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
14970011	宋思亭	组长	早上班	负责实现方块掉落调度逻辑，整理实验报告	25%
15331220	刘沅昊	组员	早上班	负责实现登录，上传、获取分数	25%
15331288	唐玄昭	组员	晚上班	负责实现游戏基础逻辑，录制视频	35%
15331067	鄧子漫	组员	晚上班	负责游戏素材收集	15%

二、开发环境

操作系统：windows 10
语言：C++
开发工具：Visual Studio

三、 项目阐述

名称：二重奏
简介：同步控制两个小球，躲避掉落的方块，保持冷静，坚持尽量长的时间
功能：左右按键控制两个小球旋转，躲避掉下的方块，游戏难度会随着分数提高而增大
运行方式：在 server 文件夹中 npm install 安装包，npm start 运行后台服务器后，方可正常进行游戏

四、 项目展示

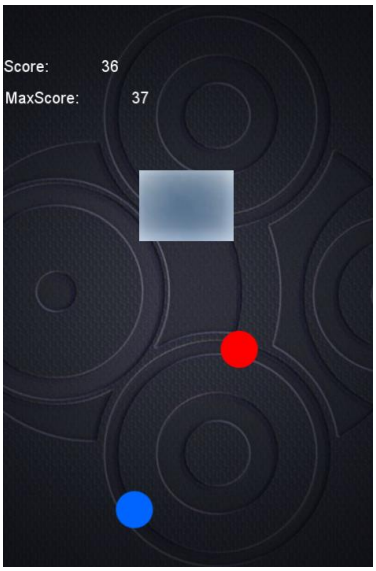
1. 登录页面

服务器后台

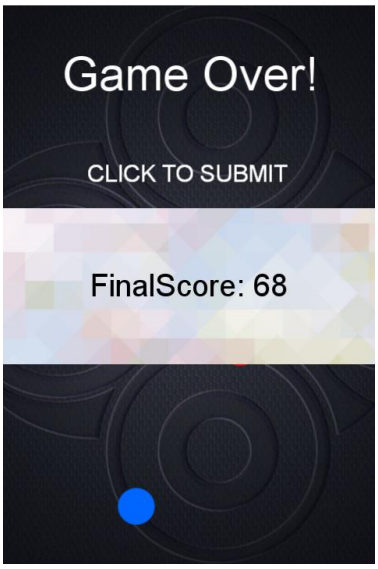
```
> server@0.1.0 start D:\Study\现操\期末\Cocos2d-Final-Project\server
> node bin/www

  <-- POST /
[ { username: '123', session: 'j4u2sqbw04qce2fe5d9x5vw4' } ]
POST / - 8ms
  --> POST / 200 24ms 24b
  <-- POST /
[ { username: '123', session: 'j4u2sqbw04qce2fe5d9x5vw4' } ]
POST / - 3ms
  --> POST / 200 6ms 24b
  <-- POST /
[ { username: '123', session: 'j4u2sqbw04qce2fe5d9x5vw4' } ]
POST / - 1ms
  --> POST / 200 3ms 24b
  <-- POST /
```

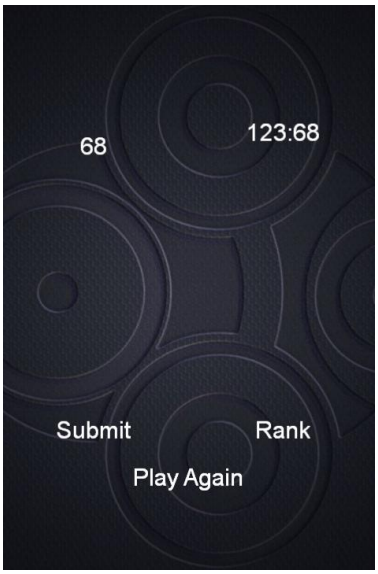
2. 游戏页面



3. 方块碰到小球时，游戏结束



4. 上传分数并获取游戏排名



服务器后台

```
POST / - 2ms
  --> POST / 200 13ms 24b
  <-- POST /score
[ { username: '123',
    session: 'j4u2sqbw04qce2fe5d9x5vw4',
    score: '68' } ]
POST /score - 2ms
  --> POST /score 200 8ms 7b
  <-- GET /score
GET /score - 0ms
  --> GET /score 200 31ms 7b
```

五、 项目难点及解决方案

1. 设置小球旋转的函数：

```
auto x = redBall->getPosition().x - origin.x - visibleWidth / 2;
auto y = redBall->getPosition().y - origin.x - visibleHeight / 4;
float num;
if (x / (visibleWidth / 4) > -1 && x / (visibleWidth / 4) < 1) {
    num = x / (visibleWidth / 4);
}
else if (x / (visibleWidth / 4) <= -1) {
    num = -1;
}
else {
    num = 1;
}
auto angle = acos(num) * 180 / PI;
if (y < 0) {
    angle = 360 - angle;
}
//CCLOG("-%f %f %f", x, y, angle);
// 每次旋转10°
angle += 10;
x = visibleWidth / 4 * cos(angle * PI / 180) + origin.x + visibleWidth / 2;
y = visibleWidth / 4 * sin(angle * PI / 180) + origin.y + visibleHeight / 4;
//CCLOG("%f %f %f", x, y, angle);
redBall->runAction(MoveTo::create(0.05f, Vec2(x, y)));
```

由于没有选择设置物理世界，小球旋转由纯数学计算位置得到，实现难度上各有利弊。

2. 设置方块掉落调度

```
/*
决定哪个方块掉落，决定掉落所需时间
*/
void GameScene::decideWhichToFall() {
    // Todo: 掉落调度函数
    if (cnt >= 25) {
        cnt = 0;
        if (countBlock < difficulity + 3) {
            countBlock++;
            if (CCRANDOM_0_1() < 0.2) //20%掉落短方块
                fall('1', (int)(CCRANDOM_0_1() * 3) - 1, 20.0 / (float)(difficulity + 10)); //从左中右掉落短方块
            else
                fall('0', CCRANDOM_0_1() * 2 - 1 > 0 ? 1 : -1, 20.0 / (float)(difficulity + 10)); //从左右掉落长方块
        } else {
            if (wait < 2) {
                wait++;
            } else {
                wait = 0;
                countBlock = 0;
            }
        }
        return;
    } else {
        cnt++;
    }
}
```

考虑了游戏难度变化，且方块下落间隔时间和位置都不确定，增加了游戏随机性。

六、 项目总结

本次期末项目初始设计是期望完成一个下落躲避的游戏，主要采用数据的网络传输、音乐音效、事件处理等技术。在实际的完成过程中，由于时间所限，不得不放弃许多设计计划，例如游戏血量和难度调整、游戏模式选择等，期望在今后的学习之余有机会继续完成这些设计。

现代操作系统应用开发这门课程的学习，到这个期末项目的完成，算是告一段落。平台应用和物理引擎的设计使用是当今软件工程应用中除 web 之外的又一大方向，通过这一个学期 15 次课程作业以及期中期末项目实践，我们对 UWP 应用设计和 cocos-2d 物理引擎使用有了简单初步的了解，同时也树立了这方面学习的兴趣点，今后将更加深入学习相关知识，为从事应用开发方向的工作做准备。

最后，感谢老师和 TA 们的辛苦指导，祝（可能有的）暑假快乐~