1. AppDelegate 类

是cocos2d-x引擎要求实现的游戏应用委托对象，在游戏运行的不同生命周期会调用不同的函数。

1.类成员函数

1. virtual bool applicationDidFinishLaunching()//游戏启动时调用的函数，初始化导演对象
2. virtual void applicationDidEnterBackground()//游戏进入后台时调用的函数
3. virtual void applicationWillEnterForeground()//游戏进入前台时调用的函数

2.类成员函数定义

USING\_NS\_CC 相当于using namespace cocos2d

1. applicationDidFinishLaunching()//游戏进入后台时调用的函数

bool AppDelegate::applicationDidFinishLaunching() {

// initialize director

auto director = Director::getInstance();/**/初始化导演类**

auto glview = director->getOpenGLView();

if(!glview) {  **//设置导演类的OPENGL视图**

#if (CC\_TARGET\_PLATFORM == CC\_PLATFORM\_WIN32) || (CC\_TARGET\_PLATFORM == CC\_PLATFORM\_MAC) || (CC\_TARGET\_PLATFORM == CC\_PLATFORM\_LINUX)

glview = GLViewImpl::createWithRect("HelloWarld", cocos2d::Rect(0, 0, designResolutionSize.width, designResolutionSize.height));

#else

glview = GLViewImpl::create("HelloWarld");

#endif

director->setOpenGLView(glview);

}

// turn on display FPS

director->setDisplayStats(true);**//设置是否在屏幕上显示帧类等信息，一般发布游戏时不显示**

// set FPS. the default value is 1.0/60 if you don't call this

director->setAnimationInterval(1.0f / 60);**//帧率设定为60帧**

// Set the design resolution

glview->setDesignResolutionSize(designResolutionSize.width, designResolutionSize.height, ResolutionPolicy::NO\_BORDER);

auto frameSize = glview->getFrameSize();

// if the frame's height is larger than the height of medium size.

if (frameSize.height > mediumResolutionSize.height)

{

director->setContentScaleFactor(MIN(largeResolutionSize.height/designResolutionSize.height, largeResolutionSize.width/designResolutionSize.width));

}

// if the frame's height is larger than the height of small size.

else if (frameSize.height > smallResolutionSize.height)

{

director->setContentScaleFactor(MIN(mediumResolutionSize.height/designResolutionSize.height, mediumResolutionSize.width/designResolutionSize.width));

}

// if the frame's height is smaller than the height of medium size.

else

{

director->setContentScaleFactor(MIN(smallResolutionSize.height/designResolutionSize.height, smallResolutionSize.width/designResolutionSize.width));

}

register\_all\_packages();

// create a scene. it's an autorelease object

auto scene = HelloWorld::createScene();**//创建场景对象scene**

// run

director->runWithScene(scene);**//运行该场景，即让游戏进入到该场景**

return true;

}

(2) applicationDidEnterBackground()

// This function will be called when the app is inactive. Note, when receiving a phone call it is invoked.

void AppDelegate::applicationDidEnterBackground() {

Director::getInstance()->stopAnimation();**//停止场景中的动画**

#if USE\_AUDIO\_ENGINE

AudioEngine::pauseAll();

#elif USE\_SIMPLE\_AUDIO\_ENGINE

SimpleAudioEngine::getInstance()->pauseBackgroundMusic();**//停止音乐**

SimpleAudioEngine::getInstance()->pauseAllEffects();

#endif

}

(3) applicationWillEnterForeground()

// this function will be called when the app is active again

void AppDelegate::applicationWillEnterForeground() {

Director::getInstance()->startAnimation();//开始场景中的动画

#if USE\_AUDIO\_ENGINE

AudioEngine::resumeAll();

#elif USE\_SIMPLE\_AUDIO\_ENGINE

SimpleAudioEngine::getInstance()->resumeBackgroundMusic();//音乐继续

SimpleAudioEngine::getInstance()->resumeAllEffects();

#endif

}

1. HelloWorld类
2. HelloWorld类继承了cocos2d::Layer类，他被称为层（layer）,layer被放到场景(scene)中

#ifndef \_\_HELLOWORLD\_SCENE\_H\_\_

#define \_\_HELLOWORLD\_SCENE\_H\_\_

#include "cocos2d.h"

class HelloWorld : public cocos2d::Scene

{

public:

static cocos2d::Scene\* createScene();//建立当前场景所在的静态函数

virtual bool init();//声明初始化层HelloWorld实例函数

// a selector callback

void menuCloseCallback(cocos2d::Ref\* pSender);//声明菜单回调函数，用于触摸菜单事件//的回调

// implement the "static create()" method manually

CREATE\_FUNC(HelloWorld);

};

#endif // \_\_HELLOWORLD\_SCENE\_H\_\_

1. Hello world.cpp
2. #include "HelloWorldScene.h"
3. #include "SimpleAudioEngine.h"
4. USING\_NS\_CC;
5. Scene\* HelloWorld::createScene()
6. {
7. return HelloWorld::create();
8. }
9. // Print useful error message instead of segfaulting when files are not there.
10. static void problemLoading(const char\* filename)
11. {
12. printf("Error while loading: %s\n", filename);
13. printf("Depending on how you compiled you might have to add 'Resources/' in front of filenames in HelloWorldScene.cpp\n");
14. }
15. // on "init" you need to initialize your instance
16. bool HelloWorld::init()
17. {
18. //////////////////////////////
19. // 初始化父类
20. if ( !Scene::init() )
21. {
22. return false;//返回true为成功，返回false则是失败
23. }
24. auto visibleSize = Director::getInstance()->getVisibleSize();//定义可视化尺寸
25. Vec2 origin = Director::getInstance()->getVisibleOrigin();//定义视图的可视化原点
26. /////////////////////////////
27. // 2. add a menu item with "X" image, which is clicked to quit the program
28. // you may modify it.
29. //增加一个菜单项单机它的时候退出程序
30. // add a "close" icon to exit the progress. it's an autorelease object
31. auto closeItem = MenuItemImage::create(
32. "CloseNormal.png",
33. "CloseSelected.png",
34. CC\_CALLBACK\_1(HelloWorld::menuCloseCallback, this));//创建一个图片菜单项对象，单击该菜单项的时候回调menuCloseCallback函数
35. if (closeItem == nullptr ||
36. closeItem->getContentSize().width <= 0 ||
37. closeItem->getContentSize().height <= 0)
38. {
39. problemLoading("'CloseNormal.png' and 'CloseSelected.png'");
40. }
41. else
42. {
43. float x = origin.x + visibleSize.width - closeItem->getContentSize().width/2;
44. float y = origin.y + closeItem->getContentSize().height/2;
45. closeItem->setPosition(Vec2(x,y));
46. }//菜单项对象的位置
47. // create menu, it's an autorelease object
48. auto menu = Menu::create(closeItem, NULL);//创建菜单menu对象
49. menu->setPosition(Vec2::ZERO);//定义菜单对象的位置
50. this->addChild(menu, 1);//将菜单对象
51. 添加到HelloWorld场景中
52. /////////////////////////////
53. // 3. add your codes below...

//在此处添加自己的代码

1. // add a label shows "Hello World"
2. // create and initialize a label
3. auto label = Label::createWithTTF("Hello World", "fonts/Marker Felt.ttf", 24)//创建对象
4. if (label == nullptr)
5. {
6. problemLoading("'fonts/Marker Felt.ttf'");
7. }
8. else
9. {
10. // position the label on the center of the screen
11. label->setPosition(Vec2(origin.x + visibleSize.width/2,
12. origin.y + visibleSize.height - label->getContentSize().height));//设置对象水平居中
13. // add the label as a child to this layer
14. this->addChild(label, 1);//将文本对象添加到层HelloWorld上
15. }
16. // add "HelloWorld" splash screen"
17. auto sprite = Sprite::create("HelloWorld.png");//创建精灵对象
18. if (sprite == nullptr)
19. {
20. problemLoading("'HelloWorld.png'");
21. }
22. else
23. {
24. // position the sprite on the center of the screen
25. sprite->setPosition(Vec2(visibleSize.width/2 + origin.x, visibleSize.height/2 + origin.y));//设置精灵对象的位置，位置是屏幕中央
26. // add the sprite as a child to this layer
27. this->addChild(sprite, 0);//将精灵对象添加到层HelloWorld上
28. }
29. return true;
30. }
31. void HelloWorld::menuCloseCallback(Ref\* pSender)
32. {
33. //Close the cocos2d-x game scene and quit the application
34. Director::getInstance()->end();
35. //EventCustom customEndEvent("game\_scene\_close\_event");
36. //\_eventDispatcher->dispatchEvent(&customEndEvent);
37. }