

Python 大作业 report

5142029014 高超

一、大作业的主要介绍

Project 名称: Bouncing Ball

小游戏介绍:

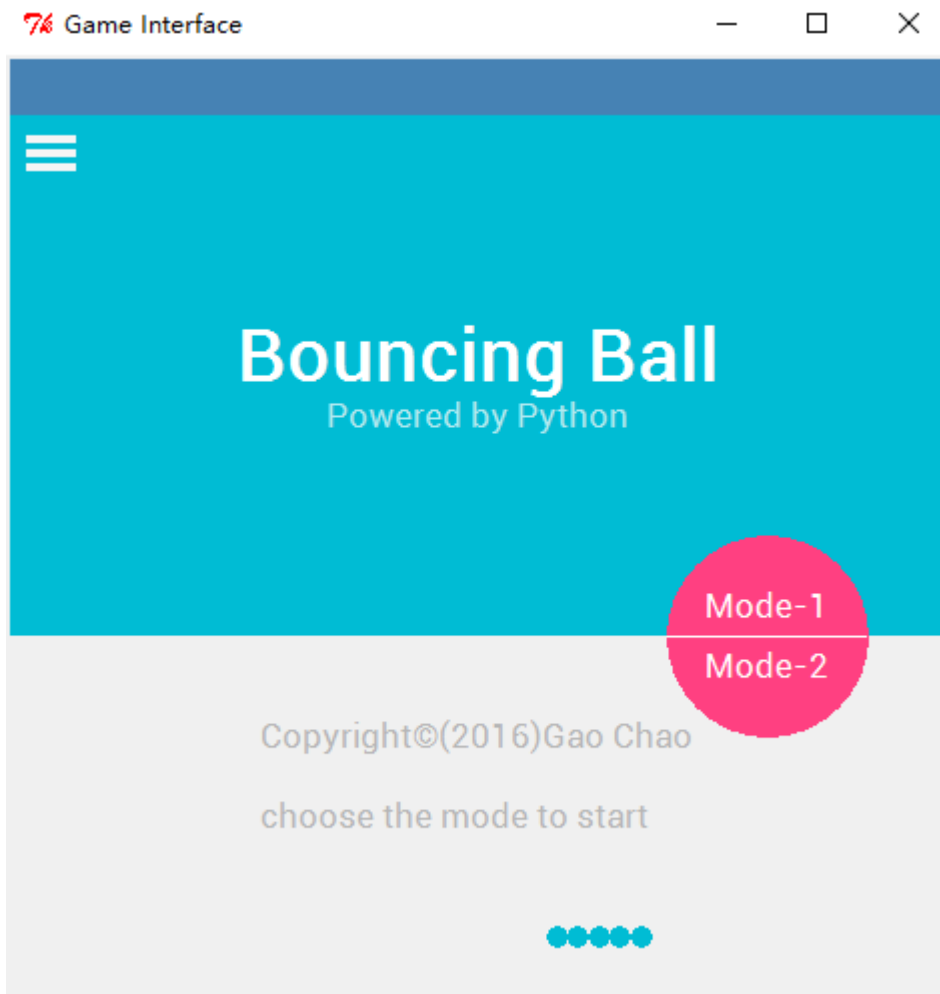
这是一个比较简单的小游戏，运行程序以后先出现一个欢迎界面，可以点击按钮（这里的按钮不是 python 控件）选择游戏模式。

Mode1 是单玩家模式，键盘上左右方向键控制挡板球遇到挡板会反弹碰到上面的砖块则将砖块消除，将全部砖块消除则获得胜利，如果球没有触及挡板落到底则失败。

Mode2 是双玩家模式，左右两个玩家分别使用 **w s** 和方向键中的上下键控制挡板上下移动，如果一方控制的挡板没有弹回小球，则对方获得胜利。

二、主要的程序实现

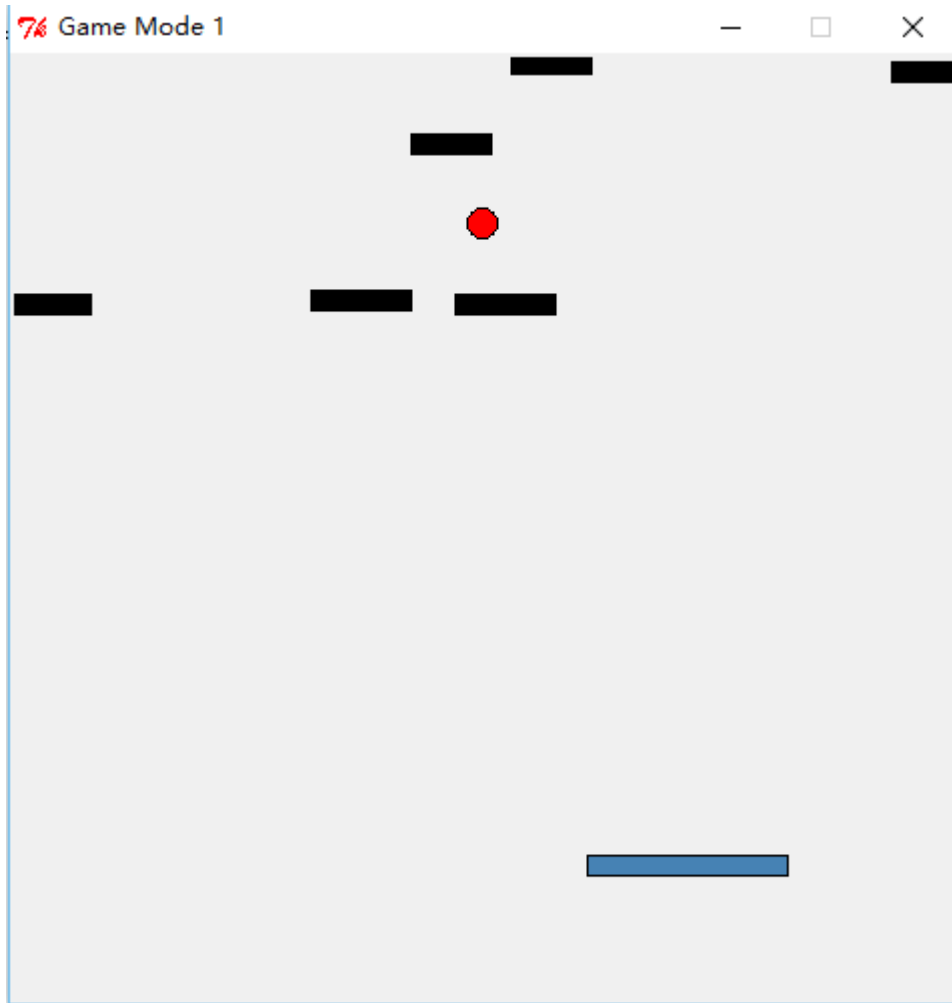
使用 **tkinter** 库画出一个界面，为了使整个界面更加好看，加入小球滚动的动画，由于涉及到屏幕的刷新，进入游戏界面后需要清除之前画布上的内容，但是清楚以后小球的动画依然存在，所以引入一个全局变量 **ballplay_enable** 来判断是否要展示底下小球的动画。



（欢迎界面）

绑定鼠标点击的事件，如果落在 **Mode-1** 的范围，进入 **Mode1**，如果落在 **Mode-2** 的范围，进入 **Mode2**，一次游戏结束后重新调用 `interface()`使出现欢迎界面。

下面是游戏部分单玩家模式的实现，写了球类和挡板类。



（单玩家模式）

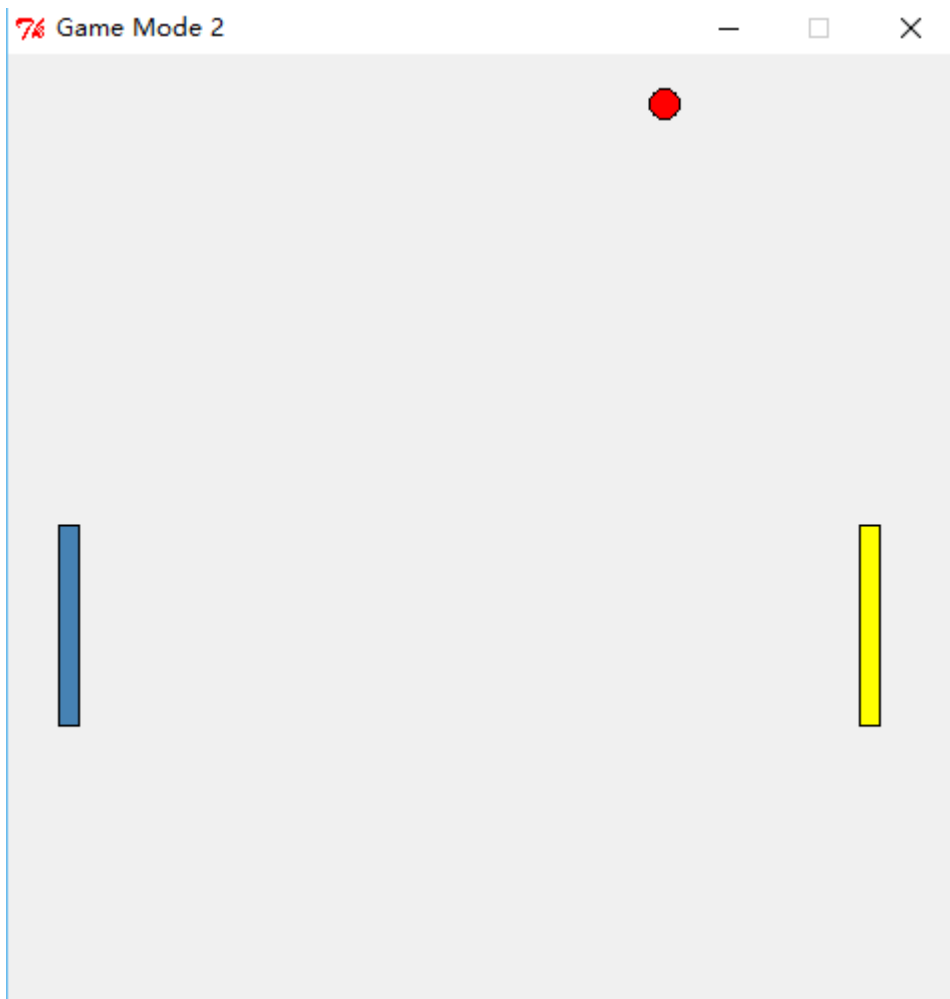
球类中的主要成员有砖块（位置有随机性），挡板，小球 x 方向的速度和 y 方向的速度。

判断小球与砖块，挡板碰撞主要通过比较横坐标在砖块或挡板中间，再比较纵坐标砖块或挡板的厚度中间，以此为基础写出 `hit_brick` 和 `hit_paddle` 方法，不过 `hit_brick` 需要将砖块消除，同时要记录已消除的砖块数量，以便判断是否获得胜利。

挡板类的实现比较简单，主要是左移和右移的实现，不过最初是按下键盘方向键一下挡板就一直会移动，后来做了一些改进，绑定 `keyRelease` 事件，使方向键释放挡板速度就变为 0。

回到主程序来，就是判断是否触及底部，如果没有触及底部且消掉所有砖块则获胜，如果触及底部则失败。

双人模式的实现事实上更加简单一些，因为少了砖块，其余的实现与单玩家模式基本相似，不再赘述。



（双玩家模式）

三、存在的问题

程序仍然存在一些问题:

1. 图形界面下感觉小球和挡板的速度不是很稳定,有时很流畅,有时很卡顿。
2. 退出程序以后,会显示一些错误(之前在上机练习中也有遇到),但是对于程序的运行没有影响。

四、实验总结

最终还是决定做个小游戏,鉴于贪吃蛇连连看有些泛滥,于是想做一个自己认为还有些新颖的游戏。

一开始的想法就是写一个欢迎界面和游戏界面,欢迎界面总体还是比较好看的,还加入了小球滚动的动画。

然后是游戏的界面,弹球的行为总体还是比较简单,于是再加入砖块让游戏更丰富一些。加入双人小游戏也是让这个小应用更丰富一些,在之前的基础上实

现起来就比较轻松了。

整体上游戏比较简单，主要的难点是控制小球的行为以及界面之间的切换。代码上存在不少缺点，可能写的比较冗余，整体上仍然是面向过程的写法，且使用了一些全局变量，没有很好的封装。

总而言之，这次的大作业让我对 **python** 有了更深的认识，也学到了许多。