

初识Cocos2d-x

陈漪凡

目录 / contents

01

cocos2d-x简介

02

Cocos2d-x安装配置

03

Hello world

Cocos2d-x 是一个MIT许可证下发布的开源的移动2D游戏框架，它有以下特征:



- **跨平台:** iOS, Android, Win32, WP...
- **多语言:** C++, Javascript (JSB), LUA
- **开源免费:** MIT License
- **简单上手, 运行高效、灵活, 且功能强大**
- **各类编辑器:** CocoStudio, Particle Designer, Tile Map...
- **第三方插件:** Plugin-X, 社交, 广告, 支付...

它有详细的开发者文档

官方API文档 <http://cocos2d-x.org/docs/api-ref/cplusplus/v3x/>

官方wiki <http://www.cocos2d-x.org/wiki>

GitHub Doc <https://github.com/cocos2d/cocos-docs>

用户索引手册 <http://www.cocos.com/doc/>

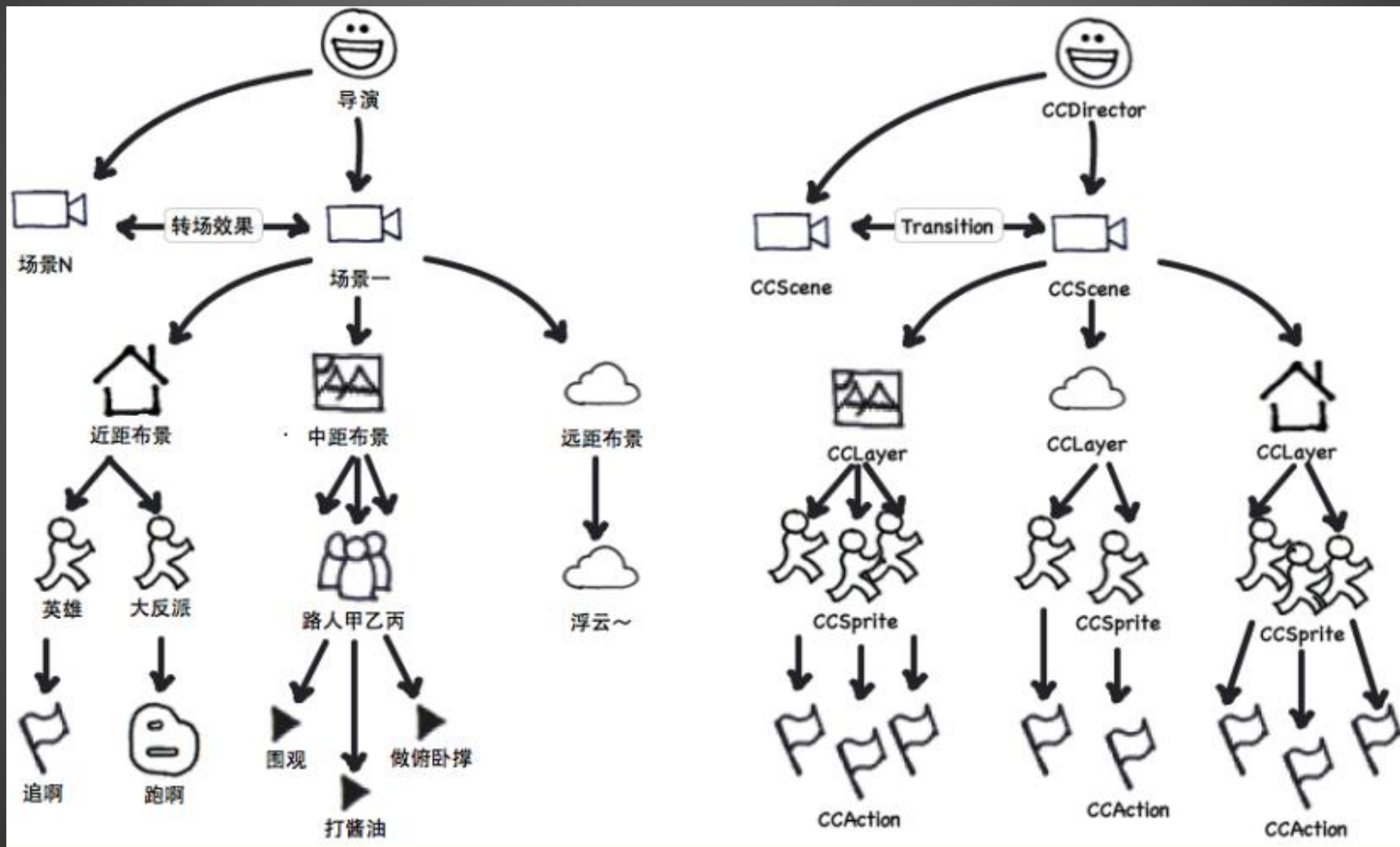
由于使用的人多，遇见了问题便能够更加方便地上网查询解决方案

开发工具：



课程：VS2015 + Cocos2d-x3.10开发环境
编程语言：C++

游戏设计理念：

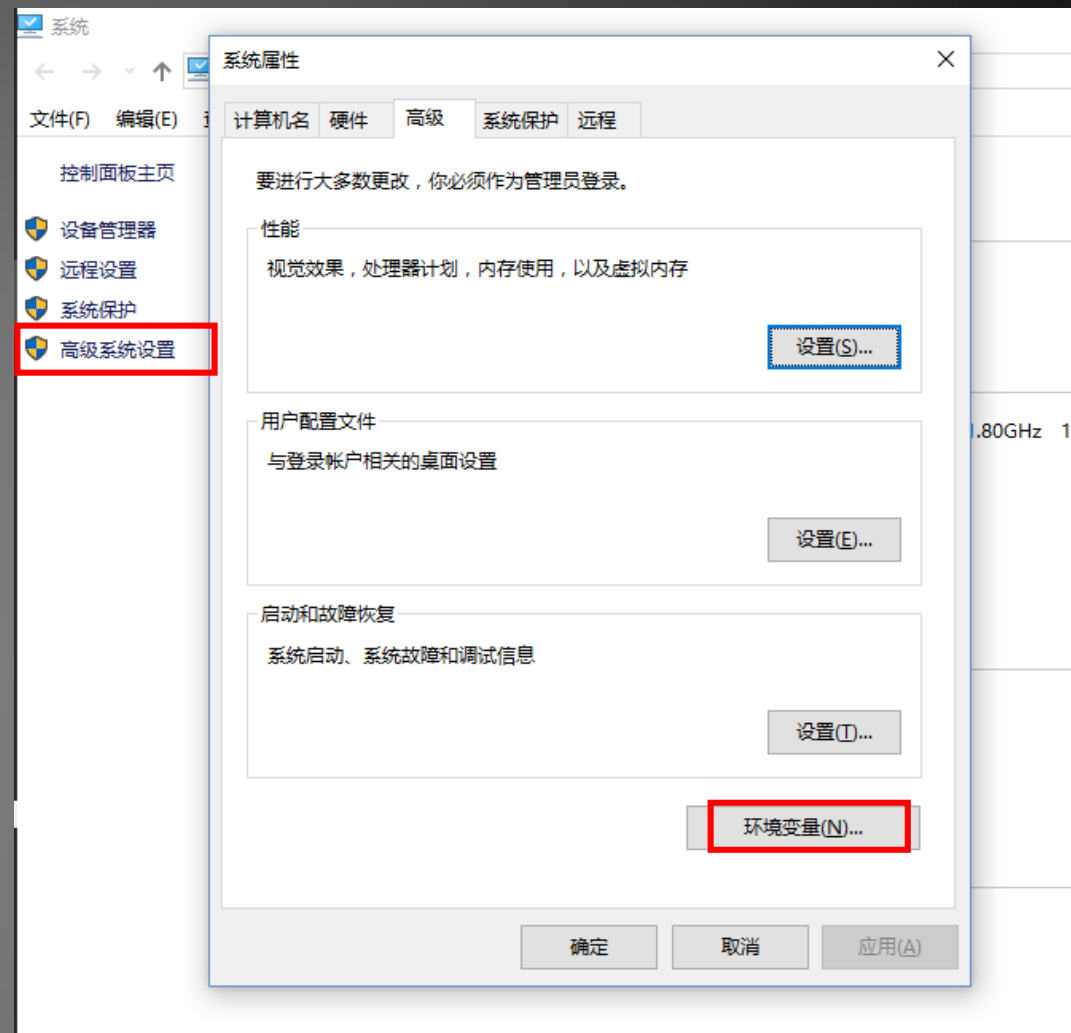


安装cocos2d-x :

- 安装python2.7
- 安装cocos2d-x 3.x
- 新建Hello World

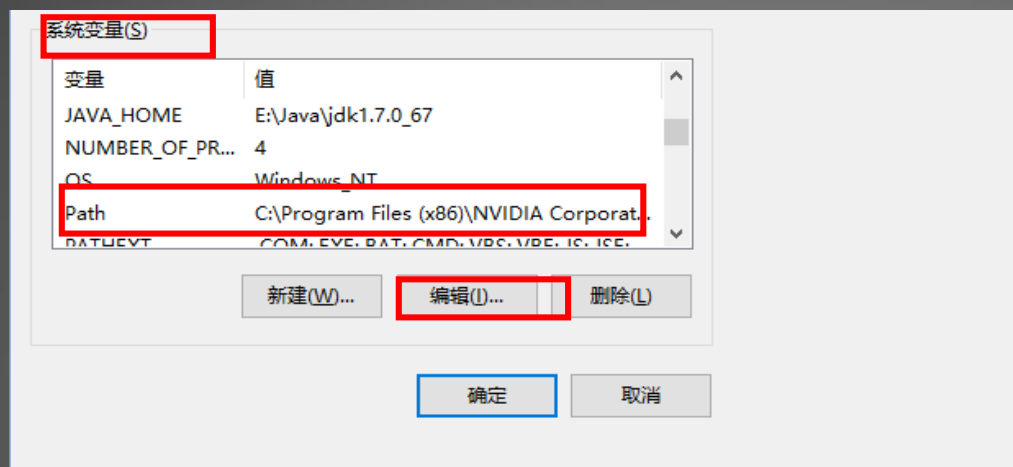
安装python2.7 (暂不支持python 3.5)

- 下载：<https://www.python.org/downloads/>
- 安装
- 设置环境变量
- 测试

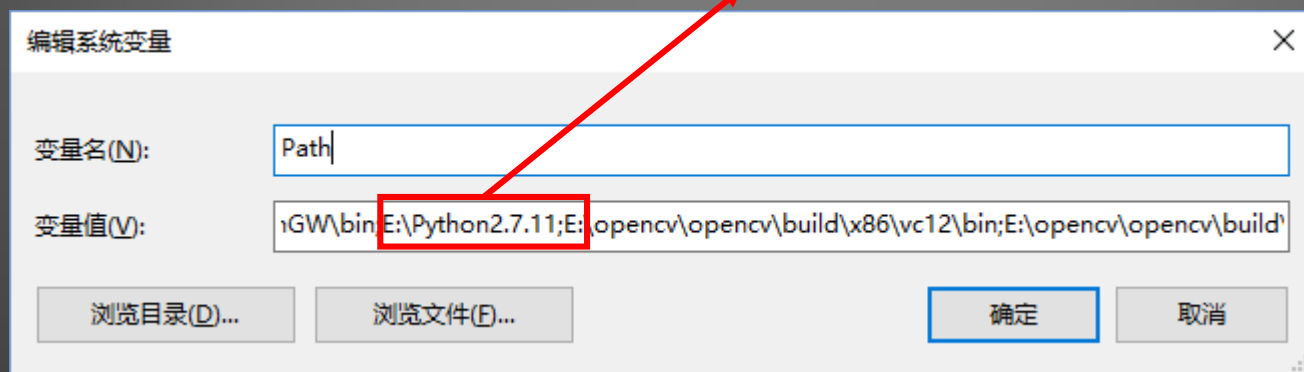


我的电脑—属性—高级系统设置—环境变量

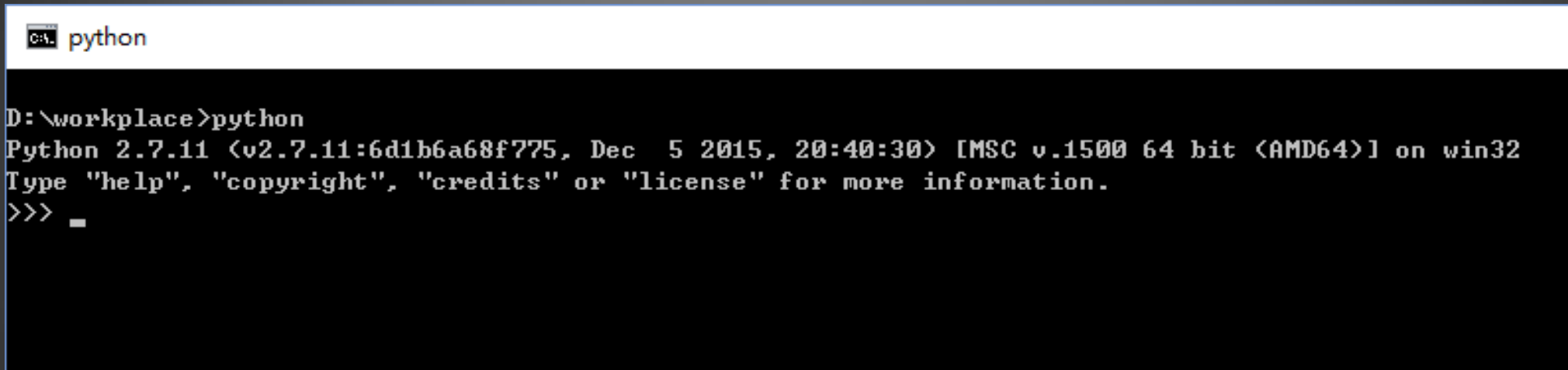
在系统变量中找到Path这一栏，编辑添加路径



你的安装目录，注意分号分隔



测试：任意目录下打开命令行，输入python，出现以下版本信息，安装成功



```
python

D:\workplace>python
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec  5 2015, 20:40:30) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

安装cocos2d-x-3.10

- 下载：<http://www.cocos2d-x.org/download>
- 解压缩
- 安装
- Sample测试
- 新建项目

在解压目录下打开命令行，输入python setup.py

如果不需要编译到安卓，一直按Enter，直至出现下图

```
->Search for environment variable ANT_ROOT...  
  ->ANT_ROOT not found  
  
->Search for command ant in system...  
  ->Command ant not found  
  
->Please enter the path of ANT_ROOT (or press Enter to skip):  
  
Please restart the terminal or restart computer to make added system variables take effect  
  
E:\cocos2d-x-3.10>
```

示例程序

- 进入解压后文件夹的build目录，打开cocos2d-win10.sln
- 设置cpp-tests为启动项，生成解决方案

常见错误：找不到 SDK “WindowsMobile, Version=10.0.10240.0” 。
解决方法：进入tests\cpp-tests\proj.win10文件夹，用记事本打开cpp-tests.vcxproj，搜索10.0.10240.0，全部替换为10.0.10586.0

- 可体验测试，测试代码可供学习参考

文件目录：

文件名	说明
build	<p>官方编译的项目解决方案。</p> <p>集中放在这个文件中，如cocos2d-win32.vc2012.sln、Debug.win32、win32-msvc-2012-x86.cmd等。</p> <p>这些在2.x版本是直接放在根目录的。</p>
cocos	<p>集中了Cocos2d-x开发中使用的所有源文件。</p> <p>除了Cocos2d-x引擎核心部分，还包含声音引擎、物理引擎、网络、GUI等。</p>
extensions	<p>扩展目录。</p> <p>包括一些2.5D特效，网络控制，以及一些GUI等，当要用到这些扩展时：</p> <pre>#include "cocos-ext.h" using namespace cocos2d::extension</pre>
external	<p>第三方目录库。</p> <p>主要包含了Box2D、Chipmunk物理引擎，以及Sqlite3轻型数据库等。</p>
tests	<p>官方样例。</p> <p>包含了官方的C++、lua例子。其中cpp-empty-test就是最简单的例子。</p> <p>另外也包含了一个 <code>cpp-tests</code> 项目，里面涵盖了cocos2dx引擎所有类的使用，所以很重要！</p>
templates	<p>模板目录。</p> <p>提供了各种平台及IDE下创建的cocos2dx新项目的项目模板。</p>
tools	<p>工具目录。</p> <p>提供了各个平台下，创建cocos2dx新项目的脚本工具。</p> <p>其中：<code>\tools\cocos2d-console\bin\cocos.py</code> 就是用来创建cocos2dx新项目的。</p>
download-deps.py	自动下载Cocos2d-x所需的第三方插件（如果有的话）
setup.py	配置Cocos2d-x的环境变量的脚本。

COCOS :

文件名	说明
base	Cocos2d-x引擎的核心部分之一。 存放一些基础类，如Ref、Director、Vector、Map、Value等。
2d	Cocos2d-x引擎的核心部分之一。 存放一些核心部分的类，如Sprite、Layer、Label、Menu等。
3d	新增了对部分3D的支持。 有Sprite3D、Animate3D等，呵呵，可能将来要出cocos3dx了吧。
math	存放引擎中与数学相关的类文件。 如坐标类Vec2.h、几何类CCGeometry.h（定义了Rect、Size）等。
renderer	引擎的渲染文件。 Texture2D、Renderer、Shader等。
deprecated	存放2.x中即将被废弃的规则，目的是为了兼容2.x。 例如，使用typedef来对CC的兼容（如：typedef Label CCLabel）。
platform	不同平台下的入口类。 CCApplication.h、CCGLView.h、CCGL.h、CCStdC.h
audio	声音引擎相关源文件。 在2.x里是放在根目录的CocosDenshion文件夹。
physics	物理引擎相关源文件。 chipmunk。
network	网络相关相关源文件。 HttpClient、WebSocket。
ui	GUI相关源文件。
editor-support	对编辑器的支持。 cocosbuilder、Cocos Studio、Spine。
scripting	lua脚本相关源文件。
cocos2d.h	Cocos2d-x引擎核心头文件，包含了引擎中的所有类。 可以打开它，查看引擎的所有源文件所在位置。

新建项目

```
cocos new Demo -p com..cc.c -l cpp -d D:\cocosExample
```

Demo是项目名称 -p后接包名

-l后接开发语言类型 -d后接项目存放目录

Hello World:













- 项目文件结构
- 游戏生命周期
- Hello world
- 作业

打开新建的项目：

需要完成的程序代码

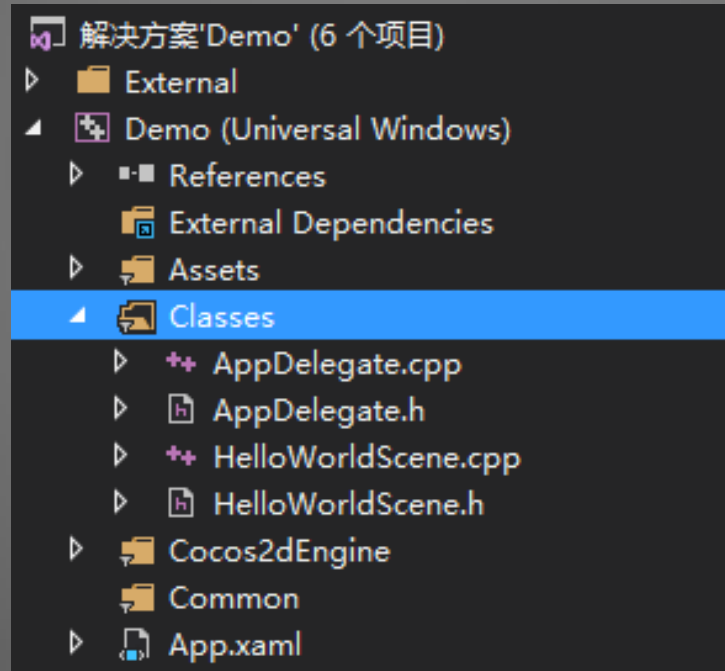
项目工程文件

引用的资源

	Classes	2016/3/20 15:26	文件夹	
	cocos2d	2016/3/20 15:27	文件夹	
	proj.android	2016/3/20 15:26	文件夹	
	proj.android-studio	2016/3/20 15:26	文件夹	
	proj.ios_mac	2016/3/20 15:27	文件夹	
	proj.linux	2016/3/20 15:26	文件夹	
	proj.win8.1-universal	2016/3/20 15:27	文件夹	
	proj.win10	2016/3/20 15:27	文件夹	
	proj.win32	2016/3/20 15:27	文件夹	
	Resources	2016/3/20 15:26	文件夹	
	.cocos-project.json	2016/3/20 15:27	JSON 文件	1 KB
	CMakeLists.txt	2016/3/20 15:27	文本文档	6 KB

项目结构：

了解AppDelegate, HelloWorldScene



AppDelegate :

游戏启动完成

应用程序进入后台

游戏从后台恢复

```
/**
 * @brief    Implement Director and Scene init code here.
 * @return true    Initialize success, app continue.
 * @return false   Initialize failed, app terminate.
 */
virtual bool applicationDidFinishLaunching();

/**
 * @brief    The function be called when the application enter background
 * @param    the pointer of the application
 */
virtual void applicationDidEnterBackground();

/**
 * @brief    The function be called when the application enter foreground
 * @param    the pointer of the application
 */
virtual void applicationWillEnterForeground();
```

AppDelegate :

创建导演

展示当前帧率

设置帧率

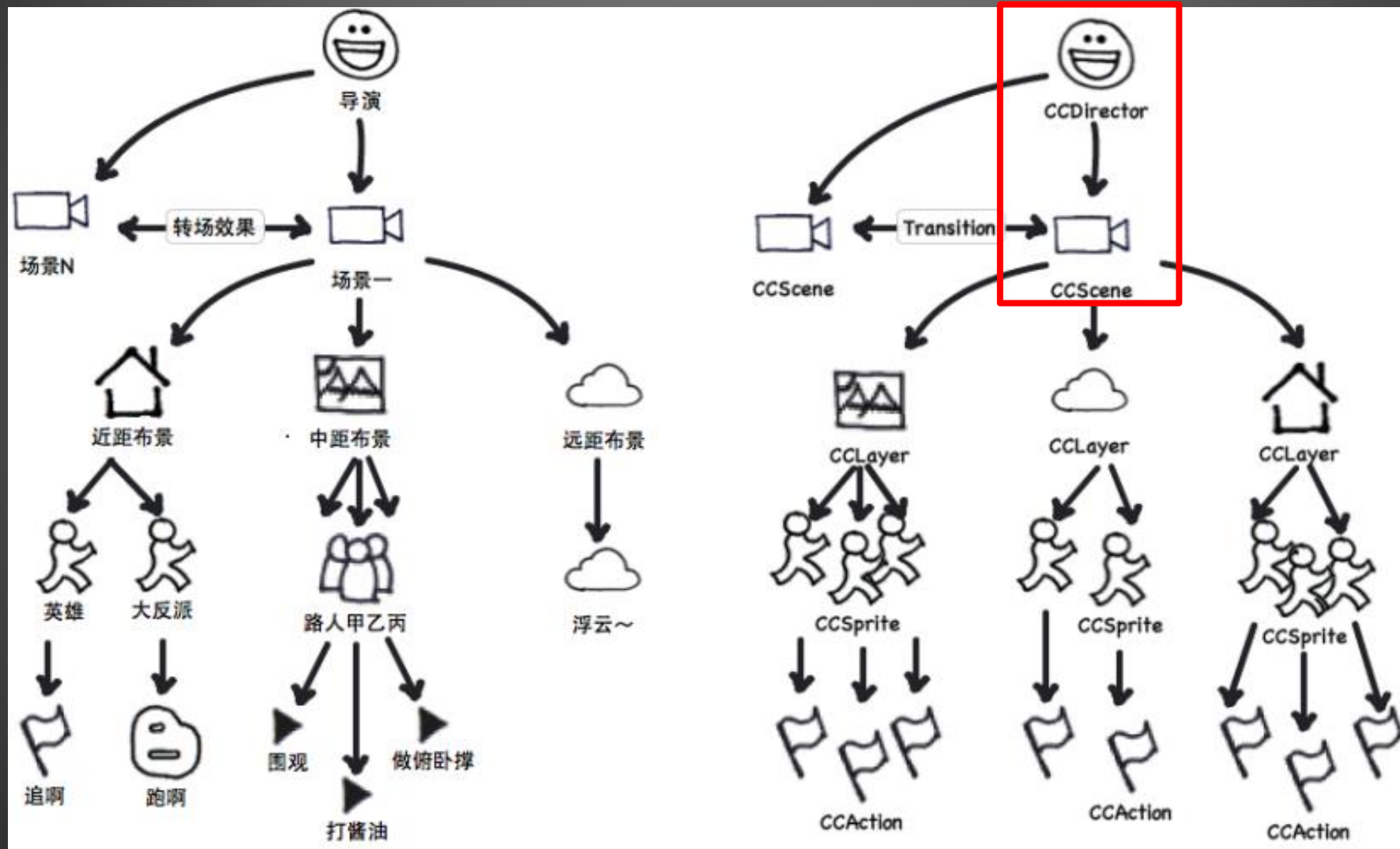
创建并设置初始场景

帧率指每秒刷新页面的次数

```
bool AppDelegate::applicationDidFinishLaunching() {  
    // initialize director  
    auto director = Director::getInstance();  
    auto glview = director->getOpenGLView();  
    if(!glview) {  
        glview = GLViewImpl::create("My Game");  
        director->setOpenGLView(glview);  
    }  
  
    // turn on display FPS  
    director->setDisplayStats(true);  
  
    // set FPS. the default value is 1.0/60 if you don't call this  
    director->setAnimationInterval(1.0 / 60);  
  
    // create a scene. it's an autorelease object  
    auto scene = HelloWorld::createScene();  
  
    // run  
    director->runWithScene(scene);  
  
    return true;  
}
```

确保只有一个
Director

游戏设计理念：



HelloWorld.h :

场景

初始化helloworld

退出按钮的函数

宏，创建helloworld对象

```
#include "cocos2d.h"

class HelloWorld : public cocos2d::Layer
{
public:
    // there's no 'id' in cpp, so we recommend returning the class instance pointer
    static cocos2d::Scene* createScene();

    // Here's a difference. Method 'init' in cocos2d-x returns bool, instead of returning void in cocos2d-1.x
    virtual bool init();

    // a selector callback
    void menuCloseCallback(cocos2d::Ref* pSender);

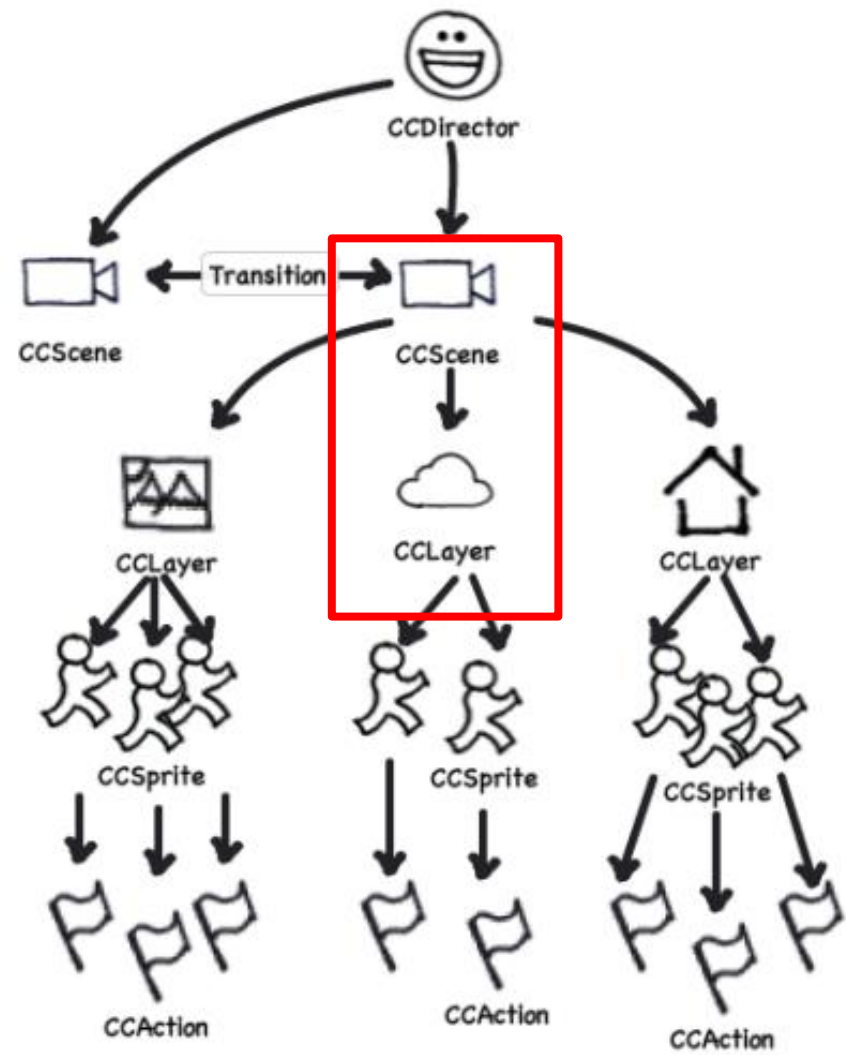
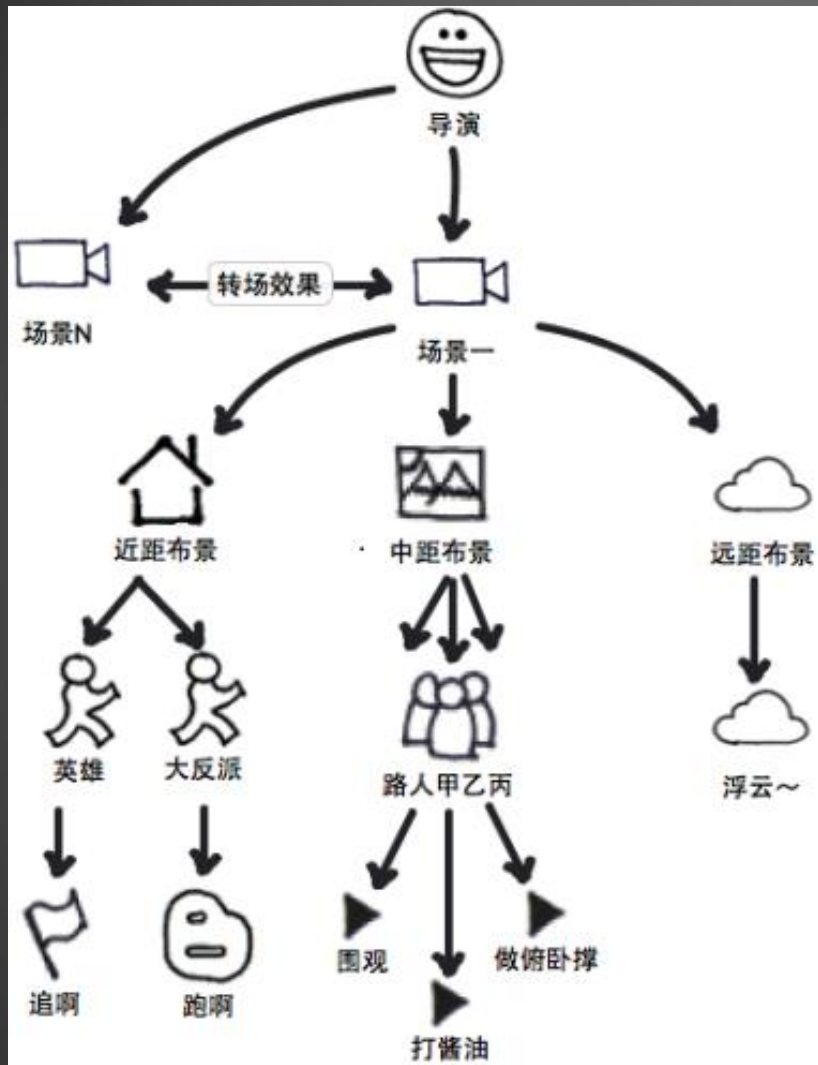
    // implement the "static create()" method manually
    CREATE_FUNC(HelloWorld);
};
```


HelloWorld.cpp :

```
Scene* HelloWorld::createScene()  
{  
    // 'scene' is an autorelease object  
    auto scene = Scene::create();  
    // 'layer' is an autorelease object  
    auto layer = HelloWorld::create();  
    // add layer as a child to scene  
    scene->addChild(layer);  
    // return the scene  
    return scene;  
}
```

在这段代码中，首先利用 Scene::create 方法创建了一个空场景，然后利用 HelloWorld::create 方法创建一个 HelloWorld 层的实例，最后调用 scene 对象的 addChild 方法来把创建的层添加到场景之中。

HelloWorld.h文件 `CREATE_FUNC(HelloWorld);`



helloworld :

初始化HelloWorld层

```
// 1. super init first
if ( !Layer::init() )
{
    return false;
}
```

```
Size visibleSize = Director::getInstance()->getVisibleSize();
```

```
Vec2 origin = Director::getInstance()->getVisibleOrigin();
```

得到窗口大小

得到坐标原点

helloworld :

在这段代码中，创建了一个MenuItem关闭按钮，并添加到了一个菜单中，设置了按钮的位置，让它在场景中显现，并且点击时将会触发窗口关闭事件

```
////////////////////////////////////  
// 2. add a menu item with "X" image, which is clicked to quit the program  
//    you may modify it.  
  
// add a "close" icon to exit the progress. it's an autorelease object  
auto closeItem = MenuItemImage::create(  
    "CloseNormal.png",  
    "CloseSelected.png",  
    CC_CALLBACK_1(HelloWorld::menuCloseCallback, this));  
  
closeItem->setPosition(Vec2(origin.x + visibleSize.width - closeItem->getContentSize().width/2 ,  
    origin.y + closeItem->getContentSize().height/2));  
  
// create menu, it's an autorelease object  
auto menu = Menu::create(closeItem, NULL);  
menu->setPosition(Vec2::ZERO);  
this->addChild(menu, 1);
```

————→ 在Resources文件夹

HelloWorld.cpp :

窗口关闭事件

```
void HelloWorld::menuCloseCallback(Ref* pSender)
{
    Director::getInstance()->end();

#if (CC_TARGET_PLATFORM == CC_PLATFORM_IOS)
    exit(0);
#endif
}
```

helloworld :

在这段代码中，创建了一个“Hello World”的label和一个精灵，这个精灵作为游戏的背景。

```
auto label = Label::createWithTTF("Hello World", "fonts/Marker Felt.ttf", 24);

// position the label on the center of the screen
label->setPosition(Vec2(origin.x + visibleSize.width/2,
                        origin.y + visibleSize.height - label->getContentSize().height));

// add the label as a child to this layer
this->addChild(label, 1);

// add "HelloWorld" splash screen"
auto sprite = Sprite::create("HelloWorld.png");

// position the sprite on the center of the screen
sprite->setPosition(Vec2(visibleSize.width/2 + origin.x, visibleSize.height/2 + origin.y));

// add the sprite as a child to this layer
this->addChild(sprite, 0);

return true;
```

作业：

- 1、安装cocos2d-x，运行测试项目
- 2、制作自己的hello world界面
其中有自己的姓名、学号，更换背景图片
加分项：添加一个MenuItem，触发事件可以为空
- 3、作业提交：提交实验报告（文档），Classes（文件夹），Resources（文件夹）。实验报告要求有截图。



THANKS FOR WATCHING