中山大学 SUN YAT-SEN UNIVERSITY 现代操作系统期末项目实验报告

一、 项目分工

学号	名字	角色	班级	职责	贡献
15331350	杨耿聪	组长	晚上班	负责障碍和道具生成	25%
15331340	许琦	组员	晚上班	负责调度，控制障碍和小球下移	25%
15331339	许慕欣	组员	晚上班	负责菜单界面、分数排行界面	25%
15331353	杨霁晗	组员	晚上班	负责生成小球(金木水火土)，小球相关逻辑	25%

二、开发环境

Windows10，Linux

三、 项目阐述

名称：五行相生

简介：该小游戏包含五种基本的元素——金木水火土。游戏内玩家控制的小球会在初始化和吃到道具时被随机设定为一种基本元素，并通过设置小球的颜色为该基本元素的代表性颜色来体现。同时，小球运动过程中遇到的障碍是不断进行旋转的圆环，圆环由五个均等的段构成，每个段的颜色皆不同，代表各段的属性。

小球只能穿过与其当时属性相生的属性障碍，具体规则为：

[水]球通过[木]障碍，

[木]球通过[火]障碍，

[火]球通过[土]障碍，

[土]球通过[金]障碍，

[金]球通过[水]障碍。

小球在开始运动后会自由掉落，玩家通过敲击键盘可以让小球弹跳，通过不断敲击，可以让小球累积弹跳高度。

小球在运动的过程中一旦碰到通不过的属性障碍就会被判定为游戏失败，在此之前，会通过累计玩家控制小球成功上升的高度来计算分数。

功能：

1. 本地存储——会在本地存储本地的最高纪录。
2. 粒子系统——在分数记录页面增加了动态效果，丰富画面。

3. 网络访问——可以为各个用户在云端保存各自的历史最高分和提供分数排行以激励用户努力。

4. 事件处理——当小球与障碍或者道具发生碰撞时，需要处理碰撞事件。

5. 基本精灵的创建等

6. 调度器——每次小球的移动，障碍的转动，及碰撞的检测，场景的变化，均通过调度器实现。

7. 数据结构——障碍其实是一个数组，利用了数据结构

亮点：1.融入了中国传统的五行元素，具有丰富的传统文化色彩。

2.采用云服务器运行后端程序，24 小时均可访问。连接服务器主要是为了让用户登录并上传自己的分数和查看分数榜；即使没有联网，游戏依然可以正常运行，只是无法上传和查看分数。

3.操作简单，但可玩性强。

四、 项目展示

首先附上我们的五行相生图，根据此图来判断能否穿过障碍

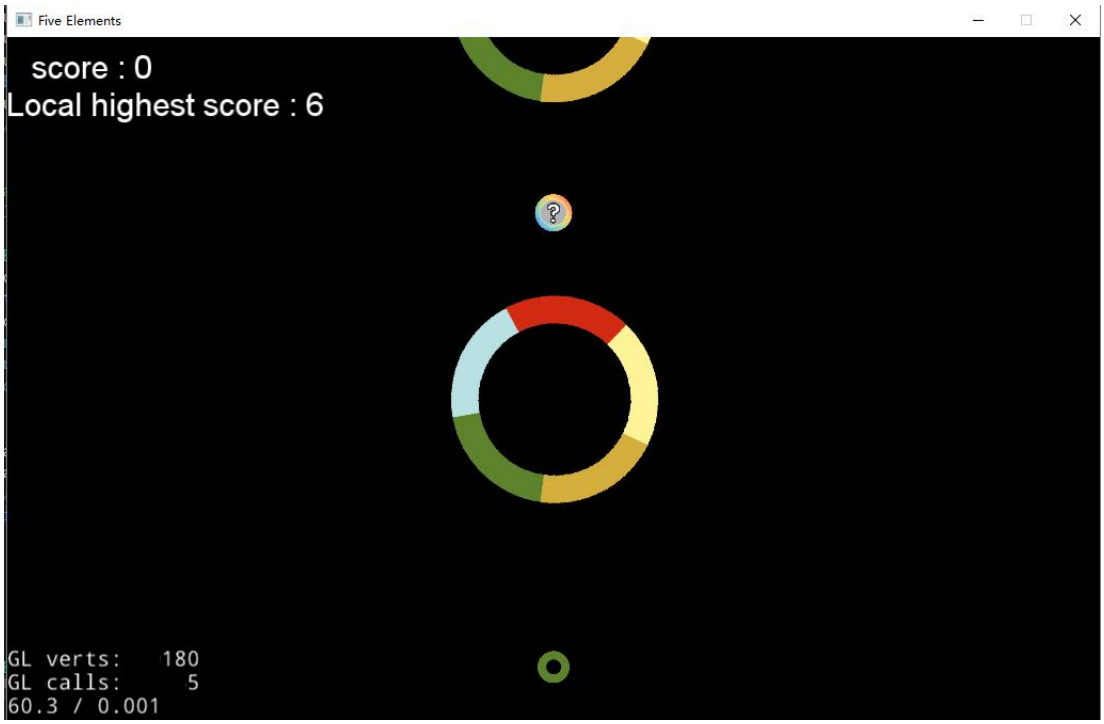
（比如水生木，故此[水]球通过[木]障碍）

1. 开始界面（包括用户登录和规则介绍）

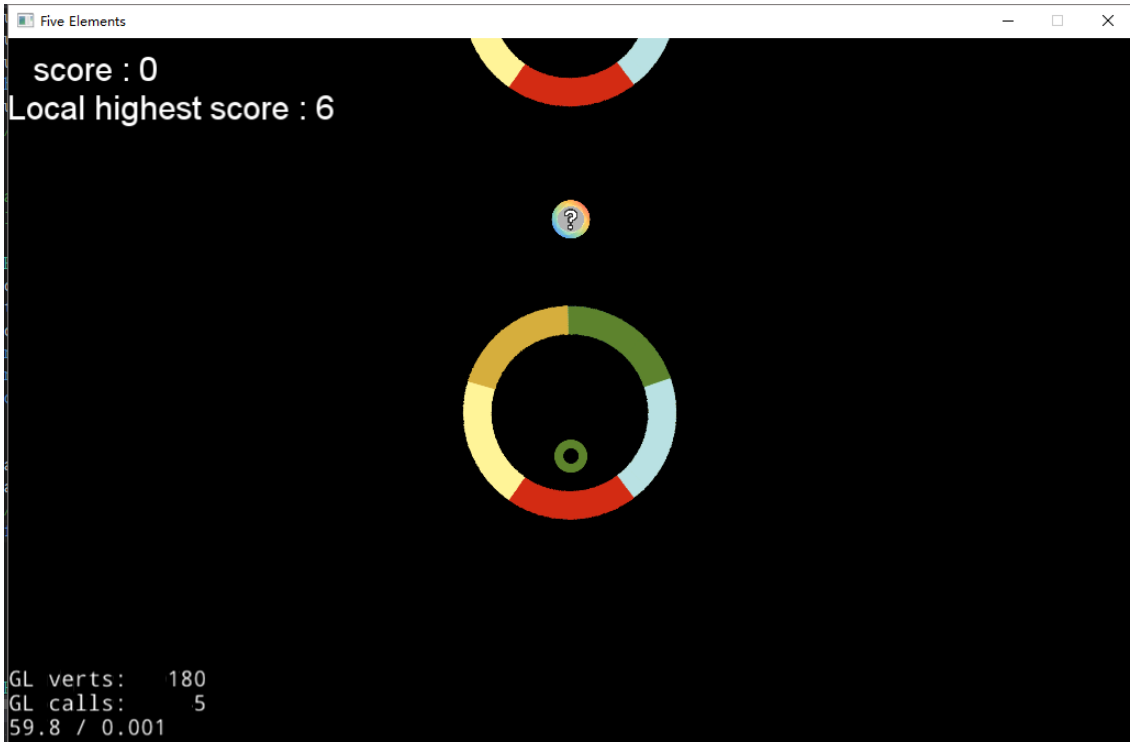
2. 输入一个新的用户名：test，来进行测试，并点击 start，进入游戏

3. 进入主游戏界面，出现小球的属性，与障碍旋转的方向皆是随机的，这次出现的是一个木球，只能通过障碍红色的区域，也就是表示着火的那部分。可以看到左上角两个分别是当前的分数，与本机记录的最高分。

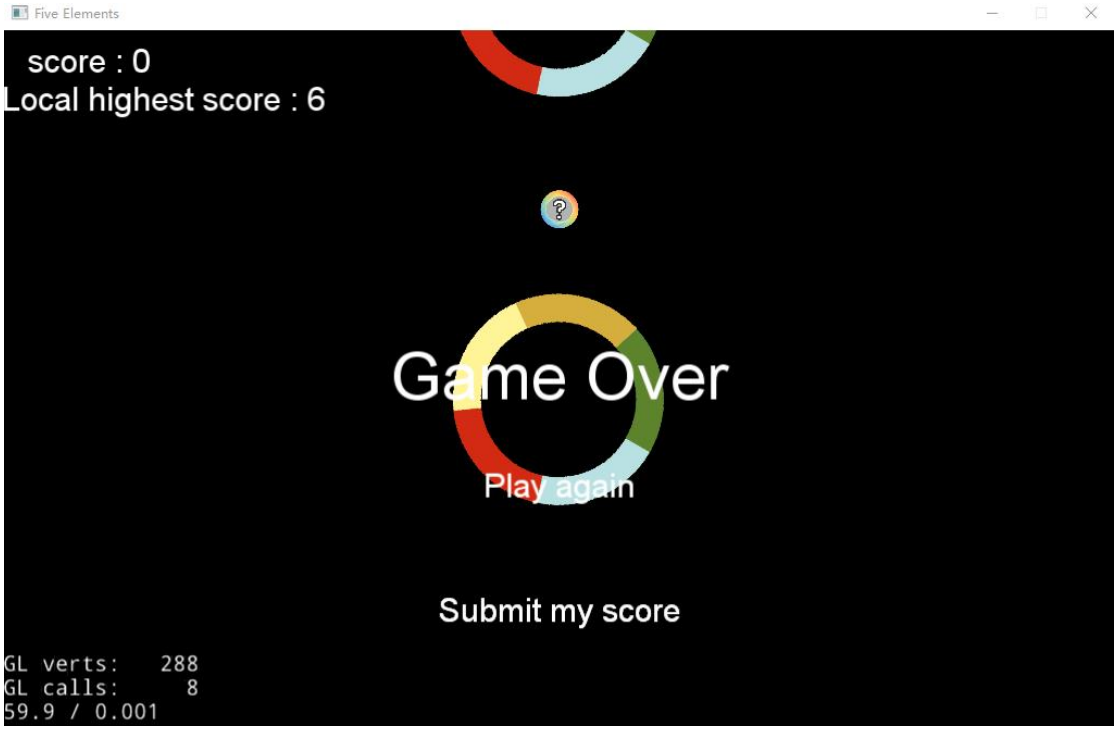
   按下空格键，小球即会开始向上跳动



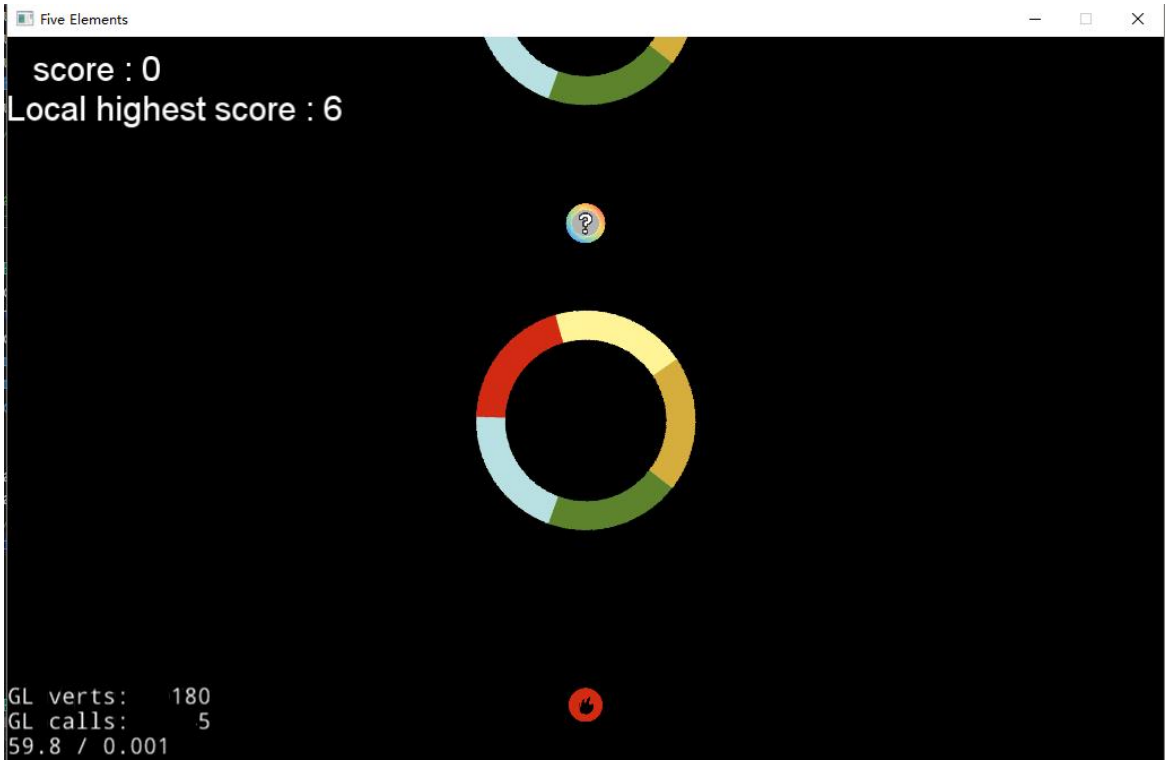
4. 测试成功通过



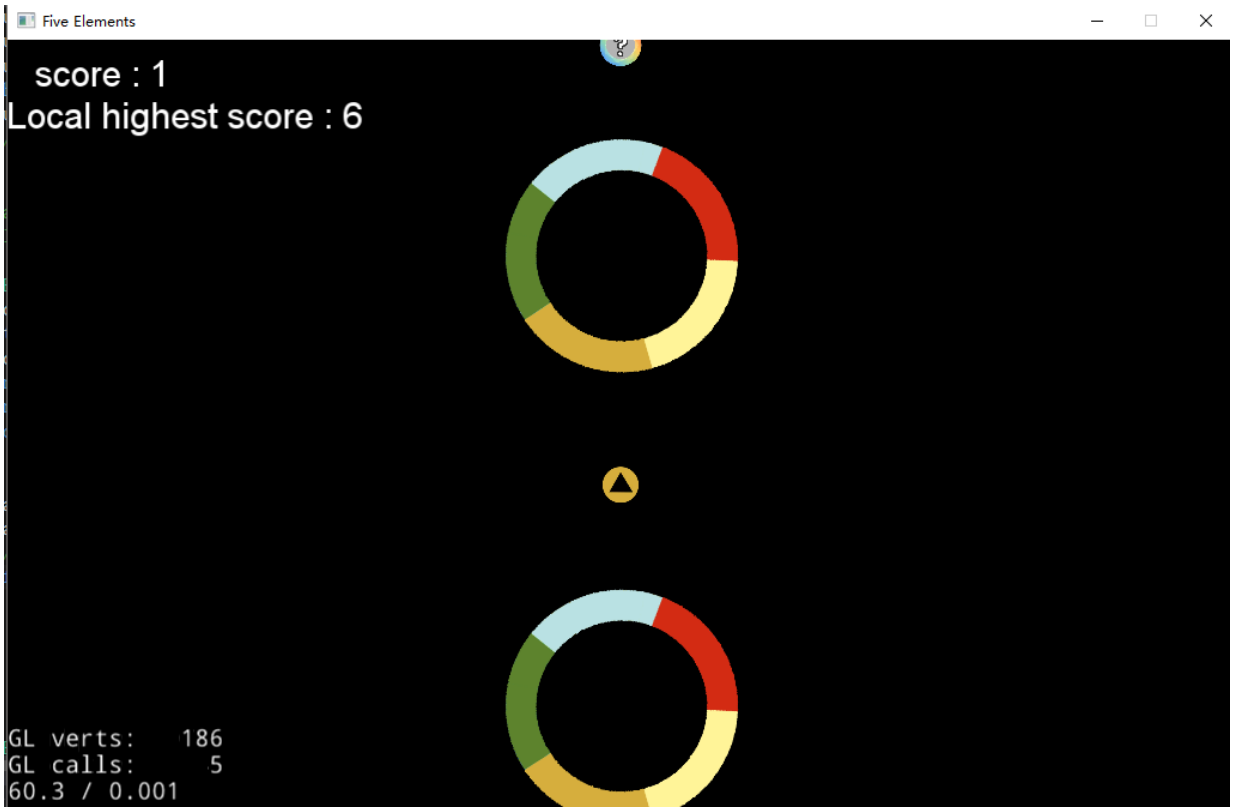
5. 测试碰到其他的属性



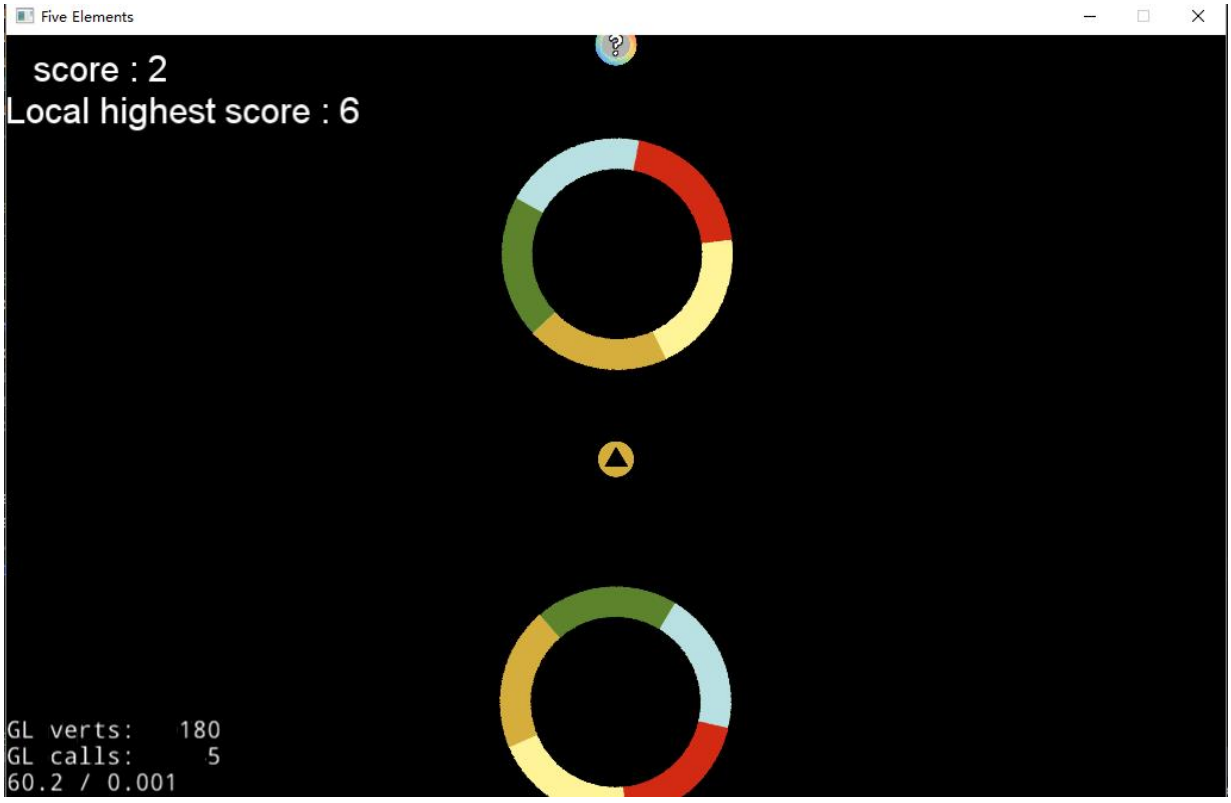
6. 点击 Play again，重玩游戏，这次初始是一个火球，只能通过淡黄色的土那部分属性的障碍



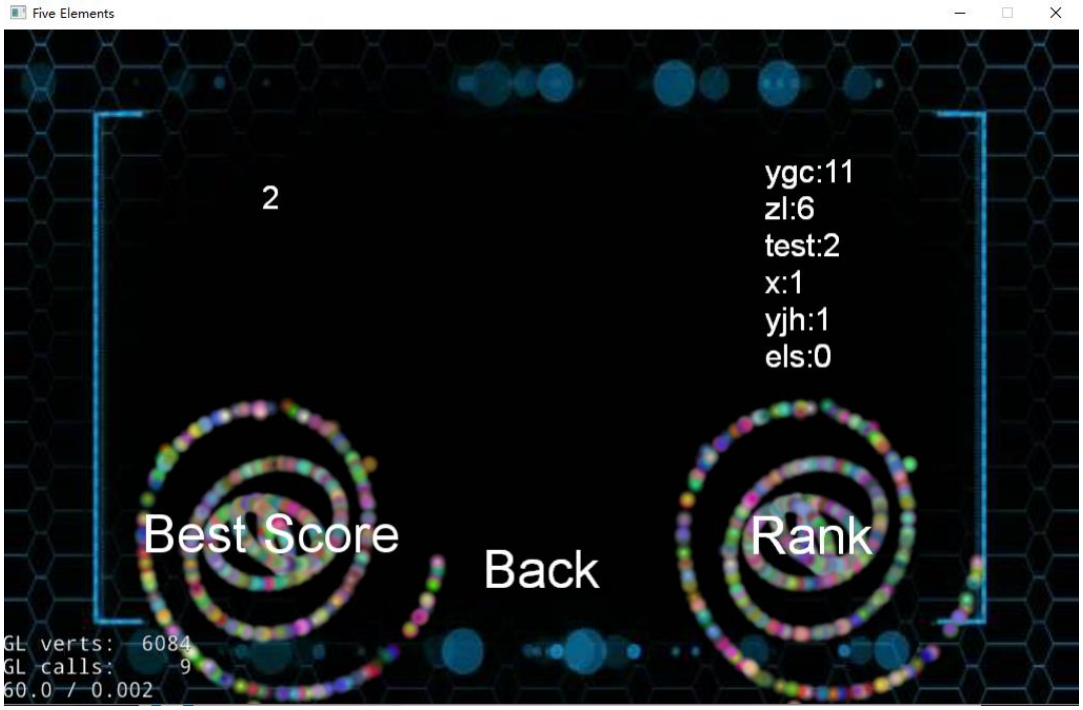
7. 当穿过障碍，碰到带有问号的彩色道具时，小球会随机变为另外一个属性（不会重复当前属性），并且场景会跟随这小球的移动而平滑移动，保证小球始终在屏幕中央。另外，当小球碰到道具时，左上角的 score 会增加，下一个障碍旋转的角速度会增加，而且旋转的方向会从顺时针和逆时针重新选择。



8. 虽然无法从截图中表现出来，但此时通过两个障碍后，旋转的角速度有明显增加，而且旋转的方向也发生了变化



9. 在 Game Over 之后点击 Submit my score，跳转到提交分数页面，可以通过 Rank 来查看所有玩过此游戏的人的排名和得分



五、 项目难点及解决方案

许慕欣:

1. 在本项目中遇到的难点是菜单界面的设计和相关素材的寻找。在找素材的过程中，或许是由于经验不足的原因很多时候会很难找到某一确定类型的素材，并且找到的素材通常会有大量水印。这就必须要对素材进行一些后期处理，如大小的剪裁等。同时，由于在布局时用了绝对像素导致在不同的窗口大小会出现布局不协调的情况，后来通过调整大小为相对于窗口大小的相对大小获得一定程度的改善。还有，在刚开始时没有对网络访问进行异常处理，导致在无法连接服务器时程序会报错，后来通过增加异常处理，使程序在本地也可以运行。

许琦:

2. 在调度器上，主要的难点便是碰撞的检测。因为在游戏中需要穿过圆环并且检测颜色，所以并不能利用库里的函数判断两个物体是否相交，因为圆环会被直接当做一个圆来处理，就算是在圆环的内部（空心的地方）也会被判定为相交的。所以只能自己写检测碰撞的方法，这个方法需要从圆内和圆外来考虑，利用小球到圆心的距离与它们俩半径之间的关系协助判断小球是否与圆环相交。
3. 在做地图的平滑移动时，利用了相对速度的原理，使得静止的小球仿佛依然在上升，其他物体在下降。设置了一个临界点，即屏幕的一半，当小球到达临界点后，小球的速度变为 0 且重力加速度为 0，其他所有障碍获得小球当时相反的速度，并且赋予我们模拟的相反的重力加速度，模拟两者之间的相对移动，完成地图的平滑移动。

杨霁晗:

4. 图片的 boundedSize 问题，因为对于处理素材没有什么经验，而且之前素材都是 TA 给的，一开始因为图片的像素不太对，周围透明的边界比本身的范围大太多了，然后判断碰撞会出现问题，要重新调像素，使其边界贴合小球，并且通过 setScale 来进行缩放，然后调整到合适的大小。
5. 不通过物理事件来仿平滑移动。因为一开始使用物理世界遇到很麻烦的问题，就改成不使用物理世界。然后是通过模仿 flappybird 的实现，通过设置全局的重力、然后每次点击就设置速度，来使其模仿平滑的移动。
6. 关于 removeAndCleanup 函数的问题。在 GameOver 函数触发之后，我调用了停止调度器，并且讲小球 removeAndCleanup，但是可能因为停止并非立刻停止，所以此时会触发检测小球位置的中断，所以后来要增加一个 flag 变量，来判断是否已经触发过 game over，如果已经触发了，就不再进行小球位置的检测。

杨耿聪：

7. 在实现障碍的功能时，考虑到需要检测小球与不断旋转的障碍的哪一部分发生了碰撞，因为当时仍旧执着于物理世界，于是天真地想到可以用物理世界的关节将多段圆弧拼接成一个整体。实际动手后才发现这样做极其复杂，或许确实有办法让这些圆弧拼到一起，但鉴于物理世界的深邃，可想而知，结果看起来一定非常别扭。所幸在 TA 的提醒下，我才意识到我们的游戏完全可以抛弃物理世界，从而以更巧妙的方式实现。大概思路就是，将圆环作为一个整体，根据其与小球的相对位置来判断是否发生碰撞，然后再通过其旋转角度来判断发生碰撞的部分。

六、 项目总结

许慕欣:

在本次中我负责的部分比较杂而琐碎，但实现起来仍然发现有很多需要值得注意的问题和需要去学习的方面。例如简单常用的 PS 技巧，如果有掌握的话可以在很多时候节省大量时间。同时发现小游戏的开发没有想象中那么难，关键是有趣的点子加上正确合适的概念抽象将游戏拆分各个部分，再加上合理的团队分工，小游戏的开发可以很有效率。

杨霁晗:

我负责的主要是小球部分的生成和逻辑，还有就是刚开始生成 project 的一些基本逻辑。这部分的难点不多，都是易于解决的。一开始是使用物理引擎，但是因为在实现方面有诸多意料之外的奇怪 bug，不得已又转为类似于 TA 最早给过的 flappybird 的那种通过调度器来不断设置位置的方式实现移动。我觉得这次比较大的收获不在于 project 的本身实现，而在于通过 github 进行项目的管

理，一开始虽然有些不熟练，遇到一些问题，但是熟练之后，这样子进行版本控制，非常的方便，比之期中项目要互传文件，而且大家之间都不知道谁修改了哪里，易于管理的多。

许琦：

在处理小球与圆环障碍之间的距离是，发现调用系统函数的 contentsize 返还的是图片的实际大小，如果对物体进行了缩放，并不会影响 contentsize，使得计算小球和圆环的半径大小时需要乘以一个放缩的因子，才能接近于实际的大小，由于存在误差，需要自己加减一个比较小的数来填补误差，只能不断的尝试，不断的调整，并没有发现其他很好的方法来解决问题。另外在实现无尽模式时，需要不断的生成新的道具与障碍，并且需要定期的删除已经 pass 掉了障碍，保证游戏的流畅度。另外，我觉得一个游戏的难度是很不好把控的，很多东西并不能只靠理论去想象，需要亲身的去测试，再加以调整，毕竟实践才能出真知。一个好的游戏，在难度上的循序渐进是很重要的，这样才能吸引玩家去挑战

杨耿聪：

1. 又是一次团队项目，在多次经历了代码版本难以统一的痛苦后，这一次我们毅然决定用 git 及 github 来管理我们的项目。诚然，我们起初对该工具的使用并不熟悉，只能现学现用，其间也浪费了不少时间在踩坑上，但我相信这是值得的。它让我认识到，过去那种复制粘贴的代码共享方式多少有些“用战术上的勤奋掩盖战略上的懒惰”的意味。我们容易被惯性思维蒙蔽，很勤快地按照原始的方法行事，却疏于反思这种方法的低效率性。大概工程化的目的便是要克服人性的这类缺陷吧。

2. 另外，我发现，一个团队的效率取决于团队全员同时“在线”的频率。坦然地说，这一次的项目代码量很小，即使一个人做也可以很快写完，而我们这个四人团队却在 ddl 前不久才完成项目。虽然可以以期末考试为借口，但事实上我们完全可能在复习周前就把项目大致做好了。低效的根源就是团队很少在同一段时间内集中起来讨论问题，时常出现有人提出了一个问题而很久以后才有稀少回复的情况，但是彼此的任务关联度又很高，不将问题讨论清楚就无法各得其所地继续。