# Q&A

## 放了栅栏看不见

必须是预制(prefab)

## 配置

PatchUrl.lua

PatchUrl.lua

PatchUrl.lua

PatchUrl.lua

## AOI重复问题

Entity.createRenderObj() 碰到重复的不能立即删除（特别是主角）（不然相机丢失目标）

+1 或 2

1. 断线删除AOI
2. 重复利用Entity old并刷新：（**开销小**）

cv.SendMessage("onAppearanceCreated", SendMessageOptions.DontRequireReceiver);

## Objevent

view.objEvent.addEvent(this, "onAniMovement");

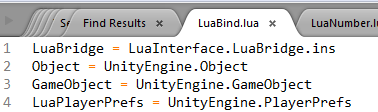
给动画控制器传参数用的

## 



## Connect.clear什么时候调用

## Lua缩写



## SVN

业务层分支地址：

svn://192.168.150.63/timefire/branches/branche\_xiaojun

svn://192.168.150.63/timefire/branches/branche\_ayu

svn://192.168.150.63/timefire/branches/branche\_framwork

后台框架层分支地址：

svn://192.168.150.63/mb/zeus/branches\_timefire/branch\_xiaojun

svn://192.168.150.63/mb/zeus/branches\_timefire/branch\_ayu

svn://192.168.150.63/mb/zeus/branches\_timefire/branch\_framwork

前台框架层分支地址：

svn://192.168.150.63/mb/cratos/branches\_timefire/branch\_xiaojun

svn://192.168.150.63/mb/cratos/branches\_timefire/branch\_ayu

svn://192.168.150.63/mb/cratos/branches\_timefire/branch\_framwork

注:外链需要自己本地checkout之后修改

## 字符数组

Console.WriteLine(BitConverter.ToString(a));

## Unsafe code requires the `unsafe’ command line option to be specified

#### 解决方案如下： 在工程的Assets目录下，创建两个文件：“csc.rsp”，“smcs.rsp”，用记事本打开，分别都输入： -unsafe 保存即可，随后关闭Unity重新打开并载入该工程，项目已经不报错了。

Zeus

//loginURL = "http://192.168.93.216:8080/login";

//loginURL = "http://192.168.93.253:8080/login"; //李永辉

Conn.cs

Gateway

Sess

## TAP

李永辉 yhli\_ex g3cz649U  
苗建超 jcmiao\_ex 9n2wl84D  
吴明生 mswu\_ex 362A8dtw  
  
登录地址： <http://tapd.tencent.com/>

## 宕机处理

try catch

## Shade

[Shader.WarmupAllShaders](http://www.baidu.com/link?url=MTt1e4HT6SkVPzdv7peBNhFSNwtNlkZr7AujVtgz3SNsCz2-m52BUicQdKhzjeVM)

## C#位运算

else if ((t & 0x8) != 0) {

yield return stream.read(4, tails);

ret = lead & 0x7UL;

ret |= (ulong)tails[0] << 3;

## kcp加密？,负载均衡服务器

string s = ConnHelper.formatBytes(buf);

## 长整数

ulongToNumber(id)

function tostring64(value)

local ret = nil

if type(value) == "number" then

ret = LuaNumber.doubleToString(value)

else

ret = tostring(value)

end

return ret

end

function equal64(value1, value2)

local v1 = tostring64(value1)

local v2 = tostring64(value2)

return v1 == v2

end

## Lua导出性能

原理分析：

http://blog.csdn.net/xingaaaxing/article/details/72869709

注册耗时，内存、查表，拆装箱

push 一个数字也不装箱

精确类型就没有

不要啥都用object传递和接收，那种出现装箱也只能怪自己

## 正则表达式查找替换

^.\*?([0-9]+\.).\*?href="(.\*?)".\*?>(.\*?数.\*?)<.\*?$

\1\3\n\2\n\n\n

^.\*public class (.\*) :.\*$

\1

**查找：**

**^.\*class (.\*) :.\*$**

**替换：**

**\_GT\(typeof\(\1\)\),**

## Redis

redis-server redis.windows.conf

## Go

GateWay会所有的lobby连接，lobby直接交互通过gateway?

[**liteide**](https://sourceforge.net/projects/liteide/)

Installing 13 tools at C:\Users\wumingsheng\go\bin

gocode

gopkgs

go-outline

go-symbols

guru

gorename

gomodifytags

impl

godef

goreturns

golint

gotests

dlv

Installing github.com/nsf/gocode SUCCEEDED

Installing github.com/uudashr/gopkgs/cmd/gopkgs SUCCEEDED

Installing github.com/ramya-rao-a/go-outline FAILED

Installing github.com/acroca/go-symbols FAILED

Installing golang.org/x/tools/cmd/guru FAILED

Installing golang.org/x/tools/cmd/gorename FAILED

Installing github.com/fatih/gomodifytags SUCCEEDED

Installing github.com/josharian/impl FAILED

Installing github.com/rogpeppe/godef SUCCEEDED

Installing sourcegraph.com/sqs/goreturns FAILED

Installing github.com/golang/lint/golint FAILED

Installing github.com/cweill/gotests/... FAILED

Installing github.com/derekparker/delve/cmd/dlv SUCCEEDED



D:\Program Files\Redis>"D:\Program Files\Redis\redis-server.exe" "D:\Program Files\Redis\redis.windows.conf"

set GOPATH=F:\project\zeus;F:\project\server

go install Login

go install Center

go install IDIPServer

go install Lobby

go install Match

go install Room

go install Gateway

Lobby.exe.infolog.2017.11.14.23

Center.exe

Gateway.exe

Room.exe

Match.exe

Lobby.exe

IDIPServer.exe

Login.exe

### 配置读取

github.com/spf13/viper

### 日志

<https://github.com/cihub/seelog>

### CPU

github.com/shirou/gopsutil/cpu

### 定义全局变量

var (

    // redis连接池

    pool \*redis.Pool

    // 给服务器间同步状态使用的redis连接池

    poolForServer \*redis.Pool

    // 单实例redis连接池

    poolForSingleton \*redis.Pool

)

### 常量

const (

    serverPrefix = "server:"

    servers = "servers"

    serverInfoChannel = "servernotify"

    // HelloMsg 当服务器注册成功时, 向所有在线服务器广播的消息

    HelloMsg = "hello"

    // UpdateMsg 当服务器更新信息时, 向所有在线服务器广播的消息

    UpdateMsg = "update"

)

### 类型转换

//将ie转换为iEntityCtrl

ie.(iEntityCtrl)

## Events.fire怎么导给lua的

Lua中的Events.fire完全是独立的一套，跟C#完全没关系

## Svn

set GOPATH=E:\timefire\zeus;E:\timefire\server

go install Login

go install Center

go install IDIPServer

go install Lobby

go install Match

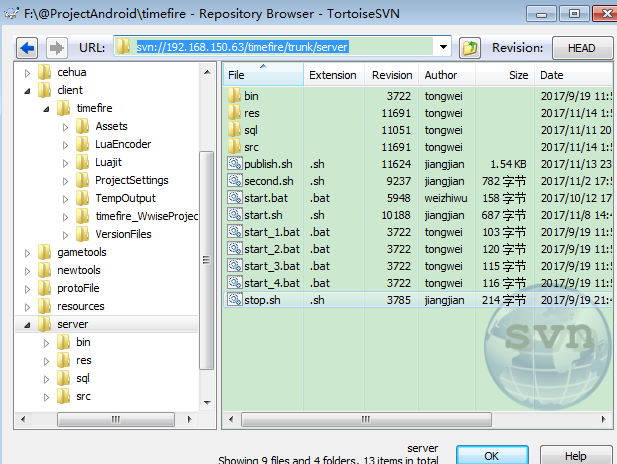
go install Room

go install Gateway

svn://192.168.150.63/mb/zeus/branch\_1

svn://192.168.150.63/timefire/trunk/server

svn://192.168.150.63/mb/zeus/branch\_1



## Terrain

## .asset

## 四元数

Quaternion shootRota = Quaternion.LookRotation(endPoint - startPoint);

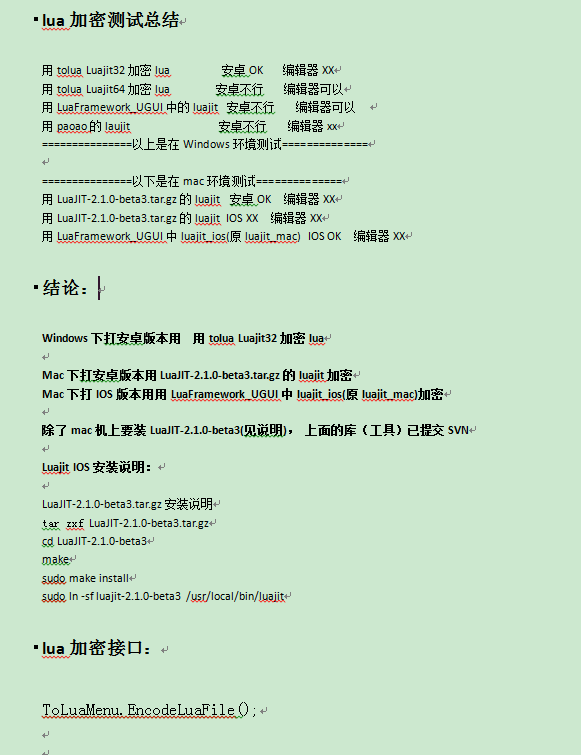
float yaw = shootRota.eulerAngles.y \* Mathf.Deg2Rad;

## Particle

particle.Stop();

particle.Play();

## Lua加密



## 安卓打包

安卓做测试包流程：

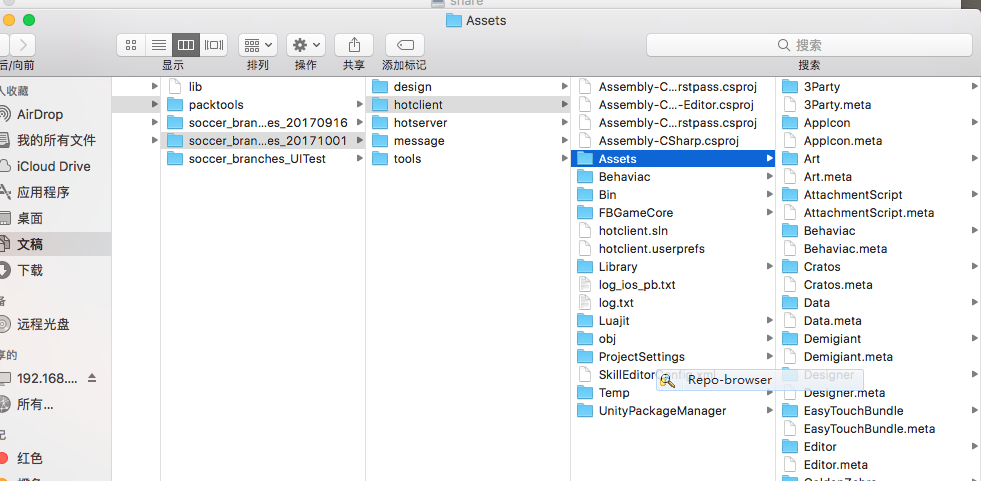
1、菜单Tools/一键打包安卓资源

2、BuildingSettings 打包

即可







## Mac mini

192.168.93.62

## 垃圾桶

192.168.96.231

soccer ztgame@123

## Luajit IOS环境

LuaJIT-2.1.0-beta3.tar.gz

tar zxf LuaJIT-2.1.0-beta3.tar.gz

cd LuaJIT-2.1.0-beta3

make

sudo make install

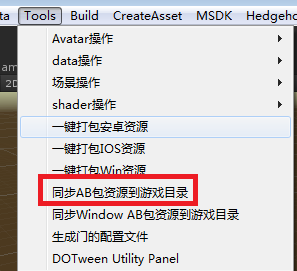
## 截屏

Ctrl+Shift+A

## lua

LuaProxy





## .asset

AssetDatabase.CreateAsset

## CG

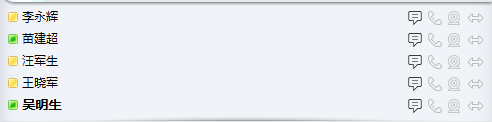
Handheld.PlayFullScreenMovie

## SVN

王晓军 14:13

svn://192.168.150.63/timefire/trunk/client

wumingsheng 12345678



## Git常用命令

### Push:

同步到服务器

git.exe push --progress "origin" Branch\_client20171019:Branch\_client20171019

### fetch:

fetch只拉取不合并

git.exe fetch -v --progress "origin"

### Pull:

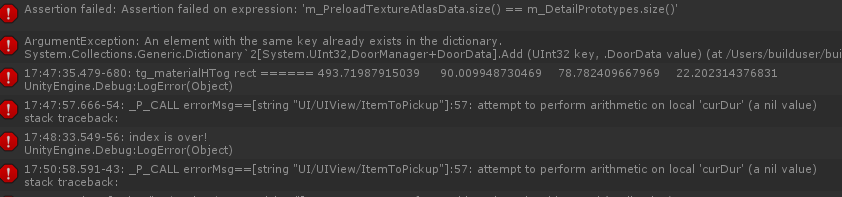
# Client

## 美术规范



## 功能

### 自动拾取



Events.fire("RefreshPickupItemList")

function M:refreshArmor()

local mainEntity = GameSceneData:getMainEntity()

if mainEntity == nil then

return

end

local i = 1

local prop = mainEntity:getProperty(Enum.PropName.HeadProp)

if prop ~= nil and prop.baseid ~= 0 and prop.reducedam ~= 0 then

self.armors[i]:refresh(prop.baseid, prop.reducedam, prop.maxreduce)

i = i + 1

end

prop = mainEntity:getProperty(Enum.PropName.BodyProp)

if prop ~= nil and prop.baseid ~= 0 and prop.reducedam ~= 0 then

self.armors[i]:refresh(prop.baseid, prop.reducedam, prop.maxreduce)

i = i + 1

end

prop = mainEntity:getProperty(Enum.PropName.BackPackProp)

if prop ~= nil and prop.baseid ~= 0 then

self.armors[i]:refresh(prop.baseid, -1)

i = i + 1

end

for j = i, 3 do

self.armors[j]:refresh(0, -1)

end

end

**PackageArmor.lua //防具**

if self.lightImg then

if itemProp.hightLight then

self.lightImg:show()

else

self.lightImg:hide()

end

end

没装备枪械，配件和子弹不需要添加标记



已经有两把枪，且未装备手枪，地面枪械，不添加标记



你这bug太多了，看文档自己挨个处理吧

PackageData.weaponItemId

PackageData.secondWeaponItemId

高亮？：

updateItemHighlightState



**table.insert(self.items, { baseid = baseid, id = id, boxid = boxid, num = num, reducedam = reducedam })**

RankingListBgFrameName = {

[1] = "record-JL\_hengtiao01",

[2] = "record-JL\_hengtiao01",

[3] = "record-JL\_hengtiao01",

[4] = "record-JL\_hengtiao01",

}

Enum.ItemType = {

None = 0,

MinWeapon = 1,

Melee = 9,

MaxWeapon = 10,

Consume = 11,

Armor = 12,

Component = 14,

SupplyBox = 15,

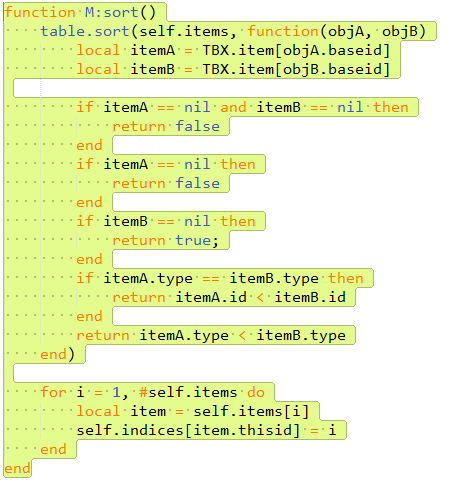
Bag = 16,

Ride = 19,

Bullet = 20,

Throw = 21,

}



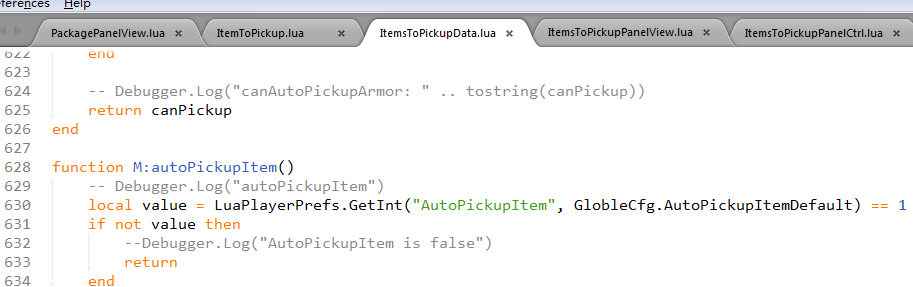
ItemsToPickupData

ItemsToPickupPanelView

ItemsToPickupPanelCtrl

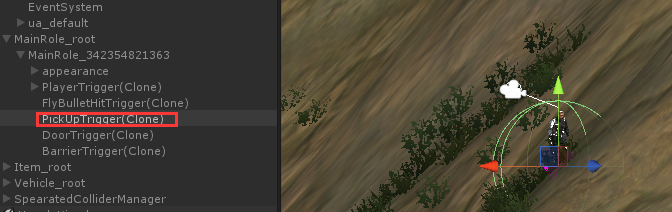
ItemToPickupCtrl

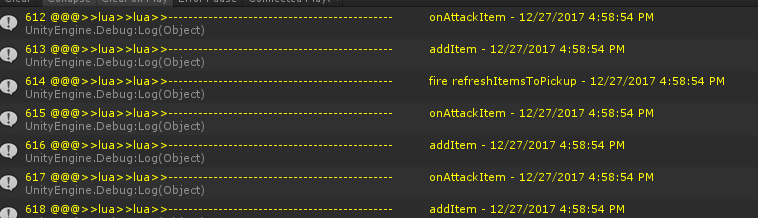
ItemToPickup



### 拾取列表

正常：





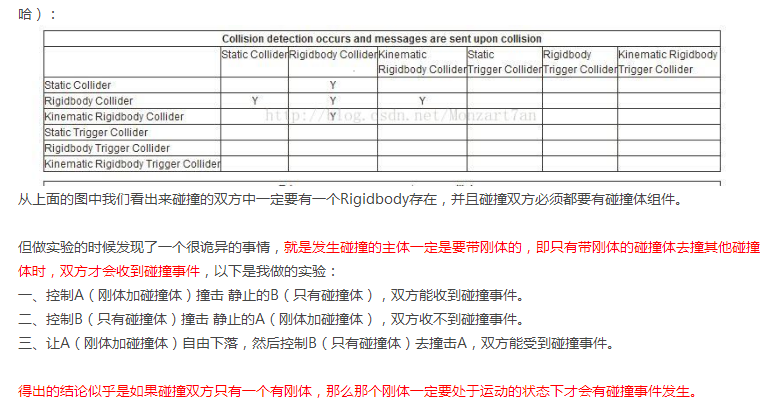


### 碰撞

VehicleTriggerCtrl

ScriptPrefab/CarrierTrigger.prefab(包含VehicleTrigger)

<http://blog.csdn.net/monzart7an/article/details/22739421>

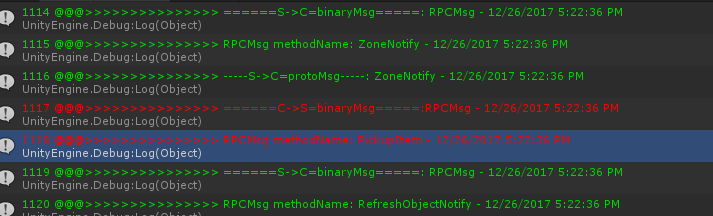




### 背包优化

ItemsToPickup

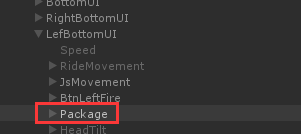




1401

1413

1414



RPC\_RefreshPackCapacity

丢弃面板跟主包裹同一个预制，但是不同的lua文件 (PackagePanel,DropPanel， PackageItemCtrl， PackageData

Events.add("RPC\_RefreshPackCellNotify", self, "onRefreshPackCellNotify")

**=============================================================================**

UISlider

**self.sd\_Hp = self:getSlider("Root/Other/Hp")**

**颜色变化：**

local color = Color.While

if percent < 1.0 and percent >= max then

color = Color.New(1.0, 1.0, 1.0, 1)

elseif percent < max and percent >= min then

color = Color.Lerp(Color.New(255.0 / 255.0, 0.0 / 255.0, 0.0 / 255.0, 1), Color.New(1.0, 1.0, 1.0, 1), percent)

elseif percent < min then

color = Color.New(255.0 / 255.0, 0.0 / 255.0, 0.0 / 255.0, 1)

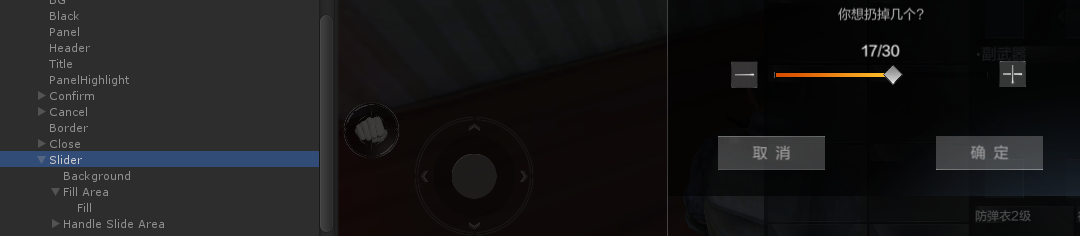
else

color = Color.New(1.0, 1.0, 1.0, 0.6)

end

self.view.hpImg:setColor(color)





**刷新包裹：**

function M:refreshItems()

if PackageData.cellCount > #self.items then

### 撞碎栅栏



生成栅栏的配置文件会自动生成： 栅栏的ID

**播放特效：**

GameObject go = EffectResource.instance.loadGameObject("Effects/" + owner.shootEffect, shootPoint.transform);

Barrier

**BarrierTrigger （主角上的碰撞体）**

DoorTriggerCtr **（主角上挂的）**

DoorTrigger （碰撞体 由DoorTriggerCtr生成）

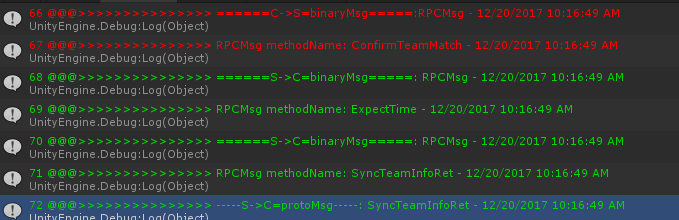
DoorEditor (编辑器)

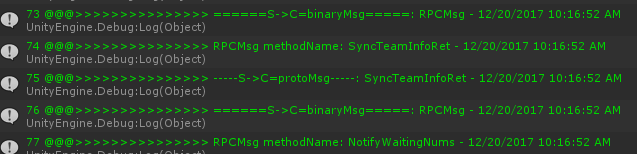
DoorManager （管理器）

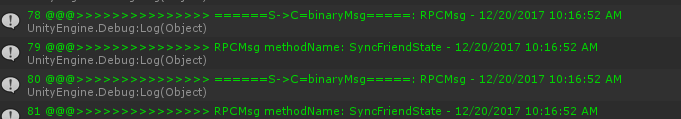
DoorView **(每个门挂的脚本)**

### 组队流程

**不主动邀请直接匹配:**

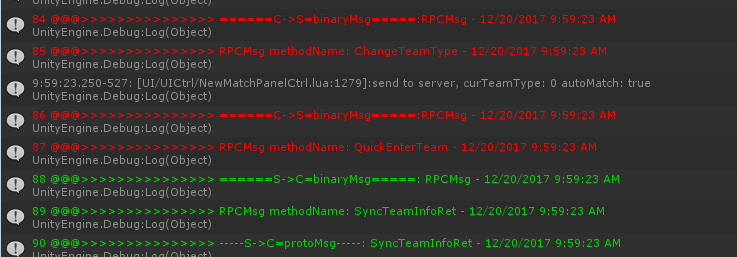






**组队流程： （绿色服务器发给客户端 红色客户端发服务器）**

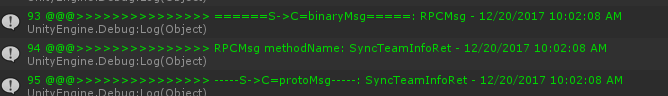
选择双人组队：



邀请好友：

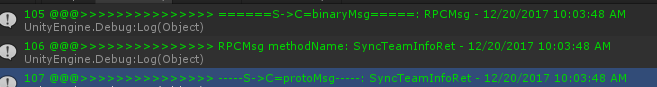


被邀请的人同意后：



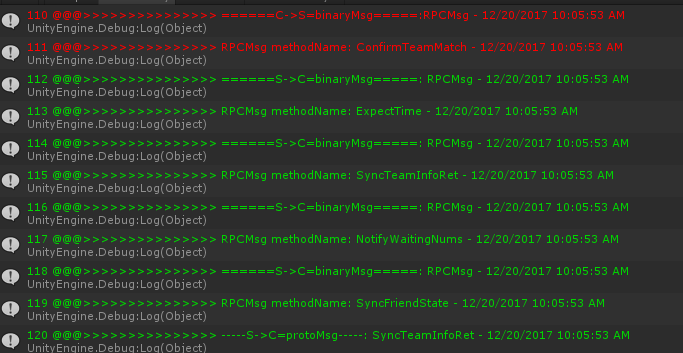


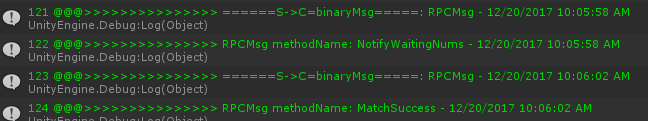
被邀请的人点击”准备”：





点击开始匹配：





### 收发消息

Zeus.inst.postToSpace

### CPU

Profiler.BeginSample()

LuaLooper.LateUpdate()

UpdateBeat:Add

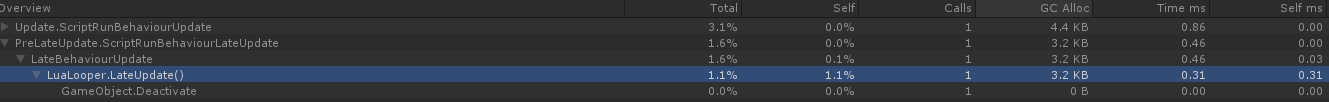
LateUpdateBeat:Add

FixedUpdateBeat:Add

updateTeamMemeberHp 有GC

* object 返回值会造成20B的GC

 有GC？



**下面语句有GC:**

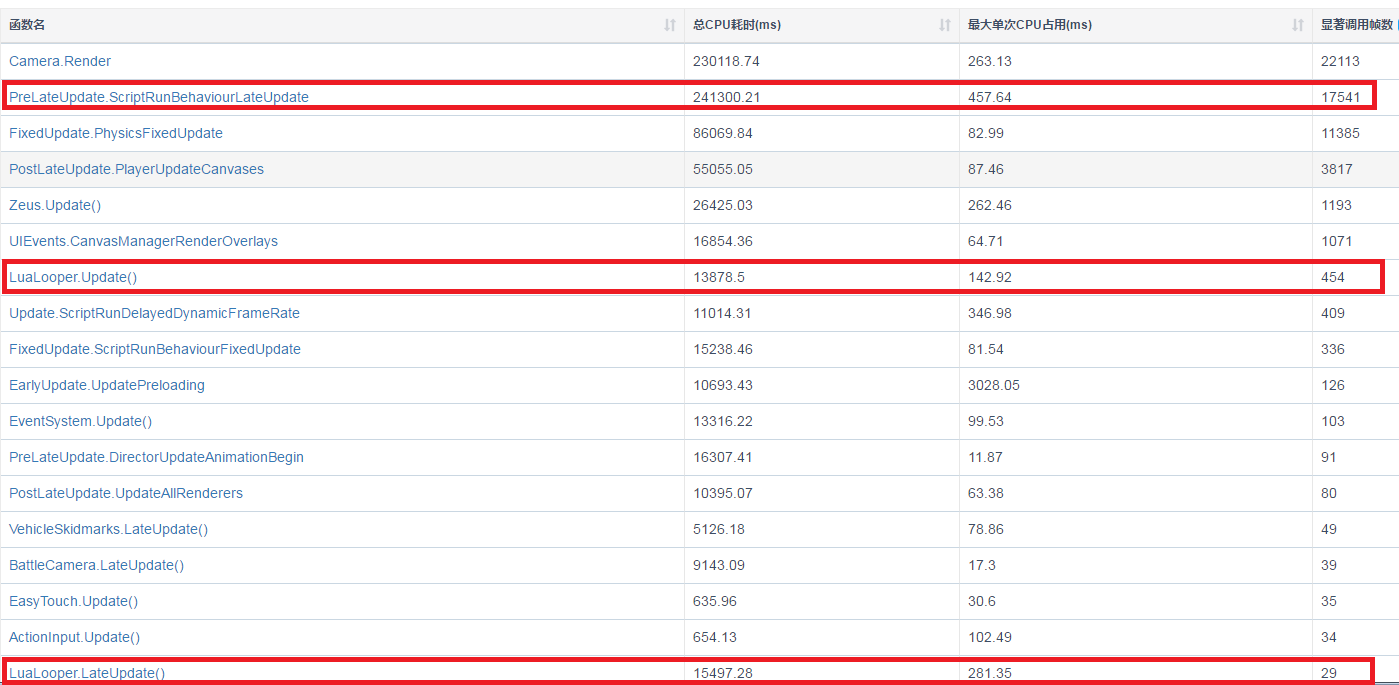
**member.hp = entity:getProperty(Enum.PropName.HP);**

**local property = LuaBridge:getEntityProperty(self.id,propName)**

LuaBridge\_Scene.**getEntityProperty**

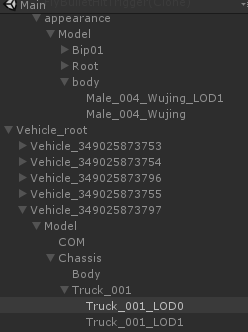
String作为参数传给C#会照成GC C#返回string好像不会有GC

C# 返回object会有GC

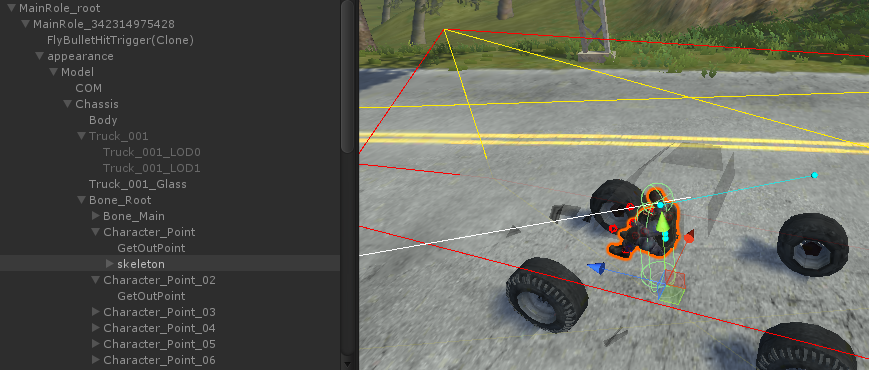


### 载具

进入前



进入后：



### 资源释放

UIManager UIBridge UIProxy

NewMatch

Rike04\_big





### 回退版本

Patcher.cs 用的zip库

### 内存

adb shell dumpsys meminfo com.tencent.cldts

### 动态秘钥

public byte[] ComputeHash(byte[] buffer);

MD5 md5 = new MD5CryptoServiceProvider();

byte[] retVal = md5.ComputeHash(file);

using System.Security.Cryptography;

clientTokenInfo.Token = "20e809c26ce362f27bd6e5ec1280668d"

### 渠道号

AccountInfo:getQQVip()

Enum.QQVip.None

Enum.QQVip.Vip

Enum.QQVip.SVip

LoginData:getPlatform()

DKConfig.PlatformType.WeiXin

SDKConfig.PlatformType.QQ

LoginData:getGameCenter()

Enum.GameCenter.QQCenter

Enum.GameCenter.WeChatCenter



73213123

@@@ SetOriginChannel: 73213123

**要清理的：**

 **最后一行**

11-27 17:13:22.400 31048 31064 I Unity : @@@Application.persistentDataPath: /storage/emulated/0/Android/data/com.tencent.cldts/files

/storage/emulated/0/Android/data/com.tencent.cldts/files/Patch

Files/**Patch**/lua

SdkManager:WGOpenUrl(url, Msdk.eMSDK\_SCREENDIR.eMSDK\_SCREENDIR\_LANDSCAPE)

local str = LuaPlayerPrefs.GetString(tostring64(AccountInfo.uid).."CareerData","")

if not IsNilOrEmpty(str) then

local jd = json.decode(str)

..

End

Util.getOriginChannel()

### 资源优化

账号：ahsdwjs@163.com密码：wjs7632466

Xcode instruments

memoprofile

### 腾讯服务条款

**参考界面：**

CareerPanel

SettingsPanel

UI.close("LoadingPanel")🡪UIManager:close(panel)

**管理器UI.lua**

打开一个界面：

UI.show("LoginPanel") 🡪 UIManager.show()

UI.show("LoginNoticePanel")

每个UI预制都挂了一个UIProxy负责连接对应的luaCtrl和luaView

local enableAutoPickup = LuaPlayerPrefs.GetInt("AutoPickupItem", 0)

Toogle:

self.tg\_AutoPickTg:onStateChanged(function(isOn)

local value = isOn and 1 or 0

LuaPlayerPrefs.SetInt("AutoPickupItem", value)

Events.fire("onAutoPickupItemChanged")

end)

Toogle:

self.view.tgg\_TabBtns:addToggleEvent("Career",self.view.tg\_CareerInfo,function(isOn)

if isOn then

self.view.img\_CenterPanel:show()

CareerData:loadCareerData(2, ulongToNumber(AccountInfo.uid))

else

self.view.img\_CenterPanel:hide()

end

end)

self.view.tgg\_TabBtns:addToggleEvent("Friend",self.view.tg\_FriendRank,function(isOn)

if isOn then

self.view.img\_FriendPanel:show()

CareerData:loadFriendRank(2, ulongToNumber(AccountInfo.uid))

else

self.view.sr\_ScrollView0:clear()

self.view.img\_FriendPanel:hide()

end

end)

self.view.tgg\_TabBtns:setAllTogglesOff()

self.view.tgg\_TabBtns:switchTogglePage("Career")

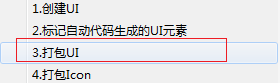
### UI

\*\*\*\*\*每个UI都有一个场景\*\*\*\*\*

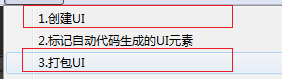
\*\*\*\*\* 通过场景编辑UI\*\*\*\*\*

Ctrl+S 自动生成和保存lua (程序用)

修改老的UI： 打开原有的场景，(策划美术用)



创建新的UI： 创建UI🡪打包UI ,(策划美术用)



\*\*\*\*要导出的控件

LoginPanelCtrl 登陆

NewMatchPanelCtrl 匹配界面

CameraPosionPanel 冒红？

MainPanel 战斗主界面

PackagePanel 背包

MapPanel 地图

EWNSPanel 方向刻度

LenXeightPanel 超级放大镜

LenXfourPanel 放大镜

LenHoloPanel 放大镜

LenRedPointPanel放大镜

SettingsPanel 设置

CustomBattlePanel 自动以操控？

### 登陆

Gateway.inst.login(accountName, passwd, data, loginURL, channel);

### ~~~~~~~~~打消息表格

客户端生成消息: newtools/protoGen/game.bat

服务端生成消息：protoFile/gen.py

打表：newtools/excelGen/genAll.bat

结构体消息：

ClientVertifyReq

**接收消息：**

(字节流)🡪(ID+字节流)🡪

RPCMsg消息只是一个特殊的结构体消息

Lua消息解析：

PARSE\_PROTO\_MSG

PARSE\_BIN\_MSG

PARSE\_HTTP\_MSG

onLuaEventParse

Lua消息发送：

function SEND\_PROTO\_MSG(msg)

local buffer = msg:SerializeToString()

local msgName = msg:GetTypeName()

Cratos.Zeus.inst:post(msgName, buffer)

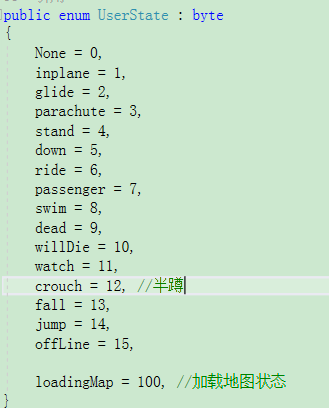
print("SEND\_PROTO\_MSG:" , "==protoMsgName==" , msgName)

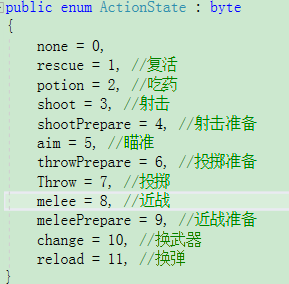
end

SEND\_RPC\_MSG

Cratos.Zeus.inst:rpc

### 状态





### OnEnterAOI

Space\_AOI

🡪GameScene\_Entity.oEnterAOI

🡪LuaBridge.onEnterAOI

🡪GameSceneData.lua OnEnterAOI

## 登陆

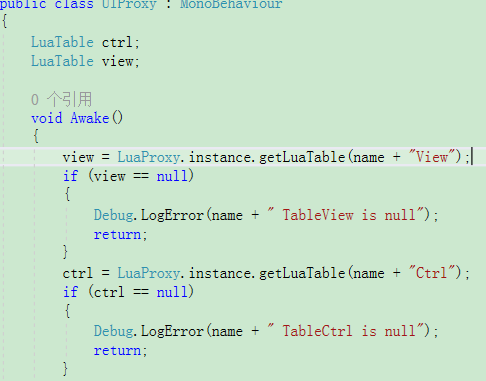
Debugger.LogError("the avatar is invalid", avatarid, userid)

## lua

用的tolua

Lua调用：

* main.lua
* 每个UI资源都挂了一个UIProxy， UIProxy负责关联对应的lua view和ctrl



## 初始化

### lua

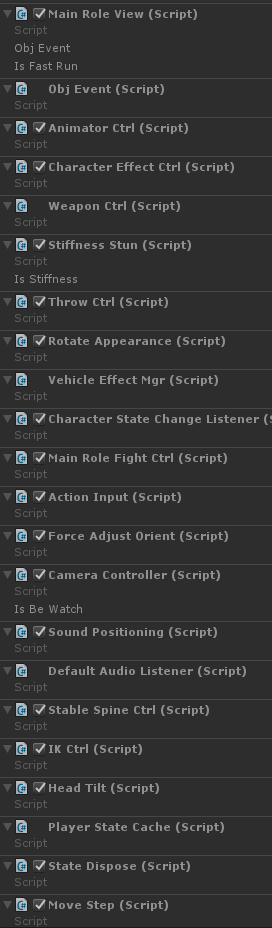
luaclient

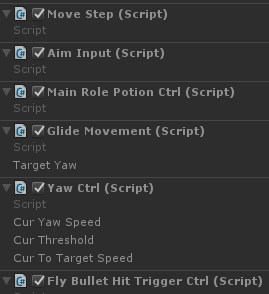
Game.Oninit🡪 LuaProxy.create()🡪LuaLoader(从luaclient继续)

## 动作

### 主角上挂的组件







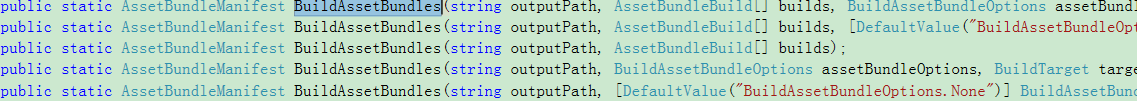
### 主角下面的RoleJumpCloud

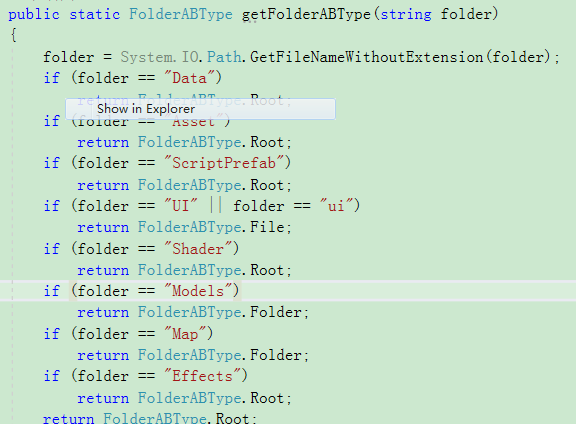
下降过程中的云效果

### GlideCameraController

人物下降过程中上下震动

## 资源打包、管理、释放





整个RawResources\Asset 打成一个ab包

整个 RawResources/Data打成一个ab包

整个RawResources/ScriptPrefab打成一个ab包

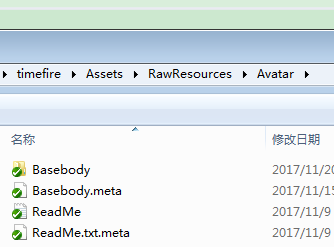
整个RawResources/Effects打成一个ab包

整个RawResources/Shaders打成一个ab包

RawResources/Models 对里面的每个叶子文件夹打一个ab包

RawResources/Avatar 对里面的每个叶子文件夹打一个ab包

RawResources/map对里面的每个叶子文件夹打一个ab包HapplyKingdom除外？？



整个basebody打成一个AB包？？

<https://www.uwa4d.com/u/overview.html>  
账号：ahsdwjs@163.com密码：wjs7632466

Resources.UnloadUnusedAssets();

**加载:**

Resources.Load

UnityEditor.AssetDatabase.LoadAssetAtPath

AssetBundleCreateRequestStatus

AssetBundle.LoadFromFileAsync(fullPath);

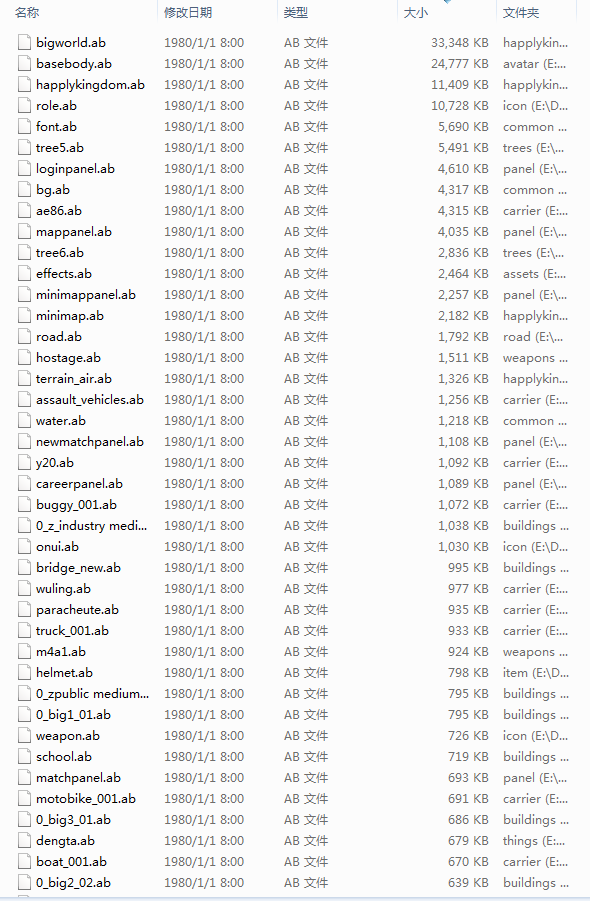
AssetBundleRequest req = loadedAB.LoadAssetAsync<T>(filePathInBundle);

while (!req.isDone)

{

yield return null;

}



## 场景

**异步加载(Unity接口)：**

SceneManager.LoadSceneAsync

**清除所有未使用资源(Unity接口)：**

Resources.UnloadUnusedAssets()

场景异步加载进度：

var sceneprocess = onSceneProcess(\_asycOperation);

### 主场景

GameScene\_Map

loadScene

🡪loadSceneAsync

🡪SceneManager.LoadSceneAsync(Unity)

编辑器下主场景(战斗场景)直接调用LoadSceneAsync加载

手机上先加载场景依赖的AB包，最后再通过LoadSceneAsync加载场景



## 角色动画

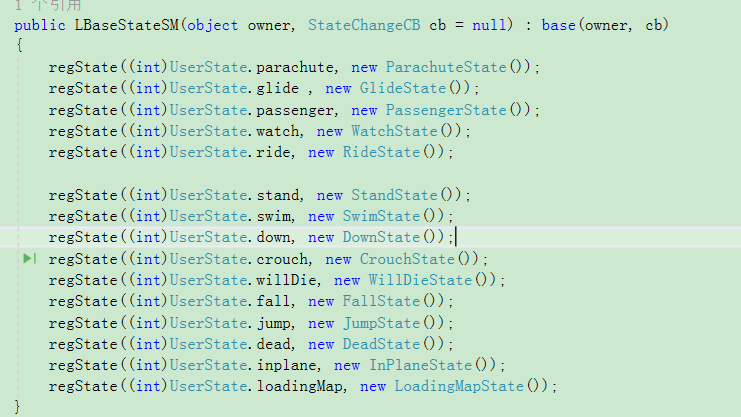
NormalStateBase

CharacterController.move

Rigidbody <->collide

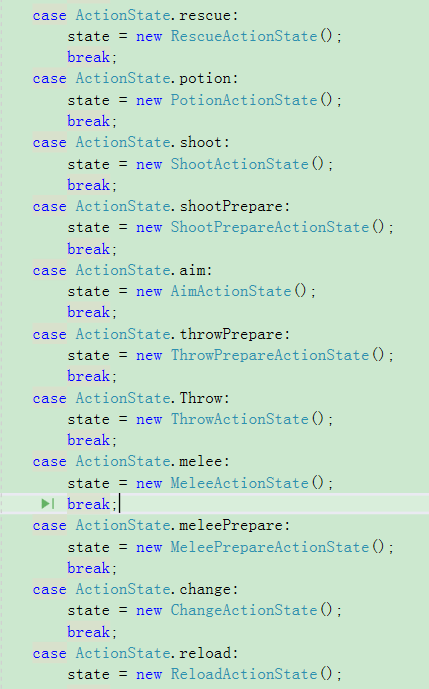
### 状态机

BaseState bsm = new LBaseStateSM(this);

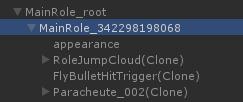


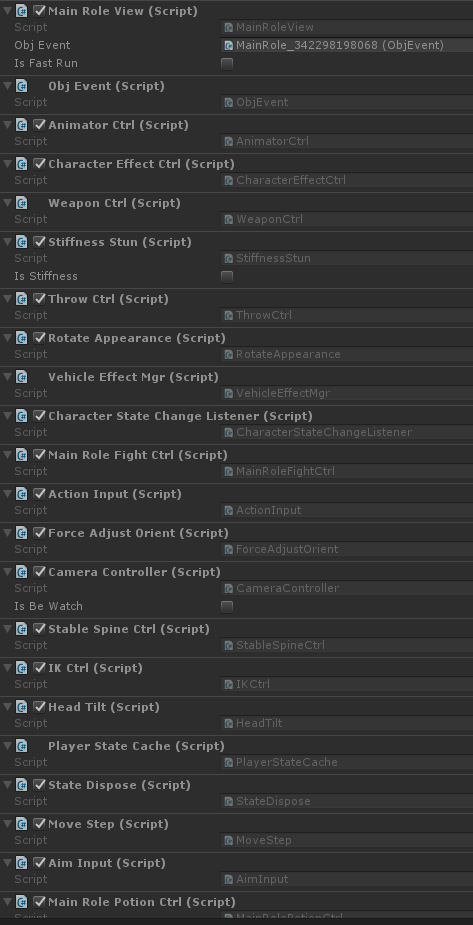
-----------------------------------------------------------------------------------

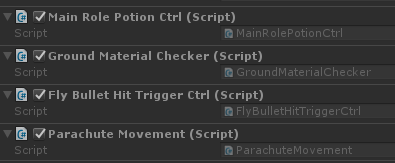
ActionState



### 主角上挂的脚本







## 数据恢复

降落伞

Debug.Log("parachute modle path: " + item.model.ToString());

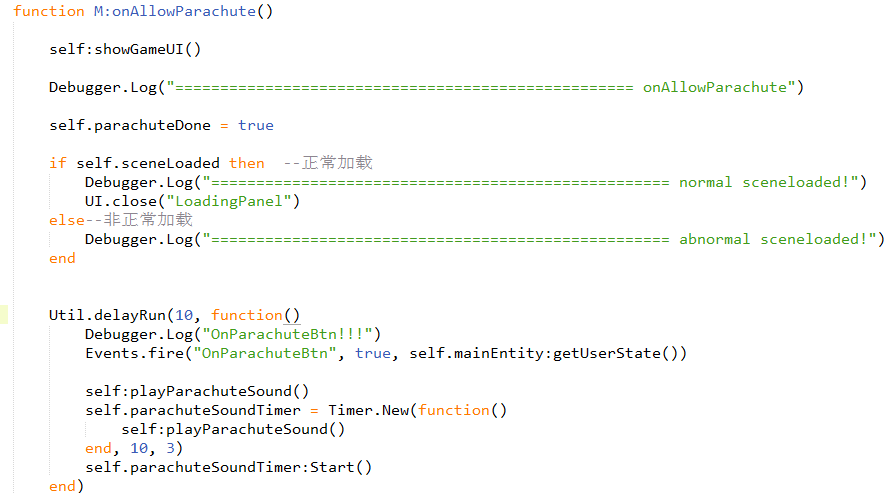
parachuteObj = ModelResource.instance.getGameObject(item.model);

飞机？

### 跳伞关键顺序

* 服务器通知可以跳伞了 （**AllowParachute**）

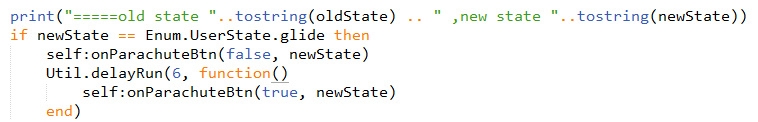
客户端设置10秒后可以跳伞



点击跳伞客户端发送RPC\_**StartParachute**

* 服务器设置状态Enum.UserState.glide

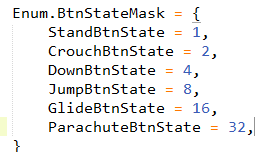
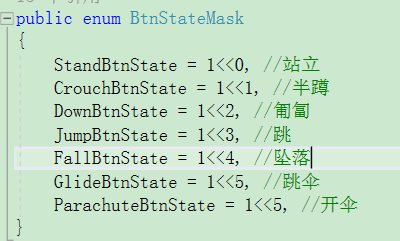
客户端滑翔状态6秒后可以打开降落伞：



点击开伞调用 🡪映射按键

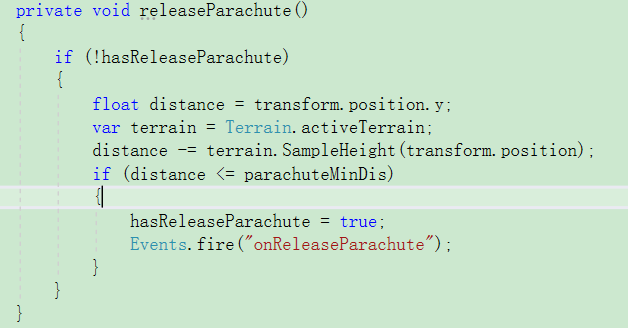
if self.canGlide then

Input.setButtonTriggerMask(**Enum.BtnStateMask.ParachuteBtnState**)

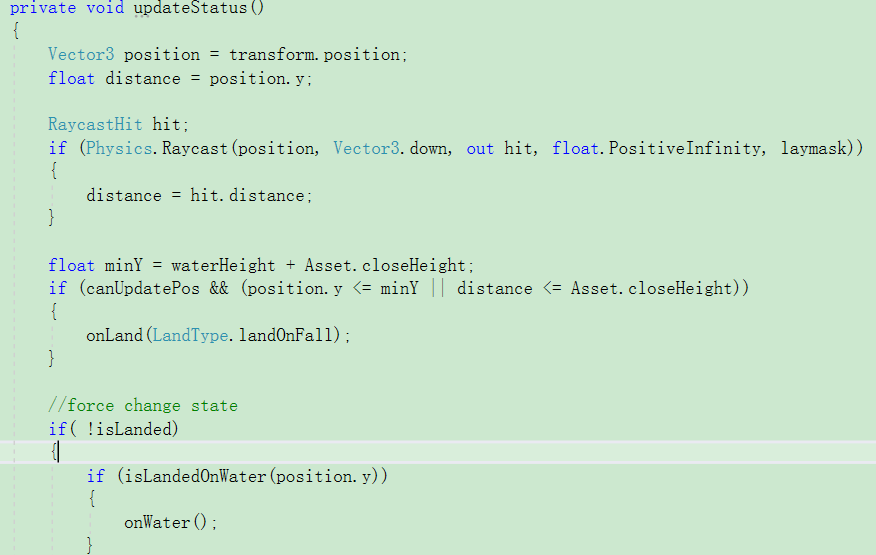
 

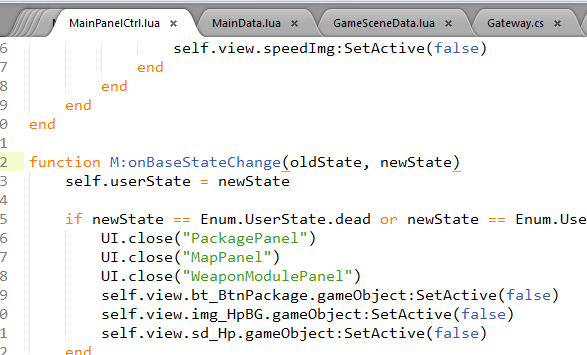
最后通过**SyncUserState （主界面上的几个按键都通过这条消息）发送给服务器**

**低于一定高度自动打开降落伞：**



**客户端自己根据条件关闭降落伞，并切换到不同的状态：**





### 消息序列

---space验证完后—

SetAutoReform (C->S rpc) (自动装备？)

ParachuteReady (C->S rpc) (自动装备？)

-----------------场景加载完毕----------------

SEND\_RPC\_MSG(Enum.ServerType.Scene, "ParachuteReady")

SEND\_RPC\_MSG(Enum.ServerType.Scene, "SpaceLoadCompleteByTeam")

🡺RPC\_SyncSpaceType (组队信息包含spaceid，用于语音房间初始化？)

**🡺AdjustUserState (位置、方向、时间戳, 通过这条消息进入InPlaneState 处理后转化为**PlayerStateAndInput(里面有2个状态机的index)**)**

🡺UpdateAliveNum (活着的玩家数量)

🡺UpdateAirLeft (舱内剩余人数)

🡺AllowParachute (允许跳伞了)

点击跳伞按钮RPC\_StartParachute (C🡪S)

🡺 ForceEjectNotify (强制跳伞通知)

🡺 RPC\_ParachutePos (self.isFinishParachut = true?)

🡺RPC\_ForceEject (强制跳 mainpanel.view.img\_leftNum:hide()？)

🡺 AdjustUserState (EntityState 主角位置方向)

🡺 AOISyncUserState (同步9屏玩家状态)

🡺 EntityAOIS (9屏玩家 EnterAOI onLeaveAOI)

🡺RPC\_ ZoneNotify (地图毒圈轰炸圈等相关)

🡺RPC\_ShrinkRefresh (毒圈收缩时间？)

🡺ZoneNotify(proto) (区域通知)

SyncUserState (C🡪S)

🡺 RPC\_RefreshBoxStart (空投)

🡺RPC\_ChatNotify (公告？)

🡺RPC\_BomRefresh(轰炸区刷新)

🡺RPC\_BombDam (播放炸弹效果) （N个）

🡺RPC\_RefreshBoxNotify (开始刷新空投道具)

🡺RPC\_AdviceNotify (公告、建议、错误提示？)

🡺RPC\_OnDamage (被击（死了？）播放被击特效)

🡺 PropSyncClient (单个实体属性同步)

🡺RPC\_DieNotifyRet (死亡通知（公告）)

🡺MapCharacterResultNotify (战斗结果？)

点返回 rpc\_leaveSpace (c->S)

### 消息

包裹数据刷新：

RPC\_RefreshPackCellNotify

EntityAOIS🡪EnterAOI

**9屏数据?**

EntityAOIS

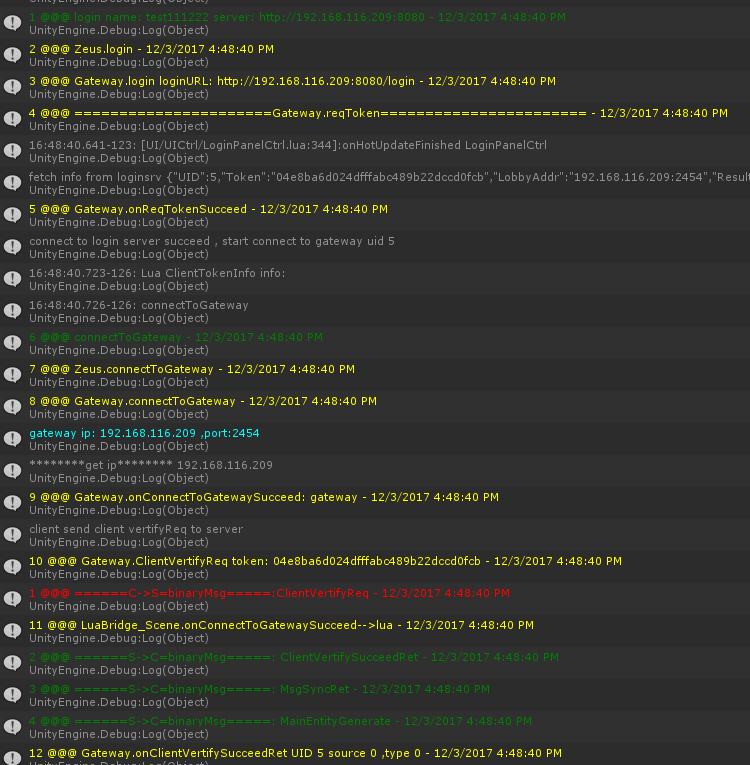
EnterAOI （）

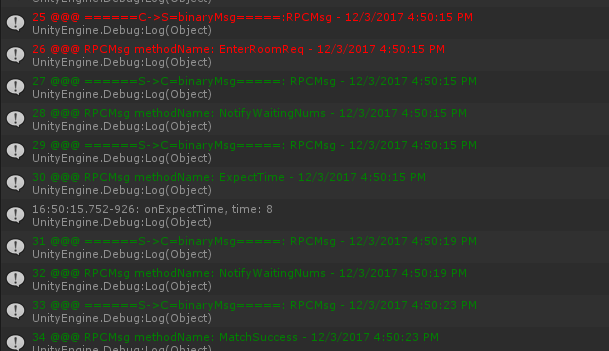
SyncUserState (C->S 频繁)

AdjustUserState（S🡪C）

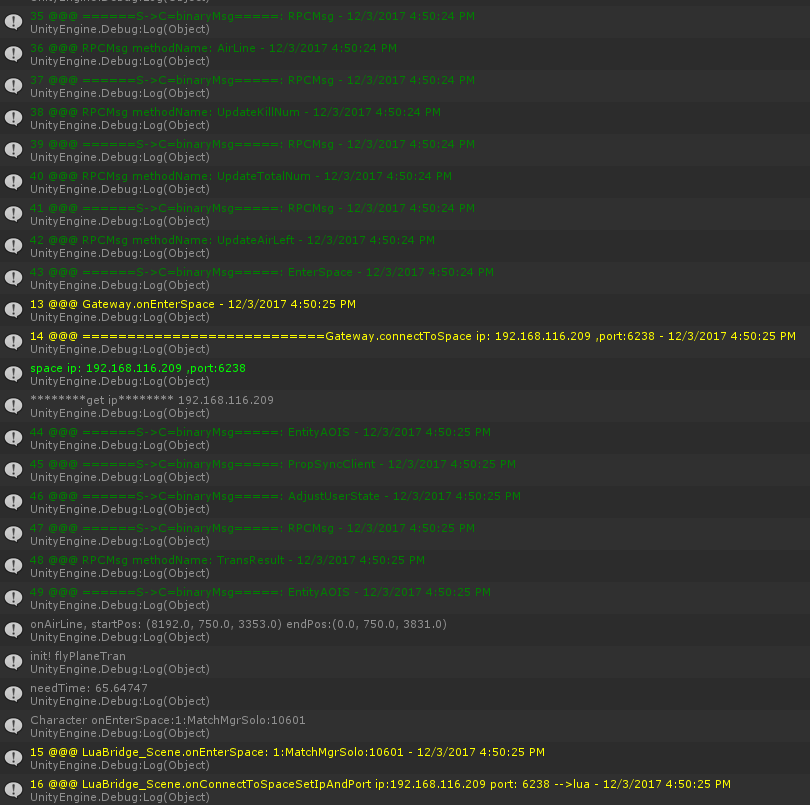
同步用户状态：（频繁）

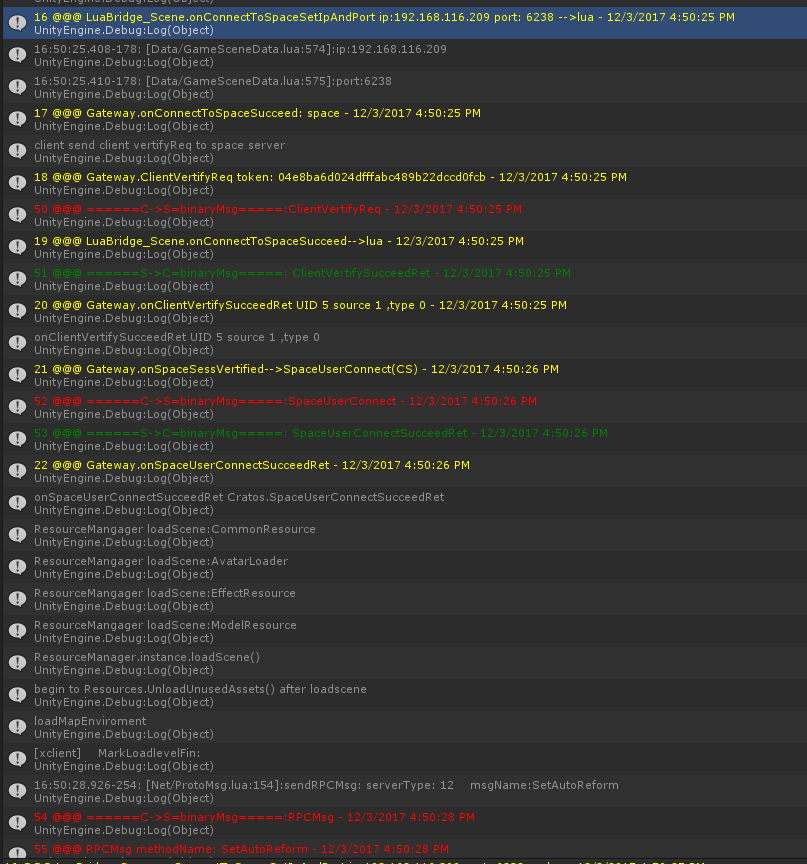
AOISyncUserState



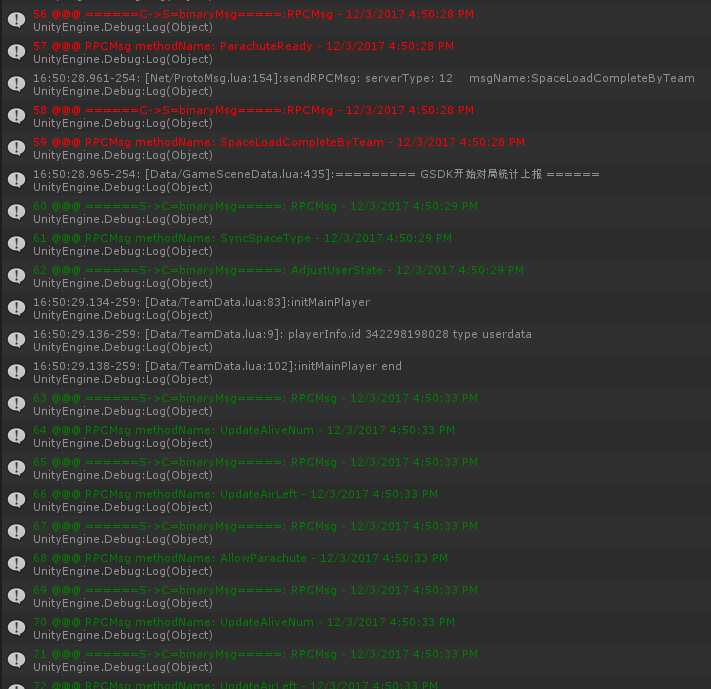


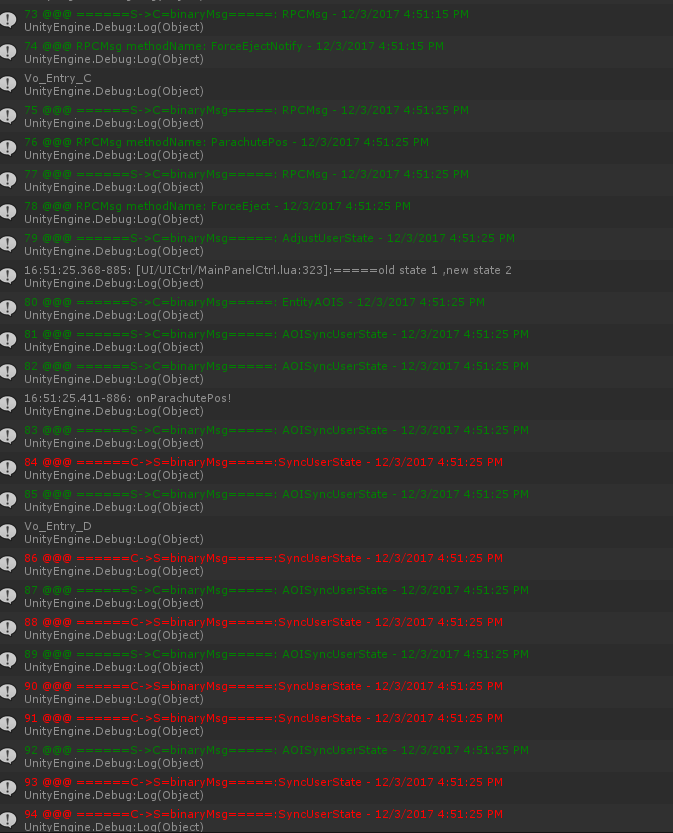
### MatchSuccess

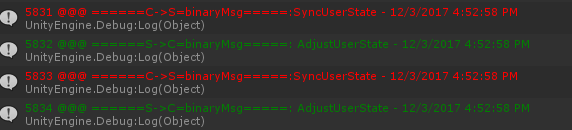


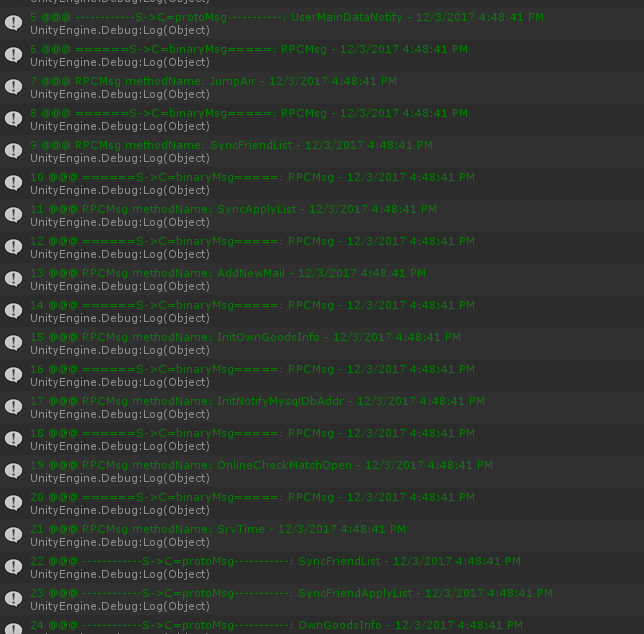


### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*降落从上面红字开始\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*









### 飞机飞

### 跳伞

## 断线重连todo

### 恢复到飞机上

🡺Events.add("RPC\_AirLine", this, "onAirLine");

🡺RPC\_UpdateKillNum

🡺RPC\_UpdateTotalNum

🡺RPC\_UpdateAirLeft

🡺EnterSpace (Space连接(ip)的信息, 也包含地图名,ID时间戳（帧）等信息)

（Gateway、 Space、GameScene）

🡺EntityAOIS (取出里面的很多)

==============9屏信息 （有一系列 EnterAOI LeaveAOI信息） 调用OnEnterAOI、onLeaveAOI

🡺PropSyncClient （状态）（刷新之前缓存的AOI信息，并且刷新这条信息包含的实体属性）

public class PropSyncClient : IMsg

{

public UInt64 entityID; Zeus.inst.getSpace().getMainRoleID()

public UInt32 num; 4

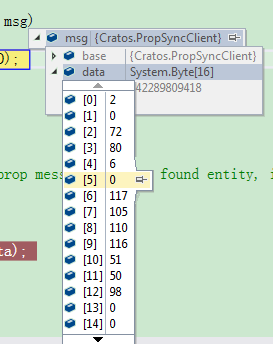
public byte[] data;

key = "SpeedRate" 1

key = "MaxHP" 100

key = "HP" value = 100

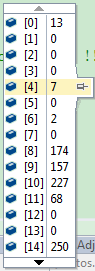
Key = "ChracterMapDataInfo" prop.Value = {protoMsg.ChracterMapDataInfo}



🡺AdjustUserState （属性）(刷新状态信息 位置角度button相机位置，状态机索引)

（view.onStateChange(oldState, newState);）



Gateway.onEnterSpace (连接Space ip)

Gateway.connectToSpace (连接Space ip)

🡺RPC\_TransResult (没有这个处理?)

🡺EntityAOIS

============主角在上面的消息更新（**主角在进网关的时候就创建了**。。。）？==========

Zeus.inst.getSpace().**getMainRole**();

public class EnterAOI : IMsg

{

public UInt64 entityID; // Zeus.inst.getSpace().getMainRoleID();

public string entityType; // "Player"

public byte[] state;

public UInt16 propNum; // 17

public byte[] properties;

}

**//-----------------** public byte[] properties **---------------属性**



"HeadProp" protoMsg.HeadProp 里面都是0

"SpeedRate" 1



"BackPackProp" protoMsg.BackPackProp 0

["SlowMove"] 0

["ParachuteID"] 1610

"MvSpeed" 0

key = "GunSight" 0

key = "Fastrun" 0

Key = "ChracterMapDataInfo" Value = {protoMsg.ChracterMapDataInfo} **//玩家地图数据**

(level =1 uid=1 pos 3.4E38)

Key = "BodyProp" prop.Value = {protoMsg.BodyProp} 0

key = "MaxHP" 100

key = "RoleModel" 2

Key = "AccessToken" ""

key = "HP" 100

Key = "VehicleProp" prop.Value = {protoMsg.VehicleProp} //载具信息(位置速度旋转) 0

Key = "AimPos" prop.Value = {protoMsg.Vector3} 0

mainRole.unPackProps(msg.propNum, msg.**properties**);

**//----------------- public byte[] state --------------- 对应PlayerStateAndInput**

**包含 位置、角度信息**

pos (2302.7, 500.0, 5457.1)

baseState 100

actionState 0

**cameraPos** (0.0, 0.0, 0.0)

button 0

wheelFRota (0.0, 0.0, 0.0)

**wheelBRota = "(0.0, 0.0, 0.0)"**

mainRole.initState(msg.**state**);

Zeus.inst.getSpace().**getMainRole**();

ClientVertifyReq

🡺ClientVertifySucceedRet

SpaceUserConnect

🡺SpaceUserConnectSucceedRet

🡺PropSyncClient

🡺AdjustUserState

🡺EntityAOIS

Gateway.onSpaceUserConnectSucceedRet

**========到上面算正式连上SpaceServer----- (isSpaceSessVertified = true)**

航线消息

RPC\_SetAutoReform

**========地图加载完成=========**

RPC\_ParachuteReady

RPC\_SpaceLoadCompleteByTeam

🡺RPC\_SyncSpaceType

🡺 AdjustUserState

🡺RPC\_UpdateKillNum

🡺RPC\_UpdateTotalNum

🡺RPC\_UpdateAirLeft

🡺 AdjustUserState

**RPC\_AllowParachute**

航线消息

Events.add("RPC\_AirLine", this, "onAirLine");

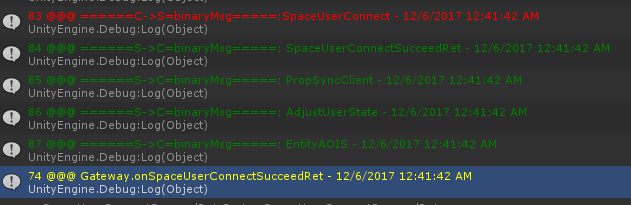




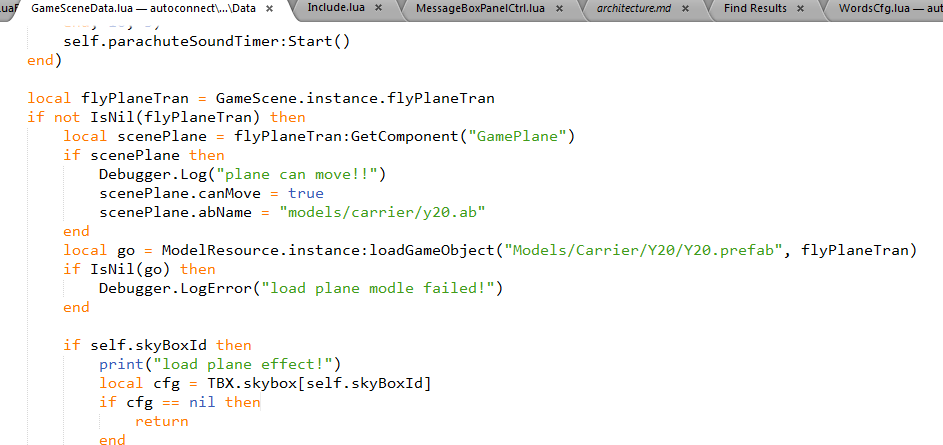
**航线消息在网关就发了**

AdjustUserState消息让在飞机上





飞机飞行处理代码：



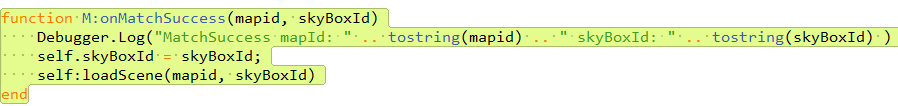
### 恢复到飞行状态

### 恢复到打开降落伞状态

### 恢复到落地状态

### 大断线

匹配成功加载场景：





UI.show("LoadingPanel")

**匹配成功后在网关上发的消息：**



## Todo

飞机飞到地图边缘处理

AdjustUserState

ParachutePos

毒圈

去掉手动clearAOI

红色 客户端给服务器发消息

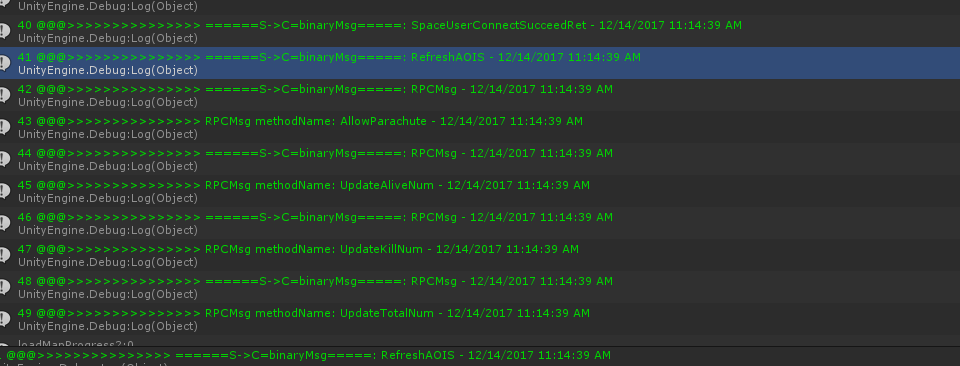
绿色 服务器给客户端发消息

黄色 断线重连

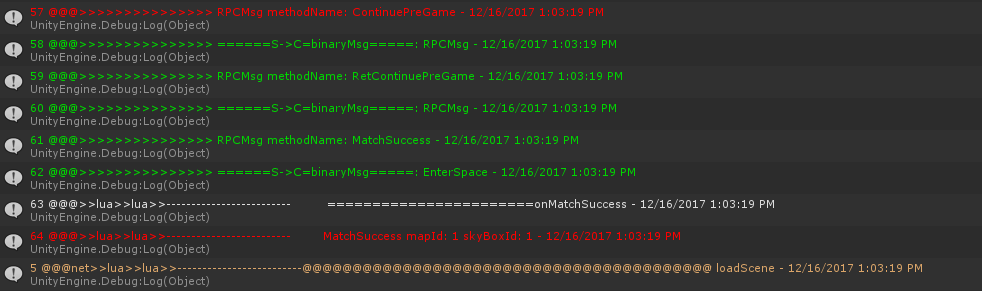
白色 lua一般性的代码日志

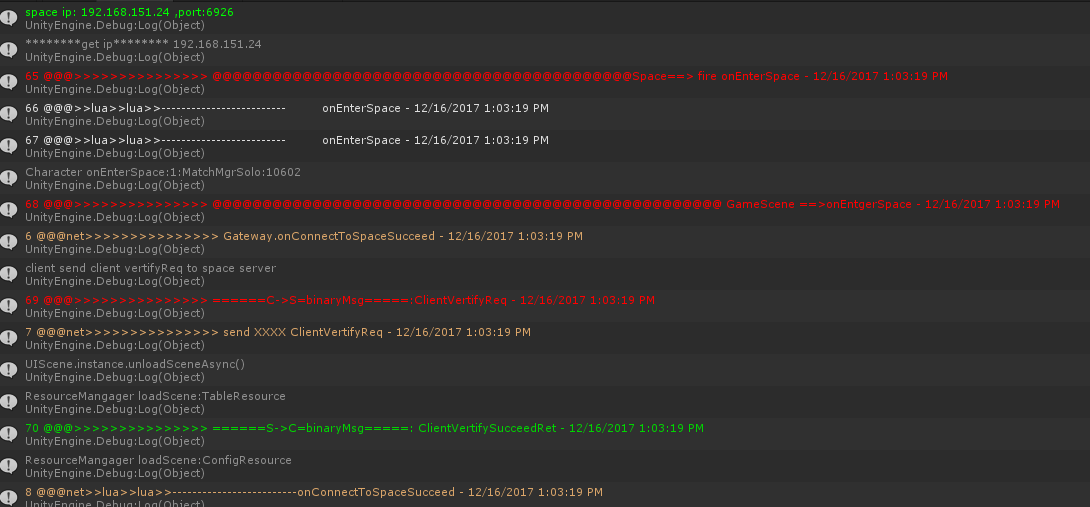
## 断线重连问题

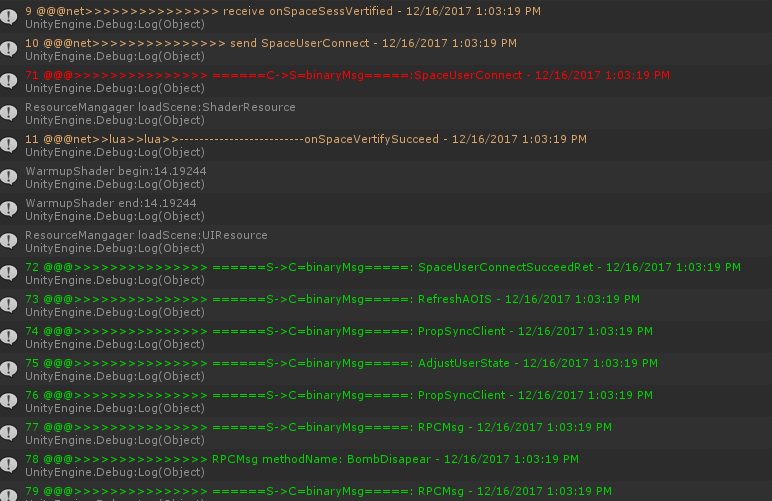
吴明生，李永辉  1.【弱网--调试版本】游戏对局中，断开wifi（USB无线网卡），角色没有立即收到断线的提示，可以在游戏中可以继续奔跑（必现）  ID：57836884  
吴明生，李永辉  2.【弱网-调试版本】游戏对局中，断线重连后（关闭USB无线网卡），己方视角--无法攻击，拾取道具，开门，上车；他人视角--无法拾取，但可以上车，同步载具，并且载具可以被打爆（必现）  ID：57836886  
吴明生，李永辉   9.【断线重连】大厅界面，断开WiFi无法出现断线loading界面（并且长时间也不会出现断线重连提示框）（必现）  ID：57837328   
  
注意一下弱网情况下断线重连应该由心跳超时触发



### 【断线重连】落地后，断网超出1分钟弹出重连提示，返回登陆界面后，连上网络，回到战斗，卡在loading界面（必现

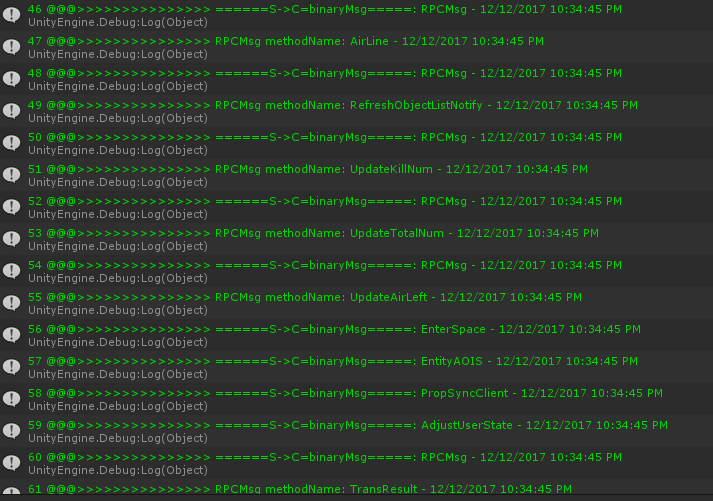


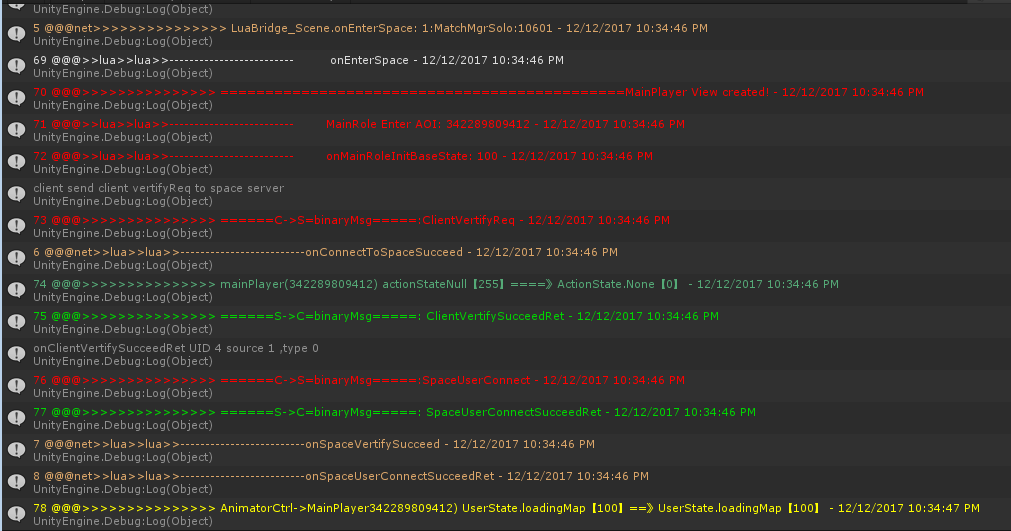


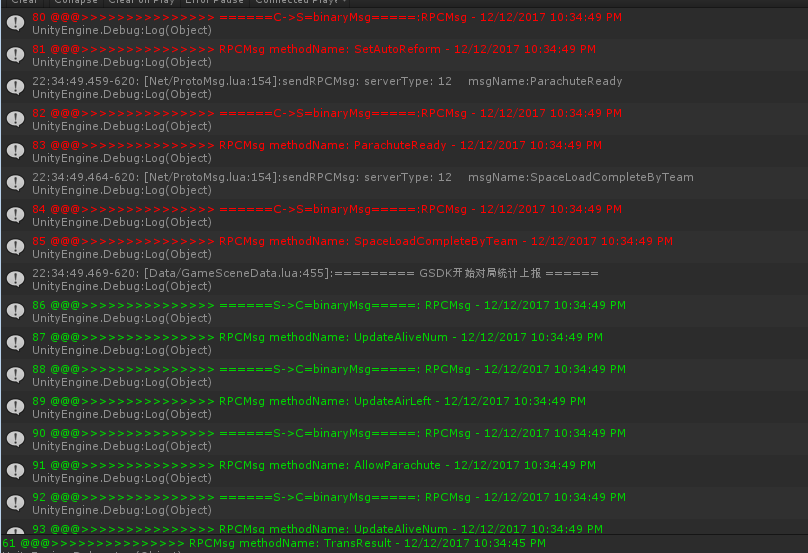


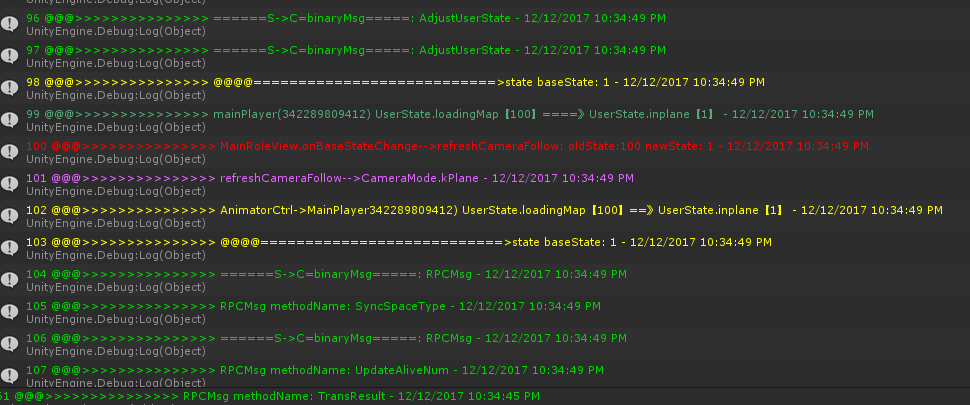
### 加载进度条断线出错

正常流程

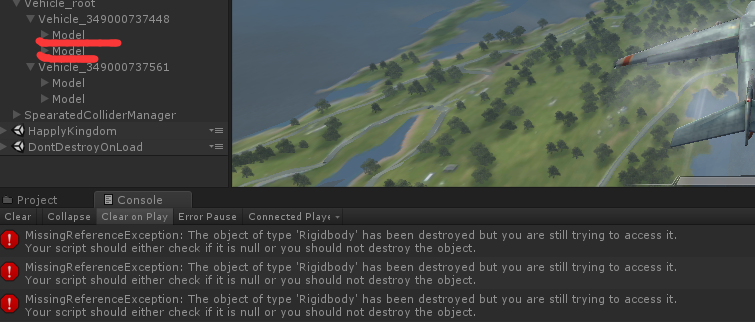




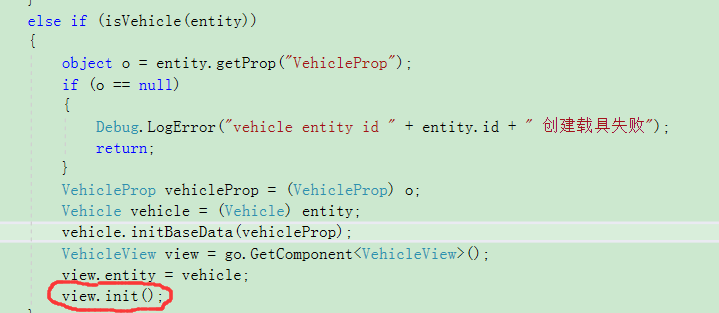




### 小断线重连上来一大堆找不到rigidbody的报错

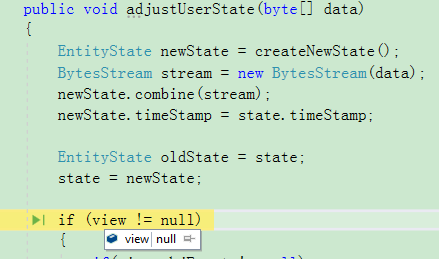


原因是onEnterAOI-createRenderObj后载具还会再load模型：

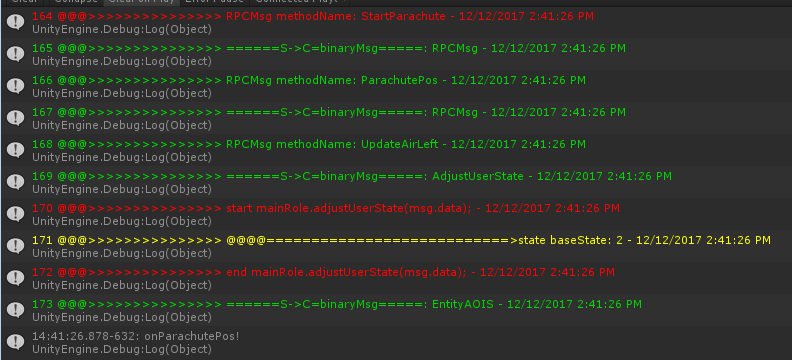


### 小断线恢复到飞机上点跳伞没反应

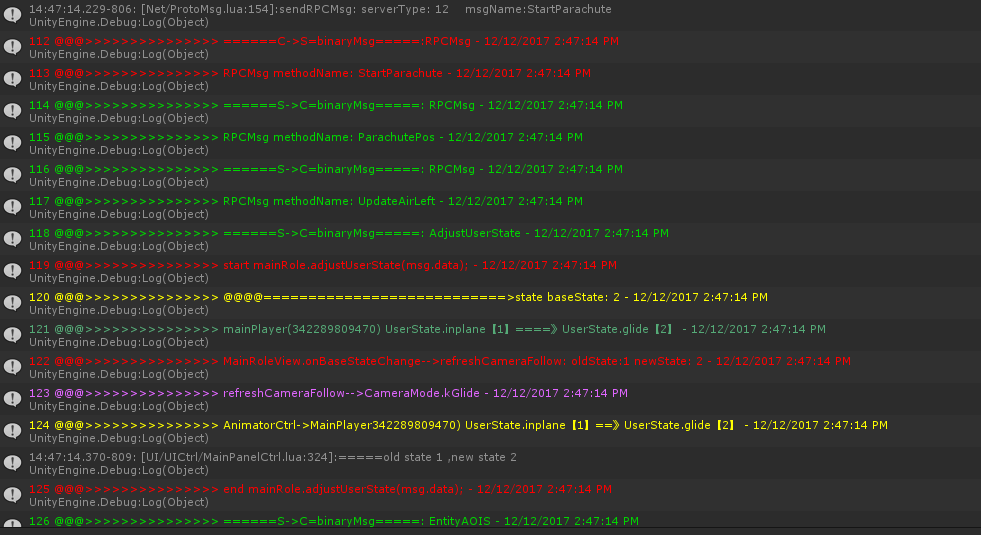
View空了。。 （最后发现竟然有2个view, entity里一个view, mainPlayer里一个view…）



错误：



没问题:



### onEnterAOI fire总是出错

类型转换出了问题，不是MainPlayer也作为MainPlayer抛了

### 小断线上来后相机错乱

MainPlayer的View都为空了，，，AdjustUserState根本没用

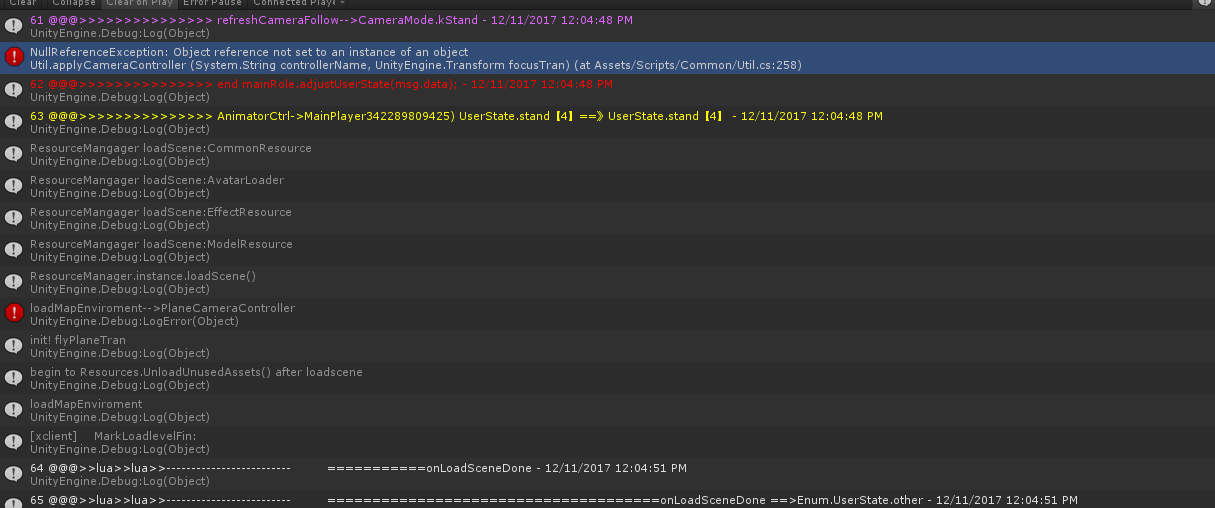
### 没有AllLiveNum的信息



### 小断线重连上来进不了了

loadMapEnviroment

### 大断线进去后相机不对

Wrong!

=======================onMatchSuccess

start===========load Scene============

RPCMsg methodName: AirLine

RPCMsg methodName: UpdateKillNum

UpdateTotalNum

UpdateAirLeft

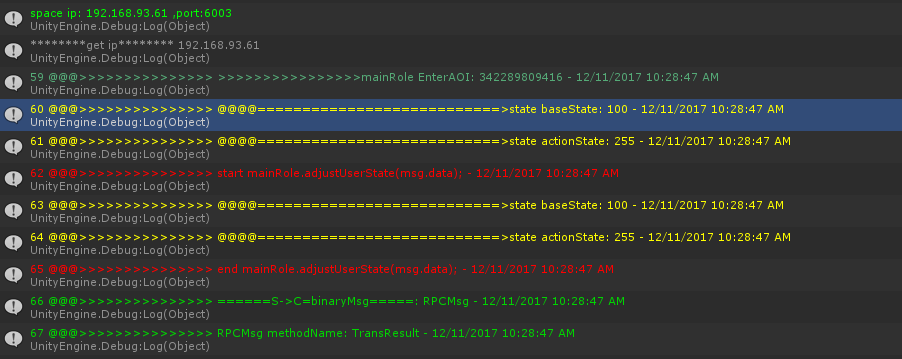
EnterSpace

EntityAOIS

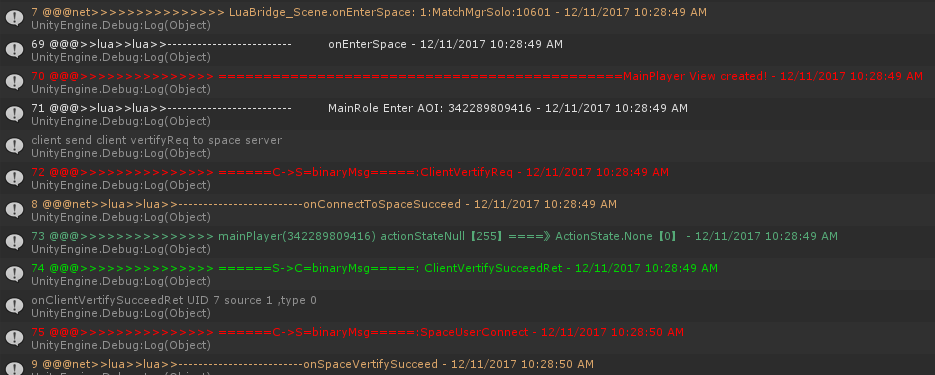
PropSyncClient

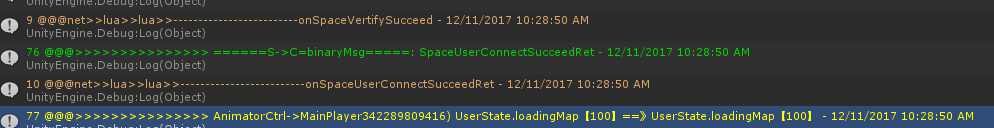
AdjustUserState

EntityAOIS









end===========onLoadSceneDown============

**关键变量：**

GameScene.instance.**isEnterSpace**

### 断线后重连上网关，但游戏结束了，但没收到通知，一直在游戏里

没有收到Space断开连接的信息

### 太多日志

### 后面断线重连修改可热更（该导的全部给lua）

验证结果：

网关 Events.fire(ZeusEvent.LoginSucceed);

SPACE Events.fire("SpaceSessVertified");

### IOS断网很长时间不触发重连

### IOS版出现重连上了马上断开的情况（陷入一个死循环？）

### IOS没网络的环境下连接是没有回调的

其实是有回调的，1分15秒才有回调。。。

### 小地图下边不应显示“搜索目标区域倒计时”

ZoneRefreshData

重连上来服务器还会发

RPC\_ShrinkRefresh", 300);的消息

### 重连上来出现2个毒圈？

Enum.ZoneNotifyType = {

cur = 0, --当前蓝圈

next = 2, -- 白圈

beforeRun = 3 --跑毒之前的蓝拳

}

self.beforeRunPos （跑毒之前的篮圈位置）

self.beforeRunRadius （跑毒之前的篮圈半径）

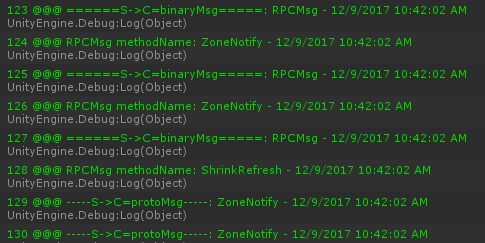
self.nextSafeAreaPos (白圈位置)

self.nextSafeAreaRadius (白圈半径)

self.safeAreaPos （消息的中心坐标赋给它）

self.safeAreaRadius （消息的半径赋给它）

self:refreshPosionFog(ret.interval)



### 重连后还看到别人在地方带着降落伞

服务器调整时间（高度）解决

### 重连后还看到别人停在空中

别人调一下就正常了

服务器刷新最新的状态就好了

### 重连上来后打开降落伞停在那很长时间不动

重连后服务器给的高度有问题

### 重连上来后已经打开伞了，还有”打开降落伞”的按钮

if self.parachuteBtnVisible then

self:onParachuteBtn(false, newState)

end

### 重连上来后相机位置不对



原因是相机控制器的目标被删了

### 重连上来一堆瞄准镜

function M:onAllowParachute()

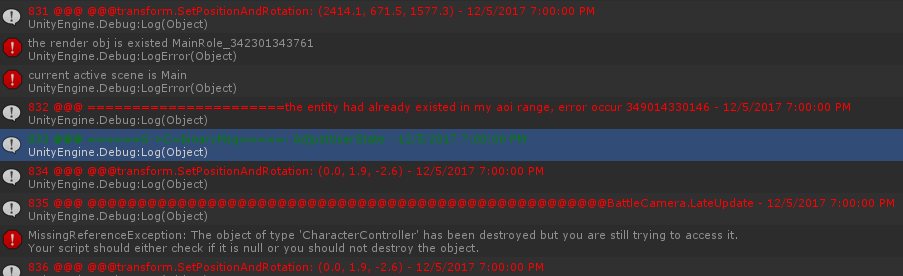
if not self.parachuteDone then

self:**showGameUI**() 造成

end

### onEnterAOI 没刷新状态？？

## 断线重连



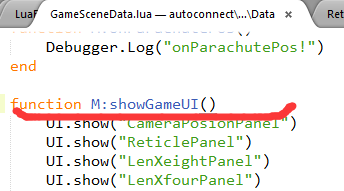
Cratos.Zeus.inst:reconnectSpace()

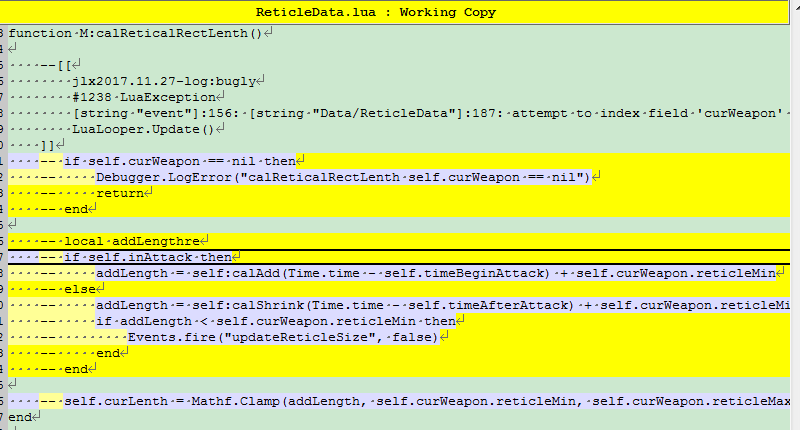
**GameScene.instance.isEnterSpace**

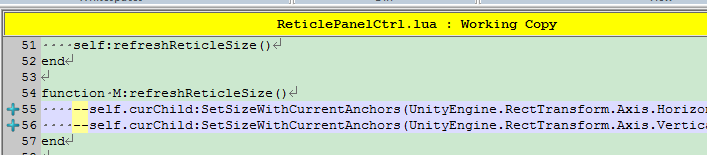
**Cratos.Zeus.inst:isSpaceValid()**

svn://192.168.150.63/timefire/branches/branch\_autoreconnect

### 重连上会出现一堆的瞄准镜界面

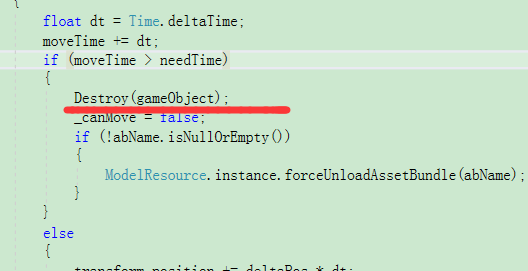






### 飞机飞结束了还没重连上

飞机被销毁了 GamePlane.cs



### 收到token的时机

https请求的时候就返回了， 2个连接共用同一个token (一次https请求)

### 状态切换的时机

enum SessState

{

offline = 0,

online,

login,

netErr

}

TcpSocket状态

public enum State {

kDisconnected = 0,

kConnecting,

kConnected

}

GameStates:getCurState()

Enum.GameState = {

None = 0,

Init = 1,

Login = 2,

Match = 3,

Game = 4,

}

### 目前的控制变量

GameScene.instance.isEnterSpace

Gateway.isSpaceSessVertified

(false closeSpaceSess()时调用)

(true 已连接(tcpsocket里的状态)并且验证通过时设置)

Game.instance.isNetConnectValid()

GameSceneData.lua

self.gatewayReconnNum = 0

self.spaceReconnNum = 0

self.reconnectMaxNum = 30

self.needReconnSpace

### 两个session

只能gateway的时候，竟然2个连接同时在刷新(update)

还没连接的时候就在刷新了。。。

进入room后两个连接都要用。。。

### 一场游戏结束链接和状态的切换

### SDK和编辑器的不同

### 客户端本地断线和与服务器失去连接

### 服务器停机

## 网络

IOSIPv6

### 消息发送

**Buff链表:**



**消息结构示意图:**

**Sess.send(object msg) send(int msgID, byte[] buf)**

{

//二进制(结构体)、protobuf序列化

MsgCoder.inst.marshal(msg, msgBuf, out msgID, out msgLen))

}

🡺

**Sess.sendMsgBuf(int msgID , byte[] buf , int len)**

**{**

**//压缩加密包体**

connHelper.fillSendBuf

**TcpSocket.send**

**}**

🡺🡺

**TcpSocket.send(byte[] data, int offset, int length)**

{

找到buff链表的最后节点

if(最后节点有足够的空间)

拷贝字节流到链表最后节点

else

创建新的节点用来拷贝这些字节流(根据字节流的大小可能创建好几个节点)

从链表取出第一个buffer

调用TcpSocket.\_send(Buffer buffer)

}

🡺🡺

TcpSocket.\_send(Buffer buffer)

{

Try

{

异步发送一个buff (如果在回调检查中没问题，继续发送下一个buff)

}

Catch (Exception)

{

清空队列

释放内存

}

}

Lua:

function SEND\_PROTO\_MSG(msg)

Cratos.Zeus.inst:post(msgName, buffer)

Gateway.inst.send(msgInfo.msgID, msgBuf);

function SEND\_RPC\_MSG( ...)

Cratos.Zeus.inst:rpc(serverType, msgName, stream:GetUsedBytes());

Cratos.Zeus.inst:post

Gateway.inst.send(msgInfo.msgID, msgBuf);

### 消息接收

//每帧循环

TcpSock.update

🡺🡺

TcpSock.\_runRecvLoop

🡺🡺 IEnumerator sess.recvLoop (咋调用过来的)

2级🡺🡺

Sess.recvMsgBuf

3级🡺🡺

connHelper.readIO

{

**//rawIO**

TcpSock.recv(byte[] buffer, int offset, int length)

{

**从m\_receivedBuffers 获取字节流**

}

}

2级🡺🡺

MsgDeliver.inst.onMsgRecv(recvMsgID, recvBuf, recvLen);

{

**//反序列化（结构体消息、proto消息）**

MsgCoder.inst.unMarshal(msgID, msgBuf , msgLen , out msgName, out msg))

Events.fire(msgName, msg);

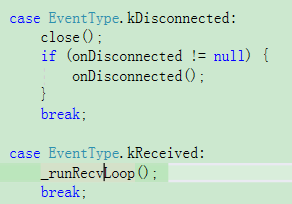
**//proto消息发到Lua层面一份（这只是放在队列里,真正传递见注2）**

rawMsg.put(new MsgContent(msgID, msgInfo.msgName, msgBuf));

}

**注1：**

**有个独立的异步接收线程不断的接收数据放到 m\_receivedBuffers中，同时更新事件列表(**private LinkedList<EventType> m\_events**)操作数据上锁, 主线程根据事件链表进行操作：**



**注2：**

循环里不停取消息传给lua函数



**消息传给lua还有这三大接口:**



**=====================================**

### 套接字C#函数

m\_socket.**BeginReceive**(buf.data, 0, buf.data.Length, 0, \_onReceive, buf);

## 调试

Debug.DrawLine(startPoint, startPoint + shootRota \* Vector3.forward \* weapon.distance, Color.green, 10, true);

## 发布、更新

**宏：**

tx\_pb

PatchUrl.lua



PatchCfg.lua



//热更新完成之后调用

//销毁luaVM后重建

//清理c#缓存的lua数据结构

//重新加载lua的ab包资源

//重新进行lua层面的初始化

版本号：

version.cs

## 第三方库

### Unity插件

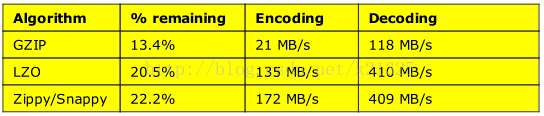
ShaderForge 、Haptic震动插件、EasyTouch 、BabeLua

### Ipv6

### LitJson

### Protobuf-net

### Snappy(压缩算法)



## 优化

### 优化措施

**峰值内存回收**

**资源打包：**

1 表格打成一个ab包

2 配置打成一个ab包

3 shader打成一个ab包

4 UI 字体合图单独打ab包，每个UI一个ab包

5 lua每个子文件夹一个ab包

6 地图 建筑、石头、树、水各一个ab包，天空盒(每个子文件夹一个ab包，一个子文件夹对应一个天空盒),物件（每个子文件夹一个ab包,一个子文件夹对应一种物件）、（草单独对应一个ab包）

7 模型 载具(一个载具对应一个ab包) 、道具(一个道具对应一个ab包)、武器(一种武器对应一个ab包)

8 Avatar(头盔、护甲、背包、身体、头发 每种都对应一个ab包(n个头盔对应n个ab包))

所有角色动画对应一个ab包

### Unity

.transform

Carmer.main

会很耗

### CPU

超过33ms的帧数占比 (?) (4.3%)

**a, 渲染模块**

**CPU消耗：**

红米Note2:

半透明(?) (1.5ms)

不透明(?) (1.8ms)

提升方法：

查看**Camera.Render**下各函数的消耗占比，对场景模型、蒙皮网格和UI进行控制，

要是Shader.SetPass消耗过高，建议对场景中的Material数量进行控制

**DrawCall:**

(?) (48--179)

**b, UI模块**

**CPU**

(?) (1.9ms)

**每帧堆内存**

(?) (2KB)

**函数：**

Canvas.SendWillRenderCanvases

**c, 动画模块**

**函数：**

Animator.Update (?) (1.8ms).

MeshSkinning.Update (?) (1.8ms)

**d, GC调用**

?帧/次 (1921帧/次)

1w帧 堆内存<30M

**检测方法：**

查看函数的堆内存

### 内存

(总内存峰值？ ) (249M)

(mono堆内存峰值 ?) (32.5M)

**a, Mono堆内存**

(?)(2.5W帧 < 40 M)

**a, 资源内存**

**纹理资源**

纹理资源数量峰值 (?) (618个)

纹理内存占用峰值 (?) (119M)

ETC1格式纹理个数 (?)

Alpha8格式纹理(?)

RGBA32(ARGB32)格式纹理：(?)

RGB24 格式纹理 (?)

RGBA16格式纹理 (?)

**Mesh资源**

资源数量 (?) (239个)

内存峰值 (?) (14M)

**AnimationClip资源**

资源数量 (?) (476个)

内存峰值 (?) (9.6M)

**AudoClip资源**

资源数量 (?) (101个)

内存峰值 (?) (7M)

### 耗电量

DrawCall

IO

流量

### 示例

A, 对频繁切换shade进行预加载

B, 减少GameObject的Active和Deactive次数 （会造成CPU浪费或间接造成CPU更大的浪费(比如Active出发Animator.Initialize)）

C, 场景合批

D，UI重新布局，降低DrawCall

E,2DUI转3DUI使支持动态合批

F,减少消息收发

# C#<-->lua接口

## 关联

LuaBridge

Luabind.lua

Events.fire("on\_RPCLuaEvent", "testplayermsg", newArgs);

## 已导出的

### 杂

\_GT(typeof(Screen)),

\_GT(typeof(HttpHelper)),

\_GT(typeof(Cratos.MsgInfo)),

\_GT(typeof(Cratos.MsgDef)),

\_GT(typeof(Cratos.Zeus)),

\_GT(typeof(LuaByteBuffer)),

\_GT(typeof(UserInput)),

\_GT(typeof(Game)),

\_GT(typeof(InputFilter)),

\_GT(typeof(DragDeltaChecker)),

LuaNumber

\_GT(typeof(WeaponCtrl.Weapon)),

\_GT(typeof(WeaponCtrl)),

\_GT(typeof(VehicleController)),

\_GT(typeof(AudioCtrl)),

\_GT(typeof(DragListener)),

\_GT(typeof(ClickListener)),

\_GT(typeof(StoppableDragDeltaChecker)),

\_GT(typeof(PressChecker)),

\_GT(typeof(ListElement)),

\_GT(typeof(ListView)),

\_GT(typeof(CycleListView)),

\_GT(typeof(AnimationCurve)),

\_GT(typeof(PressHoldButton)),

\_GT(typeof(ResourceLoader)),

\_GT(typeof(EffectResource)),

\_GT(typeof(UI.UIManager)),

\_GT(typeof(ModelResource)),

\_GT(typeof(AudioResource)),

\_GT(typeof(CommonResource)),

\_GT(typeof(UIResource)),

\_GT(typeof(ShaderResource)),

\_GT(typeof(SceneResource)),

\_GT(typeof(AvatarLoader)),

\_GT(typeof(ResourceManager)),

\_GT(typeof(GameScene)),

\_GT(typeof(RawImage)),

\_GT(typeof(GraphicQualityManager)),

\_GT(typeof(Rect)),

\_GT(typeof(MaskableGraphic)),

\_GT(typeof(Graphic)),

\_GT(typeof(RectTransform.Axis)),

\_GT(typeof(MovementJoystick)),

\_GT(typeof(Input)),

\_GT(typeof(UnityEngine.EventSystems.PointerEventData)),

\_GT(typeof(UIGallery)),

\_GT(typeof(UIGalleryItem)),

\_GT(typeof(UIRoleItemContent)),

\_GT(typeof(T\_Object)),

\_GT(typeof(EasyTouch)),

\_GT(typeof(Gesture)),

\_GT(typeof(Shader))

### C#、lua相互通讯

LuaBridge

### UI

\_GT(typeof(UIHelper)),

### 工具、扩展

\_GT(typeof(Debugger)).SetNameSpace(null),

\_GT(typeof(Extension)),

### 热更相关

//Custom Patch

\_GT(typeof(Patches.PatchChecker)),

\_GT(typeof(Patches.Patcher)),

\_GT(typeof(Patches.DownLoader)),

\_GT(typeof(Patches.UserDefault)),

\_GT(typeof(Patches.FileUtil)),

\_GT(typeof(Patches.MD5Code)),

### SDK相关

\_GT(typeof(TXSdkManager)),

\_GT(typeof(Msdk.LoginRet)),

\_GT(typeof(Msdk.WakeupRet)),

\_GT(typeof(Msdk.RelationRet)),

\_GT(typeof(Msdk.ShareRet)),

\_GT(typeof(Msdk.NoticeInfo)),

\_GT(typeof(Msdk.NoticeInfoList)),

\_GT(typeof(Msdk.PersonInfo)),

\_GT(typeof(Msdk.ePlatform)),

\_GT(typeof(Msdk.eQQScene)),

\_GT(typeof(Msdk.eStatusType)),

\_GT(typeof(Msdk.eWechatScene)),

\_GT(typeof(Msdk.TypeInfoImage)),

\_GT(typeof(Msdk.TypeInfoVideo)),

\_GT(typeof(Msdk.TypeInfoLink)),

\_GT(typeof(Msdk.TypeInfoText)),

\_GT(typeof(Msdk.ButtonApp)),

\_GT(typeof(Msdk.ButtonWebview)),

\_GT(typeof(Msdk.ButtonRankView)),

\_GT(typeof(Msdk.eMSDK\_SCREENDIR)),

\_GT(typeof(Msdk.eADType)),

\_GT(typeof(Msdk.eApiName)),

\_GT(typeof(Msdk.LocalMessageAndroid)),

\_GT(typeof(Msdk.LocalMessageIOS)),

\_GT(typeof(Msdk.GroupRet)),

\_GT(typeof(Msdk.ADRet)),

\_GT(typeof(Msdk.CardRet)),

\_GT(typeof(Msdk.WebviewRet)),

\_GT(typeof(Msdk.RealNameAuthRet)),

\_GT(typeof(ShareAssist)),

\_GT(typeof(TXSDKHelp)),

\_GT(typeof(RadarGraph)),

### 未导出的

### 还要导出的

Events（注册了没找到？）？ Frames？

Entity？视图相关的

AudioManager

DynamicGameObjectManager

读表?

BulletsMgr?

加入新的动作状态如何热更？

DoorManager

语音(? 不用。。)

**目标是把角色状态动作都导给lua**

# Todo

加密lua

C#全部接口导出给lua

AB资源规划

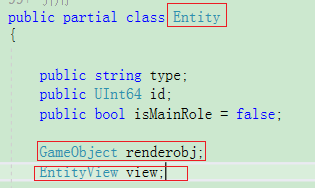
游戏优化

# 特点

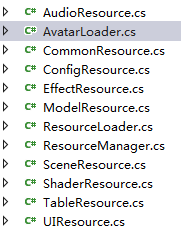
## 自动和服务器同步属性 （Entity\_Prop(文件)）

## RPC

## 逻辑视图分开



## 各种ResourceLoad



## Lua代码与UI预制名字一一对应，非常好找

## 按游戏状态划分UI

# 优化

同步！

## CPU

**渲染模块**

减DrawCall:

减少网格(减面)

合并网格

地形的pixel error参数不要太小

简化shade

减少特效复杂度

不同性能的机型使用不同档次的资源(会增加包大小)

多线程渲染

**UI模块**

Canvas.BuildBatch

Canvas.SendWillRenderCanvases

动静分离

少用Instantiate/Destory Active/Deactive

**加载模块**

**资源加载**

预加载

异步加载

分帧加载

**GC调用**

优化代码堆内存

手动调用GC其实没啥必要

//Resource.UnloadUnusedAssets、GC.Collect() 在真正需要调用的时候才用

**缓存**

## 代码优化



## 资源管理

**AssetBundle加载/卸载**

**加载方式：**



用LoadFromFile、LoadFromFileAsync

**具体AssetBundle使用情况**

**资源加载/卸载**

**GameObject实例化与激活**

## 内存

减面

不同的机型用不同分辨率的资源

卡的时候用通用的模型

### 资源打包

shader打成一个ab包、缓存



## 耗电量（手机发热）

GPU、CPU、网络、IO

**减DC(批次)**

分类绘制、自动合批

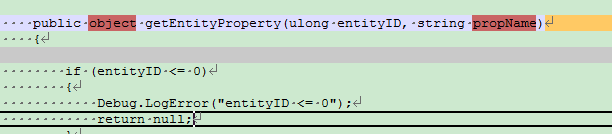
**减IO**

## 优化措施

### 峰值内存回收

### lua

1 避免C#、lua之间频繁调用



特别是别在循环里相互调用

2 string参数和object返回值都会触发GC

[**用好Lua+Unity，让性能飞起来——Lua与C#交互篇**](http://blog.csdn.net/uwa4d/article/details/54344356)

[**http://blog.csdn.net/uwa4d/article/details/54344356**](http://blog.csdn.net/uwa4d/article/details/54344356) **(good)**

原理分析：

http://blog.csdn.net/xingaaaxing/article/details/72869709

注册耗时，内存、查表，拆装箱

push 一个数字也不装箱

精确类型就没有

不要啥都用object传递和接收，那种出现装箱也只能怪自己

### 资源打包：

# [关于Unity中的资源管理，你可能遇到这些问题](https://blog.uwa4d.com/archives/QA_ResourceManagement.html)

https://blog.uwa4d.com/archives/QA\_ResourceManagement.html

1 表格打成一个ab包

2 配置打成一个ab包

3 shader打成一个ab包

4 UI 字体合图单独打ab包，每个UI一个ab包

5 lua每个子文件夹一个ab包

6 地图 建筑、石头、树、水各一个ab包，天空盒(每个子文件夹一个ab包，一个子文件夹对应一个天空盒),物件（每个子文件夹一个ab包,一个子文件夹对应一种物件）、（草单独对应一个ab包）

7 模型 载具(一个载具对应一个ab包) 、道具(一个道具对应一个ab包)、武器(一种武器对应一个ab包)

8 Avatar(头盔、护甲、背包、身体、头发 每种都对应一个ab包(n个头盔对应n个ab包))

所有角色动画对应一个ab包

头 3K drawcall :2 skinnedmeshs:1

身体 10K个面 drawcall :1 skinnedmeshs:1

枪 2K drawcall :2

背包 0.5K—1.0K drawcall :1 skinnedmeshs:1

头盔 0.25K-3.0K drawcall :1

坦克高模 20K 低模 4K



# 待改进？

C#调用lua太频繁，易产生GC

MonoBehaviour用得太多。。。