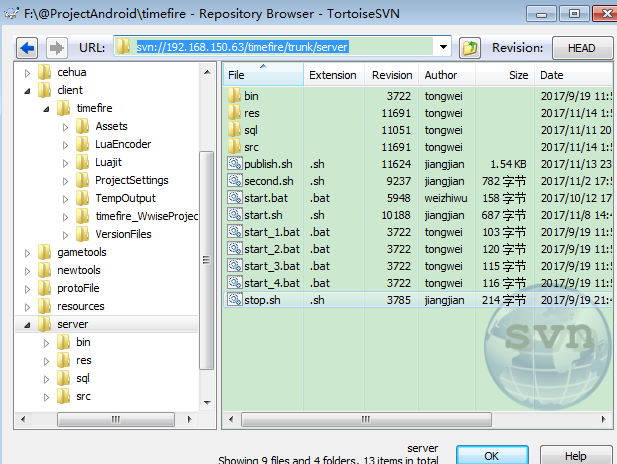
# Q&A

svn://192.168.150.63/timefire/trunk/server



## Terrain

## .asset

## 四元数

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

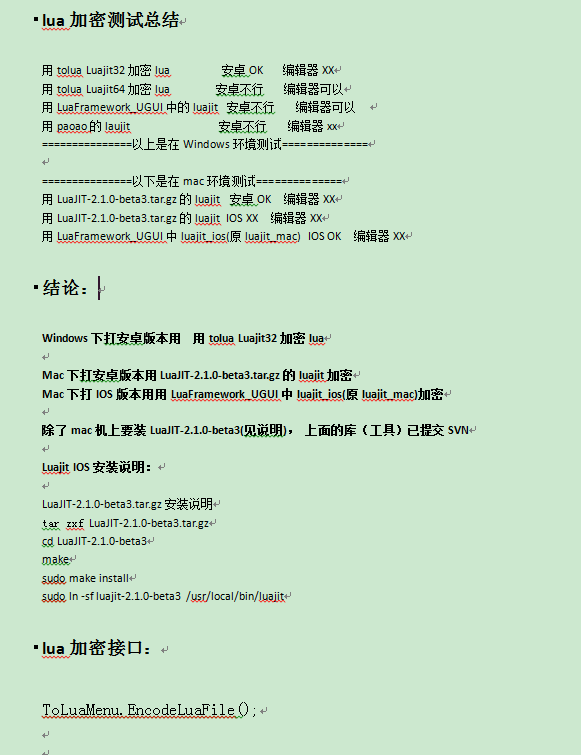
float yaw = shootRota.eulerAngles.y \* Mathf.Deg2Rad;

## Particle

particle.Stop();

particle.Play();

## Lua加密



## 安卓打包

安卓做测试包流程：

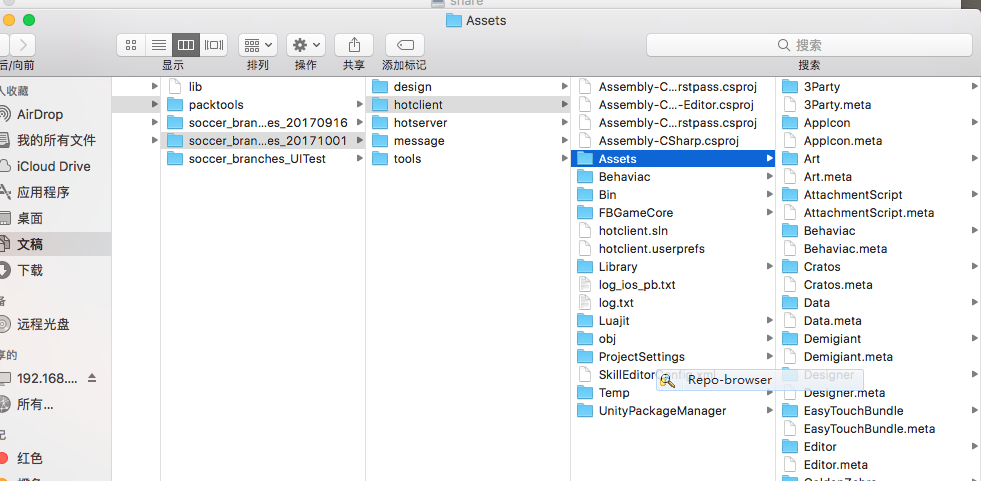
1、菜单Tools/一键打包安卓资源

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

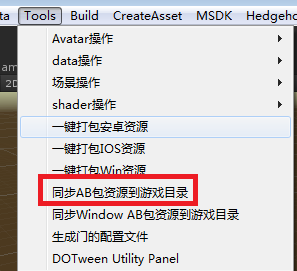
## 截屏

Ctrl+Shift+A

## lua

LuaProxy





## .asset

AssetDatabase.CreateAsset

## CG

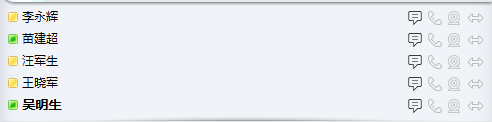
Handheld.PlayFullScreenMovie

## 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:

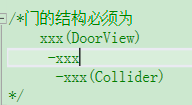
fetch只拉取不合并

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

### Pull:

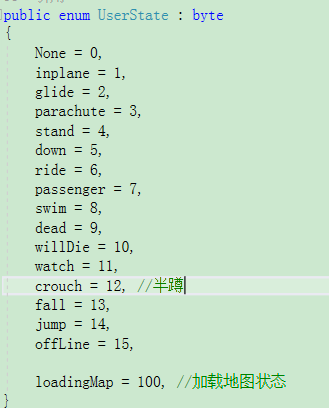
# Client

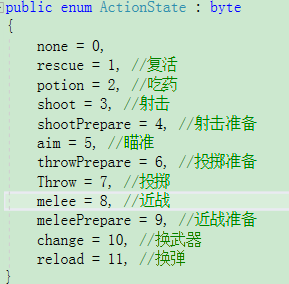
## 美术规范



## 功能

### 状态





## 登陆

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

## 初始化

### lua

luaclient

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

## 资源打包、管理、释放

Resources.UnloadUnusedAssets();

## 热更

\_GT(typeof(Patches.PatchChecker)),

\_GT(typeof(Patches.Patcher)),

\_GT(typeof(Patches.DownLoader)),

\_GT(typeof(Patches.UserDefault)),

\_GT(typeof(Patches.FileUtil)),

\_GT(typeof(Patches.MD5Code)),

## 动画

## 网络

IOSIPv6

## 调试

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

## 第三方库

### Unity插件

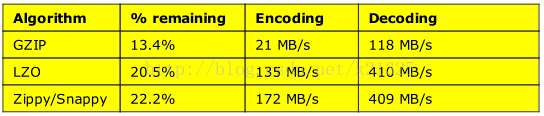
ShaderForge 、Haptic震动插件、EasyTouch

### Ipv6

### LitJson

### Protobuf-net

### Snappy(压缩算法)



## 优化

### 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

## 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

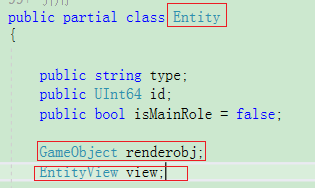
游戏优化

# 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

