



一、项目分工

学号	名字	角色	班级	职责	贡献
16340008	蔡梓珩	组长	软（1）	素材准备，游戏设计，代码编写	100%

二、开发环境

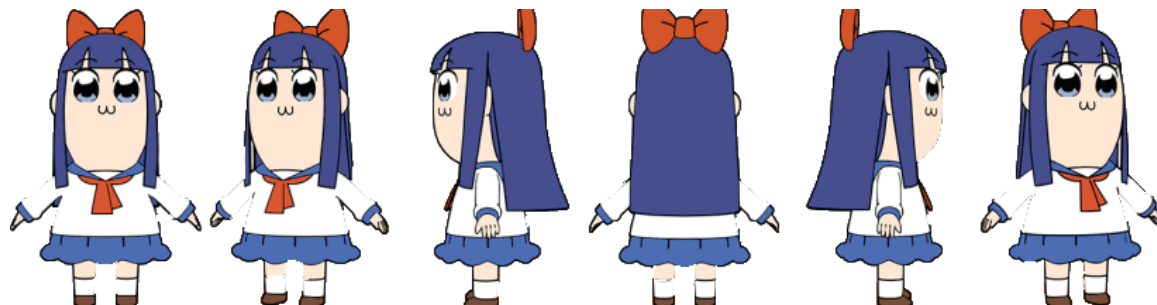
Win10, Cocos2d-x v3.16, Visual Studio 2017, Visual Studio Code

三、项目阐述

这个游戏是一个小型的弹幕游戏。玩家将控制 pipi 美，在躲避 pop 子的子弹之余，射杀 pop 子。名字打算叫“pop 子的深邃黑暗幻想”。

游戏的设计主要借鉴了东方系列的弹幕游戏，借鉴内容包括 pop 子发射的各种弹，以及部分的关卡设计。

玩家控制的角色使用了声优阵容豪华、作画精良的动画ポップテピピックの女主角之一 pipi 美。（下图为裁剪后用作帧动画的 png）



敌人则是同作品的另一主角 pop 子。

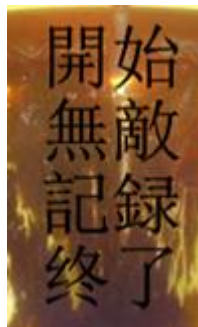


游戏的玩法与雷电或雷霆战机类似（实际上弹幕的设计比起这两款更接近东方系列的游戏，考虑到东方的受众，这里用这两款游戏举例），玩家将使用方向键控制角色移动，躲避敌人发射的密集的子弹，同时游戏会从玩家角色发射子弹，玩家不需要另外按键来主动发射子弹。

游戏主界面有四个菜单按钮，分别是开始游戏，作弊模式，查看记录，退出。因为游戏用了不少日本



ACG 元素，因此菜单按钮也用了日文，这里需要用 xml 载入文字。作弊模式能免疫子弹上海。查看记录能查看历史游戏最快通关记录（如果能通关的话）。



玩家的子弹有三种模式，会随着敌人血量减少而升级。第一种为普通的单发子弹（图为蝴蝶结）。第二种为双发的子弹（左右手比中指）。第一种升为第二种时会有路飞开二档的声音，由于是从动画录的音，因此效果不怎么好。第三种为画符，能自动追向敌人，是考虑到后面的关卡设计可能无法一直瞄准敌人而设计的模式。第二种升第三种时，会有承太郎的声效。攻击模式的升级不会替代旧模式，而是添加。



敌人有六种攻击模式，有三种类型的子弹。具体的关卡设计可以在游戏中无敌状态中体验，或者看下面的项目展示。每个关卡对应一个攻击模式。攻击模式会随着血量减少而改变。玩家通过某个关卡时，会清空子弹，并双方休息 3 秒。第三种子弹的发射有音效。



游戏上方会记录本次游戏花费的时间，游戏如果通关，则会判断这次成绩能否进历史成绩的前十，如果可以，会提示ニューレコード（new record），并将成绩按排名添加到记录中。首次启动时成绩默认为 59: 59: 59。

玩家角色身上有一个红边点，是碰撞的判定点。为了降低游戏难度，实际判定面积是红边点面积的 2/3（点的半径为 7.5，刚体创建时设置的半径为 5）。游戏界面右上角为玩家剩余生命，初始为 3。玩家被子弹触摸就会立即死亡，死亡时会清空子弹，死亡后会闪烁，并有两秒的无敌时间。

背景地图会随着敌人生命减少移动。

知识点：

1. 物理引擎。双方的子弹，双方的角色都有设置刚体。设置刚体方便了子弹自己运动以及触摸检测，关



节绑定等。

2. 调度器。子弹的发射，血量，计时，攻击模式的变化，边界的检测等都用了调度器。
3. 音效。使用了 AudioEngine 加载音效。
4. 帧动画。Player 在循环帧动画旋转。
5. 动作。Player 第三种模式的追踪子弹，以及敌人一直在旋转，都使用了动作。
6. 汉字（日文）读取。使用了 xml 读取文字，解决了汉字支持问题。
7. 数据存取。使用了 UserDefault，存储历史记录。
8. 监听器。键盘监听，控制角色。
9. 预加载。预加载图片和音效。
10. 粒子系统。子弹被清空时（通过关卡或死亡），子弹会有爆炸效果。游戏胜利时，敌人会有漩涡效果。
11. 画面跳转。



四、项目展示

开始菜单：



下面的游戏截图都是在无敌状态下载图。

游戏第一个关卡：





游戏第二个关卡:

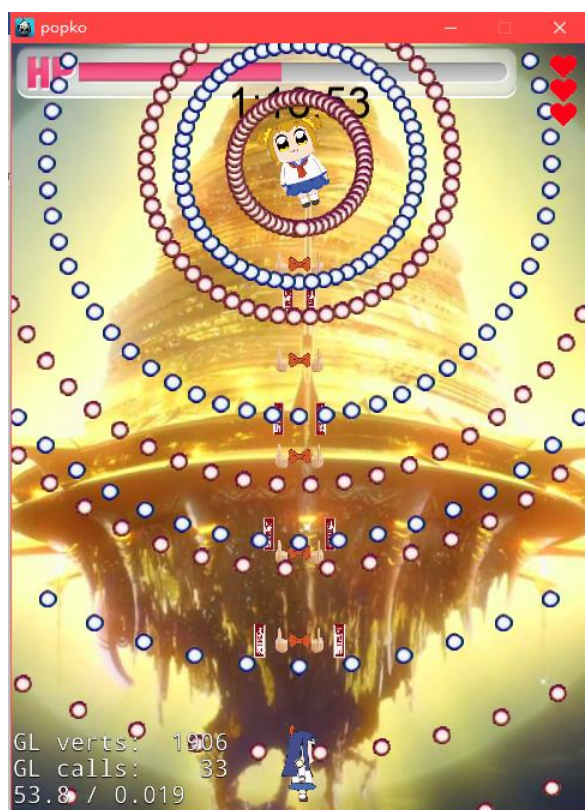


游戏第三个关卡:

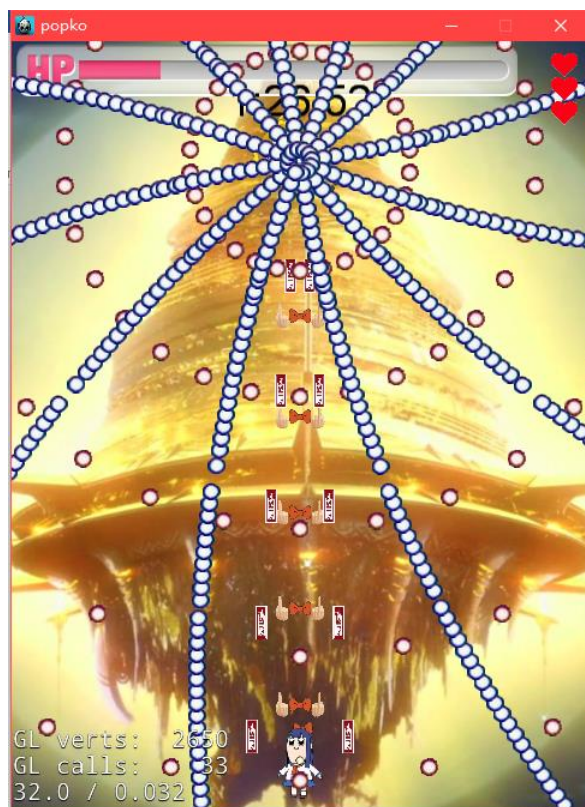




游戏第四个关卡:

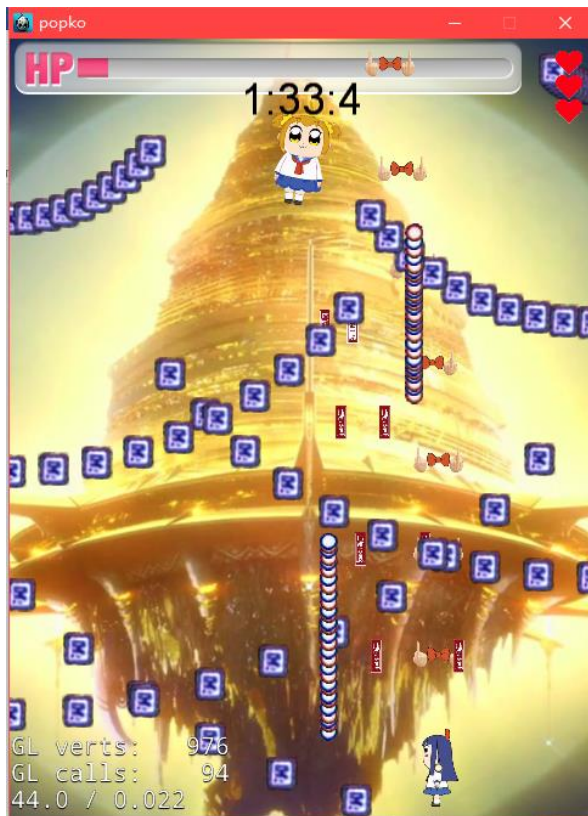


游戏第五个关卡:





游戏第六个关卡：

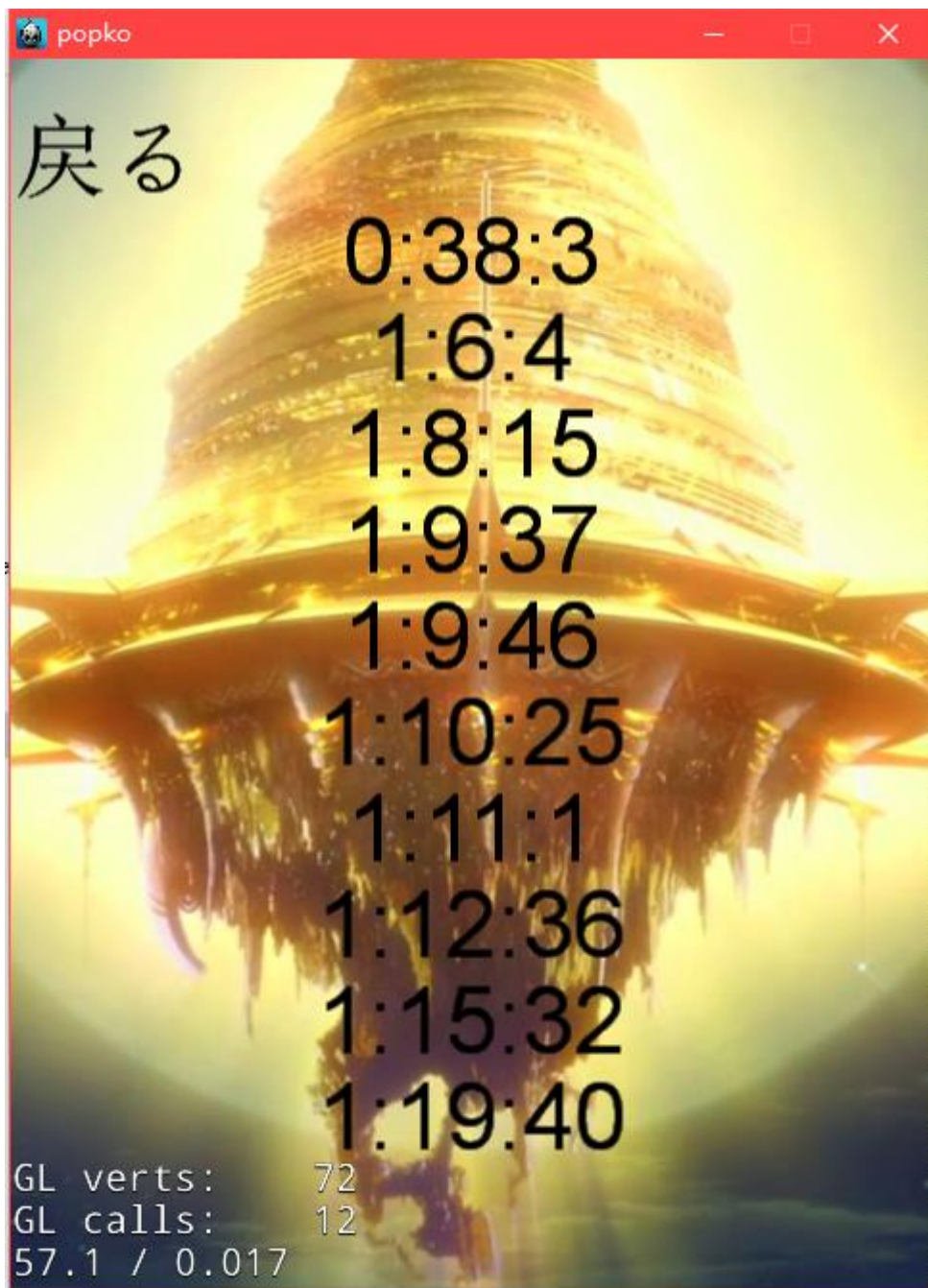


胜利界面（可以对比第一关，移动到了地图上方）：





查看记录（部分数据是测试代码时留下，会有如同开挂一般的成绩（其实就是开挂））：



五、项目难点及解决方案

- ① 抠图和排图。由于需要透明背景的 png 图片作为精灵图源，因此需要手动抠图。对于帧动画，则要对 gif 的每一帧抠出，然后再等距排列，对此稍微学习了一点 photoshop 的使用。
- ② 音效。SimpleAudioEngine 无法满足需求，一方面是有时候子弹太密集，SimpleAudioEngine 可能会明显卡顿，其次，SimpleAudioEngine 无法读取我准备的背景音乐（可能是体积 4M+ 过大）。于是使用了 AudioEngine 作为声音引擎。同时音效在网络不一定能找到现成的，需要自己去录。实际上在素材（图，音）上花费了大量时间。



- ③ 刚体之间的碰撞和接触。因为涉及的刚体较多，要对每种刚体的掩码进行修正，做到刚体之间的行为符合游戏需求。
- ④ 日文支持。由于试用了不少日文字体，字库似乎都不够全，会显示不出来，最终还是用了提供的 Arial。
- ⑤ 帧冲突。测试时遇到了一种特殊的情况，就是敌人死亡与玩家死亡在同一帧发生，结果就是出线如下图所示的清空。因此需要一个类似锁的变量，当胜利时，死亡不会发生。



- ⑥ 硬件超荷。由于有可能出现大量的精灵和刚体，粒子系统制造大量单位，游戏播放大量音效，造成会卡顿，因此需要优化精灵出界面后的清除算法，达到最快的轻除效果。需要观察 GL verts 数值确定是否在合理范围内。
- ⑦ 关卡设计。关卡设计十分消耗时间。要考虑到不能过难又不失去游戏可玩性。弹幕的设计部分借鉴了东方系列游戏，但东方系列的关卡设计十分困难，并不能套用。设计过程需要不断调整子弹刚体参数，



以及发射子弹的密集程度等，不断重复试玩，以在游戏性和难度之间找到一个平衡，并在关卡间存在一定的难度梯度。

- ⑧ 场景传参。因为无敌和普通两种模式使用的是同一个代码，因此需要在主菜单传参过去。一开始试过试用全局变量，或修改静态变量，都有无法解决的报错。最终在 `CreateScene` 函数中添加个参数，创造时传入即可。

六、项目总结

由于与期中一样是单人成组，好处就是有任何想法都能马上添加进游戏，游戏的修改变得十分方便。如果没有一个足够高水平的领导者进行分工，实际上 `teamwork` 项目将十分难进行。但是单人成组的同时，许多工作都要自己完成，尤其是素材的准备，将耗费不少时间。游戏的制作从项目创建到写报告总共使用了 10 天。

制作游戏时会发现，代码逻辑以外反而会遇到许多问题。类似关卡设计，要尽可能地在难度上找到一个平衡点，又要保证弹幕不过于单调。计算机硬件的能力也是代码以外会遇到的问题。即使代码逻辑正确，如果造成显卡，内存或 CPU 的超荷使用，则会卡顿，这时就要对代码或者关卡设计进行优化，以改善游戏体验。

游戏设计时对调度器十分依赖。如果想实现 n 个时间单位后进行某种操作，要依赖调度器（也可能是我不知道其他更方便的方法）。检测游戏状态的改变也需要依赖调度器。一开始以为自定义事件来分发可以自动检测，但似乎并没有那么理想。最终发现，将想要达到及时监听效果的内容都添加进 `update` 函数中重载，即是最方便的方法。

使用刚体可以避免大量的使用调度器检测碰撞的代码，确实减少了很多代码量。一个物理引擎对于游戏开发想必是十分重要的。

实际上开发前还有不少想法，最终由于实现难度以及个人精力的原因，都放弃了。例如音效，实际一开始是打算为每一个子弹发射都添加音效，后来发现体验并不好，就删去了。还有就是一开始打算给 `player` 添加类似奥义等内容，考虑到游戏内容少，游戏时间并不长，不适宜添加过多元素，便放弃了。

通过课程的学习，对游戏的开发有了很多新的理解。现在对平时看到的小游戏或者手游的实现逻辑都有了更深刻的理解和看法。这次开发过程中，最困难的部分是玩法的设计和游戏素材的收集。相信对于普通的非大型游戏而言（没有太多代码实现上的技术困难的游戏），`idea` 是最难的部分。

总体而言，虽然课程任务较繁重，项目也用了许多精力，但确实是学到了很多。类似数据库的应用，`xml` 文件操作，亦或物理引擎、粒子系统这些经常能听到但没实际接触体会过的东西，甚至开发过程中自学 `Photoshop`，都是收获。