# CrossApp之HelloWorld详解

学习任何一门开发语言，通常都是由HelloWorld开始的，CrossApp也不例外，CrossApp创建的工程就是一个最简单的HelloWorld程序。

运行CrossApp创建的工程，在工程的Classes目录下，工程为我们创建了AppDelegate、RootWindow和FirstViewCOntroller三个类。首先打开AppDelegate.cpp，找到applicationDidFinishLaunching方法。

bool AppDelegate::applicationDidFinishLaunching()

{

// initialize director

CAApplication\* pDirector = CAApplication::getApplication();

CCEGLView\* pEGLView = CCEGLView::sharedOpenGLView();

pDirector->setOpenGLView(pEGLView);

// run

pDirector->runWindow(RootWindow::create());

}

在这个方法的最后一行pDirector->runWindow(RootWindow::create())，从这里开始，就真正的运行了我们的CrossApp工程，我们可以不用去彻底研究AppDelegate的底层是如何实现的，CrossApp在底层为我们屏蔽了平台的差异性，使我们可以在不同平台上获得统一的编码入口。在这行代码中运行了RootWindow创建的window，这里的RootWindow类是派生自CAWindow，我们通过工厂方法create创建一个window作为整个程序的界面容器。

每个CrossApp程序都必须要有一个window，并且这个window是唯一的，我们所能看到的界面都是在window上显示的。打开RootWindow.cpp，我们看到在create方法中创建了一个window，同时在init方法中，我们给window设置了一个根视图管理器。

bool RootWindow::init()

{

if (!CAWindow::init())

{

return false;

}

FirstViewController\* \_viewController = new FirstViewController();

\_viewController->init();

this->setRootViewController(\_viewController);

\_viewController->release();

return true;

}

和window一样，每个CrossApp应用程序也需要设置一个根视图管理器，用于管理界面，在上面的代码中，FirstViewController就是当前HelloWorld中的根视图管理器。打开FirstViewController.cpp，在viewDidLoad方法中，我们创建了一个imageView用于显示图片和一个label用于显示文字。

void FirstViewController::viewDidLoad()

{

// Do any additional setup after loading the view from its nib.

CCRect winRect = this->getView()->getBounds();

CAImageView\* imageView = CAImageView::createWithImage(CAImage::create("HelloWorld.png"));

imageView->setFrame(winRect);

this->getView()->addSubview(imageView);

CALabel\* label = CALabel::createWithCenter(CCRect(winRect.size.width\*0.5, winRect.size.height\*0.5-270, winRect.size.width, 200));

label->setTextAlignment(CATextAlignmentCenter);

label->setVerticalTextAlignmet(CAVerticalTextAlignmentCenter);

label->setFontSize(72 \* CROSSAPP\_ADPTATION\_RATIO);

label->setText("Hello World!");

label->setColor(CAColor\_white);

this->getView()->insertSubview(label, 1);

}

上面说过window用于显示界面，我们在应用程序中所能看见的界面其实都是view，window也是view，每个视图管理器中都包含一个根view。在上面的方法中，我们就是通过this->getView()来获取到FirstViewController的根view，然后把图片和文字添加到根view上面显示出来，这样就运行起了一个最简单的CrossApp应用程序。