现代操作系统应用开发实验报告

姓名：_____刘宇庭_____

学号：_____16340158_____

实验名称：____cocos2d-x_____

一、参考资料

1. UserDefault 保存的 XML 文件位置
    a) https://blog.csdn.net/w337198302/article/details/41791429
2. tilemap 学习
    a) http://doc.mapeditor.org/en/stable/
    b) http://docs.cocos.com/creator/api/zh/classes/TiledMap.html
    c) http://www.taikr.com/article/1924
3. CC_CALLBACK
    a) https://blog.csdn.net/joncke/article/details/40781357
4. Sqlite 配置与使用
    a) https://blog.csdn.net/u010075060/article/details/41386545
5. 物理引擎
    a) https://blog.csdn.net/z104207/article/details/44591197
    b) http://www.cocos2d-x.org/docs/cocos2d-x/en/physics/queries.html

c) http://www.cocos2d-x.org/reference/native-cpp/V3.3rc0/d4/d11/classcocos2d_1_1_physics_joint_fixed.html

6. 粒子系统

a) http://www.cocos.com/docs/native/v3/particle-system/zh.html

二、实验步骤

1. Hw12

a) 实现怪物随机生成并且向角色移动

b) 完成怪物与角色之间的动作交互，怪物攻击角色，角色扣血，角色攻击怪物，人物加血

c) 使用 tilemap 创建地图

d) 本地数据存储，记录本轮打到的怪物数量以及历史最高，都在游戏中显示

2. Hw13

a) 实现键盘左右移动飞船以及子弹发送

b) 实现鼠标点击拖动飞船移动

c) 完成自定义子弹和陨石碰撞产生爆炸事件

d) 陨石向下移动生成新的一行陨石

e) 音效载入

f) 对细节的处理，飞船可移动的范围，游戏的结束

3. Hw14

a) 键盘可控制板子左右移动

b) 使用关节固定球和板子

c) 顶部生成小砖块

d) 完成碰撞事件定义，当砖、球碰撞则消去砖头，球与地板碰撞则游戏结束

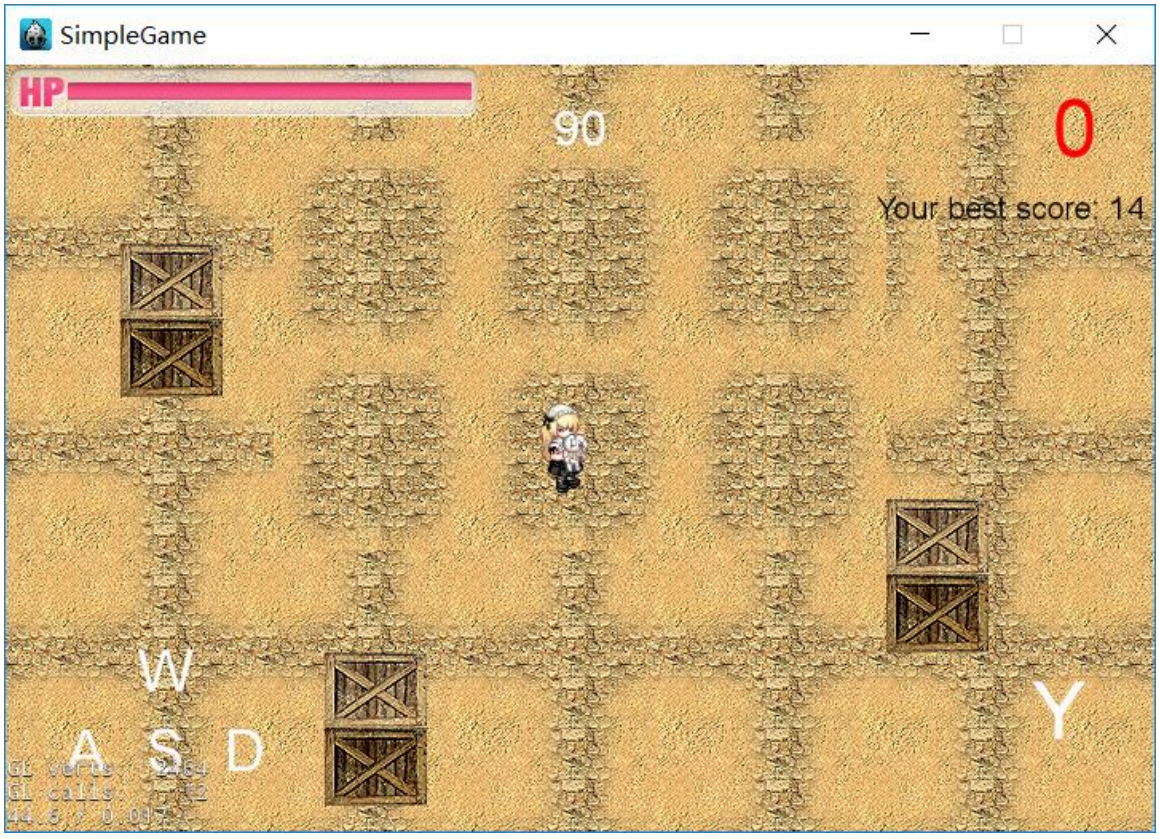
e) 修改板子、球、以及砖块的物理属性，优化游戏的体验

f) 添加粒子效果

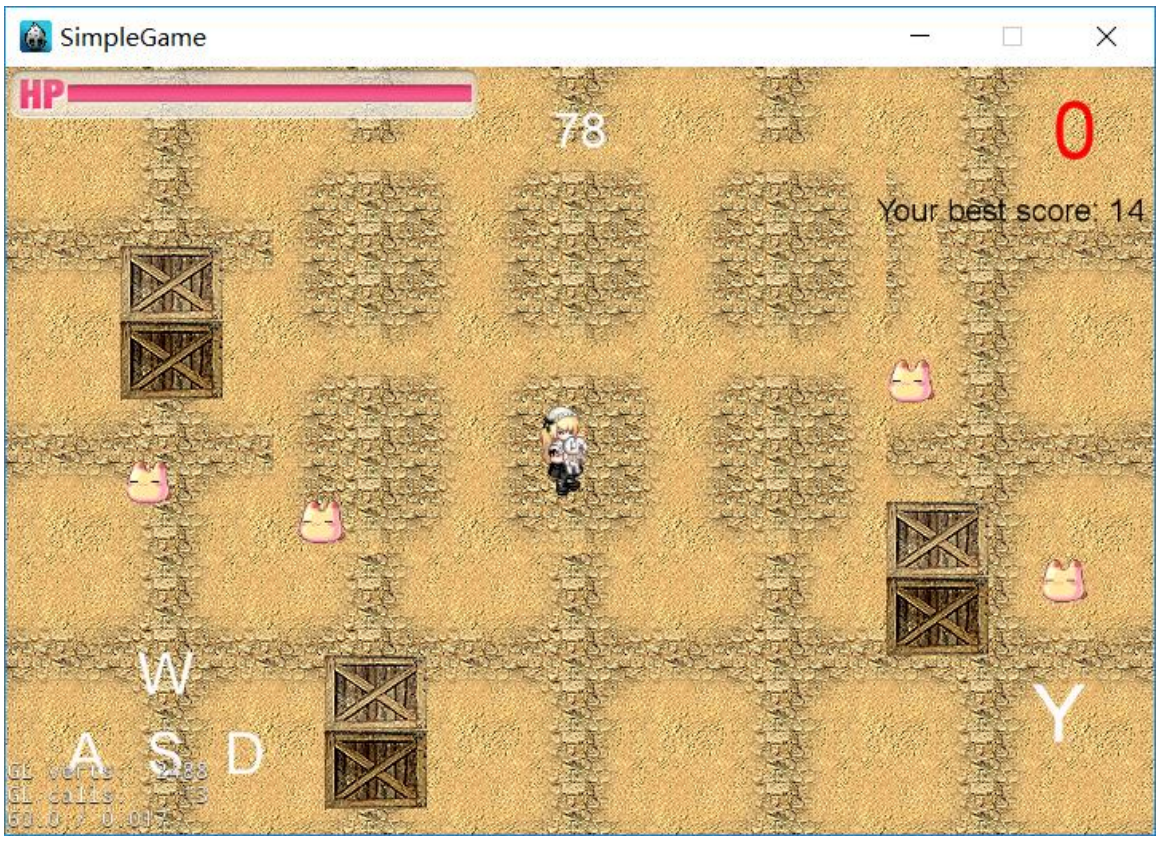
三、关键步骤截图

1. Hw12

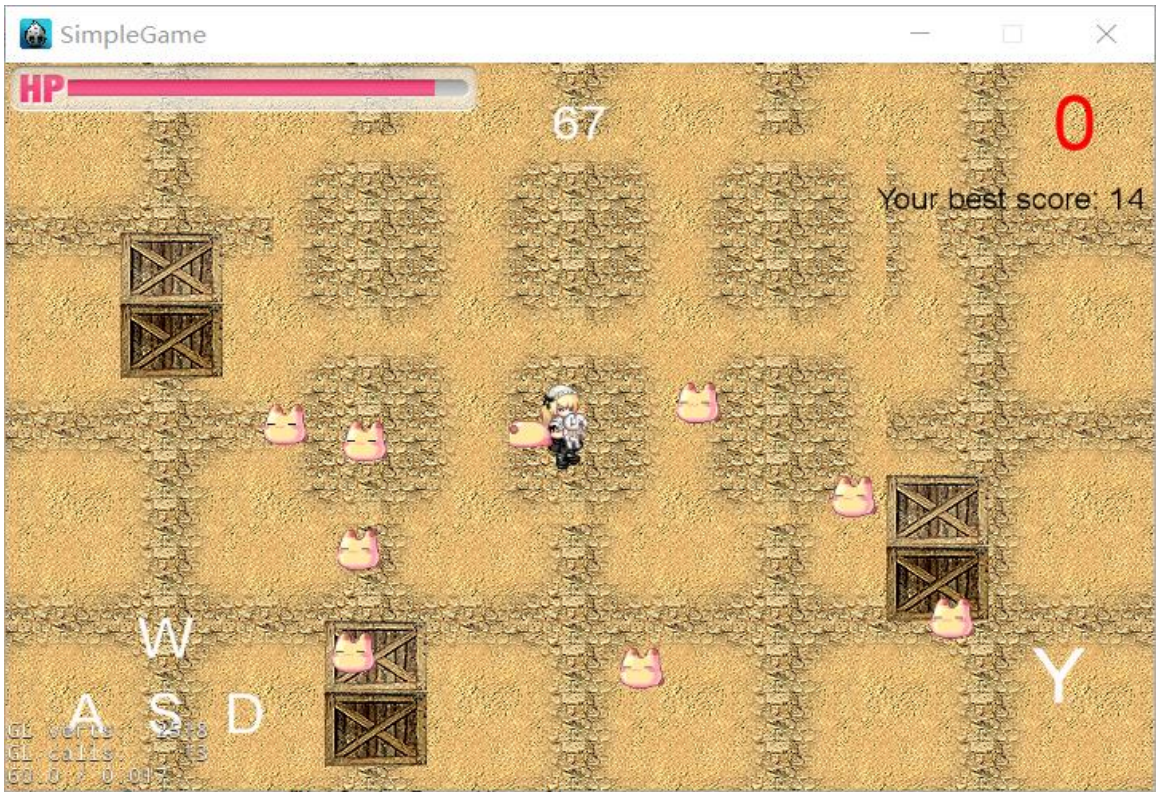
a) 游戏开始



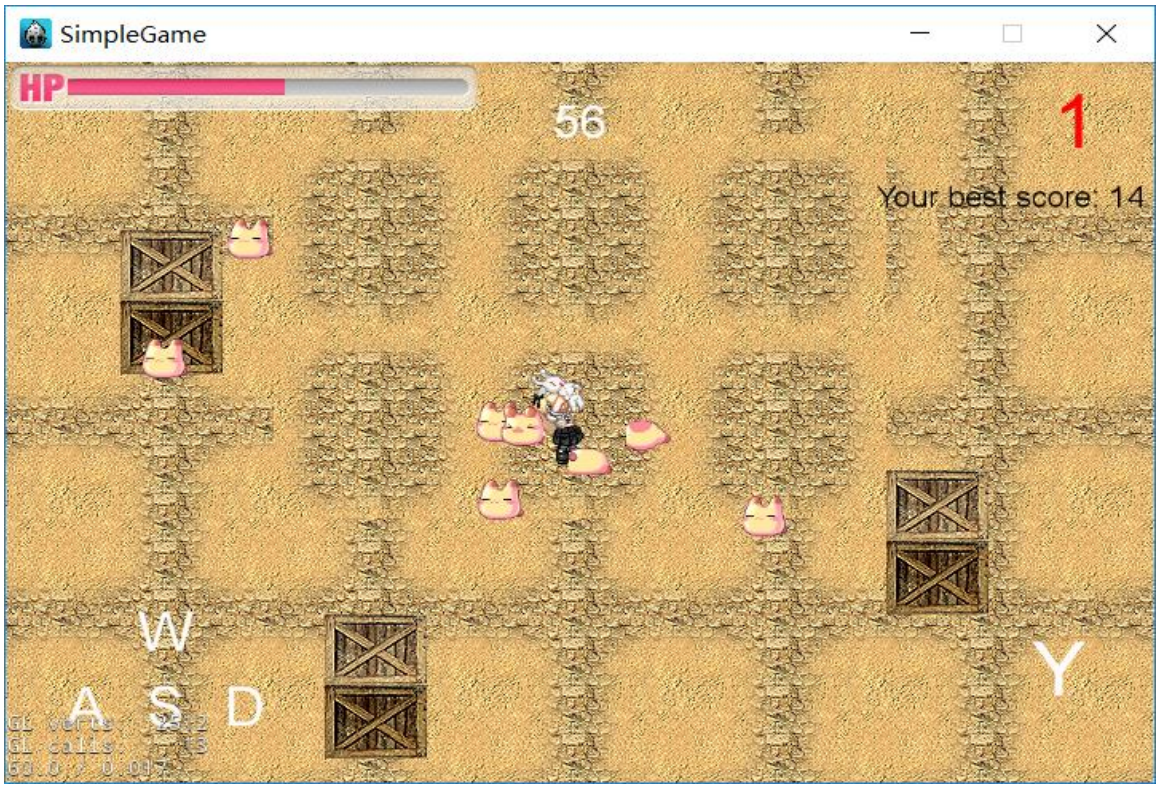
b) 怪物随机生成并且移向人物



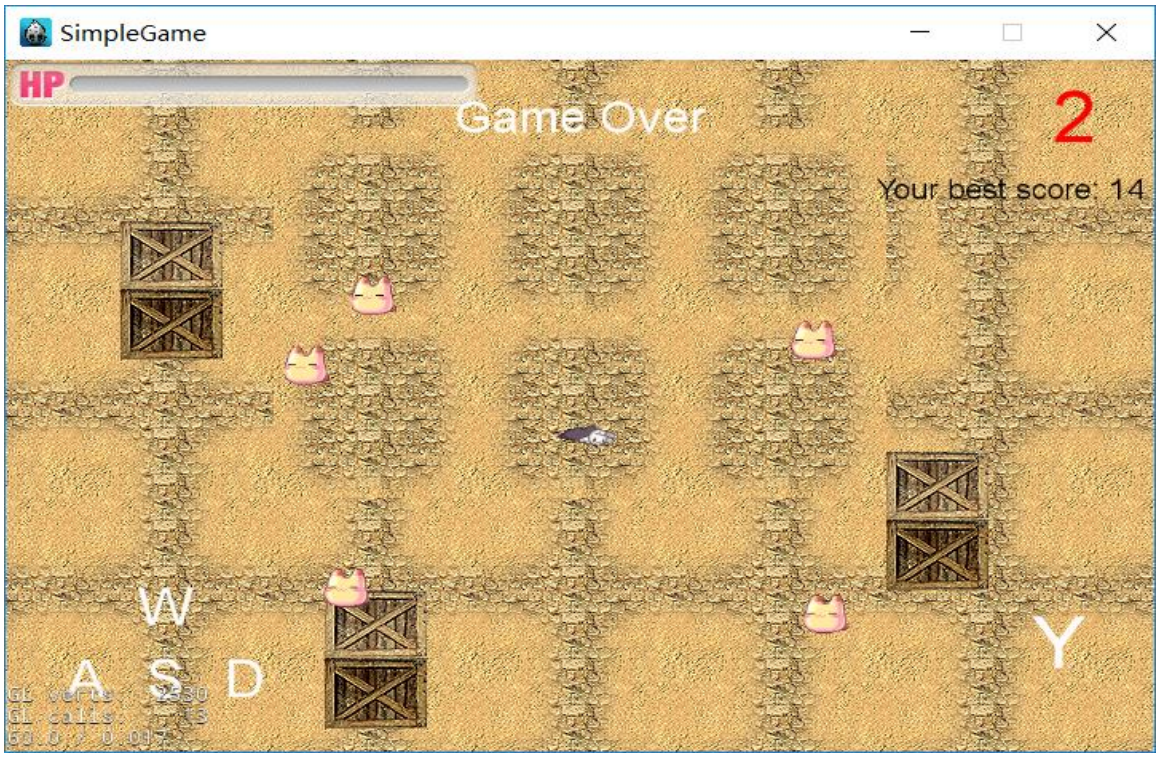
c) 怪物攻击角色，角色扣血



d) 角色攻击怪物，角色加血

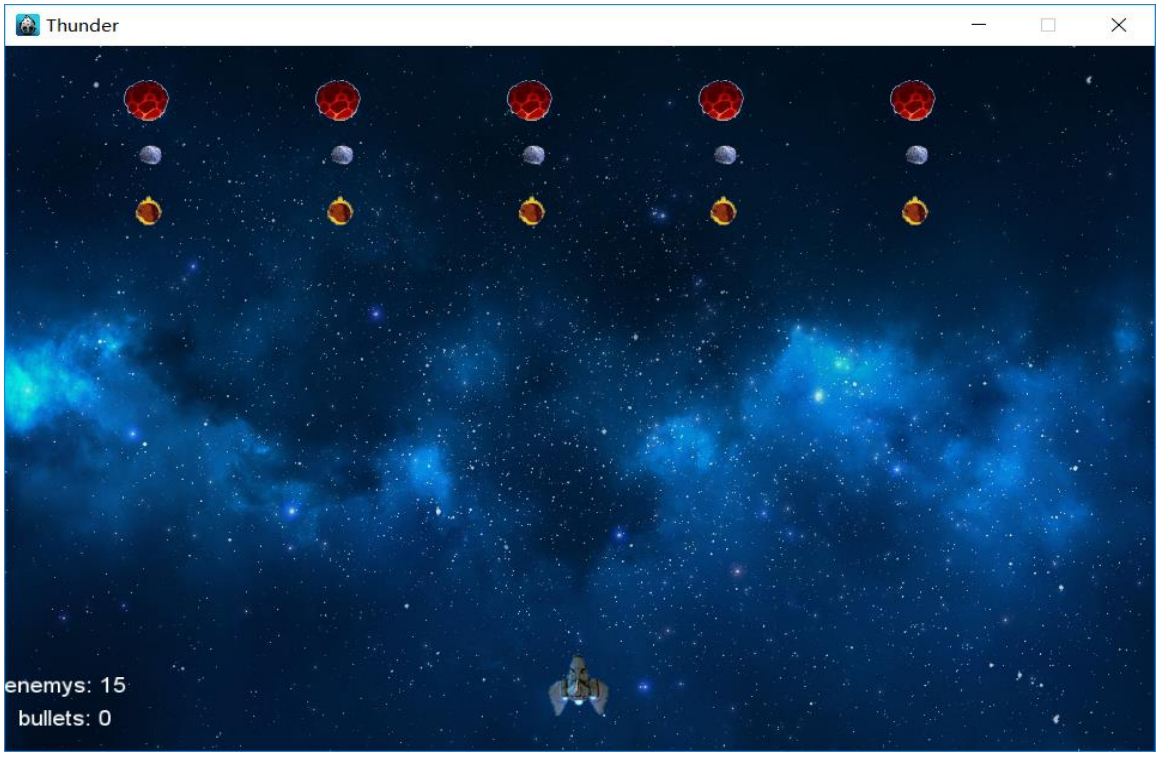


e) 时间结束或者角色死亡，游戏结束

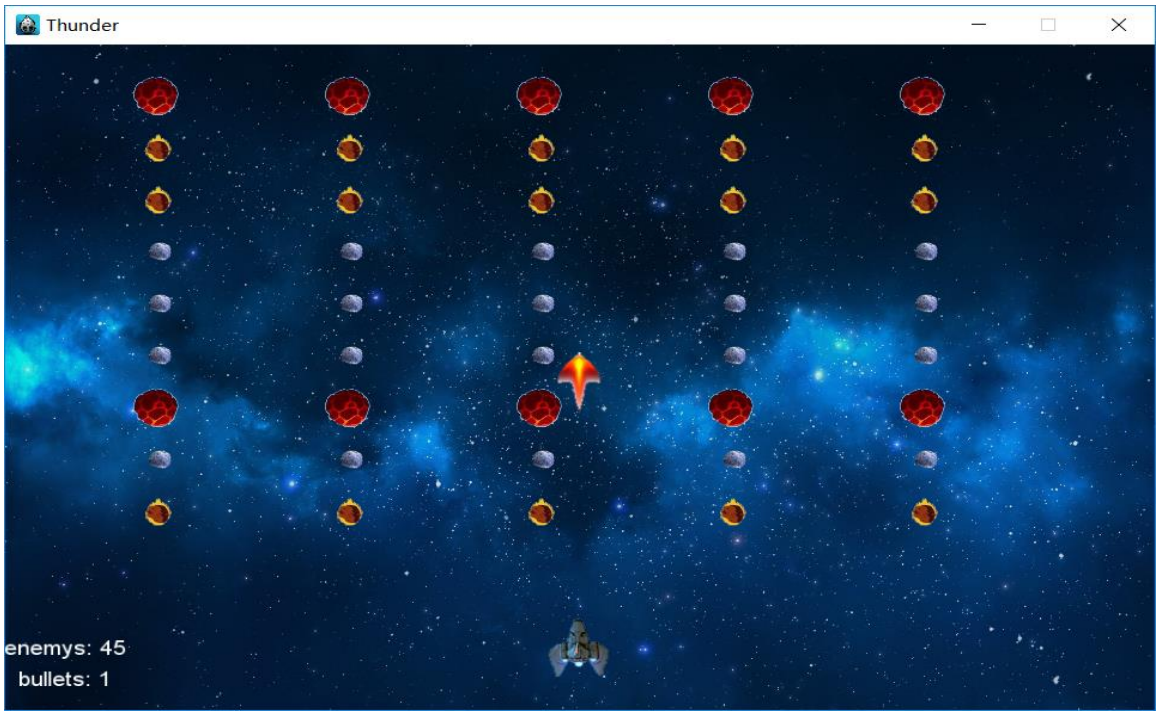


2. Hw13

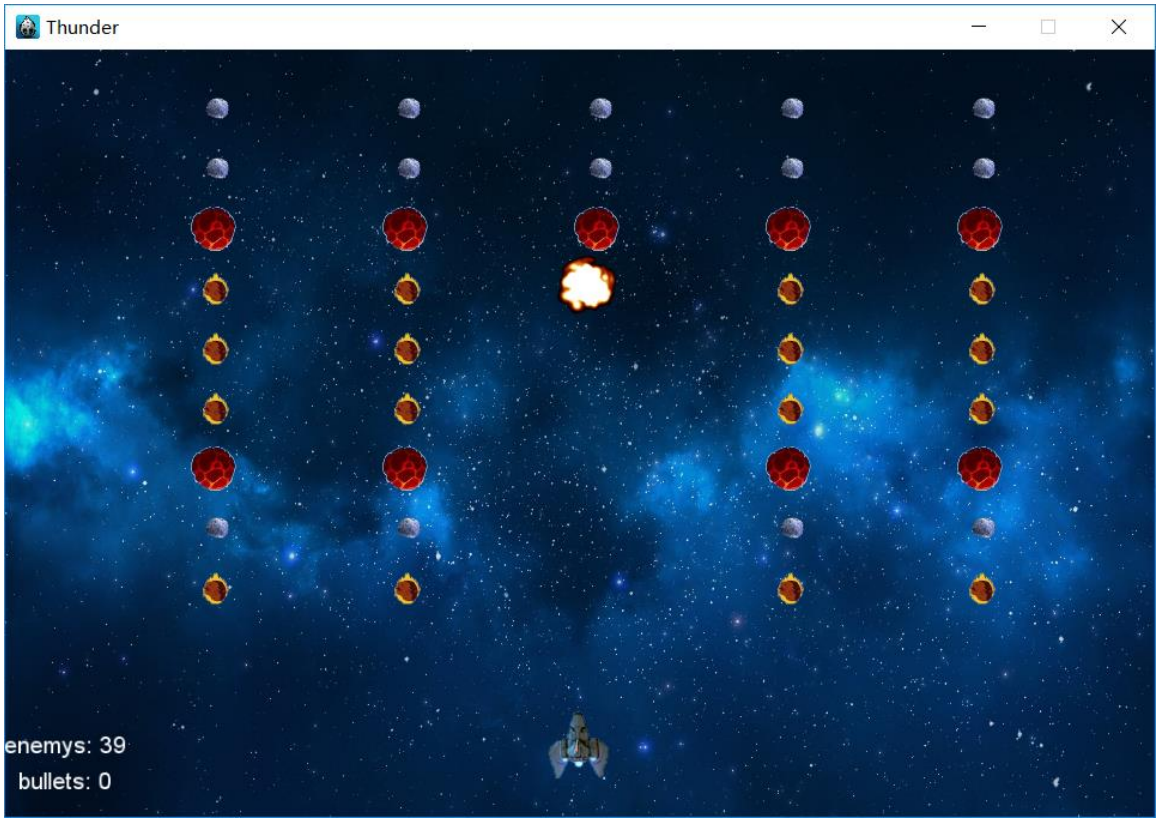
a) 游戏开始



b) 陨石下移



c) 击中陨石


Thunder
enemys: 39
bullets: 0

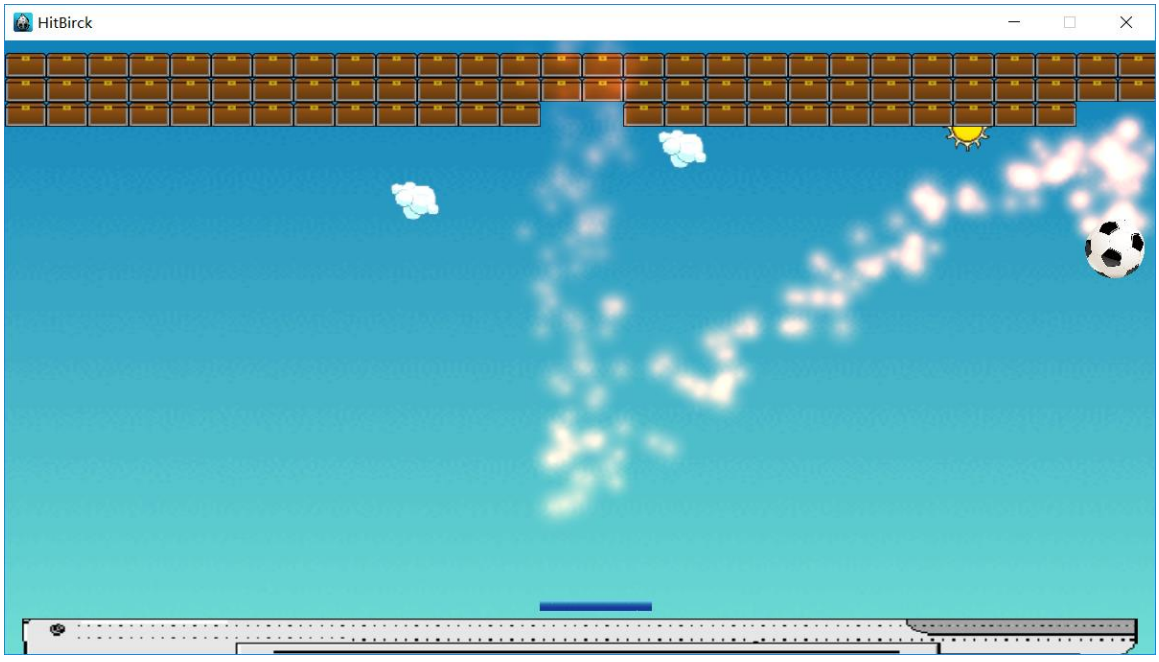
d) 游戏结束


Thunder
GAMEOVER
enemys: 54
bullets: 0

3. Hw14

a) 游戏开始

b) 游戏进行



c) 游戏结束

四、亮点与改进（可选）

1. Hw12

   a) 分别使用 UserDefault 和 SQLite 做本地数据存储

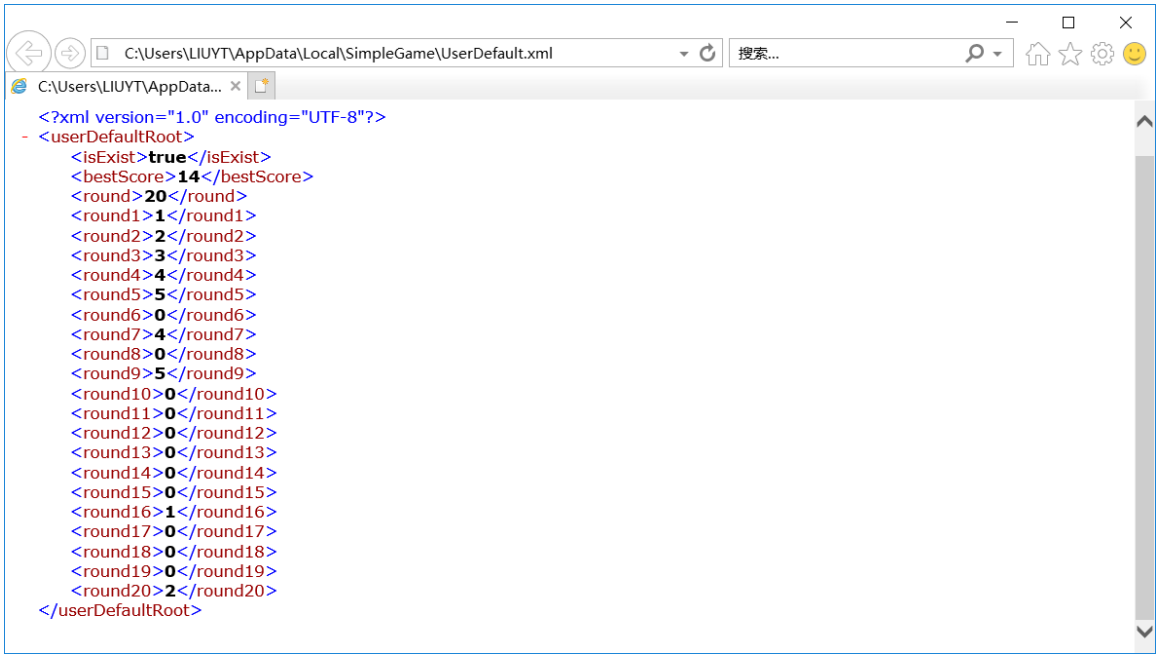
   （两个数据本地存储先完成 UserDefault 的本地存储，故两个显示的轮数不一样）

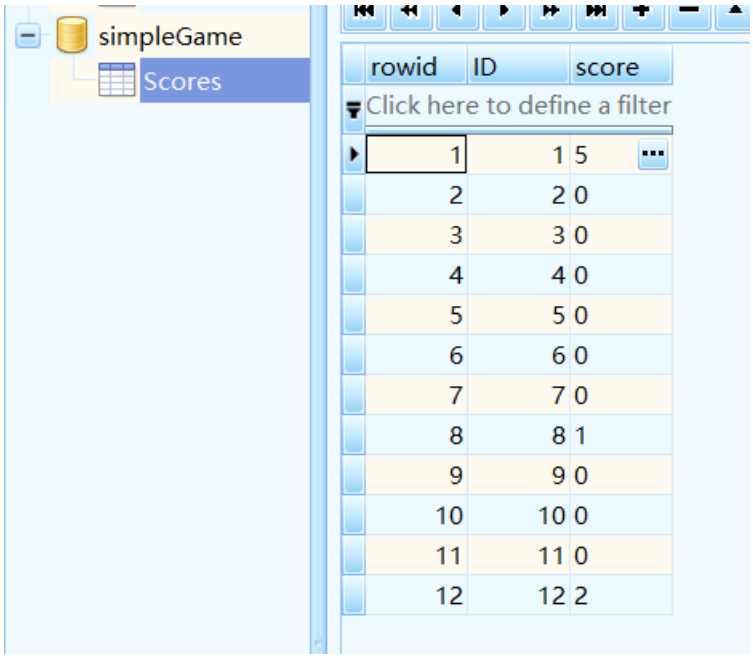
   实现主要代码：

```
1.  //存
2.  void HelloWorld::store() {
3.
4.      /* sqlite */
5.      char * errMsg = NULL;   //错误信息
6.      std::string sqlstr = " insert into Scores( score ) values ( '" + attackNum->
    getString() + "' ) ";
7.      sqlite3_exec(db, sqlstr.c_str(), NULL, NULL, &errMsg);
8.      //关闭数据库
9.      sqlite3_close(db);
10.
11.     /* userdefault */
12.     //检测 xml 文件是否存在（非必须）
13.     if(!database->getBoolForKey("isExist")) {
```

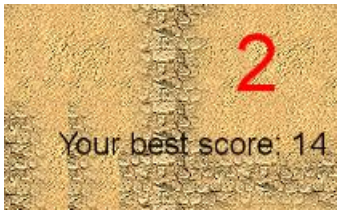
```
        database->setBoolForKey("isExist", true);
    }

    //存
    int kill = std::atoi(attackNum->getString().c_str());
    if(kill > database->getIntegerForKey("bestScore", 0))
        database->setIntegerForKey("bestScore", kill);

    int round = database->getIntegerForKey("round", 0) + 1;
    database->setIntegerForKey("round", round);

    std::string newKey = "round" + Value(round).asString();
    database->setIntegerForKey(newKey.c_str(), kill);
}

//取
int HelloWorld::get() {
    /* sqlite */

    /* userdefault */
    log("%s", FileUtils::getInstance()->getWritablePath().c_str());

    //检测 xml 文件是否存在（非必须）
    if (!database->getBoolForKey("isExist")) {
        database->setBoolForKey("isExist", true);
    }

    //取
    int bestScore = database->getIntegerForKey("bestScore", 0);

    return bestScore;
}
```

展示:





b) 游戏显示历史最高分



2. Hw13

a) 利用触摸事件实现飞船移动

需要将偏移的向量的 x 设置为 0，不然飞船跟着鼠标一同上下移动，不能再一条水平线上移动

```
// 当鼠标按住飞船后可控制飞船移动 (加分项)
void Thunder::onTouchMoved(Touch *touch, Event *event) {
    // Todo
    if (isClick) {
        Vec2 delta = touch->getDelta();
        delta.y = 0;
        if(player->getPosition().x + delta.x >= player->getContentSize().width &
& player->getPosition().x + delta.x <= visibleSize.width - player->getContentSiz
e().width)
            player->runAction(MoveBy::create(0.0, delta));
    }
}
```

b) 陨石向下移动并生成新的陨石

c) 子弹和陨石的数量显示正确

3. Hw14

a) 实现两种粒子效果

i. 足球上使用粒子系统 ParticleFire

```
fireBall->setPosition(ball->getPosition());
```

ii. 游戏结束时候使用粒子系统 ParticleExplosion

```
auto explosion = ParticleExplosion::create();
this->addChild(explosion, 4);
```

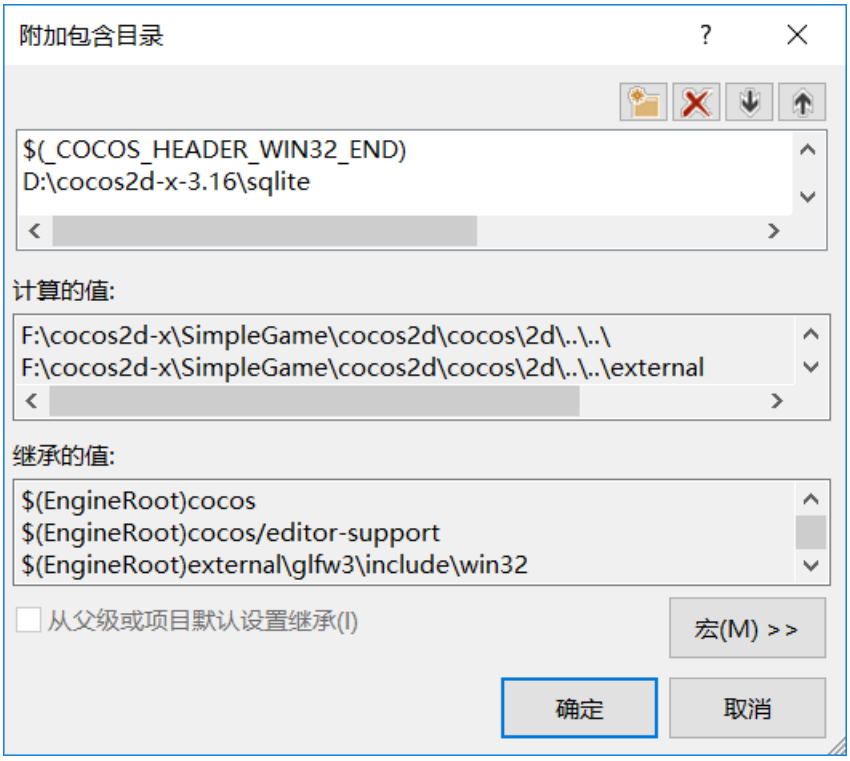
五、遇到的问题

1. Hw12

a) sqlite 的配置和使用

i. 将 sqlite 文件导入项目

右键工程->属性->配置属性->C/C++ ->常规->附加包含目录中增加相对应的目录



ii. Sqlite 数据库使用

b) UserDefault 的存储位置

查看源码:

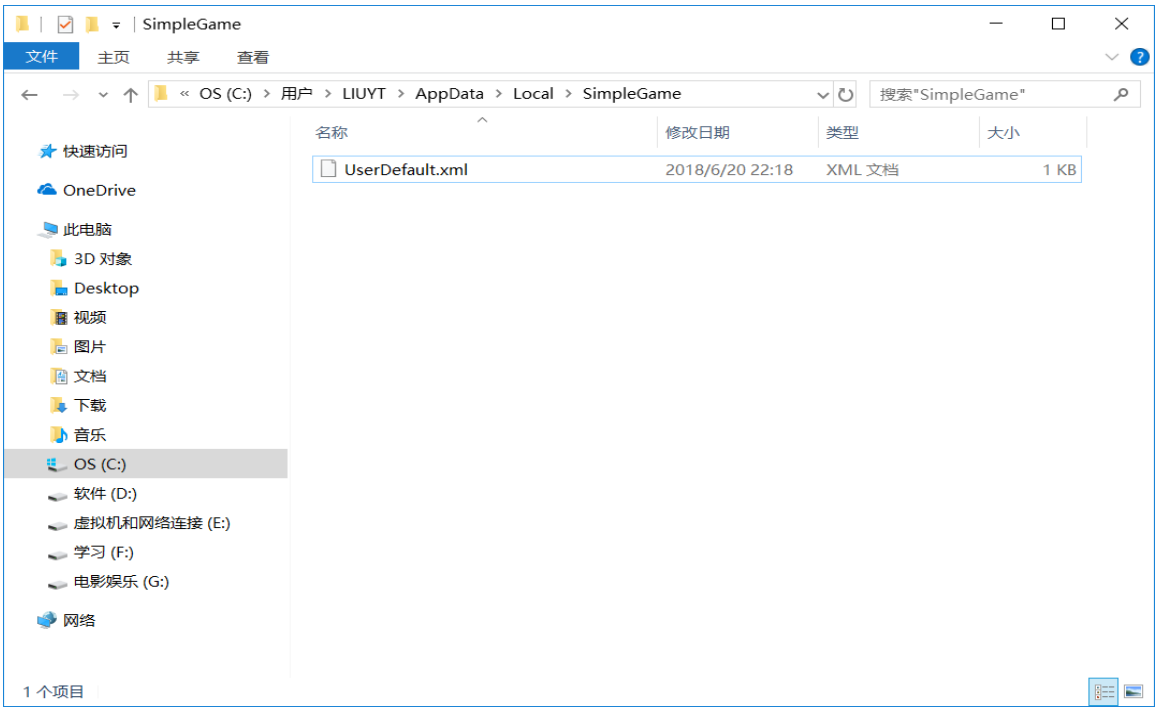
```
1.  void UserDefault::initXMLFilePath()
2.  {
3.      if (! _isFilePathInitialized)
4.      {
5.          _filePath += FileUtils::getInstance()->getWritablePath() + XML_FILE_NAME
    ;
6.          _isFilePathInitialized = true;
7.      }
8.  }
```

输出相对应的地址信息：

```
1.  log("%s", FileUtils::getInstance()->getWritablePath().c_str());
```

得到结果：

```
C:/Users/LIUYT/AppData/Local/SimpleGame/
```



2. hw13

   a) tilemap 的使用

3. hw14

   a) cocos 物理引擎

      网上没有比较规范的文档，很多完全靠不同版本的猜测或者看源码学习

六、思考与总结

作业对于 tilemap 的使用是挺基础的，感觉如果要对地图里面的对象层和地图层进行具体设计的话还是挺难的，感觉如果想做好一个地图好难，还有资源也好难找。这三周的作业总的来说感觉不是很难，cocos 的基础也基本学完了，但是第三次作业物理引擎与粒子系统这一部分写的比较头痛，很多东西没有说清楚，官方的文档对于函数有些就给个参数列表，没有任何的解释，只能自己去阅读源码和翻看可能相关的博客来学习。Cocos 的基础都学到差不读了，但是这半学期学的好像都比较浅，学的只是使用，感觉对于如何做一个完整的游戏还是有点迷茫，对于游戏的设计模式之类的也不是很清楚，只会填充现有的框架，接下来的大作业可能需要好好的构思一下了把。