ear"), this._hudController.updateWinAmount(0)), this.refreshSlotSymbols(!0),
this._cumulativeLostLimit = this._cumulativeLostLimit + o.instance.playerBetAmount,
this._SHS_GameEngineManager.sendPlayRequest("spin", o.instance.playerBetAmount,
"SpinStart"), l.instance.resetShiftPositions(), o.instance.freePlayPro &&

```

```

(o.instance.fPProSpinsPlayed++, l.instance._hudController.showHudMessages()), this._isAutoPlay
&& (this._hudController.showStopButton(!0), -1 != this._autoPlayCounter &&
this._autoPlayCounter--, this._hudController.setSpinCounter(this._autoPlayCounter, 2),
this._autoPlayCounter <= 0 && (this._hudController.showStopButton(!1), this._isAutoPlay = !1,
this._isLastAutoPlay = !0, this._cumulativeLWinLimit = 0)), this._isFreeSpin ||
o.instance.freePlayPro || (this._hudController.lblMessageNode.active = !0,
this._hudController.lblMoonMessageNode.active = !1)
    },
    refreshSlotSymbols: function(e) {
        this._isRelaxDataAvailable = !1;
        for (var t = 0; t < this.reelsArray.length; t++)
            e ? (this._hudController.EnabledHUDElements(!1),
this._reelScript[t].startSpinning(0, t)) : (this._reelScript[t].setSlotItemsInReel(t, !e),
this._reelScript[t].initReel());
        this.playBonusReel(e, !1)
    },
    playBonusReel: function(e, t) {
        var i = this,
            n = [];
        n.push(cc.callFunc(function() {
            t || (e ? i._bonusReelScript.startSpinning(0, 0) :
(i._bonusReelScript.setSlotItemsInReel(0, !e), i._bonusReelScript.initReel()))
        })), o.instance.minSpinTime >= 3e3 ? n.push(cc.delayTime(.4)) :
n.push(cc.delayTime(.1)), n.push(i.stopBonusReel()), n.push(cc.delayTime(.3)),
l.instance._isSpining && !l.instance._isBounsReelStopPlayed && n.push(cc.callFunc(function() {
            l.instance._audioController.playReelEndSound(0),
l.instance._isBounsReelStopPlayed = !0
        })), n.push(cc.delayTime(.2)), n.push(cc.callFunc(function() {
            i._isBonusReelStop && i.resetSlotMask()
        })), n.push(cc.delayTime(.1)), n.push(cc.callFunc(function() {
            i._isBonusReelStop && (i._isStartReelStop = !0)
        })), this.node.runAction(cc.sequence(n))
    },
    finishedReelSpinning: function(e) {
        s.instance.gameLog("Finish spinning on ID: " + e), 1 == e &&
l.instance._hudController.showTurboButton(!1), 4 == e && (l.instance._isSpining = !1,
this.postReelSpinAction())
    },
    onPostReelSpinActionCompleted: function() {
        this.showWinMessage()
    },
    onWinMessageCompleted: function() {
        var e = o.instance._i18n.t("HUD_MESSAGES.TRIGGER_6_FREE_SPINS"),
            t = "";
        "EX" == this._selectedScheme ? (e =
o.instance._i18n.t("HUD_MESSAGES.EXTEND_MATRIX"), t = "EXTEND_MATRIX") : "RL"
== this._selectedScheme ? (e =
o.instance._i18n.t("HUD_MESSAGES.PAY_BOTH_WAYS_MATRIX"), t =
"PAY_BOTH_WAYS_MATRIX") : "MP" == this._selectedScheme && (e =
o.instance._i18n.t("HUD_MESSAGES.ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX
"), t = "ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX"), "" == t ||
o.instance.freePlayPro || l.instance._hudController.updateWinLineMoonMessage(e, t),

```

```

this.showWinLines()
    },
    onWinLineCompleted: function() {
        this._isWinLineFinished || (this._isWinLineFinished = !0,
this.showFreeSpinMessage())
    },
    onFreeSpinMessageCompleted: function() {
        this.finalizePlayerHand()
    },
    finalizePlayerHand: function() {
        if (l.instance._stopAnimation = !1, l.instance._isClicked = !1, null !==
l.instance._freeSpinData)
            l.instance._currentFreeSpinIndex < l.instance._freeSpinData.length ?
l.instance._isAutoPlayPaused ||
(this._SHS_GameEngineManager.saveFreeSpin(this._currentFreeSpinIndex,
this._displayFreeSpinCount), setTimeout(function() {
                l.instance.playFreeSpin()
            }, 200)) : (l.instance._displayFreeSpinCount = 0,
this._SHS_GameEngineManager.setRestoreState("FreeSpinEnd"), this.finishedFreeSpin());
        else if (l.instance.finishCurrentHand(), o.instance.showFreeProMsgAfterHistoryHand)
        {
            var e = o.instance.freePlayProObj.spins;
            l.instance._hudController.updateFreePlayMessage("You have been awarded " + e + "
free spins"), o.instance.showFreeProMsgAfterHistoryHand = !1
        }
    },
    finishedFreeSpin: function() {
        l.instance._hudController.setFreeSpinCounter(0);
        var e = [];
        e.push(cc.callFunc(function() {})), e.push(cc.callFunc(function() {
            l.instance.postFreeSpinMessage(!0), l.instance._ShowWinLine = !1,
l.instance.hideWinningSymbols()
        })), e.push(cc.callFunc(function() {
            l.instance._audioController.playSummaryEffect()
        })), e.push(cc.callFunc(function() {
            l.instance._audioController.playSummarySFXEffect()
        })), e.push(cc.callFunc(function() {
            l.instance.reSetReelToBonusStartup()
        })), e.push(cc.delayTime(6)), e.push(cc.callFunc(function() {
            l.instance.postFreeSpinMessage(!1)
        })), e.push(cc.callFunc(function() {
            l.instance.finishCurrentHand()
        })), this.node.runAction(cc.sequence(e))
    },
    postReelSpinAction: function() {
        var e = this,
            t = [];
        if (t.push(cc.callFunc(function() {})), this._isFreeSpinFirstTime) {
            e._hudController.EnabledHUDElements(!1);
            var i = [];
            o.instance.freeSpinBeforeSelectedReelSet = o.instance.selectedReelSet, i =
i.concat(this._reelScript[0]._bonusPos, this._reelScript[1]._bonusPos,

```

```

this._reelScript[2]._bonusPos, this._reelScript[3]._bonusPos, this._reelScript[4]._bonusPos),
t.push(cc.callFunc(function() {
    e._SHS_GameEngineManager.setRestoreState("FreeSpinStart"),
o.instance.freeSpinBeforeBonusReelLayout = o.instance.bonusReelLayout
    })), t.push(cc.delayTime(.2)), t.push(e.showFreeSpinBonusEffect(i)),
t.push(cc.callFunc(function() {
    l.instance._audioController.playAudioBonusAnimate()
    })), t.push(cc.delayTime(1))
    }
    var n = cc.callFunc(e.onPostReelSpinActionCompleted, e);
    t.push(n), this.node.runAction(cc.sequence(t))
    },
    showFreeSpinBonusEffect: function(e) {
        var t = this;
        return cc.callFunc(function() {
            o.instance.reelSetArray[o.instance.selectedReelSet];
            for (var i = 0; i < e.length; i++) {
                var n = e[i];
                t.winningSlotPrefabs[n - 1].showBonusWinLogo(o.instance.reelLayout[n - 1])
            }
        })
    },
    hideFreeSpinBonusSymbol: function(e) {
        for (var t = 0; t < e.length; t++) {
            var i = e[t];
            this.winningSlotPrefabs[i - 1].hideLogos()
        }
    },
    showWinMessage: function() {
        var e = this,
            t = [];
        t.push(cc.callFunc(function() {})), this._countdownLevel = 0;
        if (!(null !== l.instance._freeSpinData && l.instance._currentFreeSpinIndex > 0 &&
l.instance._currentFreeSpinIndex - 1 < l.instance._freeSpinData.length) ||
l.instance._isFreeSpinFirstTime && l.instance._isFreeSpinRetrigger)
            if (null !== l.instance._winningLines && l.instance._winningLines.length > 0) {
                if (l.instance._isFreeSpin) {
                    i = l.instance._freeSpinData[0].runningTotal;
                    l.instance._creditsWonThisHand = i
                } else
                    l.instance._creditsWonThisHand += l.instance._winAmount;
                if (l.instance._creditsWonWinLines = l.instance._creditsWonThisHand, !
l.instance._isFreeSpin && l.instance._creditsWonThisHand / o.instance.playerBetAmount >= 10)
                    return t.push(cc.callFunc(function() {
                        e.showFireBallWinMessage(!0)
                    })), t.push(cc.delayTime(.01)), t = t.concat(this.postWinMessageActionEx(100,
0, !1)), void this.node.runAction(cc.sequence(t));
                !0
            } else
                o.instance.freePlayPro && (l.instance._creditsWonThisHand =
l.instance._winAmount);
            else {

```

```

        for (var i = l.instance._freeSpinData[l.instance._currentFreeSpinIndex -
1].runningTotal, n = 0, a = l.instance._freeSpinData[l.instance._currentFreeSpinIndex - 1].winnings,
s = 0; s < a.length; s++)
            n = parseFloat(n) + parseFloat(a[s][2]);
        var r = i + n;
        if (l.instance._creditsWinLineAmount = n, l.instance._creditsWonThisHand = r, n /
o.instance.playerBetAmount >= 10)
            return t.push(cc.callFunc(function() {
                e.showFireBallWinMessage(!0)
            })), t.push(cc.delayTime(.01)), t = t.concat(this.postWinMessageActionEx(100,
0, !1)), void this.node.runAction(cc.sequence(t))
        }
        var c = Math.pow(10, o.instance.gameCurrencyData.decimalPlaces);
        l.instance._creditsWonThisHand = Math.floor(l.instance._creditsWonThisHand * c) /
c, l.instance._TotalLoss += o.instance.playerBetAmount - l.instance._creditsWonThisHand;
        var d = [];
        d.push(cc.callFunc(function() {})), d.push(cc.callFunc(function() {
            e.winAmountUpdateAction()
        })), this.node.runAction(cc.sequence(d))
    },
    postWinMessageActionEx: function(e, t, i) {
        var n = this;
        n._isCoinMsgDisplay = !0;
        var a = [];
        return a.push(cc.callFunc(function() {
            n._audioController.playWinBackgroundEffect(o.instance.WIN_LOOP_START),
n._audioController.pauseMusic(), n.postWinMessageAction(e, t, i)
        })), a.push(cc.delayTime(.5)), a.push(cc.callFunc(function() {
            n._audioController.playWinBackgroundEffect(o.instance.WIN_LOOP)
        })), a
    },
    postWinMessageAction: function(e, t, i) {
        var n = l.instance._creditsWonThisHand;
        if ((l.instance._isFreeSpin || o.instance.freePlayPro) && (n =
l.instance._creditsWinLineAmount), l.instance._countdownLevel < n) {
l.instance.winMessageHolder.getChildByName("lblWinAmount").getComponent(cc.Label).string
= s.instance.ToCurrency(l.instance._countdownLevel);
            var a = l.instance._countdownLevel / o.instance.playerBetAmount,
                r = 10,
                c = o.instance._i18n.t("MSG.BIG_WIN");
            switch (!0) {
                case a >= 50:
                    r = 50, c = o.instance._i18n.t("MSG.MEGA_WIN");
                    break;
                case a >= 30:
                    r = 30, c = o.instance._i18n.t("MSG.EPIC_WIN");
                    break;
                case a >= 20:
                    r = 20, c = o.instance._i18n.t("MSG.HUGE_WIN");
                    break;
                case a >= 10:

```

```

    r = 10, c = o.instance._i18n.t("MSG.BIG_WIN")
  }
  if (t != r)

```

```

l.instance.winMessageHolder.getChildByName("lblWinType").getComponent(cc.Label).string = c;
  l.instance._countdownLevel += n > 1e3 ? Math.ceil(.005 * n + .01) : 1.02,
l.instance._countdownLevel = Math.ceil(l.instance._countdownLevel), setTimeout(function() {
  l.instance.postWinMessageAction(e < 25 ? 25 : --e, r, i)
}, e)
} else {
  var d = [];
  d.push(cc.callFunc(function() {})), d.push(cc.callFunc(function() {

```

```

l.instance._audioController.playWinBackgroundEffect(o.instance.WIN_LOOP_END),
l.instance.winMessageHolder.getChildByName("lblWinAmount").getComponent(cc.Label).string
= s.instance.ToCurrency(n)

```

```

  })), d.push(cc.delayTime(.2)), d.push(cc.callFunc(function() {
  l.instance._isFreeSpin ?

```

```

l.instance._audioController.playMusic(o.instance.MUSIC_FREE_SPIN) :

```

```

l.instance._audioController.playMusic(o.instance.MUSIC_IDLE)
  })), d.push(cc.delayTime(1)), d.push(cc.callFunc(function() {
  l.instance.showFireBallWinMessage(!1)
  })), d.push(cc.delayTime(1)), i ? d.push(cc.callFunc(function() {
  l.instance.finishedFreeSpin()
  })): d.push(cc.callFunc(function() {
  l.instance.winAmountUpdateAction()
  })), l.instance.node.runAction(cc.sequence(d))
}

```

```

},
showSingleWinAction: function() {
  var e = l.instance._creditsWonThisHand;
  (l.instance._isFreeSpin || o.instance.freePlayPro) && (e =

```

```

l.instance._creditsWinLineAmount) > 0 && (l.instance._hudController.singleWinHolder.active = !
0,

```

```

l.instance._hudController.singleWinHolder.getChildByName("singleWinAmount").getComponent(
cc.Label).string = o.instance._i18n.t("MSG.WIN_AMOUNT") + " " + s.instance.ToCurrency(e))
},

```

```

winAmountUpdateAction: function() {
  var e = [];
  e.push(cc.callFunc(function() {})), e.push(cc.callFunc(function() {})),

```

```

l.instance._creditsWonThisHand > 0 && e.push(cc.delayTime(.01));
  var t = cc.callFunc(this.onWinMessageCompleted, this);
  e.push(t), this.node.runAction(cc.sequence(e))
},

```

```

showWinAmount: function() {
  return cc.callFunc(function() {

```

```

  l.instance._hudController.updateWinAmount(l.instance._creditsWonThisHand),
l.instance._isFreeSpin || o.instance.freePlayPro ? (null != l.instance._winningLines &&
l.instance._winningLines.length > 0 && !l.instance._isWinParticlePlay &&
l.instance._audioController.playWinLineEffect(), l.instance._isWinParticlePlay = !0) :
l.instance._isWinParticlePlay || (l.instance._hudController.playWinParticle(),
l.instance._isWinParticlePlay = !0, l.instance._creditsWonThisHand > 0 &&

```

```

l.instance._audioController.playWinLineEffect());
    var e = !1;
    l.instance._creditsWinLineAmount < 1 && l.instance._isFreeSpin && (e = !0), (1 !=
l.instance._creditsWonThisHand || e) && l.instance._creditsWonThisHand
    })
    },
    showFreeSpinMessage: function() {
        var e = l.instance,
            t = [];
        t.push(cc.callFunc(function() {})), (0 === this._currentFreeSpinIndex && null !==
this._freeSpinData || this._isFreeSpinRetrigger) && (t.push(cc.callFunc(function() {
            e._audioController.playTransitionEffect()
        })), t.push(cc.callFunc(function() {
            e.preFreeSpinMessage(!0)
        })), t.push(cc.callFunc(function() {
            e._audioController.playMusic(o.instance.MUSIC_FREE_SPIN)
        })), l.instance._isFreeSpinRetrigger && (t.push(cc.delayTime(5)),
t.push(cc.callFunc(function() {
            e.preFreeSpinMessage(!1)
        }))))), this._isFreeSpin && 0 === this._currentFreeSpinIndex ||
l.instance._isFreeSpinRetrigger ? t.push(cc.callFunc(function() {})) : t.push(cc.callFunc(function()
{
    l.instance.onFreeSpinMessageCompleted()
    })), this.node.runAction(cc.sequence(t))
    },
    playFreeSpin: function() {
        var e = l.instance._freeSpinData[l.instance._currentFreeSpinIndex];
        l.instance._isTurbo = !1, o.instance._isAutoOffInFreeSpin ? l.instance._isAutoPlay = !
1 : l.instance._isAutoPlay ? l.instance._isAutoPlay = !0 : l.instance._isAutoPlay = !1,
l.instance._hudController.singleWinHolder.active = !1, l.instance._isFirstLoad = !1,
l.instance._isMPAAudioPlayed = !1, l.instance._isFirstBonusLandPlayed = !1,
l.instance._isSecondBonusLandPlayed = !1, this._isWinParticlePlay = !1,
l.instance.resetShiftPositions(), 0 == l.instance._currentFreeSpinIndex &&
(o.instance.selectedReelSet = 55555, l.instance.resetSlotMask(),
l.instance._freeSpinBeforeWinningLines = l.instance._winningLines), null != e &&
(l.instance._displayFreeSpinCount--,
l.instance._hudController.setFreeSpinCounter(l.instance._displayFreeSpinCount),
l.instance._displayFreeSpinCount > 1 || l.instance._displayFreeSpinCount, null != e.symbols &&
l.instance.formatReelLayout(e.symbols), l.instance._winningLines = e.winnings || null,
l.instance._bonusInfo = e.bonus, l.instance._bonusReelInfo = e.bonus,
l.instance.hideWinningSymbols(), l.instance.node.stopAllActions(), l.instance._MatchPositions =
null, l.instance._winLinesArray = null, this._isShiftSymbolPlayed = !1,
l.instance.refreshSlotSymbols(!0), l.instance.preProcessHand(), l.instance._isFreeSpinFirstTime = !
1, o.instance.isFreeSpinHistory ? l.instance._isRelaxDataAvailable = !1 :
l.instance._isRelaxDataAvailable = !0, l.instance._currentFreeSpinIndex++,
l.instance._isPlayIdleMusic = !(l.instance._currentFreeSpinIndex <
l.instance._freeSpinData.length))
    },
    finishCurrentHand: function() {
        l.instance._isBigReelChanged = !1;
        var e = !1;
        if (o.instance.freePlayPro && o.instance.fPProSpinsPlayed >=

```

```

o.instance.freePlayProObj.spins && (this.freeRoundPromoFinishMsg(), e = !0), this._freeSpinData
= null, this._winningLines = null, this._bonusInfo = null, this._bonusReelInfo = null,
this._isGameIdle = !0, this.resetShiftPositions(), this._isFirstLoad || this._isAutoPlay || e ||
l.instance._isFreePlayMsgOpen || this._hudController.EnabledHUDElements(!0),
this._SHS_GameEngineManager.saveFreeSpin(-1, -1),
this._SHS_GameEngineManager.gameFinished(), this._isFreeSpin = !1, this._freeSpinCount = 0,
this._currentFreeSpinIndex = 0, l.instance._isAutoPlay && !o.instance.isBlockedUI) {
    var t = !1,
        i = this._winLimit * o.instance.playerBetAmount,
        n = this._lossLimit * o.instance.playerBetAmount;
    if (o.instance.freePlayPro || (i < l.instance._creditsWonThisHand ? (t = !0,
this._hudController.changeErrorButton(o.instance._i18n.t("BUTTON.OK")),
this._hudController.updateErrorMessage(o.instance._i18n.t("AUTOPLAY.WIN_LIMIT_EXCEED
")) : n <= this._cumulativeLostLimit - this._cumulativeLWinLimit && (t = !0,
this._hudController.changeErrorButton(o.instance._i18n.t("BUTTON.OK")),
this._hudController.updateErrorMessage(o.instance._i18n.t("AUTOPLAY.LOSS_LIMIT_EXCEE
D"))), t && (this._isLimitExceed = t, this._hudController.showStopButton(!1), this._isAutoPlay = !
1, this._isLastAutoPlay = !0, this._cumulativeLWinLimit = 0, this._cumulativeLostLimit = 0)),
l.instance._autoPlayCounter > 0 && !l.instance._isAutoPlayPaused) {
        var a = l.instance._creditsWonThisHand > 0 ? 1e3 : 800;
        o.instance.minSpinTime < 3e3 && (a = 0 == l.instance._creditsWonThisHand ?
200 : 1e3), setTimeout(function() {
            l.instance.StartSpinning()
        }, a)
    }
},
processFreeSpin: function() {
    var e = this._freeSpinData;
    return null === l.instance._freeSpinData || l.instance._isFreeSpin ? (-1 !=
this._freeSpinReTriggerIndexes.indexOf(this._currentFreeSpinIndex) &&
(l.instance._isFreeSpinRetrigger = !0, this._freeSpinCount = 6), !0) : null !== e ?
(l.instance._isFreeSpin = !0, l.instance._isFreeSpinFirstTime = !0, l.instance._isPlayIdleMusic = !1,
o.instance.freeSpinBeforeReel = o.instance.reelLayout, !0) : (l.instance._isPlayIdleMusic = !0, !1)
},
showWinLines: function() {
    var e = l.instance,
        t = !0,
        i = 0;
    l.instance._MatchPositions = [], l.instance._winLinesArray = [];
    var n = [];
    if (n.push(cc.callFunc(function() {
        "MP" == e._selectedScheme ? e.shiftMpSymbols() : "RL" == e._selectedScheme
&& (t = !1)
    })), "MP" == e._selectedScheme && (e._isAutoPlay || e._isFreeSpin ||
l.instance._isFirstLoad || (o.instance.minSpinTime >= 3e3 && n.push(cc.delayTime(1)),
n.push(cc.callFunc(function() {
        e._hudController.EnabledHUDElements(!0)
    }))), n.push(cc.delayTime(2)), t = !1), (e._isAutoPlay || e._isFreeSpin ||
l.instance._isFirstLoad || o.instance.freePlayPro) && (t = !1), t && (o.instance.minSpinTime >=
3e3 ? n.push(cc.delayTime(.8)) : n.push(cc.delayTime(.05)), n.push(cc.callFunc(function() {
        e._hudController.EnabledHUDElements(!0)
    })))
}

```

```

    })), null == l.instance._winningLines || null === l.instance._winningLines)
    ;
    else {
        n.push(cc.callFunc(function() {
            e.showSingleWinAction()
        })), n.push(e.showWinAmount()), i = l.instance._winningLines.length,
l.instance._stackAnimationNotShown = !0;
        for (var a = 0; a < l.instance._winningLines.length; a++) {
            var s = l.instance._winningLines[a],
                r = [],
                c = o.instance.reelSymbolMap.indexOf(s[4]);
            if ("scatter" != s[0] || !e._isFreeSpin) {
                for (var d in s[3]) {
                    var u = s[3][d];
                    r.push(o.instance.reelToViewMap[u.x - 1][u.y - 1])
                }
                l.instance._winLinesArray[a] = {
                    symId: c,
                    matchPos: r,
                    lineData: s,
                    coinValIndex: l.instance._hudController._coinValIndex
                }
            }
        }
        if (null !== l.instance._winLinesArray)
            for (var S = 0; S < l.instance._winLinesArray.length; S++)
                n.push(e.hidePreviousWinSymbols()),
n.push(e.showWinLine(l.instance._winLinesArray[S])), n.push(cc.delayTime(1.5));
        "MP" == e._selectedScheme && l.instance._winningLines.length < 2 &&
(this.ShapeShiftSymbolWinFrame.active = !0), l.instance._isFreeSpin || o.instance.freePlayPro ||
n.push(cc.callFunc(function() {
            l.instance._hudController.updateBalance(o.instance.playerBalance)
        })))
    }
    var E = cc.callFunc(e.onWinLineCompleted, e);
    n.push(E), e._isWinLineFinished = !1;
    var _ = cc.sequence(n);
    i > 1 ? e.node.runAction(cc.repeatForever(_)) : e.node.runAction(_)
},
shiftMpSymbols: function() {
    if (!this._isShiftSymbolPlayed) {

l.instance._hudController.updateWinLineMessage(o.instance._i18n.t("HUD_MESSAGES.IS_ACTIVE_ALL"), "IS_ACTIVE_ALL");
        for (var e = 0; e < o.instance.reelLayout.length; e++)
            2 != o.instance.reelLayout[e] && 1 != o.instance.reelLayout[e] && 3 !=
o.instance.reelLayout[e] || (l.instance._isMPAudioPlayed || (l.instance._isMixedPaly ?
l.instance._audioController.playMixedShapeShiftAudio() :
l.instance._audioController.playShapeShiftAudio(o.instance.reelLayout[e]),
l.instance._isMPAudioPlayed = !0),
l.instance.winningSlotPrefabs[e].showShiftWinLogo(o.instance.reelLayout[e], !1),
l.instance._isAutoPlay || l.instance._isFreeSpin ||

```

```

l.instance.shiftSymbolPrefabs[e].showShiftWinLogoNormalSpin(o.instance.reelLayout[e], !1));
    if (l.instance._resetShiftPositions.length > 0)
        for (var t = 0; t < l.instance._resetShiftPositions.length; t++)
            l.instance._resetShiftPositions[t].length > 0 &&
l.instance._reelScript[t].reSetDisplayPositions(l.instance._resetShiftPositions[t]);
    var i = [];
    i = i.concat(this._reelScript[0]._bonusPos, this._reelScript[1]._bonusPos,
this._reelScript[2]._bonusPos, this._reelScript[3]._bonusPos, this._reelScript[4]._bonusPos),
this.hideFreeSpinBonusSymbol(i), l.instance._isMixedPaly = !1
    }
},
showWinLine: function(e) {
    var t = e.matchPos,
        i = (e.symId, this);
    return cc.callFunc(function() {
        if (i._ShowWinLine && (i._winLineDisplayRounds < 4 || i._isFreeSpin ||
i._isAutoPlay)) {
            if (null !== i.reelStopSlotItems && null !== o.instance.reelLayout) {
                for (var e = 0; e < i.reelStopSlotItems.length; e++)
                    i.reelStopSlotItems[e] && i.reelStopSlotItems[e].setOpacity(100);
                o.instance.reelSetArray[o.instance.selectedReelSet];
                for (var n = 0; n < t.length; n++) {
                    var a = t[n];
                    i.reelStopSlotItems[a - 1], i.winningSlotPrefabs[a - 1] && ("MP" !==
i._selectedScheme || 2 !== o.instance.reelLayout[a - 1] && 1 !== o.instance.reelLayout[a - 1] && 3 !==
o.instance.reelLayout[a - 1] ? (2 !== o.instance.reelLayout[a - 1] && 1 !== o.instance.reelLayout[a -
1] && 3 !== o.instance.reelLayout[a - 1] || i.winningSlotPrefabs[a -
1].resetWinLogo(o.instance.reelLayout[a - 1]), i.winningSlotPrefabs[a -
1].showWinLogo(o.instance.reelLayout[a - 1], !0)) : l.instance.winningSlotPrefabs[a -
1].showWinLogo(o.instance.reelLayout[a - 1] + 12, !0)), i._isWinLineFinished
                }
                i._isShiftSymbolPlayed = !0
            }
            "MP" === i._selectedScheme && i._winLineDisplayRounds++
        }
        "MP" === i._selectedScheme && i._winLineDisplayRounds > 3 && !i._isFreeSpin
&& !i._isAutoPlay && (i.ShapeShiftSymbolWinFrame.active = !0)
    })
},
restShiftSymbolsForQuickClick: function() {
    for (var e = 0; e < o.instance.reelLayout.length; e++) {
        var t = e;
        this.winningSlotPrefabs[t] && (2 !== o.instance.reelLayout[t] && 1 !==
o.instance.reelLayout[t] && 3 !== o.instance.reelLayout[t] ||
this.winningSlotPrefabs[t].resetWinLogo(o.instance.reelLayout[t]))
    }
},
showBonusSymbol: function(e) {
    var t = e.matchPos,
        i = e.symId;
    o.instance.reelSetArray[o.instance.selectedReelSet];
    if (null !== this.reelStopSlotItems && null !== o.instance.reelLayout) {

```

```

        for (var n = 0; n < t.length; n++) {
            var a = t[n];
            this.winningSlotPrefabs[a - 1].showWinLogo(o.instance.reelLayout[a - 1], !0)
        }
        l.instance._audioController.playSymbolSound(i)
    }
},
hidePreviousWinSymbols: function() {
    var e,
        t = this;
    return cc.callFunc(function() {
        if (null != t.reelStopSlotItems) {
            for (e = 0; e < t.reelStopSlotItems.length; e++)
                t.reelStopSlotItems[e] && t.reelStopSlotItems[e].setOpacity(255);
            for (e = 0; e < t.winningSlotPrefabs.length; e++)
                t.winningSlotPrefabs[e] && t.winningSlotPrefabs[e].hideLogos()
        }
    })
},
setAutoPlay: function(e, t, i) {
    0 === e ? (l.instance._hudController.setSpinCounter(0, -1), this._isAutoPlay = !1,
this._lossLimit = t, this._winLimit = i, this._autoPlayCounter = e, this._TotalLoss = 0) :
(this._TotalLoss = 0, o.instance.autoPlayDefault = e, this._isAutoPlay = !0, this._lossLimit = t,
this._winLimit = i, this._autoPlayCounter = e,
l.instance._hudController.setSpinCounter(this._autoPlayCounter, 2),
this._SHS_GameEngineManager.saveAutoPlayValue(e, this._winLimit, this._lossLimit))
},
setBetValue: function(e) {
    o.instance.playerBetAmount = e * this._betDivisor,
l.instance._hudController.updateBetAmount(o.instance.playerBetAmount),
l.instance._hudController.updateCoinValue(e)
},
preProcessHand: function() {
    if (this._selectedScheme = "", this._isFirstLoad = !1, l.instance.processFreeSpin(),
null != l.instance._bonusReelInfo) {
        this._selectedScheme = l.instance._bonusReelInfo.symbol.toUpperCase();
        var e = o.instance.bonusReelSymbolMap.indexOf(this._selectedScheme);
        o.instance.bonusReelLayout = [Math.floor(4 * Math.random()) + 1, e + 1,
Math.floor(4 * Math.random()) + 1]
    }
    var t = [!1, !1, !1, !1, !1];
    this._reelScript[0]._bonusPos = [], this._reelScript[1]._bonusPos = [],
this._reelScript[2]._bonusPos = [], this._reelScript[3]._bonusPos = [], this._reelScript[4]._bonusPos
= [];
    for (var i = 0; i < o.instance.reelLayout.length; i++)
        if (12 == o.instance.reelLayout[i]) {
            var n = i + 1 == 1 || i + 1 == 6 || i + 1 == 11 || i + 1 == 16 || i + 1 == 21,
                a = i + 1 == 2 || i + 1 == 7 || i + 1 == 12 || i + 1 == 17 || i + 1 == 22,
                s = i + 1 == 3 || i + 1 == 8 || i + 1 == 13 || i + 1 == 18 || i + 1 == 23,
                r = i + 1 == 4 || i + 1 == 9 || i + 1 == 14 || i + 1 == 19 || i + 1 == 24;
            t[0] = n || a, t[1] = t[1] || s, t[2] = t[2] || r, this._reelScript[n ? 0 : a ? 1 : s ? 2 : r ? 3 :
4]._bonusPos.push(i + 1)
        }
}

```

```

    }
    this._reelScript[0]._bonusSFXId = this._reelScript[0]._bonusPos.length,
this._reelScript[1]._bonusSFXId = this._reelScript[1]._bonusPos.length,
this._reelScript[2]._bonusSFXId = this._reelScript[2]._bonusPos.length,
this._reelScript[3]._bonusSFXId = this._reelScript[3]._bonusPos.length,
this._reelScript[4]._bonusSFXId = this._reelScript[4]._bonusPos.length;
    var c = this._reelScript[0]._bonusSFXId > 0 && this._reelScript[1]._bonusSFXId > 0,
        d = this._reelScript[1]._bonusSFXId > 0 && this._reelScript[2]._bonusSFXId > 0,
        u = this._reelScript[0]._bonusSFXId > 0 && this._reelScript[2]._bonusSFXId > 0,
        S = this._reelScript[0]._bonusSFXId > 0 && this._reelScript[3]._bonusSFXId > 0,
        E = this._reelScript[1]._bonusSFXId > 0 && this._reelScript[3]._bonusSFXId > 0,
        _ = this._reelScript[2]._bonusSFXId > 0 && this._reelScript[3]._bonusSFXId > 0,
        p = this._reelScript[0]._bonusSFXId > 0 && this._reelScript[1]._bonusSFXId > 0
&& this._reelScript[2]._bonusSFXId > 0,
        A = this._reelScript[0]._bonusSFXId > 0 && this._reelScript[1]._bonusSFXId > 0
&& this._reelScript[3]._bonusSFXId > 0,
        m = this._reelScript[0]._bonusSFXId > 0 && this._reelScript[2]._bonusSFXId > 0
&& this._reelScript[3]._bonusSFXId > 0,
        h = this._reelScript[1]._bonusSFXId > 0 && this._reelScript[2]._bonusSFXId > 0
&& this._reelScript[3]._bonusSFXId > 0,
        I = (c || u || d || S || E || _) && !(p || A || m || h);
    o.instance.minSpinTime >= 3e3 ? (this._reelScript[2]._spinCount += c ? 2 : 0,
this._reelScript[2]._isSuspense = c, this._reelScript[3]._isSuspense = (c || d || u) && !p,
this._reelScript[3]._spinCount += (c || d || u) && !p ? 3 : this._reelScript[2]._isSuspense ? 2 : 0,
this._reelScript[4]._spinCount += I ? 4 : this._reelScript[3]._isSuspense ? 3 :
this._reelScript[2]._isSuspense ? 2 : 0) : (this._reelScript[2]._spinCount += c ? 10 : 0,
this._reelScript[2]._isSuspense = c, this._reelScript[3]._isSuspense = (c || d || u) && !p,
this._reelScript[3]._spinCount += (c || d || u) && !p ? this._reelScript[2]._spinCount + 5 :
this._reelScript[2]._isSuspense ? 10 : 0, this._reelScript[4]._spinCount += I ?
this._reelScript[3]._spinCount : this._reelScript[3]._isSuspense ? this._reelScript[2]._spinCount +
5 : this._reelScript[2]._isSuspense ? 10 : 0, this._reelScript[3]._isSuspense && I &&
(this._reelScript[4]._spinCount = this._reelScript[3]._spinCount + 10)),
this._reelScript[4]._isSuspense = I, this._winLineDisplayRounds = 0,
this.ShapeShiftSymbolWinFrame.active = !1
    },
    fierceFactorOnClick: function() {
        this.windowHolder.removeAllChildren(!0);
        var e = cc.instantiate(this.ffPreFab);
        this.windowHolder.addChild(e)
    },
    preFreeSpinMessage: function(e) {
        var t = cc.instantiate(l.instance.freeSpinMessagePreFab);
        t.name = "winNode";
        var i = t.getChildByName("PreBonusMessage");
        if (o.instance.mPortrait ? (l.instance.windowHolder.setRotation(-90),
l.instance.windowHolder.scaleX = .75, l.instance.windowHolder.scaleY = .75) :
(l.instance.windowHolder.scaleX = 1, l.instance.windowHolder.scaleY = 1,
l.instance.windowHolder.setRotation(0)), e) {
            l.instance._hudController.showFreeSpinHudMessages();
            l.instance._freeSpinData;
            var n = l.instance._freeSpinCount;
            i.getChildByName("FreeSpin_Additional").active = !1,

```

```

i.getChildByName("FreeSpinAwarded").active = !1, l.instance._isFreeSpinRetrigger ?
(i.getChildByName("FreeSpin_Additional").active = !0,
i.getChildByName("lbl_clicktostart").active = !1) : (i.getChildByName("FreeSpinAwarded").active
= !0, i.getChildByName("lbl_clicktostart").active = !0),
t.getComponent(cc.Animation).play("pre_bonus_In"),
i.getChildByName("freeSpinCount").getComponent(cc.Label).string = n,
i.getChildByName("FreeSpinAwarded").getComponent(cc.Sprite).spriteFrame =
o.instance.winmsg_strings.getSpriteFrame(o.instance.localSpecifier + ".free_spins_awarded"),
i.getChildByName("FreeSpin_Additional").getComponent(cc.Sprite).spriteFrame =
o.instance.winmsg_strings.getSpriteFrame(o.instance.localSpecifier +
".additional_free_spins_awarded"),
i.getChildByName("lbl_clicktostart").getComponent(cc.Sprite).spriteFrame =
o.instance.winmsg_strings.getSpriteFrame(o.instance.localSpecifier + ".click_to_start"),
l.instance._displayFreeSpinCount += n, t.getChildByName("PostBonusMessage").active = !1,
l.instance.windowHolder.addChild(t)
    } else
        l.instance._hudController.lblWin.getComponent(cc.Sprite).spriteFrame =
o.instance.hud_strings.getSpriteFrame(o.instance.localSpecifier + ".total_win"),
l.instance._cumulativeLWinLimit = 0, l.instance._cumulativeLostLimit = 0, o.instance.mPortrait ?
this.freeSpinPotraitModeNode.active = !0 : this.freeSpinModeNode.active = !0,
this.freeSpinFrameModeNode1.active = !0, this.freeSpinFrameModeNode2.active = !0,
this.freeSpinFrameModeNode3.active = !0, l.instance._isFreeSpinRetrigger &&
(l.instance._isFreeSpinRetrigger = !1), (t =
l.instance.windowHolder.getChildByName("winNode")).getComponent(cc.Animation).play("pre_b
onus_out"), l.instance.onFreeSpinMessageCompleted(), setTimeout(function() {
    l.instance.windowHolder.removeAllChildren(!0)
}, 300)
},
postFreeSpinMessage: function(e) {
    var t = cc.instantiate(l.instance.freeSpinMessagePreFab);
    t.name = "winNode", o.instance.mPortrait ? (l.instance.windowHolder.setRotation(-
90), l.instance.windowHolder.scaleX = .75, l.instance.windowHolder.scaleY = .75) :
(l.instance.windowHolder.scaleX = 1, l.instance.windowHolder.scaleY = 1,
l.instance.windowHolder.setRotation(0));
    var i = t.getChildByName("PostBonusMessage");
    if (e)
        t.getComponent(cc.Animation).play("post_bonus_In"),
i.getChildByName("YOU_HAVE_WON").getComponent(cc.Sprite).spriteFrame =
o.instance.winmsg_strings.getSpriteFrame(o.instance.localSpecifier + ".you_have_won"),
i.getChildByName("lblCreditAmount").getComponent(cc.Label).string =
s.instance.ToCurrency(l.instance._creditsWonThisHand),
t.getChildByName("PreBonusMessage").active = !1, l.instance.windowHolder.addChild(t);
    else if (o.instance.mPortrait ? this.freeSpinPotraitModeNode.active = !1 :
this.freeSpinModeNode.active = !1, this.freeSpinFrameModeNode1.active = !1,
this.freeSpinFrameModeNode2.active = !1, this.freeSpinFrameModeNode3.active = !1, i = (t =
l.instance.windowHolder.getChildByName("winNode")).getChildByName("PostBonusMessage"),
t.getComponent(cc.Animation).play("post_bonus_out"), o.instance.isHistoryDataAvailable = !1,
l.instance._hudController.showHudMessages(),
l.instance._audioController.playMusic(o.instance.MUSIC_IDLE), o.instance.freePlayPro ||
this.updateWinAmountAftterPostMsg(), o.instance.freeSpinBeforeReel = null,
o.instance.freeSpinBeforeBonusReelLayout = null, o.instance.freeSpinBeforeSelectedReelSet =
null, o.instance.freePlayPro) {

```

```

        if (o.instance.showFreeProMsgAfterHistoryHand) {
            o.instance.showFreeProMsgAfterHistoryHand = !1;
            var n = o.instance.freePlayProObj.spins;

l.instance._hudController.updateFreePlayMessage(o.instance._i18n.t("HUD_MESSAGES.FREE_S
PIN_AWARDED", {
            spinCount: n
        })))
        }
    } else
        l.instance._hudController.lblWin.getComponent(cc.Sprite).spriteFrame =
o.instance.hud_strings.getSpriteFrame(o.instance.localSpecifier + ".win")
    },
    updateWinAmountAffterPostMsg: function() {
        setTimeout(function() {
            l.instance._hudController.playWinParticle(),
l.instance._audioController.playWinLineEffect()
        }, 500), setTimeout(function() {
            l.instance._hudController.updateBalance(o.instance.playerBalance),
l.instance.windowHolder.removeAllChildren(10), l.instance.hideWinningSymbols(),
l.instance._MatchPositions = null, l.instance._winLinesArray = null,
l.instance.node.stopAllActions()
        }, 1e3)
    },
    reSetReelToBonusStartup: function() {
        var e = o.instance.freeSpinBeforeSelectedReelSet,
            t = o.instance.freeSpinBeforeReel,
            i = o.instance.freeSpinBeforeBonusReelLayout;
        o.instance.reelLayout = null != t ? t : o.instance.reelLayout, o.instance.selectedReelSet
= null != e ? e : o.instance.selectedReelSet, o.instance.bonusReelLayout = null != i ? i :
o.instance.bonusReelLayout, l.instance.resetSlotMask();
        for (var n = 0; n < this.reelsArray.length; n++)
            this._reelScript[n].setDisplayPositions(n);
        this._bonusReelScript.setDisplayPositions(0)
    },
    showFireBallWinMessage: function(e) {
        e ? (l.instance.winMessageHolder.getComponent(cc.Animation).play("winmsg"),
l.instance.winMessageHolder.active = !0) :
(l.instance.winMessageHolder.getComponent(cc.Animation).play("hideWinMsg"),
setTimeout(function() {
            l.instance.winMessageHolder.active = !1
        }, 1500))
    },
    loadWinPositionsAndPrefabs: function() {
        for (var e = 0; e < this.slotWinArrayPositions.length; e++) {
            var t = cc.instantiate(this.winSlotPrefab);
            this.winningSlotPrefabs[e] = t,
this.slotWinArrayPositions[e].addChild(this.winningSlotPrefabs[e]), this.winningSlotPrefabs[e] =
this.winningSlotPrefabs[e].getComponent("SHS_SlotWinningControl"),
this.winningSlotPrefabs[e].loadLogos()
        }
        for (e = 0; e < this.ShapeShiftSymbolWinFrame.children.length; e++) {

```

```

        t = cc.instantiate(this.winSlotPrefab);
        this.shiftSymbolPrefabs[e] = t, this.ShapeShiftSymbolWinFrame.getChildren()
[e].addChild(this.shiftSymbolPrefabs[e]), this.shiftSymbolPrefabs[e] =
this.shiftSymbolPrefabs[e].getComponent("SHS_SlotWinningControl"),
this.shiftSymbolPrefabs[e].loadLogos()
    }
},
initSymbolsPool: function() {
    this._symbolsPool = [];
    for (var e = 0; e < this.slotAnimArray.length; e++)
        this._symbolsPool[e] = new cc.NodePool("PoolSym" + e);
    this._moonSymbolsPool = [];
    for (e = 0; e < this.slotMoonAnimArray.length; e++)
        this._moonSymbolsPool[e] = new cc.NodePool("TopRawPoolSym" + e)
},
updateMusicStatus: function(e) {
    o.instance.audioMusic = e, this._isFreeSpin ||
this._SHS_GameEngineManager.saveSoundSetting()
},
updateSoundState: function(e) {
    o.instance.audioSfx = e, this._isFreeSpin ||
this._SHS_GameEngineManager.saveSoundSetting()
},
updatePaytableStatus: function(e) {},
updateOptionStatus: function(e) {
    this._SHS_GameEngineManager.sendControlPanelStatusToServer(e)
},
updateTotalBet: function() {
    var e = this;

this._SHS_GameEngineManager.updateBetValueOnServer(o.instance.playerBetAmount),
setTimeout(function() {
    e._SHS_GameEngineManager.saveBetValue(o.instance.playerBetAmount)
}, 30)
},
clickOnSplashScreen: function() {
    if (!this._isClickSplash) {
        l.instance._audioController.playButtonSound(), this.EnabledSplashBackground(!1),
l.instance._hudController.settingUI.active = !1, l.instance._hudController.settingMobileUI.active
= !1, cc.sys.isMobile && (this.setFullScreen(),
l.instance._hudController.settingMobileUI.getChildByName("BtnPortraitBar").active = !1),
l.instance._isFreeSpin ? l.instance._audioController.playMusic(o.instance.MUSIC_FREE_SPIN) :
l.instance._audioController.playMusic(o.instance.MUSIC_IDLE);
        if (this._isClickSplash = !0, o.instance.isHistoryDataAvailable ||
(this._hudController.EnabledHUDElements(!0), l.instance._hudController.setSpinButtonState(!0)),
o.instance.isHistoryDataAvailable)
            this.redirectToHistory();
        else if (o.instance.freePlayPro) {
            var e = o.instance.freePlayProObj.spins;

this._hudController.updateFreePlayMessage(o.instance._i18n.t("HUD_MESSAGES.FREE_SPIN_
AWARDED", {

```

```

        spinCount: e
    )))
}

```

```

l.instance._hudController.settingMobileUI.getChildByName("volsunga_logo").setPosition(280, 0),
l.instance._hudController.settingsButton.interactable = !0
    }
},
redirectToHistory: function() {
    o.instance.freePlayPro && (o.instance.showFreeProMsgAfterHistoryHand = !0),
o.instance.isHistoryDataAvailable && (l.instance._isSeverError = !0,
this._hudController.updateErrorMessage(o.instance._i18n.t("MSG.UNFINISHED_GAME_ROUND")), this._hudController.changeErrorButton(o.instance._i18n.t("BUTTON.OK")))
    },
closeSplash: function() {
    setTimeout(function() {
        l.instance._hudController.settingUI.active = !1,
l.instance._hudController.settingMobileUI.active = !1
    }, 300)
},
openAutoPlayWindow: function() {
    this._SHS_GameEngineManager.sendAutoPlayRequestToServer()
},
requestAutoPlay: function() {
    setTimeout(function() {
        setTimeout(function() {
            l.instance._isPauseAutoPlay = !1, l.instance.StartSpinning()
        }, 100)
    }, 100)
},
requestAutoPlayCancel: function() {
    l.instance.setAutoPlay(0, 0, 0)
},
resetLayout: function() {
    var e,
        t = this;
    if (l.instance._ShowWinLine = !1, l.instance._hudController.updateWinAmount(0),
null != t.reelStopSlotItems) {
        for (e = 0; e < t.reelStopSlotItems.length; e++)
            t.reelStopSlotItems[e] && t.reelStopSlotItems[e].setOpacity(255);
        for (var i = 0; i < 4; i++)
            for (e = 0; e < t.winningSlotPrefabs[i].length; e++)
                t.winningSlotPrefabs[i][e] && t.winningSlotPrefabs[i][e].hideLogos()
    }
},
clickOnBackground: function() {
    this._isClicked = !0, l.instance._isCoinMsgDisplay ? l.instance._countdownLevel =
l.instance._creditsWonThisHand : this._isClickSplash && cc.sys.isMobile && !this._isFullScreen
&& (this.setFullScreen(), this._isFullScreen = !0)
},
resetTapCount: function() {
    l.instance._tapCount = 0, clearTimeout(l.instance._timerCheck)
}

```

```

    },
    stopBonusReel: function() {
        var e = this;
        return cc.callFunc(function() {
            e._bonusReelScript.finishSpinning(0)
        })
    },
    onPromotionActivePlaysRemaining: function(e) {
        e && 0 === e.data && (l.instance._isPromotionAvailable = !1)
    },
    onPromotionActivePlaysDone: function(e) {
        l.instance.initPromotion(l.instance.gameFreePlayData)
    },
    stopAutoPlay: function() {
        l.instance._isAutoPlay && l.instance._hudController.showStopButton(!1)
    },
    pauseAutoPlay: function() {
        l.instance._isAutoPlay && !l.instance._isPauseAutoPlay &&
(l.instance._isPauseAutoPlay = !0, l.instance._hudController.stopSpinButton.node.active = !1)
    },
    resumeAutoSpin: function() {
        l.instance._isResumeClick || (l.instance._isResumeClick = !0, l.instance._isSpining ||
l.instance.resumeAutoSpinEx())
    },
    resumeAutoSpinEx: function() {
        l.instance._isPauseAutoPlay && l.instance._isAutoPlay &&
(l.instance._isResumeClick = !1, l.instance._isPauseAutoPlay = !1, l.instance.StartSpinning(),
l.instance._hudController.stopSpinButton.node.active = !0)
    },
    releaseMemory: function() {
        cc.sys.os, cc.sys.OS_ANDROID
    },
    clearMemoryInGameLoad: function() {},
    openDebugPanel: function() {
        l.instance.debugPanel.getComponent("SHS_DebugPanelController").togglePanel()
    },
    setforcePlay: function(e) {
        l.instance._SHS_GameEngineManager.callForPresetHand(e)
    },
    resetSlotMask: function() {
        var e = !1,
            t = this._prvSelectedReelSet + "_" + o.instance.selectedReelSet;
        "" != this._prvSelectedReelSet && "55555" != this._prvSelectedReelSet || (t =
o.instance.selectedReelSet + "_Appear"), this._prvSelectedReelSet != o.instance.selectedReelSet ?
this._prvSelectedReelSet = o.instance.selectedReelSet : e = !0;
        var i = ["53135_Appear", "53335_Appear", "55355_Appear", "53135_53335",
"53135_55355", "53135_55555", "53335_55355", "55355_55555", "53335_55555",
"53335_53135", "55355_53135", "55355_53335"].indexOf(t);
        if (s.instance.gameLog(t), !e)
            if (i > -1 && i < 12) {
                this.reelSlotMaskArr[i].active || (this.reelSlotMaskArr[i].active = !0,
this.reelSlotMaskArr[i].getComponent(cc.Animation).play(t), -1 != ["53135_Appear",

```

```

"53335_Appear", "55355_Appear", "55355_53135", "53335_53135", "55355_53135"].indexOf(t) ?
l.instance._audioController.playAudioReelLock() :
l.instance._audioController.playAudioReelUnLock());
    for (var n = 0; n < this.reelSlotMaskArr.length; n++)
        i != n && (this.reelSlotMaskArr[n].active = !1)
    } else
        for (n = 0; n < this.reelSlotMaskArr.length; n++)
            this.reelSlotMaskArr[n].active = !1
    },
    loadMaskSpriteFarames: function() {
        for (var e, t, i = "", n = 0; n < 4; n++)
            e = "resources/SlotMask/texture/SlotMask0" + n + ".png", i = cc.url.raw(e), t =
cc.textureCache.addImage(i), l.instance._arrTextureFileNames[n] = t
    },
    loadSlotWinArrayPositions: function() {
        for (var e = this.gameWinFrame.getChildren(), t = 0; t < 4; t++) {
            this.slotWinArryPositions[t] = [];
            for (var i = t, n = 0; n < o.instance.reelPositionsArray[i].length; n++) {
                var a = o.instance.reelPositionsArray[i][n];
                null != a && this.slotWinArryPositions[t].push(e[a - 1])
            }
        }
        this.loadWinPositionsAndPrefabs()
    },
    formatReelLayout: function(e) {
        var t = [],
            i = 0,
            n = 0,
            a = 0;
        o.instance.reelToViewMap.forEach(function(s, r) {
            s.forEach(function(s, l) {
                var c = e[r][l];
                "A" == c && i++, "B" == c && n++, "C" == c && a++, t[s - 1] = "mask" !== c ?
o.instance.reelSymbolMap.indexOf(c) + 1 : 10
            })
        }), (i > 0 && n > 0 || i > 0 && a > 0 || n > 0 && a > 0) && (this._isMixedPaly = !0),
o.instance.reelLayout = t
    },
    extractSavedData: function(e) {
        l.instance._savedDataObj = e;
        e.autoPValParams;
        -1 != e.freeSpinData.freeSpinIndex && (o.instance.isFreeSpinHistory = !0,
l.instance._currentFreeSpinIndex = e.freeSpinData.freeSpinIndex,
l.instance._displayFreeSpinCount = e.freeSpinData.displayFreeSpinCount,
o.instance.freeSpinBeforeReel = e.freeSpinData.beforeFreeSpinReelLayout,
o.instance.freeSpinBeforeBonusReelLayout = e.freeSpinData.beforeFreeSpinBonusReelLayout,
o.instance.freeSpinBeforeSelectedReelSet = e.freeSpinData.beforeFreeSpinSelectedReelSet), null !
= e.soundSetting ? (o.instance.audioSfx = e.soundSetting.animationSound, o.instance.audioMusic =
e.soundSetting.backGroundSound) : (o.instance.audioSfx = !0, o.instance.audioMusic = !0),
o.instance.audioMusic ? l.instance._audioController.resumeMusic() :
l.instance._audioController.pauseMusic()
    },

```

```

getGameData: function() {
    this._SHS_GameEngineManager.getGameData()
},
gameNoneFatalErrorHandler: function(e, t, i) {
    o.instance.relaxErrCode = e;
    var n = "";
    if (0 == parseInt(e))
        l.instance._isRealityCheck &&
(this._hudController.changeRealityErrorButton(o.instance._i18n.t("BUTTON.CONTINUE_PLAYING")), this._hudController.changeAHButton(o.instance._i18n.t("BUTTON.AH")),
this._hudController.changeOHButton(o.instance._i18n.t("BUTTON.RL")));
    else {
        o.instance.bonusReelLayout = [1, 2, 3],
this.formatReelLayout(o.instance.noPriceSymbolLayout);
        n = o.instance._i18n.t("ERROR." + e);
        this._hudController.changeErrorButton(o.instance._i18n.t("BUTTON.OK"))
    }
    null != n && "" != n || (n = t), this._frontError && this.showLoadingError(n), -1 != r.comErrorCodes.indexOf(parseInt(e)) && (i = !0), i ?
this._hudController.updateComErrorMessage(n) : this._hudController.updateErrorMessage(n)
},
hideProgressBar: function() {
    this._frontError = !1, o.instance.rcenable && initRgPostMessageAPI(),
setTimeout(function() {
    l.instance._isLoadingComplete = !0,
document.getElementById("splash").style.display = "none";
    navigator.userAgent.match(/SamsungBrowser/i);
    cc.sys.isMobile && cc.sys.os == cc.sys.OS_IOS && o.instance.fullscreen &&
(viewSwipSplash(), getCanvasConfiguration()), l.instance._isFreeSpin ||
l.instance._audioController.playMusic(o.instance.MUSIC_IDLE), o.instance.rcenable &&
readyRgPostMessageAPI()
    }, 200)
},
showLoadingError: function(e) {
    document.getElementById("errortxt").innerHTML = e,
document.getElementById("relaxerror").style.display = "block"
},
initializeRealityCheck: function(e) {
    var t = !1,
        i = "";
    null != e && (l.instance._sessionTime = e.totalSessionMinutes + " Min",
l.instance._realityWinAmount = s.instance.ToCurrency(e.winSum), l.instance._lossAmount =
s.instance.ToCurrency(e.betSum), null != e.sessionLimitSeconds && e.sessionMinutes >=
e.sessionLimitSeconds / 60 && (t = !0, i = "MSG.REALITY_CHECK_SESSION_LENGTH"), t
&& (o.instance.rcshowresult && this._SHS_GameEngineManager.getSessionResult(), "" ==
o.instance.rciframeurl && (this._isRealityCheck = !0, this.gameNoneFatalErrorHandler(0,
o.instance._i18n.t(i), !1), this._SHS_GameEngineManager.gameResetRealityCheck(!1, t)))
},
resetShiftPositions: function() {
    l.instance._resetShiftPositions = [];
    for (var e = 0; e < this.reelsArray.length; e++)
        this._reelScript.push(this.reelsArray[e].getComponent("SHS_SlotReel")),

```

```

this._resetShiftPositions[e] = []
    },
    setLandscapeOrPortrait: function(e) {
        this._isClickSplash ? this._isFullScreen = !1 : cc.sys.isMobile ?
(l.instance._hudController.settingMobileUI.active = !0,
l.instance._hudController.settingMobileUI.getChildByName("Btns").active = !1,
l.instance._hudController.settingMobileUI.getChildByName("Btns").getChildByName("mobile_bot
tom_bar").active = !1,
l.instance._hudController.settingMobileUI.getChildByName("autoPlayClose").active = !1,
l.instance._hudController.settingMobileUI.getChildByName("btnPlay").active = !0) :
(l.instance._hudController.settingUI.active = !0,
l.instance._hudController.settingUI.getChildByName("Btns").active = !1,
l.instance._hudController.settingUI.getChildByName("autoPlayClose").active = !1,
l.instance._hudController.settingUI.getChildByName("btnPlay").active = !0,
l.instance._hudController.settingUI.getComponent("SHS_SettingsController").btnHistory.active = !
1);
        var t = /(android)/i.test(navigator.userAgent),
            i = this.isTablet(),
            n = this.isIPhoneSixToEight(),
            a = this.isIPhone5();
        o.instance.urlJurisdiction;
        if (o.instance.fullscreen || (e = cc.sys.os === cc.sys.OS_IOS ? window.orientation :
window.innerHeight > window.innerWidth ? 0 : 90), !cc.sys.isMobile || "landscape-primary" !== e
&& "landscape-secondary" !== e && 90 !== e && -90 !== e) {
            if (cc.sys.isMobile && ("portrait-primary" === e || "portrait-secondary" === e || 0 ===
e)) {
                if (this._isIpad ? (this.mPortraitBackground.getChildByName("SHS_BG").scaleX
= this._portraitBgScale, this.mPortraitBackground.getChildByName("SHS_BG").scaleY =
this._portraitBgScale, this.node.parent.parent.getComponent(cc.Canvas).fitHeight = !1,
this.node.parent.parent.getComponent(cc.Canvas).fitWidth = !0) :
(this.node.parent.parent.getComponent(cc.Canvas).fitHeight = !0, this._isSafari &&
o.instance.fullscreen ? this.node.parent.parent.getComponent(cc.Canvas).fitWidth = !1 : cc.sys.os
=== cc.sys.OS_ANDROID && document.webkitIsFullScreen ?
this.node.parent.parent.getComponent(cc.Canvas).fitWidth = !1 :
this.node.parent.parent.getComponent(cc.Canvas).fitWidth = !0), this._isRotation = !0,
this.setRotationType(!1, !1, !0), this.hudController.active = !1,
this.mobileLandscapeHudController.active = !1, this.mobilePortraitHudController.active = !0,
this.baseBackground.active = !1, this.mLandscapeBackground.active = !1,
this.mPortraitBackground.active = !0, this._hudController =
this.mobilePortraitHudController.getComponent("SHS_PSPHudController"),
l.instance._isFreeSpin)
                    l.instance._hudController.EnabledHUDElements(!1),
this.freeSpinPotraitModeNode.active = !0, this.freeSpinModeNode.active = !1;
                else {
                    if (this.freeSpinPotraitModeNode.active = !1, this.freeSpinModeNode.active = !
1, this._isClickSplash) {
                        this.mobileLandscapeHudController.getComponent("SHS_PSPHudController");
                            l.instance._isGameIdle ? o.instance.freePlayPro && null ===
o.instance.fPProSpinsPlayed && l.instance._isFreePlayMsgOpen ? l.instance._isFreePlayMsgOpen
&& l.instance._hudController.EnabledHUDElements(!1) :
l.instance._hudController.EnabledHUDElements(!0) :

```

```

l.instance._hudController.EnabledHUDElements(!1)
    } else
        l.instance._hudController.EnabledHUDElements(!1);
        l.instance._hudController.EnabledCoinValButton()
    }
    if (l.instance.winMessageHolder.scaleX = .8, l.instance.winMessageHolder.scaleY
= .8, l.instance.windowHolder.setRotation(-90), l.instance.winMessageHolder.setRotation(-90),
l.instance._hudController.comErrorMessage.scaleX = 1,
l.instance._hudController.comErrorMessage.scaleY = 1,
l.instance._hudController.errorMessage.scaleX = 1, l.instance._hudController.errorMessage.scaleY
= 1, l.instance._hudController.freePlayMessage.scaleX = 1,
l.instance._hudController.freePlayMessage.scaleY = 1,
l.instance._hudController.comErrorMessage.setRotation(-90),
l.instance._hudController.errorMessage.setRotation(-90),
l.instance._hudController.realityErrorMessage.setRotation(-90),
l.instance._hudController.freePlayMessage.setRotation(-90), cc.sys.isMobile)

l.instance._hudController.settingMobileUI.getChildByName("page_2").setRotation(-90),
l.instance._hudController.settingMobileUI.getChildByName("page_3").setRotation(-90),
l.instance._hudController.settingMobileUI.getChildByName("page_4").setRotation(-90), 11 == (c
= this.iPadVersion()) || 12 == c || 9 == c || 4 == c || t ?
(l.instance._hudController.settingMobileUI.scaleX = 1.7,
l.instance._hudController.settingMobileUI.scaleY = 1.7) :
(l.instance._hudController.settingMobileUI.scaleX = 1.8,
l.instance._hudController.settingMobileUI.scaleY = 1.8),
l.instance._hudController.settingMobileUI.getChildByName("Paytable_Mobile_Portrait").active = !
0,
l.instance._hudController.settingMobileUI.getChildByName("Paytable_Mobile_Landscape").active
= !1;
    else
        l.instance.splashScreen.setRotation(-90), l.instance.splashScreen.scaleX = .75,
l.instance.splashScreen.scaleY = .75;
        cc.sys.isMobile &&
l.instance._hudController.settingMobileUI.getComponent("SHS_SettingsController").changeSettin
gUIElementsPositions(), l.instance.slotFrame.setRotation(-90);
        var s = 0;
        t ? (s = -30, this.mobilePortraitHudController.setPosition(-60, 0),
this.mobilePortraitHudController.getChildByName("topBar").setPosition(0, -650)) : (n || a) && (s =
-50), l.instance.slotFrame.setPosition(s, 0), l.instance.slotFrame.scaleX = .75,
l.instance.slotFrame.scaleY = .75, this._isSafari && !n && !a && o.instance.fullscreen ?
(this.mobilePortraitHudController.scaleX = 1.05, this.mobilePortraitHudController.scaleY = 1.05,
this._isIpad ? this.mobilePortraitHudController.setPositionX(-40) :
this.mobilePortraitHudController.setPositionX(0)) : (this.mobilePortraitHudController.scaleX = 1,
this.mobilePortraitHudController.scaleY = 1), this._isFirstLoad ? this._isFreeSpin ||
o.instance.freePlayPro || cc.sys.isMobile && l.instance.updateHudElementsInFirstLoad() :
(this._hudController.updateBetAmount(o.instance.playerBetAmount), this._isFreeSpin ?
o.instance.freePlayPro ? this._hudController.updateBalance(o.instance.playerBalance) :
this._hudController.updateBalance(o.instance.playerBalance - l.instance._winAmount) :
this._hudController.updateBalance(o.instance.playerBalance),
this._hudController.updateWinAmount(l.instance._creditsWonThisHand))
    }
} else {

```

```

var r = navigator.userAgent.match(/SamsungBrowser/i);
this._isClickSplash && null != r &&
(document.getElementById("safariView").style.width = "100%");
var c = this.iPadVersion();
if (this._isIpad ?
(this.mLandscapeBackground.getChildByName("SHS_BG").scaleX = this._landscapeBgScale,
this.mLandscapeBackground.getChildByName("SHS_BG").scaleY = this._landscapeBgScale,
this.freeSpinModeNode.getChildByName("SHS_BG").scaleX = this._landscapeBgScale,
this.freeSpinModeNode.getChildByName("SHS_BG").scaleY = this._landscapeBgScale,
this.node.parent.parent.getComponent(cc.Canvas).fitHeight = !1,
this.node.parent.parent.getComponent(cc.Canvas).fitWidth = !0, 9 == c || 11 == c || 12 == c ?
this._isSafari && (this.mLandscapeBackground.getChildByName("SHS_BG").setPositionY(-300),
this.freeSpinModeNode.getChildByName("SHS_BG").setPositionY(50),
this.mobileLandscapeHudController.setPositionY(-30)) : this._isSafari &&
this.mobileLandscapeHudController.setPositionY(-70)) :
(this.node.parent.parent.getComponent(cc.Canvas).fitHeight = !0,
this.node.parent.parent.getComponent(cc.Canvas).fitWidth = !1), this._isRotation = !0,
this.setRotationType(!1, !0, !1), this.hudController.active = !1,
this.mobilePortraitHudController.active = !1, this.mobileLandscapeHudController.active = !0,
this.baseBackground.active = !1, this.mLandscapeBackground.active = !0,
this.mPortraitBackground.active = !1, this._hudController =
this.mobileLandscapeHudController.getComponent("SHS_PSPHudController"),
l.instance.slotFrame.scaleX = .9, l.instance.slotFrame.scaleY = .9, l.instance._isFreeSpin ?
(this.freeSpinPotraitModeNode.active = !1, this.freeSpinModeNode.active = !0,
l.instance._hudController.EnabledHUDElements(!1)) : (this.freeSpinPotraitModeNode.active = !1,
this.freeSpinModeNode.active = !1, this._isClickSplash && l.instance._isGameIdle ?
o.instance.freePlayPro && null == o.instance.fPProSpinsPlayed &&
l.instance._isFreePlayMsgOpen ? l.instance._isFreePlayMsgOpen &&
l.instance._hudController.EnabledHUDElements(!1) :
l.instance._hudController.EnabledHUDElements(!0) :
l.instance._hudController.EnabledHUDElements(!1),
l.instance._hudController.EnabledCoinValButton()), l.instance.windowHolder.scaleX = .9,
l.instance.windowHolder.scaleY = .9, l.instance.windowHolder.setRotation(0),
l.instance.winMessageHolder.scaleX = 1, l.instance.winMessageHolder.scaleY = 1,
l.instance.winMessageHolder.setRotation(0), l.instance._hudController.comErrorMessage.scaleX =
1, l.instance._hudController.comErrorMessage.scaleY = 1,
l.instance._hudController.errorMessage.scaleX = 1, l.instance._hudController.errorMessage.scaleY
= 1, l.instance._hudController.freePlayMessage.scaleX = 1,
l.instance._hudController.freePlayMessage.scaleY = 1,
l.instance._hudController.comErrorMessage.setRotation(0),
l.instance._hudController.errorMessage.setRotation(0),
l.instance._hudController.realityErrorMessage.setRotation(0),
l.instance._hudController.freePlayMessage.setRotation(0), cc.sys.isMobile) {
    if
(l.instance._hudController.settingMobileUI.getChildByName("page_2").setRotation(0),
l.instance._hudController.settingMobileUI.getChildByName("page_3").setRotation(0),
l.instance._hudController.settingMobileUI.getChildByName("page_4").setRotation(0), 11 == (c =
this.iPadVersion()) || 12 == c || 9 == c || 4 == c) {
        var d = 1.6;
        11 == c && (d = 1.5), l.instance._hudController.settingMobileUI.scaleX = d,
l.instance._hudController.settingMobileUI.scaleY = d,
this.mobileLandscapeHudController.getChildByName("BtnsNode").scaleX = .94,

```

```

this.mobileLandscapeHudController.getChildByName("BttnsNode").scaleY = .94
    } else
        l.instance._hudController.settingMobileUI.scaleX = 1.8,
l.instance._hudController.settingMobileUI.scaleY = 1.8;

l.instance._hudController.settingMobileUI.getChildByName("Paytable_Mobile_Portrait").active = !
1,
l.instance._hudController.settingMobileUI.getChildByName("Paytable_Mobile_Landscape").active
= !0
    } else
        l.instance.splashScreen.setRotation(0), l.instance.splashScreen.scaleX = 1,
l.instance.splashScreen.scaleY = 1;
        cc.sys.isMobile &&
l.instance._hudController.settingMobileUI.getComponent("SHS_SettingsController").changeSettin
gUIElementsPositions(), l.instance.slotFrame.setRotation(0), l.instance.slotFrame.setPosition(3,
18.3), this._isFirstLoad ? this._isFreeSpin || o.instance.freePlayPro || cc.sys.isMobile &&
l.instance.updateHudElementsInFirstLoad() :
(this._hudController.updateBetAmount(o.instance.playerBetAmount), this._isFreeSpin ?
o.instance.freePlayPro ? this._hudController.updateBalance(o.instance.playerBalance) :
this._hudController.updateBalance(o.instance.playerBalance - l.instance._winAmount) :
this._hudController.updateBalance(o.instance.playerBalance),
this._hudController.updateWinAmount(l.instance._creditsWonThisHand)), t && i &&
(this.mobileLandscapeHudController.getChildByName("BttnsNode").scaleX = .85,
this.mobileLandscapeHudController.getChildByName("BttnsNode").scaleY = .85,
this.mobileLandscapeHudController.getChildByName("BttnsNode").setPosition(40, 0))
    }
    0 != this._autoPlayCounter ?
l.instance._hudController.setSpinCounter(this._autoPlayCounter, 2) :
(l.instance._hudController.showStopButton(!1), l.instance.requestAutoPlayCancel()),
l.instance._hudController.setFreeSpinCounter(l.instance._displayFreeSpinCount),
l.instance._hudController.showTurboButton(!1), l.instance._isFreeSpin ?
l.instance._hudController.showFreeSpinHudMessages() :
l.instance._hudController.showHudMessages(), (o.instance.freePlayPro || l.instance._isFreeSpin)
&& !this._isLocalized && o.instance.hud_strings &&
(l.instance._hudController.lblWin.getComponent(cc.Sprite).spriteFrame =
o.instance.hud_strings.getSpriteFrame(o.instance.localSpecifier + ".total_win"))
    },
    checkOrientation: function() {
        return o.instance.fullscreen ? "iOS" == cc.sys.os ? window.orientation :
screen.orientation.type : window.innerHeight > window.innerWidth ? 0 : 90
    },
    setRotationType: function(e, t, i) {
        o.instance.desktop = e, o.instance.mLandscape = t, o.instance.mPortrait = i
    },
    clickReplay: function() {
        this._SHS_GameEngineManager.sendRePlayRequest()
    },
    startTime: function() {
        var e = new Date,
            t = e.getHours(),
            i = e.getMinutes(),
            n = e.getSeconds();

```

```

        i = l.instance.checkTime(i), n = l.instance.checkTime(n),
l.instance.timeLabelPotrait.getComponent(cc.Label).string = t + ":" + i + ":" + n,
l.instance.timeLabelLandscape.getComponent(cc.Label).string = t + ":" + i + ":" + n,
l.instance.timeLabelLandscapeMobile.getComponent(cc.Label).string = t + ":" + i + ":" + n,
l.instance.fgtimeLabelLandscape.getComponent(cc.Label).string = t + ":" + i + ":" + n,
l.instance.timeLabelSettingUI.getComponent(cc.Label).string = t + ":" + i + ":" + n;
        setTimeout(l.instance.startTime, 500)
    },
    checkTime: function(e) {
        return e < 10 && (e = "0" + e), e
    },
    sendFreeProAction: function(e) {
        l.instance._SHS_GameEngineManager.sendFreePlayProActionRequest(e)
    },
    freeRoundPromoFinishMsg: function() {
        o.instance.fPProSpinsPlayed = null,
l.instance._hudController.onAutoPlayStopSelected(), o.instance.gameIdle = !0,
l.instance._hudController.updateErrorMessage(o.instance._i18n.t("HUD_MESSAGES.FREE_SPIN_OVER")), l.instance._hudController.showHudMessages()
    },
    updateHudElementsInFirstLoad: function() {
        0 != o.instance.playerBetAmount &&
(this.mobileLandscapeHudController.getComponent("SHS_PSPHudController").updateBetAmount(o.instance.playerBetAmount),
this.mobilePortraitHudController.getComponent("SHS_PSPHudController").updateBetAmount(o.instance.playerBetAmount)), 0 != o.instance._creditsWonThisHand &&
(this.mobileLandscapeHudController.getComponent("SHS_PSPHudController").updateWinAmount(l.instance._creditsWonThisHand),
this.mobilePortraitHudController.getComponent("SHS_PSPHudController").updateWinAmount(l.instance._creditsWonThisHand)), 0 != o.instance.playerBalance &&
(this.mobileLandscapeHudController.getComponent("SHS_PSPHudController").updateBalance(o.instance.playerBalance - l.instance._winAmount),
this.mobilePortraitHudController.getComponent("SHS_PSPHudController").updateBalance(o.instance.playerBalance - l.instance._winAmount)),
this.mobileLandscapeHudController.getComponent("SHS_PSPHudController").EnabledCoinValButton(),
this.mobileLandscapeHudController.getComponent("SHS_PSPHudController")._coinValIndex = o.instance._commonCoinValIndex,
this.mobilePortraitHudController.getComponent("SHS_PSPHudController").EnabledCoinValButton(), this.mobilePortraitHudController.getComponent("SHS_PSPHudController")._coinValIndex = o.instance._commonCoinValIndex
    },
    setFullScreen: function() {
        var e = document.getElementById("safariView");
        navigator.userAgent.match(/SamsungBrowser/i);
        cc.sys.os != cc.sys.OS_IOS && o.instance.fullscreen && (document.fullscreenElement || document.mozFullScreenElement || document.webkitFullscreenElement || document.msFullscreenElement ? this._isFullScreen = !1 : (e.requestFullscreen ? (e.requestFullscreen(), document.getElementById("safariView").style.width = "100%") : e.msRequestFullscreen ? (e.msRequestFullscreen(), document.getElementById("safariView").style.width = "100%") : e.mozRequestFullscreen ? (e.mozRequestFullscreen(), document.getElementById("safariView").style.width = "100%") :

```

```

e.webkitRequestFullscreen ? (e.webkitRequestFullscreen(),
document.getElementById("safariView").style.width = "100%") :
this.setUIForFullScreenModeDiffDevice(), this._isFullScreen = !0))
    },
    checkIsFullScreen: function(e) {
        o.instance.isSafariFullScreen = e
    },
    setPotraitPositions: function(e) {
        this._isSafari ? e > 2 ? (l.instance.slotFrame.scaleX = .8, l.instance.slotFrame.scaleY = .
8, this.mobilePortraitHudController.scaleX = 1.05, this.mobilePortraitHudController.scaleY = 1.05,
this.mobilePortraitHudController.setPositionX(0)) : (l.instance.slotFrame.scaleX = .9,
l.instance.slotFrame.scaleY = .9, this.mobilePortraitHudController.scaleX = 1.15,
this.mobilePortraitHudController.scaleY = 1.15, this.mobilePortraitHudController.setPositionX(-
45)) : (l.instance.slotFrame.scaleX = .75, l.instance.slotFrame.scaleY = .75,
this.mobilePortraitHudController.scaleX = 1, this.mobilePortraitHudController.scaleY = 1)
    },
    EnabledSplashBackground: function(e) {
        o.instance.mPortrait && this._isSafari
    },
    setUIForNormalModeDifferentDevice: function() {
        var e = /(android)/i.test(navigator.userAgent),
            t = this.isSmartphone(),
            i = (this.isTablet(), /iPad/.test(navigator.userAgent) && !window.MSStream),
            n = /iPhone/.test(navigator.userAgent) && !window.MSStream,
            a = (/iPod/.test(navigator.userAgent) && window.MSStream, this.isiPhoneX()),
            s = this.isiPhoneXr(),
            r = this.isiPhoneXs(),
            c = this.isiPhoneSix8Plus(),
            d = this.isiPhoneSixToEight(),
            u = this.iPadVersion(),
            S = this.isiPhone5();
        navigator.userAgent.match(/SamsungBrowser/i);
        e && t ? this.mobilePortraitHudController.setPosition(-30, 0) : e ?
(this.mobilePortraitHudController.setPosition(-60, 0),
this.mobilePortraitHudController.getChildByName("topBar").setPosition(0, -650),
this.mobilePortraitHudController.setPosition(-60, 0),
this.mobilePortraitHudController.getChildByName("topBar").setPosition(0, -650),
this.mobileLandscapeHudController.getChildByName("BtnsNode").scaleX = .85,
this.mobileLandscapeHudController.getChildByName("BtnsNode").scaleY = .85,
this.mobileLandscapeHudController.getChildByName("BtnsNode").setPosition(40, 0)) : i ? 9 == u
|| 11 == u || 12 == u ? (this.mobilePortraitHudController.setPosition(-40, 0), this._isSafari &&
this.mobileLandscapeHudController.setPosition(0, 0)) : o.instance.mPortrait ?
this.mobilePortraitHudController.setPosition(-40, 0) : this._isSafari &&
this.mobileLandscapeHudController.setPosition(0, -80) : a ?
this.mobilePortraitHudController.setPosition(0, 0) : s ?
this.mobilePortraitHudController.setPosition(0, 0) : r ?
this.mobilePortraitHudController.setPosition(-80, 0) : c ?
this.mobilePortraitHudController.setPosition(-20, 0) : d ?
(this.mobilePortraitHudController.getChildByName("topBar").setPosition(0, -650),
this.mobilePortraitHudController.setPosition(-80, 0), this.mobilePortraitHudController.scaleX = 1,
this.mobilePortraitHudController.scaleY = 1, l.instance.slotFrame.setPosition(-50, 0)) : S ?
(l.instance._hudController.settingMobileUI.scaleX = 1.5,

```

```

l.instance._hudController.settingMobileUI.scaleY = 1.5,
l.instance._hudController.settingMobileUI.setPosition(-75, 0),
l.instance._hudController.settingMobileUI.scaleX = 1.5,
l.instance._hudController.settingMobileUI.scaleY = 1.5,
l.instance._hudController.settingMobileUI.setPosition(0, 0),
this.mobilePortraitHudController.scaleX = 1, this.mobilePortraitHudController.scaleY = 1,
this.mobilePortraitHudController.getChildByName("topBar").setPosition(0, -650),
this.mobilePortraitHudController.setPositionX(-60)) : n &&
this.mobilePortraitHudController.setPosition(-80, 0)
},
setUIForFullScreenModeDiffDevice: function() {
    var e = /(android)/i.test(navigator.userAgent),
        t = this.isSmartphone(),
        i = (this.isTablet(), /iPad/.test(navigator.userAgent) && !window.MSStream),
        n = /iPhone/.test(navigator.userAgent) && !window.MSStream,
        a = (/iPod/.test(navigator.userAgent) && window.MSStream, this.isiPhoneX()),
        s = this.isiPhoneXr(),
        r = this.isiPhoneXs(),
        c = this.isiPhoneSix8Plus(),
        d = this.isiPhoneSixToEight(),
        u = this.iPadVersion(),
        S = this.isiPhone5();
    navigator.userAgent.match(/SamsungBrowser/i);
    e && t ? this.mobilePortraitHudController.setPosition(-30, 0) : e ?
(this.mobilePortraitHudController.setPosition(-60, 0),
this.mobilePortraitHudController.getChildByName("topBar").setPosition(0, -650),
this.mobilePortraitHudController.setPosition(-60, 0),
this.mobilePortraitHudController.getChildByName("topBar").setPosition(0, -650),
this.mobileLandscapeHudController.getChildByName("BtnsNode").scaleX = .85,
this.mobileLandscapeHudController.getChildByName("BtnsNode").scaleY = .85,
this.mobileLandscapeHudController.getChildByName("BtnsNode").setPosition(40, 0)) : i ? 9 == u
|| 11 == u || 12 == u ? (this.mobilePortraitHudController.setPosition(-30, 0), this._isSafari &&
this.mobileLandscapeHudController.setPosition(0, 0)) : o.instance.mPortrait ?
this.mobilePortraitHudController.setPosition(-30, 0) : this._isSafari &&
this.mobileLandscapeHudController.setPosition(0, -80) : a ?
(l.instance._hudController.settingMobileUI.scaleX = 1.8,
l.instance._hudController.settingMobileUI.scaleY = 1.8,
l.instance._hudController.settingMobileUI.setPosition(-80, 0),
this.mobilePortraitHudController.setPosition(0, 0)) : s ?
(l.instance._hudController.settingMobileUI.scaleX = 1.8,
l.instance._hudController.settingMobileUI.scaleY = 1.8,
l.instance._hudController.settingMobileUI.setPosition(-80, 0),
this.mobilePortraitHudController.getChildByName("topBar").setPosition(0, -450),
this.mobilePortraitHudController.setPosition(0, 0)) : S ?
(l.instance._hudController.settingMobileUI.scaleX = 1.5,
l.instance._hudController.settingMobileUI.scaleY = 1.5,
l.instance._hudController.settingMobileUI.setPosition(-75, 0),
l.instance._hudController.settingMobileUI.scaleX = 1.5,
l.instance._hudController.settingMobileUI.scaleY = 1.5,
l.instance._hudController.settingMobileUI.setPosition(0, 0),
this.mobilePortraitHudController.scaleX = 1, this.mobilePortraitHudController.scaleY = 1,
this.mobilePortraitHudController.getChildByName("topBar").setPosition(0, -650),

```

```

this.mobilePortraitHudController.setPositionX(-60)) : r ?
(l.instance._hudController.settingMobileUI.scaleX = 1.8,
l.instance._hudController.settingMobileUI.scaleY = 1.8,
l.instance._hudController.settingMobileUI.setPosition(-65, 0),
this.mobilePortraitHudController.getChildByName("topBar").setPosition(0, -450),
this.mobilePortraitHudController.setPosition(0, 0)) : c ?
(l.instance._hudController.settingMobileUI.scaleX = 1.6,
l.instance._hudController.settingMobileUI.scaleY = 1.6,
l.instance._hudController.settingMobileUI.setPosition(-75, 0),
l.instance._hudController.settingMobileUI.scaleX = 1.6,
l.instance._hudController.settingMobileUI.scaleY = 1.6,
l.instance._hudController.settingMobileUI.setPosition(0, 0),
this.mobilePortraitHudController.scaleX = 1.05, this.mobilePortraitHudController.scaleY = 1.05,
this.mobilePortraitHudController.setPositionX(-70),
this.mobilePortraitHudController.getChildByName("topBar").setPosition(0, -650),
l.instance.slotFrame.setPosition(-50, 0)) : d ?
(this.mobilePortraitHudController.getChildByName("topBar").setPosition(0, -650),
this.mobilePortraitHudController.setPosition(-80, 0), this.mobilePortraitHudController.scaleX = 1,
this.mobilePortraitHudController.scaleY = 1, l.instance.slotFrame.setPosition(-50, 0)) : n &&
(l.instance._hudController.settingMobileUI.scaleX = 1.6,
l.instance._hudController.settingMobileUI.scaleY = 1.6,
l.instance._hudController.settingMobileUI.setPosition(-75, 0),
l.instance._hudController.settingMobileUI.scaleX = 1.6,
l.instance._hudController.settingMobileUI.scaleY = 1.6,
l.instance._hudController.settingMobileUI.setPosition(0, 0),
this.mobilePortraitHudController.scaleX = 1.05, this.mobilePortraitHudController.scaleY = 1.05,
this.mobilePortraitHudController.setPositionX(-70),
this.mobilePortraitHudController.getChildByName("topBar").setPosition(0, -650))
},
isSmartphone: function() {
    return window.innerWidth <= 384 && window.innerHeight <= 640
},
isTablet: function() {
    return window.innerWidth <= 1280 && window.innerHeight >= 800
},
isiPhoneX: function() {
    var e = /iPhone/.test(navigator.userAgent) && !window.MSStream,
        t = !1,
        i = window.devicePixelRatio || 1,
        n = window.screen.width * i,
        a = window.screen.height * i;
    return e && 1125 == n && 2436 === a && (t = !0), t
},
isiPhoneXr: function() {
    var e = /iPhone/.test(navigator.userAgent) && !window.MSStream,
        t = !1,
        i = window.devicePixelRatio || 1,
        n = window.screen.width * i,
        a = window.screen.height * i;
    return e && (828 == n && 1792 === a || 750 == n && 1624 === a) && (t = !0), t
},
isiPhoneXs: function() {

```

```

var e = /iPhone/.test(navigator.userAgent) && !window.MSStream,
    t = !1,
    i = window.devicePixelRatio || 1,
    n = window.screen.width * i,
    a = window.screen.height * i;
return e && 1242 == n && 2688 === a && (t = !0), t
},
isIPhoneSix8Plus: function() {
var e = /iPhone/.test(navigator.userAgent) && !window.MSStream,
    t = null,
    i = window.devicePixelRatio || 1,
    n = window.screen.width * i,
    a = window.screen.height * i;
return e && 1242 == n && 2208 === a && (t = !0), t
},
isIPhoneSixToEight: function() {
var e = /iPhone/.test(navigator.userAgent) && !window.MSStream,
    t = null,
    i = window.devicePixelRatio || 1,
    n = window.screen.width * i,
    a = window.screen.height * i;
return e && 750 == n && 1334 === a && (t = !0), t
},
isIPhone5: function() {
var e = /iPhone/.test(navigator.userAgent) && !window.MSStream,
    t = null,
    i = window.devicePixelRatio || 1,
    n = window.screen.width * i,
    a = window.screen.height * i;
return e && 640 == n && 1136 === a && (t = !0), t
},
iPadVersion: function() {
var e = /iPad/.test(navigator.userAgent) && !window.MSStream,
    t = null,
    i = window.devicePixelRatio || 1,
    n = window.screen.width * i,
    a = window.screen.height * i;
return e && (1668 == n && 2388 === a ? t = 11 : 2048 == n && 2732 === a ? t = 12 :
1536 == n && 2048 === a && (t = 9)), t
}
});
cc._RF.pop()
}, {
SHS_GameEngineManager: "SHS_GameEngineManager",
SHS_GameGlobal: "SHS_GameGlobal",
SHS_GameHelper: "SHS_GameHelper",
SHS_RelaxError: "SHS_RelaxError",
SHS_SlotWinningControl: "SHS_SlotWinningControl",
i18n: "i18n"
}],
SHS_GameEngineManager: [function(e, t, i) {
"use strict";

```

```

cc._RF.push(t, "387717ZSKdKjpD6qZSc6E/i", "SHS_GameEngineManager");
var n = e("SHS_RelaxHandler"),
    a = e("SHS_GameGlobal"),
    o = e("SHS_GameHelper");
cc.Class({
  extends: cc.Component,
  properties: {
    _legaHandler: null,
    _SHS_GameController: null,
    _gameEngine: null
  },
  statics: {
    instance: null
  },
  assignGameEngine: function(e) {
    this._gameEngine = new n
  },
  initHandler: function(e, t) {
    this.assignGameEngine(e), this._SHS_GameController = t,
    this._gameEngine.assignController(this), this._gameEngine.initHandler()
  },
  OpenPlayRequest: function() {
    this._gameEngine.OpenPlayRequest()
  },
  gameInitializeHandler: function(e) {
    a.instance.isHistoryDataAvailable = !1, e.refresh && e.refresh.restoreState &&
    "SpinStart" == e.refresh.restoreState && (a.instance.isHistoryDataAvailable = !0,
    this._SHS_GameController._isRelaxDataAvailable = !1, a.instance.historyData = e),
    this._SHS_GameController.gameInitializeHandler(e)
  },
  sendPlayRequest: function(e, t, i) {
    this._gameEngine.sendPlayRequest(e, t, i, this._SHS_GameController._restoreVersion)
  },
  sendRePlayRequest: function() {
    this._gameEngine.sendRepPlayRequest()
  },
  getBetResponseHandler: function(e) {
    this._SHS_GameController.getBetResponseHandler(e)
  },
  saveAutoPlayValue: function(e, t, i) {
    this._gameEngine.saveAutoPlayValue(e, t, i)
  },
  getGameData: function() {
    this._gameEngine.getGameData()
  },
  extractSavedData: function(e) {
    this._SHS_GameController.extractSavedData(e)
  },
  saveFreeSpin: function(e, t) {
    this._gameEngine.saveFreeSpin(e, t)
  },
  saveSoundSetting: function() {

```

```

    this._gameEngine.saveSoundSetting()
  },
  setRestoreState: function(e) {
    this._SHS_GameController._restoreVersion++, this._gameEngine.setRestoreState(e,
this._SHS_GameController._restoreVersion)
  },
  gameFinished: function() {
    this._gameEngine.gameFinished(this._SHS_GameController._restoreVersion)
  },
  redirectToHistory: function() {
    this._gameEngine.redirectToHistory()
  },
  assignParam: function() {
    var e = document.location.search || document.location.hash,
        t = decodeURIComponent(o.instance.getQueryParamValue(e, "clientid")),
        i = decodeURIComponent(o.instance.getQueryParamValue(e, "ticket")),
        n = decodeURIComponent(o.instance.getQueryParamValue(e, "currency")),
        s = decodeURIComponent(o.instance.getQueryParamValue(e, "partnerid")),
        r = decodeURIComponent(o.instance.getQueryParamValue(e, "moneymode")),
        l = decodeURIComponent(o.instance.getQueryParamValue(e, "lang")),
        c = decodeURIComponent(o.instance.getQueryParamValue(e, "minSpinTime")),
        d = decodeURIComponent(o.instance.getQueryParamValue(e, "jurisdiction")),
        u = decodeURIComponent(o.instance.getQueryParamValue(e, "rcinterval")),
        S = decodeURIComponent(o.instance.getQueryParamValue(e, "rcelapsed")),
        E = decodeURIComponent(o.instance.getQueryParamValue(e, "rchistoryurl")),
        _ = decodeURIComponent(o.instance.getQueryParamValue(e, "homeurl")),
        p = decodeURIComponent(o.instance.getQueryParamValue(e, "rcenable")),
        A = decodeURIComponent(o.instance.getQueryParamValue(e, "fullscreen")),
        m = decodeURIComponent(o.instance.getQueryParamValue(e, "rciframeurl")),
        h = decodeURIComponent(o.instance.getQueryParamValue(e, "rcshowresult"));
    "" !== A && "false" === A && (a.instance.fullscreen = !1), ("!" === p && "true" === p ||
"" !== d || "SE" === d.toUpperCase()) && (a.instance.rcenable = !0), ("!" === h && "true" === h &&
(a.instance.rcshowresult = !0), "" !== d && (a.instance.urlJurisdiction = d.toUpperCase()), "" !== m
&& (a.instance.rciframeurl = m), "" !== c && (a.instance.urlMinSpinTime = c),
a.instance._rePlayLang = l, a.instance.localSpecifier = l, a.instance.clientId = t, a.instance.ticketId =
i, "fun" !== r && "" !== r || (n = a.instance.currencyId), "real" !== r && "" === i ||
(a.instance.isRmLogin = !0), a.instance.currencyId = n, a.instance.urlPartnerId = s,
a.instance.moneyMode = r, a.instance.rcInterval = u, a.instance.rcElapsed = S,
a.instance.rcHistoryUrl = E, a.instance.rcHomeUrl = _
  },
  gameNoneFatalErrorHandler: function(e, t, i) {
    this._SHS_GameController.gameNoneFatalErrorHandler(e, t, i)
  },
  gameSetRealityCheck: function(e, t, i) {
    this._gameEngine.gameSetRealityCheck(e, t, i, function(e) {})
  },
  gameResetRealityCheck: function(e, t) {
    this._gameEngine.gameResetRealityCheck(e, t)
  },
  sendFreePlayProActionRequest: function(e) {
    this._gameEngine.sendFreePlayProActionRequest(e)
  },
}

```

```

    getSessionResult: function() {
        this._gameEngine.getSessionResult()
    }
});
cc._RF.pop()
}, {
    SHS_GameGlobal: "SHS_GameGlobal",
    SHS_GameHelper: "SHS_GameHelper",
    SHS_RelaxHandler: "SHS_RelaxHandler"
}],
SHS_GameGlobal: [function(e, t, i) {
    "use strict";
    var n;
    function a(e, t, i) {
        return t in e ? Object.defineProperty(e, t, {
            value: i,
            enumerable: !0,
            configurable: !0,
            writable: !0
        }) : e[t] = i, e
    }
}
cc._RF.push(t, "8d5fdOS8JhLX4YJOmZqEqLj", "SHS_GameGlobal");
var o = cc.Class({
    extends: cc.Component,
    properties: (n = {
        APPLICATION_VERSION: "V 1.0.9 10-04-2019",
        TUTORIAL_STEP: 0,
        SERVER_MODE: "PROD",
        BET_LINES: 100,
        sessionId: null,
        gameId: null,
        userId: null,
        lastPlayId: null,
        ERROR_CONNECTION: 99,
        STR_ERROR_CONNECTION: "Connection Error.",
        ERROR_OPEN_SESSION_SIGN: 999,
        STR_ERROR_OPEN_SESSION_SIGN: "Recovering previous open play.",
        audioMusic: !0,
        audioSfx: !0,
        isBackgroundMusicPlay: !1,
        timeoutFunc: null,
        playerBalance: 0,
        playerFreeBalance: 0,
        playerBetAmount: 0,
        _autoPlayCount: 0,
        playerBetFrom7DG: 0,
        _autoPlayCountFrom7DG: 0,
        gameCurrencyData: null,
        TIME_OUT: 3e5,
        isLEGAServerTimeout: !1,
        betPreLineList: [],
        lineLayout: [],
    })

```

reelLayout: [],  
noPriceSymbolLayout: [],  
bonusReelLayout: [],  
noPriceReelLayout: [],  
noPriceBonusReelLayout: [],  
autoPlayDefault: 1,  
autoPlayList: [],  
gameInitDataBuffer: null,  
gameRTP: 97.73,  
isBlockedUI: !1,  
isGameLobbyAvailable: !1,  
isHistoryDataAvailable: !1,  
historyEvent: null,  
payTableInfo: null,  
selectedReelSet: 53135,  
reelSetArray: null,  
sessionIdEvent: null,  
serverRequestCompletedEvent: null,  
serverRequestFailedEvent: null,  
EVENT\_PLACEBET: "EventPlaceBetResponse",  
EVENT\_GAME\_INITIALIZE: "EventGameInitializeResponse",  
EVENT\_OPEN\_PLAY: "EventOpenPlayResponse",  
EVENT\_GET\_BET\_RESULTS: "EventGetBetResponse",  
EVENT\_SETTLE\_PLAY: "EventSettlePlayResponse",  
EVENT\_CLOSE\_PLAY: "EventClosePlayResponse",  
EVENT\_MUSIC\_STATUS\_CHANGE: "EventMusicStatusChangeResponse",  
EVENT\_SFX\_STATUS\_CHANGE: "EventSFXStatusChangeResponse",  
EVENT\_BALANCE\_CHANGE: "EventBalanceChangeResponse",  
EVENT\_OPTION\_WINDOW\_CHANGE: "EventOptionWindowChangeResponse",  
EVENT\_PAYTABLE\_WINDOW\_CHANGE:  
"EventpaytableWindowChangeResponse",  
EVENT\_JACKPOT\_DATA: "EventJackPotDataResponse",  
SYMBOL\_SOUND\_BEAR: 2,  
SYMBOL\_SOUND\_WOLF: 1,  
SYMBOL\_SOUND\_EAGLE: 3,  
MUSIC\_IDLE: 0,  
MUSIC\_SPIN: 1,  
MUSIC\_FREE\_SPIN: 2,  
WIN\_LOOP\_START: 0,  
WIN\_LOOP: 1,  
WIN\_LOOP\_END: 2,  
\_i18n: null,  
font\_red: null,  
font\_yellow: null,  
font\_white: null,  
hud\_strings: null,  
paytable\_strings: null,  
sound\_strings: null,  
auto\_strings: null,  
gamerules\_strings: null,  
features\_strings: null,  
localSpecifier: "en",

```

    GAME_ENGINE_ID: "RTK",
    BackgroundMusic: null,
    AudioClipID: null,
    AudioClipArray: [],
    audioContext: null,
    audioBuffer: [],
    gameFreePlayData: null,
    legaClockDisplyWorker: null,
    reelToViewMap: [],
    rawsToViewMap: [],
    reelPositionsArray: [],
    reelDisplayPositionsArray: [],
    gameMode: 0,
    historyFreeSpinCount: null,
    historyDisplayFreeSpinCount: null,
    historyBonus: null,
    historyCumulativeWin: null,
    historyCreditsWonWinLines: null,
    isFreeSpinHistory: !1,
    freeSpinBeforeReel: [],
    freeSpinBeforeSelectedReelSet: null,
    freeSpinBeforeBonusReelLayout: null
  }, a(n, "gameId", "volsunga"), a(n, "mode", ""), a(n, "clientId", "casino_desktop-windows"),
a(n, "partnerId", ""), a(n, "urlPartnerId", ""), a(n, "ticketId", ""), a(n, "currencyId", "EUR"), a(n,
"channel", "web"), a(n, "gameServerAPIEndpointURL", ""), a(n, "tokenAPIEndpointURL", ""),
a(n, "gameReplayServiceURL", ""), a(n, "moneyMode", ""), a(n, "isRmLogin", !1), a(n,
"urlJurisdiction", null), a(n, "reelSymbolMap", []), a(n, "bonusReelSymbolMap", []), a(n,
"isHistory", !1), a(n, "isGameInilized", !1), a(n, "historyData", !1), a(n, "desktop", !1), a(n,
"mLandscape", !1), a(n, "mPortrait", !1), a(n, "font_normal", null), a(n, "font_bold", null), a(n,
"font_w10_bold", null), a(n, "font_w10_regular", null), a(n, "_rePlayLang", null), a(n,
"freePlayPro", !1), a(n, "freePlayProObj", null), a(n, "freePlayMsgShowed", !1), a(n,
"fPProSpinsPlayed", null), a(n, "showFreeProMsgAfterHistoryHand", !1), a(n, "minSpinTime",
3e3), a(n, "urlMinSpinTime", ""), a(n, "_commonCoinValIndex", 0), a(n, "rcInterval", 0), a(n,
"rcElapsed", 0), a(n, "rcHistoryUrl", ""), a(n, "rcHomeUrl", ""), a(n, "relaxErrCode", 0), a(n,
"winmsg_strings", ""), a(n, "isSafariFullScreen", !1), a(n, "_isAutoOffInFreeSpin", !1), a(n,
"_scatterPayouts", null), a(n, "rcenable", !1), a(n, "fullscreen", !0), a(n, "rciframeurl", ""), a(n,
"rcshowresult", !1), n),
  statics: {
    instance: null
  },
  initGameGlobal: function() {
    o.instance = this
  }
});
cc._RF.pop()
}, {}],
SHS_GameHelper: [function(e, t, i) {
  "use strict";
  cc._RF.push(t, "6644fE3mCZOy5a2fYv9sym5", "SHS_GameHelper");
  var n = e("SHS_GameGlobal"),
  a = cc.Class({
    extends: cc.Component,

```

```

statics: {
  instance: null
},
initGameHelper: function() {
  a.instance = this
},
gameLog: function(e) {},
winStatusToId: function(e) {
  switch (e) {
    case "win":
      return HAND_WIN;
    case "tie":
      return HAND_TIE;
    default:
      return HAND_FAIL
  }
},
ToCurrency: function(e, t) {
  var i = (e /= Math.pow(10, n.instance.gameCurrencyData.decimalPlaces)).toString(),
      a = i.indexOf("."),
      o = n.instance.gameCurrencyData.decimalPlaces,
      s = "\\d(?:=\\d{" + (t || 3) + "})+" + (o > 0 ? "\\D" : "$") + ")",
      r = (e = parseFloat(a > -1 ? i.substr(0, a) + "." + i.substr(a + 1, o) :
i)).toFixed(Math.max(0, ~o));
  return n.instance.gameCurrencyData.currencyPrefix + "" +
(n.instance.gameCurrencyData.decimalSeparator ? r.replace(".",
n.instance.gameCurrencyData.decimalSeparator) : r).replace(new RegExp(s, "g"), "$&" +
(n.instance.gameCurrencyData.groupingSeparator || ",")) +
n.instance.gameCurrencyData.currencySuffix
},
ToFormat: function(e, t) {
  var i = (e /= Math.pow(10, n.instance.gameCurrencyData.decimalPlaces)).toString(),
      a = i.indexOf("."),
      o = n.instance.gameCurrencyData.decimalPlaces,
      s = "\\d(?:=\\d{" + (t || 3) + "})+" + (o > 0 ? "\\D" : "$") + ")",
      r = (e = parseFloat(a > -1 ? i.substr(0, a) + "." + i.substr(a + 1, o) :
i)).toFixed(Math.max(0, ~o));
  return (n.instance.gameCurrencyData.decimalSeparator ? r.replace(".",
n.instance.gameCurrencyData.decimalSeparator) : r).replace(new RegExp(s, "g"), "$&" +
(n.instance.gameCurrencyData.groupingSeparator || ","))
},
isInt: function(e) {
  return "" != e && !isNaN(e) && Math.round(e) == e
},
getByType: function(e, t) {
  for (var i = e.length, n = 0; n < i; n++)
    if (e[n].type === t)
      return e[n]
},
getBonusByName: function(e, t) {
  if (Array.isArray(e))
    for (var i = e.length, n = 0; n < i; n++)

```

```

        if (e[n].bonusName === t)
            return e[n];
        return null
    },
    resizeViewPort: function() {
        devWidth = window.innerWidth, devHeight = window.innerHeight;
        var e = document.getElementById("safariView");
        !(/Safari/.test(navigator.userAgent) && /Apple Computer/.test(navigator.vendor)) ||
568 != window.screen.height || 90 !== window.orientation && -90 != window.orientation ?
e.style.overflowY = "hidden" : (devHeight > 400 && (devHeight -= 200), window.scroll(0, 0)),
e.style.width = devWidth + "px", e.style.height = devHeight + "px"
    },
    getIntDivisor: function(e) {
        var t = e.toString(),
            i = t.indexOf(".");
        if (i > -1) {
            var n = t.length - i - 1;
            return Math.pow(10, n)
        }
        return 1
    },
    getMaxPower: function(e, t) {
        var i = e.toString(),
            a = t.toString(),
            o = n.instance.gameCurrencyData.decimalPlaces,
            s = i.length - i.indexOf("."),
            r = a.length - a.indexOf(".");
        return Math.max(o, s, r)
    },
    getQueryParamValue: function(e, t) {
        if (e && t) {
            var i;
            /^?.test(e) && (i = e.split("?")[1]);
            for (var n = i.split("&"), a = 0; a < n.length; a++)
                if (n[a].substring(0, n[a].indexOf("=")) == t)
                    return n[a].substring(n[a].indexOf("=") + 1)
        }
        return ""
    }
});
cc._RF.pop()
}, {
    SHS_GameGlobal: "SHS_GameGlobal"
}],
SHS_HudLocal: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "05612ncySVNZZP4mBYWedrI", "SHS_HudLocal");
    var n = e("SHS_GameGlobal"),
        a = e("SHS_GameController");
    cc.Class({
        extends: cc.Component,
        properties: {

```

```

    lblBalance: {
        default: null,
        type: cc.Node
    },
    lblTotalBet: {
        default: null,
        type: cc.Node
    },
    lblWin: {
        default: null,
        type: cc.Node
    },
    lblAutoPlay: {
        default: null,
        type: cc.Node
    },
    lblMaxBet: {
        default: null,
        type: cc.Node
    },
    _isLocalized: !1
},
onLoad: function() {},
update: function(e) {
    !this._isLocalized && n.instance.hud_strings && (this.localizedComponent(),
this._isLocalized = !0)
},
localizedComponent: function() {
    this.lblBalance.getComponent(cc.Sprite).spriteFrame =
n.instance.hud_strings.getSpriteFrame(n.instance.localSpecifier + ".balance"),
this.lblTotalBet.getComponent(cc.Sprite).spriteFrame =
n.instance.hud_strings.getSpriteFrame(n.instance.localSpecifier + ".total_bet"),
a.instance._isFreeSpin || n.instance.freePlayPro ? this.lblWin.getComponent(cc.Sprite).spriteFrame
= n.instance.hud_strings.getSpriteFrame(n.instance.localSpecifier + ".total_win") :
this.lblWin.getComponent(cc.Sprite).spriteFrame =
n.instance.hud_strings.getSpriteFrame(n.instance.localSpecifier + ".win"), cc.sys.isMobile ?
(a.instance.settingsBalanceLabel.getComponent(cc.Sprite).spriteFrame =
n.instance.hud_strings.getSpriteFrame(n.instance.localSpecifier + ".balance"),
a.instance.settingsBetLabel.getComponent(cc.Sprite).spriteFrame =
n.instance.hud_strings.getSpriteFrame(n.instance.localSpecifier + ".total_bet")) :
(this.lblAutoPlay.getComponent(cc.Sprite).spriteFrame =
n.instance.hud_strings.getSpriteFrame(n.instance.localSpecifier + ".auto"),
this.lblMaxBet.getComponent(cc.Sprite).spriteFrame =
n.instance.hud_strings.getSpriteFrame(n.instance.localSpecifier + ".maxbet"))
}
}), cc._RF.pop()
}, {
    SHS_GameController: "SHS_GameController",
    SHS_GameGlobal: "SHS_GameGlobal"
}],
SHS_InfoController: [function(e, t, i) {
    "use strict";

```

```

cc._RF.push(t, "61d2a1qc/FPd4K+YeZuGwlm", "SHS_InfoController");
var n = e("SHS_GameController");
cc.Class({
  extends: cc.Component,
  properties: {
    payablePage: {
      default: [],
      type: [cc.Node]
    },
    lblVersion: {
      default: null,
      type: cc.Label
    },
    _pageIndex: 0
  },
  onLoad: function() {
    this._pageIndex = 0, this.showWindow(this._pageIndex)
  },
  clickOnNext: function() {
    this._pageIndex++, this._pageIndex = Math.abs(this._pageIndex %
this.payablePage.length), this.showWindow(this._pageIndex),
n.instance._audioController.playMenuAudioEffect(!1)
  },
  clickOnPrev: function() {
    this._pageIndex--, -1 == this._pageIndex && (this._pageIndex = 8), this._pageIndex =
Math.abs(this._pageIndex % this.payablePage.length), this.showWindow(this._pageIndex),
n.instance._audioController.playMenuAudioEffect(!1)
  },
  showWindow: function(e) {
    for (var t = 0; t < this.payablePage.length; t++)
      this.payablePage[t].active = t === e
  },
  changeScreenView: function(e, t) {
    3 === this._previousWindowId && n.instance.updateOptionStatus(!1), 0 ===
this._previousWindowId && n.instance.updatePaytableStatus(!1), "0" == t ?
(this._currentWindowId = 0, n.instance.updatePaytableStatus(!0)) : "1" == t ?
this._currentWindowId = 1 : "2" == t ? this._currentWindowId = 2 : "3" == t &&
(this._currentWindowId = 3, n.instance.updateOptionStatus(!0)), this._previousWindowId =
this._currentWindowId, this.showWindow(parseInt(t)),
n.instance._audioController.playButtonSound()
  },
  onClose: function() {
    n.instance._hudController.onSettingsCloseClick()
  },
  clickBottomButton: function(e, t) {
    this._pageIndex = Math.abs(t % this.payablePage.length),
this.showWindow(this._pageIndex), n.instance._audioController.playMenuAudioEffect(!1)
  }
});
cc._RF.pop()
}, {
  SHS_GameController: "SHS_GameController"
}

```

```

}],
SHS_PSPAutoPlayController: [function(e, t, i) {
  "use strict";
  cc._RF.push(t, "5de56dqrMhI35DhU+m6ZnPP", "SHS_PSPAutoPlayController");
  var n = e("SHS_GameController");
  e("SHS_GameGlobal"), cc.Class({
    extends: cc.Component,
    properties: {
      _anim: null,
      _autoPlayCount: 0,
      _isShowWindow: !1,
      autoplayPanel: {
        default: null,
        type: cc.Layout
      },
      btnTopPref: {
        default: null,
        type: cc.Prefab
      },
      btnMidPref: {
        default: null,
        type: cc.Prefab
      },
      btnBottomPref: {
        default: null,
        type: cc.Prefab
      }
    },
    onLoad: function() {
      this._anim = this.node.getComponent(cc.Animation), this._isShowWindow = !1,
      this._anim.play("AP_out"), this.node.removeAllChildren(!0)
    },
    initAutoPlayPanel: function(e) {
      for (var t = 0; t < e.length; t++) {
        var i;
        i = 0 === t ? cc.instantiate(this.btnBottomPref) : t === e.length - 1 ?
cc.instantiate(this.btnTopPref) : cc.instantiate(this.btnMidPref), this.ButtonS =
i.getComponent(cc.Button), this.ButtonS.name = "btnAutoPlay_" + t;
        var n = new cc.Component.EventHandler;
        n.target = this.node, n.component = "SHS_PSPAutoPlayController", n.handler =
"onButtonClick", n.customEventData = e[t], this.ButtonS.clickEvents.push(n), this.lblNode =
this.ButtonS.node.getChildByName("value"), this.lbl = this.lblNode.getComponent(cc.Label),
this.lbl.string = e[t], this.node.addChild(i)
      }
    },
    togglePanel: function() {
      this._isShowWindow = !this._isShowWindow, this._isShowWindow ?
this._anim.play("AP_in") : this._anim.play("AP_out")
    },
    onButtonClick: function(e, t) {
      var i = e.target;
      cc.log("Auto play button clicked: " + i + " Event Data: " + t), this._autoPlayCount = t,

```

```

n.instance._audioController.playMenuAudioEffect(!0),
n.instance.setAutoPlay(this._autoPlayCount, 0, 0), n.instance._hudController.onAutoPlaySelected()
    }
});
cc._RF.pop()
}, {
  SHS_GameController: "SHS_GameController",
  SHS_GameGlobal: "SHS_GameGlobal"
}],
SHS_PSPHudController: [function(e, t, i) {
  "use strict";
  cc._RF.push(t, "5ee59SjCaJOAL4RENigMwBD", "SHS_PSPHudController");
  var n = e("SHS_GameController"),
      a = e("SHS_GameHelper"),
      o = e("SHS_GameGlobal");
  cc.Class({
    extends: cc.Component,
    properties: {
      _isHUDEnabled: !0,
      _coinValIndex: 0,
      _levelIndex: 0,
      lblBetValue: {
        default: null,
        type: cc.Label
      },
      lblWin: {
        default: null,
        type: cc.Node
      },
      lblWinAmount: {
        default: null,
        type: cc.Label
      },
      lblBalance: {
        default: null,
        type: cc.Label
      },
      spinButton: {
        default: null,
        type: cc.Button
      },
      stopSpinButton: {
        default: null,
        type: cc.Button
      },
      quickStopBtn: {
        default: null,
        type: cc.Node
      },
      quickTurboBtn: {
        default: null,
        type: cc.Node
      }
    }
  });
}
];

```

```
    },
    lblMaxBet: {
      default: null,
      type: cc.Button
    },
    betLines: {
      default: [],
      type: [cc.Node]
    },
    maxBetButton: {
      default: null,
      type: cc.Button
    },
    autoPlayButton: {
      default: null,
      type: cc.Button
    },
    settingsButton: {
      default: null,
      type: cc.Button
    },
    coinValUpButton: {
      default: null,
      type: cc.Button
    },
    coinValDownButton: {
      default: null,
      type: cc.Button
    },
    settingUI: {
      default: null,
      type: cc.Node
    },
    settingMobileUI: {
      default: null,
      type: cc.Node
    },
    payTableUI: {
      default: null,
      type: cc.Node
    },
    payTableMobileUI: {
      default: null,
      type: cc.Node
    },
    lblSpinCounter: {
      default: null,
      type: cc.Label
    },
    darkBackground: {
      default: null,
      type: cc.Node
    }
  }
}
```

```
    },
    hudDarkBackground: {
        default: null,
        type: cc.Node
    },
    BlackPanel: {
        default: null,
        type: cc.Node
    },
    winAnimationBox: {
        default: null,
        type: cc.Node
    },
    errorMessage: {
        default: null,
        type: cc.Node
    },
    comErrorMessage: {
        default: null,
        type: cc.Node
    },
    realityErrorMessage: {
        default: null,
        type: cc.Node
    },
    freePlayMessage: {
        default: null,
        type: cc.Node
    },
    lblFreeSpinMsgNode: {
        default: null,
        type: cc.Node
    },
    lblFreeSpinCounter: {
        default: null,
        type: cc.Label
    },
    lblMessageNode: {
        default: null,
        type: cc.Node
    },
    lblMoonMessageNode: {
        default: null,
        type: cc.Node
    },
    singleWinHolder: {
        default: null,
        type: cc.Node
    },
    isSpinButtonClick: !1,
    _payTableWindow: null
},
```

```

onLoad: function() {
    this._spinBtnClickCount = 0, (null == o.instance || null != o.instance && !
o.instance.freePlayPro) && this.showStopButton(!1), this.showQuickStopButton(!1),
cc.sys.isMobile ? this._payTableWindow =
this.payTableMobileUI.getComponent("SHS_PayTableController") : this._payTableWindow =
this.payTableUI.getComponent("SHS_PayTableController"), this.setSpinCounter(0, -1)
},
updateBetAmount: function(e) {
    this.lblBetValue.string = a.instance.ToCurrency(e), cc.sys.isMobile &&
(n.instance.settingsBetValue.string = a.instance.ToCurrency(e))
},
updateBalance: function(e) {
    this.lblBalance.string = a.instance.ToCurrency(e), cc.sys.isMobile &&
(n.instance.settingsBalanceValue.string = a.instance.ToCurrency(e))
},
playWinParticle: function() {
    n.instance._winAmount > 0 && (cc.sys.isMobile ? o.instance.mPortrait ?
this.winAnimationBox.getComponent(cc.Animation).play("Win2Balance_Portrait_Mobile") :
this.winAnimationBox.getComponent(cc.Animation).play("Win2Balance_Land_Mobile") :
this.winAnimationBox.getComponent(cc.Animation).play("Win2Balance_Desk"))
},
updateCoinValue: function(e) {
    o.instance.playerBetAmount = e * n.instance._betDivisor,
this.updateBetAmount(o.instance.playerBetAmount)
},
updateWinAmount: function(e) {
    if (e > 0) {
        var t = a.instance.getMaxPower(n.instance._cumulativeLWinLimit, e);
        t = Math.pow(10, t), n.instance._cumulativeLWinLimit =
(n.instance._cumulativeLWinLimit * t + e * t) / t, this.lblWinAmount.string =
a.instance.ToCurrency(e)
    } else
        this.lblWinAmount.string = ""
},
updateCoins: function(e) {},
updateMessage: function(e) {
    this.lblMessages.string = e, this.imgMessage.active = !1, this.lblMessages.node.active = !
0
},
updateWinLineMessage: function(e, t) {
    this.singleWinHolder.active = !1, this.lblMoonMessageNode.active = !1,
this.lblMessageNode.active = !0, this.clearWinLineMessage(),
this.lblMessageNode.getComponent("SHS_WinLableMessage").showWinLineMessage(e, t)
},
updateWinLineMoonMessage: function(e, t) {
    this.singleWinHolder.active = !1, this.lblMessageNode.active = !1,
this.lblMoonMessageNode.active = !0, this.clearWinLineMoonMessage(),
this.lblMoonMessageNode.getComponent("SHS_WinLableMessage").showWinLineMoonMessage
(e, t)
},
updateErrorMessage: function(e) {
    o.instance.mPortrait ?

```

```

(this.realityErrorMessage.getChildByName("error_message_box").scaleX = .65,
this.realityErrorMessage.setRotation(-90),
this.errorMessage.getChildByName("error_message_box").scaleX = .65,
this.errorMessage.setRotation(-90)) :
(this.realityErrorMessage.getChildByName("error_message_box").scaleX = 1,
this.realityErrorMessage.setRotation(0),
this.errorMessage.getChildByName("error_message_box").scaleX = 1,
this.errorMessage.setRotation(0)), n.instance._isRealityCheck ? this.realityErrorMessage.active = !
0 : this.errorMessage.active = !0, n.instance._isRealityCheck ? (" " != o.instance.rcHistoryUrl && " "
!= o.instance.rcHomeUrl ?
this.realityErrorMessage.getComponent(cc.Animation).play("Error_Msg_Box_Reality_OH") : " " !
= o.instance.rcHistoryUrl ?
this.realityErrorMessage.getComponent(cc.Animation).play("Error_Msg_Box_Reality_AH") : " " !
= o.instance.rcHomeUrl ?
this.realityErrorMessage.getComponent(cc.Animation).play("Error_Msg_Box_Reality_OH") :
this.realityErrorMessage.getComponent(cc.Animation).play("Error_Msg_Box_Reality"),
this.realityErrorMessage.getChildByName("lbl_SessionTime").getComponent(cc.Label).string =
o.instance._i18n.t("MSG.SESSION_TIME"),
this.realityErrorMessage.getChildByName("lbl_SessionTime_Amount").getComponent(cc.Label).s
tring = n.instance._sessionTime,
this.realityErrorMessage.getChildByName("lbl_LossAmount").getComponent(cc.Label).string =
o.instance._i18n.t("MSG.LOSS_AMOUNT"),
this.realityErrorMessage.getChildByName("lbl_Loss_Amount_Amount").getComponent(cc.Label).
string = n.instance._lossAmount,
this.realityErrorMessage.getChildByName("lbl_WinAmount").getComponent(cc.Label).string =
o.instance._i18n.t("MSG.WIN_AMOUNT"),
this.realityErrorMessage.getChildByName("lbl_Win_Amount_Amount").getComponent(cc.Label).
string = n.instance._realityWinAmount, n.instance._isRelaxDataAvailable = !1,
this.realityErrorMessage.getComponent("SHS_ErrorMessageController").showRealityMessage(e)
: (o.instance.freePlayPro && null == o.instance.fPProSpinsPlayed ?
(this.errorMessage.getChildByName("lbl_Free_Spins_Over").getComponent(cc.Label).string = e,
this.errorMessage.getChildByName("lbl_YouWin").getComponent(cc.Label).string =
o.instance._i18n.t("MSG.YOU_HAVE_WON"),
this.errorMessage.getChildByName("lbl_Win").getComponent(cc.Label).string =
a.instance.ToCurrency(n.instance._winAmount), this.updateWinAmount(n.instance._winAmount),
this.errorMessage.getComponent(cc.Animation).play("Error_Msg_Box_postfreeplaymsg"),
this.EnabledHUDElements(!1), n.instance._isFreePlayMsgOpen = !0) :
this.errorMessage.getComponent(cc.Animation).play("Error_Msg_Box_With_button"),
this.errorMessage.getComponent("SHS_ErrorMessageController").showErrorMessage(e))
},
updateFreePlayMessage: function(e) {
o.instance.mPortrait ?
(this.freePlayMessage.getChildByName("error_message_box").scaleX = .65,
this.freePlayMessage.setRotation(-90)) :
(this.freePlayMessage.getChildByName("error_message_box").scaleX = 1,
this.freePlayMessage.setRotation(0));
var t = this.freePlayMessage.getChildByName("PlayNow_button"),
i = this.freePlayMessage.getChildByName("PlayLater_button"),
a = this.freePlayMessage.getChildByName("NoThanks_button");
t.getChildByName("lbl_PlayNow").getComponent(cc.Label).string =
o.instance._i18n.t("HUD_MESSAGES.PLAY_NOW"),
i.getChildByName("lbl_PlayLater").getComponent(cc.Label).string =

```

```

o.instance._i18n.t("HUD_MESSAGES.PLAY_LATER"),
a.getChildByName("lbl_NoThanks").getComponent(cc.Label).string =
o.instance._i18n.t("HUD_MESSAGES.NO_THANKS"),
n.instance._hudController.EnabledHUDElements(!1), this.freePlayMessage.active = !0,
this.freePlayMessage.getComponent(cc.Animation).play("Error_Msg_Box_Replay"),
this.freePlayMessage.getComponent("SHS_ErrorMessageController").showErrorMessage(e),
n.instance._isGameIdle = !1, n.instance._isFreePlayMsgOpen = !0
    },
    updateComErrorMessage: function(e) {
        o.instance.mPortrait ?
(this.comErrorMessage.getChildByName("error_message_box").scaleX = .65,
this.comErrorMessage.setRotation(-90)) :
(this.comErrorMessage.getChildByName("error_message_box").scaleX = 1,
this.comErrorMessage.setRotation(0)), this.comErrorMessage.active = !0,
this.comErrorMessage.getComponent(cc.Animation).play("Error_Msg_Box"),
this.comErrorMessage.getComponent("SHS_ErrorMessageController").showComErrorMessage(e)
    },
    changeErrorButton: function(e) {

this.errorMessage.getChildByName("ok_button").getChildByName("lbl_Ok").getComponent(cc.Label).string = e
    },
    changeRealityErrorButton: function(e) {

this.realityErrorMessage.getChildByName("ok_button").getChildByName("lbl_Ok").getComponent(cc.Label).string = e
    },
    changeAHButton: function(e) {

this.realityErrorMessage.getChildByName("ah_button").getChildByName("lbl_AH").getComponent(cc.Label).string = e
    },
    changeOHButton: function(e) {

this.realityErrorMessage.getChildByName("oh_button").getChildByName("lbl_AH").getComponent(cc.Label).string = e
    },
    clearWinLineMessage: function() {
        this.lblMessageNode.getComponent("SHS_WinLableMessage").clearWinLineMessage()
    },
    clearWinLineMoonMessage: function() {

this.lblMoonMessageNode.getComponent("SHS_WinLableMessage").clearWinLineMoonMessage()
    },
    onMaxBetSelected: function() {
        if (!o.instance.isBlockedUI) {
            cc.sys.isMobile || (this.coinValDownButton.interactable = !0,
this.coinValUpButton.interactable = !1), n.instance._isMaxBet = !0,
n.instance._audioController.playButtonSound(), this._coinValIndex =
o.instance.betPreLineList.length - 1, o.instance._commonCoinValIndex = this._coinValIndex;
            var e = o.instance.betPreLineList[this._coinValIndex];

```

```

        o.instance.playerBetAmount = e * n.instance._betDivisor,
this.updateBetAmount(o.instance.playerBetAmount), cc.sys.isMobile ?
this.settingMobileUI.getComponent("SHS_SettingsController").refreshBetAmountUI() :
this.settingUI.getComponent("SHS_SettingsController").refreshBetAmountUI()
    }
},
onAutoPlaySelected: function() {
    o.instance.isBlockedUI || (this.EnabledHUDElements(!1),
n.instance.EnabledSplashBackground(!0), n.instance._audioController.playButtonSound(),
o.instance._isSettingWindowOpen = !0, o.instance.autoPlayDefault = 0, cc.sys.isMobile ?
(this.settingMobileUI.active = !0,
this.settingMobileUI.getComponent("SHS_SettingsController").showPage(2),
this.settingMobileUI.getChildByName("Btns").active = !0,
this.settingMobileUI.getChildByName("Btns").getChildByName("mobile_bottom_bar").active = !
1, this.settingMobileUI.getChildByName("BtnPortraitBar").active = !1,
this.settingMobileUI.getChildByName("autoPlayClose").active = !0,
this.settingMobileUI.getChildByName("btnPlay").active = !1,
n.instance._hudController.settingMobileUI.getComponent("SHS_SettingsController").changeSettingUIElementsPositions() : (this.settingUI.active = !0,
this.settingUI.getComponent("SHS_SettingsController").showPage(2),
this.settingUI.getChildByName("Btns").active = !1,
this.settingUI.getChildByName("autoPlayClose").active = !0,
this.settingUI.getChildByName("btnPlay").active = !1))
    },
onAutoPlayStopSelected: function() {
    n.instance._isAutoPlayPaused || (n.instance._audioController.playButtonSound(),
this.showStopButton(!1), o.instance.autoPlayDefault = 0, n.instance.requestAutoPlayCancel())
    },
resetAutoPlayWindow: function() {
    this.autoPlayPanel.getComponent("SHS_PSPAutoPlayController")._isShowWindow &&
(this.autoPlayPanel.getComponent("SHS_PSPAutoPlayController").togglePanel(),
this.spinButton.interactable = !0)
    },
onSpinBtnClick: function() {
    if (!o.instance.isBlockedUI) {
        n.instance._cumulativeLWinLimit = 0, n.instance._cumulativeLostLimit = 0,
a.instance.gameLog("On spinning clicked"), n.instance._isTurbo = !1,
n.instance._audioController.playReelStartSound(), o.instance.minSpinTime < 3e3 ?
n.instance._hudController.showTurboButton(!0) : n.instance._hudController.showTurboButton(!1);
        var e = this.getComponent(cc.Animation);
        e && e.play("hideSpinButton"), n.instance._isAutoPlay ? n.instance.requestAutoPlay()
: null !== n.instance && (n.instance.StartSpinning(), this._spinBtnClickCount = 0)
    }
},
setSpinButtonState: function(e) {
    this.spinButton.interactable = e
},
onSettingsShowClick: function() {
    o.instance.isBlockedUI || (this.EnabledHUDElements(!1),
n.instance.EnabledSplashBackground(!0), this._payTableWindow.setupPayLabelIds(),
n.instance._audioController.playButtonSound(), cc.sys.isMobile ? (this.settingMobileUI.active = !0,
this.settingMobileUI.getComponent("SHS_SettingsController").showPage(0),

```

```

this.settingMobileUI.getChildByName("Btns").active = !0,
this.settingMobileUI.getChildByName("Btns").getChildByName("mobile_bottom_bar").active = !
0, this.settingMobileUI.getChildByName("autoPlayClose").active = !1,
this.settingMobileUI.getChildByName("btnPlay").active = !1,
this.settingMobileUI.getComponent("SHS_SettingsController").updateSoundUI() :
(this.settingUI.active = !0, this.settingUI.getComponent("SHS_SettingsController").showPage(0),
this.settingUI.getChildByName("Btns").active = !0,
this.settingUI.getChildByName("autoPlayClose").active = !1,
this.settingUI.getChildByName("btnPlay").active = !1,
this.settingUI.getComponent("SHS_SettingsController").btnHistory.active = !0,
this.settingUI.getComponent("SHS_SettingsController").updateSoundUI()),
o.instance._isSettingWindowOpen = !0,
this._payTableWindow.refreshPayTable(this._coinValIndex), n.instance.updatePaytableStatus(!0))
},
onBetSettingsShowClick: function() {
o.instance.isBlockedUI || (this.EnabledHUDElements(!1),
n.instance.EnabledSplashBackground(!0), this._payTableWindow.setupPayLabelIds(),
n.instance._audioController.playButtonSound(), cc.sys.isMobile ? (this.settingMobileUI.active = !0,
this.settingMobileUI.getComponent("SHS_SettingsController").showPage(1),
this.settingMobileUI.getChildByName("Btns").active = !0,
this.settingMobileUI.getChildByName("Btns").getChildByName("mobile_bottom_bar").active = !
0, this.settingMobileUI.getChildByName("autoPlayClose").active = !1,
this.settingMobileUI.getChildByName("btnPlay").active = !1) : (this.settingUI.active = !0,
this.settingUI.getComponent("SHS_SettingsController").showPage(1),
this.settingUI.getChildByName("Btns").active = !0,
this.settingUI.getChildByName("autoPlayClose").active = !1,
this.settingUI.getChildByName("btnPlay").active = !1,
this.settingUI.getComponent("SHS_SettingsController").btnHistory.active = !0),
o.instance._isSettingWindowOpen = !0,
this._payTableWindow.refreshPayTable(this._coinValIndex), n.instance.updatePaytableStatus(!0))
},
onSettingsShowTrigger: function() {
this.onSettingsShowClick()
},
showOptionWindow: function(e) {
e.data ? (this.EnabledHUDElements(!1), cc.sys.isMobile ?
(this.settingMobileUI.getComponent("SHS_SettingsController").showSettingsFromGameCtrl(),
this.settingMobileUI.active = !0) :
(this.settingUI.getComponent("SHS_SettingsController").showSettingsFromGameCtrl(),
this.settingUI.active = !0)) : (this.EnabledHUDElements(!0), cc.sys.isMobile ?
(this.settingMobileUI.active = !1,
this.settingMobileUI.getComponent("SHS_SettingsController").pgView.enabled = !0) :
(this.settingUI.active = !1,
this.settingUI.getComponent("SHS_SettingsController").pgView.enabled = !0))
},
showPayTable: function(e) {
e.data ? (this.EnabledHUDElements(!1), cc.sys.isMobile ?
(this.settingMobileUI.getComponent("SHS_SettingsController").showPayTableFromGameCtrl(),
this.settingMobileUI.active = !0) :
(this.settingUI.getComponent("SHS_SettingsController").showPayTableFromGameCtrl(),
this.settingUI.active = !0)) : (this.EnabledHUDElements(!0), cc.sys.isMobile ?
(this.settingMobileUI.active = !1,

```

```

this.settingMobileUI.getComponent("SHS_SettingsController").pgView.enabled = !0) :
(this.settingUI.active = !1,
this.settingUI.getComponent("SHS_SettingsController").pgView.enabled = !0))
    },
    onCoinValueUp: function() {
        if (!o.instance.isBlockedUI) {
            if (this._coinValIndex++, this._coinValIndex <= o.instance.betPreLineList.length - 1) {
                this._coinValIndex === o.instance.betPreLineList.length - 1 &&
                (n.instance._isMaxBet = !0, cc.sys.isMobile || (this.coinValUpButton.interactable = !1)),
                this.coinValDownButton.interactable = !0, n.instance.resetLayout(),
                n.instance._audioController.playButtonSound();
                var e = o.instance.betPreLineList[this._coinValIndex],
                    t = a.instance.getIntDivisor(e);
                o.instance.playerBetAmount = parseInt(e * t) * n.instance._betDivisor / t,
                this.updateBetAmount(o.instance.playerBetAmount), cc.sys.isMobile ?
                this.settingMobileUI.getComponent("SHS_SettingsController").refreshBetAmountUI() :
                this.settingUI.getComponent("SHS_SettingsController").refreshBetAmountUI()
            } else
                cc.sys.isMobile || (this.coinValUpButton.interactable = !1,
                this.coinValDownButton.interactable = !0), this._coinValIndex = o.instance.betPreLineList.length -
                1, n.instance._isMaxBet = !0;
            o.instance._commonCoinValIndex = this._coinValIndex
        }
    },
    onCoinValueDown: function() {
        if (!o.instance.isBlockedUI) {
            if (this._coinValIndex--, this._coinValIndex >= 0) {
                cc.sys.isMobile || 0 !== this._coinValIndex ? !cc.sys.isMobile && this._coinValIndex
                > 0 && (this.coinValUpButton.interactable = !0, this.coinValDownButton.interactable = !0) :
                (this.coinValUpButton.interactable = !0, this.coinValDownButton.interactable = !1),
                n.instance._isMaxBet = !1, n.instance.resetLayout(), n.instance._isMaxBet = !1,
                n.instance._audioController.playButtonSound();
                var e = o.instance.betPreLineList[this._coinValIndex],
                    t = a.instance.getIntDivisor(e);
                o.instance.playerBetAmount = parseInt(e * t) * n.instance._betDivisor / t,
                this.updateBetAmount(o.instance.playerBetAmount), cc.sys.isMobile ?
                this.settingMobileUI.getComponent("SHS_SettingsController").refreshBetAmountUI() :
                this.settingUI.getComponent("SHS_SettingsController").refreshBetAmountUI()
            } else
                this._coinValIndex = 0, cc.sys.isMobile || (this.coinValUpButton.interactable = !0,
                this.coinValDownButton.interactable = !1);
            o.instance._commonCoinValIndex = this._coinValIndex
        }
    },
    EnabledCoinValButton: function() {
        var e = o.instance.betPreLineList.indexOf(o.instance.playerBetAmount);
        e === o.instance.betPreLineList.length - 1 ? cc.sys.isMobile ||
        (this.coinValUpButton.interactable = !1, this.coinValDownButton.interactable = !0) : 0 === e ?
        cc.sys.isMobile || (this.coinValUpButton.interactable = !0, this.coinValDownButton.interactable = !
        1) : cc.sys.isMobile || (this.coinValUpButton.interactable = !0, this.coinValDownButton.interactable
        = !0)
    },

```

```

onSettingsCloseClick: function() {
    o.instance.freePlayPro || n.instance._isFreeSpin || o.instance.isHistoryDataAvailable || !
n.instance._isGameIdle || n.instance._isAutoPlayPaused || this.EnabledHUDElements(!0),
n.instance.EnabledSplashBackground(!1), n.instance._audioController.playButtonSound(),
cc.sys.isMobile ? this.settingMobileUI.active = !1 : this.settingUI.active = !1,
o.instance._isSettingWindowOpen = !1, n.instance.updatePaytableStatus(!1)
},
onAutoPlayMenuCloseClick: function() {
    n.instance._isFreeSpin || this.EnabledHUDElements(!0),
n.instance.EnabledSplashBackground(!1), n.instance._audioController.playButtonSound(),
o.instance._isSettingWindowOpen = !1, cc.sys.isMobile ? this.settingMobileUI.active = !1 :
this.settingUI.active = !1
},
setSpinCounter: function(e, t) {
    if (-1 != t) {
        switch (t) {
            case 1:
                "FREE SPINS";
                break;
            case 2:
                "";
                break;
            case 3:
                "START FREE SPINS";
                break;
            default:
                ""
        }
        this.lblFreeSpinCounter.string = "", this.lblFreeSpinMsgNode.active = !1,
this.lblSpinCounter.string = e
    }
},
setFreeSpinCounter: function(e) {
    0 == e ? (this.lblFreeSpinCounter.string = "", this.lblFreeSpinMsgNode.active = !1,
n.instance._isAutoPlay && (this.stopSpinButton.node.active = !0)) :
(this.lblFreeSpinMsgNode.active = !0, this.lblFreeSpinCounter.string = e,
this.stopSpinButton.node.active = !1)
},
showStopButton: function(e) {
    this.stopSpinButton.node.active = e, this.stopSpinButton.node.interactable = !e, e || null !
== n.instance && (n.instance._autoPlayCounter = 0, n.instance._isAutoPlay = !1,
this.setSpinCounter(0, -1))
},
showQuickStopButton: function(e) {},
showTurboButton: function(e) {
    n.instance._isAutoPlay || (this.quickTurboBtn.active = e, this.quickTurboBtn.interactable
= !e)
},
EnabledHUDElements: function(e) {
    n.instance._isAutoPlayPaused && (e = !1), this._isHUDEnabled = e,
this.maxBetButton.interactable = e, this.autoPlayButton.interactable = e,
this.lblMaxBet.interactable = e;
}

```

```

    var t = !0;
    if (!n.instance._isGameIdle)
        t = !1;
    cc.sys.isMobile ?
this.settingMobileUI.getComponent("SHS_SettingsController").EnabledSettingElements(t) :
this.settingUI.getComponent("SHS_SettingsController").EnabledSettingElements(t),
this.spinButton.interactable = e, cc.sys.isMobile || (n.instance._isMaxBet ?
this.coinValUpButton.interactable = !1 : this.coinValUpButton.interactable = e), 0 ===
this._coinValIndex ? cc.sys.isMobile || (this.coinValDownButton.interactable = !1) :
cc.sys.isMobile || (this.coinValDownButton.interactable = e), o.instance.freePlayPro &&
this.EnabledHUDElementsInFreePlay(!1)
    },
    EnabledHUDElementsInFreePlay: function(e) {
    this._isHUDEnabled = e, this.maxBetButton.interactable = e, this.lblMaxBet.interactable
= e;
    if (!n.instance._isGameIdle)
        ;
    cc.sys.isMobile ?
this.settingMobileUI.getComponent("SHS_SettingsController").EnabledSettingElements(e) :
this.settingUI.getComponent("SHS_SettingsController").EnabledSettingElements(e),
cc.sys.isMobile || (this.coinValUpButton.interactable = e, this.coinValDownButton.interactable = e)
    },
    showLobbyButton: function(e) {},
    onLobbyButtonClick: function() {
    leanderGMApi.goToLobby()
    },
    toggleSpinAndStopBtn: function(e) {
    this.spinButton.active = !e, this.quickStopBtn.active = e
    },
    onTurboButtonClick: function() {
    n.instance._isTurbo = !0, n.instance._hudController.showTurboButton(!1)
    },
    onQuickStopButtonClick: function() {
    n.instance._isClicked = !0, n.instance._isClickQuickStop = !0, this.quickStopBtn.active
= !1, this.spinButton.active = !0
    },
    showHudWinMessage: function() {
    null != n.instance._winningLines && n.instance._winningLines.length > 0 && "RL" ==
n.instance._selectedScheme && !n.instance._isFreeSpin && !n.instance._isFirstLoad &&
n.instance.lblMessageNode.getComponent(cc.Animation).play("Paysbothways_In")
    },
    showHudMessages: function() {
    var e = [];
    if (this.lblMessageNode.stopAllActions(), this.lblMessageNode.active = !0,
o.instance.freePlayPro) {
    var t = o.instance.freePlayProObj.spins - o.instance.fPProSpinsPlayed;
    if (0 == t) {
    var i = o.instance._i18n.t("HUD_MESSAGES.FREE_SPIN_OVER");
    this.updateWinLineMessage(i.toLowerCase(), "X_FREE_PLAY")
    } else
this.updateWinLineMessage(o.instance._i18n.t("HUD_MESSAGES.X_FREE_PLAY"), {

```

```

        remainingCount: t
    }}, "X_FREE_PLAY")
    } else {
        var n = ["EXTEND_MATRIX", "PAY_BOTH_WAYS_MATRIX",
"ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX", "IS_ACTIVE_ALL",
"DURING_THE_FREE_SPIN", "TRIGGER_6_FREE_SPINS",
"FREE_SPIN_CAN_RETRIGGER"];
        e.push(cc.callFunc(function() {}));
        for (var a = 0; a < n.length; a++)
            e.push(this.showLineText(n[a]), e.push(cc.delayTime(3)));
        this.lblMessageNode.runAction(cc.repeatForever(cc.sequence(e)))
    }
},
showFreeSpinHudMessages: function() {
    var e = [];
    this.lblMessageNode.stopAllActions(), this.lblMessageNode.active = !0;
    var t = ["FREE_SPIN_CAN_RETRIGGER", "DURING_THE_FREE_SPIN"];
    e.push(cc.callFunc(function() {}));
    for (var i = 0; i < t.length; i++)
        e.push(this.showLineText(t[i]), e.push(cc.delayTime(6)));
    this.lblMessageNode.runAction(cc.repeatForever(cc.sequence(e)))
},
showLineText: function(e) {
    var t = this;
    return cc.callFunc(function() {
        t.updateWinLineMessage(o.instance._i18n.t("HUD_MESSAGES." + e), e)
    })
},
handleDuplicateButtonList: function(e) {}
});
cc._RF.pop()
}, {
    SHS_GameController: "SHS_GameController",
    SHS_GameGlobal: "SHS_GameGlobal",
    SHS_GameHelper: "SHS_GameHelper"
}],
SHS_PayTableController: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "331e7nSKahESLtrfh2tnPFh", "SHS_PayTableController");
    var n = e("SHS_GameController"),
        a = e("SHS_GameHelper"),
        o = e("SHS_GameGlobal");
    cc.Class({
        extends: cc.Component,
        properties: {
            _coinValIndex: 0,
            payItemValues: {
                default: [],
                type: [cc.Label]
            },
        },
        payItemValuesPortrait: {
            default: [],

```

```

    type: [cc.Label]
  },
  lblCoinValuePotrait: {
    default: [],
    type: [cc.Label]
  },
  lblCoinValue: {
    default: [],
    type: [cc.Label]
  },
  pages: {
    default: [],
    type: [cc.Node]
  },
  _payLabelIds: [],
  isViewCreated: !1,
  _isLocalized: !1
},
statics: {
  instance: null
},
onLoad: function() {
  cc.sys.isMobile || this.showPage(0)
},
showPage: function(e) {
  for (var t = 0; t < this.pages.length; t++)
    this.pages[t].active = !1;
  this.pages[e].active = !0
},
setPayTable: function() {
  var e = o.instance.betPreLineList[this._coinValIndex];
  if (this.updateCoinValue(e, !this.isViewCreated) {
    this.isViewCreated = !0;
    for (var t in o.instance.payTableInfo) {
      var i = o.instance.payTableInfo[t],
          s = i.symbol,
          r = parseInt(i.pay),
          l = s + "" + parseInt(i.n);
      null != this._payLabelIds[l] && (r = o.instance.betPreLineList[this._coinValIndex] *
r / n.instance._baseBet, this.payItemValues[this._payLabelIds[l]].string = a.instance.ToCurrency(r),
cc.sys.isMobile && (this.payItemValuesPotrait[this._payLabelIds[l]].string =
a.instance.ToCurrency(r)))
    }
    if (null != o.instance._scatterPayouts)
      for (var c = 0; c < o.instance._scatterPayouts.length; c++) {
        l = "FG" + c;
        if (null != this._payLabelIds[l]) {
          r = parseInt(o.instance._scatterPayouts[c]);
          r = o.instance.betPreLineList[this._coinValIndex] * r,
this.payItemValues[this._payLabelIds[l]].string = a.instance.ToCurrency(r), cc.sys.isMobile &&
(this.payItemValuesPotrait[this._payLabelIds[l]].string = a.instance.ToCurrency(r))
        }
      }
  }
}

```

```

    }
  }
},
setupPayLabelIds: function() {
  for (var e = 0; e < this.payItemValues.length; e++) {
    var t = this.payItemValues[e].getComponent(cc.Label).name.replace("<Label>", "");
    this._payLabelIds[t] = e
  }
},
onCoinValueUp: function() {
  if (!o.instance.isBlockedUI)
    if (this._coinValIndex < o.instance.betPreLineList.length - 1) {
      this.isViewCreated = !1, n.instance.resetLayout(),
n.instance._audioController.playMenuAudioEffect(!0), this._coinValIndex++;
      var e = o.instance.betPreLineList[this._coinValIndex];
      n.instance._hudController.updateCoinValue(e),
n.instance._hudController._coinValIndex = this._coinValIndex, this.setPayTable(),
o.instance.playerBetAmount = e * n.instance._betDivisor,
n.instance._hudController.updateBetAmount(o.instance.playerBetAmount),
n.instance._hudController.EnabledCoinValButton()
    } else
      this._coinValIndex = o.instance.betPreLineList.length - 1
  },
onCoinValueDown: function() {
  if (!o.instance.isBlockedUI)
    if (this._coinValIndex > 0) {
      this.isViewCreated = !1, n.instance.resetLayout(),
n.instance._audioController.playMenuAudioEffect(!0), this._coinValIndex--;
      var e = o.instance.betPreLineList[this._coinValIndex];
      n.instance._hudController.updateCoinValue(e),
n.instance._hudController._coinValIndex = this._coinValIndex, this.setPayTable(),
o.instance.playerBetAmount = e * n.instance._betDivisor,
n.instance._hudController.updateBetAmount(o.instance.playerBetAmount),
n.instance._hudController.EnabledCoinValButton()
    } else
      this._coinValIndex = 0
  },
updateCoinValue: function(e) {
  for (var t in e *= n.instance._betDivisor, this.lblCoinValue)
    this.lblCoinValue[t].string = a.instance.ToCurrency(e);
  if (this.lblCoinValuePotrait.length > 0)
    for (var t in this.lblCoinValuePotrait)
      this.lblCoinValuePotrait[t].string = a.instance.ToCurrency(e)
},
changePayTableView: function(e, t) {
  this.showPage(parseInt(t)), n.instance._audioController.playButtonSound()
},
refreshPayTable: function(e) {
  this._coinValIndex = e, this.isViewCreated = !1;
  var t = o.instance.betPreLineList[this._coinValIndex];
  this.updateCoinValue(t), this.setPayTable()
}
}

```

```

    });
    cc._RF.pop()
  }, {
    SHS_GameController: "SHS_GameController",
    SHS_GameGlobal: "SHS_GameGlobal",
    SHS_GameHelper: "SHS_GameHelper"
  }],
SHS_PaytableLocal: [function(e, t, i) {
  "use strict";
  cc._RF.push(t, "96b11YKT2dAoauPcTfCy8Qo", "SHS_PaytableLocal");
  var n = e("SHS_GameGlobal");
  e("SHS_GameHelper");
  cc.Class({
    extends: cc.Component,
    properties: {
      PAYTABLE1: {
        default: null,
        type: cc.Node
      },
      PAYTABLE2: {
        default: null,
        type: cc.Node
      },
      PAYTABLE3: {
        default: null,
        type: cc.Node
      },
      PAYTABLE4: {
        default: null,
        type: cc.Node
      },
      PAYTABLE5: {
        default: null,
        type: cc.Node
      },
      PAYS_BOTH_WAYS: {
        default: null,
        type: cc.Node
      },
      EXTENDS_MATRIX: {
        default: null,
        type: cc.Node
      },
      ACTIVATES_MIXED: {
        default: null,
        type: cc.Node
      },
      TRIGGER_6_FREE_SPIN: {
        default: null,
        type: cc.Node
      },
      AND_PAYS: {

```

```
    default: null,
    type: cc.Node
  },
  WHEN_THE_RED: {
    default: null,
    type: cc.Node
  },
  COIN_VAL1: {
    default: null,
    type: cc.Node
  },
  MALFUNCTIONTEXT1: {
    default: null,
    type: cc.Node
  },
  COIN_VAL2: {
    default: null,
    type: cc.Node
  },
  MALFUNCTIONTEXT2: {
    default: null,
    type: cc.Node
  },
  COIN_VAL3: {
    default: null,
    type: cc.Node
  },
  MALFUNCTIONTEXT3: {
    default: null,
    type: cc.Node
  },
  MALFUNCTIONTEXT4: {
    default: null,
    type: cc.Node
  },
  MALFUNCTIONTEXT5: {
    default: null,
    type: cc.Node
  },
  MALFUNCTIONTEXT6: {
    default: null,
    type: cc.Node
  },
  GAME_RULES: {
    default: null,
    type: cc.Node
  },
  GAME_RULE1: {
    default: null,
    type: cc.Node
  },
  GAME_RULE_LIST_01: {
```

```

        default: null,
        type: cc.Node
    },
    GAME_RULE_LIST_02: {
        default: null,
        type: cc.Node
    },
    GAME_RULE_LIST_03: {
        default: null,
        type: cc.Node
    },
    btnHistory: {
        default: null,
        type: cc.Node
    },
    _isLocalized: !!
},
update: function(e) {
    !this._isLocalized && n.instance.auto_strings && n.instance.paytable_strings &&
n.instance.gamerules_strings && n.instance.sound_strings && (this.localizedComponent(),
this._isLocalized = !0)
},
localizedComponent: function() {
    this.PAYTABLE1.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".paytable"),
this.PAYTABLE2.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".paytable"),
this.PAYTABLE3.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".paytable"),
this.PAYTABLE4.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".paytable"),
this.PAYTABLE5.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".paytable"),
this.PAYS_BOTH_WAYS.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".pays_both_ways"),
this.EXTENDS_MATRIX.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".extends_matrix"),
this.ACTIVATES_MIXED.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".activates_mixed"),
this.TRIGGER_6_FREE_SPIN.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".trigger_6"),
this.AND_PAYS.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".5x5"),
this.WHEN_THE_RED.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".when_the_red_moon"),
this.COIN_VAL1.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".coin_value"),
this.COIN_VAL2.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".coin_value"),
this.COIN_VAL3.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".coin_value"),
this.GAME_RULES.getComponent(cc.Sprite).spriteFrame =

```

```

n.instance.gamerules_strings.getSpriteFrame(n.instance.localSpecifier + ".game_rules"),
cc.sys.isMobile ? (this.MALFUNCTIONTEXT1.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction_portrait"),
this.MALFUNCTIONTEXT2.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction_portrait"),
this.GAME_RULE_LIST_01.getComponent(cc.Sprite).spriteFrame =
n.instance.gamerules_strings.getSpriteFrame(n.instance.localSpecifier + ".gamerules_portrait")) :
(this.GAME_RULE1.getComponent(cc.Sprite).spriteFrame =
n.instance.gamerules_strings.getSpriteFrame(n.instance.localSpecifier + ".game_rules"),
this.MALFUNCTIONTEXT1.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction"),
this.MALFUNCTIONTEXT2.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction"),
this.MALFUNCTIONTEXT3.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction"),
this.MALFUNCTIONTEXT4.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction"),
this.MALFUNCTIONTEXT5.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction"),
this.MALFUNCTIONTEXT6.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction"),
this.GAME_RULE_LIST_01.getComponent(cc.Sprite).spriteFrame =
n.instance.gamerules_strings.getSpriteFrame(n.instance.localSpecifier + ".gamerules_list_01"),
this.GAME_RULE_LIST_02.getComponent(cc.Sprite).spriteFrame =
n.instance.gamerules_strings.getSpriteFrame(n.instance.localSpecifier + ".gamerules_list_02"),
this.GAME_RULE_LIST_03.getComponent(cc.Sprite).spriteFrame =
n.instance.gamerules_strings.getSpriteFrame(n.instance.localSpecifier + ".gamerules_list_03")),
this.btnHistory.getComponent(cc.Sprite).spriteFrame =
n.instance.gamerules_strings.getSpriteFrame(n.instance.localSpecifier + ".history")
    }
    }), cc._RF.pop()
}, {
    SHS_GameGlobal: "SHS_GameGlobal",
    SHS_GameHelper: "SHS_GameHelper"
}],
SHS_PopupWindowController: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "c35750tX4ZJ3b2hvKf1vzni", "SHS_PopupWindowController");
    var n = e("SHS_GameController"),
        a = e("SHS_GameGlobal");
    cc.Class({
        extends: cc.Component,
        properties: {},
        onCloseSettings: function() {
            n.instance._hudController.onSettingsCloseClick()
        },
        onCloseDemoPayTable: function() {
            n.instance._hudController.onCloseDemoPayTable()
        },
        onCloseAutoPlay: function() {
            n.instance._hudController.onAutoPlayMenuCloseClick()
        },
    }

```

```

    onCloseGameRule: function() {
        n.instance._hudController.onGameRuleCloseClick()
    },
    onCloseFreeSpinMessage: function() {
        n.instance._isAutoPlayPaused || (n.instance.preFreeSpinMessage(!1),
n.instance._audioController.playMusic(a.instance.MUSIC_FREE_SPIN))
    }
});
cc._RF.pop()
}, {
    SHS_GameController: "SHS_GameController",
    SHS_GameGlobal: "SHS_GameGlobal"
}],
SHS_RadioBtn: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "1a500fMfStKE4pIUrbGCNDP", "SHS_RadioBtn");
    var n = e("SHS_GameController");
    cc.Class({
        extends: cc.Component,
        properties: {
            _currentState: !0,
            OnState: {
                default: null,
                type: cc.Sprite
            },
            OffState: {
                default: null,
                type: cc.Sprite
            }
        },
        On: function() {
            n.instance._audioController.playButtonSound(), this.OnState.node.active = !0,
this.OffState.node.active = !1, this._currentState = !0
        },
        Off: function() {
            n.instance._audioController.playButtonSound(), this.OnState.node.active = !1,
this.OffState.node.active = !0, this._currentState = !1
        },
        stateChanged: function() {
            this._currentState ? this.Off() : this.On()
        },
        currentState: function() {
            return this._currentState
        }
    });
    cc._RF.pop()
}, {
    SHS_GameController: "SHS_GameController"
}],
SHS_RelaxError: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "7c645z6XXFE2bNy2rA4HiwV", "SHS_RelaxError");

```

```

cc.Class({
  extends: cc.Component,
  statics: {
    comErrorCodes: [505, 519, 521, 525, 528, 533, 549, 550, 551],
    ERRORS: {
      501: "System error. Please contact support with error code 501.",
      502: "Session error. Please contact support with error code 502.",
      503: "Session error. Please contact support with error code 503.",
      504: "System error. Please contact support with error code 504.",
      505: "Invalid game progress. Please contact support with error code 505.",
      506: "Not enough funds for chosen betsize",
      507: "Invalid bet. Please contact support with error code 507.",
      508: "System command error. Please contact support with error code 508.",
      509: "System error. Please contact support with error code 509.",
      516: "System error. Please contact support with error code 516.",
      517: "System error. Please contact support with error code 517.",
      518: "System error. Please contact support with error code 518.",
      519: "Invalid login. Please contact support with error code 519.",
      520: "Game is active in another browser.",
      521: "Technical error during login",
      522: "System error. Please contact support with error code 522.",
      523: "System error. Please contact support with error code 523.",
      524: "Game disabled.",
      525: "Currency error. Please contact support with error code 525.",
      526: "Login error. Please contact support with error code 526.",
      527: "Freespin error. Please contact support with error code 527.",
      528: "Invalid URL.",
      531: "Blocked from playing",
      532: "Freespin error. Please contact support with error code 532.",
      533: "Token error. Please contact support with error code 533.",
      534: "Login attempt comes from blocked region",
      535: "Game version too old.",
      536: "Token error. Please contact support with error code 536.",
      537: "Login error. Please contact support with error code 537.",
      538: "Jackpot error. Please contact support with error code 538.",
      539: "Jackpot error. Please contact support with error code 539.",
      540: "Player has active game in another server.",
      541: "System error. Please contact support with error code 541.",
      542: "Version error. Please contact support with error code 542.",
      543: "System error. Please contact support with error code 543.",
      544: "Spending budget exceeded",
      545: "Reality check loss limit exceeded",
      546: "Reality check session length exceeded",
      547: "Currency error. Please contact support with error code 547.",
      548: "Playtech reality check session expired",
      549: "Token error. Please contact support with error code 549.",
      550: "Win cap reached",
      551: "System error. Please contact support with error code 551."
    }
  }
});
cc._RF.pop()

```

```

}, {}],
SHS_RelaxHandler: [function(e, t, i) {
  "use strict";
  cc._RF.push(t, "cfaf5q04UFI5Iygmj382y8U", "SHS_RelaxHandler");
  var n = e("SHS_Relax"),
      a = e("SHS_GameGlobal"),
      o = e("SHS_GameHelper"),
      s = e("SHS_RelaxError");
  cc.Class({
    extends: cc.Component,
    properties: {
      _gameEngine: null
    },
    initHandler: function() {
      this._gameEngine = new n, this._gameEngine.onCommunicationErrorCallbackHandler =
this.onCommunicationError.bind(this), this._gameEngine.onServerErrorCallbackHandler =
this.onRelaxError.bind(this), a.instance.mode = RELAX_CONFIG.mode, a.instance.partnerId =
RELAX_CONFIG.partnerId, "" != a.instance.urlPartnerId && a.instance.urlPartnerId !=
RELAX_CONFIG.partnerId && (a.instance.partnerId = a.instance.urlPartnerId), void 0 !==
RELAX_CONFIG.channel && "" != RELAX_CONFIG.channel && (a.instance.channel =
RELAX_CONFIG.channel), a.instance.gameServerAPIEndpointURL =
RELAX_CONFIG.gameServerAPIEndpointURL, a.instance.tokenAPIEndpointURL =
RELAX_CONFIG.tokenAPIEndpointURL, a.instance.gameReplayServiceURL =
RELAX_CONFIG.gameReplayServiceURL, a.instance.minSpinTime =
RELAX_CONFIG.minSpinTime;
      var e = RELAX_CONFIG.threeSecMarket,
          t = a.instance.urlJurisdiction;
      "" != t && -1 != e.indexOf(t) ? a.instance.urlMinSpinTime = 4e3 :
a.instance.urlMinSpinTime = 1e3, "" != a.instance.urlMinSpinTime && (a.instance.minSpinTime =
a.instance.urlMinSpinTime), this._gameEngine.initializeGame(a.instance.gameId, a.instance.mode,
a.instance.clientId, a.instance.partnerId, a.instance.ticketId, a.instance.currencyId,
a.instance.channel, a.instance.gameServerAPIEndpointURL, a.instance.tokenAPIEndpointURL,
a.instance.gameReplayServiceURL, a.instance.moneyMode)
    },
    onCommunicationError: function(e) {
      this._gameController.gameNoneFatalErrorHandler(e, " COMMUNICATION ERROR ", !
1)
    },
    onRelaxError: function(e, t) {
      this._gameController._isRemoveButton = !1, t = s.ERRORS[e] || t,
this._gameController.gameNoneFatalErrorHandler(e, t, !1)
    },
    assignController: function(e) {
      this._gameController = e
    },
    OpenPlayRequest: function() {
      var e = this;
      "fun" == a.instance.moneyMode ? e.performUserFunMoneyLogin() :
a.instance.isRmLogin && this._gameEngine.getTokenRequest(function(t, i, n) {
        e.performUserLogin()
      })
    },
  },

```

```

performUserLogin: function() {
    var e = this;
    this._gameEngine.performUserLogin(function(t) {
        e.startGamePlaySession(t)
    })
},
performUserFunMoneyLogin: function() {
    var e = this;
    this._gameEngine.performUserFunMoneyLogin(function(t) {
        e.startGamePlaySession(t)
    })
},
performUserDemoRmLogin: function() {
    var e = this;
    this._gameEngine.performUserDemoRmLogin(function(t) {
        e.startGamePlaySession(t)
    })
},
startGamePlaySession: function(e) {
    this.extractInitializeGameResponse(e)
},
sendPlayRequest: function(e, t, i, n) {
    var a = this;
    this._gameEngine.gamePlay(e, t, i, n, function(e) {
        a._extractBetResultsResponse(e)
    })
},
sendRepPlayRequest: function() {
    this._gameEngine.sendRepPlayRequest(a.instance.gameReplayServiceURL,
a.instance._rePlayLang)
},
extractInitializeGameResponse: function(e) {
    a.instance.gameCurrencyData = {
        decimalPlaces: 2,
        currencyPrefix: e.stats.currency,
        decimalSeparator: ".",
        groupingSeparator: ",",
        currencySuffix: ""
    }, a.instance.sessionId = this._gameEngine._sessionId, a.instance.gameId =
this._gameEngine._gameId, a.instance.userId = this._gameEngine._userId,
a.instance.isGameInilized = !0, this._gameController.gameInitializeHandler(e)
},
_extractBetResultsResponse: function(e) {
    this._gameController.getBetResponseHandler(e)
},
saveAutoPlayValue: function(e, t, i) {
    var n = {
        betParams: {
            playerBet: a.instance.playerBetAmount,
            sessionId: this._gameEngine._sessionId
        },
        autoPValParams: {

```

```

        autoPlayDefault: e,
        winLimit: t,
        lossLimit: a.instance.autoPlayLostAmount,
        sessionId: this._gameEngine._sessionId
    },
    freeSpinData: {
        freeSpinIndex: -1,
        displayFreeSpinCount: -1,
        beforeFreeSpinReelLayout: a.instance.freeSpinBeforeReel,
        beforeFreeSpinBonusReelLayout: a.instance.freeSpinBeforeBonusReelLayout,
        beforeFreeSpinSelectedReelSet: a.instance.freeSpinBeforeSelectedReelSet
    },
    soundSetting: {
        backGroundSound: a.instance.audioMusic,
        animationSound: a.instance.audioSfx
    }
};
this.saveData(n)
},
getData: function() {
    var e = this;
    this._gameEngine.getData(function(t) {
        e._extractSavedData(t)
    })
},
_extractSavedData: function(e) {
    this._gameController.extractSavedData(e)
},
saveFreeSpin: function(e, t, i) {
    var n = {
        betParams: {
            playerBet: a.instance.playerBetAmount,
            sessionId: this._gameEngine._sessionId
        },
        autoPValParams: {
            autoPlayDefault: 0,
            winLimit: 75,
            lossLimit: a.instance.autoPlayLostAmount,
            sessionId: this._gameEngine._sessionId
        },
        freeSpinData: {
            freeSpinIndex: e,
            displayFreeSpinCount: t,
            beforeFreeSpinReelLayout: a.instance.freeSpinBeforeReel,
            beforeFreeSpinBonusReelLayout: a.instance.freeSpinBeforeBonusReelLayout,
            beforeFreeSpinSelectedReelSet: a.instance.freeSpinBeforeSelectedReelSet
        },
        soundSetting: {
            backGroundSound: a.instance.audioMusic,
            animationSound: a.instance.audioSfx
        }
    };
};

```

```

    this.saveData(n)
  },
  saveSoundSetting: function() {
    var e = {
      betParams: {
        playerBet: a.instance.playerBetAmount,
        sessionId: this._gameEngine._sessionId
      },
      autoPValParams: {
        autoPlayDefault: 0,
        winLimit: 75,
        lossLimit: a.instance.autoPlayLostAmount,
        sessionId: this._gameEngine._sessionId
      },
      freeSpinData: {
        freeSpinIndex: -1,
        displayFreeSpinCount: -1,
        beforeFreeSpinReelLayout: a.instance.freeSpinBeforeReel,
        beforeFreeSpinBonusReelLayout: a.instance.freeSpinBeforeBonusReelLayout,
        beforeFreeSpinSelectedReelSet: a.instance.freeSpinBeforeSelectedReelSet
      },
      soundSetting: {
        backGroundSound: a.instance.audioMusic,
        animationSound: a.instance.audioSfx
      }
    };
    this.saveData(e)
  },
  saveData: function(e) {
    this._gameEngine.saveGameData(e, function() {})
  },
  setRestoreState: function(e, t) {
    this._gameEngine.setRestoreState(e, t, function(e) {})
  },
  gameFinished: function(e) {
    this._gameEngine.gameFinished(e, function(e) {})
  },
  gameResetRealityCheck: function(e, t) {
    this._gameEngine.gameResetRealityCheck(e, t, function(e) {})
  },
  gameSetRealityCheck: function(e, t, i) {
    this._gameEngine.setRealityCheck(e, t, i, function(e) {})
  },
  sendFreePlayProActionRequest: function(e) {
    var t = this;
    this._gameEngine.setFreeSpinAction(e, function(e) {
      t._extractFreePlayProActionResponse(e)
    })
  },
  _extractFreePlayProActionResponse: function(e) {
    o.instance.gameLog("Free play promo response : " + JSON.stringify(e))
  },

```

```

    getSessionResult: function() {
        this._gameEngine.getSessionResult(function(e) {
            console.log(JSON.stringify(e))
        })
    }
});
cc._RF.pop()
}, {
    SHS_GameGlobal: "SHS_GameGlobal",
    SHS_GameHelper: "SHS_GameHelper",
    SHS_Relax: "SHS_Relax",
    SHS_RelaxError: "SHS_RelaxError"
}],
SHS_Relax: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "f2bd7A1g5dDnIxGtz7MnOJo", "SHS_Relax");
    var n = "function" == typeof Symbol && "symbol" == typeof Symbol.iterator ? function(e) {
        return typeof e
    } : function(e) {
        return e && "function" == typeof Symbol && e.constructor === Symbol && e !==
Symbol.prototype ? "symbol" : typeof e
    };
    cc.Class({
        extends: cc.Component,
        properties: {
            _ticketId: "",
            _partnerId: 1,
            _gameId: "",
            _mode: "",
            _clientId: "",
            _currencyId: "",
            _channel: "",
            _moneyMode: "",
            _lastToken: "",
            _userId: "",
            _sessionId: "",
            _gameController: null,
            _gameServerAPIEndpointURL: "",
            _tokenAPIEndpointURL: "",
            _gameReplayServiceURL: "",
            onCommunicationErrorCallbackHandler: null,
            onServerErrorCallbackHandler: null
        },
        assignController: function(e) {
            this._gameController = e
        },
        initializeGame: function(e, t, i, n, a, o, s, r, l, c, d) {
            this._ticketId = a, this._partnerId = n, this._gameId = e, this._mode = t, this._clientId = i,
this._currencyId = o, this._channel = s, this._gameServerAPIEndpointURL = r,
this._tokenAPIEndpointURL = l, this._gameReplayServiceURL = c, this._moneyMode = d
        },
        _displayDebugMessage: function(e) {}},

```

```

    _sendServerRequest: function(e, t, i) {
        var n = cc.loader.getXMLHttpRequest();
        n.open("POST", e, !0), n.setRequestHeader("Content-Type", "application/json"),
n.onerror = function() {
            i(n, 0 == n.status ? 404 : n.status)
        }, n.onreadystatechange = function() {
            4 == n.readyState && 200 == n.status && i(n, n.status)
        }, n.send(t)
    },
    getTokenRequest: function(e) {
        var t = {
            ticket: this._ticketId,
            partnerid: this._partnerId,
            channel: this._channel,
            gameref: this._gameId,
            mode: this._mode,
            clientid: this._clientId,
            currency: this._currencyId
        },
        i = this;
        this._sendServerRequest(i._tokenAPIEndpointURL, JSON.stringify(t), function(t, n) {
            if (200 != n)
                i.onCommunicationErrorCallbackHandler(n);
            else if (" " != t.response) {
                var a = JSON.parse(t.response);
                "ok" == a.status ? (i._lastToken = a.response.token, i._userId = a.response.userid,
i._displayDebugMessage("New token received from game play server : " + i._lastToken),
i._displayDebugMessage("User id received from game play server : " + i._userId), e(i._lastToken,
i._userId, a)) : i.onServerErrorCallbackHandler(a.errorCode, a.errorMessage)
            } else
                i.onCommunicationErrorCallbackHandler(400)
        })
    },
    performUserLogin: function(e) {
        var t = {
            token: this._lastToken,
            userid: this._userId,
            mode: this._mode,
            partnerid: this._partnerId,
            clientid: this._clientId,
            g: this._gameId,
            channel: this._channel
        },
        i = this;
        this._sendServerRequest(i._gameServerAPIEndpointURL + "/rmlogin",
JSON.stringify(t), function(t, n) {
            if (200 != n)
                i.onCommunicationErrorCallbackHandler(n);
            else if (" " != t.response) {
                var a = JSON.parse(t.response);
                void 0 === a.errorCode ? (i._sessionId = a.sid, i._displayDebugMessage("Session id
received from game play server : " + i._sessionId), e(a)) :

```

```

i.onServerErrorCallbackHandler(a.errorCode, a.errorMessage)
    } else
        i.onCommunicationErrorCallbackHandler(400)
    })
},
performUserFunMoneyLogin: function(e) {
    var t = {
        g: this._gameId,
        channel: this._channel,
        currency: this._currencyId
    },
    i = this;
    this._sendServerRequest(i._gameServerAPIEndpointURL + "/funmoneylogin",
JSON.stringify(t), function(t, n) {
        if (200 != n)
            i.onCommunicationErrorCallbackHandler(n);
        else if (" " != t.response) {
            var a = JSON.parse(t.response);
            void 0 === a.errorCode ? (i._sessionId = a.sid, i._displayDebugMessage("Session id
received from game play server : " + i._sessionId), e(a)) :
i.onServerErrorCallbackHandler(a.errorCode, a.errorMessage)
        } else
            i.onCommunicationErrorCallbackHandler(400)
    })
},
performUserDemoRmLogin: function(e) {
    var t = {
        g: this._gameId,
        channel: this._channel,
        currency: this._currencyId
    },
    i = this;
    this._sendServerRequest(i._gameServerAPIEndpointURL + "/demormlogin",
JSON.stringify(t), function(t, n) {
        if (200 != n)
            i.onCommunicationErrorCallbackHandler(n);
        else if (" " != t.response) {
            var a = JSON.parse(t.response);
            void 0 === a.errorCode ? (i._sessionId = a.sid, i._displayDebugMessage("Session id
received from game play server : " + i._sessionId), e(a)) :
i.onServerErrorCallbackHandler(a.errorCode, a.errorMessage)
        } else
            i.onCommunicationErrorCallbackHandler(400)
    })
},
_sendPostRequest: function(e, t, i) {
    var a = this;
    (function i(o, s) {
        a._sendServerRequest(t, JSON.stringify(e()), function(e, t) {
            if (200 != t)
                a.onCommunicationErrorCallbackHandler(t);
            else if (" " != e.response) {

```

```

        var r = JSON.parse(e.response);
        if (void 0 === r.errorCode) {
            if (!o && n(r.errorCode) != n(void 0) && 503 == r.errorCode)
                return "fun" == this._moneyMode ? void
a.performUserFunMoneyLogin(function(e) {
                    i(!0, s)
                }) : void (SHS_GameGlobal.instance.isRmLogin &&
a.getTokenRequest(function(e, t, n) {
                    "" == e ? a.onCommunicationErrorCallbackHandler(401) :
a.performUserLogin(function(e) {
                            i(!0, s)
                        })
                    });
                s(r)
            } else
                a.onServerErrorCallbackHandler(r.errorCode, r.errorMessage)
        } else
            a.onCommunicationErrorCallbackHandler(400)
    })
})(!1, i)
},
gamePlay: function(e, t, i, n, a) {
    var o = this;
    this._sendPostRequest(function() {
        return {
            sid: o._sessionId,
            g: o._gameId,
            ga: e,
            ba: t,
            restoreState: i,
            restoreVersion: n
        }
    }, o._gameServerAPIEndpointURL + "/play", a)
},
getBalance: function(e) {
    var t = this;
    this._sendPostRequest(function() {
        return {
            sid: t._sessionId
        }
    }, t._gameServerAPIEndpointURL + "/getbalance", e)
},
getGameStatus: function(e) {
    var t = this;
    this._sendPostRequest(function() {
        return {
            sid: t._sessionId,
            g: t._gameId
        }
    }, t._gameServerAPIEndpointURL + "/refresh", e)
},
gameReplay: function(e, t, i) {

```

```

    this._sendPostRequest(function() {
        return {
            sid: e,
            g: t
        }
    }, this._gameServerAPIEndpointURL + "/replay", i)
},
setFreeSpinAction: function(e, t) {
    var i = this;
    this._sendPostRequest(function() {
        return {
            sid: i._sessionId,
            action: e
        }
    }, i._gameServerAPIEndpointURL + "/freespins", t)
},
getJackpot: function(e) {
    var t = this;
    this._sendPostRequest(function() {
        return {
            sid: t._sessionId,
            g: t._gameId
        }
    }, t._gameServerAPIEndpointURL + "/getjackpot", e)
},
getBetLimits: function(e) {
    var t = this;
    this._sendPostRequest(function() {
        return {
            sid: t._sessionId,
            g: t._gameId
        }
    }, t._gameServerAPIEndpointURL + "/limits", e)
},
getAchievements: function(e) {
    var t = this;
    this._sendPostRequest(function() {
        return {
            sid: t._sessionId,
            g: t._gameId
        }
    }, t._gameServerAPIEndpointURL + "/achievements", e)
},
ping: function(e) {
    var t = this;
    this._sendPostRequest(function() {
        return {
            sid: t._sessionId
        }
    }, t._gameServerAPIEndpointURL + "/ping", e)
},
gameResetRealityCheck: function(e, t, i) {

```

```

var n = this;
this._sendPostRequest(function() {
    return {
        sid: n._sessionId,
        loss: e,
        sessionLength: t
    }
}, n._gameServerAPIEndpointURL + "/resetrealitycheck", i)
},
setRealityCheck: function(e, t, i, n) {
    var a = this;
    this._sendPostRequest(function() {
        return {
            sid: a._sessionId,
            lossLimit: e,
            sessionLimitSeconds: t,
            sessionElapsedSeconds: i
        }
    }, a._gameServerAPIEndpointURL + "/setrealitychecklimits", n)
},
selectGame: function(e) {
    var t = this;
    this._sendPostRequest(function() {
        return {
            sid: t._sessionId,
            g: t._gameId
        }
    }, t._gameServerAPIEndpointURL + "/selectgame", e)
},
saveGameData: function(e, t) {
    var i = this,
        n = i._gameServerAPIEndpointURL + "/setsavedsettings?g=" + this._gameId +
"&sid=" + this._sessionId + "&savedSettings=" + btoa(JSON.stringify(e));
    this._sendServerRequest(n, "", function(e, n) {
        200 != n ? i.onCommunicationErrorCallbackHandler(n) : "" != e.response ? t() :
i.onCommunicationErrorCallbackHandler(400)
    })
},
getGameData: function(e) {
    var t = this;
    this._sendPostRequest(function() {
        return {
            sid: t._sessionId,
            g: t._gameId
        }
    }, t._gameServerAPIEndpointURL + "/getsavedsettings", function(t) {
        var i;
        i = "" != t ? JSON.parse(atob(t.savedSettings)) : {}, e(i)
    })
},
setRestoreState: function(e, t, i) {
    var n = this;

```

```

    this._sendPostRequest(function() {
        return {
            sid: n._sessionId,
            g: n._gameId,
            restoreState: e,
            restoreVersion: t
        }
    }, n._gameServerAPIEndpointURL + "/setrestorestate", i)
},
gameFinished: function(e, t) {
    var i = this;
    this._sendPostRequest(function() {
        return {
            sid: i._sessionId,
            g: i._gameId,
            restoreVersion: e
        }
    }, i._gameServerAPIEndpointURL + "/gamefinished", t)
},
sendRepPlayRequest: function(e, t) {
    var i = e + "?lang=" + t + "&sid=" + this._sessionId + "&gameid=" + this._gameId;
    cc.sys.openURL(i)
},
getSessionResult: function(e) {
    var t = this;
    this._sendPostRequest(function() {
        return {
            sid: t._sessionId
        }
    }, t._gameServerAPIEndpointURL + "/getsessionresult", e)
}
});
cc._RF.pop()
}, {}],
SHS_SettingsController: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "cf654eRd9NKfpx1/KWAvS3h", "SHS_SettingsController");
    var n = e("SHS_GameController"),
        a = e("SHS_GameGlobal");
    cc.Class({
        extends: cc.Component,
        properties: {
            _isShowWindow: !1,
            _currentWindowId: 0,
            _previousWindowId: -1,
            pgView: cc.PageView,
            isShowingPagesFromConsole: !1,
            audioMusicNode: {
                default: null,
                type: cc.Node
            },
            audioSfxNode: {

```

```
    default: null,
    type: cc.Node
  },
  stopAutoPlayInFreeSpinNode: {
    default: null,
    type: cc.Node
  },
  pages: {
    default: [],
    type: [cc.Node]
  },
  btnHistory: {
    default: null,
    type: cc.Node
  },
  settingsButtonSection: {
    default: null,
    type: cc.Node
  },
  betController: {
    default: null,
    type: cc.Node
  },
  applicationVersionLandscape: {
    default: null,
    type: cc.Label
  },
  applicationVersionPotrait: {
    default: null,
    type: cc.Label
  },
  payTableWindow: {
    default: null,
    type: cc.Node
  },
  autoPlayButton: {
    default: null,
    type: cc.Button
  },
  betSettingButton: {
    default: null,
    type: cc.Button
  },
  coinBtnArr: {
    default: [],
    type: cc.Button
  },
  landGameScrollView: {
    default: null,
    type: cc.ScrollView
  },
  portraitGameScrollView: {
```

```

        default: null,
        type: cc.ScrollView
    }
},
onLoad: function() {
    this._previousWindowId = -1, this.audioSfxBtn =
this.audioSfxNode.getComponent("SHS_RadioBtn"), this.stopAutoPlayInFreeSpinBtn =
this.stopAutoPlayInFreeSpinNode.getComponent("SHS_AutoPlayRadioBtn"),
this.audioMusicRdBtn = this.audioMusicNode.getComponent("SHS_RadioBtn"),
this.applicationVersionLandscape.string = a.instance.APPLICATION_VERSION,
this.applicationVersionPotrait.string = a.instance.APPLICATION_VERSION,
this.payTableWindow = this.payTableWindow.getComponent("SHS_PaytableController"),
this.showPage(0)
},
loadPaytableContent: function() {
    cc.log("===== loadPaytableContent ===== :::: " +
this.payTableWindow.isViewCreated), this.payTableWindow.isViewCreated ||
this.payTableWindow.setPaytable(n.instance.playtableInfo)
},
disablePages: function() {
    this.settingsButtonSection.active = !1;
    for (var e = 0; e < this.pages.length; e++)
        this.pages[e].active = !1;
    this.pgView.enabled = !1
},
enablePages: function() {
    this.settingsButtonSection.active = !0;
    for (var e = 0; e < this.pages.length; e++)
        this.pages[e].active = !0;
    this.pgView.enabled = !0
},
showPage: function(e) {
    for (var t = 0; t < this.pages.length; t++)
        this.pages[t].active = !1;
    this.pages[e].active = !0
},
changeScreenView: function(e, t) {
    switch (t) {
        case "0":
            this._currentWindowId = 0;
            break;
        case "1":
            this._currentWindowId = 1;
            break;
        case "2":
            this._currentWindowId = 2;
            break;
        case "3":
            this._currentWindowId = 3
    }
    this._previousWindowId = this._currentWindowId, this.showPage(parseInt(t)),
n.instance._audioController.playButtonSound(), cc.sys.isMobile &&

```

```

this.changeSettingUIElementsPositions()
    },
    audioMusicStateChanged: function() {
        n.instance._audioController.playButtonSound(), this.audioMusicRdBtn.stateChanged(),
a.instance.audioMusic = this.audioMusicRdBtn.currentState(), a.instance.audioMusic ?
(n.instance._audioController.resumeMusic(), n.instance._isFreeSpin ?
n.instance._audioController.playMusic(a.instance.MUSIC_FREE_SPIN) :
n.instance._audioController.playMusic(a.instance.MUSIC_IDLE)) :
n.instance._audioController.pauseMusic(), n.instance.updateMusicStatus(a.instance.audioMusic)
    },
    audioMusicStateTrigger: function() {
        this.audioMusicStateChanged()
    },
    audioSfxStateChanged: function() {
        this.audioSfxBtn.stateChanged(), a.instance.audioSfx = this.audioSfxBtn.currentState(),
n.instance.updateSoundState(a.instance.audioSfx), n.instance._audioController.playButtonSound()
    },
    audioSfxStateChangedTrigger: function() {
        this.audioSfxStateChanged()
    },
    stopAutoPlayInFreeStateChanged: function() {
        this.stopAutoPlayInFreeSpinBtn.stateChanged(), a.instance._isAutoOffInFreeSpin =
this.stopAutoPlayInFreeSpinBtn.currentState()
    },
    updateSoundUI: function() {
        a.instance.audioSfx ? this.audioSfxBtn.On() : this.audioSfxBtn.Off(),
a.instance.audioMusic ? this.audioMusicRdBtn.On() : this.audioMusicRdBtn.Off()
    },
    showSettingsFromGameCtrl: function() {
        n.instance._audioController.playButtonSound(), 0 === this._previousWindowId &&
n.instance.updatePaytableStatus(!1), this._currentWindowId = 3, this._previousWindowId = 3,
this.showPage(3), this.settingsButtonSection.active = !1
    },
    showPayTableFromGameCtrl: function() {
        3 === this._previousWindowId && n.instance.updateOptionStatus(!1),
this._currentWindowId = 0, this._previousWindowId = 0, this.showPage(0),
this.settingsButtonSection.active = !1
    },
    EnabledSettingElements: function(e) {
        n.instance._isAutoPlayPaused && (e = !1), this.autoPlayButton.interactable = e,
this.betSettingButton.interactable = e;
        for (var t = 0; t < this.coinBtnArr.length; t++)
            this.coinBtnArr[t].interactable = e
    },
    refreshBetAmountUI: function() {
        this.betController.getComponent("SHS_BetController").setCurrentBetValue()
    },
    _onTouchBegan: function(e) {},
    _onTouchEnded: function() {},
    _onTouchMoved: function(e, t) {},
    _onTouchCancelled: function(e, t) {},
    _onMouseWheel: function() {},

```

```

changeSettingUIElementsPositions: function() {
    if (cc.sys.isMobile) {
        var e =
this.pages[2].getChildByName("Autoplay_Page_Mobile").getChildByName("Autoplay_Page");

this.pages[2].getChildByName("Autoplay_Page_Mobile").getComponent("SHS_AutoPlayController").setAutoPlayValues(n.instance._autoPlayList);
        var t = 250,
            i = 161,
            o = 0;
        if
(this.betController.getComponent("SHS_BetController").setBetValues(a.instance.betPreLineList),
a.instance.mPortrait) {
            this.landGameScrollView.node.active = !1, this.portraitGameScrollView.node.active
= !0, t = 478, this.pages[2].getChildByName("Autoplay_Page_Mobile").scaleX = 1.6,
this.pages[2].getChildByName("Autoplay_Page_Mobile").scaleY = 1.6,
this.pages[2].getChildByName("Autoplay_Page_Mobile").setPosition(0, 130),
n.instance._hudController.settingMobileUI.setPosition(0, 0),
n.instance._hudController.settingMobileUI.getChildByName("Btns").getChildByName("mobile_bo
ttom_bar").setPosition(0, -484),
n.instance._hudController.settingMobileUI.getChildByName("Btns").setRotation(-90),
this.pages[3].getChildByName("sound").getChildByName("lblsound").setPosition(-131, 171),
this.pages[3].getChildByName("sound").getChildByName("lblbackgroundsound").setPosition(-
131, 64), this.pages[3].getChildByName("sound").getChildByName("BtnSound").setPosition(159,
170),
this.pages[3].getChildByName("sound").getChildByName("BtnBackgroundSound").setPosition(15
9, 64);
            n.instance._isSafari && !0,
n.instance._hudController.settingMobileUI.getChildByName("Btns").getChildByName("Mobile_B
alance").setPosition(-180, -510),
n.instance._hudController.settingMobileUI.getChildByName("Btns").getChildByName("Mobile_T
otalBet").setPosition(180, -510), n.instance.timeLabelSettingUI.node.setPosition(280, 194),
n.instance.timeLabelSettingUI.node.setRotation(-90), n.instance._isClickSplash ?
n.instance._hudController.settingMobileUI.getChildByName("volsunga_logo").setPosition(280,
0) :
n.instance._hudController.settingMobileUI.getChildByName("volsunga_logo").setPosition(280,
-144), n.instance._hudController.settingMobileUI.getChildByName("volsunga_logo").setRotation(-
90), n.instance._hudController.settingMobileUI.getChildByName("btnPlay").setRotation(-90),
n.instance._hudController.settingMobileUI.getChildByName("btnPlay").setPosition(235, 0),
this.pages[2].active ?
(n.instance._hudController.settingMobileUI.getChildByName("Btns").getChildByName("mobile_b
ottom_bar").active || (this.pages[2].getChildByName("Autoplay_Page_Mobile").scaleX = 1.8,
this.pages[2].getChildByName("Autoplay_Page_Mobile").scaleY = 1.8,
this.pages[2].getChildByName("Autoplay_Page_Mobile").setPosition(0, 90)),
n.instance._hudController.settingMobileUI.getChildByName("BtnPortraitBar").active = !1) :
n.instance._hudController.settingMobileUI.getChildByName("BtnPortraitBar").active = !0,
n.instance._hudController.settingMobileUI.getChildByName("BtnPortraitBar").setRotation(0),
n.instance._hudController.settingMobileUI.getChildByName("BtnPortraitBar").setPosition(350, 0)
        } else {
            if (this.landGameScrollView.node.active = !0,
this.portraitGameScrollView.node.active = !1, i = 201, o = 400,
this.pages[2].getChildByName("Autoplay_Page_Mobile").scaleX = 1,

```

```

this.pages[2].getChildByName("Autoplay_Page_Mobile").scaleY = 1,
this.pages[2].getChildByName("Autoplay_Page_Mobile").setPosition(0, 320), n.instance._isIpad) {
    var s = n.instance.iPadVersion();
    n.instance._hudController.settingMobileUI.setPosition(0, -170);
    var r = -220;
    11 == s && (r = -180),
n.instance._hudController.settingMobileUI.getChildByName("Btns").getChildByName("mobile_bottom_bar").setPosition(0, r),
n.instance._hudController.settingMobileUI.getChildByName("Btns").getChildByName("Mobile_Balance").setPosition(-380, r),
n.instance._hudController.settingMobileUI.getChildByName("Btns").getChildByName("Mobile_TotalBet").setPosition(380, r)
    } else

n.instance._hudController.settingMobileUI.getChildByName("Btns").getChildByName("mobile_bottom_bar").setPosition(0, -90), n.instance._hudController.settingMobileUI.setPosition(0, -280),
n.instance._hudController.settingMobileUI.getChildByName("Btns").getChildByName("Mobile_Balance").setPosition(-380, -30),
n.instance._hudController.settingMobileUI.getChildByName("Btns").getChildByName("Mobile_TotalBet").setPosition(380, -30);
    n.instance._hudController.settingMobileUI.getChildByName("Btns").setRotation(0),
this.pages[3].getChildByName("sound").getChildByName("lblsound").setPosition(-131, 171),
this.pages[3].getChildByName("sound").getChildByName("lblbackgroundsound").setPosition(-131, 124), this.pages[3].getChildByName("sound").getChildByName("BtnSound").setPosition(159, 170),
this.pages[3].getChildByName("sound").getChildByName("BtnBackgroundSound").setPosition(159, 124),
n.instance._hudController.settingMobileUI.getChildByName("volsunga_logo").setPosition(-290, 300), n.instance.timeLabelSettingUI.node.setPosition(300, 350),
n.instance.timeLabelSettingUI.node.setRotation(0),
n.instance._hudController.settingMobileUI.getChildByName("volsunga_logo").setRotation(0),
n.instance._hudController.settingMobileUI.getChildByName("btnPlay").setRotation(0),
n.instance._hudController.settingMobileUI.getChildByName("btnPlay").setPosition(280, 150),
n.instance._hudController.settingMobileUI.getChildByName("BtnPortraitBar").active = !1
    }
    e.getChildByName("select_number").setPositionX(t),
e.getChildByName("stop_auto").setPositionX(t), e.getChildByName("if_loss").setPositionX(t),
e.getChildByName("if_single").setPositionX(t),
e.getChildByName("stop_if_freespin_won").setPositionX(t),
e.getChildByName("Clear_Btn").setPosition(544 + o, i),
e.getChildByName("Play_Btn").setPosition(733 + o, i)
    }
    },
    hideDuplicateSettingsButtons: function(e) {}
    });
cc._RF.pop()
}, {
    SHS_GameController: "SHS_GameController",
    SHS_GameGlobal: "SHS_GameGlobal"
}],
SHS_SettingsLocal: [function(e, t, i) {
    "use strict";

```

```
cc._RF.push(t, "6fd1e4clUIIR5bNH0ozhZmb", "SHS_SettingsLocal");
var n = e("SHS_GameGlobal");
e("SHS_GameHelper");
cc.Class({
  extends: cc.Component,
  properties: {
    PAYTABLE1: {
      default: null,
      type: cc.Node
    },
    PAYTABLE2: {
      default: null,
      type: cc.Node
    },
    PAYTABLE3: {
      default: null,
      type: cc.Node
    },
    PAYTABLE4: {
      default: null,
      type: cc.Node
    },
    PAYTABLE5: {
      default: null,
      type: cc.Node
    },
    PAYS_BOTH_WAYS: {
      default: null,
      type: cc.Node
    },
    EXTENDS_MATRIX: {
      default: null,
      type: cc.Node
    },
    ACTIVATES_MIXED: {
      default: null,
      type: cc.Node
    },
    TRIGGER_6_FREE_SPIN: {
      default: null,
      type: cc.Node
    },
    AND_PAYS: {
      default: null,
      type: cc.Node
    },
    WHEN_THE_RED: {
      default: null,
      type: cc.Node
    },
    COIN_VAL1: {
      default: null,
```

```
    type: cc.Node
  },
  MALFUNCTIONTEXT1: {
    default: null,
    type: cc.Node
  },
  COIN_VAL2: {
    default: null,
    type: cc.Node
  },
  MALFUNCTIONTEXT2: {
    default: null,
    type: cc.Node
  },
  COIN_VAL3: {
    default: null,
    type: cc.Node
  },
  MALFUNCTIONTEXT3: {
    default: null,
    type: cc.Node
  },
  MALFUNCTIONTEXT4: {
    default: null,
    type: cc.Node
  },
  MALFUNCTIONTEXT5: {
    default: null,
    type: cc.Node
  },
  MALFUNCTIONTEXT6: {
    default: null,
    type: cc.Node
  },
  GAME_RULES: {
    default: null,
    type: cc.Node
  },
  GAME_RULE1: {
    default: null,
    type: cc.Node
  },
  GAME_RULE2: {
    default: null,
    type: cc.Node
  },
  GAME_RULE_LIST_01: {
    default: null,
    type: cc.Node
  },
  GAME_RULE_LIST_02: {
    default: null,
```

```
    type: cc.Node
  },
  GAME_RULE_LIST_03: {
    default: null,
    type: cc.Node
  },
  btnHistory: {
    default: null,
    type: cc.Node
  },
  lblTitleBetSettings: {
    default: null,
    type: cc.Node
  },
  lblAutoPlayTitle: {
    default: null,
    type: cc.Node
  },
  lblSelectNumber: {
    default: null,
    type: cc.Node
  },
  lblStopAuto: {
    default: null,
    type: cc.Node
  },
  lblLossExceeds: {
    default: null,
    type: cc.Node
  },
  lblSingleWinExceeds: {
    default: null,
    type: cc.Node
  },
  lblSoundTitle: {
    default: null,
    type: cc.Node
  },
  lblSound: {
    default: null,
    type: cc.Node
  },
  lblBGSound: {
    default: null,
    type: cc.Node
  },
  lblClear: {
    default: null,
    type: cc.Node
  },
  lblPlay: {
    default: null,
```

```

        type: cc.Node
    },
    lblStopIf: {
        default: null,
        type: cc.Node
    },
    BUTTONPLAY: {
        default: null,
        type: cc.Node
    },
    _isLocalized: !1
},
update: function(e) {
    !this._isLocalized && n.instance.auto_strings && n.instance.paytable_strings &&
n.instance.gamerules_strings && n.instance.sound_strings && (this.localizedComponent(),
this._isLocalized = !0)
},
localizedComponent: function() {
    cc.sys.isMobile || (this.PAYTABLE1.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".paytable"),
this.PAYTABLE2.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".paytable"),
this.PAYTABLE3.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".paytable"),
this.PAYTABLE4.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".paytable"),
this.PAYTABLE5.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".paytable"),
this.PAYS_BOTH_WAYS.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".pays_both_ways"),
this.EXTENDS_MATRIX.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".extends_matrix"),
this.ACTIVATES_MIXED.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".activates_mixed"),
this.TRIGGER_6_FREE_SPIN.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".trigger_6"),
this.AND_PAYS.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".5x5"),
this.WHEN_THE_RED.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".when_the_red_moon"),
this.COIN_VAL1.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".coin_value"),
this.COIN_VAL2.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".coin_value"),
this.COIN_VAL3.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".coin_value"),
this.GAME_RULES.getComponent(cc.Sprite).spriteFrame =
n.instance.gamerules_strings.getSpriteFrame(n.instance.localSpecifier + ".game_rules"),
cc.sys.isMobile ? (this.MALFUNCTIONTEXT1.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction_portrait"),
this.MALFUNCTIONTEXT2.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction_portrait"),

```

```
this.GAME_RULE_LIST_01.getComponent(cc.Sprite).spriteFrame =
n.instance.gamerules_strings.getSpriteFrame(n.instance.localSpecifier + ".gamerules_portrait")) :
(this.GAME_RULE1.getComponent(cc.Sprite).spriteFrame =
n.instance.gamerules_strings.getSpriteFrame(n.instance.localSpecifier + ".game_rules"),
this.GAME_RULE2.getComponent(cc.Sprite).spriteFrame =
n.instance.gamerules_strings.getSpriteFrame(n.instance.localSpecifier + ".game_rules"),
this.MALFUNCTIONTEXT1.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction"),
this.MALFUNCTIONTEXT2.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction"),
this.MALFUNCTIONTEXT3.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction"),
this.MALFUNCTIONTEXT4.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction"),
this.MALFUNCTIONTEXT5.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction"),
this.MALFUNCTIONTEXT6.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".malfunction"),
this.GAME_RULE_LIST_01.getComponent(cc.Sprite).spriteFrame =
n.instance.gamerules_strings.getSpriteFrame(n.instance.localSpecifier + ".gamerules_list_01"),
this.GAME_RULE_LIST_02.getComponent(cc.Sprite).spriteFrame =
n.instance.gamerules_strings.getSpriteFrame(n.instance.localSpecifier + ".gamerules_list_02"),
this.GAME_RULE_LIST_03.getComponent(cc.Sprite).spriteFrame =
n.instance.gamerules_strings.getSpriteFrame(n.instance.localSpecifier + ".gamerules_list_03")),
this.btnHistory.getComponent(cc.Sprite).spriteFrame =
n.instance.gamerules_strings.getSpriteFrame(n.instance.localSpecifier + ".history")),
this.BUTTONPLAY.getComponent(cc.Sprite).spriteFrame =
n.instance.auto_strings.getSpriteFrame(n.instance.localSpecifier + ".play"),
this.lblAutoPlayTitle.getComponent(cc.Sprite).spriteFrame =
n.instance.auto_strings.getSpriteFrame(n.instance.localSpecifier + ".autoplay"),
this.lblSelectNumber.getComponent(cc.Sprite).spriteFrame =
n.instance.auto_strings.getSpriteFrame(n.instance.localSpecifier + ".select_number"),
this.lblStopAuto.getComponent(cc.Sprite).spriteFrame =
n.instance.auto_strings.getSpriteFrame(n.instance.localSpecifier + ".stop_auto"),
this.lblLossExceeds.getComponent(cc.Sprite).spriteFrame =
n.instance.auto_strings.getSpriteFrame(n.instance.localSpecifier + ".if_loss"),
this.lblSingleWinExceeds.getComponent(cc.Sprite).spriteFrame =
n.instance.auto_strings.getSpriteFrame(n.instance.localSpecifier + ".if_single"),
this.lblStopIf.getComponent(cc.Sprite).spriteFrame =
n.instance.auto_strings.getSpriteFrame(n.instance.localSpecifier + ".stop_if"),
this.lblClear.getComponent(cc.Sprite).spriteFrame =
n.instance.auto_strings.getSpriteFrame(n.instance.localSpecifier + ".clear"),
this.lblPlay.getComponent(cc.Sprite).spriteFrame =
n.instance.auto_strings.getSpriteFrame(n.instance.localSpecifier + ".play"),
this.lblSoundTitle.getComponent(cc.Sprite).spriteFrame =
n.instance.sound_strings.getSpriteFrame(n.instance.localSpecifier + ".sound"),
this.lblSound.getComponent(cc.Sprite).spriteFrame =
n.instance.sound_strings.getSpriteFrame(n.instance.localSpecifier + ".sound"),
this.lblBGSound.getComponent(cc.Sprite).spriteFrame =
n.instance.sound_strings.getSpriteFrame(n.instance.localSpecifier + ".background_sound"),
this.lblTitleBetSettings.getComponent(cc.Sprite).spriteFrame =
n.instance.paytable_strings.getSpriteFrame(n.instance.localSpecifier + ".bet_settings")
```

```

    }
  }, cc._RF.pop()
}, {
  SHS_GameGlobal: "SHS_GameGlobal",
  SHS_GameHelper: "SHS_GameHelper"
}],
SHS_SlotItem: [function(e, t, i) {
  "use strict";
  cc._RF.push(t, "507e0lv+2VIgJ/f0Q/zGYQG", "SHS_SlotItem");
  var n = e("SHS_GameController"),
    a = e("SHS_GameHelper");
  cc.Class({
    extends: cc.Component,
    properties: {
      _colorTable: null,
      symbolID: 0,
      prefabSymNode: null,
      anim: null,
      blurInstance: null,
      normalInstance: null,
      symbolCollection: null,
      faceSymbol: {
        default: null,
        type: cc.Sprite
      }
    },
    onLoad: function() {
      this._colorTable = [], this.prefabSymNode = null
    },
    changeSymbol: function(e) {
      this.changeSymbolEx(e, -1, !1, !0)
    },
    changeSymbolEx: function(e, t, i) {
      null !== this.prefabSymNode && (null == n.instance._symbolsPool[e] &&
a.instance.gameLog("Change Symbol: " + e + " item ID: " + t + " Reel item length: " +
n.instance.reelStopSlotItems.length), null !== n.instance._symbolsPool[this.symbolID] &&
n.instance._symbolsPool[this.symbolID].put(this.prefabSymNode)), this.node.removeAllChildren(!
0), this.symbolID = e;
      var o = null;
      null !== n.instance._symbolsPool[e] && (o = n.instance._symbolsPool[e].get()), o || (o =
cc.instantiate(n.instance.slotAnimArray[e])), this.prefabSymNode = o, null !== this.prefabSymNode
&& (this.anim = this.prefabSymNode.getComponent(cc.Animation), this.blurInstance =
this.prefabSymNode.getChildByName("Blur"), this.normalInstance =
this.prefabSymNode.getChildByName("Normal"), this.node.addChild(this.prefabSymNode)), i &&
(n.instance.reelStopSlotItems[t - 1] = this.prefabSymNode, n.instance.reelStopSlotItems[t -
1].setOpacity(255)), this.showNormalImage()
    },
    addGhostSymbol: function() {
      var e = cc.instantiate(n.instance.slotAnimArray[13]),
        t = this.getWinAnimationNameById(13);
      e.getComponent(cc.Animation).play(t), this.node.addChild(e)
    },
  }],

```

```

addBackGhostSymbol: function() {
    this.node.removeAllChildren(!0);
    var e = cc.instantiate(n.instance.slotAnimArray[14]),
        t = this.getWinAnimationNameById(14);
    e.getComponent(cc.Animation).play(t), this.node.addChild(e)
},
playWinAnim: function() {
    null != this.prefabSymNode && (this.faceSymbol.setVisible(!1),
this.prefabSymNode.getChildByName("Blur").setOpacity(0))
},
playIdleAnim: function() {
    null != this.prefabSymNode && this.faceSymbol.setVisible(!1)
},
showBlurImage: function() {
    null !== this.blurInstance && (this.blurInstance.active = !0, this.normalInstance.active = !
1)
},
showNormalImage: function() {
    null !== this.blurInstance && (this.blurInstance.active = !1, this.normalInstance.active = !
0)
},
SetSlotOpacity: function(e) {
    if (null != this.prefabSymNode) {
        this.prefabSymNode.getChildByName("Blur").setOpacity(0);
        for (var t = this.prefabSymNode.getChildByName("Normal")._children, i = 0; i <
t.length; i++) {
            var n = t[i];
            n.color = 255 === e ? this._colorTable[n._name] : new cc.Color(80, 80, 80, 255)
        }
    }
});
cc._RF.pop()
}, {
    SHS_GameController: "SHS_GameController",
    SHS_GameHelper: "SHS_GameHelper"
}],
SHS_SlotReel: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "0efd0PMm1hIKI2d2v/xksGm", "SHS_SlotReel");
    var n = e("SHS_GameController"),
        a = e("SHS_GameGlobal");
    cc.Class({
        extends: cc.Component,
        properties: {
            _spinCount: 0,
            _isSuspense: !1,
            _reelFreeSpinCount: 0,
            _bonusSFXId: -1,
            _bonusPos: [],
            reelIndex: -1,
            anim: null,

```

```

    animState: null,
    slotItemsArray: {
      default: [],
      type: [cc.Node]
    },
    _turboSpeed: .1
  },
  initReel: function() {
    this._isNoPrizeLayout = !0, this.anim = this.getComponent(cc.Animation), this.animState
= this.anim.play("slotSpinStopReel05"), this.showNormalImages(), this.resetParam()
  },
  resetParam: function() {
    this._reelFreeSpinCount = 2, this._spinCount = this.reelIndex + this._reelFreeSpinCount,
this._isSuspense = !1, this._bonusSFXId = -1
  },
  setSlotItemsInReel: function(e) {
    this.reelIndex = e;
    for (var t = a.instance.reelToViewMap[e], i = 0; i <
a.instance.reelDisplayPositionsArray.length; i++) {
      var n,
        o;
      o = t[i], n = a.instance.reelLayout[o - 1],
this.slotItemsArray[a.instance.reelDisplayPositionsArray[i]].setOpacity(255),
this.slotItemsArray[a.instance.reelDisplayPositionsArray[i]].getComponent("SHS_SlotItem").chan
geSymbol(n - 1, e)
    }
    this.setNonDisplayPositions()
  },
  setDisplayPositions: function(e) {
    this.reelIndex = e;
    for (var t = a.instance.reelToViewMap[e], i = 0; i <
a.instance.reelDisplayPositionsArray.length; i++) {
      var o,
        s;
      s = t[i], o = a.instance.reelLayout[s - 1], "MP" != n.instance._selectedScheme || 2 != o
&& 1 != o && 3 != o || n.instance._resetShiftPositions[e].push([s,
a.instance.reelDisplayPositionsArray[i], o]),
this.slotItemsArray[a.instance.reelDisplayPositionsArray[i]].setOpacity(255),
this.slotItemsArray[a.instance.reelDisplayPositionsArray[i]].getComponent("SHS_SlotItem").chan
geSymbolEx(o - 1, s, !0)
    }
  },
  reSetDisplayPositions: function(e) {
    for (var t = 0; t < e.length; t++) {
      var i = e[t][1],
        n = e[t][0],
        a = e[t][2] + 12;
      this.slotItemsArray[i].setOpacity(255),
this.slotItemsArray[i].getComponent("SHS_SlotItem").changeSymbolEx(a - 1, n, !0)
    }
  },
  setNonDisplayPositions: function() {

```

```

a.instance.selectedReelSet.toString()[this.reelIndex];
for (var e = a.instance.reelDisplayPositionsArray, t = 0; t < 15; t++) {
    var i;
    this.slotItemsArray[t].setOpacity(255), -1 == e.indexOf(t) && (1 == (i = Math.floor(7
* Math.random() + 1)) && (i = 2),
this.slotItemsArray[t].getComponent("SHS_SlotItem").changeSymbol(i - 1, this.reelIndex))
    }
},
startSpinning: function(e, t) {
    this.reelIndex = t;
    var i = cc.callFunc(this.playStartAnimCompleted, this),
        n = cc.sequence(this.playStartAnim(this), cc.delayTime(.22), i);
    this.node.runAction(n)
},
playStartAnim: function(e) {
    e = this;
    return cc.callFunc(function() {
        e.anim = e.getComponent(cc.Animation), e.animState = e.anim.play("slotSpinStart"),
n.instance._isTurbo ? e.animState.speed = e._turboSpeed : e.animState.speed = .85
    })
},
playStartAnimCompleted: function() {
    var e = cc.callFunc(this.finishSpinningEx, this),
        t = .1;
    a.instance.minSpinTime >= 3e3 && (0 == this.reelIndex ? t = .5 : 1 == this.reelIndex ? t
= .5 : 2 == this.reelIndex ? t = .45 : 3 == this.reelIndex ? t = .4 : 4 == this.reelIndex && (t = .38));
    var i = cc.sequence(this.playSpinAnim(this), cc.delayTime(t), e);
    this.node.runAction(i)
},
playStartAnimExcuted: function() {
    var e = cc.callFunc(this.finishSpinningEx, this),
        t = .1;
    a.instance.minSpinTime >= 3e3 && (0 == this.reelIndex ? t = .5 : 1 == this.reelIndex ? t
= .5 : 2 == this.reelIndex ? t = .45 : 3 == this.reelIndex ? t = .4 : 4 == this.reelIndex && (t = .38));
    var i = cc.sequence(cc.delayTime(t), e);
    this.node.runAction(i)
},
playSpinAnim: function(e) {
    return cc.callFunc(function() {
        e.anim.stop(), e.showBlurImages(), e.anim = e.getComponent(cc.Animation),
e.animState = e.anim.play("slotSpin")
    })
},
SpinStartCompleted: function() {},
finishSpinning: function() {},
finishSpinningEx: function() {
    var e = !!navigator.userAgent.match(/Version\/[\d\.\.]+.*Safari/),
        t = !0;
    if (!/iPad/.test(navigator.userAgent) && !window.MSStream) && !
a.instance.isSafariFullScreen && e && cc.sys.isMobile && cc.sys.os == cc.sys.OS_IOS &&
a.instance.fullscreen && (t = !1), n.instance._isRelaxDataAvailable && t && !
a.instance._isSettingWindowOpen) {

```

```

        this._spinCount--;
        var i = 9;
        if (a.instance.minSpinTime >= 3e3 && (i = 3, 4 == this.reelIndex && (i = 2)),
this._isSuspense && this._spinCount == i)
            return n.instance._audioController.playAnticipationEffect(this.reelIndex - 2),
n.instance.suspensNodeArr[this.reelIndex -
2].getComponent(cc.Animation).play("Suspense_Ani_01"), void this.playStartAnimExcuted();
            if (0 == this.reelIndex && n.instance._isCommError)
                return void (n.instance._isRelaxDataAvailable = !1);
            if (1 == this.reelIndex && n.instance._isTurbo && n.instance.resetSlotMask(),
this._spinCount <= 0 || a.instance.minSpinTime < 3e3 && (n.instance._stopAnimation ||
n.instance._isClicked || n.instance._isTurbo)) {
                this.setDisplayPositions(this.reelIndex), this.setNonDisplayPositions();
                this.reelIndex;
                var o = cc.callFunc(this.removeReelEndEffectEx, this),
                    s = cc.callFunc(this.playReelEndSound, this),
                    r = .1,
                    l = .52;
                if (a.instance.minSpinTime >= 3e3)
                    if (0 == this.reelIndex)
                        r = .04, l = .57;
                    else if (1 == this.reelIndex)
                        r = .05, l = .57;
                    else if (2 == this.reelIndex)
                        r = .1, l = .52;
                    else if (3 == this.reelIndex)
                        r = .15, l = .47;
                    else if (4 == this.reelIndex)
                        r = .2, l = .42;
                var c = cc.sequence(this.playSpinStopAnim(this), cc.delayTime(r), s,
cc.delayTime(l), o);
                return void this.node.runAction(c)
            }
        }
    },
    this.playStartAnimExcuted()
},
playSpinStopAnim: function(e) {
    return cc.callFunc(function() {
        e.anim = e.getComponent(cc.Animation);
        a.instance.selectedReelSet.toString()[e.reelIndex];
        if (e.animState = e.anim.play("slotSpinStopReel05"), e._bonusSFXId > 0) {
            var t = .7;
            2 == e.reelIndex && (t = .6), 0 != e.reelIndex && 1 != e.reelIndex && 2 !=
e.reelIndex || n.instance._isFirstBonusLandPlayed ||
(n.instance._audioController.playAudioBonusLand(t), n.instance._isFirstBonusLandPlayed = !0),
1 != e.reelIndex && 2 != e.reelIndex && 3 != e.reelIndex || n.instance._isSecondBonusLandPlayed
|| !n.instance._isFirstBonusLandPlayed || (n.instance._audioController.playAudioBonusLand(t),
n.instance._isSecondBonusLandPlayed = !0)
        }
    })
},
playReelEndSound: function() {

```

```

        n.instance._audioController.playReelEndSound(this.reelIndex)
    },
    removeReelEndEffect: function() {},
    removeReelEndEffectEx: function() {
        null !== this.reelStopEffect && n.instance.finishedReelSpinning(this.reelIndex),
this.resetParam()
    },
    showNormalImages: function() {
        for (var e = this.slotItemsArray.length, t = 0; t < e; t++)
            this.slotItemsArray[t].getComponent("SHS_SlotItem").showNormalImage()
    },
    showBlurImages: function() {
        for (var e = this.slotItemsArray.length, t = 0; t < e; t++)
            this.slotItemsArray[t].getComponent("SHS_SlotItem").showBlurImage()
    }
});
cc._RF.pop()
}, {
    SHS_GameController: "SHS_GameController",
    SHS_GameGlobal: "SHS_GameGlobal"
}],
SHS_SlotWinningControl: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "5b092qy/WJPO5pPFEUQISf1", "SHS_SlotWinningControl");
    cc.Class({
        extends: cc.Component,
        properties: {
            _normalInstance: null,
            slotLogos: {
                default: [],
                type: cc.Node
            },
            winFrame: {
                default: null,
                type: cc.Node
            }
        }
    },
    loadLogos: function() {
        this._normalInstance = [];
        for (var e = 0; e < this.slotLogos.length; e++)
            this._normalInstance &&
this._normalInstance.push(this.slotLogos[e].getChildByName("Normal"))
    },
    hideLogos: function() {
        for (var e = 0; e < this.slotLogos.length; e++)
            this._normalInstance && (this._normalInstance[e].active = !1);
        this.winFrame.active = !1
    },
    showLogo: function(e) {
        this._normalInstance && (this._normalInstance[e - 1].active = !0), this.winFrame.active =
!1
    },
}

```

```

        showWinLogo: function(e, t) {
            e > 0 && null != this._normalInstance[e - 1] && (this._normalInstance &&
(this._normalInstance[e - 1].active = !0), this.winFrame.active = t, this.slotLogos[e -
1].getComponent(cc.Animation).play("Reset"))
        },
        resetWinLogo: function(e, t) {
            e > 0 && null != this._normalInstance[e - 1] && t && (this._normalInstance &&
(this._normalInstance[e - 1].active = !0), this.slotLogos[e -
1].getComponent(cc.Animation).play("Reset"))
        },
        showShiftWinLogo: function(e, t) {
            e > 0 && null != this._normalInstance[e - 1] && (this._normalInstance &&
(this._normalInstance[e - 1].active = !0), this.winFrame.active = t, this.slotLogos[e -
1].getComponent(cc.Animation).play("winShiftAnim").wrapMode = cc.WrapMode.Normal)
        },
        showShiftWinLogoNormalSpin: function(e, t) {
            e > 0 && null != this._normalInstance[e - 1] && (this._normalInstance &&
(this._normalInstance[e - 1].active = !0), this.winFrame.active = t, this.slotLogos[e -
1].getComponent(cc.Animation).play("winShiftAnim").wrapMode = cc.WrapMode.PingPong)
        },
        showBonusWinLogo: function(e) {
            e > 0 && (this._normalInstance && (this._normalInstance[e - 1].active = !0),
this.winFrame.active = !1, this.slotLogos[e -
1].getComponent(cc.Animation).play("winAnimBonus"))
        }
    });
    cc._RF.pop()
}, {}],
SHS_WinLableMessage: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "e9ee9up/21Pr56o7E/06q39", "SHS_WinLableMessage");
    e("SHS_GameGlobal"), e("SHS_GameHelper"), cc.Class({
        extends: cc.Component,
        properties: {
            lblMsgArr: {
                default: [],
                type: [cc.Node]
            },
            lblMoonMsgArr: {
                default: [],
                type: [cc.Node]
            }
        }
    },
    onLoad: function() {
        this.initNode()
    },
    initNode: function() {
        for (var e = 0; e < this.lblMsgArr.length; e++)
            this.lblMsgArr[e].active = !1;
        for (e = 0; e < this.lblMoonMsgArr.length; e++)
            this.lblMoonMsgArr[e].active = !1
    },
}

```

```

showWinLineMessage: function(e, t) {
    for (var i = 0; i < this.lblMsgArr.length; i++)
        t == this.lblMsgArr[i].name ?
(this.lblMsgArr[i].getChildByName(t).getComponent(cc.Label).string = e, this.lblMsgArr[i].active
= !0) : this.lblMsgArr[i].active = !1
    },
clearWinLineMessage: function() {
    for (var e = 0; e < this.lblMsgArr.length; e++)
        this.lblMsgArr[e].active = !1
    },
showWinLineMoonMessage: function(e, t) {
    for (var i = 0; i < this.lblMoonMsgArr.length; i++)
        t == this.lblMoonMsgArr[i].name ?
(this.lblMoonMsgArr[i].getChildByName(t).getComponent(cc.Label).string = e,
this.lblMoonMsgArr[i].active = !0) : this.lblMoonMsgArr[i].active = !1
    },
clearWinLineMoonMessage: function() {
    for (var e = 0; e < this.lblMoonMsgArr.length; e++)
        this.lblMoonMsgArr[e].active = !1
    }
});
cc._RF.pop()
}, {
    SHS_GameGlobal: "SHS_GameGlobal",
    SHS_GameHelper: "SHS_GameHelper"
}],
bg: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "ddd09AxaRZOILLUAhXL7f0G", "bg"), t.exports = {
        WELCOME: "Welcome, %{name}!",
        Click_anywhere_on_the_screen_to_start_the_game: "Натиснете където и да е по екрана,
за да започнете играта",
        HUD: {
            BET: "ЗАЛОГ",
            CREDITS: "КРЕДИТИ",
            COIN_VALUE: "МОНЕТНА СТОЙНОСТ",
            WIN: "ПЕЧАЛБА",
            AUTOPLAY: "АВТОМАТИЧНО ИЗПЪЛНЕНИЕ",
            МАХВЕТ: "МАКСИМАЛЕН ЗАЛОГ",
            "576WAYS_TO_WIN": "576 НАЧИНА ДА СПЕЧЕЛИТЕ!",
            WIN_UP_TO_128_FREE_SPINS: "СПЕЧЕЛЕТЕ ДО 128 БЕЗПЛАТНИ
ЗАВЪРТАНИЯ",
            FREE: "Безплатно"
        },
        HUD_MESSAGES: {
            EXTEND_MATRIX: "Extends matrix",
            PAY_BOTH_WAYS_MATRIX: "Pays both ways and extends matrix",
            ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Activates mixed pays and
extends matrix",
            IS_ACTIVE_ALL: " is active, all characters shape-shift into their animal form and share
wins (mixed pays)",
            "3X": "3x",

```

```

    DURING_THE_FREE_SPIN: "During free spins, the game is always 5x5 and always
pays both ways",
    TRIGGER_6_FREE_SPINS: "trigger 6 free spins is active.",
    FREE_SPIN_CAN_RETRIGGER: "Free spins can retrigger",
    GOOD_LUCK: "Good Luck",
    ADDITIONAL_FREE_SPINS_AWARDED: "ADDITIONAL FREE SPINS
AWARDED"
},
PAYTABLE: {
    PAYTABLE: "PAYTABLE",
    PAYS_BOTH_WAYS: "Pays both ways and extends matrix",
    EXTENDS_MATRIX: "Extends Matrix",
    ACTIVATES_MIXED: "Activates mixed pays and extends matrix",
    TRIGGER_6_FREE_SPIN: "trigger 6 free spins",
    AND_PAYS: "5x5 and pays both ways",
    WHEN_THE_RED: "When the Red Moon is active, all characters shape-shift into their
animal form and share wins (mixed pays)",
    MALFUNCTION_TEXT: "Malfunction voids all pays and plays",
    GAME_RULES: "GAME_RULES",
    GAME_RULES1: "Volsunga is a ways game with an extending 53135, 53335, 55355 and
55555 matrix",
    GAME_RULES2: "Players have 3125 ways to win",
    GAME_RULES3: "The game has a 97.16% RTP",
    GAME_RULES4: "All pays except scatter pays are left-to-right starting from the leftmost
reel",
    GAME_RULES5: "Scatter pays are paid for symbols appearing in any position on the
reels",
    GAME_RULES6: "Only the highest symbol win is paid per winning combination",
    GAME_RULES7: "Only the highest scatter win is paid per play",
    GAME_RULES8: "Scatter pays are paid in addition to ways pays",
    GAME_RULES9: "Ways pays are multiplied by the bet multiple",
    GAME_RULES10: "Scatter pays are multiplied by the total credits bet",
    GAME_RULES11: "During free spins the game is always 5x5 and always pays both
ways",
    GAME_RULES12: "Free spins can retrigger"
},
BETSETTINGS: "НАСТРОЙКИ НА ЗАЛОГА",
AUTOPLAY: {
    AUTOPLAY_TITLE: "АВТОМАТИЧНО ИЗПЪЛНЕНИЕ",
    AUTOPLAY_SUBTITLE_1: "Изберете брой завъртания",
    AUTOPLAY_SUBTITLE_2: "Спри автоматичното изпълнение",
    AUTOPLAY_DES_1: "Ако преобладава загуба",
    AUTOPLAY_DES_2: "Ако преобладава единична печална",
    CLEAR: "Изчисти",
    PLAY: "Играй"
},
SOUNDSETTINGS: {
    SOUND: "ЗВУК",
    BACKGROUND_SOUND: "ФОНОВ ЗВУК"
},
MSG: {
    BIG_WIN: "ДОБРА ПЕЧАЛБА",

```

```

HUGE_WIN: "ЕКСТРЕМНА ПЕЧАЛБА",
EPIC_WIN: "СОЛИДНА ПЕЧАЛБА",
MEGA_WIN: "ОГРОМНА ПЕЧАЛБА",
CONGRATULATIONS: "ПОЗДРАВЕНИЯ",
CREDITS_WON: "СПЕЧЕЛЕНИ КРЕДИТИ",
COMBINATION: "КОМБИНАЦИЯ/КОМБИНАЦИИ",
AWARDED: "НАГРАДЕНИ"
}
}, cc._RF.pop()
}, {}],
cs: [function(e, t, i) {
"use strict";
cc._RF.push(t, "c5e7ePy3X1H04uRLCRJoeco", "cs"), t.exports = {
WELCOME: "Welcome, %{name}!",
Click_anywhere_on_the_screen_to_start_the_game: "Klepnutím kdekoliv na obrazovce
spustíte hru",
HUD: {
BET: "SÁZKA",
CREDITS: "KREDITY",
COIN_VALUE: "HODNOTA MINCE",
WIN: "VÝHRA",
AUTOPLAY: "AUTOMATICKÁ HRA",
MAXBET: "MAXIMÁLNÍ SÁZKA",
"576WAYS_TO_WIN": "576 ZPŮSOBŮ, JAK VYHRÁT",
WIN_UP_TO_128_FREE_SPINS: "VYHRAJTE AŽ 128 ROZTOČENÍ ZDARMA",
FREE: "ZDARMA"
},
HUD_MESSAGES: {
EXTEND_MATRIX: "Rozšiřuje mřížku",
PAY_BOTH_WAYS_MATRIX: "Vyplácí oběma směry a rozšiřuje mřížku",
ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Aktivuje smíšené výplaty a
rozšiřuje mřížku",
IS_ACTIVE_ALL: "Pokud je Red Moon aktivní, všechny postavy změni tvar do zvířecí
podoby a sdílejí výhry (smíšené výplaty)",
"3X": "3x",
DURING_THE_FREE_SPIN: "Během roztočení zdarma je uspořádání hry vždy 5x5 a
výhry se vždy vyplácejí oběma směry",
TRIGGER_6_FREE_SPINS: "spouští 6 roztočení zdarma",
FREE_SPIN_CAN_RETRIGGER: "Roztočení zdarma je možné spustit znovu",
GOOD_LUCK: "Good Luck",
ADDITIONAL_FREE_SPINS_AWARDED: "ADDITIONAL FREE SPINS
AWARDED",
X_FREE_PLAY: "zbývá %{remainingCount} her zdarma",
X_FREE_PLAY_COMPLETED: "zbývá her zdarma",
FREE_SPIN_AWARDED: "získali jste %{spinCount} her zdarma",
FREE_SPIN_OVER: "hry zdarma skončily",
NO_THANKS: "ne, díky",
PLAY_LATER: "hrát později",
PLAY_NOW: "hrát teď"
},
PAYTABLE: {
PAYTABLE: "TABULKA VÝPLAT",

```

PAYS\_BOTH\_WAYS: "Vyplácí oběma směry a rozšiřuje mřížku",  
EXTENDS\_MATRIX: "Rozšiřuje mřížku",  
ACTIVATES\_MIXED: "Aktivuje smíšené výplaty a rozšiřuje mřížku",  
TRIGGER\_6\_FREE\_SPIN: "spouští 6 roztočení zdarma",  
AND\_PAYS: "Uspořádání 5x5 a vyplácení výher oběma směry",  
WHEN\_THE\_RED: "Pokud je Red Moon aktivní, všechny postavy změny tvar do zvířecí podoby a sdílejí výhry (smíšené výplaty)",  
MALFUNCTION\_TEXT: "PORUCHA RUŠÍ VŠECHNY HRY A VÝHRY",  
GAME\_RULES: "PRAVIDLA HRY",  
GAME\_RULES1: "VOLSUNGA JE HRA S ROZŠIŘUJÍCÍ SE MŘÍŽKOU 53135, 53335, 55355 A 55555",  
GAME\_RULES2: "HRÁČ MÁ K DISPOZICI 3125 VÝHERNÍCH TRAS",  
GAME\_RULES3: "NÁVRATNOST HRÁČI (RTP) TÉTO HRY ČINÍ 97,16 %",  
GAME\_RULES4: "VŠECHNY VÝHRY KROMĚ VÝHER SE SYMBOLY SCATTER SE VYPLÁCEJÍ ZLEVA DOPRAVA POČÍNÁJE VÁLCEM ÚPLNĚ VLEVO",  
GAME\_RULES5: "VÝHRY SE SYMBOLY SCATTER SE VYPLÁCEJÍ ZA SYMBOLY SCATTER VE KTERÉKOLI POZICI NA VÁLCÍCH",  
GAME\_RULES6: "VYPLÁCÍ SE VŽDY JEN NEJVYŠŠÍ VÝHRA VE VÝHERNÍ KOMBINACI SYMBOLŮ",  
GAME\_RULES7: "VYPLÁCÍ SE VŽDY JEN NEJVYŠŠÍ VÝHRA ZA SYMBOLY SCATTER NA HRU",  
GAME\_RULES8: "VÝHRY SE SYMBOLY SCATTER SE PŘÍČITAJÍ K BĚŽNÝM VÝHRÁM NA LINIÍCH",  
GAME\_RULES9: "VÝHRY NA LINIÍCH JSOU NÁSOBENY NÁSOBKEM SÁZKY",  
GAME\_RULES10: "VÝHRY SE SYMBOLY SCATTER JSOU NÁSOBENY CELKOVOU SÁZKOU",  
GAME\_RULES11: "BĚHEM ROZTOČENÍ ZDARMA JE HRA VŽDY V REŽIMU 5X5 A VÝHRY SE VYPLÁCEJÍ OBĚMA SMĚRY",  
GAME\_RULES12: "ROZTOČENÍ ZDARMA SE MOHOU SPUSTIT ZNOVU",  
},  
BETSETTINGS: "NASTAVENÍ SÁZEK",  
AUTOPLAY: {  
  AUTOPLAY\_TITLE: "AUTOMATICKÁ HRA",  
  AUTOPLAY\_SUBTITLE\_1: "Vyberte počet otočení",  
  AUTOPLAY\_SUBTITLE\_2: "zastavit automatickou hru",  
  AUTOPLAY\_DES\_1: "Pokud ztráta přesáhne",  
  AUTOPLAY\_DES\_2: "Pokud jednotlivá výhra přesáhne",  
  CLEAR: "Vyčistit",  
  PLAY: "Hrát",  
  WIN\_LIMIT\_EXCEED: "výhra přesáhne",  
  LOSS\_LIMIT\_EXCEED: "ztráta přesáhne"  
},  
SOUNDSETTINGS: {  
  SOUND: "ZVUK",  
  BACKGROUND\_SOUND: "ZVUK NA POZADÍ"  
},  
MSG: {  
  BIG\_WIN: "HEZKÁ VÝHRA",  
  HUGE\_WIN: "EXTRÉMNI VÝHRA",  
  EPIC\_WIN: "MASIVNÍ VÝHRA",  
  MEGA\_WIN: "OBROVSKÁ VÝHRA",  
  CONGRATULATIONS: "GRATULUJEME",

CREDITS\_WON: "VYHRANÉ KREDITY",  
COMBINATION: "KOMBINACE",  
AWARDED: "ZÍSKÁVÁTE",  
FREE\_SPINS: "ROZTOČENÍ ZDARMA",  
ADDITIONAL\_FREE\_SPINS\_AWARDED: "ZÍSKALI JSTE DALŠÍ ROZTOČENÍ  
ZDARMA",  
YOU\_HAVE\_WON: "VYHRÁLI JSTE",  
FREE\_SPINS\_AWARDED: "ZÍSKALI JSTE ROZTOČENÍ ZDARMA",  
REALITY\_CHECK\_LOSS\_LIMIT: "V kontrole reality překročen limit maximální  
ztráty",  
REALITY\_CHECK\_SESSION\_LENGTH: "V kontrole reality překročena délka relace",  
CLICK\_TO\_START: "Kliknutím začnete",  
SESSION\_TIME: "Čas relace",  
WIN\_AMOUNT: "Částka výhry",  
LOSS\_AMOUNT: "Částka prohry",  
UNFINISHED\_GAME\_ROUND: "NEDOKONČENÁ KOLA HRY"  
},  
BUTTON: {  
  OK: "OK",  
  CANCEL: "ZRUŠIT",  
  CONTINUE\_PLAYING: "POKRAČOVAT VE HŘE",  
  CLEAR: "SMAZAT",  
  CONTINUE: "POKRAČOVAT",  
  AH: "Historie účtu",  
  RL: "ZPĚT DO LOBBY"  
},  
ERROR: {  
  101: "Neplatná vstupenka",  
  404: "Chyba komunikace",  
  501: "Chyba systému. Kontaktujte podporu a sdělte jí kód chyby 501. ",  
  502: "Chyba systému. Kontaktujte podporu a sdělte jí kód chyby 502",  
  503: "Relace chybí nebo jí vypršela platnost",  
  504: "Chyba systému. Kontaktujte podporu a sdělte jí kód chyby 504",  
  505: "Neplatný postup ve hře. Kontaktujte podporu a sdělte jí chybový kód 505",  
  506: "Na provedení zvolené sázky nemáte dostatek prostředků ",  
  507: "Neplatná sázka. Kontaktujte podporu a sdělte jí chybový kód 507 ",  
  508: "Chyba příkazu systému. Kontaktujte podporu a sdělte jí chybový kód 508",  
  509: "Chyba systému. Kontaktujte podporu a sdělte jí kód chyby 509",  
  516: "Chyba systému. Kontaktujte podporu a sdělte jí kód chyby 516",  
  517: "Chyba systému. Kontaktujte podporu a sdělte jí kód chyby 517",  
  518: "Chyba systému. Kontaktujte podporu a sdělte jí kód chyby 518",  
  519: "Neplatné přihlášení. Kontaktujte podporu a sdělte jí chybový kód 519 ",  
  520: "Hra je aktivní v jiném prohlížeči. ",  
  521: "Technická chyba během přihlašování ",  
  522: "Chyba systému. Kontaktujte podporu a sdělte jí kód chyby 522",  
  523: "Chyba systému. Kontaktujte podporu a sdělte jí kód chyby 523",  
  524: "Hra vypnuta. ",  
  525: "Chyba měny. Kontaktujte podporu a sdělte jí chybový kód 525",  
  526: "Chyba přihlášení. Kontaktujte podporu a sdělte jí chybový kód 526",  
  527: "Chyba roztočení zdarma. Kontaktujte podporu a sdělte jí chybový kód 527",  
  528: "Neplatná adresa URL. ",  
  531: "Hra zablokována ",

```

532: "Chyba roztočení zdarma. Kontaktujte podporu a sdělte jí chybový kód 532",
533: "Chyba tokenu. Kontaktujte podporu a sdělte jí chybový kód 533",
534: "Pokus o přihlášení z blokováného regionu ",
535: "Verze hry je příliš stará. ",
536: "Chyba tokenu. Kontaktujte podporu a sdělte jí chybový kód 536",
537: "Chyba přihlášení. Kontaktujte podporu a sdělte jí chybový kód 537",
538: "Chyba jackpotu. Kontaktujte podporu a sdělte jí chybový kód 538",
539: "Chyba jackpotu. Kontaktujte podporu a sdělte jí chybový kód 539",
540: "Hráč má aktivní hru na jiném serveru. ",
541: "Chyba systému. Kontaktujte podporu a sdělte jí kód chyby 541 ",
542: "Chyba verze. Kontaktujte podporu a sdělte jí chybový kód 542",
543: "Chyba systému. Kontaktujte podporu a sdělte jí kód chyby 543",
544: "Překročen rozpočet útraty ",
545: "V kontrole reality překročen limit maximální ztráty ",
546: "V kontrole reality překročena délka relace ",
547: "Chyba měny. Kontaktujte podporu a sdělte jí chybový kód 547",
548: "Relace kontroly času stráveného ve hře Playtech vypršela ",
549: "Chyba tokenu. Kontaktujte podporu a sdělte jí chybový kód 549",
550: "Dosáhli jste maximální výhry ",
551: "Chyba systému. Kontaktujte podporu a sdělte jí kód chyby 551 ",
554: "Blocked from playing - daily time limit exceeded",
555: "Blocked from playing - weekly time limit exceeded",
556: "Blocked from playing - monthly time limit exceeded"
}
}, cc._RF.pop()
}, {}],
da: [function(e, t, i) {
"use strict";
cc._RF.push(t, "5c065gCRQNEMoZXSYYvdiE6", "da"), t.exports = {
WELCOME: "Welcome, %{name}!",
Click_anywhere_on_the_screen_to_start_the_game: "Klik hvor som helst på skærmen for at
starte spillet",
HUD: {
BET: "INDSATS",
CREDITS: "KREDITTER",
COIN_VALUE: "MØNT VÆRDI",
WIN: "GEVISNT",
AUTOPLAY: "AUTO SPIL",
MAXBET: "MAKS INDSATS",
"576WAYS_TO_WIN": "567 MÅDER AT VINDE PÅ!",
WIN_UP_TO_128_FREE_SPINS: "VIND OP TIL 128 GRATIS SPIL",
FREE: "GRATIS"
},
HUD_MESSAGES: {
EXTEND_MATRIX: "Udvider matrixen",
PAY_BOTH_WAYS_MATRIX: "Udbetaler begge veje og udvider matrixen",
ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Aktiverer blandet
udbetaling og udvider matrixen",
IS_ACTIVE_ALL: " Når den Røde Måne er aktiv, antager alle figur deres dyreform og
deler gevinster (blandet udbetaling)",
"3X": "3x",
DURING_THE_FREE_SPIN: "Under gratis spins er spillet altid 5x5 og udbetaler begge

```

```
veje",
  TRIGGER_6_FREE_SPINS: "udløser 6 gratis spins.",
  FREE_SPIN_CAN_RETRIGGER: "Gratis spins kan genudløses",
  GOOD_LUCK: "Good Luck",
  ADDITIONAL_FREE_SPINS_AWARDED: "ADDITIONAL FREE SPINS
AWARDED",
  X_FREE_PLAY: "%{remainingCount} gratis spil tilbage",
  X_FREE_PLAY_COMPLETED: "free plays remaining",
  FREE_SPIN_AWARDED: "du har fået tildelt %{spinCount} gratis spil",
  FREE_SPIN_OVER: "gratis spil er slut",
  NO_THANKS: "nej, tak",
  PLAY_LATER: "spil senere",
  PLAY_NOW: "spil nu"
},
PAYTABLE: {
  PAYTABLE: "SPILLEREGLER",
  PAYS_BOTH_WAYS: "Udbetaler begge veje og udvider matrixen",
  EXTENDS_MATRIX: "Udvider matrixen",
  ACTIVATES_MIXED: "Aktiverer blandet udbetaling og udvider matrixen",
  TRIGGER_6_FREE_SPIN: "udløser 6 gratis spins",
  AND_PAYS: "5x5 og udbetaler begge veje",
  WHEN_THE_RED: "Når den Røde Måne er aktiv, antager alle figur deres dyreform og
deler gevinster (blandet udbetaling)",
  MALFUNCTION_TEXT: "TEKNISKE FEJL ANNULLERER ALLE
UDBETALINGER OG SPIL",
  GAME_RULES: "SPILLEREGLER",
  GAME_RULES1: "VOLSUNGA ER ET MÅDE-SPIL MED EN UDVIDENDE 53135,
53335, 55355 OG 55555 MATRIX",
  GAME_RULES2: "SPILLERE KAN VINDE PÅ 3125 MÅDER",
  GAME_RULES3: "SPILLET HAR EN RTP PÅ 97,16 %",
  GAME_RULES4: "ALLE UDBETALINGER, BORTSET FRA SCATTER-
UDBETALINGER, UDBETALER FRA VENSTRE MOD HØJRE, STARTENDE PÅ HJULET
LÆNGST TIL VENSTRE",
  GAME_RULES5: "SCATTER-UDBETALINGER BLIVER UDBETALT FOR
SYMBOLER, DER FOREKOMMER HVOR SOM HELST PÅ HJULENE",
  GAME_RULES6: "KUN DEN HØJESTE SYMBOLGEVINST PR. VINDENDE
KOMBINATION BLIVER UDBETALT",
  GAME_RULES7: "KUN DEN HØJESTE SCATTERGEVINST BLIVER UDBETALT
PR. SPIL ",
  GAME_RULES8: "SCATTER-UDBETALINGER UDBETALES UD OVER MÅDE-
UDBETALINGER",
  GAME_RULES9: "MÅDE-UDBETALINGER BLIVER GANGET MED
INDSATSEN",
  GAME_RULES10: "SCATTER-UDBETALINGER BLIVER GANGET MED DEN
SAMLEDE INDSATS",
  GAME_RULES11: "UNDER GRATIS SPINS ER HJULENE ALTID 5X5 OG
UDBETALER ALTID BEGGE VEJE",
  GAME_RULES12: "GRATIS SPINS KAN BLIVE GENUDLØST"
},
BETSETTINGS: "INDSATS INDSTILLINGER",
AUTOPLAY: {
  AUTOPLAY_TITLE: "AUTO SPIL",
```

```

AUTOPLAY_SUBTITLE_1: "Vælg antal spil",
AUTOPLAY_SUBTITLE_2: "Stop auto spil",
AUTOPLAY_DES_1: "Hvis tab overstiger",
AUTOPLAY_DES_2: "Hvis enkelt gevinst overstiger",
CLEAR: "Klar",
PLAY: "Spil",
WIN_LIMIT_EXCEED: "gevinst overstiger",
LOSS_LIMIT_EXCEED: "tab overstiger"
},
SOUNDSETTINGS: {
    SOUND: "LYD",
    BACKGROUND SOUND: "BAGGRUNDSLYD"
},
MSG: {
    BIG_WIN: "LÆKKER GEVINST",
    HUGE_WIN: "EKSTREM GEVINST",
    EPIC_WIN: "MASSIV GEVINST",
    MEGA_WIN: "ENORM GEVINST",
    CONGRATULATIONS: "TILLYKKE",
    CREDITS_WON: "KREDITTER VUNDET",
    COMBINATION: "KOMBINATION(ER)",
    AWARDED: "TILDDDELT",
    FREE_SPINS: "GRATIS SPINS",
    ADDITIONAL_FREE_SPINS_AWARDED: "EKSTRA GRATIS SPINS TILDELDT",
    YOU_HAVE_WON: "DU HAR VUNDET",
    FREE_SPINS_AWARDED: "GRATIS SPINS TILDELDT",
    REALITY_CHECK_LOSS_LIMIT: "Realitetstjek tabsgrænse overskredet",
    REALITY_CHECK_SESSION_LENGTH: "Realitetstjek sessionslængde overskredet",
    CLICK_TO_START: "Klik for at starte",
    SESSION_TIME: "Sessionstid",
    WIN_AMOUNT: "Vundet beløb",
    LOSS_AMOUNT: "Tabt beløb",
    UNFINISHED_GAME_ROUND: "UAFSLUTTEDE SPILLERUNDER"
},
BUTTON: {
    OK: "OK",
    CANCEL: "ANNULLERET",
    CONTINUE_PLAYING: "FORTSÆT MED AT SPILLE",
    CLEAR: "RYD",
    CONTINUE: "FORTSÆT",
    AH: "Kontohistorik",
    RL: "VEND TILBAGE TIL LOBBY"
},
ERROR: {
    101: "Ugyldig billet",
    404: "kommunikationsfejl",
    501: "Systemfejl. Kontakt venligst support med fejlkoden 501 ",
    502: "Systemfejl. Kontakt venligst support med fejlkoden 502 ",
    503: "Sessionen mangler eller er udløbet",
    504: "Systemfejl. Kontakt venligst support med fejlkoden 504 ",
    509: "Systemfejl. Kontakt venligst support med fejlkoden 509 ",
    516: "Systemfejl. Kontakt venligst support med fejlkoden 516 ",

```

517: "Systemfejl. Kontakt venligst support med fejlkoden 517 ",  
518: "Systemfejl. Kontakt venligst support med fejlkoden 518 ",  
522: "Systemfejl. Kontakt venligst support med fejlkoden 522 ",  
523: "Systemfejl. Kontakt venligst support med fejlkoden 523 ",  
541: "Systemfejl. Kontakt venligst support med fejlkoden 541 ",  
543: "Systemfejl. Kontakt venligst support med fejlkoden 543 ",  
551: "Systemfejl. Kontakt venligst support med fejlkoden 551 ",  
505: "Ugyldig spilfremgang. Kontakt venligst support med fejlkode 505",  
507: "Ugyldig indsats. Kontakt venligst support med fejlkode 507 ",  
508: "Systemkommandofejl. Kontakt venligst support med fejlkode 508 ",  
519: "Ugyldigt login. Kontakt venligst support med fejlkode 519",  
520: "Spillet er aktivt i en anden browser. ",  
525: "Valutafejl. Kontakt venligst support med fejlkode 525",  
526: "Login-fejl. Kontakt venligst support med fejlkode 526",  
527: "Gratis spin-fejl. Kontakt venligst support med fejlkode 527",  
528: "Ugyldig URL. ",  
532: "Gratis spin-fejl. Kontakt venligst support med fejlkode 532",  
533: "Token-fejl. Kontakt venligst support med fejlkode 533",  
535: "Spilversionen er forældet. ",  
536: "Token-fejl. Kontakt venligst support med fejlkode 536",  
537: "Login-fejl. Kontakt venligst support med fejlkode 537",  
538: "Jackpotfejl. Kontakt venligst support med fejlkode 538",  
539: "Jackpotfejl. Kontakt venligst support med fejlkode 539",  
540: "Spilleren har et aktivt spil i en anden server. ",  
542: "Versionsfejl. Kontakt venligst support med fejlkode 542",  
544: "Forbrugsbudget overskredet ",  
547: "Valutafejl. Kontakt venligst support med fejlkode 547",  
549: "Token-fejl. Kontakt venligst support med fejlkode 549",  
550: "Gevinstgrænse nået ",  
506: "Der er ikke nok midler til den valgte indsatsstørrelse ",  
521: "Der opstod en teknisk fejl under login ",  
524: "Spil deaktiveret ",  
531: "Blokeret fra at spille",  
534: "Login forsøgt fra en blokeret region ",  
545: "Realitetstjek tabsgrænse overskredet ",  
546: "Realitetstjek sessionslængde overskredet ",  
548: "Playtech realitetstjek session udløbet ",  
554: "Blocked from playing - daily time limit exceeded",  
555: "Blocked from playing - weekly time limit exceeded",  
556: "Blocked from playing - monthly time limit exceeded"

}

}, cc.\_RF.pop()

}, {}],

de: [function(e, t, i) {

"use strict";

cc.\_RF.push(t, "31196t6IY1AgIVLLc3yVXd9", "de"), t.exports = {

WELCOME: "Welcome, %{name}!",

Click\_anywhere\_on\_the\_screen\_to\_start\_the\_game: "Klicke irgendwo auf dem Bildschirm um das Spiel zu starten",

HUD: {

BET: "EINSATZ",

CREDITS: "CREDITS",

```

COIN_VALUE: "MÜNZWERT",
WIN: "GEWINN",
AUTOPLAY: "AUTO SPIEL",
MAXBET: "MAX EINSATZ",
"576WAYS_TO_WIN": "576 WEGE ZU GEWINNEN!",
WIN_UP_TO_128_FREE_SPINS: "GEWINNE BIS ZU 128 GRATIS DREHS",
FREE: "GRATIS"
},
HUD_MESSAGES: {
    EXTEND_MATRIX: "Erweitert die Matrix",
    PAY_BOTH_WAYS_MATRIX: "Zahlt in beide Richtungen aus und erweitert die
Matrix",
    ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Aktiviert gemischte
Auszahlungen und erweitert die Matrix",
    IS_ACTIVE_ALL: " Wenn der Rote Mond aktiv ist, verwandeln sich alle Buchstaben in
ihre Tierform und teilen die Gewinne (gemischte Auszahlung)",
    "3X": "3x",
    DURING_THE_FREE_SPIN: "Während der Freispiele ist das Spiel immer 5x5 groß und
zahlt in beide Richtungen aus",
    TRIGGER_6_FREE_SPINS: "lösen 6 Freispiele aus.",
    FREE_SPIN_CAN_RETRIGGER: "FREISPIELE KÖNNEN ERNEUT AUSGELÖST
WERDEN",
    GOOD_LUCK: "Good Luck",
    ADDITIONAL_FREE_SPINS_AWARDED: "ADDITIONAL FREE SPINS
AWARDED",
    X_FREE_PLAY: "%{remainingCount} Freispiele verbleibend",
    X_FREE_PLAY_COMPLETED: "Freispiele verbleibend",
    FREE_SPIN_AWARDED: "Sie haben %{spinCount} Freispiele erhalten",
    FREE_SPIN_OVER: "Freispiele beendet",
    NO_THANKS: "Nein, danke",
    PLAY_LATER: "Später spielen",
    PLAY_NOW: "Jetzt spielen"
},
PAYTABLE: {
    PAYTABLE: "GEWINNTABELLE",
    PAYS_BOTH_WAYS: "Zahlt in beide Richtungen aus und erweitert die Matrix",
    EXTENDS_MATRIX: "Erweitert die Matrix",
    ACTIVATES_MIXED: "Aktiviert gemischte Auszahlungen und erweitert die Matrix",
    TRIGGER_6_FREE_SPIN: "lösen 6 Freispiele aus",
    AND_PAYS: "5x5 und zahlt in beide Richtungen aus",
    WHEN_THE_RED: "Wenn der Rote Mond aktiv ist, verwandeln sich alle Buchstaben in
ihre Tierform und teilen die Gewinne (gemischte Auszahlung)",
    MALFUNCTION_TEXT: "BEI EINER FEHLFUNKTION WERDEN ALLE
AUSZAHLUNGEN UND SPIELE UNGÜLTIG",
    GAME_RULES: "SPIELREGELN",
    GAME_RULES1: "VOLSUNGA IST EIN GEWINNWEGE-SPIEL MIT EINER
ERWEITERNDEN 53135-, 53335-, 53355- UND 55555-MATRIX",
    GAME_RULES2: "DIE SPIELER GEWINNEN AUF 3.125 GEWINNWEGEN",
    GAME_RULES3: "DIE AUSSCHÜTTUNGSQUOTE BETRÄGT 97,16 %",
    GAME_RULES4: "ALLE GEWINNE, AUSSER SCATTER, WERDEN VON LINKS
NACH RECHTS, BEGINNEND AUF DER ÄUSSERSTEN LINKEN WALZE, AUSGEZAHLT",
    GAME_RULES5: "SCATTER-GEWINNE WERDEN FÜR SYMBOLE AUF

```

BELIEBIGEN WALZENPOSITIONEN AUSGEZAHLT",  
 GAME\_RULES6: "ES WIRD LEDIGLICH DER HÖCHSTE SYMBOLGEWINN PRO  
 GEWINNKOMBINATION AUSGEZAHLT",  
 GAME\_RULES7: "ES WIRD LEDIGLICH DER HÖCHSTE SYMBOLGEWINN PRO  
 SPIEL AUSGEZAHLT",  
 GAME\_RULES8: "SCATTER-GEWINNE WERDEN ZUSÄTZLICH ZU DEN  
 GEWINNWEGEN AUSGEZAHLT",  
 GAME\_RULES9: "GEWINNWEGE WERDEN MIT DEM  
 EINSATZMULTIPLIKATOR MULTIPLIZIERT",  
 GAME\_RULES10: "SCATTER-GEWINNE WERDEN MIT DEM  
 GESAMTGUTHABENEINSATZ MULTIPLIZIERT",  
 GAME\_RULES11: "WÄHREND DER FREISPIELE HAT DAS SPIEL IMMER EINE  
 GRÖSSE VON 5X5 UND ZAHLT IN BEIDE RICHTUNGEN AUS",  
 GAME\_RULES12: "FREISPIELE KÖNNEN ERNEUT AUSGELÖST WERDEN"  
 },  
 BETSETTINGS: "EINSATZ EINSTELLUNGEN",  
 AUTOPLAY: {  
 AUTOPLAY\_TITLE: "AUTO SPIEL",  
 AUTOPLAY\_SUBTITLE\_1: "Wähle Anzahl an Drehs",  
 AUTOPLAY\_SUBTITLE\_2: "Stopp Auto Spiel",  
 AUTOPLAY\_DES\_1: "Wenn Verluste höher sind als",  
 AUTOPLAY\_DES\_2: "Wenn ein einzelner Gewinn höher ist als",  
 CLEAR: "Leeren",  
 PLAY: "Spielen",  
 WIN\_LIMIT\_EXCEED: "Gewinn höher ist",  
 LOSS\_LIMIT\_EXCEED: "höher sind als"  
 },  
 SOUNDSETTINGS: {  
 SOUND: "TON",  
 BACKGROUND\_SOUND: "HINTERGRUNDMUSIK"  
 },  
 MSG: {  
 BIG\_WIN: "SCHÖNER GEWINN",  
 HUGE\_WIN: "EXTREMER GEWINN",  
 EPIC\_WIN: "MASSIVER GEWINN",  
 MEGA\_WIN: "ENORMER GEWINN",  
 CONGRATULATIONS: "GLÜCKWUNSCH",  
 CREDITS\_WON: "CREDITS GEWONNEN",  
 COMBINATION: "KOMBINATION(EN)",  
 AWARDED: "GEWONNEN",  
 FREE\_SPINS: "FREISPIELE",  
 ADDITIONAL\_FREE\_SPINS\_AWARDED: "ZUSÄTZLICHE FREISPIELE  
 GEWONNEN",  
 YOU\_HAVE\_WON: "SIE GEWINNEN",  
 FREE\_SPINS\_AWARDED: "FREISPIELE GEWONNEN",  
 REALITY\_CHECK\_LOSS\_LIMIT: "Realitäts-Check-Verlustlimit überschritten",  
 REALITY\_CHECK\_SESSION\_LENGTH: "Realitäts-Check-Sitzungsdauer  
 überschritten",  
 CLICK\_TO\_START: "Zum Starten klicken",  
 SESSION\_TIME: "Sitzungszeit",  
 WIN\_AMOUNT: "Gewinnbetrag",  
 LOSS\_AMOUNT: "Verlustbetrag",

```

UNFINISHED_GAME_ROUND: "UNBEENDETE SPIELRUNDEN"
},
BUTTON: {
    OK: "OKAY",
    CANCEL: "ABBRECHEN",
    CONTINUE_PLAYING: "WEITERSPIELEN",
    CLEAR: "LÖSCHEN",
    CONTINUE: "WEITER",
    AH: "Kontoverlauf",
    RL: "ZURÜCK ZUR LOBBY"
},
ERROR: {
    101: "Ungültiges Ticket",
    404: "Kommunikationsfehler",
    501: "Systemfehler. Bitte Support kontaktieren, unter Angabe von Fehlercode 501",
    502: "Systemfehler. Bitte Support kontaktieren, unter Angabe von Fehlercode 502",
    503: "Sitzung fehlt oder ist abgelaufen",
    504: "Systemfehler. Bitte Support kontaktieren, unter Angabe von Fehlercode 504",
    509: "Systemfehler. Bitte Support kontaktieren, unter Angabe von Fehlercode 509",
    516: "Systemfehler. Bitte Support kontaktieren, unter Angabe von Fehlercode 516",
    517: "Systemfehler. Bitte Support kontaktieren, unter Angabe von Fehlercode 517",
    518: "Systemfehler. Bitte Support kontaktieren, unter Angabe von Fehlercode 518",
    522: "Systemfehler. Bitte Support kontaktieren, unter Angabe von Fehlercode 522",
    523: "Systemfehler. Bitte Support kontaktieren, unter Angabe von Fehlercode 523",
    541: "Systemfehler. Bitte Support kontaktieren, unter Angabe von Fehlercode 541",
    543: "Systemfehler. Bitte Support kontaktieren, unter Angabe von Fehlercode 543",
    551: "Systemfehler. Bitte Support kontaktieren, unter Angabe von Fehlercode 551",
    505: "Ungültiger Spielfortschritt. Bitte wenden Sie sich mit dem Fehlercode 505 an den
Support. ",
    507: "Ungültiger Einsatz. Bitte wenden Sie sich mit dem Fehlercode 507 an den Support.
",
    508: "Systembefehlsfehler. Bitte wenden Sie sich mit dem Fehlercode 508 an den Support.
",
    519: "Ungültige Anmeldung. Bitte wenden Sie sich mit dem Fehlercode 519 an den
Support. ",
    520: "Spiel ist in einem anderen Browser aktiv.",
    525: "Währungsfehler. Bitte wenden Sie sich mit dem Fehlercode 525 an den Support. ",
    526: "Anmeldefehler. Bitte wenden Sie sich mit dem Fehlercode 526 an den Support.",
    537: "Anmeldefehler. Bitte wenden Sie sich mit dem Fehlercode 537 an den Support.",
    527: "Freispielfehler. Bitte wenden Sie sich mit dem Fehlercode 527 an den Support. ",
    532: "Freispielfehler. Bitte wenden Sie sich mit dem Fehlercode 532 an den Support. ",
    528: "Ungültige URL. ",
    533: "Token-Fehler. Bitte wenden Sie sich mit dem Fehlercode 533 an den Support. ",
    536: "Token-Fehler. Bitte wenden Sie sich mit dem Fehlercode 536 an den Support. ",
    549: "Token-Fehler. Bitte wenden Sie sich mit dem Fehlercode 549 an den Support. ",
    535: "Spielversion ist veraltet.",
    538: "Jackpotfehler. Bitte wenden Sie sich mit dem Fehlercode 538 an den Support. ",
    539: "Jackpotfehler. Bitte wenden Sie sich mit dem Fehlercode 539 an den Support. ",
    540: "Spieler hat ein aktives Spiel auf einem anderen Server. ",
    542: "Versionsfehler. Bitte wenden Sie sich mit dem Fehlercode 542 an den Support. ",
    544: "Ausgabebudget überschritten ",
    547: "Währungsfehler. Bitte wenden Sie sich mit dem Fehlercode 547 an den Support. ",

```

```

550: "Gewinnmaximum erreicht",
506: "Guthaben reicht für ausgewählte Einsatzgröße nicht aus",
521: "Technischer Fehler beim Login",
524: "Spiel deaktiviert",
531: "Vom Spielen ausgeschlossen",
534: "Login-Versuch aus ausgeschlossener Region ",
545: "Realitäts-Check-Verlustlimit überschritten ",
546: "Realitäts-Check-Sitzungsdauer überschritten ",
548: "Playtech-Realitäts-Check-Sitzung abgelaufen ",
554: "Blocked from playing - daily time limit exceeded",
555: "Blocked from playing - weekly time limit exceeded",
556: "Blocked from playing - monthly time limit exceeded"
}
}, cc._RF.pop()
}, {}],
el: [function(e, t, i) {
"use strict";
cc._RF.push(t, "640da6a11pDmp+KYHL9XdVo", "el"), t.exports = {
WELCOME: "Welcome, %{name}!",
Click_anywhere_on_the_screen_to_start_the_game: "Κάντε κλικ οπουδήποτε στην οθόνη
για να ξεκινήσετε το παιχνίδι",
HUD: {
BET: "ΠΟΝΤΑΡΙΣΜΑ",
CREDITS: "ΝΟΜΙΣΜΑΤΑ",
COIN_VALUE: "ΑΞΙΑ ΝΟΜΙΣΜΑΤΟΣ",
WIN: "ΝΙΚΗ",
AUTOPLAY: "ΑΥΤΟΜ. ΠΑΙΧΝΙΔΙ",
MAXBET: "ΜΕΓΙΣΤΟ ΠΟΝΤΑΡΙΣΜΑ",
"576WAYS_TO_WIN": "576 ΤΡΟΠΟΙ ΝΑ ΚΕΡΔΙΣΕΤΕ!",
WIN_UP_TO_128_FREE_SPINS: "ΚΕΡΔΙΣΤΕ ΜΕΧΡΙ ΚΑΙ 128 ΔΩΡΕΑΝ
ΠΕΡΙΣΤΡΟΦΕΣ",
FREE: "ΔΩΡΕΑΝ"
},
HUD_MESSAGES: {
EXTEND_MATRIX: "Επεκτείνει το πλέγμα",
PAY_BOTH_WAYS_MATRIX: "Πληρώνει και με τους δύο τρόπους και επεκτείνει το
πλέγμα",
ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Ενεργοποιεί μεικτά
παιχνίδια και επεκτείνει το πλέγμα",
IS_ACTIVE_ALL: "Όταν ενεργοποιηθεί η Κόκκινη Σελήνη, όλοι οι χαρακτήρες
αλλάζουν στη ζωώδη μορφή τους και μοιράζονται τα κέρδη (μεικτές πληρωμές)",
"3X": "3x",
DURING_THE_FREE_SPIN: "Κατά τις δωρεάν περιστροφές, το παιχνίδι είναι πάντα
5x5 και πληρώνει πάντα και με τους δύο τρόπους",
TRIGGER_6_FREE_SPINS: "ενεργοποιούν 5 δωρεάν περιστροφές.",
FREE_SPIN_CAN_RETRIGGER: "Οι Δωρεάν περιστροφές μπορούν να ενεργοποιηθούν
ξανά",
GOOD_LUCK: "Good Luck",
ADDITIONAL_FREE_SPINS_AWARDED: "ADDITIONAL FREE SPINS
AWARDED",
X_FREE_PLAY: "απομένουν %{remainingCount} δωρεάν παιχνίδια",
X_FREE_PLAY_COMPLETED: "απομένουν δωρεάν παιχνίδια",

```

```

FREE_SPIN_AWARDED: "λάβετε % {spinCount} δωρεάν παιχνίδια",
FREE_SPIN_OVER: "τα δωρεάν παιχνίδια τελείωσαν",
NO_THANKS: "όχι, ευχαριστώ",
PLAY_LATER: "παίξτε αργότερα",
PLAY_NOW: "παίξτε τώρα"
},
PAYTABLE: {
  PAYTABLE: "ΠΙΝΑΚΑΣ ΠΛΗΡΩΜΩΝ",
  PAYS_BOTH_WAYS: "Πληρώνει και με τους δύο τρόπους και επεκτείνει το πλέγμα",
  EXTENDS_MATRIX: "Επεκτείνει το πλέγμα",
  ACTIVATES_MIXED: "Ενεργοποιεί μεικτά παιχνίδια και επεκτείνει το πλέγμα",
  TRIGGER_6_FREE_SPIN: "ενεργοποιούν 6 δωρεάν περιστροφές",
  AND_PAYS: "5x5 και πληρώνει και με τους δύο τρόπους",
  WHEN_THE_RED: "Όταν ενεργοποιηθεί η Κόκκινη Σελήνη, όλοι οι χαρακτήρες
αλλάζουν στη ζωώδη μορφή τους και μοιράζονται τα κέρδη (μεικτές πληρωμές)",
  MALFUNCTION_TEXT: "ΤΥΧΟΝ ΔΥΣΛΕΙΤΟΥΡΓΙΕΣ ΑΚΥΡΩΝΟΥΝ ΟΛΕΣ ΤΙΣ
ΠΛΗΡΩΜΕΣ ΚΑΙ ΤΑ ΠΑΙΧΝΙΔΙΑ",
  GAME_RULES: "ΚΑΝΟΝΕΣ ΠΑΙΧΝΙΔΙΟΥ",
  GAME_RULES1: "ΤΟ VOLSUNGA ΕΙΝΑΙ ΠΑΙΧΝΙΔΙ ΤΡΟΠΩΝ ΜΕ
ΕΠΕΚΤΕΙΝΟΜΕΝΟ ΠΛΕΓΜΑ 53135, 53335, 55355 ΚΑΙ 55555",
  GAME_RULES2: "ΟΙ ΠΑΙΚΤΕΣ/ΠΑΙΚΤΡΙΕΣ ΕΧΟΥΝ 3125 ΤΡΟΠΟΥΣ ΝΑ
ΚΕΡΔΙΣΟΥΝ",
  GAME_RULES3: "ΤΟ ΠΑΙΧΝΙΔΙ ΈΧΕΙ 97,16% RTP.",
  GAME_RULES4: "ΟΛΕΣ ΟΙ ΠΛΗΡΩΜΕΣ ΕΚΤΟΣ ΤΩΝ ΠΛΗΡΩΜΩΝ SCATTER
ΓΙΝΟΝΤΑΙ ΑΠΟ ΑΡΙΣΤΕΡΑ ΠΡΟΣ ΤΑ ΔΕΞΙΑ ΞΕΚΙΝΩΤΑΣ ΑΠΟ ΤΟΝ ΠΡΩΤΟ ΑΠΟ
ΑΡΙΣΤΕΡΑ ΚΥΛΙΝΔΡΟ",
  GAME_RULES5: "ΟΙ ΠΛΗΡΩΜΕΣ SCATTER ΠΛΗΡΩΝΟΥΝ ΓΙΑ ΣΥΜΒΟΛΑ ΠΟΥ
ΕΜΦΑΝΙΖΟΝΤΑΙ ΣΕ ΟΠΟΙΑΔΗΠΟΤΕ ΘΕΣΗ ΣΤΟΥΣ ΚΥΛΙΝΔΡΟΥΣ",
  GAME_RULES6: "ΜΟΝΟ ΤΟ ΥΨΗΛΟΤΕΡΟ ΚΕΡΔΟΣ ΣΥΜΒΟΛΟΥ ΠΛΗΡΩΝΕΙ
ΑΝΑ ΝΙΚΗΤΗΡΙΟ ΣΥΝΔΥΑΣΜΟ",
  GAME_RULES7: "ΜΟΝΟ ΤΟ ΥΨΗΛΟΤΕΡΟ ΚΕΡΔΟΣ SCATTER ΠΛΗΡΩΝΕΙ ΑΝΑ
ΠΑΙΧΝΙΔΙ",
  GAME_RULES8: "ΟΙ ΠΛΗΡΩΜΕΣ SCATTER ΓΙΝΟΝΤΑΙ ΕΠΙΠΛΕΟΝ ΤΩΝ
ΠΛΗΡΩΜΩΝ ΤΡΟΠΩΝ",
  GAME_RULES9: "ΟΙ ΠΛΗΡΩΜΕΣ ΤΡΟΠΩΝ ΠΟΛΛΑΠΛΑΣΙΑΖΟΝΤΑΙ ΜΕ ΤΟΝ
ΠΟΛΛΑΠΛΑΣΙΑΣΤΗ ΠΟΝΤΑΡΙΣΜΑΤΟΣ",
  GAME_RULES10: "ΟΙ ΠΛΗΡΩΜΕΣ SCATTER ΠΟΛΛΑΠΛΑΣΙΑΖΟΝΤΑΙ ΜΕ ΤΟ
ΣΥΝΟΛΙΚΟ ΠΟΝΤΑΡΙΣΜΑ ΜΟΝΑΔΩΝ",
  GAME_RULES11: "ΚΑΤΑ ΤΙΣ ΔΩΡΕΑΝ ΠΕΡΙΣΤΡΟΦΕΣ, ΤΟ ΠΑΙΧΝΙΔΙ ΕΙΝΑΙ
ΠΑΝΤΑ 5X5 ΚΑΙ ΠΛΗΡΩΝΕΙ ΠΑΝΤΑ ΚΑΙ ΜΕ ΤΟΥΣ ΔΥΟ ΤΡΟΠΟΥΣ",
  GAME_RULES12: "Οι Δωρεάν περιστροφές μπορούν να ενεργοποιηθούν ξανά"
},
BETSETTINGS: "ΡΥΘΜΙΣΕΙΣ ΠΟΝΤΑΡΙΣΜΑΤΟΣ",
AUTOPLAY: {
  AUTOPLAY_TITLE: "ΑΥΤΟΜ. ΠΑΙΧΝΙΔΙ",
  AUTOPLAY_SUBTITLE_1: "Επιλέξτε αριθμό περιστροφών",
  AUTOPLAY_SUBTITLE_2: "Διακοπή αυτομ. παιχνιδιού",
  AUTOPLAY_DES_1: "Εάν η ήττα υπερβαίνει",
  AUTOPLAY_DES_2: "Εάν η απλή νίκη υπερβαίνει",
  CLEAR: "Καθαρισμός",
  PLAY: "Παιχνίδι",
  WIN_LIMIT_EXCEED: "νίκη υπερβαίνει",

```

```

LOSS_LIMIT_EXCEED: "ήττα υπερβαίνει"
},
SOUNDSETTINGS: {
    SOUND: "ΗΧΟΣ",
    BACKGROUNDSOUND: "ΗΧΟΣ ΥΠΟΒΑΘΡΟΥ"
},
MSG: {
    BIG_WIN: "ΩΡΑΙΟ ΚΕΡΔΟΣ",
    HUGE_WIN: "ΤΕΡΑΣΤΙΑ ΝΙΚΗ",
    EPIC_WIN: "ΠΟΛΥ ΜΕΓΑΛΗ ΝΙΚΗ",
    MEGA_WIN: "ΤΕΡΑΣΤΙΑ ΝΙΚΗ",
    CONGRATULATIONS: "ΣΥΓΧΑΡΗΤΗΡΙΑ",
    CREDITS_WON: "ΚΕΡΔΙΣΜΕΝΑ ΝΟΜΙΣΜΑΤΑ",
    COMBINATION: "ΣΥΝΔΥΑΣΜΟΣ (ΟΙ)",
    AWARDED: "ΑΠΟΝΕΜΗΘΗΚΑΝ",
    FREE_SPINS: "ΔΩΡΕΑΝ ΠΕΡΙΣΤΡΟΦΕΣ",
    ADDITIONAL_FREE_SPINS_AWARDED: "ΑΠΟΝΕΜΗΘΗΚΑΝ ΕΠΙΠΛΕΟΝ
ΔΩΡΕΑΝ ΠΕΡΙΣΤΡΟΦΕΣ",
    YOU_HAVE_WON: "HAS GANADO",
    FREE_SPINS_AWARDED: "SE RECOMPENSA CON GIROS GRATIS",
    REALITY_CHECK_LOSS_LIMIT: "Υπέρβαση ορίου απώλειας ελέγχου
πραγματικότητας",
    REALITY_CHECK_SESSION_LENGTH: "Υπέρβαση διάρκειας συνεδρίας ελέγχου
πραγματικότητας",
    CLICK_TO_START: "Κάντε κλικ για έναρξη",
    SESSION_TIME: "Χρόνος συνεδρίας",
    WIN_AMOUNT: "Ποσό κερδών",
    LOSS_AMOUNT: "Ποσό απωλειών",
    UNFINISHED_GAME_ROUND: "ΗΜΙΤΕΛΕΙΣ ΓΥΡΟΙ ΠΑΙΧΝΙΔΙΟΥ"
},
BUTTON: {
    OK: "OK",
    CANCEL: "ΑΚΥΡΩΣΗ",
    CONTINUE_PLAYING: "ΣΥΝΕΧΕΙΑ ΠΑΙΧΝΙΔΙΟΥ",
    CLEAR: "ΚΑΘΑΡΙΣΜΟΣ",
    CONTINUE: "ΣΥΝΕΧΕΙΑ",
    AH: "Ιστορικό λογαριασμού",
    RL: "ΕΠΙΣΤΡΟΦΗ ΣΤΟ LOBBY"
},
ERROR: {
    101: "Μη έγκυρο εισιτήριο",
    404: "Σφάλμα επικοινωνίας",
    501: "Σφάλμα συστήματος. Επικοινωνήστε με το τμήμα υποστήριξης, δίνοντας τον
κωδικό σφάλματος 501 ",
    502: "Σφάλμα συστήματος. Επικοινωνήστε με το τμήμα υποστήριξης, δίνοντας τον
κωδικό σφάλματος 502 ",
    503: "Η συνεδρία λείπει ή έχει λήξει ",
    504: "Σφάλμα συστήματος. Επικοινωνήστε με το τμήμα υποστήριξης, δίνοντας τον
κωδικό σφάλματος 504 ",
    509: "Σφάλμα συστήματος. Επικοινωνήστε με το τμήμα υποστήριξης, δίνοντας τον
κωδικό σφάλματος 509 ",
    516: "Σφάλμα συστήματος. Επικοινωνήστε με το τμήμα υποστήριξης, δίνοντας τον

```

κωδικό σφάλματος 516 ",  
517: "Σφάλμα συστήματος. Επικοινωνήστε με το τμήμα υποστήριξης, δίνοντας τον κωδικό σφάλματος 517 ",  
518: "Σφάλμα συστήματος. Επικοινωνήστε με το τμήμα υποστήριξης, δίνοντας τον κωδικό σφάλματος 518 ",  
522: "Σφάλμα συστήματος. Επικοινωνήστε με το τμήμα υποστήριξης, δίνοντας τον κωδικό σφάλματος 522 ",  
523: "Σφάλμα συστήματος. Επικοινωνήστε με το τμήμα υποστήριξης, δίνοντας τον κωδικό σφάλματος 523 ",  
541: "Σφάλμα συστήματος. Επικοινωνήστε με το τμήμα υποστήριξης, δίνοντας τον κωδικό σφάλματος 541 ",  
543: "Σφάλμα συστήματος. Επικοινωνήστε με το τμήμα υποστήριξης, δίνοντας τον κωδικό σφάλματος 543 ",  
551: "Σφάλμα συστήματος. Επικοινωνήστε με το τμήμα υποστήριξης, δίνοντας τον κωδικό σφάλματος 551 ",  
505: "Μη έγκυρη πρόοδος παιχνιδιού. Επικοινωνήστε με την υποστήριξη, αναφέροντας τον κωδικό σφάλματος 505.",  
507: "Μη έγκυρο ποντάρισμα. Επικοινωνήστε με την υποστήριξη, αναφέροντας τον κωδικό σφάλματος 507. ",  
508: "Σφάλμα εντολής συστήματος. Επικοινωνήστε με την υποστήριξη, αναφέροντας τον κωδικό σφάλματος 508.",  
519: "Μη έγκυρη σύνδεση. Επικοινωνήστε με την υποστήριξη, αναφέροντας τον κωδικό σφάλματος 519.",  
520: "Το παιχνίδι είναι ενεργό σε άλλο πρόγραμμα περιήγησης.",  
525: "Σφάλμα νομίσιματος. Επικοινωνήστε με την υποστήριξη, αναφέροντας τον κωδικό σφάλματος 525.",  
526: "Σφάλμα σύνδεσης. Επικοινωνήστε με την υποστήριξη, αναφέροντας τον κωδικό σφάλματος 526.",  
537: "Σφάλμα σύνδεσης. Επικοινωνήστε με την υποστήριξη, αναφέροντας τον κωδικό σφάλματος 537.",  
527: "Σφάλμα δωρεάν περιστροφών. Επικοινωνήστε με την υποστήριξη, αναφέροντας τον κωδικό σφάλματος 527.",  
532: "Σφάλμα δωρεάν περιστροφών. Επικοινωνήστε με την υποστήριξη, αναφέροντας τον κωδικό σφάλματος 532.",  
528: "Μη έγκυρη διεύθυνση URL.",  
533: "Σφάλμα διακριτικού. Επικοινωνήστε με την υποστήριξη, αναφέροντας τον κωδικό σφάλματος 533.",  
536: "Σφάλμα διακριτικού. Επικοινωνήστε με την υποστήριξη, αναφέροντας τον κωδικό σφάλματος 536.",  
549: "Σφάλμα διακριτικού. Επικοινωνήστε με την υποστήριξη, αναφέροντας τον κωδικό σφάλματος 549.",  
535: "Η έκδοση του παιχνιδιού είναι πολύ παλιά.",  
538: "Σφάλμα Τζακπότ. Επικοινωνήστε με την υποστήριξη, αναφέροντας τον κωδικό σφάλματος 538.",  
539: "Σφάλμα Τζακπότ. Επικοινωνήστε με την υποστήριξη, αναφέροντας τον κωδικό σφάλματος 539.",  
540: "Ο παίκτης έχει ενεργό παιχνίδι σε άλλον διακομιστή.",  
542: "Σφάλμα έκδοσης. Επικοινωνήστε με την υποστήριξη, αναφέροντας τον κωδικό σφάλματος 542.",  
544: "Υπέρβαση διαθέσιμου υπολοίπου",  
547: "Σφάλμα νομίσιματος. Επικοινωνήστε με την υποστήριξη, αναφέροντας τον κωδικό σφάλματος 547.",

```

550: "Συμπλήρωση ορίου κέρδους",
506: "Δεν υπάρχουν αρκετά χρήματα για το επιλεγμένο μέγεθος πονταρίσματος",
521: "Τεχνικό σφάλμα κατά τη σύνδεση",
524: "Το παιχνίδι απενεργοποιήθηκε",
531: "Υπό φραγή παιχνιδιού",
534: "Η απόπειρα σύνδεσης προέρχεται από φραγμένη περιοχή",
545: "Υπέρβαση ορίου απώλειας ελέγχου πραγματικότητας",
546: "Υπέρβαση διάρκειας συνεδρίας ελέγχου πραγματικότητας",
548: "Εληξε η περίοδος λειτουργίας ελέγχου πραγματικότητας Playtech",
554: "Blocked from playing - daily time limit exceeded",
555: "Blocked from playing - weekly time limit exceeded",
556: "Blocked from playing - monthly time limit exceeded"
}
}, cc._RF.pop()
}, {}],
en: [function(e, t, i) {
"use strict";
var n;
function a(e, t, i) {
return t in e ? Object.defineProperty(e, t, {
value: i,
enumerable: !0,
configurable: !0,
writable: !0
}): e[t] = i, e
}
cc._RF.push(t, "6e514t05O9Mj7VcZtpSMeOk", "en"), t.exports = {
WELCOME: "Welcome, %{name}!",
Click_anywhere_on_the_screen_to_start_the_game: "Click anywhere on the screen to start
the game",
HUD: {
BET: "BET",
CREDITS: "CREDITS",
COIN_VALUE: "COIN VALUE",
WIN: "WIN",
AUTOPLAY: "AUTO PLAY",
MAXBET: "MAX BET",
"576WAYS_TO_WIN": "576 WAYS TO WIN!",
WIN_UP_TO_128_FREE_SPINS: "WIN UP TO 128 FREE SPINS",
FREE: "Free"
},
HUD_MESSAGES: (n = {
EXTEND_MATRIX: "White Moon extends matrix",
PAY_BOTH_WAYS_MATRIX: "Blue Moon pays both ways and extends matrix",
ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Red Moon activates mixed
pays and extends matrix",
IS_ACTIVE_ALL: "When Red Moon is active, all characters share wins",
"3X": "3x",
DURING_THE_FREE_SPIN: "During free spins, the game is always 5x5 and always
pays both ways",
TRIGGER_6_FREE_SPINS: "3x Scatter trigger 6 free spins",
FREE_SPIN_CAN_RETRIGGER: "Free spins can retrigger",

```

GOOD\_LUCK: "Good Luck",  
 RED\_MOON: "Red Moon",  
 WHITE\_MOON: "White Moon"  
 }, a(n, "RED\_MOON", "Blue Moon"), a(n, "X\_FREE\_PLAY", "%{remainingCount} free  
 plays remaining"), a(n, "X\_FREE\_PLAY\_COMPLETED", "free plays remaining"), a(n,  
 "FREE\_SPIN\_AWARDED", "You have been awarded %{spinCount} free spins"), a(n,  
 "FREE\_SPIN\_OVER", "FREE SPINS ARE OVER"), a(n, "NO\_THANKS", "NO THANKS"), a(n,  
 "PLAY\_LATER", "PLAY LATER"), a(n, "PLAY\_NOW", "PLAY NOW"), n),  
 PAYTABLE: {  
 PAYTABLE: "PAYTABLE",  
 PAYS\_BOTH\_WAYS: "Pays both ways and extends matrix",  
 EXTENDS\_MATRIX: "Extends Matrix",  
 ACTIVATES\_MIXED: "Activates mixed pays and extends matrix",  
 TRIGGER\_6\_FREE\_SPIN: "trigger 6 free spins",  
 AND\_PAYS: "5x5 and pays both ways",  
 WHEN\_THE\_RED: "When the Red Moon is active, all characters shape-shift into their  
 animal form and share wins (mixed pays)",  
 MALFUNCTION\_TEXT: "Malfunction voids all pays and plays",  
 GAME\_RULES: "GAME RULES",  
 GAME\_RULES1: "Volsunga is a ways game with an extending 53135, 53335, 55355 and  
 55555 matrix",  
 GAME\_RULES2: "Players have 3125 ways to win",  
 GAME\_RULES3: "The game has a 97.16% RTP",  
 GAME\_RULES4: "All pays except scatter pays are left-to-right starting from the leftmost  
 reel",  
 GAME\_RULES5: "Scatter pays are paid for symbols appearing in any position on the  
 reels",  
 GAME\_RULES6: "Only the highest symbol win is paid per winning combination",  
 GAME\_RULES7: "Only the highest scatter win is paid per play",  
 GAME\_RULES8: "Scatter pays are paid in addition to ways pays",  
 GAME\_RULES9: "Ways pays are multiplied by the bet multiple",  
 GAME\_RULES10: "Scatter pays are multiplied by the total credits bet",  
 GAME\_RULES11: "During free spins the game is always 5x5 and always pays both  
 ways",  
 GAME\_RULES12: "Free spins can retrigger"  
 },  
 BETSETTINGS: "BET SETTINGS",  
 AUTOPLAY: {  
 AUTOPLAY\_TITLE: "AUTO PLAY",  
 AUTOPLAY\_SUBTITLE\_1: "Select number of spins",  
 AUTOPLAY\_SUBTITLE\_2: "Stop auto play",  
 AUTOPLAY\_DES\_1: "If loss exceeds",  
 AUTOPLAY\_DES\_2: "If single win exceeds",  
 CLEAR: "Clear",  
 PLAY: "Play",  
 WIN\_LIMIT\_EXCEED: "WIN LIMIT EXCEEDED",  
 LOSS\_LIMIT\_EXCEED: "LOSS LIMIT EXCEEDED"  
 },  
 SOUNDSETTINGS: {  
 SOUND: "SOUND",  
 BACKGROUND\_SOUND: "BACKGROUND SOUND"  
 },  
 },

```

MSG: {
    BIG_WIN: "NICE WIN",
    HUGE_WIN: "HUGE WIN",
    EPIC_WIN: "MASSIVE WIN",
    MEGA_WIN: "EXTREME WIN",
    CONGRATULATIONS: "CONGRATULATIONS",
    CREDITS_WON: "CREDITS WON",
    COMBINATION: "COMBINATION(S)",
    AWARDED: "AWARDED",
    FREE_SPINS: "FREE SPINS",
    ADDITIONAL_FREE_SPINS_AWARDED: "ADDITIONAL FREE SPINS
AWARDED",
    YOU_HAVE_WON: "YOU HAVE WON",
    FREE_SPINS_AWARDED: "FREE SPINS AWARDED",
    REALITY_CHECK_LOSS_LIMIT: "Reality check loss limit exceeded",
    REALITY_CHECK_SESSION_LENGTH: "Reality check session length exceeded",
    CLICK_TO_START: "Click to start",
    SESSION_TIME: "Session Time",
    WIN_AMOUNT: "Win Amount",
    LOSS_AMOUNT: "Loss Amount",
    UNFINISHED_GAME_ROUND: "UNFINISHED GAME ROUND"
},
BUTTON: {
    OK: "OK",
    CANCEL: "CANCEL",
    CONTINUE_PLAYING: "CONTINUE PLAYING",
    CLEAR: "CLEAR",
    CONTINUE: "CONTINUE",
    AH: "ACCOUNT HISTORY",
    RL: "RETURN TO LOBBY"
},
ERROR: {
    101: "Invalid Ticket",
    404: "Communication Error",
    501: "System error. Please contact support with error code 501",
    502: "System error. Please contact support with error code 502",
    503: "Session missing or expired",
    504: "System error. Please contact support with error code 504",
    505: "Invalid game progress. Please contact support with error code 505",
    506: "Not enough funds for chosen betsize ",
    507: "Invalid bet. Please contact support with error code 507",
    508: "System command error. Please contact support with error code 508",
    509: "System error. Please contact support with error code 509",
    516: "System error. Please contact support with error code 516",
    517: "System error. Please contact support with error code 517",
    518: "System error. Please contact support with error code 518",
    519: "Invalid login. Please contact support with error code 519",
    520: "Game is active in another browser.",
    521: "Technical error during login ",
    522: "System error. Please contact support with error code 522",
    523: "System error. Please contact support with error code 523",
    524: "Game disabled. ",

```

```

525: "Currency error. Please contact support with error code 525",
526: "Login error. Please contact support with error code 526",
527: "Freespin error. Please contact support with error code 527",
528: "Invalid URL. ",
531: "Blocked from playing ",
532: "Freespin error. Please contact support with error code 532",
533: "Token error. Please contact support with error code 533",
534: "Login attempt comes from blocked region ",
535: "Game version too old. ",
536: "Token error. Please contact support with error code 536",
537: "Login error. Please contact support with error code 537",
538: "Jackpot error. Please contact support with error code 538",
539: "Jackpot error. Please contact support with error code 539",
540: "Player has active game in another server. ",
541: "System error. Please contact support with error code 541",
542: "Version error. Please contact support with error code 542",
543: "System error. Please contact support with error code 543",
544: "Spending budget exceeded ",
545: "Reality check loss limit exceeded 545",
546: "Reality check session length exceeded 546",
547: "Currency error. Please contact support with error code 547",
548: "Playtech reality check session expired 548",
549: "Token error. Please contact support with error code 549",
550: "Win cap reached 550",
551: "System error. Please contact support with error code 551",
554: "Blocked from playing - daily time limit exceeded",
555: "Blocked from playing - weekly time limit exceeded",
556: "Blocked from playing - monthly time limit exceeded"
}
}, cc._RF.pop()
}, {}],
"es-eu": [function(e, t, i) {
"use strict";
cc._RF.push(t, "af288dzBWFGpocb2onzyjdT", "es-eu"), t.exports = {
WELCOME: "Welcome, %{name}!",
Click_anywhere_on_the_screen_to_start_the_game: "Haz click en la pantalla para empezar
el juego",
HUD: {
BET: "APUESTA",
CREDITS: "CREDITOS",
COIN_VALUE: "VALOR DE MONEDA",
WIN: "GANANCIA",
AUTOPLAY: "AUTOJUEGO",
MAXBET: "APUESTA MAXIMA",
"576WAYS_TO_WIN": "¡576 MANERAS DE GANANCIA!",
WIN_UP_TO_128_FREE_SPINS: "GANANCIA HASTA 128 GIROS GRATUITOS",
FREE: "GRATUITOS"
},
HUD_MESSAGES: {
EXTEND_MATRIX: "Amplía la matriz",
PAY_BOTH_WAYS_MATRIX: "Paga en ambos sentidos y amplía la matriz",
ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Activa los pagos

```

combinados y amplía la matriz",  
 IS\_ACTIVE\_ALL: " Cuando la Luna Roja esté activa, todos los personajes cambian a su forma animal y comparten los premios (pagos combinados)",  
 "3X": "3x",  
 DURING\_THE\_FREE\_SPIN: "Durante los giros gratis, el juego es 5x5 y paga en ambos sentidos",  
 TRIGGER\_6\_FREE\_SPINS: "activa 6 giros gratis",  
 FREE\_SPIN\_CAN\_RETRIGGER: "Los giros gratis se pueden reactivar",  
 GOOD\_LUCK: "Good Luck",  
 ADDITIONAL\_FREE\_SPINS\_AWARDED: "GIROS GRATIS ADICIONALES ACTIVADOS",  
 X\_FREE\_PLAY: "quedan %{remainingCount} jugadas gratis",  
 X\_FREE\_PLAY\_COMPLETED: "quedan jugadas gratis",  
 FREE\_SPIN\_AWARDED: "has ganado %{spinCount} jugadas gratis",  
 FREE\_SPIN\_OVER: "las jugadas gratis se han terminado",  
 NO\_THANKS: "no, gracias",  
 PLAY\_LATER: "jugar más tarde",  
 PLAY\_NOW: "jugar ahora"  
 },  
 PAYTABLE: {  
 PAYTABLE: "PAYTABLE",  
 PAYS\_BOTH\_WAYS: "Paga en ambos sentidos y amplía la matriz",  
 EXTENDS\_MATRIX: "Amplía la matriz",  
 ACTIVATES\_MIXED: "Activa los pagos combinados y amplía la matriz",  
 TRIGGER\_6\_FREE\_SPIN: "activa 6 giros gratis",  
 AND\_PAYS: "5x5 y paga en ambos sentidos",  
 WHEN\_THE\_RED: "Cuando la Luna Roja esté activa, todos los personajes cambian a su forma animal y comparten los premios (pagos combinados)",  
 MALFUNCTION\_TEXT: "LOS ERRORES EN EL FUNCIONAMIENTO ANULARÁN TODOS LOS PAGOS Y JUGADAS",  
 GAME\_RULES: "REGLAS DEL JUEGO",  
 GAME\_RULES1: "VOLSUNGA ES UNA TRAGAPERRAS CON FORMAS DE GANAR Y UNA MATRIZ AMPLIABLE DE 53135, 53335, 55355 Y 55555",  
 GAME\_RULES2: "LOS JUGADORES TIENEN 3125 FORMAS DE GANAR",  
 GAME\_RULES3: "EL JUEGO TIENE UN RETORNO AL JUGADOR (RTP) DEL 97,16 %",  
 GAME\_RULES4: "TODOS LOS PAGOS EXCEPTO LOS DE LOS SCATTER SE HACEN DE IZQUIERDA A DERECHA, EMPEZANDO POR EL CARRETE DE LA IZQUIERDA",  
 GAME\_RULES5: "LOS PAGOS DE LOS SCATTER SE PAGAN POR LOS SÍMBOLOS QUE APARECE EN CUALQUIER POSICIÓN DE LOS CARRETES",  
 GAME\_RULES6: "SOLO SE PAGA EL SÍMBOLO CON EL PREMIO MÁS ALTO POR CADA COMBINACIÓN GANADORA",  
 GAME\_RULES7: "SOLO SE PAGA EL SCATTER CON EL PREMIO MÁS ALTO POR JUGADA",  
 GAME\_RULES8: "LOS PAGOS DE LOS SCATTER SE AÑADEN A LOS PAGOS POR LÍNEA",  
 GAME\_RULES9: "LOS PAGOS POR FORMA SE MULTIPLICAN POR EL MÚLTIPLO DE LA APUESTA",  
 GAME\_RULES10: "LOS PAGOS SCATTER SE MULTIPLICAN POR LA APUESTA TOTAL DE CRÉDITOS",  
 GAME\_RULES11: "DURANTE LOS GIROS GRATIS, EL JUEGO ES 5X5 Y PAGA

```

EN AMBOS SENTIDOS",
    GAME_RULES12: "LOS GIROS GRATIS SE PUEDEN REACTIVAR"
},
BETSETTINGS: "OPCIONES DE APUESTAS",
AUTOPLAY: {
    AUTOPLAY_TITLE: "AUTOJUEGO",
    AUTOPLAY_SUBTITLE_1: "Seleccione número de giros",
    AUTOPLAY_SUBTITLE_2: "Parar autojuego",
    AUTOPLAY_DES_1: "Si la pérdida excede",
    AUTOPLAY_DES_2: "Si una sola ganancia excede",
    CLEAR: "Limpiar",
    PLAY: "Juego",
    WIN_LIMIT_EXCEED: "ganancia excede",
    LOSS_LIMIT_EXCEED: "pérdida excede"
},
SOUNDSETTINGS: {
    SOUND: "SONIDO",
    BACKGROUND_SOUND: "FONDO DE SONIDO"
},
MSG: {
    BIG_WIN: "BUEN PREMIO",
    HUGE_WIN: "¡¡GANANCIA EXTREMA",
    EPIC_WIN: "¡¡GANANCIA MASIVA",
    MEGA_WIN: "¡¡ENORME GANANCIA",
    CONGRATULATIONS: "FELICITACIONES",
    CREDITS_WON: "CREDITOS GANADOS",
    COMBINATION: "COMBINACION (ES)",
    AWARDED: "PREMIADO",
    FREE_SPINS: "FREE SPINS",
    ADDITIONAL_FREE_SPINS_AWARDED: "GIROS GRATIS ADICIONALES
ACTIVADOS",
    YOU_HAVE_WON: "HAS GANADO",
    FREE_SPINS_AWARDED: "SE RECOMPENSA CON GIROS GRATIS",
    REALITY_CHECK_LOSS_LIMIT: "Has superado el límite de pérdidas de la
comprobación",
    REALITY_CHECK_SESSION_LENGTH: "Has superado la duración de sesión de la
comprobación",
    CLICK_TO_START: "Pulsa para empezar",
    SESSION_TIME: "Tiempo de la sesión",
    WIN_AMOUNT: "Cantidad ganada",
    LOSS_AMOUNT: "Cantidad perdida",
    UNFINISHED_GAME_ROUND: "RONDAS DE JUEGO INCOMPLETAS"
},
BUTTON: {
    OK: "VALE",
    CANCEL: "CANCELAR",
    CONTINUE_PLAYING: "CONTINUAR JUGANDO",
    CLEAR: "BORRAR",
    CONTINUE: "CONTINUAR",
    AH: "Historial de la cuenta",
    RL: "VOLVER AL VESTÍBULO"
},

```

ERROR: {

101: "Boleto no válido",

404: "Error de comunicación",

501: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 501",

502: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 502",

503: "No se encuentra la sesión o esta ha caducado",

504: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 504",

509: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 509",

516: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 516",

517: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 517",

518: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 518",

522: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 522",

523: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 523",

541: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 541",

543: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 543",

551: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 551",

505: "El progreso del juego es inválido. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 505.",

507: "Apuesta inválida. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 507.",

508: "Error de comando del sistema. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 508.",

519: "Inicio de sesión inválido. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 519.",

520: "El juego está activo en otro navegador.",

525: "Error en la divisa. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 525.",

526: "Error en el inicio de sesión. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 526.",

537: "Error en el inicio de sesión. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 537.",

527: "Error en el giro gratis. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 527.",

532: "Error en el giro gratis. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 532.",

528: "URL inválida. ",

533: "Error en el token. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 533.",

536: "Error en el token. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 536.",

```

549: "Error en el token. Por favor, ponte en contacto con el equipo de atención al cliente
con el código de error 549.",
535: "La versión del juego es demasiado antigua. ",
538: "Error en el jackpot. Por favor, ponte en contacto con el equipo de atención al cliente
con el código de error 538",
539: "Error en el jackpot. Por favor, ponte en contacto con el equipo de atención al cliente
con el código de error 539",
540: "El jugador tiene una partida activa en otro servidor. ",
542: "Error en la versión. Por favor, ponte en contacto con el equipo de atención al cliente
con el código de error 542.",
544: "Se ha excedido el presupuesto de gasto",
547: "Error en la divisa. Por favor, ponte en contacto con el equipo de atención al cliente
con el código de error 547.",
550: "Límite de ganancias alcanzado",
506: "No hay fondos suficientes para la apuesta del tamaño seleccionado",
521: "Error técnico al iniciar sesión",
524: "Juego deshabilitado",
531: "No se permite jugar",
534: "El intento de inicio de sesión llega desde una región bloqueada",
545: "Has superado el límite de pérdidas de la comprobación",
546: "Has superado la duración de sesión de la comprobación",
548: "Ha caducado la sesión de comprobación de Playtech",
554: "Blocked from playing - daily time limit exceeded",
555: "Blocked from playing - weekly time limit exceeded",
556: "Blocked from playing - monthly time limit exceeded"
}
}, cc._RF.pop()
}, {}],
es: [function(e, t, i) {
"use strict";
cc._RF.push(t, "7b272owWthPorezcPnQt8Vvk", "es"), t.exports = {
WELCOME: "Welcome, %{name}!",
Click_anywhere_on_the_screen_to_start_the_game: "Haz click en la pantalla para empezar
el juego",
HUD: {
BET: "APUESTA",
CREDITS: "CREDITOS",
COIN_VALUE: "VALOR DE MONEDA",
WIN: "GANANCIA",
AUTOPLAY: "AUTOJUEGO",
MAXBET: "APUESTA MAXIMA",
"576WAYS_TO_WIN": "¡576 MANERAS DE GANANCIA!",
WIN_UP_TO_128_FREE_SPINS: "GANANCIA HASTA 128 GIROS GRATUITOS",
FREE: "GRATUITOS"
},
HUD_MESSAGES: {
EXTEND_MATRIX: "Amplía la matriz",
PAY_BOTH_WAYS_MATRIX: "Paga en ambos sentidos y amplía la matriz",
ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Activa los pagos
combinados y amplía la matriz",
IS_ACTIVE_ALL: " Cuando la Luna Roja esté activa, todos los personajes cambian a su
forma animal y comparten los premios (pagos combinados)",

```

"3X": "3x",  
DURING\_THE\_FREE\_SPIN: "Durante los giros gratis, el juego es 5x5 y paga en ambos sentidos",  
TRIGGER\_6\_FREE\_SPINS: "activa 6 giros gratis",  
FREE\_SPIN\_CAN\_RETRIGGER: "Los giros gratis se pueden reactivar",  
GOOD\_LUCK: "Good Luck",  
ADDITIONAL\_FREE\_SPINS\_AWARDED: "GIROS GRATIS ADICIONALES ACTIVADOS",  
X\_FREE\_PLAY: "quedan % {remainingCount} jugadas gratis",  
X\_FREE\_PLAY\_COMPLETED: "quedan jugadas gratis",  
FREE\_SPIN\_AWARDED: "has ganado % {spinCount} jugadas gratis",  
FREE\_SPIN\_OVER: "las jugadas gratis se han terminado",  
NO\_THANKS: "no, gracias",  
PLAY\_LATER: "jugar más tarde",  
PLAY\_NOW: "jugar ahora"  
},  
PAYTABLE: {  
PAYTABLE: "PAYTABLE",  
PAYS\_BOTH\_WAYS: "Paga en ambos sentidos y amplía la matriz",  
EXTENDS\_MATRIX: "Amplía la matriz",  
ACTIVATES\_MIXED: "Activa los pagos combinados y amplía la matriz",  
TRIGGER\_6\_FREE\_SPIN: "activa 6 giros gratis",  
AND\_PAYS: "5x5 y paga en ambos sentidos",  
WHEN\_THE\_RED: "Cuando la Luna Roja esté activa, todos los personajes cambian a su forma animal y comparten los premios (pagos combinados)",  
MALFUNCTION\_TEXT: "LOS ERRORES EN EL FUNCIONAMIENTO ANULARÁN TODOS LOS PAGOS Y JUGADAS",  
GAME\_RULES: "REGLAS DEL JUEGO",  
GAME\_RULES1: "VOLSUNGA ES UNA TRAGAPERRAS CON FORMAS DE GANAR Y UNA MATRIZ AMPLIABLE DE 53135, 53335, 55355 Y 55555",  
GAME\_RULES2: "LOS JUGADORES TIENEN 3125 FORMAS DE GANAR",  
GAME\_RULES3: "EL JUEGO TIENE UN RETORNO AL JUGADOR (RTP) DEL 97,16 %",  
GAME\_RULES4: "TODOS LOS PAGOS EXCEPTO LOS DE LOS SCATTER SE HACEN DE IZQUIERDA A DERECHA, EMPEZANDO POR EL CARRETE DE LA IZQUIERDA",  
GAME\_RULES5: "LOS PAGOS DE LOS SCATTER SE PAGAN POR LOS SÍMBOLOS QUE APARECE EN CUALQUIER POSICIÓN DE LOS CARRETES",  
GAME\_RULES6: "SOLO SE PAGA EL SÍMBOLO CON EL PREMIO MÁS ALTO POR CADA COMBINACIÓN GANADORA",  
GAME\_RULES7: "SOLO SE PAGA EL SCATTER CON EL PREMIO MÁS ALTO POR JUGADA",  
GAME\_RULES8: "LOS PAGOS DE LOS SCATTER SE AÑADEN A LOS PAGOS POR LÍNEA",  
GAME\_RULES9: "LOS PAGOS POR FORMA SE MULTIPLICAN POR EL MÚLTIPLO DE LA APUESTA",  
GAME\_RULES10: "LOS PAGOS SCATTER SE MULTIPLICAN POR LA APUESTA TOTAL DE CRÉDITOS",  
GAME\_RULES11: "DURANTE LOS GIROS GRATIS, EL JUEGO ES 5X5 Y PAGA EN AMBOS SENTIDOS",  
GAME\_RULES12: "LOS GIROS GRATIS SE PUEDEN REACTIVAR"  
},

```

BETSETTINGS: "OPCIONES DE APUESTAS",
AUTOPLAY: {
  AUTOPLAY_TITLE: "AUTOJUEGO",
  AUTOPLAY_SUBTITLE_1: "Seleccione número de giros",
  AUTOPLAY_SUBTITLE_2: "Parar autojuego",
  AUTOPLAY_DES_1: "Si la pérdida excede",
  AUTOPLAY_DES_2: "Si una sola ganancia excede",
  CLEAR: "Limpiar",
  PLAY: "Juego",
  WIN_LIMIT_EXCEED: "ganancia excede",
  LOSS_LIMIT_EXCEED: "pérdida excede"
},
SOUNDSETTINGS: {
  SOUND: "SONIDO",
  BACKGROUND_SOUND: "FONDO DE SONIDO"
},
MSG: {
  BIG_WIN: "BUEN PREMIO",
  HUGE_WIN: "¡¡GANANCIA EXTREMA",
  EPIC_WIN: "¡¡GANANCIA MASIVA",
  MEGA_WIN: "¡¡ENORME GANANCIA",
  CONGRATULATIONS: "FELICITACIONES",
  CREDITS_WON: "CREDITOS GANADOS",
  COMBINATION: "COMBINACION (ES)",
  AWARDED: "PREMIADO",
  FREE_SPINS: "FREE SPINS",
  ADDITIONAL_FREE_SPINS_AWARDED: "GIROS GRATIS ADICIONALES
ACTIVADOS",
  YOU_HAVE_WON: "HAS GANADO",
  FREE_SPINS_AWARDED: "SE RECOMPENSA CON GIROS GRATIS",
  REALITY_CHECK_LOSS_LIMIT: "Has superado el límite de pérdidas de la
comprobación",
  REALITY_CHECK_SESSION_LENGTH: "Has superado la duración de sesión de la
comprobación",
  CLICK_TO_START: "Pulsa para empezar",
  SESSION_TIME: "Tiempo de la sesión",
  WIN_AMOUNT: "Cantidad ganada",
  LOSS_AMOUNT: "Cantidad perdida",
  UNFINISHED_GAME_ROUND: "RONDAS DE JUEGO INCOMPLETAS"
},
BUTTON: {
  OK: "VALE",
  CANCEL: "CANCELAR",
  CONTINUE_PLAYING: "CONTINUAR JUGANDO",
  CLEAR: "BORRAR",
  CONTINUE: "CONTINUAR",
  AH: "Historial de la cuenta",
  RL: "VOLVER AL VESTÍBULO"
},
ERROR: {
  101: "Boleto no válido",
  404: "Error de comunicación",

```

501: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 501",  
502: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 502",  
503: "No se encuentra la sesión o esta ha caducado",  
504: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 504",  
509: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 509",  
516: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 516",  
517: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 517",  
518: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 518",  
522: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 522",  
523: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 523",  
541: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 541",  
543: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 543",  
551: "Error del sistema. Contacte con el soporte técnico y comuníquenos el código de error 551",  
505: "El progreso del juego es inválido. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 505.",  
507: "Apuesta inválida. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 507.",  
508: "Error de comando del sistema. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 508.",  
519: "Inicio de sesión inválido. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 519.",  
520: "El juego está activo en otro navegador.",  
525: "Error en la divisa. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 525.",  
526: "Error en el inicio de sesión. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 526.",  
537: "Error en el inicio de sesión. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 537.",  
527: "Error en el giro gratis. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 527.",  
532: "Error en el giro gratis. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 532.",  
528: "URL inválida. ",  
533: "Error en el token. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 533.",  
536: "Error en el token. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 536.",  
549: "Error en el token. Por favor, ponte en contacto con el equipo de atención al cliente con el código de error 549.",  
535: "La versión del juego es demasiado antigua. ",

```

538: "Error en el jackpot. Por favor, ponte en contacto con el equipo de atención al cliente
con el código de error 538",
539: "Error en el jackpot. Por favor, ponte en contacto con el equipo de atención al cliente
con el código de error 539",
540: "El jugador tiene una partida activa en otro servidor. ",
542: "Error en la versión. Por favor, ponte en contacto con el equipo de atención al cliente
con el código de error 542.",
544: "Se ha excedido el presupuesto de gasto",
547: "Error en la divisa. Por favor, ponte en contacto con el equipo de atención al cliente
con el código de error 547.",
550: "Límite de ganancias alcanzado",
506: "No hay fondos suficientes para la apuesta del tamaño seleccionado",
521: "Error técnico al iniciar sesión",
524: "Juego deshabilitado",
531: "No se permite jugar",
534: "El intento de inicio de sesión llega desde una región bloqueada",
545: "Has superado el límite de pérdidas de la comprobación",
546: "Has superado la duración de sesión de la comprobación",
548: "Ha caducado la sesión de comprobación de Playtech",
554: "Blocked from playing - daily time limit exceeded",
555: "Blocked from playing - weekly time limit exceeded",
556: "Blocked from playing - monthly time limit exceeded"
}
}, cc._RF.pop()
}, {}],
et: [function(e, t, i) {
"use strict";
cc._RF.push(t, "7f17aFoObFLibfH2RJZ2FWR", "et"), t.exports = {
WELCOME: "Welcome, %{name}!",
Click_anywhere_on_the_screen_to_start_the_game: "Vajuta ekraanile, et alustada mängu",
HUD: {
BET: "PANUS",
CREDITS: "RAHA",
COIN_VALUE: "MÜNDI VÄÄRTUS",
WIN: "VÕIT",
AUTOPLAY: "AUTOMAATMÄNG",
MAXBET: "MAX PANUS",
"576WAYS_TO_WIN": "576 VÕIDUVÕIMALUSTI!",
WIN_UP_TO_128_FREE_SPINS: "VÕIDA KUNI 128 TASUTA KEERUTUST",
FREE: "TASUTA"
},
HUD_MESSAGES: {
EXTEND_MATRIX: "Pikendab maatriksit",
PAY_BOTH_WAYS_MATRIX: "Võidud mõlemas suunas ja pikendab maatriksit",
ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Aktiveerib segavõidud ja
pikendab maatriksit",
IS_ACTIVE_ALL: " Verekuu ajal võtavad kõik tegelased oma loomse kuju ja jagavad
võite (segavõidud)",
"3X": "3x",
DURING_THE_FREE_SPIN: "Tasuta keerutuste ajal on mäng alati 5x5 ja võidab
mõlemas suunas",
TRIGGER_6_FREE_SPINS: "käivitab 6 tasuta keerutust",

```

```

FREE_SPIN_CAN_RETRIGGER: "Ilmaiskierrokset voivat käynnistyä uudestaan",
GOOD_LUCK: "Good Luck",
X_FREE_PLAY: "järel on %{remainingCount} tasuta mängu",
X_FREE_PLAY_COMPLETED: "järel on tasuta mängu",
FREE_SPIN_AWARDED: "olete võitnud %{spinCount} tasuta mängu",
FREE_SPIN_OVER: "tasuta mängud läbi",
NO_THANKS: "tänan, ei",
PLAY_LATER: "mängi hiljem",
PLAY_NOW: "mängi kohe"
},
PAYTABLE: {
  PAYTABLE: "VÕIDUTABEL",
  PAYS_BOTH_WAYS: "Võidud mõlemas suunas ja pikendab maatriksit",
  EXTENDS_MATRIX: "Pikendab maatriksit",
  ACTIVATES_MIXED: "Aktiveerib segavõidud ja pikendab maatriksit",
  TRIGGER_6_FREE_SPIN: "trigger 6 free spins",
  AND_PAYS: "5x5 ja võidab mõlemas suunas",
  WHEN_THE_RED: "Verekuu ajal võtavad kõik tegelased oma loomse kuju ja jagavad
võite (segavõidud)",
  MALFUNCTION_TEXT: "TEHNILINE TÕRGE TÜHISTAB KÕIK VÕIDUD JA
MÄNGUD",
  GAME_RULES: "MÄNGU REEGLID",
  GAME_RULES1: "VOLSUNGA ON ERI VÕIDUVÕIMALUSTEGA (WAYS) MÄNG
PIKENEVA 53135, 53335, 55355 JA 55555 MAATRIKSIGA",
  GAME_RULES2: "MÄNGIJATEL ON 3125 VÕIDUVÕIMALUST",
  GAME_RULES3: "MÄNGU RTP ON 97,16%.",
  GAME_RULES4: "KÕIKI VÕITE, V.A SCATTERI VÕIDUD, ARVESTATAKSE
VASAKPOOLSEIMAST RULLIKUST VASAKULT PAREMALE",
  GAME_RULES5: "SCATTERI VÕIDUD MAKSTAKSE SÜMBOLITE EEST, MIS
ILMUVAD RULLIKUTELE MIS TAHES POSITSIOONIS",
  GAME_RULES6: "VÕIDUKOMBINATSIOONI KOHTA MAKSTAKSE VÄLJA
AINULT KÕRGEIMA SÜMBOLI VÕIT",
  GAME_RULES7: "MÄNGU KOHTA MAKSTAKSE VÄLJA AINULT KÕRGEIM
SCATTERI VÕIT",
  GAME_RULES8: "SCATTERI VÕIDUD MAKSTAKSE LISAKS
VÕIDUVÕIMALUSTE VÕITUDELE",
  GAME_RULES9: "VÕIDUVÕIMALUSTE VÕIDUD KORRUTATAKSE PANUSE
KORISTAJAGA",
  GAME_RULES10: "SCATTERI VÕIDUD KORRUTATAKSE KREDIIDI
KOGUPANUSEGA",
  GAME_RULES11: "TASUTA KEERUTUSTE AJAL ON MÄNG ALATI 5X5 JA
VÕITA SAAB ALATI MÕLEMAT VIISI",
  GAME_RULES12: "Tasuta keerutusi saab taaskäivitada"
},
BETSETTINGS: "PANUSE SEADISTUSED",
AUTOPLAY: {
  AUTOPLAY_TITLE: "AUTOMAATMÄNG",
  AUTOPLAY_SUBTITLE_1: "Vali keerutuste arv",
  AUTOPLAY_SUBTITLE_2: "Peata automaatmäng",
  AUTOPLAY_DES_1: "Kui kaotus ületab",
  AUTOPLAY_DES_2: "Kui üks võit ületab",
  CLEAR: "Puhasta",

```

```

PLAY: "Mängi",
WIN_LIMIT_EXCEED: "võit ületab",
LOSS_LIMIT_EXCEED: "kaotus ületab"
},
SOUNDSETTINGS: {
    SOUND: "HELI",
    BACKGROUND_SOUND: "TAUSTAHELI"
},
MSG: {
    BIG_WIN: "HEA VÕIT",
    HUGE_WIN: "EKSTREEMNE VÕIT",
    EPIC_WIN: "MASSIIVNE VÕIT",
    MEGA_WIN: "TOHUTU VÕIT",
    CONGRATULATIONS: "PALJU ÕNNE",
    CREDITS_WON: "VÕIDETUD RAHA",
    COMBINATION: "KOMBINATSIOON(ID)",
    AWARDED: "TASUSTATUD",
    FREE_SPINS: "TASUTA KEERUTUSED",
    ADDITIONAL_FREE_SPINS_AWARDED: "VÕITSID TÄIENDAVAID TASUTA
KEERUTUSI",
    YOU_HAVE_WON: "OLED VÕITNUD",
    FREE_SPINS_AWARDED: "SAADUD TASUTA KEERUTUSI",
    REALITY_CHECK_LOSS_LIMIT: "Reaalsuskontrolli kaotuste piirmäär ületatud",
    REALITY_CHECK_SESSION_LENGTH: "Reaalsuskontrolli pikkus ületatud",
    CLICK_TO_START: "Alustamiseks klõpsake",
    SESSION_TIME: "Seansi aeg",
    WIN_AMOUNT: "Võidu summa",
    LOSS_AMOUNT: "Kaotuse summa",
    UNFINISHED_GAME_ROUND: "LÕPETAMATA MÄNGUVOORUD"
},
BUTTON: {
    OK: "OK",
    CANCEL: "TÜHISTA",
    CONTINUE_PLAYING: "JÄTKA MÄNGIMIST",
    CLEAR: "TÜHJENDA",
    CONTINUE: "JÄTKA",
    AH: "Konto ajalugu",
    RL: "TAGASI AVALEHELE"
},
ERROR: {
    101: "Kehtetu pilet",
    404: "side tõrge",
    501: "Süsteemiviga. Teatage kasutajatoele veakood 501.",
    502: "Süsteemiviga. Teatage kasutajatoele veakood 502.",
    503: "Seanss puudub või aegunud",
    504: "Süsteemiviga. Teatage kasutajatoele veakood 504.",
    509: "Süsteemiviga. Teatage kasutajatoele veakood 509.",
    516: "Süsteemiviga. Teatage kasutajatoele veakood 516.",
    517: "Süsteemiviga. Teatage kasutajatoele veakood 517.",
    518: "Süsteemiviga. Teatage kasutajatoele veakood 518.",
    522: "Süsteemiviga. Teatage kasutajatoele veakood 522.",
    523: "Süsteemiviga. Teatage kasutajatoele veakood 523.",

```

541: "Süsteemiviga. Teatage kasutajatoele veakood 541.",  
 543: "Süsteemiviga. Teatage kasutajatoele veakood 543.",  
 551: "Süsteemiviga. Teatage kasutajatoele veakood 551.",  
 505: "Kehtetu mänguprogess. Võtke ühendust klienditoega ja teavitage veakoodist 505.",  
 507: "Kehtetu panus. Võtke ühendust klienditoega ja teavitage veakoodist 507.",  
 508: "Süsteemikäskudega seotud viga. Võtke ühendust klienditoega ja teavitage  
 veakoodist 508.",  
 519: "Kehtetu sisselogimine. Võtke ühendust klienditoega ja teavitage veakoodist 519.",  
 520: "Mäng on aktiivne teises brauseris.",  
 525: "Valuutaga seotud viga. Võtke ühendust klienditoega ja teavitage veakoodist 525.",  
 526: "Sisselogimisviga. Võtke ühendust klienditoega ja teavitage veakoodist 526.",  
 537: "Sisselogimisviga. Võtke ühendust klienditoega ja teavitage veakoodist 526.",  
 527: "Tasuta keerutusega seotud viga. Võtke ühendust klienditoega ja teavitage  
 veakoodist 527.",  
 532: "Tasuta keerutusega seotud viga. Võtke ühendust klienditoega ja teavitage  
 veakoodist 527.",  
 528: "Kehtetu URL.",  
 533: "Tokeniga seotud viga. Võtke ühendust klienditoega ja teavitage veakoodist 533.",  
 536: "Tokeniga seotud viga. Võtke ühendust klienditoega ja teavitage veakoodist 533.",  
 549: "Tokeniga seotud viga. Võtke ühendust klienditoega ja teavitage veakoodist 533.",  
 535: "Mänguversioon on liiga vana.",  
 538: "Jackpotiga seotud viga. Võtke ühendust klienditoega ja teavitage veakoodist 538.",  
 539: "Jackpotiga seotud viga. Võtke ühendust klienditoega ja teavitage veakoodist 538.",  
 540: "Mängijal on aktiivne mäng teises serveris.",  
 542: "Versiooniga seotud viga. Võtke ühendust klienditoega ja teavitage veakoodist  
 542.",  
 544: "Kulueelarve ületatud",  
 547: "Valuutaga seotud viga. Võtke ühendust klienditoega ja teavitage veakoodist 547.",  
 550: "Võitude ülemmäär saavutatud",  
 506: "Valitud panuse suuruse jaoks ei ole piisavalt raha",  
 521: "Tehniline tõrge sisselogimisel",  
 524: "Mäng keelatud",  
 531: "Mängimine blokitud",  
 534: "Sisse logida üritati blokitud piirkonnast",  
 545: "Reaalsuskontrolli kaotuste piirmäär ületatud",  
 546: "Reaalsuskontrolli pikkus ületatud",  
 548: "Playtechi reaalsuskontrolli seanss aegunud",  
 554: "Blocked from playing - daily time limit exceeded",  
 555: "Blocked from playing - weekly time limit exceeded",  
 556: "Blocked from playing - monthly time limit exceeded"

```

    }
  }, cc._RF.pop()
}, {}],
fi: [function(e, t, i) {
  "use strict";
  cc._RF.push(t, "9499eeGt0pCs4fola4jtf9t", "fi"), t.exports = {
    WELCOME: "Welcome, %{name}!",
    Click_anywhere_on_the_screen_to_start_the_game: "Klikkaa mihin tahansa näytöllä  

aloittaaksesi pelin",
    HUD: {
      BET: "PANOS",
      CREDITS: "PELIMERKIT",

```

```

COIN_VALUE: "KOLIKKOARVO",
WIN: "VOITTO",
AUTOPLAY: "AUTOMAATTIPELI",
MAXBET: "MAKSIMIPANOS",
"576WAYS_TO_WIN": "576 TAPAA VOITTA!",
WIN_UP_TO_128_FREE_SPINS: "VOITA JOPA 128 VAPAA PYÖRÄYTYSTÄ",
FREE: "VAPAA"
},
HUD_MESSAGES: {
  EXTEND_MATRIX: "Jatkaa matriisia",
  PAY_BOTH_WAYS_MATRIX: "Maksaa molempiin suuntiin ja pidentää matriisia",
  ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Aktivoi sekoitetut voitot ja
pidentää matriisia",
  IS_ACTIVE_ALL: " Kun Punainen kuu on aktiivinen, kaikki hahmot muuttavat
muotoaan eläinmuotoonsa ja jakavat voitot (sekoitetut voitot)",
  "3X": "3x",
  DURING_THE_FREE_SPIN: "Ilmaiskierrosten aikana peli on aina 5x5 ja voitot
maksetaan aina molempiin suuntiin",
  TRIGGER_6_FREE_SPINS: "trigger 6 free spins is active.",
  FREE_SPIN_CAN_RETRIGGER: "Ilmaiskierrokset voivat käynnistyä uudestaan",
  GOOD_LUCK: "Good Luck",
  X_FREE_PLAY: "jäljellä %{remainingCount} ilmaispeliä",
  X_FREE_PLAY_COMPLETED: "jäljellä ilmaispeliä",
  FREE_SPIN_AWARDED: "olet saanut %{spinCount} ilmaispeliä",
  FREE_SPIN_OVER: "ilmaispelit loppuivat",
  NO_THANKS: "ei, kiitos",
  PLAY_LATER: "pelaa myöhemmin",
  PLAY_NOW: "pelaa nyt"
},
PAYTABLE: {
  PAYTABLE: "VOITTOTAULUKKO",
  PAYS_BOTH_WAYS: "Maksaa molempiin suuntiin ja pidentää matriisia",
  EXTENDS_MATRIX: "Jatkaa matriisia",
  ACTIVATES_MIXED: "Aktivoi sekoitetut voitot ja pidentää matriisia",
  TRIGGER_6_FREE_SPIN: "trigger 6 free spins",
  AND_PAYS: "5x5 ja voitot maksetaan molempiin suuntiin",
  WHEN_THE_RED: "Kun Punainen kuu on aktiivinen, kaikki hahmot muuttavat
muotoaan eläinmuotoonsa ja jakavat voitot (sekoitetut voitot)",
  MALFUNCTION_TEXT: "TOIMINTAHÄIRIÖ MITÄTÖI KAIKKI VOITOT JA
PELIKIERROKSET",
  GAME_RULES: "PELIN SÄÄNNÖT",
  GAME_RULES1: "VOLSUNGA ON VOITTOTAPAPELI, JOSSA ON JATKUVAT
53135, 53335, 55355 JA 55555 MATRIISIT",
  GAME_RULES2: "PELAAJILLA ON KÄYTÖSSÄÄN 3 125 VOITTOTAPAA",
  GAME_RULES3: "PELIN TEOREETTINEN PALAUTUSPROSENTTI ON 97,16 %",
  GAME_RULES4: "KAIKKI PAITSI SCATTER-VOITOT MAKSETAAN
VASEMMALTA OIKEALLE VASEMMANPUOLIMMAISESTA RULLASTA ALKAEN",
  GAME_RULES5: "SCATTER-VOITOT MAKSETAAN MISSÄ TAHANSA
KOHDISSA OLEVISTA SYMBOLEISTA",
  GAME_RULES6: "VAIN SUURIN VOITTO MAKSETAAN
VOITTOYHDISTELMÄÄ KOHDEN",
  GAME_RULES7: "VAIN SUURIN SCATTER-VOITTO MAKSETAAN

```

```

PELIKIERROSTA KOHDEN",
    GAME_RULES8: "SCATTER-VOITOT MAKSETAAN VOITTOTAPAVOITTOJEN
LISÄKSI",
    GAME_RULES9: "VOITTOTAPAVOITOT KERROTAAN PANOSKERTOIMELLA",
    GAME_RULES10: "SCATTER-VOITOT MAKSETAAN KOKONAISPANOKSEN
KERRANNAISINA",
    GAME_RULES11: "ILMAISKIERROSTEN AIKANA PELISSÄ ON AINA 5X5-
RUUDUKKO JA VOITOT MAKSETAAN MOLEMPIIN SUUNTIIN",
    GAME_RULES12: "ILMAISKIERROKSET VOI KÄYNNISTÄÄ UUDESTAAN"
},
BETSETTINGS: "PANOS ASETUKSET",
AUTOPLAY: {
    AUTOPLAY_TITLE: "AUTOMAATTIPELI",
    AUTOPLAY_SUBTITLE_1: "Valitse pyöräytysten määrä",
    AUTOPLAY_SUBTITLE_2: "Pysäytä automaattipeli",
    AUTOPLAY_DES_1: "jos tappio ylittää",
    AUTOPLAY_DES_2: "Jos yksittäinen voitto ylittää",
    CLEAR: "Tyhjennä",
    PLAY: "Pelaa",
    WIN_LIMIT_EXCEED: "voitto ylittää",
    LOSS_LIMIT_EXCEED: "tappio ylittää"
},
SOUNDSETTINGS: {
    SOUND: "ÄÄNI",
    BACKGROUND_SOUND: "TAUSTAÄÄNET"
},
MSG: {
    BIG_WIN: "UPEA VOITTO",
    HUGE_WIN: "ÄÄRIMMÄINEN VOITTO",
    EPIC_WIN: "MASSIIVINEN VOITTO",
    MEGA_WIN: "VALTAVA VOITTO",
    CONGRATULATIONS: "ONNITTELUT",
    CREDITS_WON: "PELIMERKKEJÄ VOITETTU",
    COMBINATION: "YHDISTELMÄ(T)",
    AWARDED: "ANSAITTU",
    FREE_SPINS: "ILMAISKIERROKSET",
    ADDITIONAL_FREE_SPINS_AWARDED: "LISÄILMAISKIERROKSIA
ANNETTU",
    YOU_HAVE_WON: "OLET VOITTANUT",
    FREE_SPINS_AWARDED: "ILMAISKIERROKSIA ANNETTU",
    REALITY_CHECK_LOSS_LIMIT: "Pelimuistutuksen tappioraja ylitetty",
    REALITY_CHECK_SESSION_LENGTH: "Pelimuistutuksen istunnon pituus ylitetty",
    CLICK_TO_START: "Klikkaa aloittaaksesi",
    SESSION_TIME: "Istunnon kesto",
    WIN_AMOUNT: "Voittosumma",
    LOSS_AMOUNT: "Tappiosumma",
    UNFINISHED_GAME_ROUND: "KESKENERÄISIÄ PELIKIERROKSIA"
},
BUTTON: {
    OK: "OK",
    CANCEL: "PERUUTA",
    CONTINUE_PLAYING: "JATKA PELAAMISTA",

```

```
CLEAR: "TYHJENNÄ",
CONTINUE: "JATKA",
AH: "Tilihistoria",
RL: "PALAA AULAAN"
},
ERROR: {
  101: "Virheellinen lippu",
  404: "yhteysvirhe",
  501: "Järjestelmävirhe. Ota yhteyttä asiakaspalveluun ja kerro virhekoodi 501.",
  502: "Järjestelmävirhe. Ota yhteyttä asiakaspalveluun ja kerro virhekoodi 502.",
  503: "Istunto puuttuu tai on vanhentunut",
  504: "Järjestelmävirhe. Ota yhteyttä asiakaspalveluun ja kerro virhekoodi 504.",
  509: "Järjestelmävirhe. Ota yhteyttä asiakaspalveluun ja kerro virhekoodi 509.",
  516: "Järjestelmävirhe. Ota yhteyttä asiakaspalveluun ja kerro virhekoodi 516.",
  517: "Järjestelmävirhe. Ota yhteyttä asiakaspalveluun ja kerro virhekoodi 517.",
  518: "Järjestelmävirhe. Ota yhteyttä asiakaspalveluun ja kerro virhekoodi 518.",
  522: "Järjestelmävirhe. Ota yhteyttä asiakaspalveluun ja kerro virhekoodi 522.",
  523: "Järjestelmävirhe. Ota yhteyttä asiakaspalveluun ja kerro virhekoodi 523.",
  541: "Järjestelmävirhe. Ota yhteyttä asiakaspalveluun ja kerro virhekoodi 541.",
  543: "Järjestelmävirhe. Ota yhteyttä asiakaspalveluun ja kerro virhekoodi 543.",
  551: "Järjestelmävirhe. Ota yhteyttä asiakaspalveluun ja kerro virhekoodi 551.",
  505: "Virheellinen pelin edistyminen. Ota yhteyttä asiakaspalveluun virhekoodilla 505.",
  507: "Virheellinen panostus. Ota yhteyttä asiakaspalveluun virhekoodilla 507.",
  508: "Käskyvirhe järjestelmässä. Ota yhteyttä asiakaspalveluun virhekoodilla 508.",
  519: "Virheelliset kirjautumistiedot. Ota yhteyttä asiakaspalveluun virhekoodilla 519.",
  520: "Peli on aktiivinen toisessa selaimessa.",
  525: "Valuuttavirhe. Ota yhteyttä asiakaspalveluun virhekoodilla 525.",
  526: "Kirjautumisvirhe. Ota yhteyttä asiakaspalveluun virhekoodilla 526.",
  537: "Kirjautumisvirhe. Ota yhteyttä asiakaspalveluun virhekoodilla 537.",
  527: "Ilmaiskierrosvirhe. Ota yhteyttä asiakaspalveluun virhekoodilla 527.",
  532: "Ilmaiskierrosvirhe. Ota yhteyttä asiakaspalveluun virhekoodilla 532.",
  528: "Virheellinen URL.",
  533: "Polettivirhe. Ota yhteyttä asiakaspalveluun virhekoodilla 533.",
  536: "Polettivirhe. Ota yhteyttä asiakaspalveluun virhekoodilla 536.",
  549: "Polettivirhe. Ota yhteyttä asiakaspalveluun virhekoodilla 549.",
  535: "Pelin versio on liian vanha. ",
  538: "Jackpot-virhe. Ota yhteyttä asiakaspalveluun virhekoodilla 538.",
  539: "Jackpot-virhe. Ota yhteyttä asiakaspalveluun virhekoodilla 539.",
  540: "Pelaajalla on aktiivinen peli toisella palvelimella.",
  542: "Versiovirhe. Ota yhteyttä asiakaspalveluun virhekoodilla 542.",
  544: "Kulutusbudjetti ylitetty",
  547: "Valuuttavirhe. Ota yhteyttä asiakaspalveluun virhekoodilla 547.",
  550: "Voittoraja saavutettu",
  506: "Varat eivät riitä valittuun panokseen",
  521: "Tekninen virhe kirjautumisen aikana",
  524: "Peli poistettu käytöstä",
  531: "Estetty pelaamasta",
  534: "Kirjautumisyritys estetystä sijainnista",
  545: "Pelimuistutuksen tappioraja ylitetty",
  546: "Pelimuistutuksen istunnon pituus ylitetty",
  548: "Playtech-pelimuistutuksen istunto vanhentunut",
  554: "Blocked from playing - daily time limit exceeded",
```

```

555: "Blocked from playing - weekly time limit exceeded",
556: "Blocked from playing - monthly time limit exceeded"
}
}, cc._RF.pop()
}, {}],
fr: [function(e, t, i) {
"use strict";
cc._RF.push(t, "1de86yTLKtCfbzsevUzlKB5", "fr"), t.exports = {
WELCOME: "Welcome, %{name}!",
Click_anywhere_on_the_screen_to_start_the_game: "Cliquez n'importe où sur l'écran pour
commencer à jouer",
HUD: {
BET: "MISE",
CREDITS: "CRÉDITS",
COIN_VALUE: "VALEUR DES PIÈCES",
WIN: "GAIN",
AUTOPLAY: "JEU AUTO",
MAXBET: "MISE MAX",
"576WAYS_TO_WIN": "576 MANIÈRES DE GAGNER !",
WIN_UP_TO_128_FREE_SPINS: "GAGNEZ JUSQU'À 128 TOURS GRATUITS",
FREE: "GRATUITS"
},
HUD_MESSAGES: {
EXTEND_MATRIX: "Étend la matrice!!",
PAY_BOTH_WAYS_MATRIX: "Paie dans les deux sens et étend la matrice!!",
ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Active les paiements
mixtes et étend la matrice!!",
IS_ACTIVE_ALL: "Quand la Lune rouge est active, tous les personnages se
métamorphosent leur animal et partagent les gains (paiements mixtes)!!",
"3X": "3x",
DURING_THE_FREE_SPIN: "Pendant les tours gratuits, le jeu est toujours en 5x5 et
paie toujours dans les deux sens",
TRIGGER_6_FREE_SPINS: "déclenche 6 tours gratuits!!",
FREE_SPIN_CAN_RETRIGGER: "Les tours gratuits peuvent être redéclenchés",
GOOD_LUCK: "Good Luck!!",
X_FREE_PLAY: "%{remainingCount} tours gratuits restants",
X_FREE_PLAY_COMPLETED: "tours gratuits restants",
FREE_SPIN_AWARDED: "il vous a été attribué %{spinCount} tours gratuits",
FREE_SPIN_OVER: "tours gratuits terminés",
NO_THANKS: "non merci",
PLAY_LATER: "jouer plus tard",
PLAY_NOW: "jouer maintenant"
},
PAYTABLE: {
PAYTABLE: "TABLE DES PAIEMENTS",
PAYS_BOTH_WAYS: "Paie dans les deux sens et étend la matrice",
EXTENDS_MATRIX: "Étend la matrice",
ACTIVATES_MIXED: "Active les paiements mixtes et étend la matrice",
TRIGGER_6_FREE_SPIN: "déclenche 6 tours gratuits",
AND_PAYS: "5x5 et paie dans les deux sens",
WHEN_THE_RED: "Quand la Lune rouge est active, tous les personnages se
métamorphosent leur animal et partagent les gains (paiements mixtes)",

```

```

MALFUNCTION_TEXT: "UNE MAUVAISE UTILISATION OU UN
DYSFONCTIONNEMENT ANNULENT TOUTES LES ACTIONS ET TOUS LES
PAIEMENTS",
    GAME_RULES: "RÈGLES DU JEU",
    GAME_RULES1: "VOLSUNGA FONCTIONNE SUR LE PRINCIPE DES WAYS
(FAÇONS DE GAGNER), AVEC UNE MATRICE EXTENSIBLE À 53135, 53335, 55355, ET
55555.",
    GAME_RULES2: "LES JOUEURS DISPOSENT DE 3 125 FAÇONS DE GAGNER",
    GAME_RULES3: "LE JEU A UN RTP DE 97,16 %",
    GAME_RULES4: "TOUS LES GAINS, À L'EXCEPTION DE CEUX DES SCATTERS,
SONT PAYÉS DE GAUCHE À DROITE EN COMMENÇANT PAR LE ROULEAU LE PLUS À
GAUCHE.",
    GAME_RULES5: "LES GAINS DES SCATTERS SONT PAYÉS POUR
L'APPARITION DES SYMBOLES N'IMPORTE OÙ SUR LES ROULEAUX",
    GAME_RULES6: "SEUL LE GAIN DU SYMBOLE LE PLUS ÉLEVÉ EST PAYÉ PAR
COMBINAISON GAGNANTE",
    GAME_RULES7: "SEUL LE GAIN DE SCATTER LE PLUS ÉLEVÉ EST PAYÉ PAR
TOUR",
    GAME_RULES8: "LES GAINS DES SCATTERS S'AJOUTENT AUX GAINS DES
WAYS",
    GAME_RULES9: "LES GAINS DES WAYS SONT MULTIPLIÉS PAR LE
MULTIPLE DE LA MISE",
    GAME_RULES10: "LES GAINS DES SCATTERS SONT MULTIPLIÉS PAR LA
MISE TOTALE",
    GAME_RULES11: "PENDANT LES TOURS GRATUITS, LE JEU RESTE EN 5X5 ET
PAIE TOUJOURS DANS LES DEUX SENS",
    GAME_RULES12: "Les tours gratuits peuvent être redéclenchés"
},
BETSETTINGS: "PARAMÈTRES DE LA MISE",
AUTOPLAY: {
    AUTOPLAY_TITLE: "JEU AUTO",
    AUTOPLAY_SUBTITLE_1: "Sélectionnez un nombre de tours",
    AUTOPLAY_SUBTITLE_2: "Arrêter le jeu auto",
    AUTOPLAY_DES_1: "Si la perte est supérieure à",
    AUTOPLAY_DES_2: "Si un gain à lui seul est supérieur à",
    CLEAR: "Vider",
    PLAY: "Jouer",
    WIN_LIMIT_EXCEED: " gain à lui seul est supérieur",
    LOSS_LIMIT_EXCEED: "perte est supérieure"
},
SOUNDSETTINGS: {
    SOUND: "SON",
    BACKGROUND_SOUND: "MUSIQUE DE FOND"
},
MSG: {
    BIG_WIN: "BIEN JOUÉ !",
    HUGE_WIN: "GAIN EXTRÊME",
    EPIC_WIN: "GAIN MASSIF",
    MEGA_WIN: "ÉNORME GAIN ",
    CONGRATULATIONS: "FÉLICITATIONS",
    CREDITS_WON: "CRÉDITS REMPORÉS",
    COMBINATION: "COMBINAISON(S)",

```

```

    AWARDED: "OBTENUS",
    FREE_SPINS: "TOURS GRATUITS",
    ADDITIONAL_FREE_SPINS_AWARDED: "TOURS GRATUITS
SUPPLÉMENTAIRES ATTRIBUÉS",
    YOU_HAVE_WON: "VOUS AVEZ GAGNÉ",
    FREE_SPINS_AWARDED: "TOURS GRATUITS ATTRIBUÉS",
    REALITY_CHECK_LOSS_LIMIT: "Limite de perte du rappel à la réalité dépassée",
    REALITY_CHECK_SESSION_LENGTH: "Durée de session du rappel à la réalité
dépassée",
    CLICK_TO_START: "Cliquez pour commencer",
    SESSION_TIME: "Durée de session",
    WIN_AMOUNT: "Montant gagné",
    LOSS_AMOUNT: "Montant perdu",
    UNFINISHED_GAME_ROUND: "PARTIES INACHEVÉES"
},
BUTTON: {
    OK: "OK",
    CANCEL: "ANNULER",
    CONTINUE_PLAYING: "CONTINUER À JOUER",
    CLEAR: "EFFACER",
    CONTINUE: "CONTINUER",
    AH: "Historique du compte",
    RL: "RETOURNER AU SALON"
},
ERROR: {
    404: "Erreur de communication",
501",    501: "Erreur système. Veuillez contacter l'assistance technique et indiquer le code d'erreur
502",    502: "Erreur système. Veuillez contacter l'assistance technique et indiquer le code d'erreur
504",    503: "Session manquante ou expirée",
509",    504: "Erreur système. Veuillez contacter l'assistance technique et indiquer le code d'erreur
516",    509: "Erreur système. Veuillez contacter l'assistance technique et indiquer le code d'erreur
517",    516: "Erreur système. Veuillez contacter l'assistance technique et indiquer le code d'erreur
518",    517: "Erreur système. Veuillez contacter l'assistance technique et indiquer le code d'erreur
522",    518: "Erreur système. Veuillez contacter l'assistance technique et indiquer le code d'erreur
523",    522: "Erreur système. Veuillez contacter l'assistance technique et indiquer le code d'erreur
541",    523: "Erreur système. Veuillez contacter l'assistance technique et indiquer le code d'erreur
543",    541: "Erreur système. Veuillez contacter l'assistance technique et indiquer le code d'erreur
551",    543: "Erreur système. Veuillez contacter l'assistance technique et indiquer le code d'erreur
551",    551: "Erreur système. Veuillez contacter l'assistance technique et indiquer le code d'erreur
    505: "Progression du jeu non valide. Veuillez contacter l'assistance technique avec le

```

```

code d'erreur 505.",
    507: "Mise non valide. Veuillez contacter l'assistance technique avec le code d'erreur
507.",
    508: "Erreur de commande système. Veuillez contacter l'assistance technique avec le
code d'erreur 508.",
    519: "Identifiant non valide. Veuillez contacter l'assistance technique avec le code
d'erreur 519.",
    520: "Le jeu est actif dans un autre navigateur.",
    525: "Erreur de devise. Veuillez contacter l'assistance technique avec le code d'erreur
525.",
    526: "Erreur de connexion. Veuillez contacter l'assistance technique avec le code d'erreur
526.",
    537: "Erreur de connexion. Veuillez contacter l'assistance technique avec le code d'erreur
526.",
    527: "Erreur de tour gratuit. Veuillez contacter l'assistance technique avec le code
d'erreur 527.",
    532: "Erreur de tour gratuit. Veuillez contacter l'assistance technique avec le code
d'erreur 527.",
    528: "URL non valide.",
    533: "Erreur de jeton. Veuillez contacter l'assistance technique avec le code d'erreur
533.",
    536: "Erreur de jeton. Veuillez contacter l'assistance technique avec le code d'erreur
536.",
    549: "Erreur de jeton. Veuillez contacter l'assistance technique avec le code d'erreur
549.",
    535: "Version du jeu trop ancienne",
    538: "Erreur de jackpot. Veuillez contacter l'assistance technique avec le code d'erreur
538.",
    539: "Erreur de jackpot. Veuillez contacter l'assistance technique avec le code d'erreur
539.",
    540: "Le joueur a une session active du jeu sur un autre serveur.",
    542: "Erreur de version. Veuillez contacter l'assistance technique avec le code d'erreur
542.",
    544: "Budget de dépenses dépassé",
    547: "Erreur de devise. Veuillez contacter l'assistance technique avec le code d'erreur
547.",
    550: "Plafond de gain atteint",
    506: "Fonds insuffisants pour la taille de mise choisie",
    521: "Erreur technique pendant la connexion",
    524: "Jeu désactivé",
    531: "Jeu bloqué",
    534: "La tentative de connexion vient d'une région bloquée",
    545: "Limite de perte du rappel à la réalité dépassée",
    546: "Durée de session du rappel à la réalité dépassée",
    548: "Session de rappel à la réalité de Playtech expirée",
    554: "Blocked from playing - daily time limit exceeded",
    555: "Blocked from playing - weekly time limit exceeded",
    556: "Blocked from playing - monthly time limit exceeded"
}
}, cc._RF.pop()
}, {}],
hu: [function(e, t, i) {

```

```

"use strict";
cc._RF.push(t, "1bf4eJsrgZN343KzcsriHx2", "hu"), t.exports = {
  WELCOME: "Welcome, %{name}!",
  Click_anywhere_on_the_screen_to_start_the_game: "Kattintson valahova a képernyőn,
  hogy elkezdhesse a játékot!",
  HUD: {
    BET: "TÉT",
    CREDITS: "KREDITEK",
    COIN_VALUE: "ÉRME ÉRTÉKE",
    WIN: "NYEREMÉNY",
    AUTOPLAY: "AUTOMATIKUS JÁTÉK",
    MAXBET: "MAXIMÁLIS TÉT",
    "576WAYS_TO_WIN": "576 NYERÉSI ESÉLY!",
    WIN_UP_TO_128_FREE_SPINS: "AKÁR 128 INGYEN PÖRGETÉS",
    FREE: "INGYEN"
  },
  HUD_MESSAGES: {
    EXTEND_MATRIX: "Kiterjeszti a mátrixot",
    PAY_BOTH_WAYS_MATRIX: "Mindkét irányban fizet és kiterjeszti a mátrixot",
    ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Activates mixed pays and
  extends matrix",
    IS_ACTIVE_ALL: " Amikor a Piros hold aktív, minden karakter állati formára vált, és
  rész vesznek a nyerő kombinációkban (vegyes kifizetések)",
    "3X": "3x",
    DURING_THE_FREE_SPIN: "Az ingyenes pörgetések során a játék mindig 5x5-ös, és
  mindig mindkét irányban fizet",
    TRIGGER_6_FREE_SPINS: "hat szabad pörgetést indít el.",
    FREE_SPIN_CAN_RETRIGGER: "Az ingyenes pörgetések újraaktiválhatók",
    GOOD_LUCK: "Good Luck",
    X_FREE_PLAY: "%{remainingCount} ingyenes játék maradt",
    X_FREE_PLAY_COMPLETED: "ingyenes játék maradt",
    FREE_SPIN_AWARDED: "%{spinCount} ingyenes játékot nyertél",
    FREE_SPIN_OVER: "ingyenes játékok vége",
    NO_THANKS: "nem, köszönöm",
    PLAY_LATER: "játssz később",
    PLAY_NOW: "játssz most"
  },
  PAYTABLE: {
    PAYTABLE: "NYEREMÉNYTÁBLA",
    PAYS_BOTH_WAYS: "Mindkét irányban fizet és kiterjeszti a mátrixot",
    EXTENDS_MATRIX: "Kiterjeszti a mátrixot",
    ACTIVATES_MIXED: "Vegyes kifizetéseket és bővülő mátrixot indít el",
    TRIGGER_6_FREE_SPIN: "hat szabad pörgetést indít el",
    AND_PAYS: "5x5-ös és mindkét irányban fizet",
    WHEN_THE_RED: "Amikor a Piros hold aktív, minden karakter állati formára vált, és
  rész vesznek a nyerő kombinációkban (vegyes kifizetések)",
    MALFUNCTION_TEXT: "ÜZEMZAVAR ESETÉN MINDEN NYEREMÉNY ÉS
  JÁTÉK ÉRVÉNYTELEN",
    GAME_RULES: "JÁTÉKSZABÁLYOK",
    GAME_RULES1: "A VOLSUNGA EGY OLYAN NYERŐUTAS JÁTÉK, AMELY
  53135-ÖS, 53335-ÖS, 55355-ÖS ÉS 55555-ÖS BŐVÜLŐ MÁTRIXSZAL RENDELKEZIK",
    GAME_RULES2: "A JÁTÉKOSOK 3125 NYERŐÚTTAL NYERHETNEK",

```

GAME\_RULES3: "A JÁTÉK MEGTÉRÜLÉSI ÉRTÉKE A JÁTÉKOS SZÁMÁRA  
 97,16% ",  
 GAME\_RULES4: "A SCATTER KIFIZETÉSEK KIVÉTELÉVEL MINDEN  
 KIFIZETÉS A BAL SZÉLSŐ TÁRCSÁRÓL INDULVA BALRÓL JOBBRA FIZET",  
 GAME\_RULES5: "A SCATTER KIFIZETÉSEK A TÁRCSÁKON BÁRMILYEN  
 POZÍCIÓBAN MEGJELENŐ SZIMBÓLUMOKÉRT FIZETNEK",  
 GAME\_RULES6: "NYERŐ KOMBINÁCIÓNKÉNT CSAK A LEGNAGYOBB  
 SZIMBÓLUMNYEREMÉNY FIZET",  
 GAME\_RULES7: "JÁTÉKONKÉNT CSAK A LEGNAGYOBB  
 SCATTERNYEREMÉNY FIZET",  
 GAME\_RULES8: "A SCATTER KIFIZETÉSEK A NYERŐUTAS KIFIZETÉSEK  
 MELLETT KERÜLNEK KIFIZETÉSRE",  
 GAME\_RULES9: "A NYERŐUTAS KIFIZETÉSEIT FELSZOROZZUK A  
 TÉTSZORZÓVAL",  
 GAME\_RULES10: "A SCATTER KIFIZETÉSEKET FELSZOROZZUK A TELJES  
 KREDITTÉTTTEL",  
 GAME\_RULES11: "AZ INGYENES PÖRGETÉSEK SORÁN A JÁTÉK MINDIG 5X5-  
 ÖS MÉRETŰ, ÉS MINDKÉT IRÁNYBAN FIZET",  
 GAME\_RULES12: "Az ingyenes pörgetések újraaktiválhatók"  
 },  
 BETSETTINGS: "TÉT BEÁLLÍTÁSOK",  
 AUTOPLAY: {  
 AUTOPLAY\_TITLE: "AUTOMATIKUS JÁTÉK",  
 AUTOPLAY\_SUBTITLE\_1: "Válassza meg a pörgetések számát!",  
 AUTOPLAY\_SUBTITLE\_2: "Állítsa le az automatikus játékot!",  
 AUTOPLAY\_DES\_1: "Ha a veszteség meghaladja",  
 AUTOPLAY\_DES\_2: "Ha egyetlen nyeremény meghaladja",  
 CLEAR: "törlés",  
 PLAY: "játás",  
 WIN\_LIMIT\_EXCEED: "nyeremény meghaladja",  
 LOSS\_LIMIT\_EXCEED: "veszteség meghaladja"  
 },  
 SOUNDSETTINGS: {  
 SOUND: "HANG",  
 BACKGROUND\_SOUND: "HÁTTÉRZENE"  
 },  
 MSG: {  
 BIG\_WIN: "SZÉP NYEREMÉNY",  
 HUGE\_WIN: "EXTRÉM NYEREMÉNY",  
 EPIC\_WIN: "ÓRIÁSI NYEREMÉNY",  
 MEGA\_WIN: "LEGENDÁS NYEREMÉNY",  
 CONGRATULATIONS: "GRATULÁLUNK",  
 CREDITS\_WON: "NYERT KREDITEK",  
 COMBINATION: "JUTALMAZOTT",  
 AWARDED: "KOMBINÁCIÓ(K)",  
 FREE\_SPINS: "INGYENES PÖRGETÉSEK",  
 ADDITIONAL\_FREE\_SPINS\_AWARDED: "TOVÁBBI INGYENES PÖRGETÉSEK  
 MEGSZEREZVE",  
 YOU\_HAVE\_WON: "AZ ÖN NYEREMÉNYE",  
 FREE\_SPINS\_AWARDED: "INGYENES PÖRGETÉSEK MEGSZEREZVE",  
 REALITY\_CHECK\_LOSS\_LIMIT: "Valóságvizsgálat veszteséghatára túllépve",  
 REALITY\_CHECK\_SESSION\_LENGTH: "Valóságvizsgálat játékmenet-hosszúsága

túllépve",

```
CLICK_TO_START: "Kattintson a kezdéshez",  
SESSION_TIME: "Játékidő",  
WIN_AMOUNT: "Nyeremény összege",  
LOSS_AMOUNT: "Veszteség összege",  
UNFINISHED_GAME_ROUND: "BEFEJEZETLEN JÁTÉKKÖRÖK"
```

```
},
```

```
BUTTON: {
```

```
OK: "OK",
```

```
CANCEL: "MÉGSE",
```

```
CONTINUE_PLAYING: "JÁTÉK FOLYTATÁSA",
```

```
CLEAR: "TÖRLÉS",
```

```
CONTINUE: "FOLYTATÁS",
```

```
AH: "Számlelőzmények",
```

```
RL: "VISSZA A LOBBYRA"
```

```
},
```

```
ERROR: {
```

```
101: "Érvénytelen jegy",
```

```
404: "kommunikációs hiba",
```

```
501: "Rendszerhiba. Kérjük, a hibakód megjelölésével forduljon az 501  
ügyfélszolgálathoz! ",
```

```
502: "Rendszerhiba. Kérjük, a hibakód megjelölésével forduljon az 502  
ügyfélszolgálathoz! ",
```

```
503: "Hiányzó vagy lejárt munkamenet ",
```

```
504: "Rendszerhiba. Kérjük, a hibakód megjelölésével forduljon az 504  
ügyfélszolgálathoz! ",
```

```
509: "Rendszerhiba. Kérjük, a hibakód megjelölésével forduljon az 509  
ügyfélszolgálathoz! ",
```

```
516: "Rendszerhiba. Kérjük, a hibakód megjelölésével forduljon az 516  
ügyfélszolgálathoz! ",
```

```
517: "Rendszerhiba. Kérjük, a hibakód megjelölésével forduljon az 517  
ügyfélszolgálathoz! ",
```

```
518: "Rendszerhiba. Kérjük, a hibakód megjelölésével forduljon az 518  
ügyfélszolgálathoz! ",
```

```
522: "Rendszerhiba. Kérjük, a hibakód megjelölésével forduljon az 522  
ügyfélszolgálathoz! ",
```

```
523: "Rendszerhiba. Kérjük, a hibakód megjelölésével forduljon az 523  
ügyfélszolgálathoz! ",
```

```
541: "Rendszerhiba. Kérjük, a hibakód megjelölésével forduljon az 541  
ügyfélszolgálathoz! ",
```

```
543: "Rendszerhiba. Kérjük, a hibakód megjelölésével forduljon az 543  
ügyfélszolgálathoz! ",
```

```
551: "Rendszerhiba. Kérjük, a hibakód megjelölésével forduljon az 551  
ügyfélszolgálathoz! ",
```

```
505: "Érvénytelen játékfolyamat. Kérjük, hogy forduljon az ügyféltámogatáshoz az 505-  
ös hibakóddal!",
```

```
507: "Érvénytelen tét. Kérjük, hogy forduljon az ügyféltámogatáshoz az 507-es  
hibakóddal!",
```

```
508: "Rendszerparancshiba. Kérjük, hogy forduljon az ügyféltámogatáshoz az 508-as  
hibakóddal!",
```

```
519: "Érvénytelen bejelentkezés. Kérjük, hogy forduljon az ügyféltámogatáshoz az 519-es  
hibakóddal!",
```

520: "A játék egy másik böngészőben aktív.",  
525: "Pénznemhiba. Kérjük, hogy forduljon az ügyféltámogatáshoz az 525-ös hibakóddal!",  
526: "Bejelentkezési hiba. Kérjük, hogy forduljon az ügyféltámogatáshoz az 526-os hibakóddal!",  
537: "Bejelentkezési hiba. Kérjük, hogy forduljon az ügyféltámogatáshoz az 537-os hibakóddal!",  
527: "Ingyenes pörgetés hiba. Kérjük, hogy forduljon az ügyféltámogatáshoz az 527-es hibakóddal!",  
532: "Ingyenes pörgetés hiba. Kérjük, hogy forduljon az ügyféltámogatáshoz az 532-es hibakóddal!",  
528: "Érvénytelen URL.",  
533: "Tokenhiba. Kérjük, hogy forduljon az ügyféltámogatáshoz az 533-as hibakóddal!",  
536: "Tokenhiba. Kérjük, hogy forduljon az ügyféltámogatáshoz az 536-as hibakóddal!",  
549: "Tokenhiba. Kérjük, hogy forduljon az ügyféltámogatáshoz az 549-as hibakóddal!",  
535: "játékverzió túl régi.",  
538: "Jackpot hiba. Kérjük, hogy forduljon az ügyféltámogatáshoz az 538-as hibakóddal!",  
539: "Jackpot hiba. Kérjük, hogy forduljon az ügyféltámogatáshoz az 539-as hibakóddal!",  
540: "A játékos egy aktív játékkal rendelkezik egy másik kiszolgálónál.",  
542: "Verzióhiba. Kérjük, hogy forduljon az ügyféltámogatáshoz az 542-es hibakóddal!",  
544: "Költési határérték átlépve",  
547: "Pénznemhiba. Kérjük, hogy forduljon az ügyféltámogatáshoz az 547-es hibakóddal!",  
550: "Nyereménykorlát elérve",  
506: "Nincs elegendő egyenleg a kiválasztott tétthez ",  
521: "A bejelentkezés során műszaki hiba történt ",  
524: "Játék letiltva ",  
531: "Játékból kitiltva ",  
534: "A bejelentkezési kísérlet egy letiltott régióból érkezik",  
545: "Valóságvizsgálat veszteséghatára túllépve",  
546: "Valóságvizsgálat játékmenet-hosszúsága túllépve",  
548: "Playtech valóságvizsgálat munkamenete lejárt",  
554: "Blocked from playing - daily time limit exceeded",  
555: "Blocked from playing - weekly time limit exceeded",  
556: "Blocked from playing - monthly time limit exceeded"

```

}
}, cc._RF.pop()
}, {}],
i18n: [function(e, t, i) {
  "use strict";
  cc._RF.push(t, "d9aabrif3BLwLsDPI3Pis3F", "i18n");
  var n = new (e("polyglot"))({
    phrases: e("en")
  });
  t.exports = {
    init: function(t) {
      var i = e(t);
      n.replace(i)
    },
    t: function(e, t) {

```

```

        return n.t(e, t)
    }
}, cc._RF.pop()
}, {
  en: "en",
  polyglot: "polyglot"
}],
is: [function(e, t, i) {
  "use strict";
  cc._RF.push(t, "af4011brLxHqad/Cx6JhyXQ", "is"), t.exports = {
    WELCOME: "Welcome, %{name}!",
    Click_anywhere_on_the_screen_to_start_the_game: "Smelltu hvar sem er á skjánum til að hefja leik",
    HUD: {
      BET: "VEÐ",
      CREDITS: "PUNKTAR",
      COIN_VALUE: "VIRÐI SPILAPENINGS",
      WIN: "VINNINGUR",
      AUTOPLAY: "SJÁLFSPILUN",
      MAXBET: "HÁMARKSVEÐ",
      "576WAYS_TO_WIN": "576 LEIÐIR TIL AÐ VINNA!",
      WIN_UP_TO_128_FREE_SPINS: "NÆLDU Í ALLT AÐ 128 FRÍSNÚNINGA",
      FREE: "FRJÁLS"
    },
    HUD_MESSAGES: {
      EXTEND_MATRIX: "Extends matrix!!",
      PAY_BOTH_WAYS_MATRIX: "Pays both ways and extends matrix!!",
      ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Activates mixed pays and extends matrix!!",
      IS_ACTIVE_ALL: " is active, all characters shape-shift into their animal form and share wins (mixed pays)!!",
      "3X": "3x",
      DURING_THE_FREE_SPIN: "During free spins, the game is always 5x5 and always pays both ways",
      TRIGGER_6_FREE_SPINS: "trigger 6 free spins is active.!!",
      FREE_SPIN_CAN_RETRIGGER: "Free spins can retrigger",
      GOOD_LUCK: "Good Luck!!",
      X_FREE_PLAY: "%{remainingCount} free plays remaining",
      X_FREE_PLAY_COMPLETED: "free plays remaining",
      FREE_SPIN_AWARDED: "You have been awarded %{spinCount} free spins",
      FREE_SPIN_OVER: "free spins over"
    },
    PAYTABLE: {
      PAYTABLE: "PAYTABLE",
      PAYS_BOTH_WAYS: "Pays both ways and extends matrix",
      EXTENDS_MATRIX: "Extends Matrix",
      ACTIVATES_MIXED: "Activates mixed pays and extends matrix",
      TRIGGER_6_FREE_SPIN: "trigger 6 free spins",
      AND_PAYS: "5x5 and pays both ways",
      WHEN_THE_RED: "When the Red Moon is active, all characters shape-shift into their animal form and share wins (mixed pays)",
      MALFUNCTION_TEXT: "Malfunction voids all pays and plays",

```

```

    GAME_RULES: "GAME_RULES",
    GAME_RULES1: "Volsunga is a ways game with an extending 53135, 53335, 55355 and
55555 matrix",
    GAME_RULES2: "Players have 3125 ways to win",
    GAME_RULES3: "The game has a 97.16% RTP",
    GAME_RULES4: "All pays except scatter pays are left-to-right starting from the leftmost
reel",
    GAME_RULES5: "Scatter pays are paid for symbols appearing in any position on the
reels",
    GAME_RULES6: "Only the highest symbol win is paid per winning combination",
    GAME_RULES7: "Only the highest scatter win is paid per play",
    GAME_RULES8: "Scatter pays are paid in addition to ways pays",
    GAME_RULES9: "Ways pays are multiplied by the bet multiple",
    GAME_RULES10: "Scatter pays are multiplied by the total credits bet",
    GAME_RULES11: "During free spins the game is always 5x5 and always pays both
ways",
    GAME_RULES12: "Free spins can retrigger"
},
BETSETTINGS: "VEÐSTILLINGAR",
AUTOPLAY: {
    AUTOPLAY_TITLE: "SJÁLFSPILUN",
    AUTOPLAY_SUBTITLE_1: "Veldu fjölda snúninga",
    AUTOPLAY_SUBTITLE_2: "Stöðva sjálfspilun",
    AUTOPLAY_DES_1: "Ef tap fer umfram",
    AUTOPLAY_DES_2: "Ef stakur vinningur fer umfram",
    CLEAR: "Hreinsa",
    PLAY: "Spila",
    WIN_LIMIT_EXCEED: "WIN LIMIT EXCEED",
    LOSS_LIMIT_EXCEED: "LOSS LIMIT EXCEED"
},
SOUNDSETTINGS: {
    SOUND: "HLJÓÐ",
    BACKGROUND_SOUND: "BAKGRUNNSHLJÓÐ"
},
MSG: {
    BIG_WIN: "GÓÐUR VINNINGUR!!",
    HUGE_WIN: "FLOTTUR VINNINGUR!!",
    EPIC_WIN: "RISA VINNINGUR!!",
    MEGA_WIN: "GEGGJAÐUR VINNINGUR!!",
    CONGRATULATIONS: "TIL HAMINGJU",
    CREDITS_WON: "UNNIR PUNKTAR",
    COMBINATION: "FLÉTTA/FLÉTTUR",
    AWARDED: "ÚTHLUTAÐ",
    FREE_SPINS: "FREE SPINS",
    ADDITIONAL_FREE_SPINS_AWARDED: "ADDITIONAL FREE SPINS
AWARDED",
    YOU_HAVE_WON: "YOU HAVE WON",
    FREE_SPINS_AWARDED: "FREE SPINS AWARDED",
    REALITY_CHECK_LOSS_LIMIT: "Promemoria - Limite di perdita superato",
    REALITY_CHECK_SESSION_LENGTH: "Promemoria - Limite di durata della sessione
superato"
},

```

```
BUTTON: {
  OK: "OK",
  CANCEL: "CANCEL",
  CONTINUE_PLAYING: "CONTINUE PLAYING!",
  CLEAR: "CLEAR",
  CONTINUE: "CONTINUE",
  RL: "RETURN TO LOBBY"
},
ERROR: {
  101: "Invalid Ticket",
  501: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando il codice di errore 501",
  502: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando il codice di errore 502",
  503: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando il codice di errore 503",
  504: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando il codice di errore 504",
  509: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando il codice di errore 509",
  516: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando il codice di errore 516",
  517: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando il codice di errore 517",
  518: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando il codice di errore 518",
  522: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando il codice di errore 522",
  523: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando il codice di errore 523",
  541: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando il codice di errore 541",
  543: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando il codice di errore 543",
  551: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando il codice di errore 551",
  505: "----Invalid game progress. Please contact support with error code ",
  507: "----Invalid bet. Please contact support with error code ",
  508: "----System command error. Please contact support with error code ",
  519: "----Invalid login. Please contact support with error code ",
  520: "----Game is active in another browser. ",
  525: "----Currency error. Please contact support with error code ",
  526: "----Login error. Please contact support with error code ",
  537: "----Login error. Please contact support with error code ",
  527: "----Fre spins error. Please contact support with error code ",
  532: "----Fre spins error. Please contact support with error code ",
  528: "----Invalid URL. ",
  533: "----Token error. Please contact support with error code ",
  536: "----Token error. Please contact support with error code ",
  549: "----Token error. Please contact support with error code ",
  535: "----Game version too old. ",
  538: "----Jackpot error. Please contact support with error code ",
```

```

539: "----Jackpot error. Please contact support with error code ",
540: "----Player has active game in another server. ",
542: "----Version error. Please contact support with error code ",
544: "----Spending budget exceeded ",
547: "----Currency error. Please contact support with error code ",
550: "----Win cap reached ",
506: "----Not enough funds for chosen betsize ",
521: "----Technical error during login ",
524: "----Game disabled. ",
531: "----Blocked from playing ",
534: "----Login attempt comes from blocked region ",
545: "----Reality check loss limit exceeded ",
546: "----Reality check session length exceeded ",
548: "----Playtech reality check session expired ",
554: "Blocked from playing - daily time limit exceeded",
555: "Blocked from playing - weekly time limit exceeded",
556: "Blocked from playing - monthly time limit exceeded"
}
}, cc._RF.pop()
}, {}],
it: [function(e, t, i) {
"use strict";
cc._RF.push(t, "b3159B4/VVAi4agctN8J4bz", "it"), t.exports = {
WELCOME: "Welcome, %{name}!",
Click_anywhere_on_the_screen_to_start_the_game: "Clicca ovunque sullo schermo per
avviare il gioco",
HUD: {
BET: "PUNTATA",
CREDITS: "CREDITI",
COIN_VALUE: "VALORE MONETA",
WIN: "VINCITA",
AUTOPLAY: "AUTOPLAY",
MAXBET: "PUNTATA MASSIMA",
"576WAYS_TO_WIN": "576 COMBINAZIONI VINCENTI!",
WIN_UP_TO_128_FREE_SPINS: "VINCI FINO A 128 GIRI GRATIS",
FREE: "GRATIS"
},
HUD_MESSAGES: {
EXTEND_MATRIX: "Estende la griglia",
PAY_BOTH_WAYS_MATRIX: "Paga in entrambe le direzioni ed estende la griglia",
ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Attiva i pagamenti misti ed
estende la griglia",
IS_ACTIVE_ALL: " Quando la Luna Rossa è attiva, tutti i personaggi assumono la loro
forma animale e condividono le vincite (pagamenti misti)",
"3X": "3x",
DURING_THE_FREE_SPIN: "Durante i Giri Gratis, la partita mantiene lo schema 5x5 e
paga in entrambe le direzioni",
TRIGGER_6_FREE_SPINS: "attivano 6 Giri Gratis",
FREE_SPIN_CAN_RETRIGGER: "I Giri Gratis possono essere riattivati",
GOOD_LUCK: "Good Luck",
X_FREE_PLAY: "rimangono %{remainingCount} partite gratis",
X_FREE_PLAY_COMPLETED: "rimangono partite gratis",

```

```

FREE_SPIN_AWARDED: "ti sono state assegnate %{spinCount} partite gratis",
FREE_SPIN_OVER: "partite gratis terminate",
NO_THANKS: "no, grazie",
PLAY_LATER: "gioca più tardi",
PLAY_NOW: "gioca ora"
},
PAYTABLE: {
  PAYTABLE: "TABELLA DEI PAGAMENTI",
  PAYS_BOTH_WAYS: "Paga in entrambe le direzioni ed estende la griglia",
  EXTENDS_MATRIX: "Estende la griglia",
  ACTIVATES_MIXED: "Attiva i pagamenti misti ed estende la griglia",
  TRIGGER_6_FREE_SPIN: "attivano 6 Giri Gratis",
  AND_PAYS: "5x5 e pagamenti in entrambe le direzioni",
  WHEN_THE_RED: "Quando la Luna Rossa è attiva, tutti i personaggi assumono la loro
forma animale e condividono le vincite (pagamenti misti)",
  MALFUNCTION_TEXT: "I MALFUNZIONAMENTI ANNULLANO TUTTE LE
VINCITE E LE PUNTATE",
  GAME_RULES: "REGOLE DI GIOCO",
  GAME_RULES1: "VOLSUNGA È UN GIOCO CON MODALITÀ E UNA GRIGLIA
ESTENDIBILE 53135, 53335, 55355 E 55555",
  GAME_RULES2: "I GIOCATORI HANNO 3125 MODALITÀ DI VINCITA",
  GAME_RULES3: "IL GIOCO OFFRE UN RTP DEL 97,16%.",
  GAME_RULES4: "TUTTI I PREMI, TRANNE GLI SCATTER, PAGANO DA
SINISTRA A DESTRA, A PARTIRE DAL PRIMO RULLO A SINISTRA",
  GAME_RULES5: "I PREMI SCATTER SONO PAGATI PER SIMBOLI COMPARI
OVUNQUE SUI RULLI",
  GAME_RULES6: "VIENE PAGATA SOLO LA VINCITA DA SIMBOLI PIÙ ALTA
PER COMBINAZIONE VINCENTE",
  GAME_RULES7: "VIENE PAGATA SOLO LA VINCITA SCATTER PIÙ ALTA PER
PARTITA.",
  GAME_RULES8: "LE VINCITE SCATTER SONO SOMMATE A QUELLE DELLE
MODALITÀ",
  GAME_RULES9: "I PREMI DELLE MODALITÀ SONO MOLTIPLICATI PER IL
MOLTIPLICATORE DELLA PUNTATAe",
  GAME_RULES10: "LE VINCITE SCATTER SONO MOLTIPLICATE PER LA
PUNTATA TOTALE IN CREDITI",
  GAME_RULES11: "DURANTE I GIRI GRATIS, IL GIOCO È SEMPRE 5X5 E PAGA
SEMPRE IN ENTRAMBE LE DIREZIONI",
  GAME_RULES12: "I GIRI GRATIS POSSONO ESSERE RIATTIVATI"
},
BETSETTINGS: "IMPOSTAZIONI PUNTATA",
AUTOPLAY: {
  AUTOPLAY_TITLE: "AUTOPLAY",
  AUTOPLAY_SUBTITLE_1: "Seleziona il numero di giri",
  AUTOPLAY_SUBTITLE_2: "Interrompi Autoplay",
  AUTOPLAY_DES_1: "Se al perdita è maggiore",
  AUTOPLAY_DES_2: "Se la vincita singola è maggiore",
  CLEAR: "Cancella",
  PLAY: "Gioca",
  WIN_LIMIT_EXCEED: "vincita è maggiore",
  LOSS_LIMIT_EXCEED: "perdita è maggiore"
},

```

```

SOUNDSETTINGS: {
    SOUND: "AUDIO",
    BACKGROUND_SOUND: "SUONO DI SOTTOFONDO"
},
MSG: {
    BIG_WIN: "BELLA VINCITA",
    HUGE_WIN: "VINCITA ESTREMA",
    EPIC_WIN: "VINCITA MASSIVA",
    MEGA_WIN: "VINCITA GIGANTESCA",
    CONGRATULATIONS: "CONGRATULAZIONI",
    CREDITS_WON: "CREDITI VINTI",
    COMBINATION: "COMBINAZIONE(I)",
    AWARDED: "VINTI",
    FREE_SPINS: "GIRI GRATUITI",
    ADDITIONAL_FREE_SPINS_AWARDED: "GIRI GRATIS EXTRA ASSEGNATI",
    YOU_HAVE_WON: "HAI VINTO",
    FREE_SPINS_AWARDED: "GIRO GRATUITO ASSEGNATO",
    REALITY_CHECK_LOSS_LIMIT: "Promemoria - Limite di perdita superato",
    REALITY_CHECK_SESSION_LENGTH: "Promemoria - Limite di durata della sessione
superato",
    CLICK_TO_START: "Premi per iniziare",
    SESSION_TIME: "Durata della sessione",
    WIN_AMOUNT: "Importo vinto",
    LOSS_AMOUNT: "Importo perso",
    UNFINISHED_GAME_ROUND: "ROUND DI GIOCO NON TERMINATI"
},
BUTTON: {
    OK: "OK",
    CANCEL: "ANNULLA",
    CONTINUE_PLAYING: "CONTINUA A GIOCARE",
    CLEAR: "ELIMINA",
    CONTINUE: "CONTINUA",
    AH: "Cronologia del conto",
    RL: "TORNA ALLA LOBBY"
},
ERROR: {
    101: "Biglietto non valido",
    404: "Errore di comunicazione",
    501: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando
il codice di errore 501",
    502: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando
il codice di errore 502",
    503: "Sessione assente o scaduta",
    504: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando
il codice di errore 504",
    509: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando
il codice di errore 509",
    516: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando
il codice di errore 516",
    517: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando
il codice di errore 517",
    518: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando

```

il codice di errore 518",  
522: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando il codice di errore 522",  
523: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando il codice di errore 523",  
541: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando il codice di errore 541",  
543: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando il codice di errore 543",  
551: "Errore di sistema. Ti preghiamo di contattare il servizio di assistenza menzionando il codice di errore 551",  
505: "Progresso di gioco non valido. Contatta il servizio di assistenza indicando il codice di errore 505.",  
507: "Puntata non valida. Contatta il servizio di assistenza indicando il codice di errore 507.",  
508: "Errore di comando del sistema. Contatta il servizio di assistenza indicando il codice di errore 508.",  
519: "Accesso non valido. Contatta il servizio di assistenza indicando il codice di errore 519.",  
520: "Gioco attivo in un altro browser.",  
525: "Errore della valuta. Contatta il servizio di assistenza indicando il codice di errore 525.",  
526: "Errore di accesso. Contatta il servizio di assistenza indicando il codice di errore 526.",  
537: "Errore di accesso. Contatta il servizio di assistenza indicando il codice di errore 537.",  
527: "Errore dei giri gratis. Contatta il servizio di assistenza indicando il codice di errore 527.",  
532: "Errore dei giri gratis. Contatta il servizio di assistenza indicando il codice di errore 532.",  
528: "URL non valido.",  
533: "Errore del token. Contatta il servizio di assistenza indicando il codice di errore 533.",  
536: "Errore del token. Contatta il servizio di assistenza indicando il codice di errore 536.",  
549: "Errore del token. Contatta il servizio di assistenza indicando il codice di errore 549.",  
535: "La versione del gioco è obsoleta.",  
538: "Errore del Jackpot. Contatta il servizio di assistenza indicando il codice di errore 538.",  
539: "Errore del Jackpot. Contatta il servizio di assistenza indicando il codice di errore 539.",  
540: "Il giocatore ha una partita attiva su un altro server.",  
542: "Errore della versione. Contatta il servizio di assistenza indicando il codice di errore 542.",  
544: "Budget di spesa superato",  
547: "Errore della valuta. Contatta il servizio di assistenza indicando il codice di errore 547.",  
550: "Limite di vincita raggiunto",  
506: "Fondi insufficienti per la puntata scelta",  
521: "Techninè klaida prisijungiant",  
524: "Gioco disattivato",

```

531: "Gioco bloccato",
534: "Tentativo di accesso da una regione bloccata",
545: "Promemoria - Limite di perdita superato",
546: "Promemoria - Limite di durata della sessione superato",
548: "Promemoria di Playtech - Sessione scaduta",
554: "Blocked from playing - daily time limit exceeded",
555: "Blocked from playing - weekly time limit exceeded",
556: "Blocked from playing - monthly time limit exceeded"
}
}, cc._RF.pop()
}, {}],
lt: [function(e, t, i) {
"use strict";
cc._RF.push(t, "9f65bjQ5aJIIJKsV5fZnNQ", "lt"), t.exports = {
WELCOME: "Welcome, %{name}!",
Click_anywhere_on_the_screen_to_start_the_game: "Paspauskite bet kur, norėdami pradėti
žaidimą",
HUD: {
BET: "STATYMAS",
CREDITS: "KREDITAI",
COIN_VALUE: "MONETOS VERTĖ",
WIN: "LAIMĖTI",
AUTOPLAY: "AUTOMATINIS ŽAIDIMAS",
MAXBET: "MAKSIMALUS STATYMAS",
"576WAYS_TO_WIN": "576 BŪDAI LAIMĖTI!",
WIN_UP_TO_128_FREE_SPINS: "LAIMĖKITE IKI 128 NEMOKAMŲ SUKIMŲ",
FREE: "NEMOKAMŲ"
},
HUD_MESSAGES: {
EXTEND_MATRIX: "Išplečia matricą",
PAY_BOTH_WAYS_MATRIX: "Moka abiem kryptimis ir išplečia matricą",
ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Aktyvina mišriąsias
išmokas ir išplečia matricą",
IS_ACTIVE_ALL: " Kai funkcija „Red Moon“ aktyvi, visi simboliai pakeičia pavidalą į
gyvūnų, o laimėjimai tampa bendri (mišriosios išmokos)",
"3X": "3x",
DURING_THE_FREE_SPIN: "Nemokamų sukimų metu, žaidimo formatas visada būna
5 x 5, be to, visada mokama abiem kryptimis",
TRIGGER_6_FREE_SPINS: "aktyvina 6 nemokamus sukimus",
FREE_SPIN_CAN_RETRIGGER: "Nemokami sukimai gali būti pakartotinai
aktyvinami",
GOOD_LUCK: "Good Luck",
X_FREE_PLAY: "Liko %{remainingCount} nemokami žaidimai",
X_FREE_PLAY_COMPLETED: "Liko nemokami žaidimai",
FREE_SPIN_AWARDED: "jums padovanoti %{spinCount} nemokami žaidimai",
FREE_SPIN_OVER: "baigėsi nemokami žaidimai",
NO_THANKS: "ne, ačiū",
PLAY_LATER: "žaisti vėliau",
PLAY_NOW: "žaisti dabar"
},
PAYTABLE: {
PAYTABLE: "PAYTABLE",

```

PAYS\_BOTH\_WAYS: "Moka abiem kryptimis ir išplečia matricą",  
EXTENDS\_MATRIX: "Išplečia matricą",  
ACTIVATES\_MIXED: "Aktyvina mišriąsias išmokas ir išplečia matricą",  
TRIGGER\_6\_FREE\_SPIN: "trigger 6 free spins",  
AND\_PAYS: "5 x 5 ir mokama abiem kryptimis",  
WHEN\_THE\_RED: "Kai funkcija „Red Moon“ aktyvi, visi simboliai pakeičia pavidalą į gyvūnų, o laimėjimai tampa bendri (mišriosios išmokos)",  
MALFUNCTION\_TEXT: "ĮVYKUS TRIKČIUI, ANULIUOJAMI VISI MOKĖJIMAI IR ŽAIDIMAI",  
GAME\_RULES: "ŽAIDIMO TAISYKLĖS",  
GAME\_RULES1: "„VOLSUNGA“ YRA DAUGIAVARIANTINIS ŽAIDIMAS SU IŠSIPLEČIANČIA 53135, 53335, 55355 IR 55555 MATRICA",  
GAME\_RULES2: "ŽAIDĖJUI SUTEIKIAMI 3125 VARIANTAI LAIMĖTI",  
GAME\_RULES3: "ŽAIDIMAS PASIŽYMI 97,16 % RTP (GRAŽA ŽAIDĖJUI)",  
GAME\_RULES4: "VISOS IŠMOKOS, IŠSKYRUS „SCATTER“ IŠMOKAS, SKAIČIUOJAMOS IŠ KAIRĖS Į DEŠINĘ, PRADEDANT NUO KAIRIAUSIO BŪGNO",  
GAME\_RULES5: "„SCATTER“ IŠMOKOS MOKAMOS UŽ SIMBOLIUS, PASIRODANČIUS BET KURIOSE BŪGNŲ PADĖTYSE",  
GAME\_RULES6: "PER LAIMINČIAŲ KOMBINACIJĄ MOKAMA TIK UŽ AUKŠČIAUSIĄ SIMBOLIŲ LAIMĖJIMĄ",  
GAME\_RULES7: "PER VIENĄ ŽAIDIMĄ MOKAMA TIK UŽ AUKŠČIAUSIĄ „SCATTER“ LAIMĖJIMĄ",  
GAME\_RULES8: "„SCATTER“ IŠMOKOS MOKAMOS PAPILDOMAI PRIE VARIANTINIŲ IŠMOKŲ",  
GAME\_RULES9: "VARIANTINĖS IŠMOKOS DAUGINAMOS IŠ STATYMO DAUGIKLIO",  
GAME\_RULES10: "„SCATTER“ IŠMOKOS DAUGINAMOS IŠ VISŲ KREDITŲ STATYMO",  
GAME\_RULES11: "ŽAIDIMAS PER NEMOKAMUS SUKIMUS VISADA YRA 5 X 5, BE TO, VISADA MOKAMA ABIEM KRYPTIMIS",  
GAME\_RULES12: "GALIMA PAKARTOTINAI AKTYVINTI NEMOKAMUS SUKIMUS",  
},  
BETSETTINGS: "STATYMO NUSTATYMAI",  
AUTOPLAY: {  
AUTOPLAY\_TITLE: "AUTOMATINIS ŽAIDIMAS",  
AUTOPLAY\_SUBTITLE\_1: "Pasirinkite sukimų skaičių",  
AUTOPLAY\_SUBTITLE\_2: "Sustabdykite automatinį žaidimą",  
AUTOPLAY\_DES\_1: "Jei nuostolis viršija",  
AUTOPLAY\_DES\_2: "Jei vienintelis laimėjimas viršija",  
CLEAR: "Išvalyti",  
PLAY: "Žaisti",  
WIN\_LIMIT\_EXCEED: "laimėjimas viršija",  
LOSS\_LIMIT\_EXCEED: "nuostolis viršija"  
},  
SOUNDSETTINGS: {  
SOUND: "GARSAS",  
BACKGROUND\_SOUND: "FONO GARSAS"  
},  
MSG: {  
BIG\_WIN: "GRAŽUS LAIMĖJIMAS",  
HUGE\_WIN: "EKSTREMALUS LAIMĖJIMAS",

```

EPIC_WIN: "DIDELIS LAIMĖJIMAS",
MEGA_WIN: "MILŽINIŠKAS LAIMĖJIMAS",
CONGRATULATIONS: "SVEIKINAME",
CREDITS_WON: "LAIMĖTI KREDITAI",
COMBINATION: "KOMBINACIJA(OS)",
AWARDED: "APDOVANOTA",
FREE_SPINS: "NEMOKAMI SUKIMAI",
ADDITIONAL_FREE_SPINS_AWARDED: "LAIMĖTA PAPILDOMŲ NEMOKAMŲ
SUKIMŲ",
YOU_HAVE_WON: "LAIMĖJOTE",
FREE_SPINS_AWARDED: "DOVANOTI NEMOKAMI SUKIMAI",
REALITY_CHECK_LOSS_LIMIT: "Viršyta realybės patikros pralaimėjimo riba",
REALITY_CHECK_SESSION_LENGTH: "Viršyta realybės patikros sesijos trukmė",
CLICK_TÖ_START: "Spragtelėkite norėdami pradėti",
SESSION_TIME: "Sesijos laikas",
WIN_AMOUNT: "Laimėjimo suma",
LOSS_AMOUNT: "Prarandama suma",
UNFINISHED_GAME_ROUND: "NEBAIGTI ŽAIDIMO RAUNDAI"
},
BUTTON: {
    OK: "GERAI",
    CANCEL: "ATŠAUKTI",
    CONTINUE_PLAYING: "TOLIAU ŽAISTI",
    CLEAR: "PAŠALINTI",
    CONTINUE: "TĘSTI",
    AH: "Paskyros istorija",
    RL: "GRAŽINIMAS Į LOBBĄ"
},
ERROR: {
    101: "Neteisingas bilietas",
    404: "komunikacijos klaida",
    501: "Sistemos klaida. Kreipkitės pagalbos su klaidos kodu 501",
    502: "Sistemos klaida. Kreipkitės pagalbos su klaidos kodu 502",
    503: "Trūksta sesijos arba ji nustojo galioti",
    504: "Sistemos klaida. Kreipkitės pagalbos su klaidos kodu 504",
    509: "Sistemos klaida. Kreipkitės pagalbos su klaidos kodu 509",
    516: "Sistemos klaida. Kreipkitės pagalbos su klaidos kodu 516",
    517: "Sistemos klaida. Kreipkitės pagalbos su klaidos kodu 517",
    518: "Sistemos klaida. Kreipkitės pagalbos su klaidos kodu 518",
    522: "Sistemos klaida. Kreipkitės pagalbos su klaidos kodu 522",
    523: "Sistemos klaida. Kreipkitės pagalbos su klaidos kodu 523",
    541: "Sistemos klaida. Kreipkitės pagalbos su klaidos kodu 541",
    543: "Sistemos klaida. Kreipkitės pagalbos su klaidos kodu 543",
    551: "Sistemos klaida. Kreipkitės pagalbos su klaidos kodu 551",
    505: "Negaliojanti žaidimo pažanga. Kreipkitės į pagalbos skyrių su klaidos kodu 505.",
    507: "Negaliojantis statymas. Kreipkitės į pagalbos skyrių su klaidos kodu 507.",
    508: "Sistemos komandos klaida. Kreipkitės į pagalbos skyrių su klaidos kodu 508.",
    519: "Negaliojantys prisijungimo duomenys. Kreipkitės į pagalbos skyrių su klaidos kodu
519.",
    520: "Žaidimas aktyvus kitoje naršyklėje. ",
    525: "Valiutos klaida. Kreipkitės į pagalbos skyrių su klaidos kodu 525.",
    526: "Prisijungimo klaida. Kreipkitės į pagalbos skyrių su klaidos kodu 526."
}

```

537: "Prisijungimo klaida. Kreipkitės į pagalbos skyrių su klaidos kodu 537.",  
527: "Nemokamų sukimų klaida. Kreipkitės į pagalbos skyrių su klaidos kodu 527.",  
532: "Nemokamų sukimų klaida. Kreipkitės į pagalbos skyrių su klaidos kodu 532.",  
528: "Negaliojantis URL adresas.",  
533: "Žetonų klaida. Kreipkitės į pagalbos skyrių su klaidos kodu 533.",  
536: "Žetonų klaida. Kreipkitės į pagalbos skyrių su klaidos kodu 536.",  
549: "Žetonų klaida. Kreipkitės į pagalbos skyrių su klaidos kodu 549.",  
535: "Žaidimo versija per sena.",  
538: "Aukso puodo klaida. Kreipkitės į pagalbos skyrių su klaidos kodu 538.",  
539: "Aukso puodo klaida. Kreipkitės į pagalbos skyrių su klaidos kodu 539.",  
540: "Žaidėjas kitame serveryje turi aktyvų žaidimą.",  
542: "Versijos klaida. Kreipkitės į pagalbos skyrių su klaidos kodu 542.",  
544: "Viršytas išlaidų biudžetas",  
547: "Valiutos klaida. Kreipkitės į pagalbos skyrių su klaidos kodu 547.",  
550: "Pasiekta laimėjimo riba",  
506: "Pasirinktam statymo dydžiui neužtenka lėšų",  
521: "Techninė klaida prisijungiant",  
524: "Žaidimas išjungtas",  
531: "Užblokuota galimybė žaisti",  
534: "Bandoma prisijungti iš užblokuoto regiono.",  
545: "Viršyta realybės patikros pralaimėjimo riba",  
546: "Viršyta realybės patikros sesijos trukmė",  
548: "Baigėsi „Playtech“ realybės patikros sesijos galiojimas",  
554: "Blocked from playing - daily time limit exceeded",  
555: "Blocked from playing - weekly time limit exceeded",  
556: "Blocked from playing - monthly time limit exceeded"

```
    }  
  }, cc._RF.pop()  
}, {}],  
lv: [function(e, t, i) {  
  "use strict";  
  cc._RF.push(t, "72078sac7xNbretxTGliB9f", "lv"), t.exports = {  
    WELCOME: "Welcome, %{name}!",  
    Click_anywhere_on_the_screen_to_start_the_game: "Click anywhere on the screen to start  
the game",  
    HUD: {  
      BET: "LIKME",  
      CREDITS: "SUMMA",  
      COIN_VALUE: "MONĖTAS VĒRTĪBA",  
      WIN: "UZVARA",  
      AUTOPLAY: "AUTO SPĒLE",  
      MAXBET: "MAX LIKMME",  
      "576WAYS_TO_WIN": "576 VEIDI, LAI UZVARĒTU!",  
      WIN_UP_TO_128_FREE_SPINS: "UZVARI LĪDZ 128 BEZMAKSAS GRIEZIENU",  
      FREE: "BEZMAKSAS"  
    },  
    HUD_MESSAGES: {  
      EXTEND_MATRIX: "Izvērš matricu",  
      PAY_BOTH_WAYS_MATRIX: "Izmaksā laimestus abos virzienos un izvērš matricu!!",  
      ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Aktivizē jauktos laimestus  
un izvērš matricu!!",  
      IS_ACTIVE_ALL: " Kad ir aktivizēts sarkanais mēness, visi personāži ieņem savu
```

dzīvnieka veidolu un kopīgi laimē (jauktie laimesti).!!",  
 "3X": "3x",  
 DURING\_THE\_FREE\_SPIN: "Bezmaksas griezienu laikā vienmēr spēlē ar 5x5 spolēm un laimesti vienmēr tiek izmaksāti abos virzienos",  
 TRIGGER\_6\_FREE\_SPINS: "ierosina 6 bezmaksas griezienus.!!",  
 FREE\_SPIN\_CAN\_RETRIGGER: "Bezmaksas griezieni var tikt ierosināti atkārtoti",  
 GOOD\_LUCK: "Good Luck!!",  
 X\_FREE\_PLAY: "atlikušas %{remainingCount} bezmaksas spēles",  
 X\_FREE\_PLAY\_COMPLETED: "atlikušas bezmaksas spēles",  
 FREE\_SPIN\_AWARDED: "jums ir piešķirtas %{spinCount} bezmaksas spēles",  
 FREE\_SPIN\_OVER: "bezmaksas spēles beigušās",  
 NO\_THANKS: "nē, paldies",  
 PLAY\_LATER: "spēlēt vēlāk",  
 PLAY\_NOW: "spēlēt tagad"  
 },  
 PAYTABLE: {  
 PAYTABLE: "LAIMESTU TABULA",  
 PAYS\_BOTH\_WAYS: "Izmaksā laimestus abos virzienos un izvērš matricu",  
 EXTENDS\_MATRIX: "Izvērš matricu",  
 ACTIVATES\_MIXED: "Aktivizē jauktos laimestus un izvērš matricu",  
 TRIGGER\_6\_FREE\_SPIN: "ierosina 6 bezmaksas griezienus",  
 AND\_PAYS: "5x5 spoles, un laimesti tiek izmaksāti abos virzienos",  
 WHEN\_THE\_RED: "Kad ir aktivizēts sarkanais mēness, visi personāži ieņem savu dzīvnieka veidolu un kopīgi laimē (jauktie laimesti).",  
 MALFUNCTION\_TEXT: "SISTĒMAS DISFUNKCIJA ATCEĻ VISUS IZMAKSĀJAMOS LAIMESTUS UN SPĒLES",  
 GAME\_RULES: "SPĒLES NOTEIKUMI",  
 GAME\_RULES1: "“VOLŠUNGA” IR VARIANTU SPĒLE AR IZVĒRŠAMĀM 53135, 53335, 55355 UN 55555 MATRICĀM.",  
 GAME\_RULES2: "SPĒLĒTĀJU RĪCĪBĀ IR 3125 LAIMESTU VARIANTI.",  
 GAME\_RULES3: "SPĒLĒ IEGULDĪTĀS NAUDAS ATMAKSĀŠANĀS LĪMENIS SPĒLĒTĀJAM IR 97,16 %.",  
 GAME\_RULES4: "VISI LAIMESTI TIEK IZMAKSĀTI NO KREISĀS UZ LABO PUSI, SĀKOT AR MALĒJO, KREISO SPOLI.",  
 GAME\_RULES5: "IZKAISĪTO SIMBOLU LAIMESTUS IZMAKSĀ PAR SIMBOLIEM, KĀS PARĀDĀS JEBKURĀ POZĪCIJĀ UZ SPOLĒM.",  
 GAME\_RULES6: "TIEK IZMAKSĀTS TIKAI AUGSTĀKAIS SIMBOLA LAIMESTS LAIMĒJUŠAJĀ KOMBINĀCIJĀ.",  
 GAME\_RULES7: "TIEK IZMAKSĀTS TIKAI AUGSTĀKAIS IZKAISĪTO SIMBOLU LAIMESTS SPĒLĒ.",  
 GAME\_RULES8: "IZKAISĪTO SIMBOLU LAIMESTI TIEK IZMAKSĀTI PAPILDUS VARIANTU LAIMESTIEM.",  
 GAME\_RULES9: "VARIANTU LAIMESTI TIEK SAREIZINĀTI AR LIKMES REIZINĀTĀJU.",  
 GAME\_RULES10: "IZKAISĪTO SIMBOLU LAIMESTI TIEK SAREIZINĀTI AR KOPĒJO KREDĪTPUNKTU LIKMI.",  
 GAME\_RULES11: "BEZMAKSAS GRIEZIENU LAIKĀ SPĒLI VIENMĒR SPĒLĒ 5X5 FORMĀTĀ, UN LAIMESTUS VIENMĒR IZMAKSĀ ABOS VIRZIENOS.",  
 GAME\_RULES12: "BEZMAKSAS GRIEZIENI VAR TIKT IEROSINĀTI ATKĀRTOTI,"  
 },  
 BETSETTINGS: "LIKMES UZSTĀDĪJUMI",

```

AUTOPLAY: {
    AUTOPLAY_TITLE: "AUTOMĀTISKĀ SPĒLE",
    AUTOPLAY_SUBTITLE_1: "Izvēlies griezienu skaitu",
    AUTOPLAY_SUBTITLE_2: "Apturēt automātisko spēlēšanu",
    AUTOPLAY_DES_1: "Ja zaudējums pārsniedz",
    AUTOPLAY_DES_2: "Ja viena līnija pārsniedz",
    CLEAR: "Notīrīt",
    PLAY: "Spēlēt",
    WIN_LIMIT_EXCEED: "viena līnija pārsniedz",
    LOSS_LIMIT_EXCEED: "zaudējums pārsniedz"
},
SOUNDSETTINGS: {
    SOUND: "SKAŅA",
    BACKGROUND_SOUND: "FONA SKAŅA"
},
MSG: {
    BIG_WIN: "LIELISKS LAIMESTS",
    HUGE_WIN: "MILZĪGA UZVARA!!",
    EPIC_WIN: "EPISKA UZVARA!!",
    MEGA_WIN: "MEGA UZVARA!!",
    CONGRATULATIONS: "UZSTĀDĪJUMI",
    CREDITS_WON: "KREDĪTI UZVARĒTI",
    COMBINATION: "KOMBINĀCIJA(S)",
    AWARDED: "APBALVOTS",
    FREE_SPINS: "BEZMAKSAS GRIEZIENI",
    ADDITIONAL_FREE_SPINS_AWARDED: "PIEŠĶIRTI PAPILDU BEZMAKSAS
GRIEZIENI",
    YOU_HAVE_WON: "JŪS ESAT LAIMĒJIS",
    FREE_SPINS_AWARDED: "PIEŠĶIRTI BEZMAKSAS GRIEZIENI",
    REALITY_CHECK_LOSS_LIMIT: "Pārsniegts realitātes kontroles zaudējuma
ierobežojums",
    REALITY_CHECK_SESSION_LENGTH: "Pārsniegts realitātes kontroles sesijas
ilgums",
    CLICK_TO_START: "Noklikšķiniet, lai sāktu.",
    SESSION_TIME: "Sesijas ilgums",
    WIN_AMOUNT: "Laimesta summa",
    LOSS_AMOUNT: "Zaudējuma summa",
    UNFINISHED_GAME_ROUND: "NEPABEIGTI SPĒLES RAUNDI"
},
BUTTON: {
    OK: "LABI",
    CANCEL: "ATCELT",
    CONTINUE_PLAYING: "TURPINĀT SPĒLĒT",
    CLEAR: "NOTĪRĪT",
    CONTINUE: "TURPINĀT",
    AH: "Konta vēsture",
    RL: "ATGRIEZTIES UZ LOBBIJU"
},
ERROR: {
    101: "Nederīga biļete",
    404: "sakarū klūda",
    501: "Sistēmas klūda. Lūdzu, sazinieties ar atbalsta dienestu, norādot klūdas kodu 501."
}

```

502: "Sistēmas kļūda. Lūdzu, sazinieties ar atbalsta dienestu, norādot kļūdas kodu 502.",  
503: "Nav sesijas vai tā ir beigusies",  
504: "Sistēmas kļūda. Lūdzu, sazinieties ar atbalsta dienestu, norādot kļūdas kodu 504.",  
509: "Sistēmas kļūda. Lūdzu, sazinieties ar atbalsta dienestu, norādot kļūdas kodu 509.",  
516: "Sistēmas kļūda. Lūdzu, sazinieties ar atbalsta dienestu, norādot kļūdas kodu 516.",  
517: "Sistēmas kļūda. Lūdzu, sazinieties ar atbalsta dienestu, norādot kļūdas kodu 517.",  
518: "Sistēmas kļūda. Lūdzu, sazinieties ar atbalsta dienestu, norādot kļūdas kodu 518.",  
522: "Sistēmas kļūda. Lūdzu, sazinieties ar atbalsta dienestu, norādot kļūdas kodu 522.",  
523: "Sistēmas kļūda. Lūdzu, sazinieties ar atbalsta dienestu, norādot kļūdas kodu 523.",  
541: "Sistēmas kļūda. Lūdzu, sazinieties ar atbalsta dienestu, norādot kļūdas kodu 541.",  
543: "Sistēmas kļūda. Lūdzu, sazinieties ar atbalsta dienestu, norādot kļūdas kodu 543.",  
551: "Sistēmas kļūda. Lūdzu, sazinieties ar atbalsta dienestu, norādot kļūdas kodu 551.",  
505: "Nederīgs spēles stāvoklis. Lūdzu, paziņojiet atbalsta dienestam kļūdas kodu 505.",  
507: "Nederīga likme. Lūdzu, paziņojiet atbalsta dienestam kļūdas kodu 507.",  
508: "Sistēmas komandas kļūda. Lūdzu, paziņojiet atbalsta dienestam kļūdas kodu 508.",  
519: "Nederīga pieteikšanās. Lūdzu, paziņojiet atbalsta dienestam kļūdas kodu 519.",  
520: "Spēle ir aktīva citā pārļūkā.",  
525: "Valūtas kļūda. Lūdzu, paziņojiet atbalsta dienestam kļūdas kodu 525.",  
526: "Pieteikšanās kļūda. Lūdzu, paziņojiet atbalsta dienestam kļūdas kodu 526.",  
537: "Pieteikšanās kļūda. Lūdzu, paziņojiet atbalsta dienestam kļūdas kodu 537.",  
527: "Bezmaksas griezienu kļūda. Lūdzu, paziņojiet atbalsta dienestam kļūdas kodu  
527.",  
532: "Bezmaksas griezienu kļūda. Lūdzu, paziņojiet atbalsta dienestam kļūdas kodu  
532.",  
528: "Nederīgs URL.",  
533: "Žetona kļūda. Lūdzu, paziņojiet atbalsta dienestam kļūdas kodu 533.",  
536: "Žetona kļūda. Lūdzu, paziņojiet atbalsta dienestam kļūdas kodu 536.",  
549: "Žetona kļūda. Lūdzu, paziņojiet atbalsta dienestam kļūdas kodu 549.",  
535: "Spēles versija ir novecojusi.",  
538: "Džekpota kļūda. Lūdzu, paziņojiet atbalsta dienestam kļūdas kodu 538.",  
539: "Džekpota kļūda. Lūdzu, paziņojiet atbalsta dienestam kļūdas kodu 539.",  
540: "Spēlētājam ir aktīva spēle citā serverī.",  
542: "Versijas kļūda. Lūdzu, paziņojiet atbalsta dienestam kļūdas kodu 542.",  
544: "Pārsniegts pieejamais budžets",  
547: "Valūtas kļūda. Lūdzu, paziņojiet atbalsta dienestam kļūdas kodu 547.",  
550: "Sasniegta laimesta maksimālā robežvērtība",  
506: "Nepietiek līdzekļu izvēlētajam likmes lielumam",  
521: "Tehniska kļūda pieteikšanās laikā",  
524: "Spēle atspējota",  
531: "Bloķēta spēlēšanas iespēja",  
534: "Pieteikšanās mēģinājums no bloķēta apgabala",  
545: "Pārsniegts realitātes kontroles zaudējuma ierobežojums",  
546: "Pārsniegts realitātes kontroles sesijas ilgums",  
548: "Playtech realitātes kontroles sesija ir beigusies",  
554: "Blocked from playing - daily time limit exceeded",  
555: "Blocked from playing - weekly time limit exceeded",  
556: "Blocked from playing - monthly time limit exceeded"

```
}  
}, cc._RF.pop()  
}, {}  
nl: [function(e, t, i) {  
  "use strict";
```

```

cc._RF.push(t, "1611019WfdLYaKByet+wtAb", "nl"), t.exports = {
  WELCOME: "Welcome, %{name}!",
  Click_anywhere_on_the_screen_to_start_the_game: "Klik ergens op het scherm om het spel
te starten",
  HUD: {
    BET: "INZET",
    CREDITS: "CREDITS",
    COIN_VALUE: "MUNTWAARDE",
    WIN: "WINST",
    AUTOPLAY: "AUTOPLAY",
    MAXBET: "MAX INZET",
    "576WAYS_TO_WIN": "576 MANIEREN OM TE WINNEN!",
    WIN_UP_TO_128_FREE_SPINS: "WIN TOT 128 GRATIS DRAAIEN",
    FREE: "GRATIS"
  },
  HUD_MESSAGES: {
    EXTEND_MATRIX: "Overschrijdt de matrix!",
    PAY_BOTH_WAYS_MATRIX: "Wordt in beide richtingen uitbetaald en vergroot de
matrix!",
    ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Activeert gemengde
uitbetalingen en vergroot de matrix!",
    IS_ACTIVE_ALL: " Wanneer de Rode Maan actief is, veranderen de personages naar
hun dierlijke vorm en delen winsten (gemengde uitbetalingen)!",
    "3X": "3x",
    DURING_THE_FREE_SPIN: "Tijdens de gratis rondes is het spel altijd 5x5 en betaalt in
beide richtingen uit",
    TRIGGER_6_FREE_SPINS: "Activeren 6 gratis rondes!",
    FREE_SPIN_CAN_RETRIGGER: "Gratis rondes kunnen geheractiveerd worden",
    GOOD_LUCK: "Good Luck!",
    X_FREE_PLAY: "nog %{remainingCount} gratis spellen over",
    X_FREE_PLAY_COMPLETED: "nog gratis spellen over",
    FREE_SPIN_AWARDED: "u hebt %{spinCount} gratis spellen ontvangen",
    FREE_SPIN_OVER: "geen gratis spellen meer",
    NO_THANKS: "nee, bedankt",
    PLAY_LATER: "later spelen",
    PLAY_NOW: "nu spelen"
  },
  PAYTABLE: {
    PAYTABLE: "UITBETALINGSTABEL",
    PAYS_BOTH_WAYS: "Wordt in beide richtingen uitbetaald en vergroot de matrix",
    EXTENDS_MATRIX: "Overschrijdt de matrix",
    ACTIVATES_MIXED: "Activeert gemengde uitbetalingen en vergroot de matrix",
    TRIGGER_6_FREE_SPIN: "Activeren 6 gratis rondes",
    AND_PAYS: "5x5 en betaalt in beide richtingen uit",
    WHEN_THE_RED: "Wanneer de Rode Maan actief is, veranderen de personages naar
hun dierlijke vorm en delen winsten (gemengde uitbetalingen)",
    MALFUNCTION_TEXT: "BIJ STORING ZIJN ALLE SPELRONDES EN WINSTEN
ONGELDIG",
    GAME_RULES: "spelregelsSS",
    GAME_RULES1: "VOLSUNGA IS EEN SPEL MET WINLIJNEN EN EEN GROTER
WORDENDE 53135-, 53335-, 55355- EN 55555-MATRIX",
    GAME_RULES2: "SPELERS HEBBEN 3125 WINLIJNEN",

```

```

    GAME_RULES3: "HET SPEL HEEFT EEN RTP VAN 97,16%",
    GAME_RULES4: "ALLE UITBETALINGEN, BEHALVE SCATTERWINSTEN,
WORDEN VAN LINKS NAAR RECHTS UITBETAALD, BEGINNEND BIJ DE MEEST
LINKSE ROL",
    GAME_RULES5: "SCATTERWINSTEN WORDEN UITBETAALD VOOR
SYMBOLEN IN ELKE POSITIE OP DE ROLLEN",
    GAME_RULES6: "ALLEEN DE GROOTSTE SYMBOOLWINST PER WINNENDE
COMBINATIE WORDT UITBETAALD",
    GAME_RULES7: "ALLEEN DE GROOTSTE SCATTERWINST PER SPEL WORDT
UITBETAALD",
    GAME_RULES8: "SCATTERWINSTEN WORDEN BIJ VAN LIJNWINSTEN
OPGETELD",
    GAME_RULES9: "LIJNWINSTEN WORDEN VERMENIGVULDIGD MET DE
INZETVERMENIGVULDIGER",
    GAME_RULES10: "SCATTERWINSTEN WORDEN VERMENIGVULDIGD MET DE
TOTALE INZET",
    GAME_RULES11: "TIJDENS GRATIS RONDES IS HET SPEL ALTIJD 5X5 EN
BETAALT ALTIJD IN BEIDE RICHTINGEN UIT",
    GAME_RULES12: "GRATIS RONDES KUNNEN OPNIEUW GEACTIVEERD
WORDEN"
},
BETSETTINGS: "INZET INSTELLINGEN",
AUTOPLAY: {
    AUTOPLAY_TITLE: "AUTOPLAY",
    AUTOPLAY_SUBTITLE_1: "Selecteer aantal draaien",
    AUTOPLAY_SUBTITLE_2: "Stop autoplay",
    AUTOPLAY_DES_1: "Als verlies groter is dan",
    AUTOPLAY_DES_2: "Als enkele draai groter is dan",
    CLEAR: "Wissen",
    PLAY: "Spelen",
    WIN_LIMIT_EXCEED: "draai groter is",
    LOSS_LIMIT_EXCEED: "verlies groter is"
},
SOUNDSETTINGS: {
    SOUND: "GELUID",
    BACKGROUND_SOUND: "ACHTERGRONDGELUID"
},
MSG: {
    BIG_WIN: "MOOIE WINST",
    HUGE_WIN: "EXTREME WINST!!",
    EPIC_WIN: "GIGANTISCHE WINST!!",
    MEGA_WIN: "REUSACHTIGE WINST!!",
    CONGRATULATIONS: "GEFELICITEERD",
    CREDITS_WON: "CREDITS GEWONNEN",
    COMBINATION: "COMBINATIE(S)",
    AWARDED: "BELOOND",
    FREE_SPINS: "GRATIS RONDES",
    ADDITIONAL_FREE_SPINS_AWARDED: "EXTRA GRATIS RONDES
TOEGEKEND",
    YOU_HAVE_WON: "U WINT",
    FREE_SPINS_AWARDED: "GRATIS RONDES VERDIEND",
    REALITY_CHECK_LOSS_LIMIT: "Verlies-limiet bij werkelijkheidscontrole

```

```

overschreden",
  REALITY_CHECK_SESSION_LENGTH: "Sessieduur van werkelijkheidscontrole
overschreden",
  CLICK_TO_START: "Klik om te starten",
  SESSION_TIME: "Sessietijd",
  WIN_AMOUNT: "Gewonnen bedrag",
  LOSS_AMOUNT: "Verloren bedrag",
  UNFINISHED_GAME_ROUND: "ONVOLTOOIDE SPELRONDES"
},
BUTTON: {
  OK: "OK",
  CANCEL: "ANNULEREN",
  CONTINUE_PLAYING: "BLIJVEN SPELEN",
  CLEAR: "WISSEN",
  CONTINUE: "DOORGAAN",
  AH: "Accountgeschiedenis",
  RL: "TERUG NAAR LOBBY"
},
ERROR: {
  101: "Ongeldig ticket",
  404: "communicatiefout",
  501: "Systeemfout. Neem contact op met support en vermeld de foutcode 501",
  502: "Systeemfout. Neem contact op met support en vermeld de foutcode 502",
  503: "Økt mangler eller er utløpt",
  504: "Systeemfout. Neem contact op met support en vermeld de foutcode 504",
  509: "Systeemfout. Neem contact op met support en vermeld de foutcode 509",
  516: "Systeemfout. Neem contact op met support en vermeld de foutcode 516",
  517: "Systeemfout. Neem contact op met support en vermeld de foutcode 517",
  518: "Systeemfout. Neem contact op met support en vermeld de foutcode 518",
  522: "Systeemfout. Neem contact op met support en vermeld de foutcode 522",
  523: "Systeemfout. Neem contact op met support en vermeld de foutcode 523",
  541: "Systeemfout. Neem contact op met support en vermeld de foutcode 541",
  543: "Systeemfout. Neem contact op met support en vermeld de foutcode 543",
  551: "Systeemfout. Neem contact op met support en vermeld de foutcode 551",
  505: "Ongeldige spelvoortgang. Neem contact op met de klantenservice onder vermelding
van foutcode 505.",
  507: "Ongeldige inzet. Neem contact op met de klantenservice onder vermelding van
foutcode 507.",
  508: "Systeemopdrachtfout. Neem contact op met de klantenservice onder vermelding
van foutcode 508.",
  519: "Ongeldige login. Neem contact op met de klantenservice onder vermelding van
foutcode 519.",
  520: "Spel is actief in een andere browser.",
  525: "Valutafout. Neem contact op met de klantenservice onder vermelding van foutcode
525.",
  526: "Loginfout. Neem contact op met de klantenservice onder vermelding van foutcode
526.",
  537: "Loginfout. Neem contact op met de klantenservice onder vermelding van foutcode
537.",
  527: "Fout bij gratis rondes. Neem contact op met de klantenservice onder vermelding
van foutcode 527.",
  532: "Fout bij gratis rondes. Neem contact op met de klantenservice onder vermelding

```

```

van foutcode 532.",
    528: "Ongeldige URL.",
    533: "Tokenfout. Neem contact op met de klantenservice onder vermelding van foutcode
533.",
    536: "Tokenfout. Neem contact op met de klantenservice onder vermelding van foutcode
536.",
    549: "Tokenfout. Neem contact op met de klantenservice onder vermelding van foutcode
549.",
    535: "Spelversie te oud.",
    538: "Jackpotfout. Neem contact op met de klantenservice onder vermelding van foutcode
538.",
    539: "Jackpotfout. Neem contact op met de klantenservice onder vermelding van foutcode
539.",
    540: "Speler heeft een actief spel op een andere server.",
    542: "Versiefout. Neem contact op met de klantenservice onder vermelding van foutcode
542.",
    544: "Bestedingsbudget overschreden",
    547: "Valutafout. Neem contact op met de klantenservice onder vermelding van foutcode
547.",
    550: "Winstgrens bereikt",
    506: "--Niet genoeg geld voor gekozen inzet",
    521: "--Technische fout bij login",
    524: "--Spel stopgezet",
    531: "--Geblokkeerd van het spelen",
    534: "--Loginpoging vanuit een geblokkeerde regio",
    545: "--Reality check verlieslimiet overschreden",
    546: "--Reality check sessielengte overschreden",
    548: "--Playtech reality check-sessie verlopen",
    554: "Blocked from playing - daily time limit exceeded",
    555: "Blocked from playing - weekly time limit exceeded",
    556: "Blocked from playing - monthly time limit exceeded"
}
}, cc._RF.pop()
}, {}],
no: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "4902aE/EQ9JXq9SxPHc8N+Q", "no"), t.exports = {
        WELCOME: "Welcome, %{name}!",
        Click_anywhere_on_the_screen_to_start_the_game: "Klikk hvor som helst på skjermen for
å starte spillet",
        HUD: {
            BET: "INNSATS",
            CREDITS: "KREDITTER",
            COIN_VALUE: "MYNTVERDI",
            WIN: "VINN",
            AUTOPLAY: "AUTOSPINN",
            MAXBET: "MAKSINNSATS",
            "576WAYS_TO_WIN": "576 VINNERMÅTER!",
            WIN_UP_TO_128_FREE_SPINS: "VINN OPP TIL 128 GRATIS SPINN",
            FREE: "GRATIS"
        },
        HUD_MESSAGES: {

```

```

EXTEND_MATRIX: "Utvider matrisen",
PAY_BOTH_WAYS_MATRIX: "Betaler begge veier og utvider matrisen",
ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Aktiverer blandede
betalinger og utvider matrisen",
IS_ACTIVE_ALL: " Når den røde månen er aktiv, vil alle figurer forvandle seg til sin
dyreform og dele gevinstene (blandede betalinger)",
"3X": "3x",
DURING_THE_FREE_SPIN: "Under gratisspinnene er spillet alltid 5x5 og betaler alltid
begge veier",
TRIGGER_6_FREE_SPINS: "aktiver 6 gratisspinn",
FREE_SPIN_CAN_RETRIGGER: "Gratisspinnene kan reaktiveres",
GOOD_LUCK: "Good Luck",
X_FREE_PLAY: "%{remainingCount} gratisspill igjen",
X_FREE_PLAY_COMPLETED: "gratisspill igjen",
FREE_SPIN_AWARDED: "du har fått tildelt %{spinCount} gratisspinn",
FREE_SPIN_OVER: "gratisspillene er over",
NO_THANKS: "nei takk",
PLAY_LATER: "spill senere",
PLAY_NOW: "spill nå"
},
PAYTABLE: {
PAYTABLE: "GEVINSTTABELL",
PAYS_BOTH_WAYS: "Betaler begge veier og utvider matrisen",
EXTENDS_MATRIX: "Utvider matrisen",
ACTIVATES_MIXED: "Aktiverer blandede betalinger og utvider matrisen",
TRIGGER_6_FREE_SPIN: "aktiver 6 gratisspinn",
AND_PAYS: "5x5 og betaler begge veier",
WHEN_THE_RED: "Når den røde månen er aktiv, vil alle figurer forvandle seg til sin
dyreform og dele gevinstene (blandede betalinger)",
MALFUNCTION_TEXT: "SYSTEMFEIL ANNULLERER ALLE GEVINSTER OG
SPILL",
GAME_RULES: "SPILLREGLER",
GAME_RULES1: "VOLSUNGA ER ET VINNERVEI-SPILL MED UTVIDENDE 53
135, 53 335, 55 355 OG 55 555-MATRISER",
GAME_RULES2: "SPILLERE HAR 3 125 VINNERVEIER",
GAME_RULES3: "SPILLET HAR EN RTP PÅ 97,16 %",
GAME_RULES4: "ALLE GEVINSTER UNNTATT SPREDTE BETALES VENSTRE-
TIL-HØYRE, MED START FRA HJULET LENGST TIL VENSTRE",
GAME_RULES5: "SPREDTE GEVINSTER BETALES FOR SYMBOLER SOM
DUKKER OPP I ÆNHVER POSISJON PÅ HJULENE",
GAME_RULES6: "KUN DEN HØYESTE SYMBOLGEVINSTEN UTBETALES PER
VINNERKOMBINASJON",
GAME_RULES7: "KUN DEN HØYESTE SPREDTE GEVINSTEN UTBETALES PER
RUNDE",
GAME_RULES8: "SPREDTE GEVINSTER UTBETALES I TILLEGG TIL
GEVINSTVEIENE",
GAME_RULES9: "GEVINSTVEIENE MULTIPLISERES MED
INNSATSMULTIPLIKATOREN",
GAME_RULES10: "SPREDTE GEVINSTER MULTIPLISERES MED
TOTALINNSATSEN I KREDITTER",
GAME_RULES11: "UNDER GRATISSPINNENE ER SPILLET ALLTID 5X5 OG
BETALER ALLTID BEGGE VEIER",

```

```

    GAME_RULES12: "GRATISSPINNENE KAN REAKTIVERES"
  },
  BETSETTINGS: "INNSATSINNSTILLINGER",
  AUTOPLAY: {
    AUTOPLAY_TITLE: "AUTOSPILL",
    AUTOPLAY_SUBTITLE_1: "Velg antall spill",
    AUTOPLAY_SUBTITLE_2: "Stopp autospill",
    AUTOPLAY_DES_1: "Hvis tap overstiger",
    AUTOPLAY_DES_2: "Hvis en enkel gevinst overstiger",
    CLEAR: "Tøm",
    PLAY: "Spill",
    WIN_LIMIT_EXCEED: "gevinst overstiger",
    LOSS_LIMIT_EXCEED: "tap overstiger"
  },
  SOUNDSETTINGS: {
    SOUND: "LYD",
    BACKGROUND_SOUND: "BAKGRUNNSLYDER"
  },
  MSG: {
    BIG_WIN: "HERLIG GEVINST",
    HUGE_WIN: "EKSTREM GEVINST",
    EPIC_WIN: "MASSIV GEVINST",
    MEGA_WIN: "ENORM GEVINST",
    CONGRATULATIONS: "GRATULERER",
    CREDITS_WON: "KREDITTER VUNNET",
    COMBINATION: "KOMBINASJON(ER)",
    AWARDED: "TILDELT",
    FREE_SPINS: "GRATISSPINN",
    ADDITIONAL_FREE_SPINS_AWARDED: "YTTERLIGERE GRATISSPINN
    AKTIVERT",
    YOU_HAVE_WON: "DU HAR VUNNET",
    FREE_SPINS_AWARDED: "GRATISSPINN AKTIVERT",
    REALITY_CHECK_LOSS_LIMIT: "Virkelighetskontrollens tapsgrense overskredet",
    REALITY_CHECK_SESSION_LENGTH: "Virkelighetskontrollens øktlengde
    overskredet",
    CLICK_TO_START: "Trykk for å starte",
    SESSION_TIME: "Økttid",
    WIN_AMOUNT: "Gevinstbeløp",
    LOSS_AMOUNT: "Tapsbeløp",
    UNFINISHED_GAME_ROUND: "UFERDIGE SPILLRUNDER"
  },
  BUTTON: {
    OK: "OK",
    CANCEL: "AVBRYT",
    CONTINUE_PLAYING: "FORTSETT SPILLET",
    CLEAR: "FJERN",
    CONTINUE: "FORTSETT",
    AH: "Kontohistorikk",
    RL: "TILBAKE TIL LOBBY"
  },
  ERROR: {
    101: "Ugyldig billett",

```

```

404: "kommunikasjonsfeil",
501: "Systemfeil. Vennligst kontakt support med feilkoden 501",
502: "Systemfeil. Vennligst kontakt support med feilkoden 502",
503: "Sessie ontbreekt of verlopen",
504: "Systemfeil. Vennligst kontakt support med feilkoden 504",
509: "Systemfeil. Vennligst kontakt support med feilkoden 509",
516: "Systemfeil. Vennligst kontakt support med feilkoden 516",
517: "Systemfeil. Vennligst kontakt support med feilkoden 517",
518: "Systemfeil. Vennligst kontakt support med feilkoden 518",
522: "Systemfeil. Vennligst kontakt support med feilkoden 522",
523: "Systemfeil. Vennligst kontakt support med feilkoden 523",
541: "Systemfeil. Vennligst kontakt support med feilkoden 541",
543: "Systemfeil. Vennligst kontakt support med feilkoden 543",
551: "Systemfeil. Vennligst kontakt support med feilkoden 551",
505: "Ugyldig spillprosess. Vennligst kontakt kundeservice med feilkoden 505.",
507: "Ugyldig innsats. Vennligst kontakt kundeservice med feilkoden 507.",
508: "Feil i systembeskjed. Vennligst kontakt kundeservice med feilkoden 508.",
519: "Ugyldig innlogging. Vennligst kontakt kundeservice med feilkoden 519.",
520: "Spillet er aktivt i en annen nettleser.",
525: "Valutafeil. Vennligst kontakt kundeservice med feilkoden 525.",
526: "Innloggingsfeil. Vennligst kontakt kundeservice med feilkoden 526.",
537: "Innloggingsfeil. Vennligst kontakt kundeservice med feilkoden 537.",
527: "Gratisspinnfeil. Vennligst kontakt kundeservice med feilkoden 527.",
532: "Gratisspinnfeil. Vennligst kontakt kundeservice med feilkoden 532.",
528: "Ugyldig URL.",
533: "Sjetongfeil. Vennligst kontakt kundeservice med feilkoden 533. ",
536: "Sjetongfeil. Vennligst kontakt kundeservice med feilkoden 536. ",
549: "Sjetongfeil. Vennligst kontakt kundeservice med feilkoden 549. ",
535: "Spillversjon for gammel.",
538: "Jackpotfeil. Vennligst kontakt kundeservice med feilkoden 538.",
539: "Jackpotfeil. Vennligst kontakt kundeservice med feilkoden 539.",
540: "Spilleren har et aktivt spill på en annen server.",
542: "Versjonfeil. Vennligst kontakt kundeservice med feilkoden 542.",
544: "Budsjett overskredet",
547: "Valutafeil. Vennligst kontakt kundeservice med feilkoden 547.",
550: "Gevinstgrense oppnådd",
506: "Niet genoeg saldo voor gekozen inzetgrootte",
521: "Technische fout tijdens inloggen",
524: "Spel uitgeschakeld",
531: "Geblokkeerd van spelen",
534: "Inlogpoging komt vanuit een geblokkeerde regio",
545: "Verlies-limiet bij werkelijkheidscontrole overschreden",
546: "Sessieduur van werkelijkheidscontrole overschreden",
548: "Sessie van Playtech-werkelijkheidscontrole verlopen",
554: "Blocked from playing - daily time limit exceeded",
555: "Blocked from playing - weekly time limit exceeded",
556: "Blocked from playing - monthly time limit exceeded"
}
}, cc._RF.pop()
}, {}],
pl: [function(e, t, i) {
"use strict";

```

```

cc._RF.push(t, "a2a9dPH9GhH9aN2qXBKraVp", "pl"), t.exports = {
  WELCOME: "Welcome, %{name}!",
  Click_anywhere_on_the_screen_to_start_the_game: "Kliknij gdziekolwiek , aby rozpocząć
grę",
  HUD: {
    BET: "ZAKŁAD",
    CREDITS: "KREDYTY",
    COIN_VALUE: "WARTOŚĆ MONETY",
    WIN: "WYGRANA",
    AUTOPLAY: "GRA AUTOMATYCZNA",
    MAXBET: "MAKSYMALNY ZAKŁAD",
    "576WAYS_TO_WIN": "576 SPOSOBÓW NA WYGRANĄ!",
    WIN_UP_TO_128_FREE_SPINS: "WYGRAJ DO 128 DARMOWYCH SPINÓW",
    FREE: "DARMOWYCH"
  },
  HUD_MESSAGES: {
    EXTEND_MATRIX: "Rozszerza macierz",
    PAY_BOTH_WAYS_MATRIX: "Wypłaca wygrane w obu kierunkach i rozszerza
macierz",
    ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Aktywuje mieszane
wypłaty i rozszerza macierz",
    IS_ACTIVE_ALL: " Gdy Czerwony Księżyc jest aktywny, wszystkie postacie zamieniają
się w zwierzęta i dzielą wygranymi (mieszane wypłaty)",
    "3X": "3x",
    DURING_THE_FREE_SPIN: "Podczas darmowych obrotów gra ma zawsze format 5 x
5, a wygrane liczy się w obu kierunkach",
    TRIGGER_6_FREE_SPINS: "uruchamia 6 obrotów darmowych",
    FREE_SPIN_CAN_RETRIGGER: "Darmowe obroty można uruchomić ponownie",
    GOOD_LUCK: "Good Luck",
    X_FREE_PLAY: "zostało %{remainingCount} gier darmowych",
    X_FREE_PLAY_COMPLETED: "zostało gier darmowych",
    FREE_SPIN_AWARDED: "otrzymujesz %{spinCount} darmowych gier",
    FREE_SPIN_OVER: "koniec gier darmowych",
    NO_THANKS: "nie, dziękuję",
    PLAY_LATER: "zagraj później",
    PLAY_NOW: "graj teraz"
  },
  PAYTABLE: {
    PAYTABLE: "TABELA WYPŁAT",
    PAYS_BOTH_WAYS: "Wypłaca wygrane w obu kierunkach i rozszerza macierz",
    EXTENDS_MATRIX: "Rozszerza macierz",
    ACTIVATES_MIXED: "Aktywuje mieszane wypłaty i rozszerza macierz",
    TRIGGER_6_FREE_SPIN: "uruchamia 6 obrotów darmowych",
    AND_PAYS: "Format 5 x 5 i premie w obu kierunkach",
    WHEN_THE_RED: "Gdy Czerwony Księżyc jest aktywny, wszystkie postacie
zamieniają się w zwierzęta i dzielą wygranymi (mieszane wypłaty)",
    MALFUNCTION_TEXT: "AWARIA ANULUJE WSZYSTKIE WYPŁATY I
ROZGRYWKI",
    GAME_RULES: "ZASADY GRY",
    GAME_RULES1: "VOLSUNGA TO GRA Z WARIACJAMI WYGRANYCH I
ROZSZERZANĄ MACIERZĄ 53135, 53335, 55355 ORAZ 55555",
    GAME_RULES2: "W GRZE JEST 3125 WARIACJI WYGRANYCH",

```

```
    GAME_RULES3: "WSKAŹNIK ZWROTU DLA GRACZA (RTP) WYNOSI 97,16%",
    GAME_RULES4: "WSZYSTKIE WYPŁATY Z WYJĄTKIEM WYPŁAT SCATTER
SĄ PRZYZNAWANE OD LEWEJ DO PRAWEJ, ZACZYNAJĄC OD PIERWSZEGO BĘBNA
PO LEWEJ",
    GAME_RULES5: "WYPŁATY SCATTER SĄ PRZYZNAWANE ZA SYMBOLE
POJAWIAJĄCE SIĘ W DOWOLNEJ POZYCJI NA BĘBNACH",
    GAME_RULES6: "WYPŁACANA JEST TYLKO NAJWYŻSZA WYGRANA ZA
SYMBOLE NA WYGRYWAJĄCĄ KOMBINACJĘ",
    GAME_RULES7: "WYPŁACANA JEST TYLKO NAJWYŻSZA WYGRANA
SCATTER NA GRĘ",
    GAME_RULES8: "WYPŁATY SCATTER SĄ PRZYZNAWANE DODATKOWO DO
WYPŁAT WARIACJI",
    GAME_RULES9: "WYPŁATY WARIACJI SĄ MNOŻONE PRZEZ MNOŻNIK
ZAKŁADU",
    GAME_RULES10: "WYPŁATY SCATTER SĄ MNOŻONE PRZEZ CAŁKOWITĄ
KWOTĘ ZAKŁADU W KREDYTACH",
    GAME_RULES11: "PODCZAS OBROTÓW DARMOWYCH GRA MA ZAWSZE
FORMAT 5 NA 5 I WYPŁACA WYGRANE W OBU KIERUNKACH",
    GAME_RULES12: "OBROTY DARMOWE MOGĄ ZOSTAĆ URUCHOMIONE
PONOWNIE"
},
BETSETTINGS: "USTAWIENIA ZAKŁADU",
AUTOPLAY: {
    AUTOPLAY_TITLE: "GRA AUTOMATYCZNA",
    AUTOPLAY_SUBTITLE_1: "Wybierz liczbę spinów",
    AUTOPLAY_SUBTITLE_2: "Zatrzymaj grę automatyczną",
    AUTOPLAY_DES_1: "Gdy przegrana przekracza",
    AUTOPLAY_DES_2: "Gdy pojedyncza wygrana przekracza",
    CLEAR: "Wyczyść",
    PLAY: "Graj",
    WIN_LIMIT_EXCEED: "wygrana przekracza",
    LOSS_LIMIT_EXCEED: "przegrana przekracza"
},
SOUNDSETTINGS: {
    SOUND: "DŹWIĘK",
    BACKGROUND_SOUND: "DŹWIĘK W TLE"
},
MSG: {
    BIG_WIN: "NIEZŁA WYGRANA",
    HUGE_WIN: "EKSTREMALNA WYGRANA",
    EPIC_WIN: "MASYWNA WYGRANA",
    MEGA_WIN: "OGROMNA WYGRANA",
    CONGRATULATIONS: "GRATULACJE",
    CREDITS_WON: "WYGRANYCH KREDYTÓW",
    COMBINATION: "KOMBINACJI",
    AWARDED: "NAGRODZONYCH",
    FREE_SPINS: "OBROTY DARMOWE",
    ADDITIONAL_FREE_SPINS_AWARDED: "PRZYZNANO DODATKOWE
DARMOWE OBROTY",
    YOU_HAVE_WON: "WYGRYWASZ",
    FREE_SPINS_AWARDED: "WYGRANO OBROTY DARMOWE",
    REALITY_CHECK_LOSS_LIMIT: "Przekroczono limit odpowiedzialnej gry dotyczący
```

strat",

REALITY\_CHECK\_SESSION\_LENGTH: "Przekroczono limit odpowiedzialnej gry dotyczący długości sesji",

CLICK\_TO\_START: "Kliknij, żeby zacząć",

SESSION\_TIME: "Czas sesji",

WIN\_AMOUNT: "Kwota wygranej",

LOSS\_AMOUNT: "Kwota przegranej",

UNFINISHED\_GAME\_ROUND: "NIEUKOŃCZONE RUNDY GRY"

},

BUTTON: {

OK: "OK",

CANCEL: "ANULUJ",

CONTINUE\_PLAYING: "KONTYNUUJ GRĘ",

CLEAR: "WYCZYŚĆ",

CONTINUE: "KONTYNUUJ",

AH: "Historia konta",

RL: "WRÓĆ DO LOBBY"

},

ERROR: {

101: "Nieprawidłowy bilet",

404: "Błąd komunikacji",

501: "Błąd systemu. Skontaktuj się z działem pomocy, podając kod błędu 501",

502: "Błąd systemu. Skontaktuj się z działem pomocy, podając kod błędu 502",

503: "Sesja nie istnieje lub wygasła",

504: "Błąd systemu. Skontaktuj się z działem pomocy, podając kod błędu 504",

509: "Błąd systemu. Skontaktuj się z działem pomocy, podając kod błędu 509",

516: "Błąd systemu. Skontaktuj się z działem pomocy, podając kod błędu 516",

517: "Błąd systemu. Skontaktuj się z działem pomocy, podając kod błędu 517",

518: "Błąd systemu. Skontaktuj się z działem pomocy, podając kod błędu 518",

522: "Błąd systemu. Skontaktuj się z działem pomocy, podając kod błędu 522",

523: "Błąd systemu. Skontaktuj się z działem pomocy, podając kod błędu 523",

541: "Błąd systemu. Skontaktuj się z działem pomocy, podając kod błędu 541",

543: "Błąd systemu. Skontaktuj się z działem pomocy, podając kod błędu 543",

551: "Błąd systemu. Skontaktuj się z działem pomocy, podając kod błędu 551",

505: "Nieprawidłowe postępy w grze. Skontaktuj się z działem pomocy, podając kod błędu 505.",

507: "Nieprawidłowy zakład. Skontaktuj się z działem pomocy, podając kod błędu 507.",

508: "Błąd polecenia systemowego. Skontaktuj się z działem pomocy, podając kod błędu 508.",

519: "Nieprawidłowe logowanie. Skontaktuj się z działem pomocy, podając kod błędu 519.",

520: "Gra jest aktywna w innej przeglądarce.",

525: "Błąd waluty. Skontaktuj się z działem pomocy, podając kod błędu 525.",

526: "Błąd logowania. Skontaktuj się z działem pomocy, podając kod błędu 526.",

537: "Błąd logowania. Skontaktuj się z działem pomocy, podając kod błędu 537.",

527: "Błąd darmowego obrotu. Skontaktuj się z działem pomocy, podając kod błędu 527.",

532: "Błąd darmowego obrotu. Skontaktuj się z działem pomocy, podając kod błędu 532.",

528: "Nieprawidłowy adres URL.",

533: "Błąd żetonu. Skontaktuj się z działem pomocy, podając kod błędu 533.",

536: "Błąd żetonu. Skontaktuj się z działem pomocy, podając kod błędu 536.",

549: "Błąd żetonu. Skontaktuj się z działem pomocy, podając kod błędu 549.",  
 535: "Wersja gry jest zbyt stara. ",  
 538: "Błąd jackpota. Skontaktuj się z działem pomocy, podając kod błędu 538.",  
 539: "Błąd jackpota. Skontaktuj się z działem pomocy, podając kod błędu 539.",  
 540: "Gracz ma aktywną grę na innym serwerze.",  
 542: "Błąd wersji. Skontaktuj się z działem pomocy, podając kod błędu 542.",  
 544: "Przekroczono budżet wydatków",  
 547: "Błąd waluty. Skontaktuj się z działem pomocy, podając kod błędu 547.",  
 550: "Osiągnięto limit wygranych",  
 506: "Brak wystarczających środków dla wybranego zakładu",  
 521: "Błąd techniczny podczas logowania",  
 524: "Gra wyłączona",  
 531: "Zablokowano możliwość gry",  
 534: "Próba zalogowania z zablokowanego regionu",  
 545: "Przekroczono limit odpowiedzialnej gry dotyczący strat",  
 546: "Przekroczono limit odpowiedzialnej gry dotyczący długości sesji",  
 548: "Sesja odpowiedzialnej gry Playtech wygasła",  
 554: "Blocked from playing - daily time limit exceeded",  
 555: "Blocked from playing - weekly time limit exceeded",  
 556: "Blocked from playing - monthly time limit exceeded"

```

    }
  }, cc._RF.pop()
}, {}],
polyglot: [function(e, t, i) {
  (function(e) {
    "use strict";
    cc._RF.push(t, "bdf15pziSlC/KXaYhvhrOs1", "polyglot");
    var n,
        a,
        o = "function" === typeof Symbol && "symbol" === typeof Symbol.iterator ? function(e) {
          return typeof e
        } : function(e) {
          return e && "function" === typeof Symbol && e.constructor === Symbol && e !==
Symbol.prototype ? "symbol" : typeof e
        };
    n = void 0 !== e ? e : void 0, a = function(e) {
      var t = String.prototype.replace;
      function i(e) {
        e = e || {}, this.phrases = {}, this.extend(e.phrases || {}), this.currentLocale = e.locale ||
"en", this.allowMissing = !!e.allowMissing, this.warn = e.warn || d
      }
      i.VERSION = "1.0.0", i.prototype.locale = function(e) {
        return e && (this.currentLocale = e), this.currentLocale
      }, i.prototype.extend = function(e, t) {
        var i;
        for (var n in e)
          e.hasOwnProperty(n) && (i = e[n], t && (n = t + "." + n), "object" === (void 0 ===
i ? "undefined" : o(i)) ? this.extend(i, n) : this.phrases[n] = i)
      }, i.prototype.unset = function(e, t) {
        var i;
        if ("string" === typeof e)
          delete this.phrases[e];

```

```

else
  for (var n in e)
    e.hasOwnProperty(n) && (i = e[n], t && (n = t + "." + n), "object" === (void 0
=== i ? "undefined" : o(i)) ? this.unset(i, n) : delete this.phrases[n])
  }, i.prototype.clear = function() {
    this.phrases = {}
  }, i.prototype.replace = function(e) {
    this.clear(), this.extend(e)
  }, i.prototype.t = function(e, i) {
    var o,
        d;
    return "number" === typeof (i = null === i ? {} : i) && (i = {
      smart_count: i
    }), "string" === typeof this.phrases[e] ? o = this.phrases[e] : "string" === typeof i._ ? o =
i._ : this.allowMissing ? o = e : (this.warn("Missing translation for key: '" + e + "'"), d = e), "string"
=== typeof o && (i = function(e) {
    var t = {};
    for (var i in e)
      t[i] = e[i];
    return t
  })(i), d = function(e, i) {
    for (var n in i)
      if ("_" !== n && i.hasOwnProperty(n)) {
        var a = i[n];
        "string" === typeof a && (a = t.call(i[n], l, c)), e = t.call(e, new RegExp("%\\{"
+ n + "\\}", "g"), a)
      }
    return e
  }(d = function(e, i, o) {
    var l,
        c,
        d;
    null !== o && e ? (c = e.split(n), d = c[(S = i, E = o, a[_ = S, p = function(e) {
      var t,
          i,
          n,
          a = {};
      for (t in e)
        if (e.hasOwnProperty(t))
          for (n in i = e[t])
            a[i[n]] = t;
      return a
    }](s), p[_ || p.en])(E))] || c[0], u = d, l = t.call(u, r, "")) : l = e;
    var u;
    var S,
        E,
        _;
    p;
    return l
  })(o, this.currentLocale, i.smart_count, i)), d
  }, i.prototype.has = function(e) {
    return e in this.phrases
  }

```

```

};
var n = "||||",
    a = {
      chinese: function(e) {
        return 0
      },
      german: function(e) {
        return 1 !== e ? 1 : 0
      },
      french: function(e) {
        return e > 1 ? 1 : 0
      },
      russian: function(e) {
        return e % 10 === 1 && e % 100 !== 11 ? 0 : e % 10 >= 2 && e % 10 <= 4 && (e
% 100 < 10 || e % 100 >= 20) ? 1 : 2
      },
      czech: function(e) {
        return 1 === e ? 0 : e >= 2 && e <= 4 ? 1 : 2
      },
      polish: function(e) {
        return 1 === e ? 0 : e % 10 >= 2 && e % 10 <= 4 && (e % 100 < 10 || e % 100 >=
20) ? 1 : 2
      },
      icelandic: function(e) {
        return e % 10 !== 1 || e % 100 === 11 ? 1 : 0
      }
    },
    s = {
      chinese: ["fa", "id", "ja", "ko", "lo", "ms", "th", "tr", "zh"],
      german: ["da", "de", "en", "es", "fi", "el", "he", "hu", "it", "nl", "no", "pt", "sv"],
      french: ["fr", "tl", "pt-br"],
      russian: ["hr", "ru"],
      czech: ["cs", "sk"],
      polish: ["pl"],
      icelandic: ["is"]
    };
var r = /^s+|s+$/g;
var l = ^$/g,
    c = "$$$$";
function d(t) {
  e.console && e.console.warn && e.console.warn("WARNING: " + t)
}
return i
}, "function" === typeof define && define.amd ? define([], function() {
  return a(n)
}) : "object" === (void 0 === i ? "undefined" : o(i)) ? t.exports = a(n) : n.Polyglot = a(n),
cc._RF.pop()
}).call(this, "undefined" !== typeof global ? global : "undefined" !== typeof self ? self :
"undefined" !== typeof window ? window : {})
}, {}],
"pt-br": [function(e, t, i) {
  "use strict";

```

```
cc._RF.push(t, "ebc28HEc5xEeYolQpP9Jw6+", "pt-br"), t.exports = {
  WELCOME: "Welcome, %{name}!",
  Click_anywhere_on_the_screen_to_start_the_game: "Clique em qualquer lugar na tela para
começar o jogo",
  HUD: {
    BET: "APOSTA",
    CREDITS: "CRÉDITOS",
    COIN_VALUE: "VALOR DA MOEDA",
    WIN: "GANHOU",
    AUTOPLAY: "JOGO AUTOMÁTICO",
    MAXBET: "APOSTA MÁXIMA",
    "576WAYS_TO_WIN": "576 MANEIRAS DE GANHAR!",
    WIN_UP_TO_128_FREE_SPINS: "GANHE ATÉ 128 GIROS GRÁTIS",
    FREE: "GRÁTIS"
  },
  HUD_MESSAGES: {
    EXTEND_MATRIX: "Estende a matriz",
    PAY_BOTH_WAYS_MATRIX: "Paga nas duas direções e estende a matriz",
    ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Ativa pagamentos mistos e
estende a matriz",
    IS_ACTIVE_ALL: " Quando a Lua Vermelha está ativa, todos os personagens se
transformam em sua forma animal e compartilham prêmios (pagamentos mistos)",
    "3X": "3x",
    DURING_THE_FREE_SPIN: "Durante as partidas grátis, o jogo é sempre 5x5 e paga nas
duas direções",
    TRIGGER_6_FREE_SPINS: "ativa 6 partidas grátis",
    FREE_SPIN_CAN_RETRIGGER: "As partidas grátis podem ser reativadas",
    GOOD_LUCK: "Good Luck",
    X_FREE_PLAY: "%{remainingCount} partidas grátis restantes",
    X_FREE_PLAY_COMPLETED: "partidas grátis restantes",
    FREE_SPIN_AWARDED: "você recebeu %{spinCount} partidas grátis",
    FREE_SPIN_OVER: "fim das partidas grátis",
    NO_THANKS: "não, obrigado",
    PLAY_LATER: "jogar depois",
    PLAY_NOW: "jogar agora"
  },
  PAYTABLE: {
    PAYTABLE: "TABELA DE PAGAMENTO",
    PAYS_BOTH_WAYS: "Paga nas duas direções e estende a matriz",
    EXTENDS_MATRIX: "Estende a matriz",
    ACTIVATES_MIXED: "Ativa pagamentos mistos e estende a matriz",
    TRIGGER_6_FREE_SPIN: "ativa 6 partidas grátis",
    AND_PAYS: "5x5 e paga nas duas direções",
    WHEN_THE_RED: "Quando a Lua Vermelha está ativa, todos os personagens se
transformam em sua forma animal e compartilham prêmios (pagamentos mistos)",
    MALFUNCTION_TEXT: "Malfunction voids all pays and plays",
    GAME_RULES: "REGRAS DO JOGO",
    GAME_RULES1: "O VOLSUNGA É UM JOGO COM FORMAS DE PAGAMENTO,
QUE POSSUI UMA MATRIZ EXTENSIVA DE 53135, 53335, 55355 E 55555",
    GAME_RULES2: "OS JOGADORES POSSUEM 3125 FORMAS DE GANHAR",
    GAME_RULES3: "O JOGO POSSUI UM RETORNO AO JOGADOR DE 97,16%",
    GAME_RULES4: "TODAS AS COMBINAÇÕES, EXCETO SCATTER, PAGAM DA
```

ESQUERDA PARA A DIREITA, A PARTIR DO PRIMEIRO CILINDRO À ESQUERDA",  
 GAME\_RULES5: "OS PRÊMIOS DE SCATTER SÃO PAGOS PARA SÍMBOLOS  
 QUE APARECEM EM QUALQUER POSIÇÃO DOS CILINDROS",  
 GAME\_RULES6: "APENAS O SÍMBOLO COM MAIOR PAGAMENTO É PAGO  
 POR COMBINAÇÃO GANHADORA",  
 GAME\_RULES7: "APENAS O PRÊMIO DE SCATTER COM MAIOR PAGAMENTO  
 É PAGO POR PARTIDA",  
 GAME\_RULES8: "PRÊMIOS DE SCATTER SÃO SOMADOS AOS PRÊMIOS DAS  
 LINHAS",  
 GAME\_RULES9: "PRÊMIOS DAS FORMAS SÃO MULTIPLICADOS PELO  
 MULTIPLICADOR DA APOSTA",  
 GAME\_RULES10: "PRÊMIOS DE SCATTER SÃO MULTIPLICADOS PELA  
 APOSTA TOTAL EM CRÉDITOS",  
 GAME\_RULES11: "DURANTE AS PARTIDAS GRÁTIS, O JOGO É SEMPRE 5X5 E  
 PAGA NAS DUAS DIREÇÕES",  
 GAME\_RULES12: "AS PARTIDAS GRÁTIS PODEM SER REATIVADAS"  
 },  
 BETSETTINGS: "CONFIGURAÇÕES DE APOSTA",  
 AUTOPLAY: {  
 AUTOPLAY\_TITLE: "JOGADA AUTOMÁTICA",  
 AUTOPLAY\_SUBTITLE\_1: "Selecione o número de giros",  
 AUTOPLAY\_SUBTITLE\_2: "Parar a jogada automática",  
 AUTOPLAY\_DES\_1: "Se a perda exceder",  
 AUTOPLAY\_DES\_2: "Se uma única vitória exceder",  
 CLEAR: "Limpar",  
 PLAY: "Jogar",  
 WIN\_LIMIT\_EXCEED: "vitória exceder",  
 LOSS\_LIMIT\_EXCEED: "perda exceder"  
 },  
 SOUNDSETTINGS: {  
 SOUND: "EFEITOS SONOROS",  
 BACKGROUND SOUND: "SOM DE FUNDO"  
 },  
 MSG: {  
 BIG\_WIN: "BOM GANHO",  
 HUGE\_WIN: "VITÓRIA EXTREMA",  
 EPIC\_WIN: "VITÓRIA MASSIVA",  
 MEGA\_WIN: "VITÓRIA ENORME",  
 CONGRATULATIONS: "PARABÉNS",  
 CREDITS\_WON: "CRÉDITOS GANHOS",  
 COMBINATION: "COMBINAÇÃO(ÕES)",  
 AWARDED: "CONCEDIDO",  
 FREE\_SPINS: "PARTIDAS GRÁTIS",  
 ADDITIONAL\_FREE\_SPINS\_AWARDED: "PARTIDAS GRÁTIS ADICIONAIS  
 CONCEDIDAS",  
 YOU\_HAVE\_WON: "VOCÊ GANHOU",  
 FREE\_SPINS\_AWARDED: "PARTIDAS GRÁTIS CONCEDIDAS",  
 REALITY\_CHECK\_LOSS\_LIMIT: "Limite de perdas definido excedido",  
 REALITY\_CHECK\_SESSION\_LENGTH: "Duração de sessão definida excedida",  
 CLICK\_TO\_START: "Clique para iniciar",  
 SESSION\_TIME: "Tempo da sessão",  
 WIN\_AMOUNT: "Montante de ganho",

```
LOSS_AMOUNT: "Montante de perda",
UNFINISHED_GAME_ROUND: "RONDA DE JOGO POR TERMINAR"
},
BUTTON: {
    OK: "OK",
    CANCEL: "CANCELAR",
    CONTINUE_PLAYING: "CONTINUAR JOGANDO!",
    CLEAR: "LIMPAR",
    CONTINUE: "CONTINUAR",
    AH: "Histórico da conta",
    RL: "VOLTAR AO LOBBY"
},
ERROR: {
    101: "Bilhete inválido",
    404: "Erro de comunicação",
    501: "Erro do sistema. Contate o suporte com o código do erro 501",
    502: "Erro do sistema. Contate o suporte com o código do erro 502",
    503: "Sessão ausente ou expirada",
    504: "Erro do sistema. Contate o suporte com o código do erro 504",
    509: "Erro do sistema. Contate o suporte com o código do erro 509",
    516: "Erro do sistema. Contate o suporte com o código do erro 516",
    517: "Erro do sistema. Contate o suporte com o código do erro 517",
    518: "Erro do sistema. Contate o suporte com o código do erro 518",
    522: "Erro do sistema. Contate o suporte com o código do erro 522",
    523: "Erro do sistema. Contate o suporte com o código do erro 523",
    541: "Erro do sistema. Contate o suporte com o código do erro 541",
    543: "Erro do sistema. Contate o suporte com o código do erro 543",
    551: "Erro do sistema. Contate o suporte com o código do erro 551",
    505: "Progresso de jogo inválido. Contate o suporte com o código de erro 505.",
    507: "Aposta inválida. Contate o suporte com o código de erro 507.",
    508: "Erro de comando do sistema. Contate o suporte com o código de erro 508.",
    519: "Login inválido. Contate o suporte com o código de erro 519.",
    520: "O jogo está ativo em outro navegador.",
    525: "Erro de moeda. Contate o suporte com o código de erro 525.",
    526: "Erro de login. Contate o suporte com o código de erro 526.",
    537: "Erro de login. Contate o suporte com o código de erro 537.",
    527: "Erro de partidas grátis. Contate o suporte com o código de erro 527.",
    532: "Erro de partidas grátis. Contate o suporte com o código de erro 532.",
    528: "URL inválida",
    533: "Erro de token. Contate o suporte com o código de erro 533.",
    536: "Erro de token. Contate o suporte com o código de erro 536.",
    549: "Erro de token. Contate o suporte com o código de erro 549.",
    535: "A versão do jogo é muito antiga.",
    538: "Erro de jackpot. Contate o suporte com o código de erro 538.",
    539: "Erro de jackpot. Contate o suporte com o código de erro 539.",
    540: "O jogador possui um jogo ativo em outro servidor.",
    542: "Erro de versão. Contate o suporte com o código de erro 542. ",
    544: "Limite de gastos excedido",
    547: "Erro de moeda. Contate o suporte com o código de erro 547.",
    550: "Limite de ganhos atingido",
    506: "Fundos insuficientes para aposta escolhida",
    521: "Erro técnico durante o acesso",
```

```

524: "Jogo desativado",
531: "Bloqueado do jogo ",
534: "Tentativa de acesso proveniente de região bloqueada",
545: "Limite de perdas definido excedido",
546: "Duração de sessão definida excedida",
548: "Sessão definida da Playtech expirada",
554: "Blocked from playing - daily time limit exceeded",
555: "Blocked from playing - weekly time limit exceeded",
556: "Blocked from playing - monthly time limit exceeded"
}
}, cc._RF.pop()
}, {}],
pt: [function(e, t, i) {
"use strict";
cc._RF.push(t, "ea4c9FVDpxJ0rF4FMh8Cc1F", "pt"), t.exports = {
WELCOME: "Welcome, %{name}!",
Click_anywhere_on_the_screen_to_start_the_game: "Pressione em qualquer do écran para
iniciar o jogo",
HUD: {
BET: "APOSTA",
CREDITS: "CRÉDITOS",
COIN_VALUE: "VALOR DA MOEDA",
WIN: "GANHO",
AUTOPLAY: "JOGO AUTOMÁTICO",
MAXBET: "APOSTA MÁXIMA",
"576WAYS_TO_WIN": "576 MODOS DE GANHAR!",
WIN_UP_TO_128_FREE_SPINS: "GANHE ATÉ 128 VOLTAS GRÁTIS",
FREE: "GRÁTIS"
},
HUD_MESSAGES: {
EXTEND_MATRIX: "Expande a matriz",
PAY_BOTH_WAYS_MATRIX: "Paga em ambos os sentidos e expande a matriz",
ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Ativa os pagamentos
mistos e expande a matriz",
IS_ACTIVE_ALL: " Quando a Lua Vermelha está ativa, todas as personagens se
transformam em animais e partilham prémios (pagamentos mistos)",
"3X": "3x",
DURING_THE_FREE_SPIN: "Durante as rodadas grátis, o jogo é sempre 5x5 e paga
sempre em ambos os sentidos",
TRIGGER_6_FREE_SPINS: "ativa 6 rodadas grátis",
FREE_SPIN_CAN_RETRIGGER: "As rodadas grátis podem ser reativadas",
GOOD_LUCK: "Good Luck",
X_FREE_PLAY: "%{remainingCount} jogadas grátis restantes",
X_FREE_PLAY_COMPLETED: "jogadas grátis restantes",
FREE_SPIN_AWARDED: "recebeu %{spinCount} jogadas grátis",
FREE_SPIN_OVER: "as jogadas grátis terminaram",
NO_THANKS: "não, obrigado",
PLAY_LATER: "jogar depois",
PLAY_NOW: "jogar agora"
},
PAYTABLE: {
PAYTABLE: "TABELA DE PRÉMIOS",

```

PAYS\_BOTH\_WAYS: "Paga em ambos os sentidos e expande a matriz",  
 EXTENDS\_MATRIX: "Expande a matriz",  
 ACTIVATES\_MIXED: "Ativa os pagamentos mistos e expande a matriz",  
 TRIGGER\_6\_FREE\_SPIN: "ativa 6 rodadas grátis",  
 AND\_PAYS: "5x5 e paga em ambos os sentidos",  
 WHEN\_THE\_RED: "Quando a Lua Vermelha está ativa, todas as personagens se transformam em animais e partilham prémios (pagamentos mistos)",  
 MALFUNCTION\_TEXT: "AS AVARIAS ANULAM TODAS AS JOGADAS E PAGAMENTOS",  
 GAME\_RULES: "REGRAS DO JOGO",  
 GAME\_RULES1: "O VOLSUNGA É UM JOGO DE FORMAS COM MATRIZ EXPANSÍVEL 53135, 53335, 55355 E 55555",  
 GAME\_RULES2: "OS JOGADORES TÊM 3125 FORMAS DE GANHAR",  
 GAME\_RULES3: "O JOGO TEM UM RTP DE 97,16%",  
 GAME\_RULES4: "TODOS OS PAGAMENTOS, EXCETO PAGAMENTOS SCATTER, SÃO FEITOS DA ESQUERDA PARA A DIREITA, A COMEÇAR NO CILINDRO MAIS À ESQUERDA",  
 GAME\_RULES5: "OS PAGAMENTOS SCATTER SÃO PAGOS PARA SÍMBOLOS QUE APAREÇAM EM QUALQUER POSIÇÃO NOS CILINDROS",  
 GAME\_RULES6: "APENAS O PRÉMIO DE SÍMBOLO MAIS ELEVADO SERÁ PAGO POR COMBINAÇÃO VENCEDORA",  
 GAME\_RULES7: "APENAS O PRÉMIO SCATTER MAIS ELEVADO SERÁ PAGO POR JOGADA",  
 GAME\_RULES8: "OS PAGAMENTOS SCATTER SÃO PAGOS ADICIONALMENTE AOS PAGAMENTOS DAS FORMAS",  
 GAME\_RULES9: "AS FORMAS SÃO MULTIPLICADAS PELO MÚLTIPLO DA APOSTA",  
 GAME\_RULES10: "OS PAGAMENTOS SCATTER SÃO MULTIPLICADOS PELA APOSTA DE CRÉDITOS TOTAL",  
 GAME\_RULES11: "DURANTE AS RODADAS GRÁTIS, O JOGO É SEMPRE DE 5X5 E PAGA SEMPRE EM AMBOS OS SENTIDOS",  
 GAME\_RULES12: "AS RODADAS GRÁTIS PODEM REATIVAR"

},  
 BETSETTINGS: "CONFIGURAÇÕES DE APOSTA",  
 AUTOPLAY: {  
 AUTOPLAY\_TITLE: "JOGO AUTOMÁTICO",  
 AUTOPLAY\_SUBTITLE\_1: "Seleccione o número de voltas",  
 AUTOPLAY\_SUBTITLE\_2: "Parar jogo automático",  
 AUTOPLAY\_DES\_1: "Se perda exceder",  
 AUTOPLAY\_DES\_2: "Se única vitória exceder",  
 CLEAR: "Limpar",  
 PLAY: "Jogar",  
 WIN\_LIMIT\_EXCEED: "vitória exceder",  
 LOSS\_LIMIT\_EXCEED: "perda exceder"
 }
 },  
 SOUNDSETTINGS: {  
 SOUND: "SOM",  
 BACKGROUND\_SOUND: "SOM DE FUNDO"
 },  
 },  
 MSG: {  
 BIG\_WIN: "BOM GANHO",  
 HUGE\_WIN: "GANHO EXTREMO",
 }

```
    EPIC_WIN: "GANHO MASSIVO",
    MEGA_WIN: "ENORME GANHO",
    CONGRATULATIONS: "PARABÉNS",
    CREDITS_WON: "CRÉDITOS GANHOS",
    COMBINATION: "COMBINAÇÃO(ÕES)",
    AWARDED: "PREMIADOS",
    FREE_SPINS: "RODADAS GRÁTIS",
    ADDITIONAL_FREE_SPINS_AWARDED: "RODADAS GRÁTIS ADICIONAIS
ATRIBUÍDAS",
    YOU_HAVE_WON: "GANHOU",
    FREE_SPINS_AWARDED: "RODADAS GRÁTIS ATRIBUÍDAS",
    REALITY_CHECK_LOSS_LIMIT: "Limite de perdas da verificação de realidade
excedido",
    REALITY_CHECK_SESSION_LENGTH: "Duração de sessão da verificação de
realidade excedido",
    CLICK_TO_START: "Clique para começar",
    SESSION_TIME: "Tempo da sessão",
    WIN_AMOUNT: "Montante de ganho",
    LOSS_AMOUNT: "Montante de perda",
    UNFINISHED_GAME_ROUND: "RONDA DE JOGO POR TERMINAR"
},
BUTTON: {
    OK: "OK",
    CANCEL: "CANCELAR",
    CONTINUE_PLAYING: "CONTINUAR A JOGAR",
    CLEAR: "APAGAR",
    CONTINUE: "CONTINUAR",
    AH: "Histórico da conta",
    RL: "VOLTAR AO LOBBY"
},
ERROR: {
    101: "Bilhete inválido",
    404: "Erro de comunicação",
    501: "Erro do sistema. Contacte a assistência com o código de erro 501. ",
    502: "Erro do sistema. Contacte a assistência com o código de erro 502. ",
    503: "Sessão em falta ou expirada",
    504: "Erro do sistema. Contacte a assistência com o código de erro 504. ",
    509: "Erro do sistema. Contacte a assistência com o código de erro 509. ",
    516: "Erro do sistema. Contacte a assistência com o código de erro 516. ",
    517: "Erro do sistema. Contacte a assistência com o código de erro 517. ",
    518: "Erro do sistema. Contacte a assistência com o código de erro 518. ",
    522: "Erro do sistema. Contacte a assistência com o código de erro 522. ",
    523: "Erro do sistema. Contacte a assistência com o código de erro 523. ",
    541: "Erro do sistema. Contacte a assistência com o código de erro 541. ",
    543: "Erro do sistema. Contacte a assistência com o código de erro 543. ",
    551: "Erro do sistema. Contacte a assistência com o código de erro 551. ",
    505: "Progresso de jogo inválido. Contacte o suporte com o código de erro 505. ",
    507: "Aposta inválida. Contacte o suporte com o código de erro 507.",
    508: "Erro de comando do sistema. Contacte o suporte com o código de erro 508.",
    519: "Início de sessão inválido. Contacte o suporte com o código de erro 519.",
    520: "O jogo está ativo noutra navegador. ",
    525: "Erro de moeda. Contacte o suporte com o código de erro 525.",
```

```

526: "Erro de início de sessão. Contacte o suporte com o código de erro 526.",
537: "Erro de início de sessão. Contacte o suporte com o código de erro 537.",
527: "Erro de rodada grátis. Contacte o suporte com o código de erro 527.",
532: "Erro de rodada grátis. Contacte o suporte com o código de erro 532.",
528: "URL inválid",
533: "Erro de token. Contacte o suporte com o código de erro 533.",
536: "Erro de token. Contacte o suporte com o código de erro 536.",
549: "Erro de token. Contacte o suporte com o código de erro 549.",
535: "O jogo é demasiado antigo.",
538: "Erro de jackpot. Contacte o suporte com o código de erro 538.",
539: "Erro de jackpot. Contacte o suporte com o código de erro 539.",
540: "O jogador tem um jogo ativo noutra servidor.",
542: "Erro de versão. Contacte o suporte com o código de erro 542.",
544: "Orçamento de gastos excedido",
547: "Erro de moeda. Contacte o suporte com o código de erro 547.",
550: "Limite de ganhos atingido",
506: "Sem fundos suficientes para o valor de aposta escolhido",
521: "Erro técnico durante o início de sessão",
524: "Jogo desativado",
531: "Jogo bloqueado",
534: "As tentativas de início de sessão são provenientes de uma região bloqueada",
545: "Limite de perdas da verificação de realidade excedido",
546: "Duração de sessão da verificação de realidade excedido",
548: "A sessão de verificação de realidade da Playtech expirou",
554: "Blocked from playing - daily time limit exceeded",
555: "Blocked from playing - weekly time limit exceeded",
556: "Blocked from playing - monthly time limit exceeded"
}
}, cc._RF.pop()
}, {}],
ro: [function(e, t, i) {
"use strict";
cc._RF.push(t, "4c760D330NOVJ2WZA5sWuUK", "ro"), t.exports = {
WELCOME: "Welcome, %{name}!",
Click_anywhere_on_the_screen_to_start_the_game: "Apasă oriunde pe ecran pentru a
începe jocul ",
HUD: {
BET: "PARIU",
CREDITS: "CREDITE",
COIN_VALUE: "VALOARE MONEDĂ",
WIN: "CÂȘTIGĂ",
AUTOPLAY: "JOC AUTOMAT",
MAXBET: "PARIU MAX.",
"576WAYS_TO_WIN": "576 MODURI DE A CÂȘTIGA!",
WIN_UP_TO_128_FREE_SPINS: "CÂȘTIGĂ PÂNĂ LA 128 ROTIRI GRATUITE",
FREE: "GRATUITE"
},
HUD_MESSAGES: {
EXTEND_MATRIX: "Extinde matricea",
PAY_BOTH_WAYS_MATRIX: "Plătește în ambele sensuri și extinde matricea",
ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Activează plățile mixte și
extinde matricea",

```

IS\_ACTIVE\_ALL: " Când Luna roșie este activă, toate personajele își iau forma animală și împart câștigurile (plăți mixte)",  
"3X": "3x",  
DURING\_THE\_FREE\_SPIN: "În timpul învârtirilor gratuite, jocul este întotdeauna 5x5 și plătește în ambele direcții",  
TRIGGER\_6\_FREE\_SPINS: "declanșează 6 învârtiri gratuite.",  
FREE\_SPIN\_CAN\_RETRIGGER: "Învârtirile gratuite pot redeclanșa",  
GOOD\_LUCK: "Good Luck",  
X\_FREE\_PLAY: "%{remainingCount} jocuri gratuite rămase",  
X\_FREE\_PLAY\_COMPLETED: "jocuri gratuite rămase",  
FREE\_SPIN\_AWARDED: "ați primit %{spinCount} jocuri gratuite",  
FREE\_SPIN\_OVER: "jocuri gratuite încheiate",  
NO\_THANKS: "nu, mulțumesc",  
PLAY\_LATER: "joc mai târziu",  
PLAY\_NOW: "joc acum",  
},  
PAYTABLE: {  
PAYTABLE: "TABEL DE PLĂȚI",  
PAYS\_BOTH\_WAYS: "Plătește în ambele sensuri și extinde matricea",  
EXTENDS\_MATRIX: "Extinde matricea",  
ACTIVATES\_MIXED: "Activează plățile mixte și extinde matricea",  
TRIGGER\_6\_FREE\_SPIN: "declanșează 6 învârtiri gratuite",  
AND\_PAYS: "5x5 și plătește în ambele direcții",  
WHEN\_THE\_RED: "Când Luna roșie este activă, toate personajele își iau forma animală și împart câștigurile (plăți mixte)",  
MALFUNCTION\_TEXT: "DEFECȚIUNILE ANULEAZĂ TOATE PLĂȚILE ȘI JOCURILE",  
GAME\_RULES: "REGULI DE JOC",  
GAME\_RULES1: "VOLSUNGA ESTE UN MOD DE JOC CU O MATRICE CU EXTINDERE 53135, 53335, 55355 ȘI 55555",  
GAME\_RULES2: "JUCĂTORII AU 3125 DE MODURI DE A CÂȘTIGA",  
GAME\_RULES3: "JOCUL ARE UN RTP DE 97,16%",  
GAME\_RULES4: "TOATE PLĂȚILE, ÎN AFARĂ DE CELE SCATTER, SE FAC DE LA STÂNGA LA DREAPTA, ÎNCEPÂND CU ROLA DIN EXTREMITATEA STÂNGĂ",  
GAME\_RULES5: "PLĂȚILE SCATTER SUNT FĂCUTE PENTRU SIMBOLURI CARE APAR ÎN ORICE POZIȚIE PE ROLE",  
GAME\_RULES6: "NUMAI CEL MAI MARE CÂȘTIG DIN SIMBOLURI ESTE PLĂTIT PER COMBINAȚIE CÂȘTIGĂTOARE",  
GAME\_RULES7: "NUMAI CEL MAI MARE CÂȘTIG SCATTER ESTE PLĂTIT PER JOC",  
GAME\_RULES8: "CÂȘTIGURILE SCATTER SE PLĂTESC ÎN PLUS FAȚĂ DE CÂȘTIGURILE DIN MODURI",  
GAME\_RULES9: "CÂȘTIGURILE DIN MODURI SE ÎNMULȚESC CU MULTIPLICĂTORUL DE PARIU",  
GAME\_RULES10: "PLĂȚILE SCATTER SUNT ÎNMULȚITE CU PARIUL TOTAL ÎN CREDITE",  
GAME\_RULES11: "ÎN TIMPUL ÎNVÂRTIRILOR GRATUITE, JOCUL ESTE ÎNTOTDEAUNA 5X5 ȘI PLĂTEȘTE TOT TIMPUL ÎN AMBELE SENSURI",  
GAME\_RULES12: "ÎNVÂRTIRILE GRATUITE POT FI REDECLANȘATE",  
},  
BETSETTINGS: "SETĂRI PARIU",  
AUTOPLAY: {

```

AUTOPLAY_TITLE: "JOC AUTOMAT",
AUTOPLAY_SUBTITLE_1: "Selectează numărul de rotiri",
AUTOPLAY_SUBTITLE_2: "Oprește jocul automat",
AUTOPLAY_DES_1: "Dacă pierderile depășesc",
AUTOPLAY_DES_2: "Dacă un singur câștig depășește",
CLEAR: "Șterge",
PLAY: "Joacă",
WIN_LIMIT_EXCEED: "câștig depășește",
LOSS_LIMIT_EXCEED: "pierderile depășesc"
},
SOUNDSETTINGS: {
    SOUND: "SUNET",
    BACKGROUND_SOUND: "SUNET DE FUNDAL"
},
MSG: {
    BIG_WIN: "CÂȘTIG BUN",
    HUGE_WIN: "CÂȘTIG EXTREM!",
    EPIC_WIN: "CÂȘTIG GROZAV!",
    MEGA_WIN: "CÂȘTIG COLOSAL!",
    CONGRATULATIONS: "FELICITĂRI",
    CREDITS_WON: "CREDITE CÂȘTIGATE",
    COMBINATION: "COMBINAȚIE/-II",
    AWARDED: "PREMIAT",
    FREE_SPINS: "ÎNVÂRTIRI GRATUITE",
    ADDITIONAL_FREE_SPINS_AWARDED: "ÎNVÂRTIRI GRATUITE
SUPLIMENTARE ACORDATE",
    YOU_HAVE_WON: "AȚI CÂȘTIGAT",
    FREE_SPINS_AWARDED: "ÎNVÂRTIRI GRATUITE ACORDATE",
    REALITY_CHECK_LOSS_LIMIT: "Limită de pierdere depășită",
    REALITY_CHECK_SESSION_LENGTH: "Durata sesiunii de joc a fost depășită",
    CLICK_TO_START: "Faceți clic pentru a începe",
    SESSION_TIME: "Durată sesiune",
    WIN_AMOUNT: "Valoare câștig",
    LOSS_AMOUNT: "Valoare pierdere",
    UNFINISHED_GAME_ROUND: "RUNDE DE JOC NEFINALIZATE"
},
BUTTON: {
    OK: "OK",
    CANCEL: "ANULARE",
    CONTINUE_PLAYING: "CONTINUARE JOC",
    CLEAR: "ȘTERGERE",
    CONTINUE: "CONTINUARE",
    AH: "Istoric cont",
    RL: "RETURN LA LOBBY"
},
ERROR: {
    101: "Tichet nevalid",
    404: "Eroare de comunicare",
    501: "Eroare de sistem. Contactați echipa de asistență și transmiteți codul de eroare 501",
    502: "Eroare de sistem. Contactați echipa de asistență și transmiteți codul de eroare 502",
    503: "Sesiune lipsă sau expirată",
    504: "Eroare de sistem. Contactați echipa de asistență și transmiteți codul de eroare 504",

```

```

509: "Eroare de sistem. Contactați echipa de asistență și transmiteți codul de eroare 509",
516: "Eroare de sistem. Contactați echipa de asistență și transmiteți codul de eroare 516",
517: "Eroare de sistem. Contactați echipa de asistență și transmiteți codul de eroare 517",
518: "Eroare de sistem. Contactați echipa de asistență și transmiteți codul de eroare 518",
522: "Eroare de sistem. Contactați echipa de asistență și transmiteți codul de eroare 522",
523: "Eroare de sistem. Contactați echipa de asistență și transmiteți codul de eroare 523",
541: "Eroare de sistem. Contactați echipa de asistență și transmiteți codul de eroare 541",
543: "Eroare de sistem. Contactați echipa de asistență și transmiteți codul de eroare 543",
551: "Eroare de sistem. Contactați echipa de asistență și transmiteți codul de eroare 551",
505: "Curs joc nevalid. Contactați echipa de asistență cu codul de eroare 505.",
507: "Pariu nevalid. Contactați echipa de asistență cu codul de eroare 507.",
508: "Eroare comandă sistem. Contactați echipa de asistență cu codul de eroare 508.",
519: "Date de conectare nevalide. Contactați echipa de asistență cu codul de eroare 519.",
520: "Jocul este activ în alt browser.",
525: "Eroare monedă. Contactați echipa de asistență cu codul de eroare 525.",
526: "Eroare conectare. Contactați echipa de asistență cu codul de eroare 526.",
537: "Eroare conectare. Contactați echipa de asistență cu codul de eroare 537.",
527: "Eroare învârtire gratuită. Contactați echipa de asistență cu codul de eroare 527.",
532: "Eroare învârtire gratuită. Contactați echipa de asistență cu codul de eroare 532.",
528: "URL nevalid. ",
533: "Eroare token. Contactați echipa de asistență cu codul de eroare 533.",
536: "Eroare token. Contactați echipa de asistență cu codul de eroare 536.",
549: "Eroare token. Contactați echipa de asistență cu codul de eroare 549.",
535: "Versiune de joc prea veche.",
538: "Eroare Jackpot. Contactați echipa de asistență cu codul de eroare 538.",
539: "Eroare Jackpot. Contactați echipa de asistență cu codul de eroare 539.",
540: "Jucătorul are un joc activ pe alt server.",
542: "Eroare versiune. Contactați echipa de asistență cu codul de eroare 542.",
544: "Buget disponibil depășit",
547: "Eroare monedă. Contactați echipa de asistență cu codul de eroare 547.",
550: "A fost atinsă limita de câștig",
506: "Nu există fonduri suficiente pentru valoarea de pariu aleasă",
521: "Eroare tehnică în timpul conectării",
524: "Joc dezactivat",
531: "Blocat de la joc",
534: "Tentativa de conectare provine dintr-o regiune blocată",
545: "Limită de pierdere depășită",
546: "Durata sesiunii de joc a fost depășită",
548: "A expirat sesiunea Playtech de trezire la realitate",
554: "Blocked from playing - daily time limit exceeded",
555: "Blocked from playing - weekly time limit exceeded",
556: "Blocked from playing - monthly time limit exceeded"
}
}, cc._RF.pop()
}, {}],
ru: [function(e, t, i) {
"use strict";
cc._RF.push(t, "2ebc75/OoJJxIMDTkH7ewni", "ru"), t.exports = {
WELCOME: "Welcome, %{name}!",
Click_anywhere_on_the_screen_to_start_the_game: "Нажмите в любом месте на экране,
чтобы начать игру",
HUD: {

```

BET: "СТАВКА",  
 CREDITS: "НОМИНАЛЫ",  
 COIN\_VALUE: "СТОИМОСТЬ МОНЕТЫ",  
 WIN: "ВЫИГРЫШ",  
 AUTOPLAY: "АВТОИГРА",  
 MAXBET: "МАКС. СТАВКА",  
 "576WAYS\_TO\_WIN": "576 ПУТЕЙ ВЫИГРЫША!",  
 WIN\_UP\_TO\_128\_FREE\_SPINS: "ВЫИГРЫШ ДО 128 БЕСПЛАТНЫХ  
 ВРАЩЕНИЙ",  
 FREE: "БЕСПЛАТНЫХ"  
 },  
 HUD\_MESSAGES: {  
 EXTEND\_MATRIX: "Расширяет матрицу",  
 PAY\_BOTH\_WAYS\_MATRIX: "Оплачивается в обе стороны и расширяет  
 матрицу",  
 ACTIVATES\_MIXED\_PAYS\_AND\_EXTEND\_MATRIX: "Активирует смешанные  
 выплаты и расширяет матрицу",  
 IS\_ACTIVE\_ALL: " При активной Красной Луне все персонажи трансформируются  
 в свои животные формы и разделяют выигрыш (смешанные выплаты)",  
 "3X": "3x",  
 DURING\_THE\_FREE\_SPIN: "Во время бесплатных спинов в игре всегда  
 используется решетка 5x5, выплата всегда производится в обе стороны",  
 TRIGGER\_6\_FREE\_SPINS: "активируют 6 бесплатных спинов",  
 FREE\_SPIN\_CAN\_RETRIGGER: "Активация бесплатных спинов может быть  
 произведена повторно",  
 GOOD\_LUCK: "Good Luck",  
 X\_FREE\_PLAY: "осталось %{remainingCount} бесплатных розыгрышей",  
 X\_FREE\_PLAY\_COMPLETED: "осталось бесплатных розыгрышей",  
 FREE\_SPIN\_AWARDED: "вы получаете %{spinCount} бесплатных розыгрышей",  
 FREE\_SPIN\_OVER: "бесплатные розыгрыши завершены",  
 NO\_THANKS: "нет, спасибо",  
 PLAY\_LATER: "играть позже",  
 PLAY\_NOW: "играть сейчас"  
 },  
 PAYTABLE: {  
 PAYTABLE: "ТАБЛИЦА ВЫПЛАТ",  
 PAYS\_BOTH\_WAYS: "Оплачивается в обе стороны и расширяет матрицу",  
 EXTENDS\_MATRIX: "Расширяет матрицу",  
 ACTIVATES\_MIXED: "Активирует смешанные выплаты и расширяет матрицу",  
 TRIGGER\_6\_FREE\_SPIN: "активируют 6 бесплатных спинов",  
 AND\_PAYS: "5x5 и выплата в обе стороны",  
 WHEN\_THE\_RED: "При активной Красной Луне все персонажи трансформируются  
 в свои животные формы и разделяют выигрыш (смешанные выплаты)",  
 MALFUNCTION\_TEXT: "ПРИ ВОЗНИКНОВЕНИИ НЕИСПРАВНОСТИ ВСЕ  
 ВЫПЛАТЫ И ИГРЫ АННУЛИРУЮТСЯ",  
 GAME\_RULES: "ПРАВИЛА ИГРЫ",  
 GAME\_RULES1: "VOLSUNGA – ЭТО ВИДЕОСЛОТ С РАСШИРЯЮЩЕЙСЯ  
 МАТРИЦЕЙ 53135, 53335, 55355 И 55555",  
 GAME\_RULES2: "3125 ВЫИГРЫШНЫХ КОМБИНАЦИЙ",  
 GAME\_RULES3: "КОЭФФИЦИЕНТ ВОЗВРАТА ИГРОКУ (RTP) СОСТАВЛЯЕТ  
 97.16%",  
 GAME\_RULES4: "ВСЕ ВЫПЛАТЫ, ЗА ИСКЛЮЧЕНИЕМ ВЫПЛАТ ЗА

SCATTER, ВЫПЛАЧИВАЮТСЯ СЛЕВА НАПРАВО, НАЧИНАЯ С КРАЙНЕГО ЛЕВОГО БАРАБАНА",

GAME\_RULES5: "ВЫПЛАТЫ ЗА SCATTER ПРОИЗВОДЯТСЯ ЗА СИМВОЛЫ, КОТОРЫЕ ВЫПАДАЮТ В ЛЮБОЙ ПОЗИЦИИ НА БАРАБАНАХ",

GAME\_RULES6: "ЗА ВЫИГРЫШНУЮ КОМБИНАЦИЮ ВЫПЛАЧИВАЕТСЯ ТОЛЬКО САМЫЙ ВЫСОКИЙ ВЫИГРЫШ ЗА СИМВОЛ",

GAME\_RULES7: "ЗА ИГРУ ВЫПЛАЧИВАЕТСЯ ТОЛЬКО САМЫЙ ВЫСОКИЙ ВЫИГРЫШ ЗА SCATTER",

GAME\_RULES8: "ВЫПЛАТЫ ЗА SCATTER ВЫПЛАЧИВАЮТСЯ В ДОПОЛНЕНИЕ К ВЫИГРЫШНЫМ КОМБИНАЦИЯМ",

GAME\_RULES9: "ВЫИГРЫШИ ПО КОМБИНАЦИЯМ УМНОЖАЮТСЯ НА МНОЖИТЕЛЬ СТАВКИ",

GAME\_RULES10: "ВЫИГРЫШИ ЗА SCATTER УМНОЖАЮТСЯ НА ОБЩУЮ СУММУ СТАВКИ",

GAME\_RULES11: "ВО ВРЕМЯ БЕСПЛАТНЫХ СПИНОВ В ИГРЕ ВСЕГДА ИСПОЛЬЗУЕТСЯ РЕШЕТКА 5X5, ВЫПЛАТА ВСЕГДА ПРОИЗВОДИТСЯ В ОБЕ СТОРОНЫ",

GAME\_RULES12: "БЕСПЛАТНЫЕ СПИНЫ МОГУТ АКТИВИРОВАТЬСЯ ПОВТОРНО"

},

BETSETTINGS: "НАСТРОЙКИ СТАВКИ",

AUTOPLAY: {

AUTOPLAY\_TITLE: "АВТОИГРА",

AUTOPLAY\_SUBTITLE\_1: "Выберите число вращений",

AUTOPLAY\_SUBTITLE\_2: "Остановить автоигру",

AUTOPLAY\_DES\_1: "Если проигрыш превысит",

AUTOPLAY\_DES\_2: "Если единичный выигрыш превысит",

CLEAR: "Очистить",

PLAY: "Играть",

WIN\_LIMIT\_EXCEED: "WIN LIMIT EXCEED",

LOSS\_LIMIT\_EXCEED: "выигрыш превысит"

},

SOUNDSETTINGS: {

SOUND: "ЗВУК",

BACKGROUND SOUND: "ФОНОВЫЙ ЗВУК"

},

MSG: {

BIG\_WIN: "ОТЛИЧНЫЙ ВЫИГРЫШ",

HUGE\_WIN: "ЭКСТРИМАЛЬНЫЙ ВЫИГРЫШ",

EPIC\_WIN: "МАССИВНЫЙ ВЫИГРЫШ",

MEGA\_WIN: "ОГРОМНЫЙ ВЫИГРЫШ",

CONGRATULATIONS: "ПОЗДРАВЛЯЕМ",

CREDITS\_WON: "ВЫИГРЫШ КРЕДИТОВ",

COMBINATION: "КОМБИНАЦИЯ (И)",

AWARDED: "ПРИСУЖДАЕТСЯ",

FREE\_SPINS: "БЕСПЛАТНЫЕ СПИНЫ",

ADDITIONAL\_FREE\_SPINS\_AWARDED: "ПРИСУЖДЕНЫ ДОПОЛНИТЕЛЬНЫЕ БЕСПЛАТНЫЕ СПИНЫ",

YOU\_HAVE\_WON: "ВЫ ВЫИГРАЛИ",

FREE\_SPINS\_AWARDED: "ПРИСУЖДЕНО БЕСПЛАТНЫХ СПИНОВ",

REALITY\_CHECK\_LOSS\_LIMIT: "Превышен лимит проигрыша при проверке связи с реальностью",

REALITY\_CHECK\_SESSION\_LENGTH: "Превышена длина сеанса проверки связи с реальностью",  
CLICK\_TO\_START: "Щелкнуть для запуска",  
SESSION\_TIME: "Время сеанса",  
WIN\_AMOUNT: "Сумма выигрыша",  
LOSS\_AMOUNT: "Сумма проигрыша",  
UNFINISHED\_GAME\_ROUND: "НЕЗАВЕРШЕННЫЕ ИГРОВЫЕ РАУНДЫ"  
},  
BUTTON: {  
  OK: "OK",  
  CANCEL: "ОТМЕНИТЬ",  
  CONTINUE\_PLAYING: "ПРОДОЛЖИТЬ ИГРАТЬ",  
  CLEAR: "УБРАТЬ",  
  CONTINUE: "ПРОДОЛЖИТЬ",  
  AH: "История учетной записи",  
  RL: "НАЗАД В ЛОББИ"  
},  
ERROR: {  
  101: "Неверный билет",  
  404: "Ошибка связи",  
  501: "Системная ошибка. Обратитесь в службу поддержки, указав код ошибки 501",  
  502: "Системная ошибка. Обратитесь в службу поддержки, указав код ошибки 502",  
  503: "Сеанс отсутствует или срок его действия истек",  
  504: "Системная ошибка. Обратитесь в службу поддержки, указав код ошибки 504",  
  509: "Системная ошибка. Обратитесь в службу поддержки, указав код ошибки 509",  
  516: "Системная ошибка. Обратитесь в службу поддержки, указав код ошибки 516",  
  517: "Системная ошибка. Обратитесь в службу поддержки, указав код ошибки 517",  
  518: "Системная ошибка. Обратитесь в службу поддержки, указав код ошибки 518",  
  522: "Системная ошибка. Обратитесь в службу поддержки, указав код ошибки 522",  
  523: "Системная ошибка. Обратитесь в службу поддержки, указав код ошибки 523",  
  541: "Системная ошибка. Обратитесь в службу поддержки, указав код ошибки 541",  
  543: "Системная ошибка. Обратитесь в службу поддержки, указав код ошибки 543",  
  551: "Системная ошибка. Обратитесь в службу поддержки, указав код ошибки 551",  
  505: "Идет недействительная игра. Свяжитесь со службой поддержки, указав код ошибки 505.",  
  507: "Недействительная ставка. Свяжитесь со службой поддержки, указав код ошибки 507.",  
  508: "Ошибка системной команды. Свяжитесь со службой поддержки, указав код ошибки 508.",  
  519: "Недействительный логин. Свяжитесь со службой поддержки, указав код ошибки 519.",  
  520: "Игра активна в другом браузере.",  
  525: "Ошибка валюты. Свяжитесь со службой поддержки, указав код ошибки 525.",  
  526: "Ошибка логина. Свяжитесь со службой поддержки, указав код ошибки 526.",  
  537: "Ошибка логина. Свяжитесь со службой поддержки, указав код ошибки 537.",  
  527: "Ошибка бесплатного спина. Свяжитесь со службой поддержки, указав код ошибки 527.",  
  532: "Ошибка бесплатного спина. Свяжитесь со службой поддержки, указав код ошибки 532.",  
  528: "Недействительный URL.",  
  533: "Ошибка токена. Свяжитесь со службой поддержки, указав код ошибки 533.",  
  536: "Ошибка токена. Свяжитесь со службой поддержки, указав код ошибки 536.",

549: "Ошибка токена. Свяжитесь со службой поддержки, указав код ошибки 549.",  
535: "Слишком старая версия игры.",  
538: "Ошибка джекпота. Свяжитесь со службой поддержки, указав код ошибки  
538.",  
539: "Ошибка джекпота. Свяжитесь со службой поддержки, указав код ошибки  
538.",  
540: "У игрока активный сеанс игры на другом сервере.",  
542: "Ошибка версии. Свяжитесь со службой поддержки, указав код ошибки 542.",  
544: "Превышен бюджет расходов",  
547: "Ошибка валюты. Свяжитесь со службой поддержки, указав код ошибки 547.",  
550: "Достигнут предел выигрыша",  
506: "Недостаточно средств для выбранного размера ставки",  
521: "Техническая ошибка во время входа",  
524: "Игра отключена",  
531: "Заблокирован от игры",  
534: "Произведена попытка входа из заблокированного региона",  
545: "Превышен лимит проигрыша при проверке связи с реальностью",  
546: "Превышена длина сеанса проверки связи с реальностью",  
548: "Истекло время сеанса проверки связи с реальностью Playtech",  
554: "Blocked from playing - daily time limit exceeded",  
555: "Blocked from playing - weekly time limit exceeded",  
556: "Blocked from playing - monthly time limit exceeded"  
}  
}, cc.\_RF.pop()  
}, {}],  
sr: [function(e, t, i) {  
"use strict";  
cc.\_RF.push(t, "79008qDoUJMs7NUz9Huw/1P", "sr"), t.exports = {  
WELCOME: "Welcome, %{name}!",  
Click\_anywhere\_on\_the\_screen\_to\_start\_the\_game: "Kliknite bilo gde na ekranu da biste  
pokrenuli igru",  
HUD: {  
BET: "ULOG",  
CREDITS: "KREDITI",  
COIN\_VALUE: "VREDNOST NOVČIĆA",  
WIN: "DOBITAK",  
AUTOPLAY: "AUTO IGRA",  
MAXBET: "MAKSIMALNI ULOG",  
"576WAYS\_TO\_WIN": "576 NAČINA ZA DOBITAK!",  
WIN\_UP\_TO\_128\_FREE\_SPINS: "OSVOJITE DO 128 BESPLATNIH SPINOVA",  
FREE: "BESPLATNIH"  
},  
HUD\_MESSAGES: {  
EXTEND\_MATRIX: "Extends matrix",  
PAY\_BOTH\_WAYS\_MATRIX: "Pays both ways and extends matrix",  
ACTIVATES\_MIXED\_PAYS\_AND\_EXTEND\_MATRIX: "Activates mixed pays and  
extends matrix",  
IS\_ACTIVE\_ALL: " is active, all characters shape-shift into their animal form and share  
wins (mixed pays)",  
"3X": "3x",  
DURING\_THE\_FREE\_SPIN: "During free spins, the game is always 5x5 and always  
pays both ways",

```

    TRIGGER_6_FREE_SPINS: "trigger 6 free spins is active.",
    FREE_SPIN_CAN_RETRIGGER: "Free spins can retrigger",
    GOOD_LUCK: "Good Luck",
    X_FREE_PLAY: "%{remainingCount} free plays remaining",
    X_FREE_PLAY_COMPLETED: "free plays remaining",
    FREE_SPIN_AWARDED: "You have been awarded %{spinCount} free spins",
    FREE_SPIN_OVER: "free spins over"
},
PAYTABLE: {
    PAYTABLE: "PAYTABLE",
    PAYS_BOTH_WAYS: "Pays both ways and extends matrix",
    EXTENDS_MATRIX: "Extends Matrix",
    ACTIVATES_MIXED: "Activates mixed pays and extends matrix",
    TRIGGER_6_FREE_SPIN: "trigger 6 free spins",
    AND_PAYS: "5x5 and pays both ways",
    WHEN_THE_RED: "When the Red Moon is active, all characters shape-shift into their
animal form and share wins (mixed pays)",
    MALFUNCTION_TEXT: "Malfunction voids all pays and plays",
    GAME_RULES: "GAME_RULES",
    GAME_RULES1: "Volsunga is a ways game with an extending 53135, 53335, 55355 and
55555 matrix",
    GAME_RULES2: "Players have 3125 ways to win",
    GAME_RULES3: "The game has a 97.16% RTP",
    GAME_RULES4: "All pays except scatter pays are left-to-right starting from the leftmost
reel",
    GAME_RULES5: "Scatter pays are paid for symbols appearing in any position on the
reels",
    GAME_RULES6: "Only the highest symbol win is paid per winning combination",
    GAME_RULES7: "Only the highest scatter win is paid per play",
    GAME_RULES8: "Scatter pays are paid in addition to ways pays",
    GAME_RULES9: "Ways pays are multiplied by the bet multiple",
    GAME_RULES10: "Scatter pays are multiplied by the total credits bet",
    GAME_RULES11: "During free spins the game is always 5x5 and always pays both
ways",
    GAME_RULES12: "Free spins can retrigger"
},
BETSETTINGS: "PODEŠAVANJA ULOGA",
AUTOPLAY: {
    AUTOPLAY_TITLE: "AUTO IGRA",
    AUTOPLAY_SUBTITLE_1: "Izaberite broj spinova",
    AUTOPLAY_SUBTITLE_2: "Zaustavite auto igru",
    AUTOPLAY_DES_1: "Ukoliko gubitak premaši",
    AUTOPLAY_DES_2: "Ukoliko jedinstveni dobitak premaši",
    CLEAR: "Čisto",
    PLAY: "Igraj",
    WIN_LIMIT_EXCEED: "WIN LIMIT EXCEED",
    LOSS_LIMIT_EXCEED: "LOSS LIMIT EXCEED"
},
SOUNDSETTINGS: {
    SOUND: "ZVUK",
    BACKGROUND_SOUND: "ZVUK U POZADINI"
},

```

```

MSG: {
  BIG_WIN: "DOBAR DOBITAK",
  HUGE_WIN: "ODLIČAN DOBITAK",
  EPIC_WIN: "NEVEROVATAN DOBITAK",
  MEGA_WIN: "OGROMAN DOBITAK",
  CONGRATULATIONS: "ČESTITAMO",
  CREDITS_WON: "OSVOJENO KREDITA",
  COMBINATION: "KOMBINACIJA(E)",
  AWARDED: "OSVOJENO",
  FREE_SPINS: "FREE SPINS",
  ADDITIONAL_FREE_SPINS_AWARDED: "ADDITIONAL FREE SPINS
AWARDED",
  YOU_HAVE_WON: "YOU HAVE WON",
  FREE_SPINS_AWARDED: "FREE SPINS AWARDED",
  REALITY_CHECK_LOSS_LIMIT: "Reality check loss limit exceeded",
  REALITY_CHECK_SESSION_LENGTH: "Reality check session length exceeded"
},
BUTTON: {
  OK: "OK",
  CANCEL: "CANCEL",
  CONTINUE_PLAYING: "CONTINUE PLAYING!",
  CLEAR: "CLEAR",
  CONTINUE: "CONTINUE"
}
}, cc._RF.pop()
}, {}],
sv: [function(e, t, i) {
  "use strict";
  cc._RF.push(t, "18db1dG525KLaTVD7PO5ceH", "sv"), t.exports = {
    WELCOME: "Welcome, %{name}!",
    Click_anywhere_on_the_screen_to_start_the_game: "Klicka var som helst på skärmen för
att starta spelet",
    HUD: {
      BET: "INSATS",
      CREDITS: "KREDITER",
      COIN_VALUE: "MYNTVÄRDE",
      WIN: "VINST",
      AUTOPLAY: "AUTOSPEL",
      MAXBET: "MAX INSATS",
      "576WAYS_TO_WIN": "576 SÄTT ATT VINNA!",
      WIN_UP_TO_128_FREE_SPINS: "VINN UPP TILL 128 GRATIS SNURR",
      FREE: "GRATIS"
    },
    HUD_MESSAGES: {
      EXTEND_MATRIX: "Utökar matrisen",
      PAY_BOTH_WAYS_MATRIX: "Ger vinster åt båda hållen och utökar matrisen",
      ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Aktiverar blandade
vinstsätt och utökar matrisen",
      IS_ACTIVE_ALL: "När den röda månen är aktiv förvandlas alla karaktärer till sina
djuriska skepnader och delar vinster (blandade vinstsätt)",
      "3X": "3x",
      DURING_THE_FREE_SPIN: "Under gratisnurr är spelet alltid 5x5 och betalar alltid åt

```

```

båda hållen",
    TRIGGER_6_FREE_SPINS: "aktiverar 6 gratissnurr",
    FREE_SPIN_CAN_RETRIGGER: "Gratissnurr kan återaktiveras",
    GOOD_LUCK: "Good Luck",
    X_FREE_PLAY: "%{remainingCount} frispel återstår",
    X_FREE_PLAY_COMPLETED: "frispel återstår",
    FREE_SPIN_AWARDED: "du har fått %{spinCount} frispel",
    FREE_SPIN_OVER: "frispielen är slut",
    NO_THANKS: "nej tack",
    PLAY_LATER: "spela senare",
    PLAY_NOW: "spela nu"
},
PAYTABLE: {
    PAYTABLE: "VINSTTABELL",
    PAYS_BOTH_WAYS: "Ger vinster åt båda hållen och utökar matrisen",
    EXTENDS_MATRIX: "Utökar matrisen",
    ACTIVATES_MIXED: "Aktiverar blandade vinstsätt och utökar matrisen",
    TRIGGER_6_FREE_SPIN: "aktiverar 6 gratissnurr",
    AND_PAYS: "5x5 och betalar åt båda hållen",
    WHEN_THE_RED: "När den röda månen är aktiv förvandlas alla karaktärer till sina
djuriska skepnader och delar vinster (blandade vinstsätt)",
    MALFUNCTION_TEXT: "VID FEL OGILTIGFÖRKLARAS ALLA
SPELHÄNDELSER OCH SPELTRANSAKTIONER",
    GAME_RULES: "SPELREGLER",
    GAME_RULES1: "VOLSUNGA ÄR ETT KOMBINATIONSSPEL MED EN
UTÖKANDE MÄTRIS PÅ 53135, 53335, 55355 OCH 55555",
    GAME_RULES2: "DU KAN VINNA PÅ 3 125 OLIKA SÄTT",
    GAME_RULES3: "SPELET HAR 97,16 % i RTP",
    GAME_RULES4: "ALLA VINSTER FÖRUTOM SCATTERVINSTER GER
UTDELNING FRÅN VÄNSTER TILL HÖGER MED BÖRJAN FRÅN HJULET LÄNGST TILL
VÄNSTER",
    GAME_RULES5: "SCATTERVINSTER TILLDELAS NÄR SYMBOLER DYKER
UPP PÅ HJULEN OAVSETT POSITION",
    GAME_RULES6: "ENDAST DEN HÖGSTA SYMBOLVINSTEN PER VINNANDE
KOMBINATION BETALAS UT",
    GAME_RULES7: "ENDAST DEN HÖGSTA SCATTERVINSTEN PER OMGÅNG
BETALAS UT",
    GAME_RULES8: "SCATTERVINSTER BETALAS UT I TILLÄGG TILL VANLIGA
VINSTER",
    GAME_RULES9: "KOMBINATIONSVINSTER MULTIPLICERAS MED
INSATSMULTIPLIKATORN",
    GAME_RULES10: "SCATTERVINSTER MULTIPLICERAS MED
TOTALINSATSEN",
    GAME_RULES11: "UNDER GRATISSNURR ÄR SPELET ALLTID 5X5 OCH KAN
ALLTID GE VINST ÅT BÅDA HÅLLEN",
    GAME_RULES12: "GRATISSNURR KAN ÅTERAKTIVERAS"
},
BETSETTINGS: "INSATS INSTÄLLNINGAR",
AUTOPLAY: {
    AUTOPLAY_TITLE: "AUTOSPEL",
    AUTOPLAY_SUBTITLE_1: "Välj antal snurr",
    AUTOPLAY_SUBTITLE_2: "Stoppa auto",

```

```

    AUTOPLAY_DES_1: "Om förlusten överskrider",
    AUTOPLAY_DES_2: "Om en vinst är större än",
    CLEAR: "Rensa",
    PLAY: "Spela",
    WIN_LIMIT_EXCEED: "vinst är större",
    LOSS_LIMIT_EXCEED: "förlusten överskrider"
},
SOUNDSETTINGS: {
    SOUND: "LJUD",
    BACKGROUND_SOUND: "BAKGRUNDSLJUD"
},
MSG: {
    BIG_WIN: "FIN VINST",
    HUGE_WIN: "EXTREM VINST!",
    EPIC_WIN: "MASSIV VINST",
    MEGA_WIN: "GIGANTISK VINST",
    CONGRATULATIONS: "GRATTIS",
    CREDITS_WON: "KREDITER VUNNA",
    COMBINATION: "KOMBINATION(ER)",
    AWARDED: "BELÖNAD",
    FREE_SPINS: "GRATISSNURR",
    ADDITIONAL_FREE_SPINS_AWARDED: "FLER GRATISSNURR TILLDELADE",
    YOU_HAVE_WON: "DU HAR VUNNIT",
    FREE_SPINS_AWARDED: "ERHÅLLNA GRATISSNURR",
    REALITY_CHECK_LOSS_LIMIT: "Verklighetskollens förlustgräns har överskridits",
    REALITY_CHECK_SESSION_LENGTH: "Verklighetskollens sessionslängd har
överskridits",
    CLICK_TO_START: "Klicka för att starta",
    SESSION_TIME: "Sessionstid",
    WIN_AMOUNT: "Vinstbelopp",
    LOSS_AMOUNT: "Förlustbelopp",
    UNFINISHED_GAME_ROUND: "OAVSLUTADE SPELOMGÅNGAR"
},
BUTTON: {
    OK: "OK",
    CANCEL: "AVBRYT",
    CONTINUE_PLAYING: "FORTSÄTT SPELA",
    CLEAR: "RENSA",
    CONTINUE: "FORTSÄTT",
    AH: "Kontohistorik",
    RL: "TILLBAKA TILL LOBBYN"
},
ERROR: {
    101: "Ogiltig Biljett",
    404: "kommunikationsfel",
    501: "Systemfel. Vänligen kontakta supporten och uppge felkoden 501",
    502: "Systemfel. Vänligen kontakta supporten och uppge felkoden 502",
    503: "Session saknas eller har gått ut",
    504: "Systemfel. Vänligen kontakta supporten och uppge felkoden 504",
    509: "Systemfel. Vänligen kontakta supporten och uppge felkoden 509",
    516: "Systemfel. Vänligen kontakta supporten och uppge felkoden 516",
    517: "Systemfel. Vänligen kontakta supporten och uppge felkoden 517",

```

518: "Systemfel. Vänligen kontakta supporten och uppge felkoden 518",  
522: "Systemfel. Vänligen kontakta supporten och uppge felkoden 522",  
523: "Systemfel. Vänligen kontakta supporten och uppge felkoden 523",  
541: "Systemfel. Vänligen kontakta supporten och uppge felkoden 541",  
543: "Systemfel. Vänligen kontakta supporten och uppge felkoden 543",  
551: "Systemfel. Vänligen kontakta supporten och uppge felkoden 551",  
505: "Ogiltigt spelförlopp. Kontakta support och uppge felkoden 505.",  
507: "Ogiltig insats. Kontakta support och uppge felkoden 507",  
508: "Systemkommandofel. Kontakta support och uppge felkoden 508.",  
519: "Ogiltig inloggning. Kontakta support och uppge felkoden 519.",  
520: "Spelet är aktivt i en annan webbläsare.",  
525: "Valutafel. Kontakta support och uppge felkoden 525.",  
526: "Inloggningsfel. Kontakta support och uppge felkoden 526",  
537: "Inloggningsfel. Kontakta support och uppge felkoden 537",  
527: "Freeroll-fel. Kontakta support och uppge felkoden 527. ",  
532: "Freeroll-fel. Kontakta support och uppge felkoden 532. ",  
528: "Ogiltig URL.",  
533: "Pollettfel. Kontakta support och uppge felkoden 533.",  
536: "Pollettfel. Kontakta support och uppge felkoden 536.",  
549: "Pollettfel. Kontakta support och uppge felkoden 549.",  
535: "Spelversionen är för gammal.",  
538: "Jackpottfel. Kontakta support och uppge felkoden 538.",  
539: "Jackpottfel. Kontakta support och uppge felkoden 539.",  
540: "Spelaren har ett aktivt spel på en annan server.",  
542: "Versionsfel. Kontakta support och uppge felkoden 542.",  
544: "Spenderingsbudget överskriden",  
547: "Valutafel. Kontakta support och uppge felkoden 547.",  
550: "Vinsttaket har nåtts",  
506: "Inte tillräckligt med medel för vald insatsstorlek",  
521: "Tekniskt fel vid inloggning",  
524: "Spel inaktiverat",  
531: "Spärrad från att spela",  
534: "Inloggningsförsök kommer från blockerad region",  
545: "Verklighetskollens förlustgräns har överskridits",  
546: "Verklighetskollens sessionslängd har överskridits",  
548: "Verklighetskoll-sessionen för Playtech har löpt ut",  
554: "Spärrad från att spela - du har uppnått din dagliga tidsgrän",  
555: "Spärrad från att spela - du har uppnått din veckovisa tidsgräns",  
556: "Spärrad från att spela - du har uppnått din månatliga tidsgräns"

}

}, cc.\_RF.pop()

}, {}],

tr: [function(e, t, i) {

"use strict";

cc.\_RF.push(t, "fb3f4Aer5lJPrI8+reIHl8F", "tr"), t.exports = {

WELCOME: "Welcome, %{name}!",

Click\_anywhere\_on\_the\_screen\_to\_start\_the\_game: "Oyunu başlatmak için ekranda herhangi bir yere tıklayın",

HUD: {

BET: "BAHİS YAP",

CREDITS: "KREDİ",

COIN\_VALUE: "JETON DEĞERİ",

```

WIN: "KAZANÇ",
AUTOPLAY: "OTO-OYUN",
MAXBET: "MAKS BAHİS",
"576WAYS_TO_WIN": "KAZANMAK İÇİN 576 YOL!",
WIN_UP_TO_128_FREE_SPINS: "128'E KADAR SERBEST DÖNÜŞ KAZANIN!",
FREE: "SERBEST"
},
HUD_MESSAGES: {
EXTEND_MATRIX: "Extends matrix!!",
PAY_BOTH_WAYS_MATRIX: "Pays both ways and extends matrix!!",
ACTIVATES_MIXED_PAYS_AND_EXTEND_MATRIX: "Activates mixed pays and
extends matrix!!",
IS_ACTIVE_ALL: " is active, all characters shape-shift into their animal form and share
wins (mixed pays)!!",
"3X": "3x",
DURING_THE_FREE_SPIN: "During free spins, the game is always 5x5 and always
pays both ways",
TRIGGER_6_FREE_SPINS: "trigger 6 free spins is active.!!",
FREE_SPIN_CAN_RETRIGGER: "Free spins can retrigger",
GOOD_LUCK: "Good Luck!!",
X_FREE_PLAY: "%{remainingCount} free plays remaining",
X_FREE_PLAY_COMPLETED: "free plays remaining",
FREE_SPIN_AWARDED: "You have been awarded %{spinCount} free spins",
FREE_SPIN_OVER: "free spins over"
},
PAYTABLE: {
PAYTABLE: "PAYTABLE",
PAYS_BOTH_WAYS: "Pays both ways and extends matrix",
EXTENDS_MATRIX: "Extends Matrix",
ACTIVATES_MIXED: "Activates mixed pays and extends matrix",
TRIGGER_6_FREE_SPIN: "trigger 6 free spins",
AND_PAYS: "5x5 and pays both ways",
WHEN_THE_RED: "When the Red Moon is active, all characters shape-shift into their
animal form and share wins (mixed pays)",
MALFUNCTION_TEXT: "Malfunction voids all pays and plays",
GAME_RULES: "GAME_RULES",
GAME_RULES1: "Volsunga is a ways game with an extending 53135, 53335, 55355 and
55555 matrix",
GAME_RULES2: "Players have 3125 ways to win",
GAME_RULES3: "The game has a 97.16% RTP",
GAME_RULES4: "All pays except scatter pays are left-to-right starting from the leftmost
reel",
GAME_RULES5: "Scatter pays are paid for symbols appearing in any position on the
reels",
GAME_RULES6: "Only the highest symbol win is paid per winning combination",
GAME_RULES7: "Only the highest scatter win is paid per play",
GAME_RULES8: "Scatter pays are paid in addition to ways pays",
GAME_RULES9: "Ways pays are multiplied by the bet multiple",
GAME_RULES10: "Scatter pays are multiplied by the total credits bet",
GAME_RULES11: "During free spins the game is always 5x5 and always pays both
ways",
GAME_RULES12: "Free spins can retrigger"

```

```

    },
    BETSETTINGS: "BAHİS AYARLARI",
    AUTOPLAY: {
        AUTOPLAY_TITLE: "OTO-OYUN",
        AUTOPLAY_SUBTITLE_1: "Dönüş sayısını seçin",
        AUTOPLAY_SUBTITLE_2: "Oto-oyunu durdurun",
        AUTOPLAY_DES_1: "Eğer kayıp bu miktarı geçerse:",
        AUTOPLAY_DES_2: "Eğer kazanç bu miktarı geçerse:",
        CLEAR: "Temizle",
        PLAY: "Oyna",
        WIN_LIMIT_EXCEED: "WIN LIMIT EXCEED",
        LOSS_LIMIT_EXCEED: "LOSS LIMIT EXCEED"
    },
    SOUNDSETTINGS: {
        SOUND: "SES",
        BACKGROUND_SOUND: "ARKA PLAN SESİ"
    },
    MSG: {
        BIG_WIN: "İYİ KAZANÇ!!",
        HUGE_WIN: "OLAĞANÜSTÜ KAZANÇ!!",
        EPIC_WIN: "MUAZZAM KAZANÇ!!",
        MEGA_WIN: "DEVASA KAZANÇ!!",
        CONGRATULATIONS: "TEBRİKLER",
        CREDITS_WON: "KREDİ KAZANILDI",
        COMBINATION: "KOMBİNASYON(LAR)",
        AWARDED: "ÖDÜLLENDİRİLDİ",
        FREE_SPINS: "FREE SPINS",
        ADDITIONAL_FREE_SPINS_AWARDED: "ADDITIONAL FREE SPINS
AWARDED",
        YOU_HAVE_WON: "YOU HAVE WON",
        FREE_SPINS_AWARDED: "FREE SPINS AWARDED",
        REALITY_CHECK_LOSS_LIMIT: "Reality check loss limit exceeded",
        REALITY_CHECK_SESSION_LENGTH: "Reality check session length exceeded"
    },
    BUTTON: {
        OK: "OK",
        CANCEL: "CANCEL",
        CONTINUE_PLAYING: "CONTINUE PLAYING!",
        CLEAR: "CLEAR",
        CONTINUE: "CONTINUE",
        RL: "RETURN TO LOBBY"
    }
}, cc._RF.pop()
}, {}],
zh: [function(e, t, i) {
    "use strict";
    cc._RF.push(t, "bb37aYMaFRKq43tr7CMDn1u", "zh"), t.exports = {
        HELLO: "你好",
        WORLD: "世界",
        WELCOME: "欢迎 , %{name} !"
    }, cc._RF.pop()
}, {}]

```

```
}, {}, [{"RTK_Coin", "RTK_CoinMaker", "SHS_GameGlobal", "SHS_GameHelper",  
"SHS_ApplicationManager", "SHS_AudioManager", "SHS_AutoPlayController",  
"SHS_AutoPlayRadioBtn", "SHS_BetController", "SHS_BonusSlotItem", "SHS_BonusSlotReel",  
"SHS_DebugPanelController", "SHS_ErrorMessageController",  
"SHS_FeaturePayTableController", "SHS_GameController", "SHS_GameEngineManager",  
"SHS_InfoController", "SHS_PSPAutoPlayController", "SHS_PSPHudController",  
"SHS_PayTableController", "SHS_PopupWindowController", "SHS_RadioBtn", "SHS_Relax",  
"SHS_RelaxError", "SHS_RelaxHandler", "SHS_SettingsController", "SHS_SlotItem",  
"SHS_SlotReel", "SHS_SlotWinningControl", "SHS_WinLableMessage", "SHS_AutoPlayLocal",  
"SHS_Feature", "SHS_HudLocal", "SHS_PaytableLocal", "SHS_SettingsLocal", "LabelLocalized",  
"bg", "cs", "da", "de", "el", "en", "es-eu", "es", "et", "fi", "fr", "hu", "is", "it", "lt", "lv", "nl", "no",  
"pl", "pt-br", "pt", "ro", "ru", "sr", "sv", "tr", "zh", "i18n", "polyglot"]]);
```