声音设计师如何利用PD+ Heavy进行DSP插件的开发\_Part2

欢迎回来！

上一篇介绍了如何绘制图纸（Patch），这一篇我将带领大家建造“自动化车间（Heavy）”，生产产品并将产品部署到我们的火箭（游戏）当中去！

# 为什么要建造车间

原本Heavy的编译环境是有网页端的，也就是你可以在线上提交Patch，然后Heavy会返回编译报告并提供代码及插件的下载地址。

但由于去年Enzien Audio把Heavy开源了，导致现在必须要将编译器的代码克隆到本地使用。因此我才将接下来要做的事情形容为建造车间（否则就是取快递了！）。

在接着往下读之前请确保你已安装了某一版本的Wwise SDK，我安装的SDK版本为2019.1.1.6977。

## 一，下载安装python2及环境配置

首先为了能够运行Heavy，我们需要从下载python2.7版本。

**地址：**[**https://www.python.org/downloads/release/python-2716/**](https://www.python.org/downloads/release/python-2716/)

1，找到下图中的选项并点击鼠标左键。

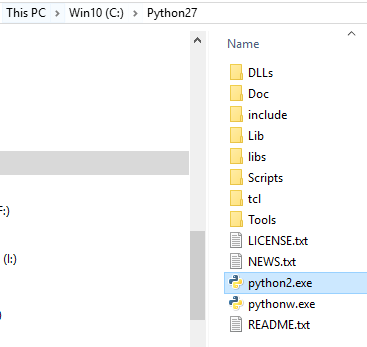


2，由于我是在PC上工作因此选择了Windows64位版本的安装包。

C://Users/Administrator/AppData/Local/YNote/data/weixinobU7VjqF_aM0hmX2X9pL8nss_ldU/b4f46a80bcaf41b6915f2a76a06a4edd/clipboard.png

3，我将Python2.7安装到了C盘根目录下（由于我的电脑上本来已经安装了python3.7，

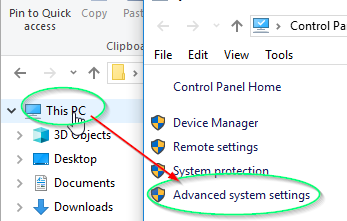
为了加以区分我需要将python2.7的可执行文件重命名为python2）。



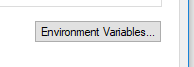
接下来要进行环境变量的设置。

## 二，环境变量的设置

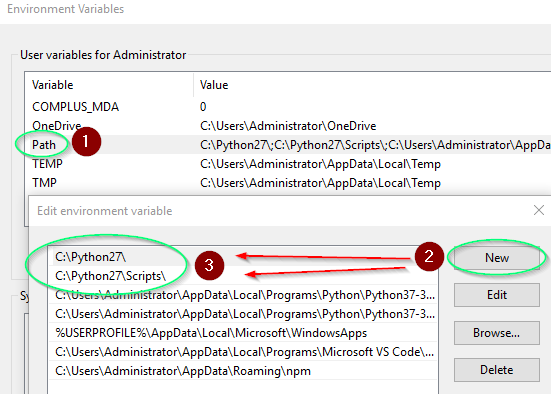
1，首先在This PC（我的电脑）处点击右键，选择Properties（属性）然后点击Advance System Settings（高级系统设置）。



2，点击Environment Variables（环境变量）。



3，然后双击下图中的Path，在弹出的窗口右侧点击New（新建），填入Python2.7的安装路径（也包含子文件夹Scripts）。

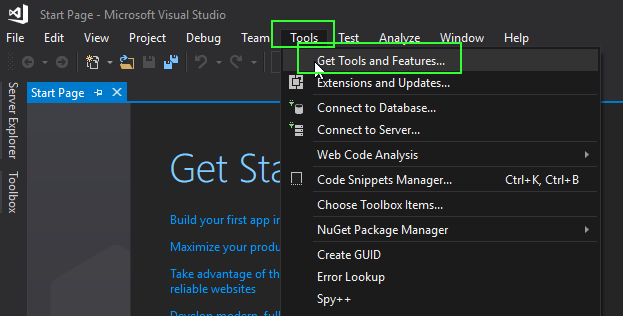


这样，我们在命令行工具中输入python2时便会运行python2.7版本了。

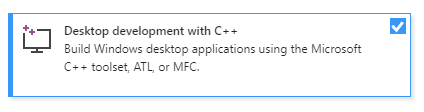
## 三，设置Visual Studio

为了使用Heavy生成的Wwise相关数据进行插件的构建，我们需要对Visual Studio进行配置。

1，打开visual studio然后在Tools栏（工具栏）中选择Get Tools and Features（获得工具及特性）。



2，选择安装Desktop development with C++（使用c++进行桌面开发）。



3，如果您使用的是VS2017版本，请确保Installation Details中勾选VC++ 2015.3 v14.00（默认是没勾上的）。

C://Users/Administrator/AppData/Local/YNote/data/weixinobU7VjqF_aM0hmX2X9pL8nss_ldU/19b23197b0144b65afb98c734928ba44/clipboard.png

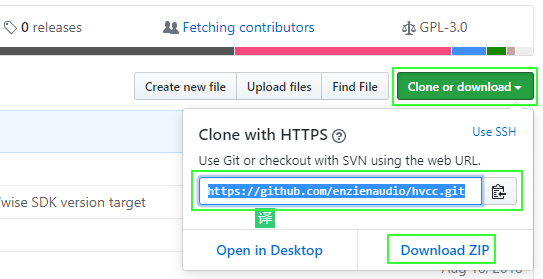
完成VS的设置，接下来该主角登场了。

## 四，克隆Heavy并设置

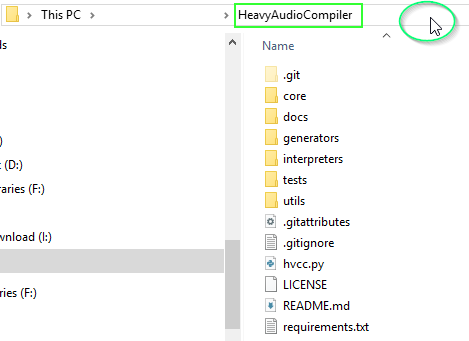
首先我们要从Github上把Heavy克隆到我们的电脑上。

地址：<https://github.com/enzienaudio/hvcc>

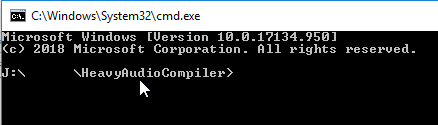
1，你可以复制Git地址到SourceTree（版本管理工具，<https://www.sourcetreeapp.com/>）里拽下来，也可以直接下载压缩包解压后使用。



2，然后进入Heavy所在的目录（如果您下载了压缩包，请先解压）。

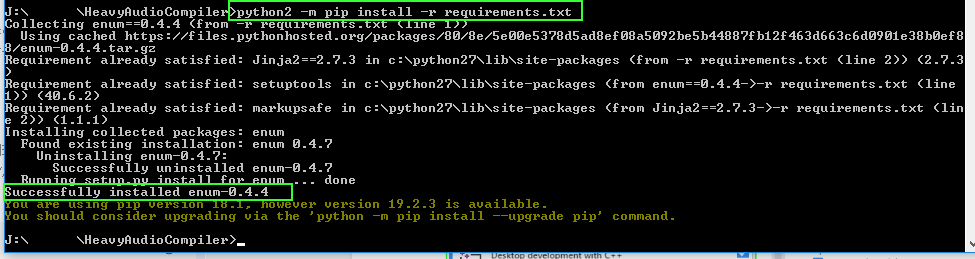


3，点击上图中绿色圆圈中的空白区域，输入CMD然后回车，这样便可以直接在Heavy所在目录处开启windows命令行工具。



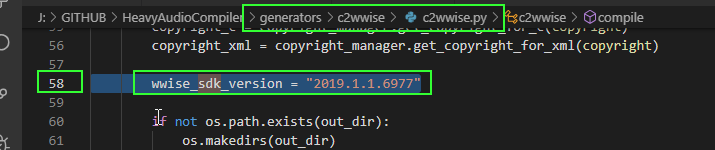
***就不用自己把地址敲一遍了。***

4，由于我们刚刚安装python2，还缺少运行heavy所需要的库，因此需要在命令行工具中输入python2 -m pip install -r requirements.txt来安装库文件。



5，接下来要让heavy代码中的Wwise SDK版本与你所安装的版本一致，否则会发生编译错误。

比如我机器上安装的是2019.1.1.6977版本的Wwise SDK，需要在c2wwise.py第58进行替换（路径见下图上方绿色方框内）。

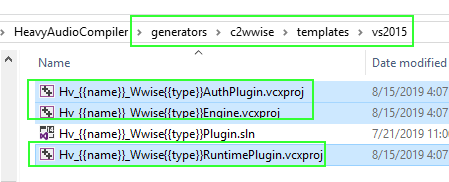


**Note：在heavy的文档中有提到这部分，地址**

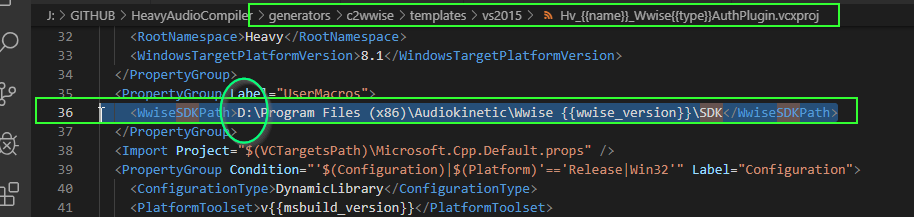
[**https://github.com/enzienaudio/hvcc/blob/master/generators/c2wwise/c2wwise.py#L58**](https://github.com/enzienaudio/hvcc/blob/master/generators/c2wwise/c2wwise.py#L58)

6，由于我的Wwise SDK被安装在了D盘，而在Heavy提供的VS Solution的模板中，默认SDK安装地址的盘符是C，所以需要都修改一下（瞧我把坑给你们填平了！路径如下图上方绿色方块内所示），

7，首先进入下图中地址，



8，打开每个vcxproj文件，将所有第36行的盘符C改为D。



如果你有耐心并且进行完上面的所有设置，祝贺你，你已经拥有自己的插件生产车间了！

## 五，运行Heavy流水线

接下来就是激动人心的时刻了，开机生产！

1，以前面所述的方式，在Heavy所在目录的地址栏输入CMD打开命令行工具，然后输入“python2 hvcc.py J:\HeavyAudioPlayground\bitcrusher.pd -n HvSampleDivider -o C:\Heavy\_Build\HvSampleDivider\ -g wwise”。

其中“**J:\HeavyAudioPlayground\bitcrusher.pd”**这段是我们之前创建的Patch所在的路径；

-n HvSampleDivider将把输出的插件命名为HvSampleDivider，否则插件会使用默认名称“Heavy”；

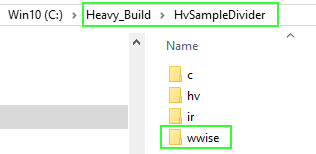
-o C:\Heavy\_Build\HvSampleDivider\ 是编译后的资源输出的路径（大家可以随意设置）；

-g wwise将指示Heavy生成构建Wwise插件需要的数据，Heavy还支持其他平台。

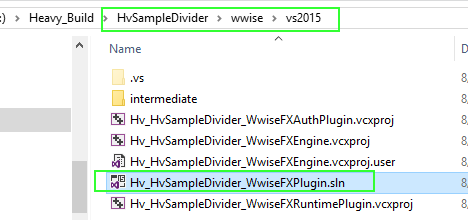
**Note：有关Heavy的具体使用方式请参考<https://github.com/enzienaudio/hvcc>中Usage一节。**

2，按下回车之后如没有出现报错，则Heavy已完成编译。

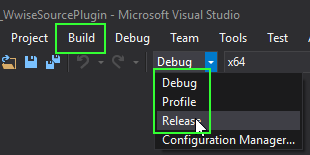
3，进入C:\Heavy\_Build\HvSampleDivider\后也就可以看到以下内容了。



4，然后我们进入wwise->vs2015文件夹，双击Hv\_HvSampleDivider\_WwiseFXPlugin.sln文件运行Visual Studio。



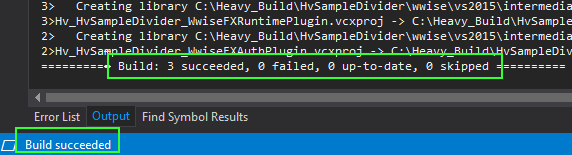
5，如下图所示，在下拉菜单中选择Release。



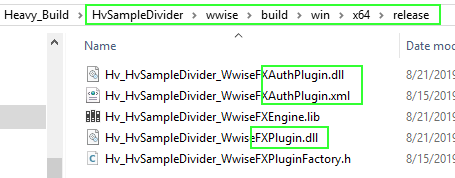
6，在Build栏中点击Build Solution开始建造。

C://Users/Administrator/AppData/Local/YNote/data/weixinobU7VjqF_aM0hmX2X9pL8nss_ldU/7bc94310dec04a14b3e98c81a7a0fc0b/clipboard.png

7，如果上面所有的设置都做对了，这里就能看到成功的报告。



8，然后进入C:\Heavy\_Build\HvSampleDivider\wwise\build\win\x64\release，就可以看到我们的劳动成果（插件）了。



# 部署

## 一，产品的安装

接下来，该将插件部署到创作工具及游戏工程当中去了。

C://Users/Administrator/AppData/Local/YNote/data/weixinobU7VjqF_aM0hmX2X9pL8nss_ldU/3f532adc162649429426db2eef6dc76c/clipboard.png

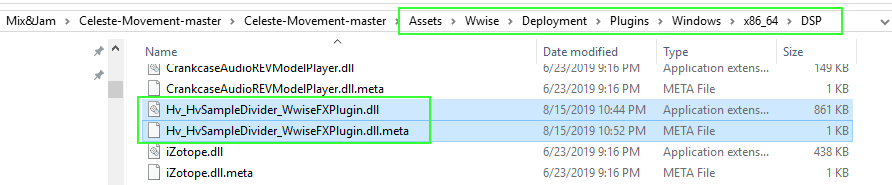
1，如果我们只在创作工具中使用插件，则要将上图中的dll及xml文件放到wwise的插件目录下，如下图所示。



**NOTE：文件名中包含的“Auth”字段代表文件应被部署到创作工具中。**

C://Users/Administrator/AppData/Local/YNote/data/weixinobU7VjqF_aM0hmX2X9pL8nss_ldU/f2cf2699eb464cbdb491e71ab0f24998/clipboard.png

2，为了让插件在引擎中运行并被打包到游戏中去，我们需要将上图中的文件放在游戏项目工程目录下的Assets->Wwise->Deployment->Plugin->Windows->x86\_64->DSP文件夹当中。

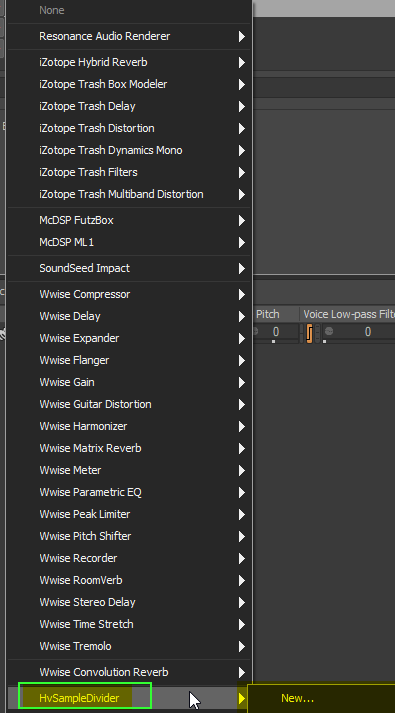


**NOTE：我使用了Mix and Jam（**[**https://www.youtube.com/channel/UCLyVUwlB\_Hahir\_VsKkGPIA**](https://www.youtube.com/channel/UCLyVUwlB_Hahir_VsKkGPIA)**）制作的项目作为展示工程，他的频道超级棒一定要去看！**

**另外，如果大家想了解插件在UE4中的部署方式，可以前往声音设计师Derrick Reyes频道<https://www.youtube.com/channel/UCkjoqOJDeT-jfCOJQPZ_F4w>**

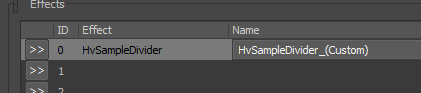
3，接下来打开Wwise。

由于我们之前在Patch中包含了adc~物件，所以生成的是效果器插件而不是源插件，因此我们应该可以在Audio Object或者Bus的Effect栏中找到它，而不是在Sound SFX->Contents Editor的Add Source菜单当中。

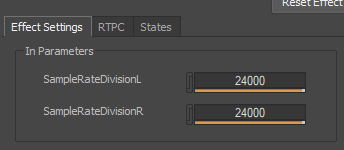


**如图，已经可以在Wwise中看到了！**

4，然后添加到Wwise某个Audio Object或Bus的效果器链当中。



5，双击打开就可以看到前面我们开放的参数都以默认值显示在插件界面上了。

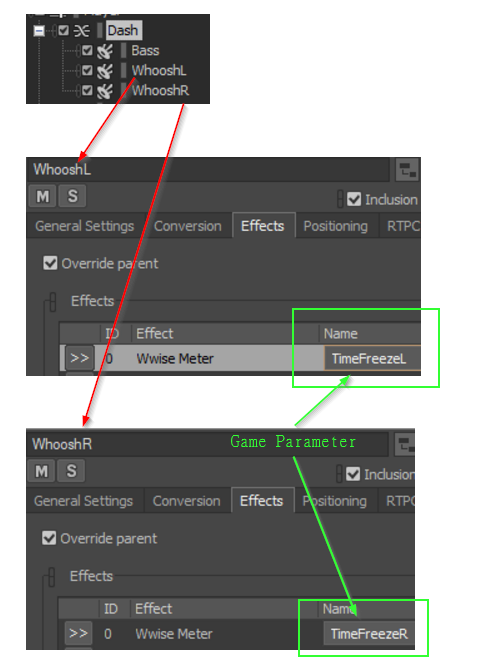


## 二，与游戏参数链接

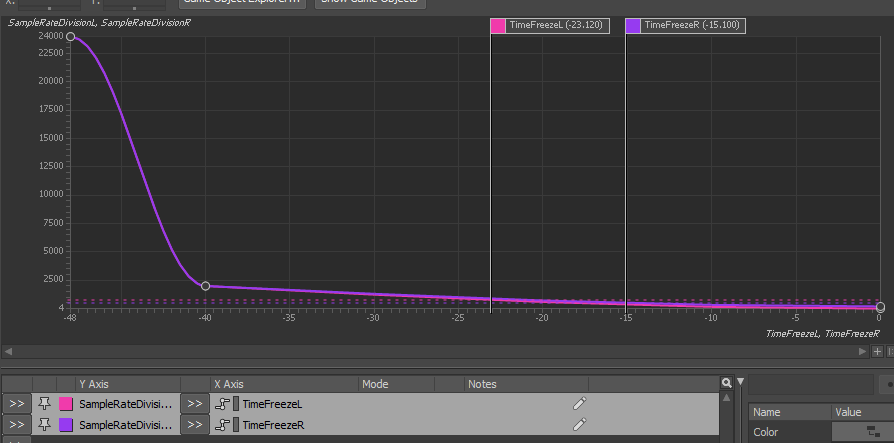
我想要利用这个插件实现的效果是：当游戏中角色冲刺时，利用冲刺声音量的提升来降低音乐及环境声的采样率。

为此，我们需要创建游戏参数来传递冲刺音量的变化。

1，我将冲刺声分成了包含左右两个声道信息的Sound SFX，并分别添加了Wwise Meter插件来捕捉音量包络，然后发给不同的两个游戏参数：



2，然后我们要为Music及AmbientBus上的HvSampleDivider的两个参数创建RTPC，并找到理想的曲线



3，接下来，就是打包、跑游戏、享受战果啦！

# 成果展示

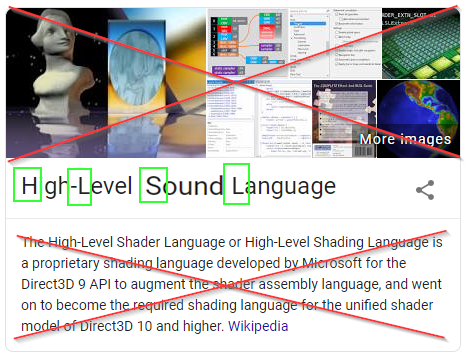
下面是HvSampleDivider在游戏中的实际表现。

**Note：Music是我写的……**

暂时拿掉

仔细听主角冲刺时音乐与环境声的变化，这都是Wwise原本无法做到的事情，现在是不是超级有成就感！

# 展望



感谢大家读到这里！

我有个疑问，既然有Max4Live，为什么不能有PD4Wwise呢？

如果可以将Heavy编译管线整合到wwise当中，使用PD的声音设计师就可以在Wwise中更方便地制作插件形式的音频资源了（就像Technical Artist制作Shader那样）！从此声音设计师也拥有了属于自己的HLSL！

让我们一起相信未来吧！

## 彩蛋

另外，在这个Demo中所有声音都是使用SynthOne作为声源实现的。

然而有一个疑问一直以来都困扰着我，那就是……既然有SynthOne，什么时候会有SynthTwo，SynthThree呢？！

等了这么久终于不用等了，我可以自己造了！

先来个FMOne！ XD



# 附录——Heavy不支持的PD物件列表

**也就是你“画图”的时候用了就会让生产线“卡壳”（error）的物件**

makefilename

value

list

cputime

realtime

expr

midiin

polytouchin

sysexin

midirealtimein

polytouchout

midiout

stripenote

oscparse

oscformat

fudiparse

fudiformat

array

savestate

netsend

netreceive

glist

textfile

text

openpanel

savepanel

bag

key

keyup

keyname

fft~

iff~

rfft~

rifft~

log~

framp~

vline~

bang~

readsf~

writesf~

print~

delread4~

pd

clone

block~

switch

struct

drawcurve

filledcurve

drawpolygon

filledpolygon

drawtext

drawsymbol

plot

drawnumber

pointer

get

set

element

getsize

setsize

append

scalar

sigmund~

bonk~

choice

hilbert~

complex-mod~

loop~

lrshift~

pd~

stdout

rev1~

rev2~

rev3~

bob~

scope~

namecanvas

template