# 2015

## 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()));

}

}

}