**Cocos2d-x之路**

**0.前序**

为什么要捡一手Cocos2d-x？

只能说，自己玩玩，学设计模式会用到，以及写给社团想入门游戏开发的新人看吧...

尽管整个高中都给了Cocos2d-x，但现在时代更替，自己也走了全栈3D的路，成了Unity的开发者，可能以后会去用虚幻，也许未来回去用别的引擎，或许会去自研引擎，总之既然选了游戏就得跟上时代，不可能吊死在某个坑里

不过入门游戏开发，起码在很长一段时间里，我还是推荐从Cocos2d-x开始，而不是直接Unity真香或者其它上层的，可视化编辑的引擎

**Unity强大的功能很容易让新人迷失自我，过渡膨胀**

事实上现在B站上，是有这样的一匹用着Unity，自己没点东西，全靠搬素材，剪视频，但也迎合了这个浮躁的时代，走近了众人的目光中的“独立游戏开发者”

但自己部门的培训，面对初学C语言，真的想入门游戏开发，想有一番作为的新人

我断然不能直接让他们迷失在Unity之中

**个人依然十分推荐，从C/C++，Cocos来入门游戏开发**

相比Unity，Cocos更接近底层，更依靠编码来实现功能，使得在学习Cocos的同时，真的能学会很多东西

在Cocos还没凉的那个年代，**让我入门游戏开发的是这款引擎，让我真正学会面向对象的是这款引擎，让我真正学会Debug的是这款引擎，让我学会在一开始就绕开Bug的是这款引擎**

**从0开始，面向对象手撸出一个完整的游戏，让我知道了怎么去抽象出对象，分配功能，设计耦合与解耦，怎么为自己铺路，怎么避免踩坑，怎么去调试，怎么去解决问题**

我的第一篇文章，2D游戏手柄的Unity实现，也是基于当初在Cocos上实现的2D虚拟手柄

**总之这款引擎真的教会了我很多**

本篇已更新**环境配置**即**初始化**相关内容

针对使用**Cocos2d-x 引擎 v3.17.2（C++）**结合**CocosStdio v3.10场景UI编辑器**，进行**windows下的游戏开发学习**

我们使用了CocosStdio v3.10版本，由于v3.10版本.ccs项目创建和v3.10项目创建被绑定

我们使用了创建一份v3.10预备工程，**并拷贝.ccs工程到v3.17.2工程中的方法**

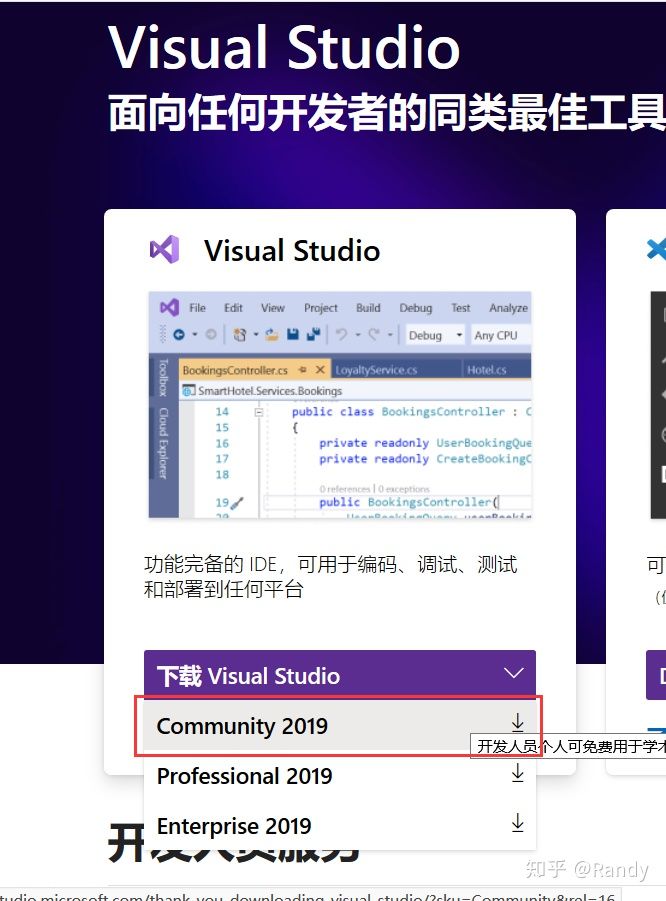
从而完成**Cocos2d-x v3.17.2与CocosStdio v3.10的结合使用**

**1.环境配置**

**---1.1 下载世界上最好的编译器**

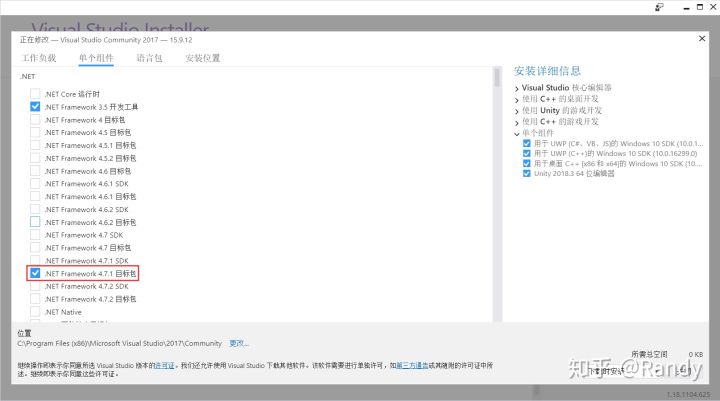
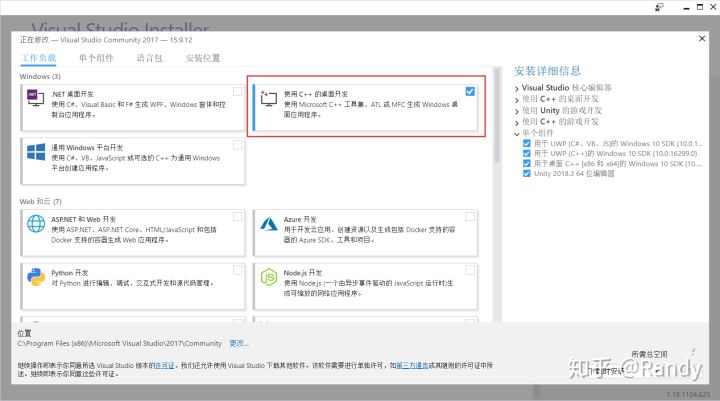
下载**Visual Stdio 2017/2019社区版**

[Visual Studio IDE, Code Editor, Azure DevOps, & App Center - Visual Studio​visualstudio.microsoft.com/](https://link.zhihu.com/?target=https%3A//visualstudio.microsoft.com/)



在安装时设定工作负载配置

1.使用C++的桌面开发 2.使用C++的游戏开发 [http://3.Net](https://link.zhihu.com/?target=http%3A//3.Net) framework支持（为了能够使用CocosStdio）

为了CocosStdio

如果你已经安装了Visual Stdio，或是错过了安装时的工作负载设置，你可以在这里进行再次配置



**---1.2 下载python2.7并配置环境变量**

Cocos2d工程的拷贝创建，以及Android打包操作，需要使用**python脚本**完成

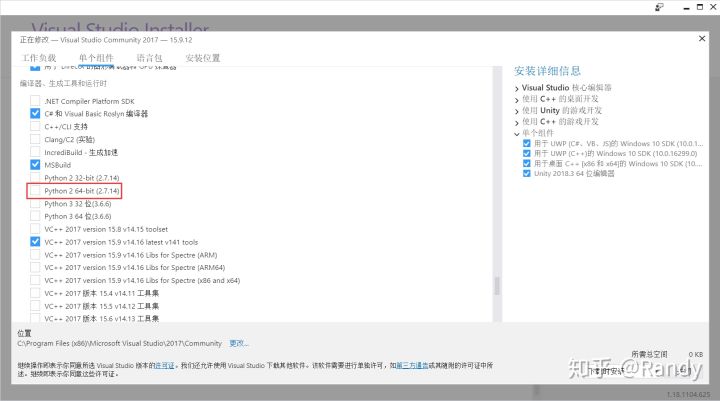
我们需要**安装python 2.7版本**并**配置系统环境变量**，保证能够使用这些脚本

首先你需要获得这样的一个安装包，可以去Python官网

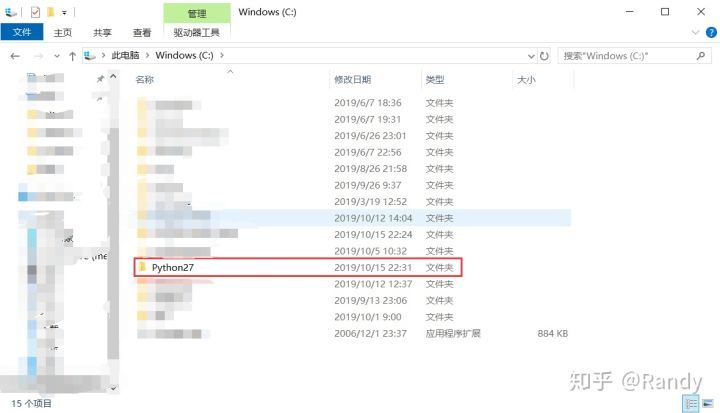
[https://www.python.org/downloads/](https://link.zhihu.com/?target=https%3A//www.python.org/downloads/)

https://pic2.zhimg.com/80/v2-fd0cfade63f83d4533bfdba5fdb25395_720w.png

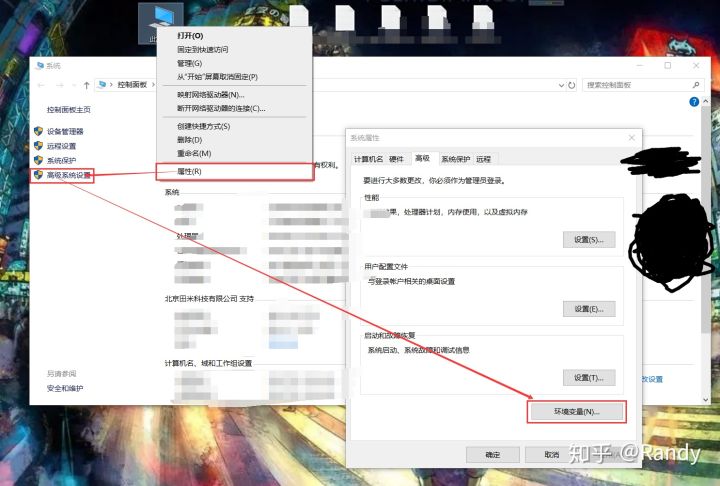
当然万能的Visual Stdio也可以帮你，但谁让上面那个安装包资源就在我手边呢（极其嚣张）

你可以通过这个安装Python 2.7 但安装目录可能需要花时间查找

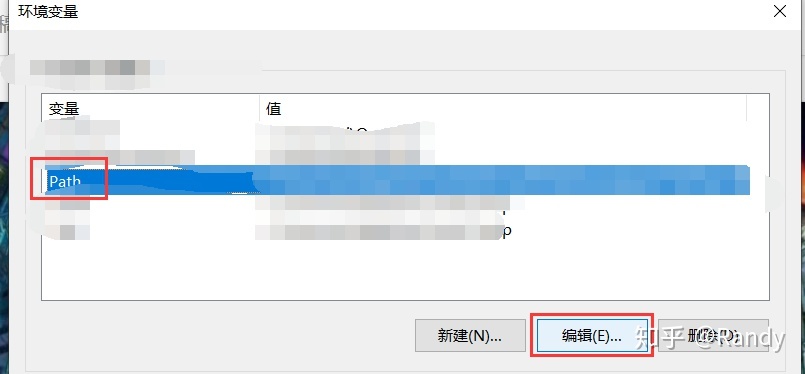
安装完成后在C盘根目录下可以找到Python27文件夹（使用安装包默认安装）



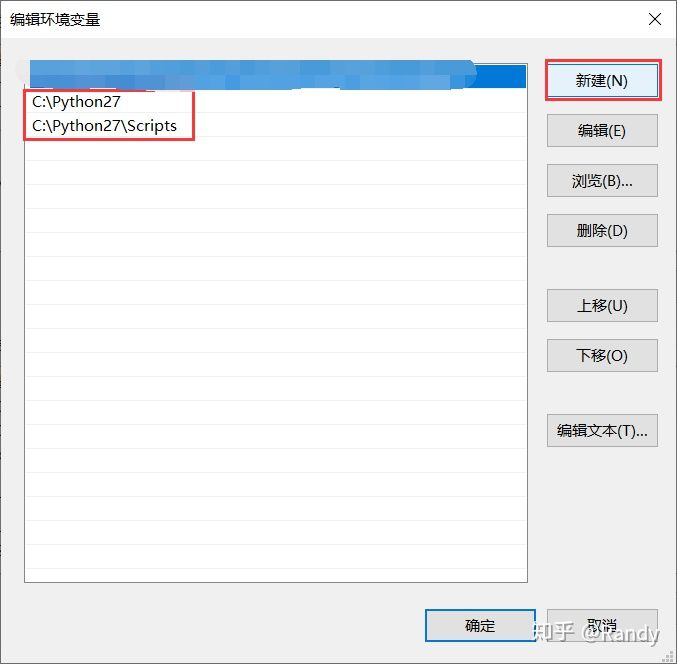
之后我们右键我的电脑---属性---高级系统设置---环境变量



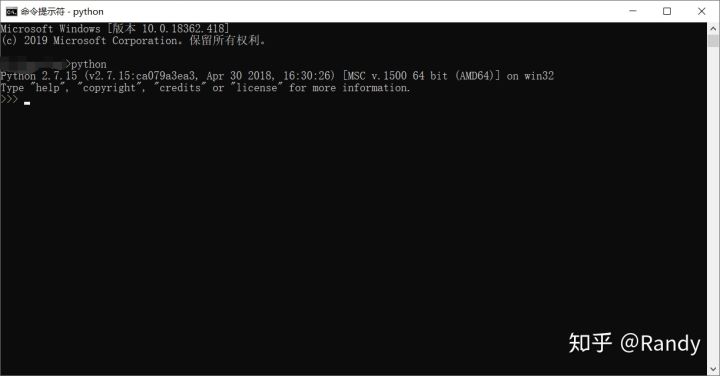
找到一个名为Path的环境变量，编辑



新建，添加这两个目录位置

如果你使用了别的安装目录，需要做出对应修改，总之你需要在Path环境变量中添加Python27和Python27/Scrpit目录在你硬盘中的位置

打开命令提示符（cmd），输入python，显示如下，说明系统环境变量配置完成



**---1.3 下载CocosStdio（附Stdio工程管理器）**

点击链接下载安装包

这里没有办法，最高版本的v3.10Stdio当年和v3.10的库还付了一个工程管理器绑在一起的...

[http://www.cocos2d-x.org/filedown/CocosForWin-v3.10.exe​www.cocos2d-x.org/filedown/CocosForWin-v3.10.exe](https://link.zhihu.com/?target=http%3A//www.cocos2d-x.org/filedown/CocosForWin-v3.10.exe)

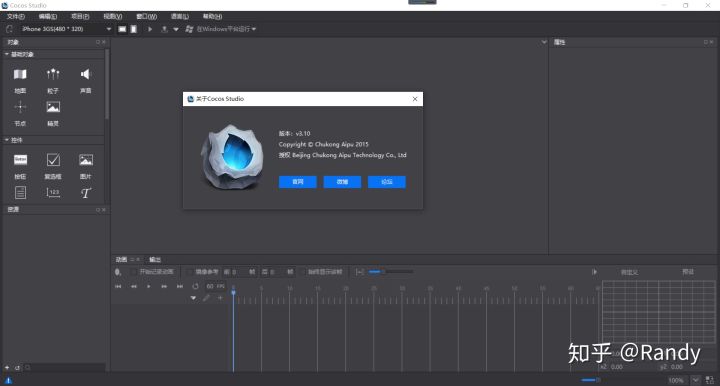
**注意如果安装过程中提示需要安装.Net Framework支持，请见1.1 Visual Stdio工作负载配置**

下载完成后你会得到这两个应用程序

这个Cocos是**v3.10出现的一个工程管理器**，可以直接完成v3.10版本的工程创建

CocosStdio是Cocos早期官方推出的一套**场景UI编辑器**，现已完全被Creator编辑器所取代

（但我就是要用，因为我不用Creator，我不写JS）

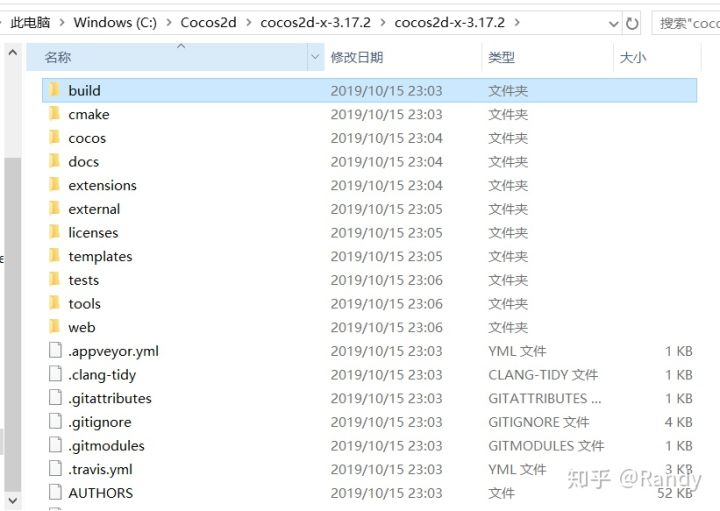


**---1.4 下载Cocos2d-x库**

点击链接下载Cocos2d-x v3.17.2库

[World’s #1 Open Source Game Development Platform​cocos2d-x.org/filedown/cocos2d-x-3.17.2](https://link.zhihu.com/?target=http%3A//cocos2d-x.org/filedown/cocos2d-x-3.17.2)

下载完成后解压到一个目录下



**---1.5 使用setpy配置环境变量**

我们需要使用Cocos2d 的setup.py设置环境变量，将cocos命令行工具添加到环境变量中

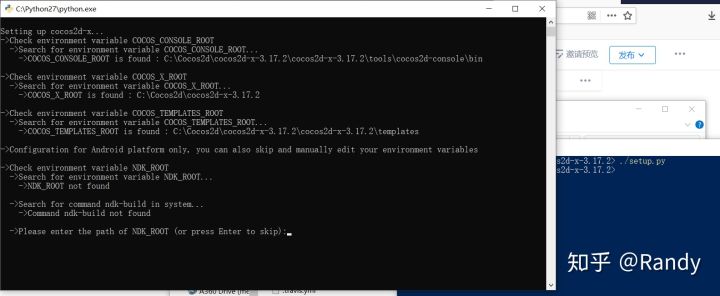
在Cocos2d库解压后的根目录，摁住Shift点击鼠标左键，在此处打开Powershell窗口

什么？为什么不用cmd？因为win10现在默认是没有“在此处打开命令窗口的”，反正Powershell也行就是了，我怎么可能说去cd目录呢？

首先键入命令

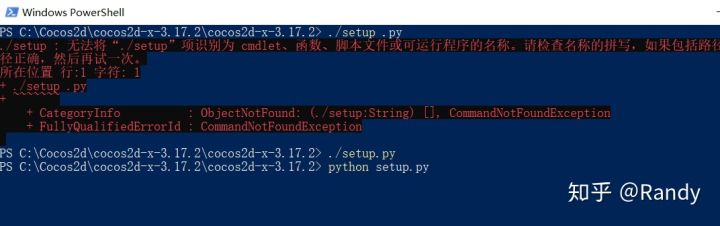
./setup.py

弹出的窗口是为设置Android跨平台开发的NDK AndroidSDK JDK相关，这里我们不用管，一路回车

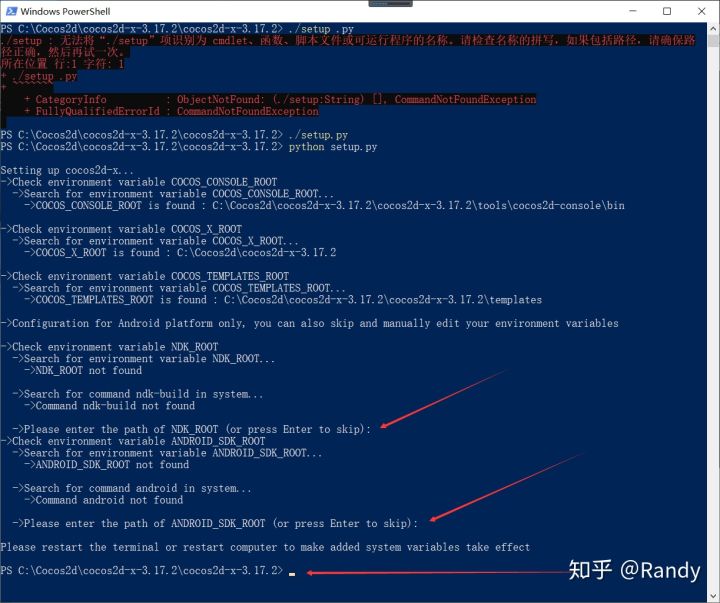


随后键入命令

python setup.py

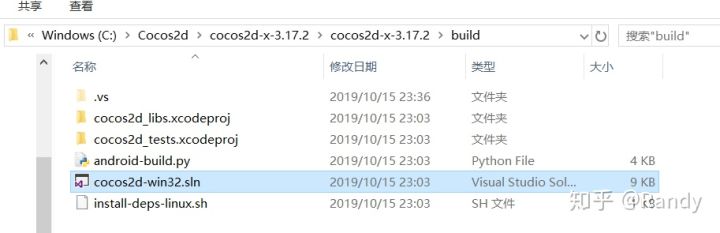


我们需要摁两下回车，直到这个python脚本运行完

运行完成如图

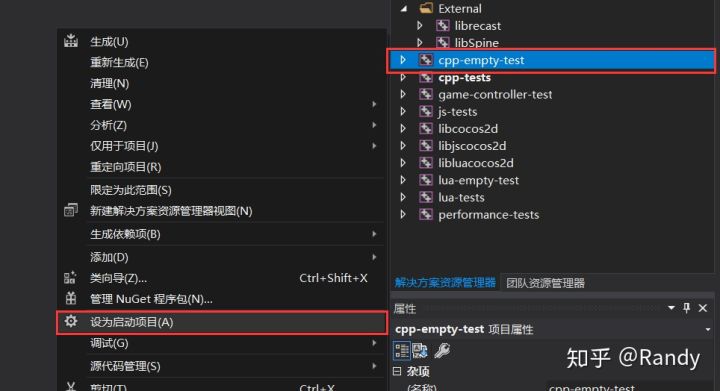
**---1.6 编译Cocosd空项目**

进入这个目录，打开cocos2d-win32.sln Visiual Stdio工程

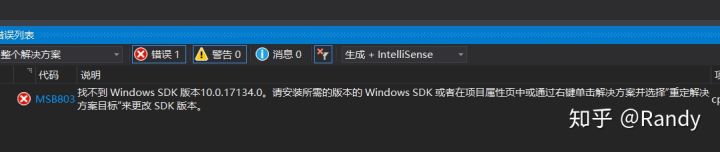


一直点确定，把里面包括的所以解决方案都打开

之后我们将cpp-empty-test设为启动项目，编译



可能会提示找不到对应的windowsSDK版本

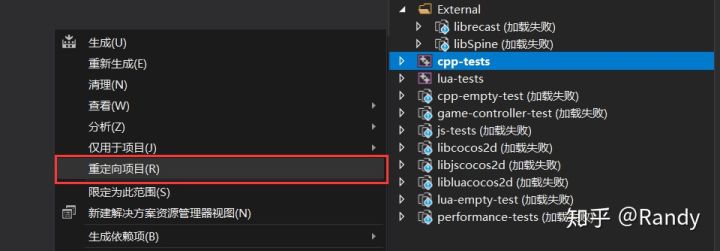


这里不要慌，**如果你在1.1中勾选了使用C++的游戏开发，那么会送你最高版本的Windows 10 SDK**

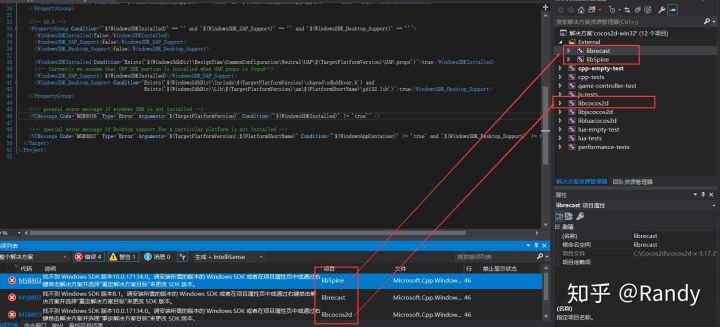
根据**向下兼容原则**，我们现在拥有的SDK版本应该可以应用到Cocosd-x v3.17.2的工程中



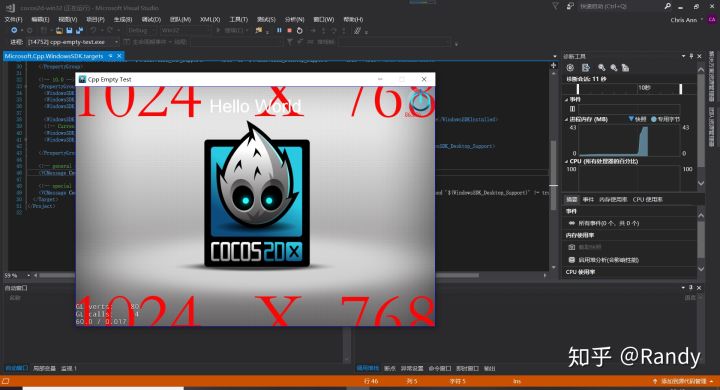
我们只需**重定向项目**，选择已有的Windows SDK版本确定即可



再次编译我们发现这三个静态链接库项目也需要进行Windows SDK版本的重定



编译成功，运行出了这样的一个空项目窗口



此时说明我们的编译开发环境已经配置完成

**2.初始化**

让我们开启Cocos2d-x之路吧！！！

**---2.1 创建Cocos2d工程**

进入这个目录，摁住Shift点击鼠标右键，选择在此处打开Powershell窗口

什么？为什么不用cmd？因为win10现在默认是没有“在此处打开命令窗口的”，反正Powershell也行就是了，我怎么可能说去cd目录呢？

在你的硬盘中**准备一个集中保存Cocos2d工程的地方**

例如：C:\Users\Public\Documents\CocosProj

我们在打开的Powershell窗口中输入：

cocos new NewProject01 -p com.cocos2dx.pack -l cpp -d C:\Users\Public\Documents\CocosProj

命令解释：

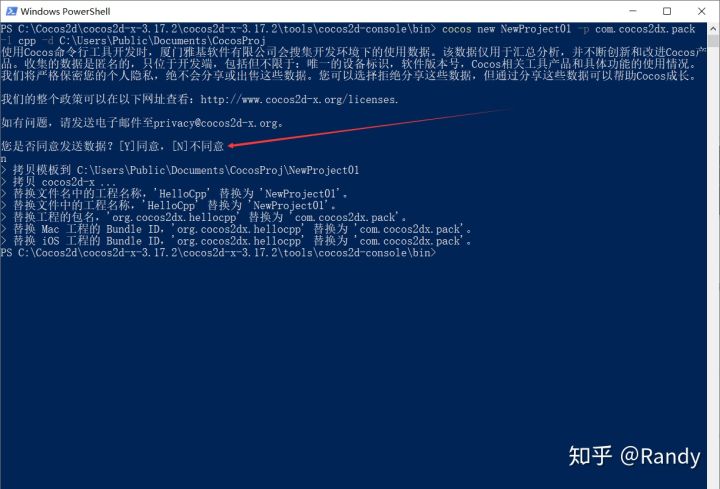
cocos new 创建新的Cocos2d工程（使用Python脚本进行复制拷贝）

NewProject01 是工程名称

-l cpp 指定语言为C++

-p com.cocos2dx.pack 指定Android包名

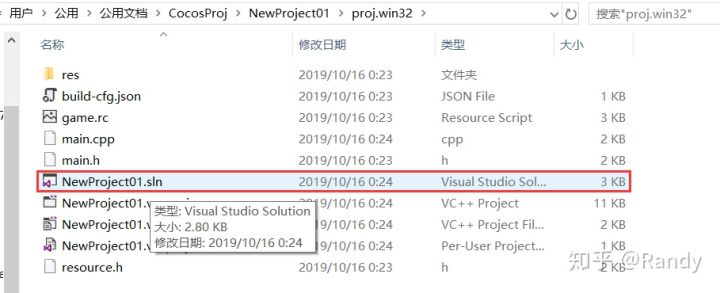
-d C:\Users\Public\Documents\CocosProj 指定保持路径

运行完成如图，中间提示你是否发送数据，这个看个人偏好选择

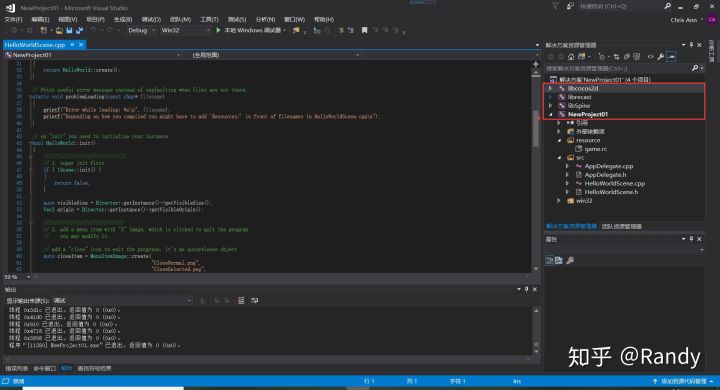
运行完成后我们设定的保存目录下就出现了拷贝好的程文件



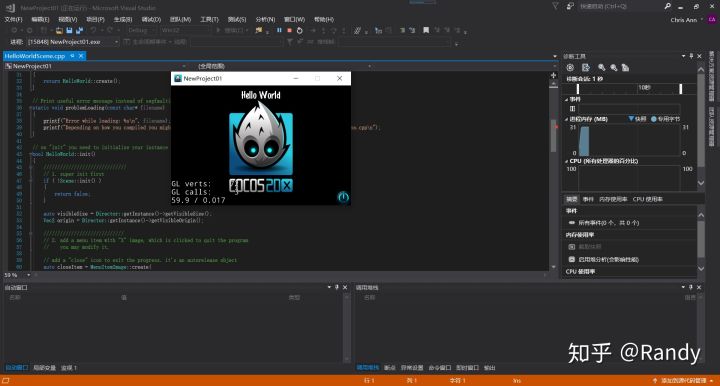
进入这个目录打开Visual Stdio工程文件



我们可以编译测试一下，如果编译不过，按照上面的方法，我们**将三个静态链接工程，和我们创建的工程逐一重定向**



编译成功如图，生成默认工程窗口

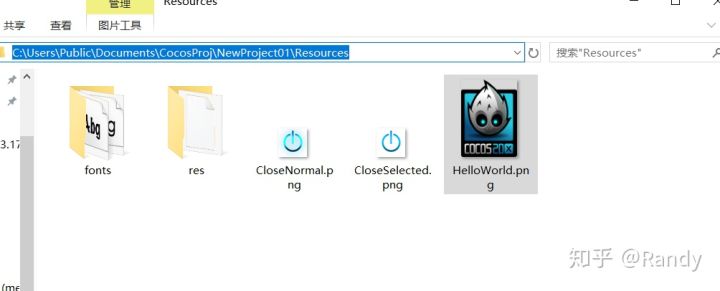


**---2.2 引入CocosStdio工程**

这个文件夹是Cocos2d项目的**资源文件目录**

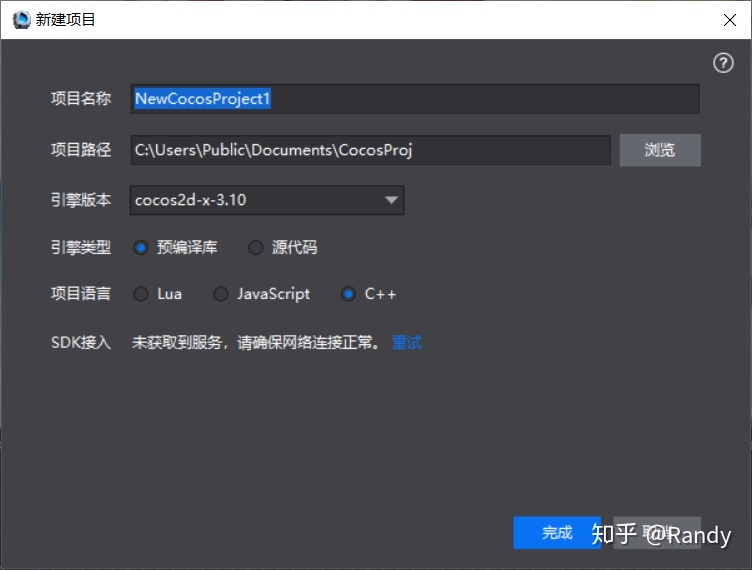
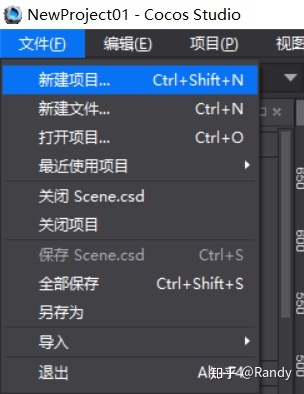
我们之后的**发布资源目录要设置为这里**

C:\Users\Public\Documents\CocosProj\NewProject01\Resources



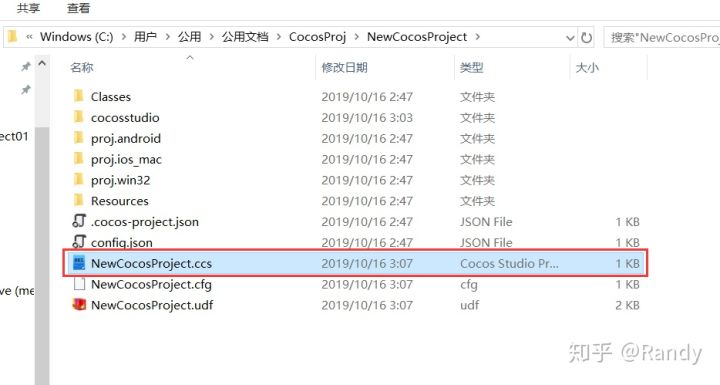
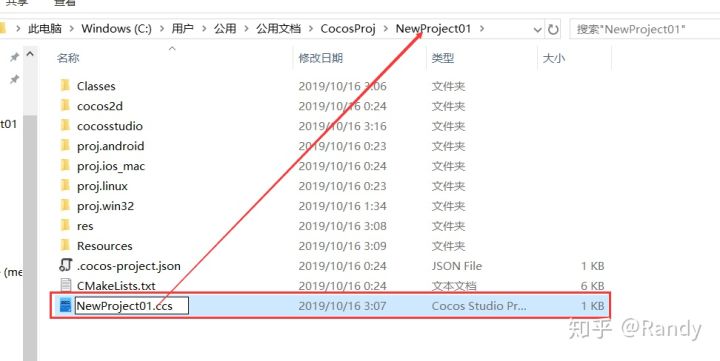
注意！！！由于**CocosStdio v3.10创建工程时与v3.10项目的创建被绑定**，因此我们需要使用“**借鸡下蛋法**”

通过CocosStdio我们新建出一个v3.10的工程

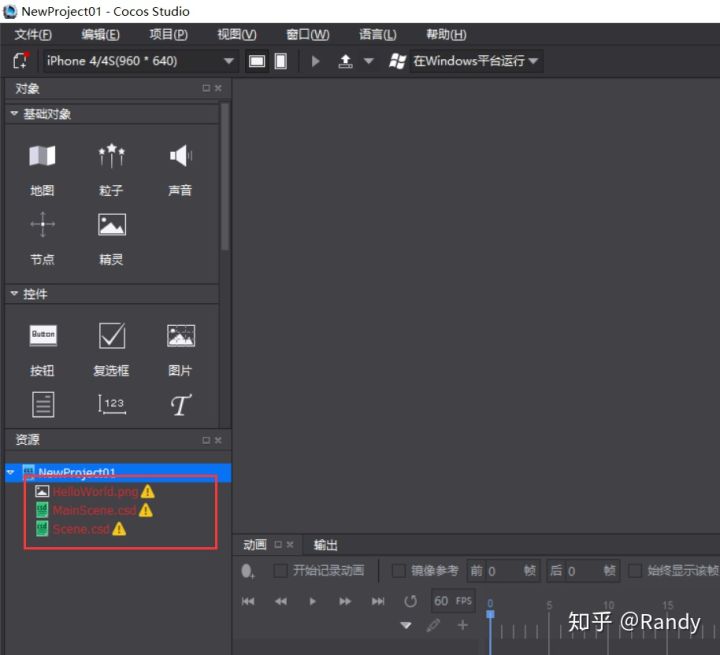


之后我们进入这个工程，将里面的.ccs CocosStdio工程拷贝到我们创建出的v3.17.2工程根目录下，并将其改名为与v3.17.2工程同名的CocosStdio工程

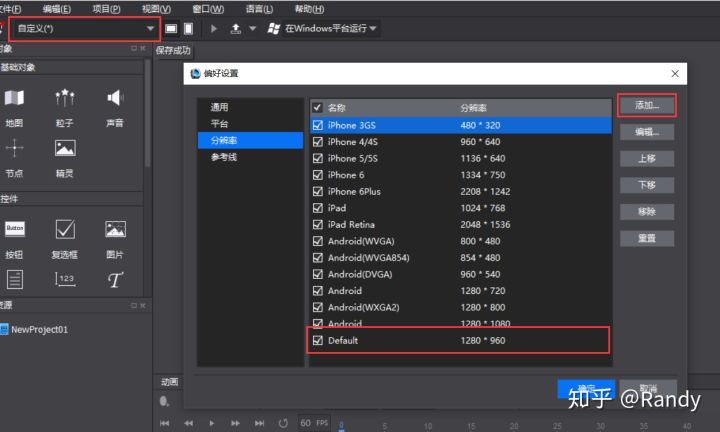
**以后我们新建了v3.17.2工程都需要来这里拷贝.ccs工程到v3.17.2工程的根目录下**

拷贝这个.css文件将其拷贝到v3.17.2的工程目录下并改名

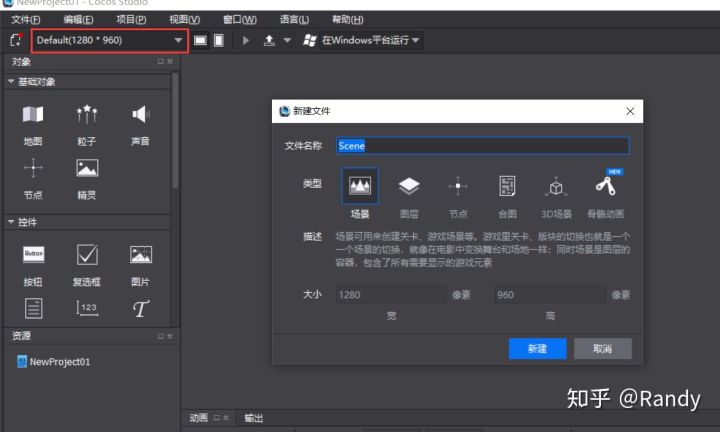
双击打开拷贝出的.ccs工程，原有的丢失文件删除即可



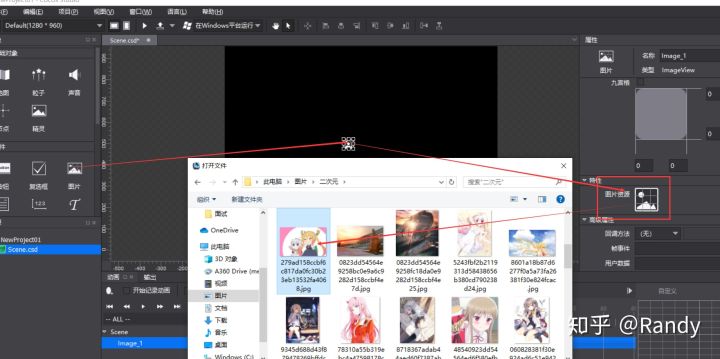
自定义添加1280\*960的分辨率



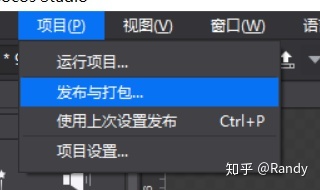
选择我们自定义的分辨率新建场景文件



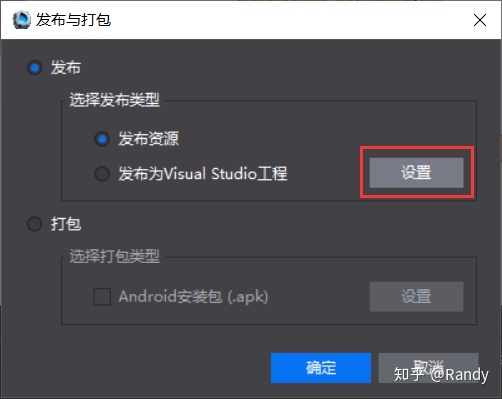
拖入一张图片，选定硬盘中的纹理资源，调整适配



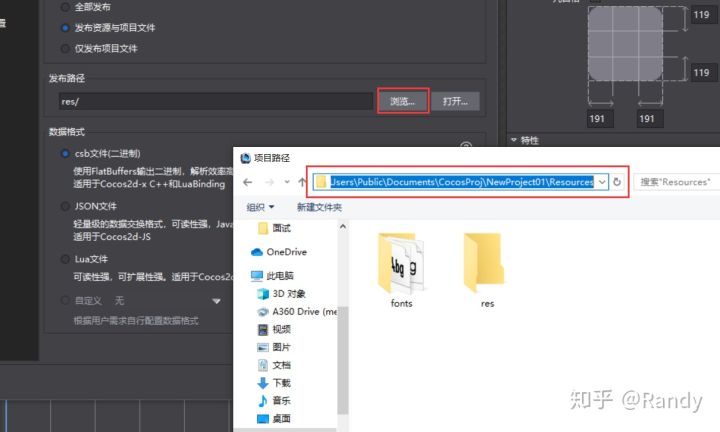
项目---发布与打包



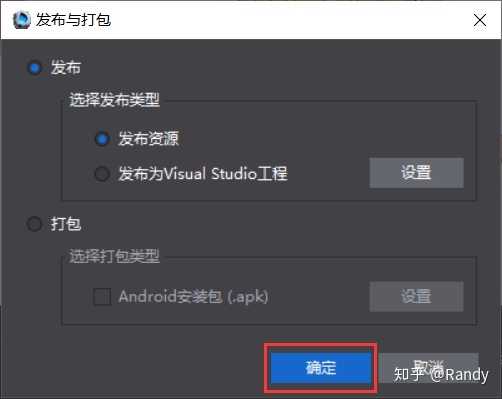
设置



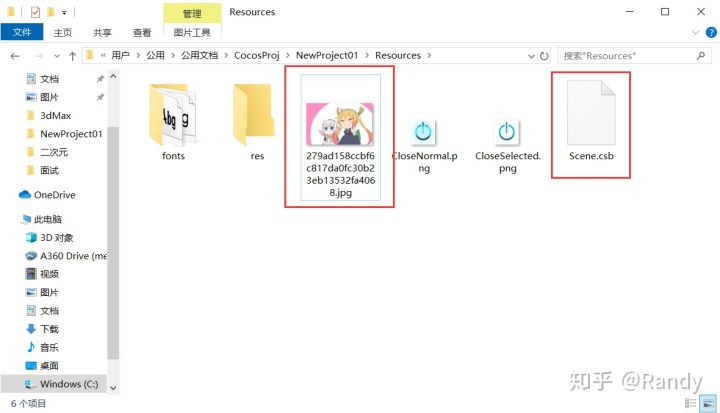
**将发布路径改为之前提到的资源目录**



随后我们**每次更改场景文件都需要进行一次资源的发布**



发布资源后我们可以发现Resource文件夹下多出了编码好的.csb场景文件和我们引用的图片资源

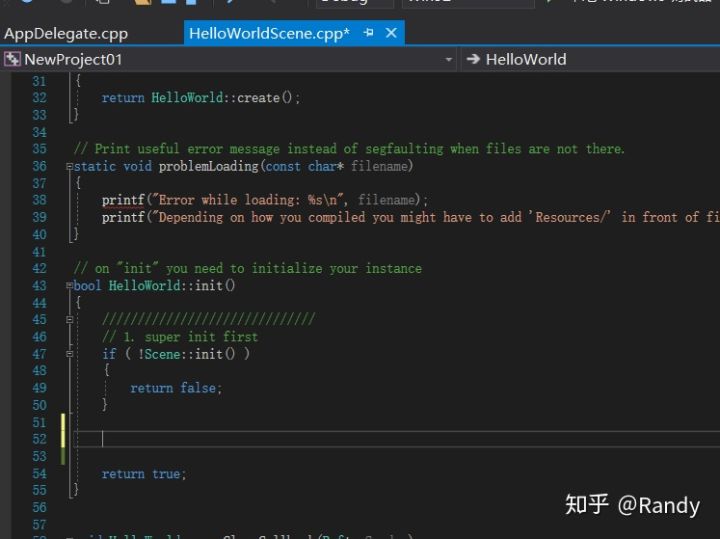


我们进入到Appdelegate.cpp中找到第一行设定分辨率的代码

static cocos2d::SizedesignResolutionSize =cocos2d::Size(480, 320);

将传参改为1280\*960（与我们创建的场景文件分辨率一致）

随后我们进入HelloWorld.cpp 删除Init中的原有代码



引入头文件和命名空间

**#include** "cocostudio/CocoStudio.h"

**#include** "ui/CocosGUI.h"

**using** **namespace** cocostudio;

**using** **namespace** cocostudio**::**timeline;

随后，在Init方法中添加以下代码

**bool** HelloWorld**::**init()

{

*//////////////////////////////*

*// 1. super init first*

**if** ( **!**Scene**::**init() )

{

**return** false;

}

**auto** myscence **=** CSLoader**::**getInstance()**->**createNode("Scene.csb");

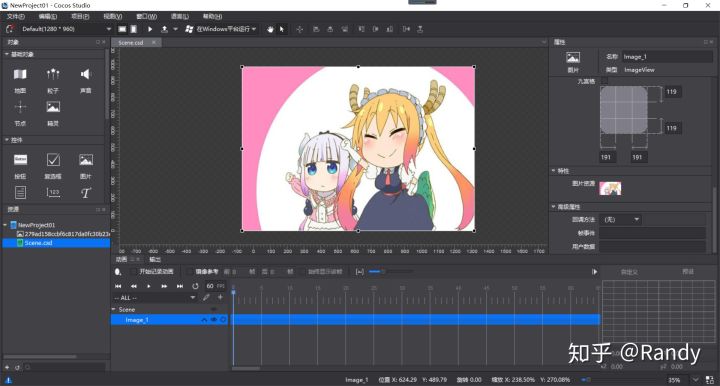
*//注意这里的"Scene.csb"要与我们发布的场景名称一致*

addChild(myscence);

**return** true;

}

编译运行，我们就**将CocosStdio v3.10制作的UI场景添加到了v3.17.2中**



事实上你完全可以删除创建出的v3.10工程，只留下那个.ccs文件，我们**每次创建完v3.17.2的工程之后，只要去拷贝那个.ccs文件即可**

/\*

\*/

编辑于 2019-10-24 09:12

[游戏开发](https://www.zhihu.com/topic/19553361" \t "_blank)

[Cocos2d-x](https://www.zhihu.com/topic/19625841" \t "_blank)

[cocos2d](https://www.zhihu.com/topic/19566188" \t "_blank)