



一、项目分工

学号	名字	角色	班级	职责	贡献
15331151	李佳	组长	早上班	负责游戏思路 and 整体架构的编写，以及地图制作和动画制作等方面；	40%
15331150	李辉旭	组员	早上班	负责游戏主界面相应功能以及协助地图制作；	30%
15331145	李果	组员	早上班	负责游戏结束页面编写以及协助动画制作；	15%
15331167	李伟杜	组员	早上班	负责游戏提示页面相应功能的编写；	15%

二、开发环境

1. 系统环境：Windows10 系统
2. IDE: Visual Studio 2015 64bit
3. 游戏引擎：cocos 2d x
3. 解决方案配置及平台：proj.win32 Debug x86
4. 编程语言：C++

三、项目阐述

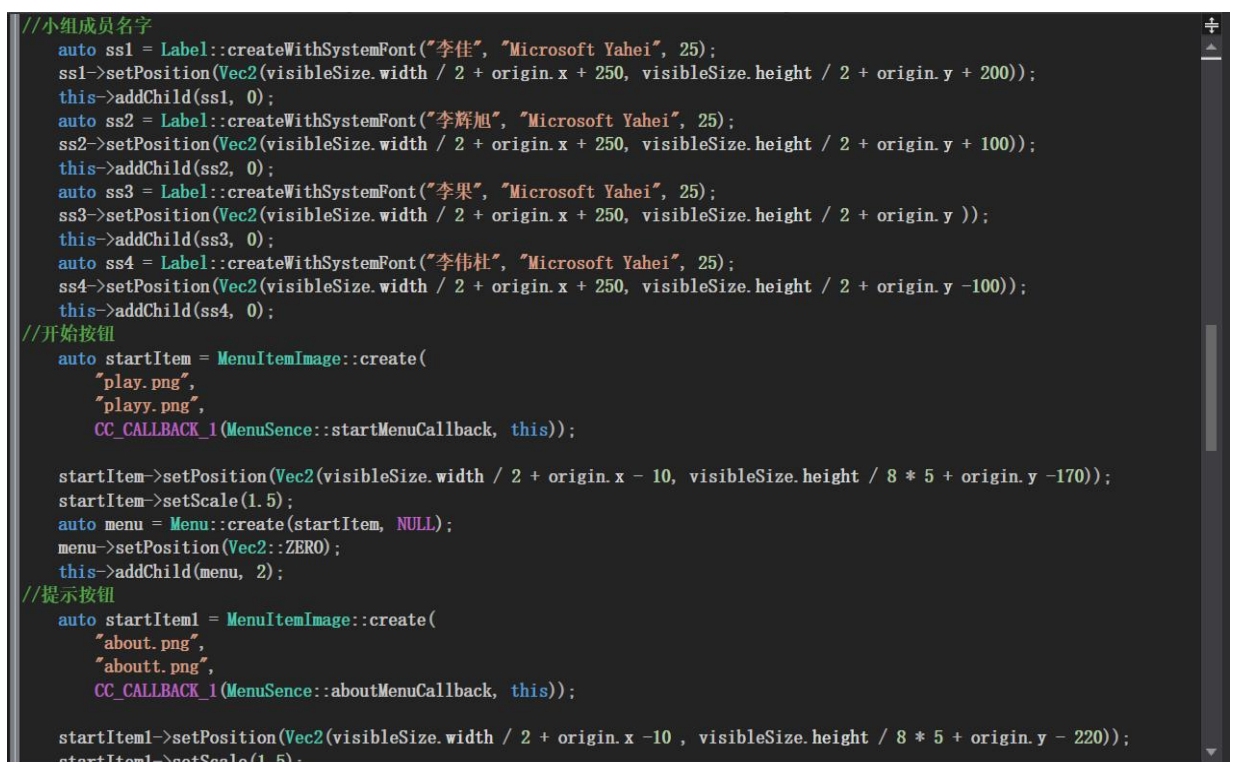
我们的期末项目叫“疯狂打砖块”，初步思路来自于流行的小游戏打砖块，在这个基础上，我们对以前游戏简单的界面进行了美化，增加了游戏提示界面以及背景音乐，优化了体验。对于游戏的地图，我们增加了 3 个特殊砖块，会增加 100 分，同时当分数大于 150 分时会变成两个球，提高了游戏难度。当游戏结束时，会跳转到游戏结束页面，显示游戏最高分记录以及本次游戏得分，我们还在页面上添加了 6 个垒起来跳舞的小人，和音乐节奏配合，显得非常生动。

这次游戏主要的亮点和创新就是增加了特殊砖块，以及当分数到达一定程度时会变成两个球，增加了游戏的难度和挑战性。游戏的最高分记录也是相比于之前这种小型游戏的改进。



四、项目展示

1. 游戏的主界面展示。右侧是小组成员名单，左侧是游戏名，中间是开始游戏按钮以及提示按钮，右下角是退出游戏按钮。

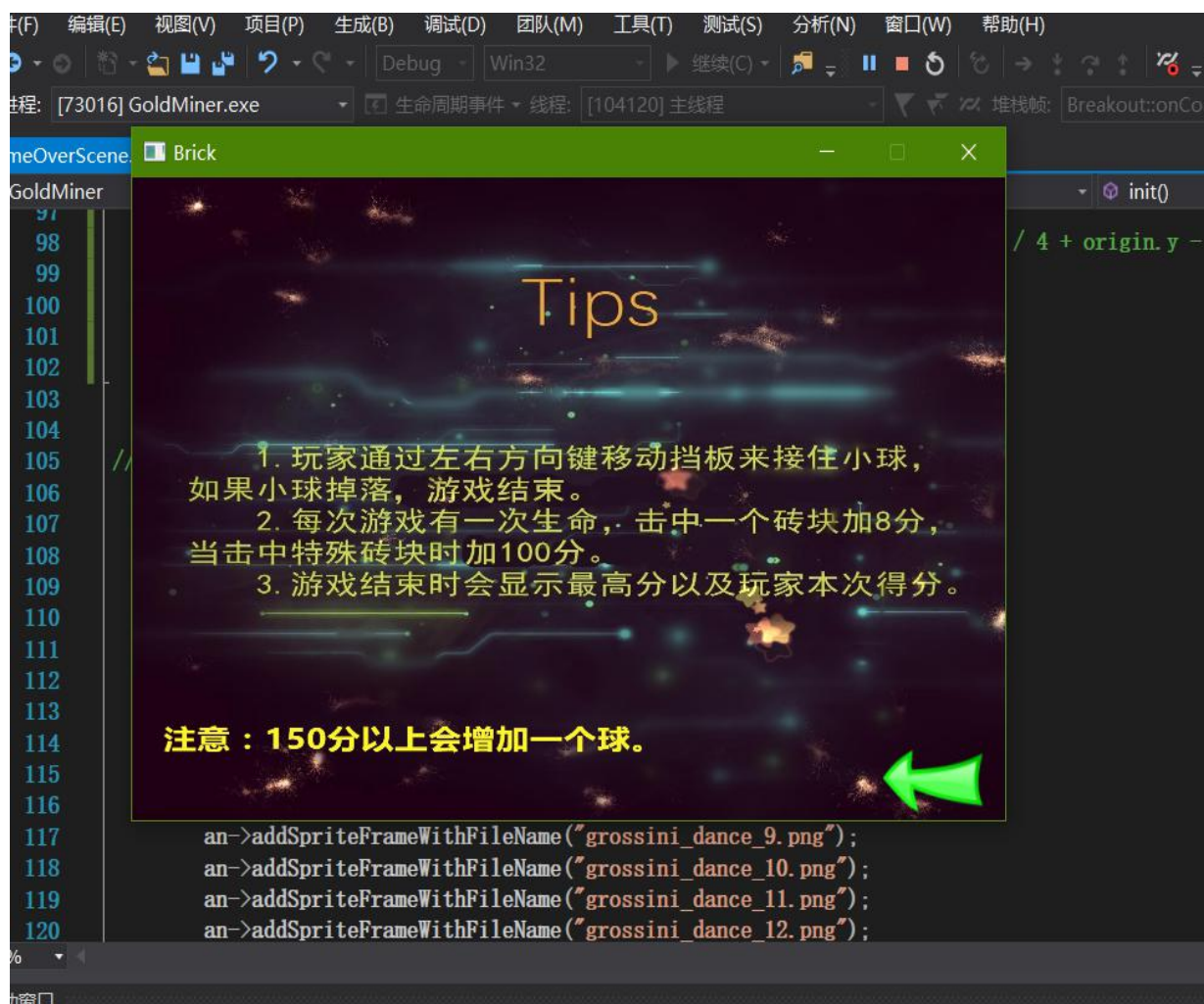




```
this->setPhysicsWorld(world);
Size visibleSize = Director::getInstance()->getVisibleSize();
Vec2 origin = Director::getInstance()->getVisibleOrigin();
//背景图片
auto bgmap = Sprite::create("1.jpg");
bgmap->setPosition(Vec2(visibleSize.width / 2 + origin.x, visibleSize.height / 2 + origin.y));
Size mywinsize = Director::getInstance()->getWinSize();
float winw = mywinsize.width; //获取屏幕宽度
float winh = mywinsize.height; //获取屏幕高度
float spx = bgmap->getTextureRect().getMaxX();
float spy = bgmap->getTextureRect().getMaxY();
bgmap->setScaleX(winw / spx); //设置精灵宽度缩放比例
bgmap->setScaleY(winh / spy);
this->addChild(bgmap, -1);
//关闭游戏按钮
auto closeItem = MenuItemImage::create(
    "CloseNormal.png",
    "CloseSelected.png",
    CC_CALLBACK_1(MenuSence::menuCloseCallback, this));

closeItem->setPosition(Vec2(origin.x + visibleSize.width - closeItem->getContentSize().width / 2 - 10,
    origin.y + closeItem->getContentSize().height / 2 + 10));
closeItem->setScale(1.5);
auto menuu = Menu::create(closeItem, NULL);
menuu->setPosition(Vec2::ZERO);
this->addChild(menuu, 1);
```

2. 提示界面展示。有游戏规则的简单介绍，以及注意的内容。点击右下角即可返回主页面。





```
if (!Layer::init())
{
    return false;
}

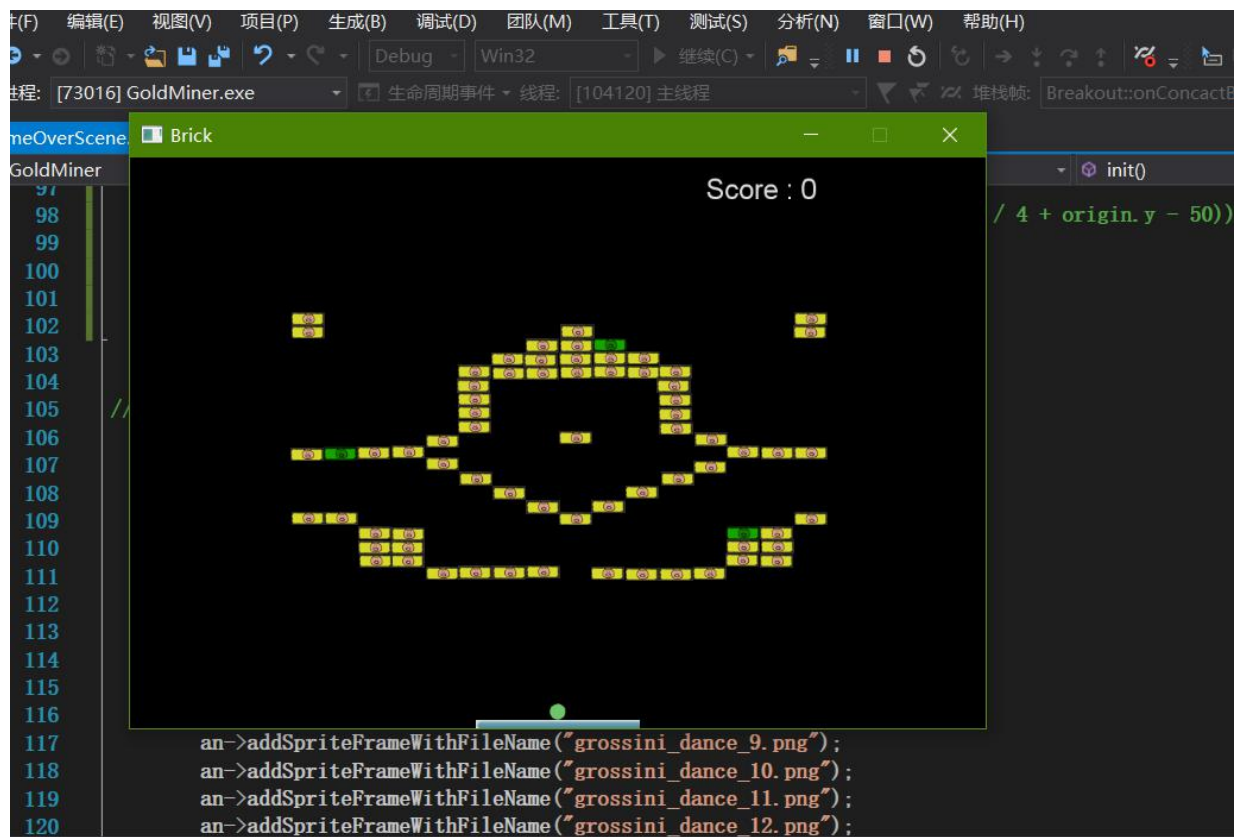
Size visibleSize = Director::getInstance()->getVisibleSize();
Vec2 origin = Director::getInstance()->getVisibleOrigin();

auto bgmap = Sprite::create("2.jpg");
bgmap->setPosition(Vec2(visibleSize.width / 2 + origin.x, visibleSize.height / 2 + origin.y));
Size mywinSize = Director::getInstance()->getWinSize();
float winw = mywinSize.width; //获取屏幕宽度
float winh = mywinSize.height; //获取屏幕高度
float scaleX = bgmap->getTextureRect().getMaxX();
float scaleY = bgmap->getTextureRect().getMaxY();
bgmap->setScaleX(winw / scaleX); //设置精灵宽度缩放比例
bgmap->setScaleY(winh / scaleY);
this->addChild(bgmap, -1);

//返回的按钮事件
auto startItem = MenuItemImage::create(
    "bga.png",
    "bgb.png",
    CC_CALLBACK_1(About::CancelMenuCallback, this));

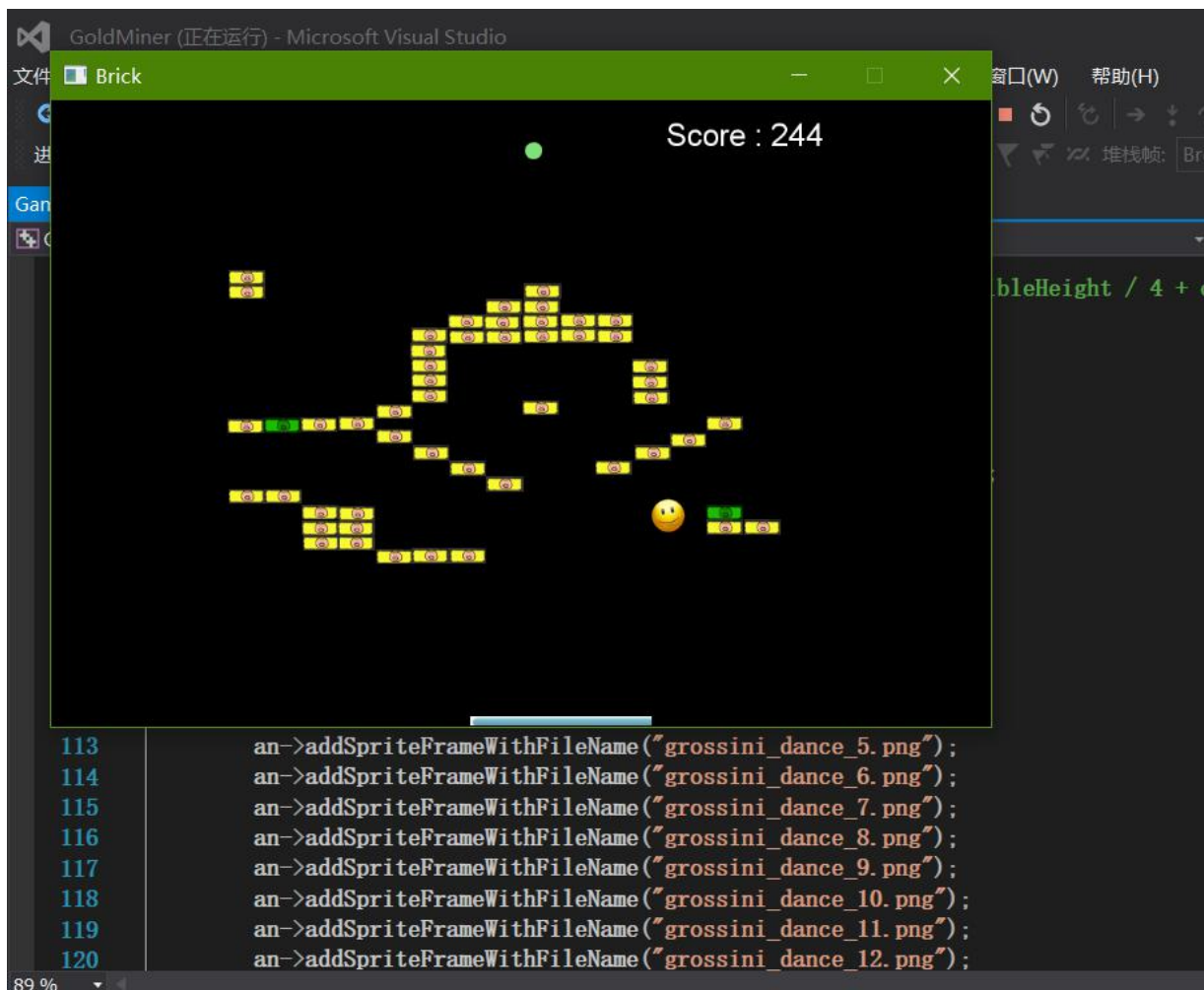
startItem->setPosition(Vec2(visibleSize.width + origin.x - 60, origin.y+30));
startItem->setScale(1.5);
auto menu = Menu::create(startItem, NULL);
menu->setPosition(Vec2::ZERO);
this->addChild(menu, 1);
```

3. 游戏界面展示。刚进入时只有一个球，地图如下图所示，绿色是特殊加分方块，黄色是普通方块。右上角是当前的分数。

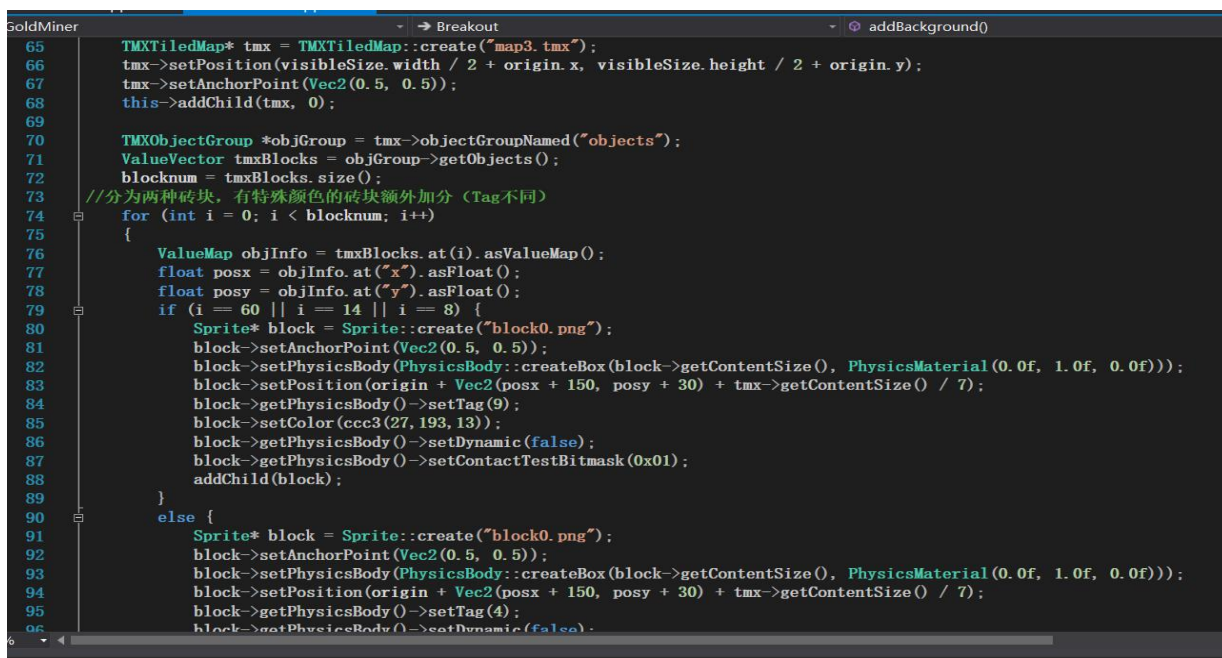




这是当游戏分数达到 150 分以上时，增加一个球，游戏难度提高。



这是加载 tmx 格式地图背景的相关代码，对于 objects 层中的方块，分别添加刚体属性，对于特殊方块利用 setColor 的 ccc3 参数重新设置颜色。





这是利用每个刚体的 Tag 进行的碰撞检测，当检测到球和砖块碰撞时，添加相应分数并移除掉砖块。当检测到球与底部边界碰撞时游戏结束，进入游戏结束页面。

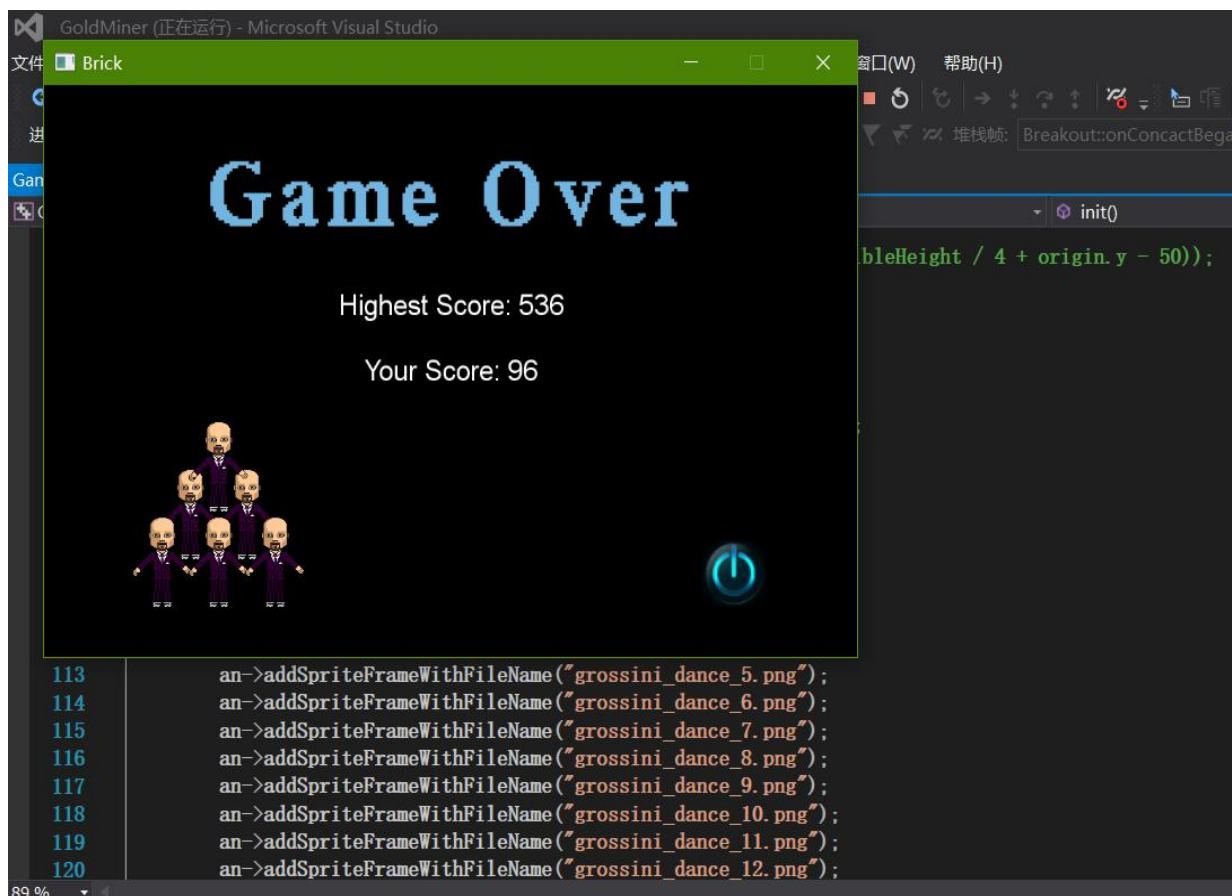
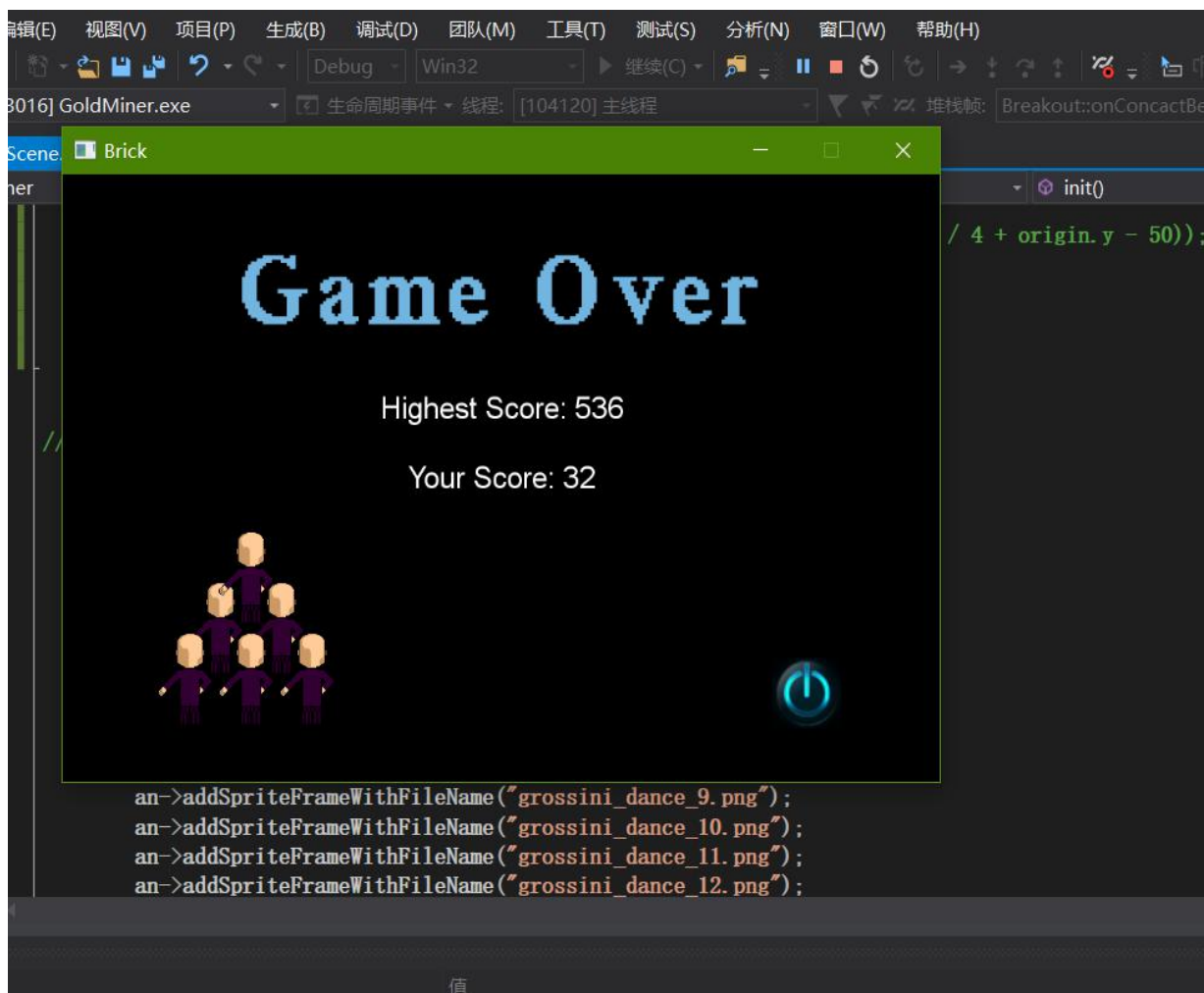
```
ScoreLabel->setString(ScoreStrrrr);
if (blocknum == 0) {
    CCDirector::sharedDirector()->replaceScene(GameOverScene::createScene());
}
}
if ((node1->getTag() == 2 && node2->getTag() == 9) || (node1->getTag() == 9 && node2->getTag() == 2)) {
    if (node1->getTag() == 2 && node2->getTag() == 9) {
        node2->getNode()->removeFromParentAndCleanup(true);
    }
    else if (node1->getTag() == 9 && node2->getTag() == 2) {
        node1->getNode()->removeFromParentAndCleanup(true);
    }
    blocknum -= 1;
    Score += 100;
    CCString *strsqw = CCString::createWithFormat("Score : %d", Score);
    std::string ScoreStrqw = strsqw->getCString();
    ScoreLabel->setString(ScoreStrqw);
    if (blocknum == 0) {
        CCDirector::sharedDirector()->replaceScene(GameOverScene::createScene());
    }
}
if ((node1->getTag() == 1 && node2->getTag() == 3) || (node1->getTag() == 3 && node2->getTag() == 1)) {
    if (node1->getTag() == 1 && node2->getTag() == 3)
        node1->setVelocity(Vec2(0, 0));
    else if (node1->getTag() == 3 && node2->getTag() == 1)
        node2->setVelocity(Vec2(0, 0));
}
if ((node1->getTag() == 2 && node2->getTag() == 5) || (node1->getTag() == 5 && node2->getTag() == 2)) {
    CCDirector::sharedDirector()->replaceScene(GameOverScene::createScene());
}
}
```

这是添加相应的角色并设置位置以及刚体属性的代码。

```
//底下的边界
22 auto deadSp = Sprite::create("deadbar.png");
23 deadSp->setAnchorPoint(Vec2(0.5, 0.5));
24 deadSp->setScaleX(visibleSize.width / deadSp->getContentSize().width);
25 deadSp->setPhysicsBody(PhysicsBody::createBox(deadSp->getContentSize()));
26 deadSp->setPosition(origin.x + visibleSize.width / 2, origin.y + deadSp->getContentSize().height);
27 deadSp->getPhysicsBody()->setDynamic(false);
28 deadSp->getPhysicsBody()->setTag(5);
29 deadSp->getPhysicsBody()->setContactTestBitmask(0x01);
30 addChild(deadSp);
31
32 //板
33 player = Sprite::create("player.jpg");
34 player->setScaleX(1.5f);
35 player->setAnchorPoint(Vec2(0.5, 0.5));
36 player->setPhysicsBody(PhysicsBody::createBox(player->getContentSize(), PhysicsMaterial(0.0f, 1.0f, 0.0f)));
37 player->setPosition(origin.x + visibleSize.width / 2, origin.y + player->getContentSize().height / 2);
38 player->getPhysicsBody()->setTag(1);
39 player->getPhysicsBody()->setAngularVelocityLimit(0);
40 player->getPhysicsBody()->setRotationEnable(false);
41 player->getPhysicsBody()->setMass(10000);
42 player->getPhysicsBody()->setContactTestBitmask(0x10);
43 addChild(player);
44
45 //球
46 void Breakout::addball() {
47     ball = Sprite::create("ball.png");
48     ball->setPosition(origin.x + visibleSize.width / 2, origin.y + ball->getContentSize().height);
49     ball->setPhysicsBody(PhysicsBody::createCircle(ball->getContentSize().height / 2, PhysicsMaterial(0.0f, 1.0f, 0.0f)));
50     ball->getPhysicsBody()->setTag(2);
51     ball->getPhysicsBody()->setVelocity(Vec2(0, 300));
52     ball->getPhysicsBody()->applyImpulse(Vec2(50, 10));
53     ball->getPhysicsBody()->setMass(10);
}
```

4. 游戏结束画面展示。游戏结束时，显示最高分记录以及本次得分。

左下角是小人跟着音乐跳舞的动画，右下角是退出游戏按钮。





这是数据库计分记录的代码以及添加小人跳舞的代码。

```
    }  
    if (!database->getBoolForKey("isExist")) {  
        database->setBoolForKey("isExist", true);  
        database->setIntegerForKey("record", 0);  
    }  
    //计分数据库  
    int record = database->getIntegerForKey("record");  
  
    Size size = Director::getInstance()->getVisibleSize();  
    Point origin = Director::getInstance()->getVisibleOrigin();  
    visibleHeight = size.height;  
    visibleWidth = size.width;  
    //加载相应的图片  
    auto over = Sprite::create("GameOver.png");  
    over->setPosition(Vec2(size.width / 2 + origin.x, size.height / 2 + origin.y + 150));  
    this->addChild(over, 0);  
  
    TTFConfig ttfConfig;  
    ttfConfig.fontFilePath = "fonts/arial.ttf";  
    ttfConfig.fontSize = 24;  
  
    CCString *str = CCString::createWithFormat("%d", Breakout::Score);  
    std::string ScoreStr = str->getCString();  
    Label* yourScoreLabel = Label::createWithTTF(ttfConfig, "Your Score: " + ScoreStr);  
    yourScoreLabel->setPosition(Vec2(origin.x + size.width / 2,  
        origin.y + size.height / 2));  
    addChild(yourScoreLabel);
```

```
03  
04    Size visibleSize = Director::getInstance()->getVisibleSize();  
05    //跳舞的动画  
06    mouse = Sprite::create("grossini.png");  
07    mouse->setPosition(Vec2(visibleSize.width / 2 - 180, 120));  
08    Animation* an = Animation::create();  
09    an->addSpriteFrameWithFileName("grossini_dance_1.png");  
10    an->addSpriteFrameWithFileName("grossini_dance_2.png");  
11    an->addSpriteFrameWithFileName("grossini_dance_3.png");  
12    an->addSpriteFrameWithFileName("grossini_dance_4.png");  
13    an->addSpriteFrameWithFileName("grossini_dance_5.png");  
14    an->addSpriteFrameWithFileName("grossini_dance_6.png");  
15    an->addSpriteFrameWithFileName("grossini_dance_7.png");  
16    an->addSpriteFrameWithFileName("grossini_dance_8.png");  
17    an->addSpriteFrameWithFileName("grossini_dance_9.png");  
18    an->addSpriteFrameWithFileName("grossini_dance_10.png");  
19    an->addSpriteFrameWithFileName("grossini_dance_11.png");  
20    an->addSpriteFrameWithFileName("grossini_dance_12.png");  
21    an->addSpriteFrameWithFileName("grossini_dance_13.png");  
22    an->addSpriteFrameWithFileName("grossini_dance_14.png");  
23    an->setDelayPerUnit(0.1f);  
24    an->setRestoreOriginalFrame(true);  
25    mouse->runAction(RepeatForever::create(Animate::create(an)));  
26    addChild(mouse);  
27    //一共六个人组成金字塔跳舞  
28    mouse1 = Sprite::create("grossini.png");  
29    mouse1->setPosition(Vec2(visibleSize.width / 2 - 230, 120));  
30    mouse1->runAction(RepeatForever::create(Animate::create(an)));  
31    addChild(mouse1);  
32  
33    mouse2 = Sprite::create("grossini.png");  
34    mouse2->setPosition(Vec2(visibleSize.width / 2 - 205, 160));  
35    mouse2->runAction(RepeatForever::create(Animate::create(an)));  
36    addChild(mouse2);  
37  
38    mouse3 = Sprite::create("grossini.png");  
39    mouse3->setPosition(Vec2(visibleSize.width / 2 - 255, 80));  
40    mouse3->runAction(RepeatForever::create(Animate::create(an)));
```

五、项目难点及解决方案

1. 在制作地图时遇到了一些问题，由于将几周前制作的方法忘掉了，制作的地图刚开始没办法将砖块放置在 objects 层，也就无法检测到。后来添加进来以后加载进游戏又出现_tiles 错误的提示。然后就看着教程一步一步制作，参考的网站：



<http://www.cnblogs.com/walzer/archive/2010/11/08/1871561.html>

和 <http://8287044.blog.51cto.com/5179921/1045274>

2. 游戏提示页面的制作，发现将提示的语言一条一条放进去还需要考虑位置坐标，比较复杂，然后就用了美图秀秀直接编辑图片添加文字，直接把编辑好的文字加载进来就可以了。

3. 游戏页面把地图加载进来以后不知道如何处理砖块，因为此时的它们没有刚体属性，也不能在后面进行碰撞检测，在查阅了资料后发现可以利用循环提取 objects 中的每个方块，然后在循环中分别对它们进行位置设定和刚体属性的设置。

4. 游戏结束的页面，要添加小人跳舞的动画，却发现以前用的从 plist 加载动画帧的方法不能用，因为我们的素材是一张一张 png 格式的图片。在网上找了教程后发现可以用 `addSpriteFrameWithFileName` 把一张一张图片添加到帧动画，然后再赋给初始的人物，这样一来就解决掉了问题。

5. 游戏界面编写增加一个球时测试发现球一直从下面冒出来不会停止，检查后发现调用 `addpea` 函数时没有判断是否添加过，只要 150 分以上就调用，造成了无限调用。最后在 .h 文件中增加了一个 flag 标记用来检测是否是第一次添加。

六、项目总结

这次的期末项目，我们做了一个简单的小游戏，游戏的名字叫做“疯狂打砖块”。这个游戏的规则比较简单，相信许多人也都玩过类似的游戏，就是一个打砖块游戏。由于时间的关系，我们只制作了一个关卡，当打完所有的砖块后，那么你就完成了该游戏。但是为了增加游戏的趣味性，我们增加了不同的砖块，当然特殊砖块的分值是与普通砖块不一样的，而且在玩家分



数达到一定程度时，我们会再发射一个小球，这时候你就要同时注意 2 个小球了，因为任何一个小球落地都会使游戏结束。虽然这个游戏玩起来比较简单，但是在制作的过程中会遇到许许多多的问题。首先我们游戏砖块的添加是通过地图实现的，所以我们得自己制作游戏地图，这个工程量比较大，而且在制作的过程中还得注意每个砖块的坐标。地图完成后，在测试的时候也遇到了许多的问题，开始的时候地图加载不进去，后来才发现是地图中一些对象的属性设置有问题，最后经过不断的修改测试，我们终于将地图部分完成。然后将地图中的每个砖块添加到游戏中，以及设置他们的物理属性也给我们带来了很大的麻烦，还有一系列的碰撞检测等等，在真正实现的时候还是会出现各种各样的问题。但是经过组员们的共同努力，不断查阅资料，最终将问题一个一个解决，完成了这次的项目。虽然做的游戏比较简单，但是涉及到的知识点还是比较多，而且经过这次的游戏制作，我们也都学到了许多，游戏的构思能力制作能力都有了很大的提升，团队的协作能力也有显著的提高，并且积累了宝贵的游戏制作经验。而且自己制作的游戏自己玩起来也别有一番感受与成就。

总之，这个学期的现代操作系统与应用开发终于结束了，真是不容易啊，每周都有不小的作业量，还有期中期末项目，虽然让我们有一点点苦不堪言，但真的是学到了很多東西，通过自己一步步的制作调试，进步很大。谢谢老师和 TA 们一学期的付出和帮助，以后我们会继续加油。