# 2015

## 2015/11/27

### protoc-gen-lua

pb.h

pb.c

https://code.google.com/p/protoc-gen-lua/

### 读取图片（读取文件）

Data data = FileUtils::getInstance()->getDataFromFile(\_filePath);

Data 包含数据指针和大小以及一些常用操作

压缩包工具

ZipUtils

CCImage.cpp Format::ETC

**是否支持ETC:**

Configuration::getInstance()->supportsETC()

### 绑定纹理

if (info.compressed)

{

glCompressedTexImage2D(GL\_TEXTURE\_2D, i, info.internalFormat, (GLsizei)width, (GLsizei)height, 0, datalen, data);

}

else

{

glTexImage2D(GL\_TEXTURE\_2D, i, info.internalFormat, (GLsizei)width, (GLsizei)height, 0, info.format, info.type, data);

}

## 2015/11/26

### 脚本说明

pngTotga.bat png🡪tga

TexturePacker.bat 生成 test.plist test-a.jpg testalpha.jpg (png🡪tga🡪jpg+alpha)

TexturePackerAll.bat 可以生成jpg pvr

Tga是texturepacker转的

Pkm+alpha 、Jpa+alpha是jpgcompress转的

### Cocos 创建工程

Cocos new myTest  –p com.game.hellogame -l cpp -d ~/Desktop

### 安卓日志查看

adb logcat | findstr cocos

### 挂机几个打日志的函数

Fir::log::instance().logMsg(“dd”) //ios日志会写文件 安卓会输出控制台

GameTime::instance().log(“”) //会输出到GM面板上，同时会调用上面的函数

## 2015/11/23

### [svn仓库转为git仓库](http://nowing.iteye.com/blog/844608)

<http://nowing.iteye.com/blog/844608>

http://www.aikaiyuan.com/6584.html

### cocos ide 输出不了lua日志

默认是按release版本来编译的，把print换成release\_print可以输出

更好的方法参考：

http://blog.k-res.net/archives/1833.html

### cocos studio创建的lua工程没有framework/runtime-src

在创建的界面里设置 framework

### cocos ide创建的lua工程没有framework/runtime-src

方式1： 创建工程的时候勾选 add native support

方式2： 已经创建的工程 右键工程🡪cocos tools🡪add native codes support

http://segmentfault.com/a/1190000000628902

### vim查找替换

全局**替换**命令为：:%s/源字符串/目的字符串/g

### mysql

SERVERLIST

1.导出整个数据库  
　　mysqldump -u用户名 -p密码  数据库名 > 导出的文件名  
　　C:\Users\jack> mysqldump -uroot -pmysql sva\_rec  > e:\sva\_rec.sql  
   
　　2.导出一个表，包括表结构和数据  
　　mysqldump -u用户名 -p 密码  数据库名 表名> 导出的文件名  
　　C:\Users\jack> mysqldump -uroot -pmysql sva\_rec date\_rec\_drv> e:\date\_rec\_drv.sql  
   
　　3.导出一个数据库结构  
　　C:\Users\jack> mysqldump -uroot -pmysql -d sva\_rec > e:\sva\_rec.sql  
     
     4.导出一个表，只有表结构  
　　mysqldump -u用户名 -p 密码 -d数据库名  表名> 导出的文件名  
　　C:\Users\jack> mysqldump -uroot -pmysql -d sva\_rec date\_rec\_drv> e:\date\_rec\_drv.sql

**导入数据库：**

mysql>source d:wcnc\_db.sql

导出数据库：

**添加用户并设置权限：**

GRANT ALL ON \*.\* TO `username`@`%` IDENTIFIED BY ‘passwd’

**导出用户数据：**

Select \* from CHARBASE where charid=12345678 into outfile ‘/tmp/user12345678.sql’;

**导入用户数据：**

Load data infile ‘/tmp/user12345678.sql’ INTO TABLE CHARBASE;

导入后修改新增记录ACCID字段成现有的ACCID

## 2015/11/15

### 非越狱ios无需审核安装ipa

http://my.oschina.net/qixiaobo025/blog/321050

## 2015/11/13

### 接管cocos读文件接口

Fileutils::getInstance()->setDelegate(new class) class从fileutilapple🡨fileutil继承

## 2015/11/04

### 内存管理、异步加载

### 手动添加动画、文件添加动画

http://www.cocoachina.com/cocos/20140514/8414.html

## 2015/11/03

### Etc图片格式如何用alpha通道

[ETC 纹理压缩和 Alpha 通道处理](http://blog.csdn.net/myarrow/article/details/21018695)

<http://blog.csdn.net/myarrow/article/details/21018695>

http://blog.ch-wind.com/cocos2d-x中shader的使用/

### Cocos像素相加（叠加）

Sprite->setBlendFunc(BlendFunc::ADDITIVE)

CocosStudio里Sprite也有混合方式

### [用protoc-gen-lua生成PB的lua代码](http://blog.csdn.net/mergerly/article/details/16717311)

http://blog.csdn.net/mergerly/article/details/16717311

## 2015/10/26

### C++和java互调

**C++调用Java:**

JniMethodInfo

JniHelper::getStaticMethodInfo

**java调用C++:**

在java层声明函数、在C++层定义函数 函数体放在extern “C”里

### [NSString 与C++ string字符串的互转](http://blog.csdn.net/typename/article/details/6730183)

<http://blog.csdn.net/typename/article/details/6730183>

std::string str("hello");

   NSString \*str=[NSString stringWithUTF8String:str.c\_str()];

   NSString \*istr=[NSString stringWithUFT8String:@"zsh"];

## 2015/10/25

### 场景异步加载？

IO异步 创建纹理渲染同步

UI全程同步

### 适配（不一定适用）

**地图：** 屏幕尺寸越大，看的地图越多

UI：640X960 + 大底图 768X1136

### 梵天修真分辨率

标准iphone4s 960X640

底图：1280X720

### ETC1

不支持alpha通道,

**解决方法;**

有通道拆开，分开压缩 rgb和alpha成2个 etc1, 用多重纹理来做，需要shade支持

### 子弹追踪 角度更新

void ControlLayer::EnemyMove(float dt){

    CCObject \*et;

    CCARRAY\_FOREACH(enemyArray,et)//遍历所有子弹

    {

        EnemySnake \*enemySnake= (EnemySnake\*)et;

        //enemySnake->setPositionX(enemySnake->getPositionX()+1);

        //计算角度

        CCPoint ePoint=enemySnake->getPosition();

        CCPoint hPoint=snakeGameLayer->SnakeBody->getPosition();

        //计算矢量

        CCPoint pVectr=ccpSub(ePoint,hPoint);

        //反tan得到弧度

        float angleRadians=atan(pVectr.y/pVectr.x);

        //得到角度 弧度除以pi乘以180

        float angleDegrees = CC\_RADIANS\_TO\_DEGREES(angleRadians);

        if(pVectr.x>0){

            angleDegrees=angleDegrees+180;

        }

        enemySnake->setRotation(-angleDegrees);

        enemySnake->setPosition(

            ccpAdd(enemySnake->getPosition(),

            ccp(cos(angleDegrees)\*iEnemySpeed,sin(angleDegrees)\*iEnemySpeed)

            )

            );

    }

}

### Cocos释放显存

removeunusetexture

removetexturebyname? 释放指定图片

显存察看工具?

相关代码:

CCTextureCache

### Cocos游戏图像深度和图片格式

**UI:** 32位png

**模型:** rgba444

**地图:** rgb565 (不需要alpha通道)

图片格式:

IOS全部用pvr格式

Android 用etc1+alpha

UI未处理

### AngryBirds基于物理的更新位置和方向

world->Step(dt, velocityIterations, positionIteratoins);

//遍历物理世界中的物体

for(b2Body\* b = world->GetBodyList(); b; b = b->GetNext())

{

if(b->GetUserData() != NULL)

{

//获取精灵对象

CCSprite \*mySprite = (CCSprite\*)b->GetUserData();

//根据物理的位置设置精灵的位置

mySprite->setPosition(ccp((b->GetPosition().x) \* PTM\_RATIO, (b->GetPosition().y) \* PTM\_RATIO));

//根据物体的旋转角度设置精灵旋转角度

if(mySprite->getTag() >= 9)

{

mySprite->setRotation(-1 \* CC\_RADIANS\_TO\_DEGREES(b->GetAngle()));

}

}

}

## 11

### 111

# 2

## 21

### 111

## 22

### 111