



一、项目分工

学号	名字	角色	班级	职责	贡献
14331210	莫浩然	组长	周二班	负责第一关卡的设计以及代码实现	1/3
14331211	莫华晓	组员	周二班	负责第二关卡的设计以及代码实现和后期代码整合界面连接	1/3
14332006	谷溢	组员	周四班	负责第二关卡的设计以及代码实现	1/3

二、开发环境

- 1) 开发系统: Win32
- 2) 编程 IDE: VS 2015 Community
- 3) 编程语言: C++

三、项目阐述

1. 名称

TinyGame

2. 简介

游戏为闯关形式,分为 3 个关卡,讲述一个英雄需要经历陆海空三层世界的各项阻拦,收集道具,最后救出被老妖魔俘虏的同伴的故事:

第一关为冒险夺宝类游戏。游戏场景是陆地上的某片被妖魔们占领的区域。英雄需要躲避老妖魔的手下一小妖魔们释放的火球、子弹、炸弹等攻击,收集齐打开第二世界大门所需的 20 颗钻石。英雄可以左右移动(按左右方向键或者 A、D 键),或者跳跃(空格键),英雄跳跃的高度由按下空格键的时间决定。由于小妖魔们中有个卧底,所以在某个时间里会扔出清屏道具(同时消除所有的攻击)和回血包(给英雄补血),但为了身份泄露,扔完道具后他也会发出炸弹攻击来掩饰自己身份。凡是触碰到火球、子弹、炸弹等攻击,英雄都会掉一次血,血量为 0 的时候游戏结束。

第二关为类似与雷霆一样的游戏。游戏场景是英雄离开了陆地之后在空中驾驶飞机避开宇宙中的陨石,去拯救被老妖魔俘虏的同伴。英雄可以操作飞机进行左右移动,按照飞机的移动速度大概 30s



之后就可以到达老妖魔所在的空中城堡，飞行了 20s 之后英雄可以得到一个宝物，这个宝物英雄使用之后可以将界面中所有陨石进行清空。在太空中还会有两个不断移动的障碍物，英雄的飞机遇到障碍物则会受到损坏，每次掉血 25,4 次后则判定失败。

第三关英雄来到了魔法城堡，但是被邪恶的老妖魔变成了龙，具有喷火技能，可以秒杀怪物，但是需要消耗魔法，像赛亚人蓄力一样，可以通过 K 键蓄力，J 键喷火，并可以四个方向移动，碰到怪物则会消耗 hp。通关秘籍是打怪数目超过 30，如果提前死亡则失败。如果英雄打败了所有的怪物就可以拯救被老妖魔俘虏的同伴。

3. 功能

1) 调度器的使用

每个关卡都用到调度器，来对定时发生的事件进行处理，比如在第一关，需要用调度器来判断钻石是否被英雄获取、判断英雄是否被攻击击中。

```
schedule(schedule_selector(GameScene1::createDiamond), 2.0f, kRepeatForever, 0);
```

2) 帧动画

英雄的运动动作使用了帧动画。

```
auto texture = Director::getInstance()->getTextureCache()->addImage("$lucia_forward.png");
for (int i = 0; i < 8; i++) {
    auto frame = SpriteFrame::createWithTexture(texture, CC_RECT_PIXELS_TO_POINTS(Rect(68
* i, 0, 68, 101)));
    run.pushBack(frame);
}
```

3) 本地储存

database 的代码部分（主要头现在以下函数部分）：

```
void HelloWorld::killMon(float dt) {
    Rect a(this->player->getPositionX()-400, this->player->getPositionY() - 70, 320, 80);
    if (this->player->isFlippedX()) {
        a.setRect(this->player->getPositionX(), this->player->getPositionY() - 70, 320, 80);
    }
    int score = database->getIntegerForKey("score");
    Vector<Sprite*> colliped = Factory::getInstance()->collider(a);
    int delta = colliped.size();
    if (colliped.size() == 0) return;
    for (auto it = colliped.begin(); it < colliped.end(); it++) {
        Factory::getInstance()->removeMonster(*it);
    }
}
```



```
score += delta;
char buf[20];
auto progress2 = ProgressTo::create(0.1, this->pT->getPercentage() + 3*delta);
this->pT->runAction(progress2);
this->time->setString(itoa(score, buf, 10));
database->setIntegerForKey("score", score);
}
```

4) Tilemap

代码部分:

```
visibleSize = Director::getInstance()->getVisibleSize();
origin = Director::getInstance()->getVisibleOrigin();

TMXTiledMap* bg = TMXTiledMap::create("map.tmx");
bg->setPosition(visibleSize.width / 2 + origin.x, visibleSize.height / 2 + origin.y);
bg->setAnchorPoint(Vec2(0.5, 0.5));
bg->setScale(Director::getInstance()->getContentScaleFactor());
this->addChild(bg, 0);
```

5) 事件处理

每个关卡都用到事件的分发和响应。在关卡一，主要是键盘事件的分发和响应。

```
void GameScene1::addKeyboardListener() {
    auto keyboardListener = EventListenerKeyboard::create();
    keyboardListener->onKeyPressed = CC_CALLBACK_2(GameScene1::onKeyPressed,
    this);
    keyboardListener->onKeyReleased = CC_CALLBACK_2(GameScene1::onKeyReleased,
    this);
    _eventDispatcher->addEventListenerWithSceneGraphPriority(keyboardListener,
    this);
}
```

6) 音乐与音效

游戏插入了很多背景音乐和音效。在场景初始化的时候，需要对音乐、音效预加载，然后在后面适合的场合播放。比如在关卡一，英雄跳跃会播放跳跃音效，获取钻石也会播放获取音效。

```
void GameScene1::preloadMusic() {
    SimpleAudioEngine::getInstance()->preloadBackgroundMusic("sound/bgm1.mp3");
    SimpleAudioEngine::getInstance()->preloadBackgroundMusic("sound/bgm2.mp3");
    SimpleAudioEngine::getInstance()->preloadBackgroundMusic("sound/bgm3.mp3");
    SimpleAudioEngine::getInstance()->preloadEffect("sound/beingAttack.wav");
    SimpleAudioEngine::getInstance()->preloadEffect("sound/blood_Tool.mp3");
    SimpleAudioEngine::getInstance()->preloadEffect("sound/bomb.wav");
    SimpleAudioEngine::getInstance()->preloadEffect("sound/bullet.wav");
    SimpleAudioEngine::getInstance()->preloadEffect("sound/clear_Tool.wav");
}
```



```
SimpleAudioEngine::getInstance()->preloadEffect("sound/get_diamond.wav");
SimpleAudioEngine::getInstance()->preloadEffect("sound/jump.wav");
SimpleAudioEngine::getInstance()->preloadEffect("sound/stone.wav");
SimpleAudioEngine::getInstance()->preloadEffect("sound/explore.wav");
}
```

```
void GameScene1::play_bgm() {
    SimpleAudioEngine::getInstance()->playBackgroundMusic("sound/bgml.mp3",
true);
}
```

7) 物理引擎

在关卡一对英雄使用了物理引擎，实现英雄跳起和下落都受到重力的效果。同时也要对场景边框添加刚体，并设置英雄的各项掩码，使得英雄会跟边框发生碰撞，不至于飞出边界。

```
Scene* GameScene1::createScene()
{
    auto scene = Scene::createWithPhysics();
    //scene->getPhysicsWorld()->setDebugDrawMask(PhysicsWorld::DEBUGDRAW_ALL);
    scene->getPhysicsWorld()->setGravity(Vec2(0, -600.0f));

    auto layer = GameScene1::create(scene->getPhysicsWorld());
    scene->addChild(layer);
    return scene;
}
```

```
player->setPhysicsBody(PhysicsBody::createCircle(35));
player->getPhysicsBody()->setCategoryBitmask(0x03);    //0011
player->getPhysicsBody()->setCollisionBitmask(0x03);    //0011
player->getPhysicsBody()->setContactTestBitmask(0x02); //0010
```

```
void GameScene1::addBackgroundEdge() {
    auto edgeSp = Sprite::create();
    auto boundbody = PhysicsBody::createEdgeBox(visibleSize);
    boundbody->setDynamic(false);

    edgeSp->setPosition(Point(visibleSize.width / 2, visibleSize.height / 2));
    edgeSp->setPhysicsBody(boundbody);

    this->addChild(edgeSp);
}
```

8) 粒子系统



在关卡一攻击爆炸的时候使用了粒子系统。

```
if (stone_set.at(i)->getPosition().getDistance(player->getPosition()) < 25) {  
    //爆炸效果  
    auto hurtExploration = ParticleSun::createWithTotalParticles(150);  
    hurtExploration->setPosition(player->getPosition());  
    hurtExploration->setDuration(1.5f);  
    this->addChild(hurtExploration);  
  
    BeingAttacked();  
}
```

4. 亮点

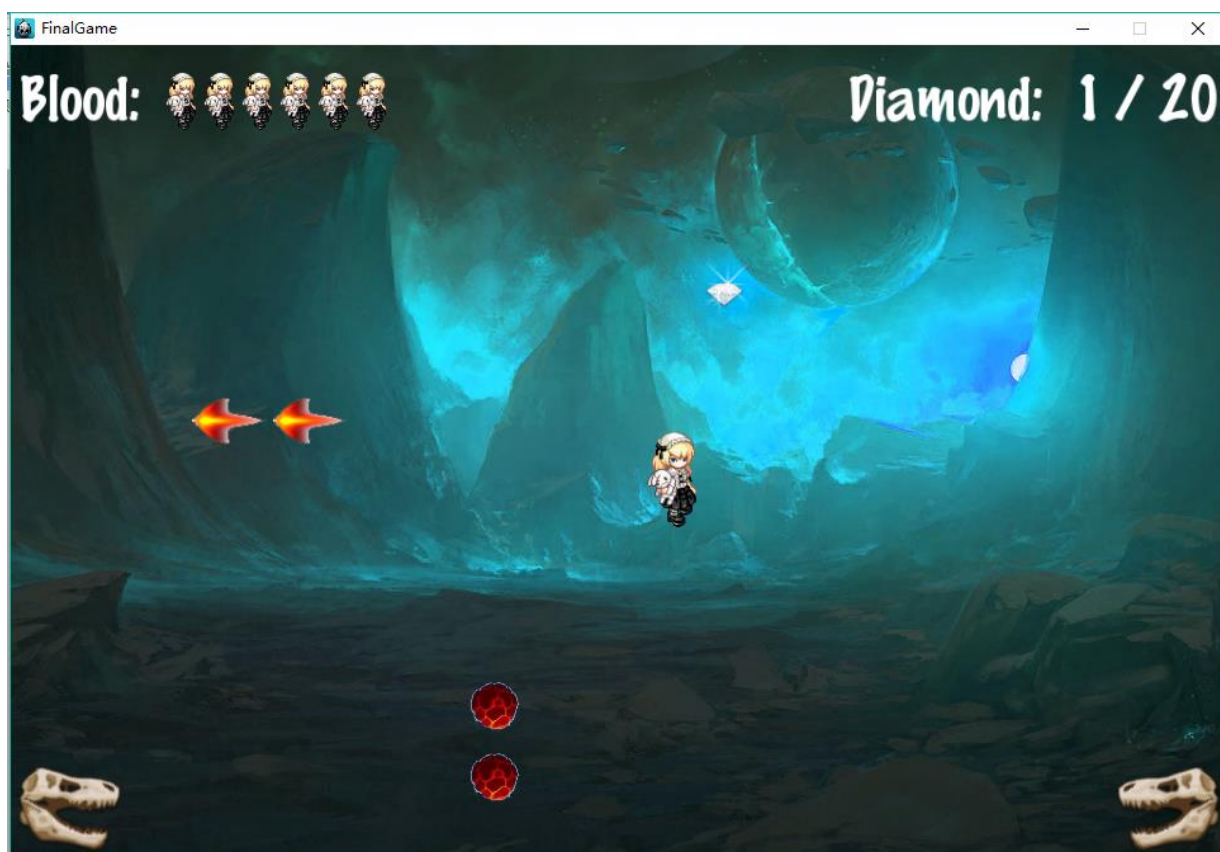
- 1) 关卡一能够设置难度系数，有 4 个系数：简单【2】；中等【3】；难【4】；变态难【5】。难度主要体现在攻击（子弹、陨石）个数；清屏工具大小、出现时间；初始血量多少。根据对难度系数的公式计算即可改变游戏难度，使游戏耐玩性提高。
- 2) 关卡二使用了物理引擎，会在特定时间内出现清屏工具，减少了游戏的难度操作。
- 3) 关卡三动画效果比较酷炫，素材挑了很久
- 4) 由于游戏过程时间较长，TA 点击 Q 即可跳转到下一关卡 2333~
- 5) 实现了自己应用 Tilemap 制作地图，绘制了我和女票的地图，hhhh...

四、项目展示

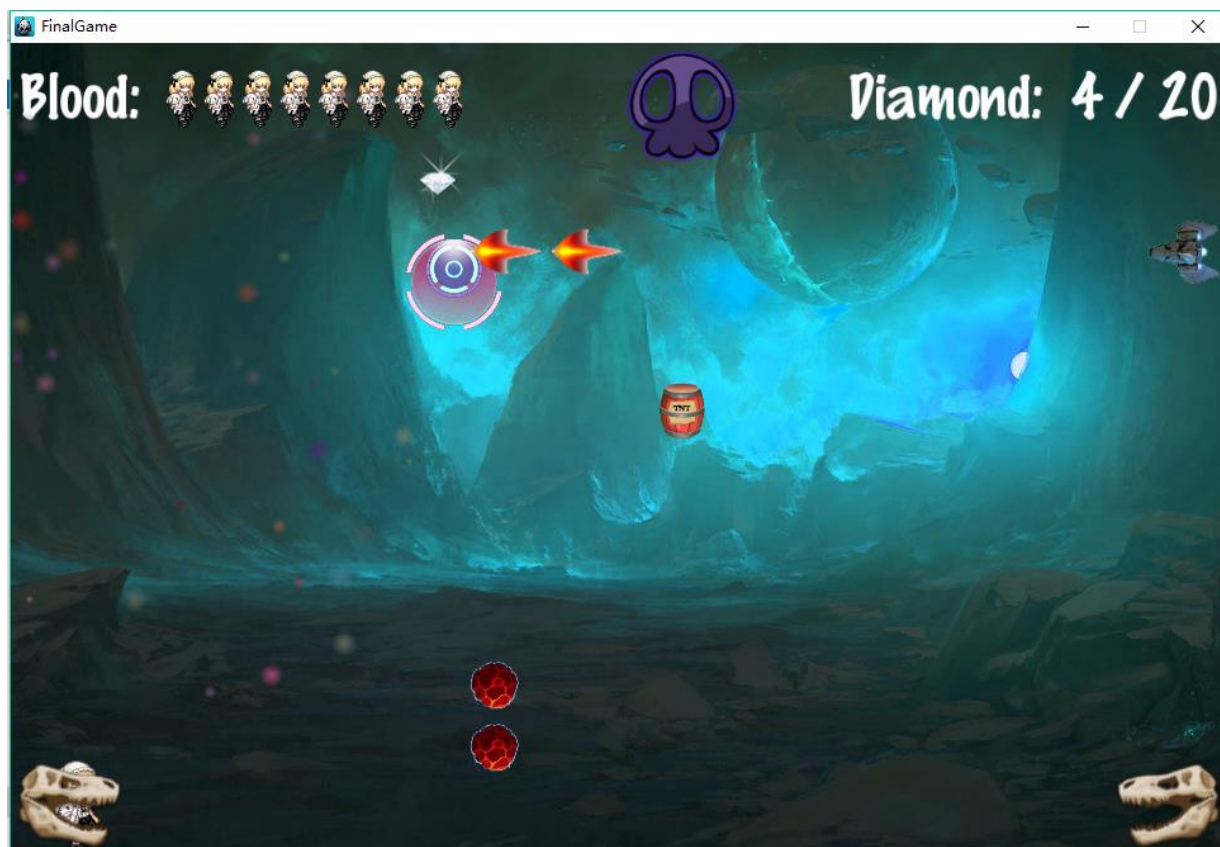
1. 启动游戏后，首先是初始页面



2. 初始页面可以选择游戏难度，一共有 4 个难度。点击 4 个 label 进行选择，被选择的难度变成红色。
3. 进入关卡一，开始第一个游戏。此关卡可以按 A、D 键或者左右方向键控制英雄左右移动。按空格键跳跃（跳跃高度看按下时间长短——英雄上方显示跳跃力度）。英雄需要躲避攻击，并获取钻石。左上方为血量，右上方为获取的钻石数：

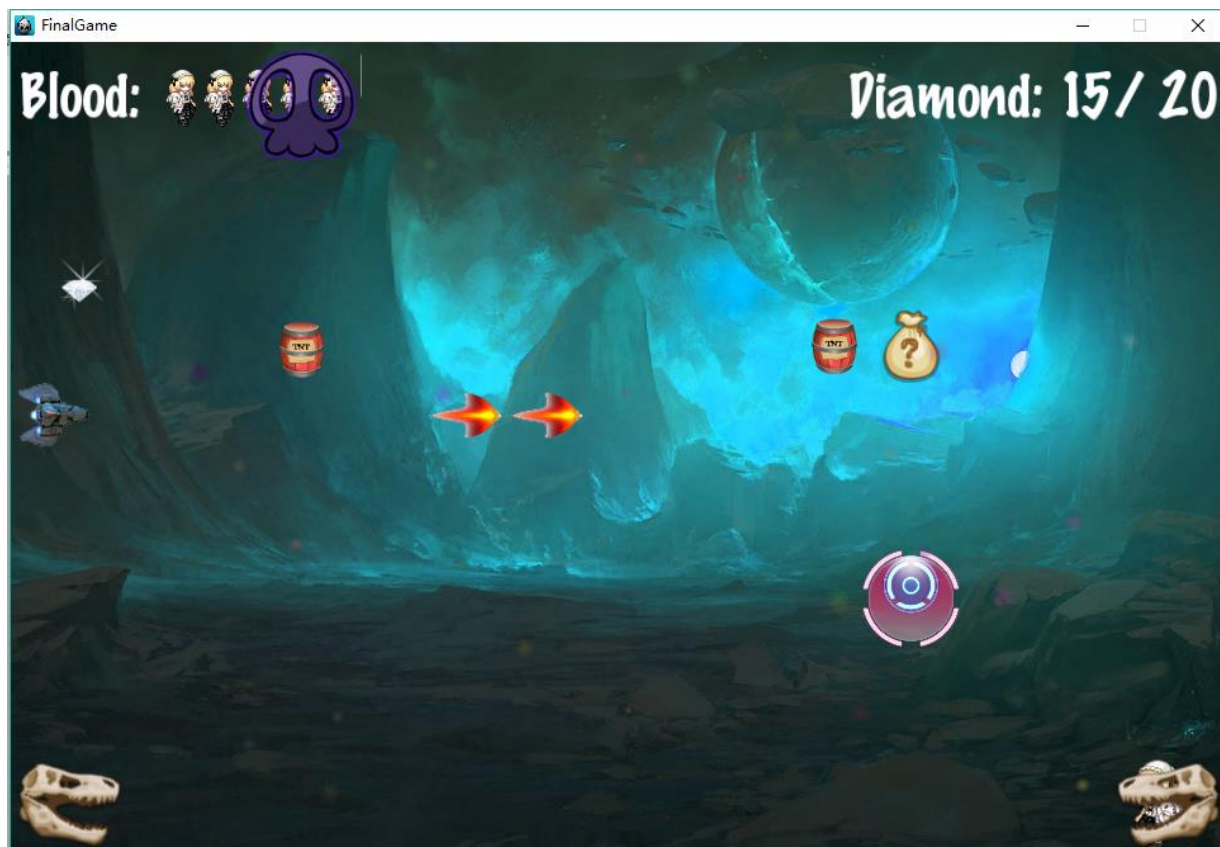


4. 当获取一定数量钻石后，会有新的妖魔（卧底）投放炸药桶（定时爆炸）。同时正如上面游戏介绍所说，妖魔卧底也会投放道具。画面中上方那个粉紫色的圈便是清屏道具（英雄触碰后，场景里所有的攻击都会自爆清除），清屏道具自投放后便永远存在。





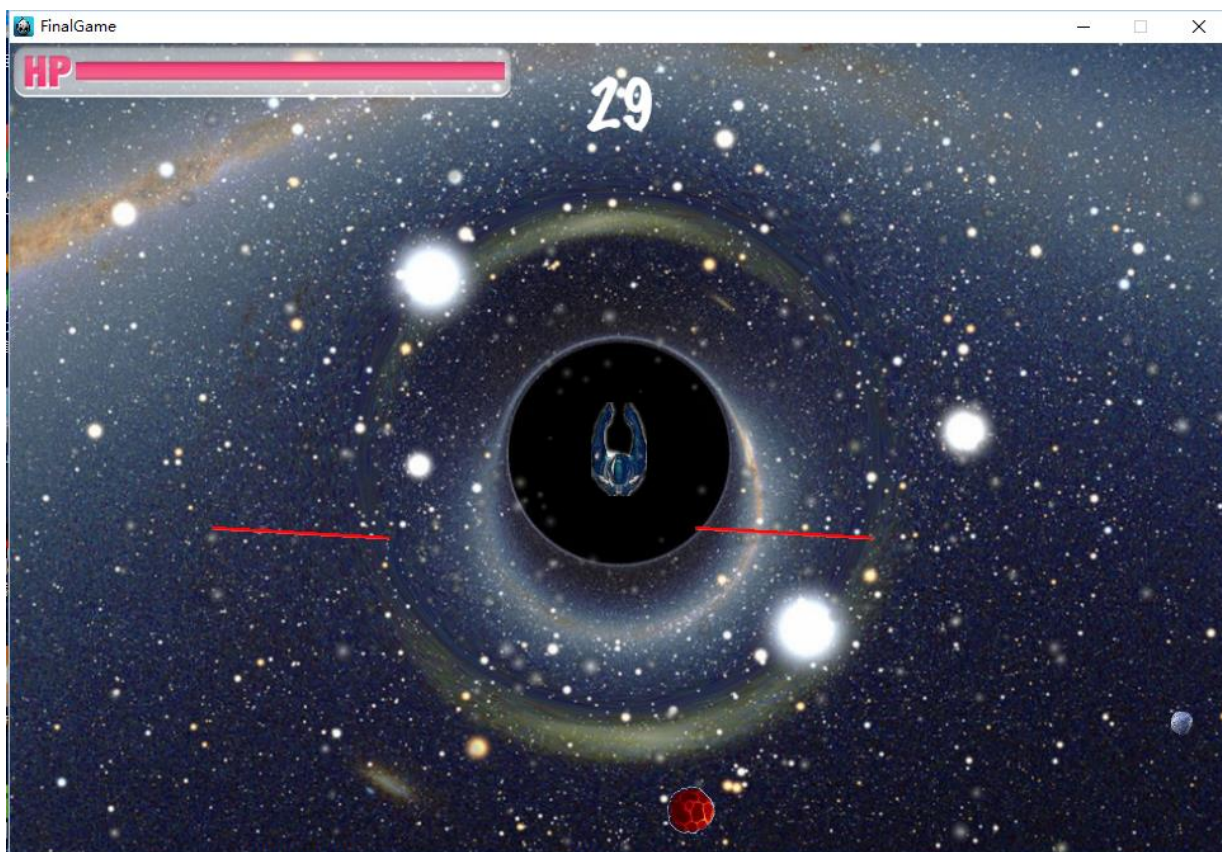
5. 在获得了 15 个钻石后，上方妖魔会投放一个回血包（有问号的袋子），英雄获取后加 3 个血。



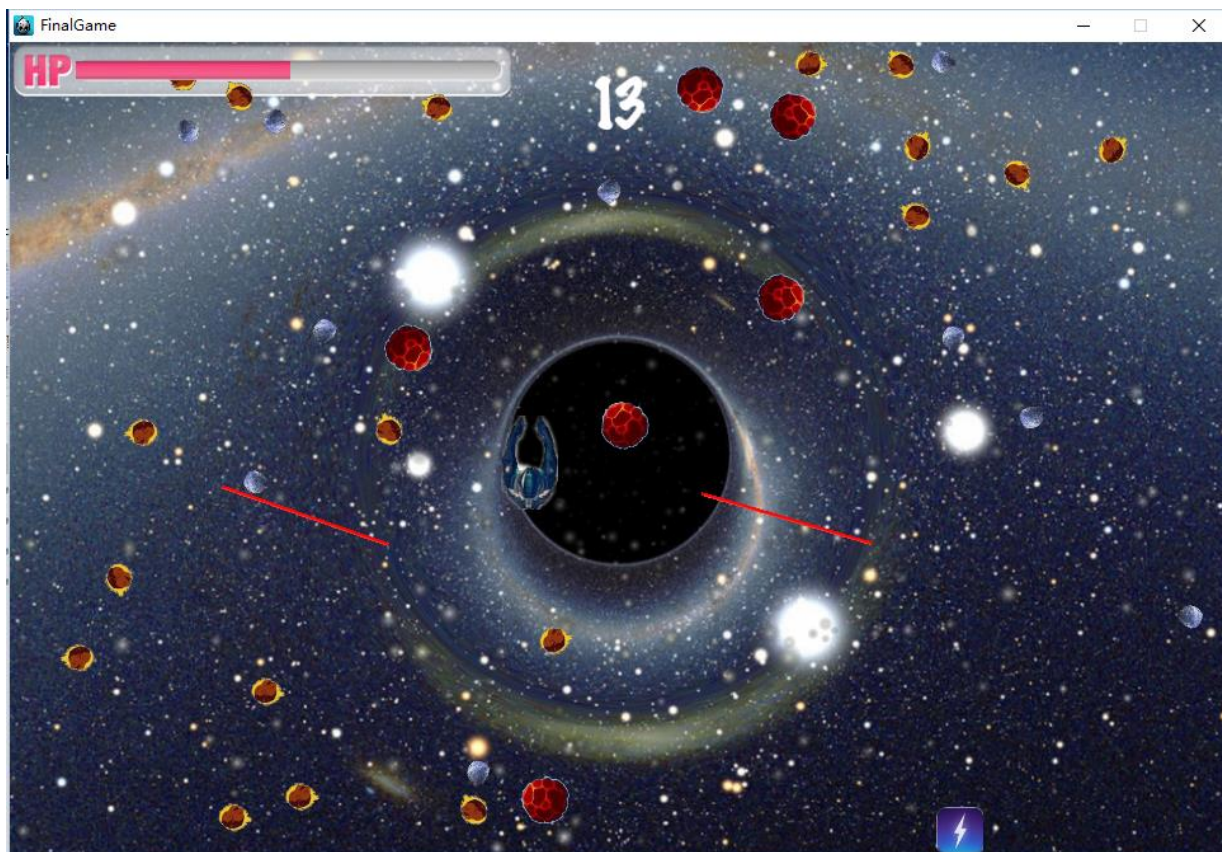
6. 获取了 20 颗钻石后，便出现胜利 logo。跳进胜利 logo 后便进入下一场景



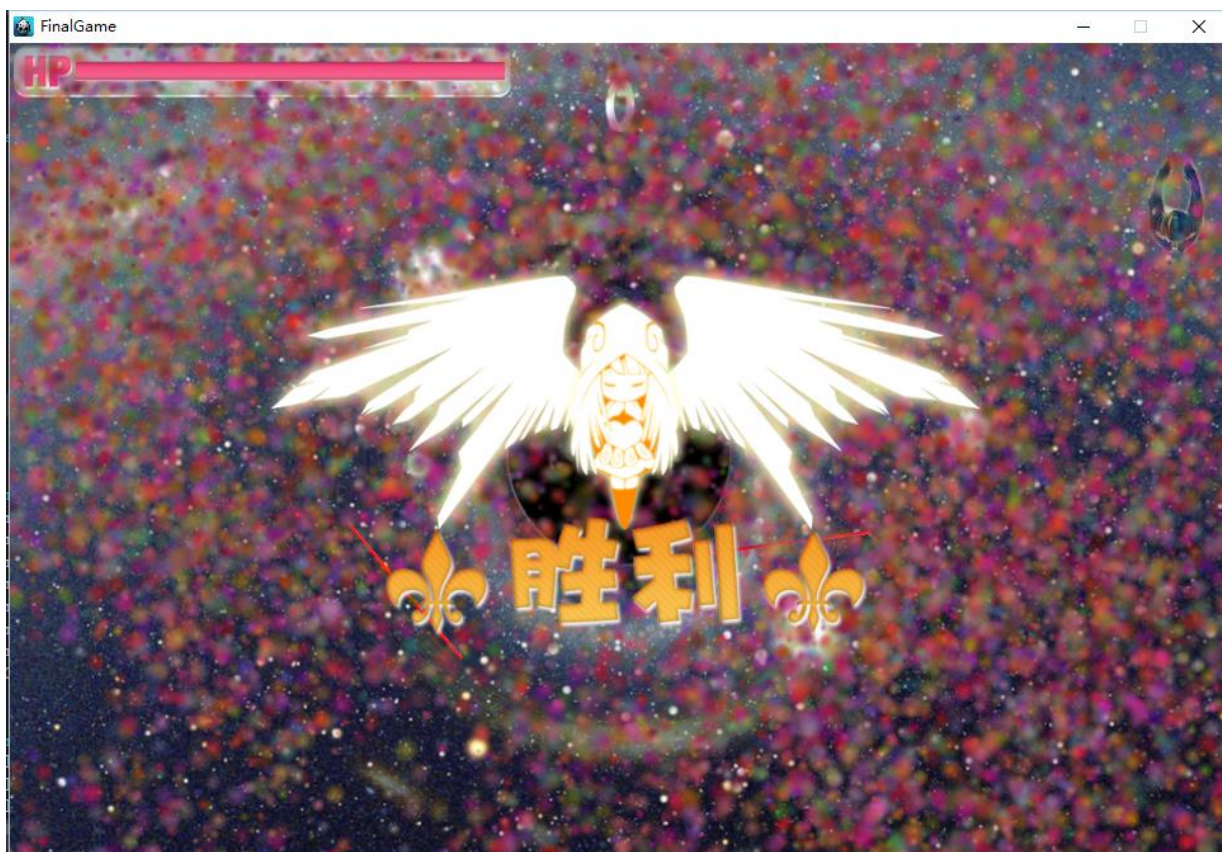
7. 画面跳转到第二关卡



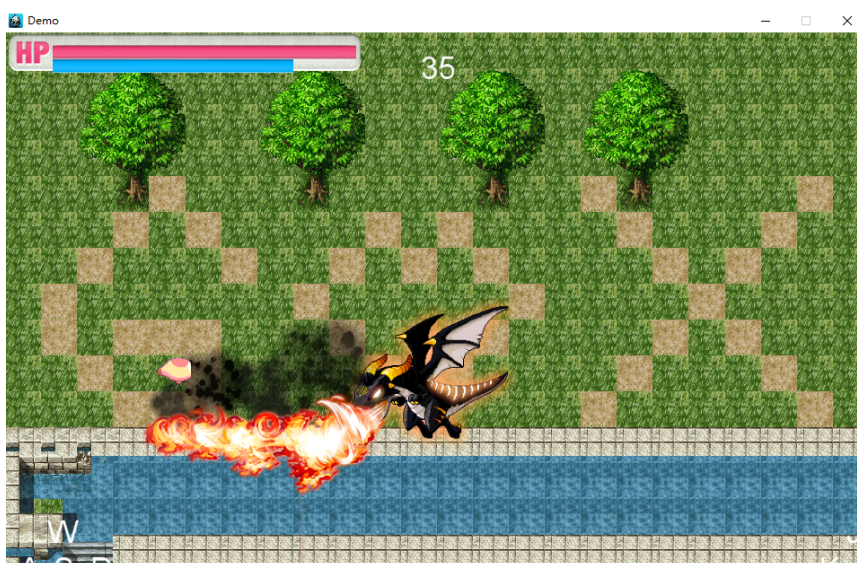
8. 躲避陨石坚持到 20s 出现清屏工具；



9. 每碰撞一次则飞机掉血 $\frac{1}{4}$ ；
10. 坚持 30s 之后出现胜利界面，跳转到第三关卡；



11. 进入第三个关卡:



wasd 控制上下左右，也可用方向键，J 喷火，K 蓄力增加 mp;



五、项目难点及解决方案

1. 总体上来说，我们这个项目所用到的知识都是在作业中所应用到的，所以在项目的难点上遇到的很少，具体的解决方法也可以通过作业上的经验进行解决。这次 **project** 有个美中不足的地方就是，我们小组是每个人完成一个小游戏，在项目合作代码链接上经验可能相对较少，希望在以后的实践中能够学到更多。

六、项目总结

这个项目是从 6.1 左右开始动工的，大概是断断续续做了 3 个星期。由于一开始考虑到我们课堂上学的东西确实不是很多，能运用的知识也只是那么一点，所以只是用已经学到的知识来做一个东西的话，可能最后成品会跟我们的作业大同小异，也可能跟大多数同学做的东西差不多。所以我们小组一开始商量就打算做一些比较少同学会想到而且最后去做的东西。经过讨论，最后我们打算做一个闯关类型的游戏，分为三个关卡，这样也便于我们的分工。在这个过程中，由于我们每个游戏都是近乎独立的，所以分工起来非常明确，大家都对这个游戏有所贡献，从中能够学到东西，这是最棒的一件事情。并且，这个 **project** 近乎和期末的复习冲突了好多，所以作业的完成也断断续续，总体上来说，还是更加地对该课程本学期内学到的内容有了更多的了解，并且应用到了自己项目实践中，这真的是一件非常棒的事情。感谢老师和助教们这学期的教导，我们小组三个人都学到了很有用的东西~