**实验报告2-flappy bird**

1. **设计思路**

就是模仿网上的flappy bird写的游戏，但是功能太单一，于是给游戏设置了子弹和石头。小鸟在达到一定的关卡的时候，前方不仅会有水管，同时会有石头随机出现，小鸟碰到水管和石头都会死亡。所以小鸟要发射子弹防御。

给游戏加了个菜单，可以调节游戏难度，和挑选小鸟的种类。一共有六个关卡，随着关卡的提升，水管的速度和数量会增加。石头的速度和射出的方向也是随机的。

1. **设计过程**

刚开始规划的只有两个类

小鸟类

水管类

然后在各自的类里面实现了对应的功能，在小鸟类里面，主要函数有小鸟的图片切换，返回对应的 mask 图片，判断小鸟是否可见的图片，用于小鸟和水管的碰撞检测。

在水管类里面，有水管的生成，水管的绘制，判断水管是否可见，水管的绘制比较麻烦，就是设置了一个矩形的子表面，然后把水管绘制上去 ，因为水管的长度是随机的，所以用循环先在子表面把水管的body画好，然后在水管顶部贴上水管头，背景设置为透明的把整个子表面绘制在屏幕上。然后有返回水管对应的 mask 图片，用于判断水管和小鸟是否发生碰撞。判断小鸟和水管是否发生碰撞的函数在水管类里实现。

然后又加了两个类，石头类和子弹类

石头类的函数就是生成石头，返回石头对应的图片进行碰撞检测。

子弹类就是根据鸟的位置生成对应的子弹，进行碰撞检测。

然后在游戏循环里面，生成的石头对象和子弹对象都存储在group里，进行碰撞检测的时候就直接进行两个group之间的碰撞检测。如果碰撞就把子弹和石头一起删除。（就是刚开始我不知道可以用组写，然后自己用循环折腾了很久也没有折腾出来，然后用pygame模块里面的函数写出来的，感觉用对了函数，python效率很高，很多繁琐的操作都不用自己写了，可以直接使用模块，一行语句就搞定了）

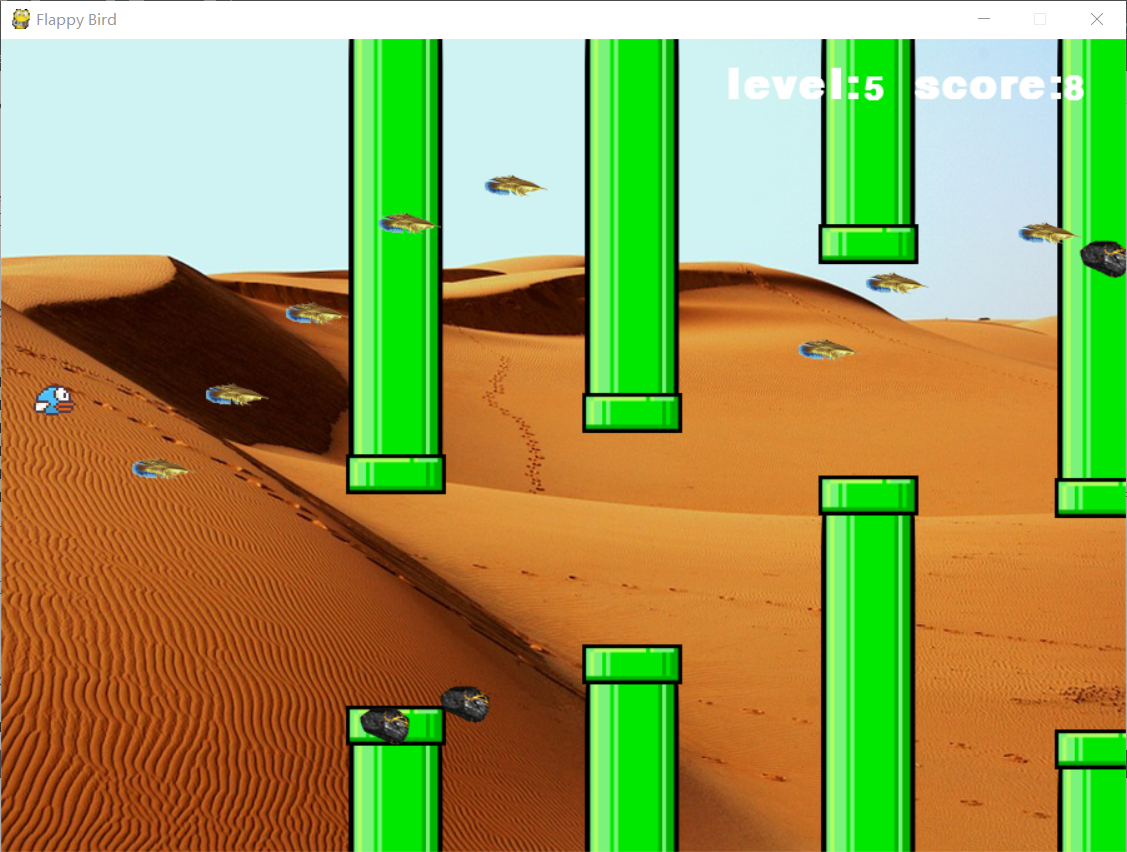
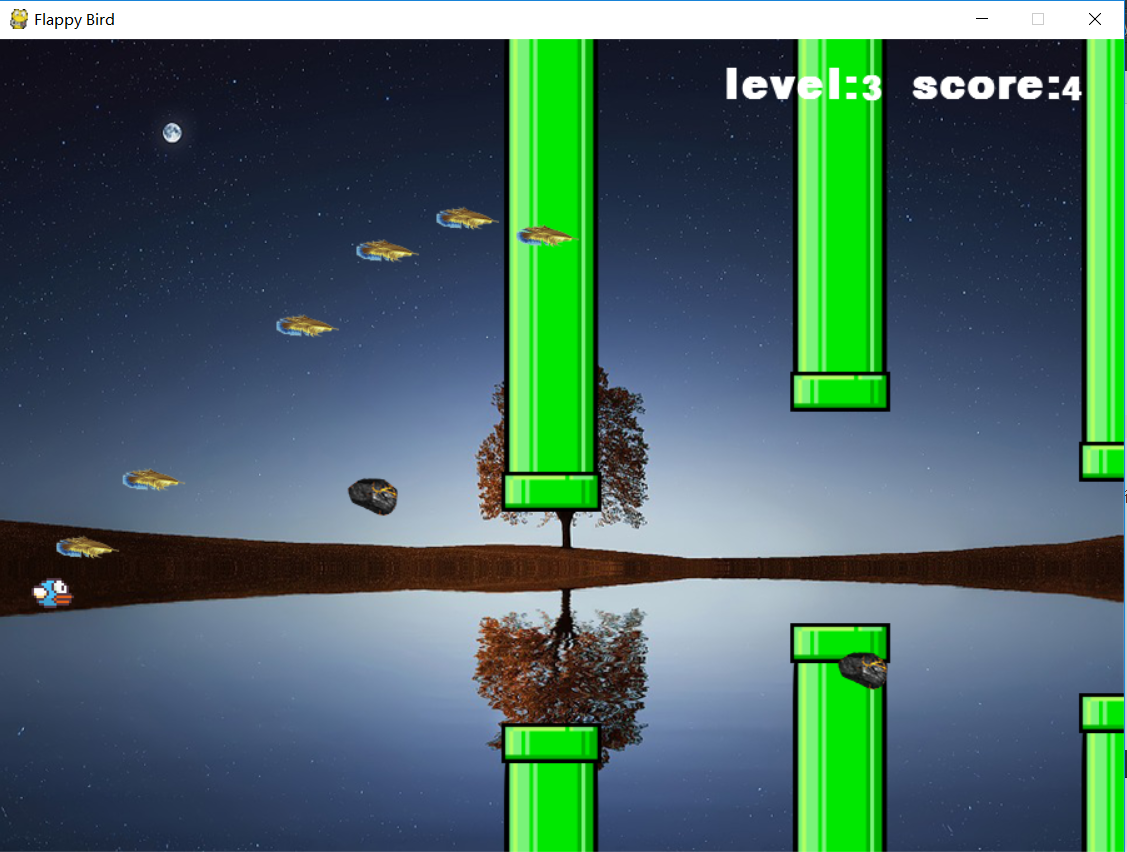
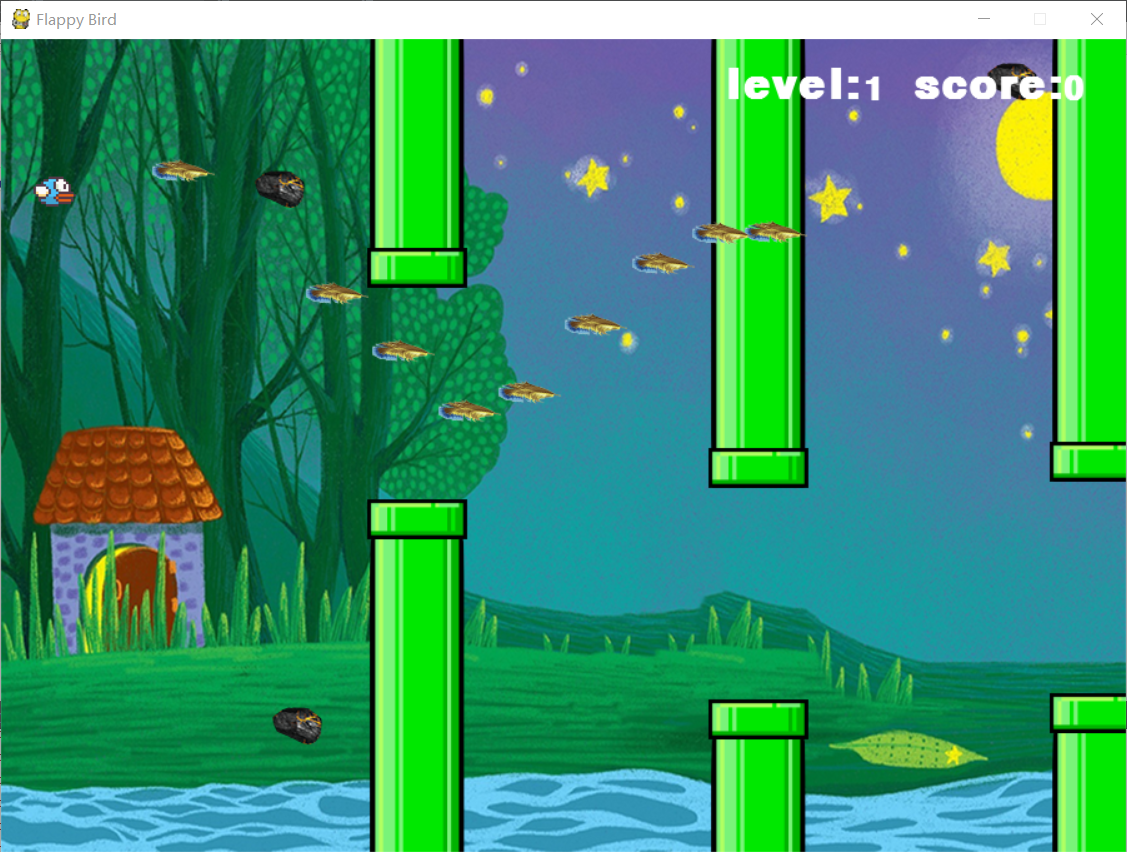
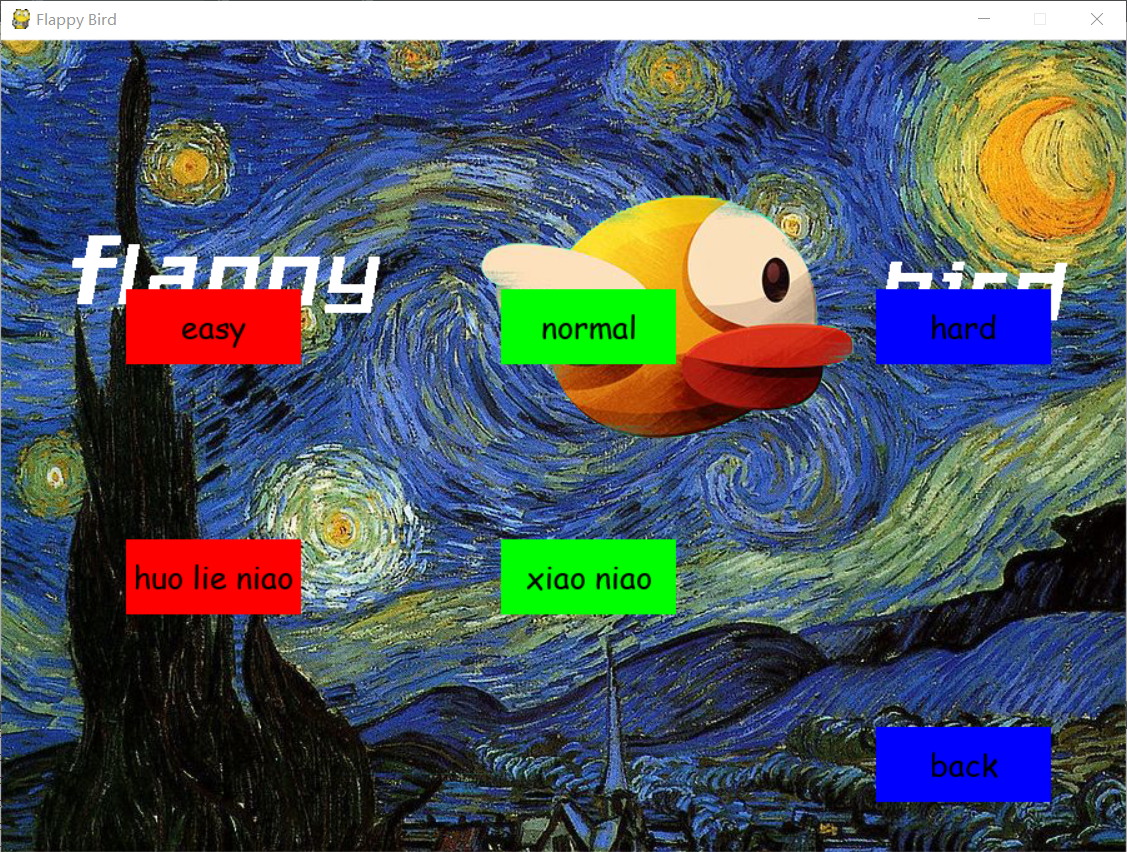
1. **代码分析**

就是使用面向对象的思想，根据自己的需要写出对应的类，然后补充类对应的函数，最后在main函数里面根据游戏的逻辑组织函数，实现自己想要的功能。

我觉得我写的代码的不足：

1. 函数的封装实现的不好，我刚开始使用了很多的全局变量，因为我想在调试的时候可以很方便的修改函数，但是后来我应该把它们去掉，而且我有很多时候还是会使用全局变量，这对函数封装很不利。
2. 我觉得我可以在类的继承这方面再改进，因为我的石头和子弹和水管有很多相似之处，比如都是以某速度朝某个方向移动，都要进行碰撞检测，都要返回mask图片，返回所在的矩形，返回图片进行绘图。他们可以抽象出共同的基类。然后我又可以减少一点代码的重复使用。应该能再减少一些行数
3. **效果展现**





1. **我学到了什么**

写第二个游戏的收获就是我终于解除了解释性语言，我觉得计算机有很多抽象的东西光靠定义很难理解，比如解释性语言和编译型语言。而实践就很容易弄懂了，当然我现在的对二者的体会并不深，了解的只是皮毛的皮毛。

python和c++都是面向对象型语言，我觉得对比二者的异同挺有意思的。

还有学会了一些python的语法，算是初级入门python了。

觉得自己的搜索查找能力比以前提高了很多，对比我刚接触github和刚接触python的时候就能感觉到。

最后我觉得我之所以能快速入门python，学了c和c++功不可没，就像老师说的，编程很多概念都是相同的，所以在学习python的语法的时候，会轻松很多，完全陌生的概念很少，很多都能从c和c++找到类似的概念。