现代操作系统应用开发实验报告

学号： 15331418 **班级 ：** 晚上班

姓名： 郑柏川 **实验名称：事件处理与音效**

一．参考资料

Ppt

二．实验步骤

1. 利用键盘事件实现飞船左右移动

```
// 添加键盘事件监听器
void Thunder::addKeyboardListener() {
    // Todo
    auto keyboardListener = EventListenerKeyboard::create();
    keyboardListener->onKeyPressed = CC_CALLBACK_2(Thunder::onKeyPressed, this);
    keyboardListener->onKeyReleased = CC_CALLBACK_2(Thunder::onKeyReleased, this);
    this->getEventDispatcher()->addEventListenerWithSceneGraphPriority(keyboardListener, this);
}

void Thunder::onKeyPressed(EventKeyboard::KeyCode code, Event* event) {
    switch (code) {
    case EventKeyboard::KeyCode::KEY_LEFT_ARROW:
    case EventKeyboard::KeyCode::KEY_CAPITAL_A:
    case EventKeyboard::KeyCode::KEY_A:
        movekey = 'A';
        isMove = true;
        break;
    case EventKeyboard::KeyCode::KEY_RIGHT_ARROW:
    case EventKeyboard::KeyCode::KEY_CAPITAL_D:
    case EventKeyboard::KeyCode::KEY_D:
        movekey = 'D';
        isMove = true;
        break;
    case EventKeyboard::KeyCode::KEY_SPACE:
        fire();
        break;
    }
}
```

```
void Thunder::onKeyReleased(EventKeyboard::KeyCode code, Event* event) {
  switch (code) {
    case EventKeyboard::KeyCode::KEY_LEFT_ARROW:
    case EventKeyboard::KeyCode::KEY_A:
    case EventKeyboard::KeyCode::KEY_CAPITAL_A:
    case EventKeyboard::KeyCode::KEY_RIGHT_ARROW:
    case EventKeyboard::KeyCode::KEY_D:
    case EventKeyboard::KeyCode::KEY_CAPITAL_D:
      isMove = false;
      break;
    }
}
```

2. 利用键盘和触摸事件实现子弹发射

```
//发射子弹
void Thunder::fire() {
  auto bullet = Sprite::create("bullet.png");
  bullet->setAnchorPoint(Vec2(0.5, 0.5));
  bullets.push_back(bullet);
  bullet->setPosition(player->getPosition());
  addChild(bullet, 1);
  SimpleAudioEngine::getInstance()->playEffect("music/fire.wav", false);

  // 移除飞出屏幕外的子弹
  // Todo
  //bullet->runAction(MoveBy::create(1.0f, Vec2(0, visibleSize.height)));
  bullet->runAction(Sequence::create(MoveBy::create(1.0f, Vec2(0, visibleSize.height)), CallFunc::create([this, bullet]() {
      bullet->removeFromParentAndCleanup(true);
      this->bullets.remove(bullet);
  }), nullptr));
}
```

让子弹向上走到边界并 remove

3. 用自定义事件实现：子弹和陨石相距小于一定距离时，陨 石爆炸，子弹消失。
游戏结束飞船爆炸，移除所有监听器

```
// 判断游戏是否结束并执行对应操作
// Todo
  for (auto it = enemys.begin(); it != enemys.end(); ++it) { //判断游戏是否结束
      if (*it && (*it)->getBoundingBox().getMinY() <= player->getBoundingBox().getMaxY()) {
          temp = player;
          player->runAction(Sequence::create(Animate::create(Animation::createWithSpriteFrames(explore, 0.05f, 1)), CallFunc::create([this, temp] {
              temp->removeFromParentAndCleanup(true);
              auto gameover = Sprite::create("gameOver.png");
              gameover->setAnchorPoint(Vec2(0.5, 0.5));
              gameover->setPosition(visibleSize / 2);
              this->addChild(gameover, 1);
          }), nullptr));

          SimpleAudioEngine::getInstance()->playEffect("music/explore.wav", false);
          unschedule(schedule_selector(Thunder::update));
          this->getEventDispatcher()->removeAllEventListeners();
          return; // 游戏结束要退出循环，player已经是空指针，不能player->getBoundingBox()
      }
  }
```

```
void Thunder::meet(EventCustom * event) {
  // 判断子弹是否打中陨石并执行对应操作
  // Todo
    Sprite* temp;
    bool meet = false;
    for (auto bullet = bullets.begin(); bullet != bullets.end();) {
        meet = false;
        for (auto enemy = enemys.begin(); enemy != enemys.end(); ++ enemy) {
            if ((*bullet)->getPosition().getDistance((*enemy)->getPosition()) < 25 && (*bullet) && (*enemy)) {
                temp = *enemy;
                (*bullet)->removeFromParentAndCleanup(true);
                (*enemy)->runAction(Sequence::create(Animate::create(Animation::createWithSpriteFrames(explore, 0.05f, 1)), CallFunc::create([this, temp] {
                    temp->removeFromParentAndCleanup(true);
                }), nullptr));
                SimpleAudioEngine::getInstance()->playEffect("music/explore.wav", false);
                meet = true;
                bullet = bullets.erase(bullet);
                enemy = enemys.erase(enemy);
                break;
            }
        }
        if (!meet) ++bullet;
    }
```

遍历 enemy 跟子弹，如果有 position 距离小于 25 则说明 meet 了，另外如果游戏结束则移除所有监听事件

4. 游戏过程中有背景音乐，发射子弹、击中陨石有音效

```
//预加载音乐文件
void Thunder::preloadMusic() {
  // Todo
    auto audio = SimpleAudioEngine::getInstance();
    audio->preloadBackgroundMusic("music/bgm.mp3");
    audio->preloadEffect("music/explore.wav");
    audio->preloadEffect("music/fire.wav");
}

//播放背景音乐
void Thunder::playBgm() {
  // Todo
    SimpleAudioEngine::getInstance()->playBackgroundMusic("music/bgm.mp3", true);
}
  SimpleAudioEngine::getInstance()->playEffect("music/fire.wav", false);
```

5. 加分项：

利用触摸事件实现飞船移动。（点击飞船后拖动鼠标）

```
// 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) {
        float x = touch->getDelta().x + player->getPositionX();
        if (x < 0) x = 0;
        if (x > visibleSize.width) x = visibleSize.width;
        player->setPosition(x, player->getPositionY());
    }
}
```

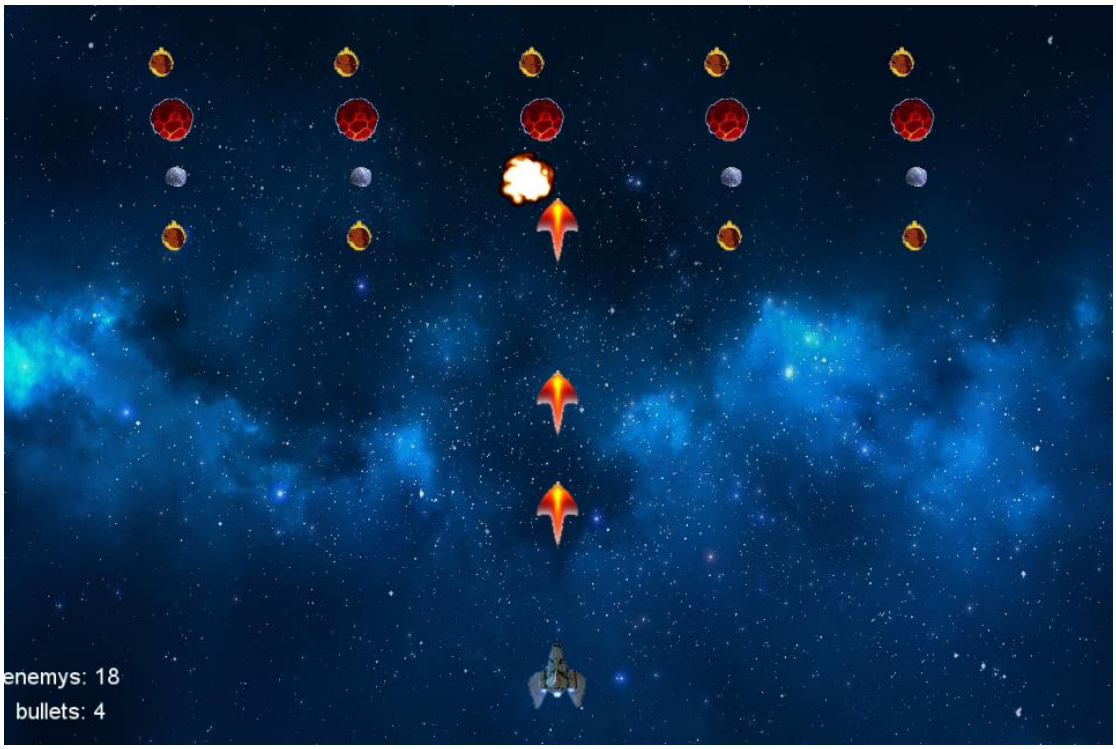
陨石向下移动并生成新的一行陨石

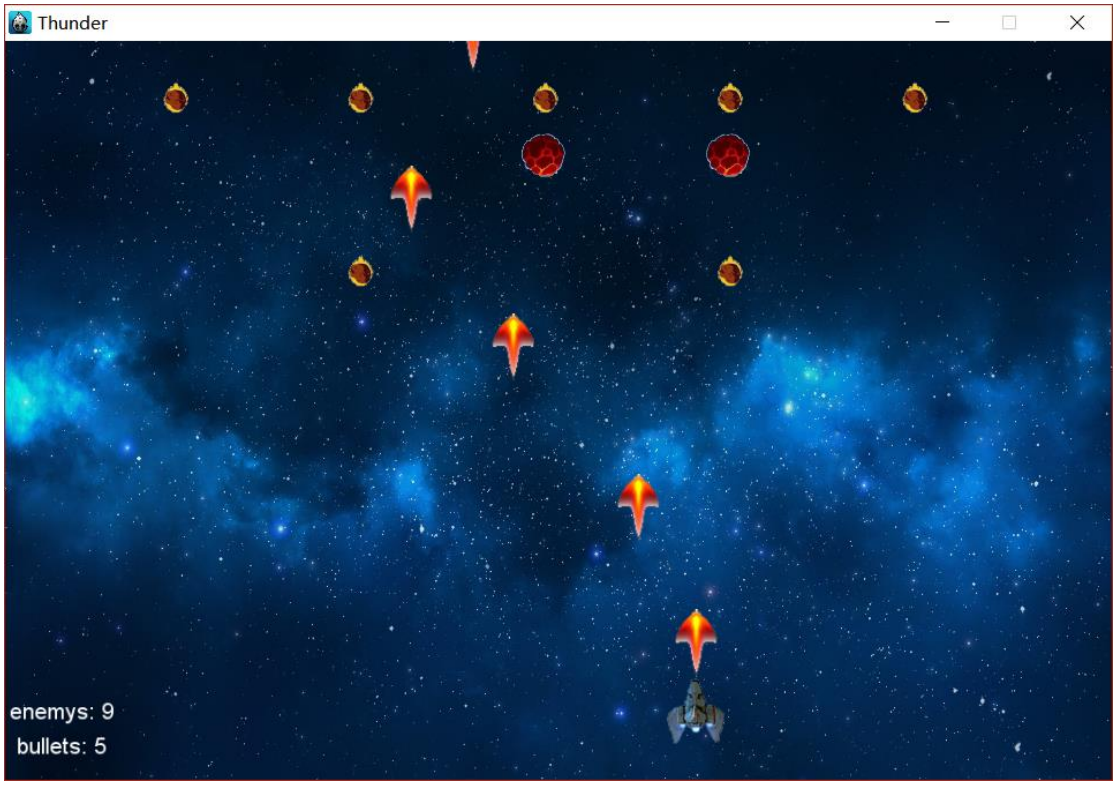
```
for (int j = 0; j < 5; ++j) {
    auto enemy = Sprite::create(enemyPath);
    enemy->setAnchorPoint(Vec2(0.5, 0.5));
    enemy->setScale(0.5, 0.5);
    enemy->setPosition(width * j + 65, height)
    enemys.push_back(enemy);
    addChild(enemy, 1);
}
stone = stone % 3 + 1;
```

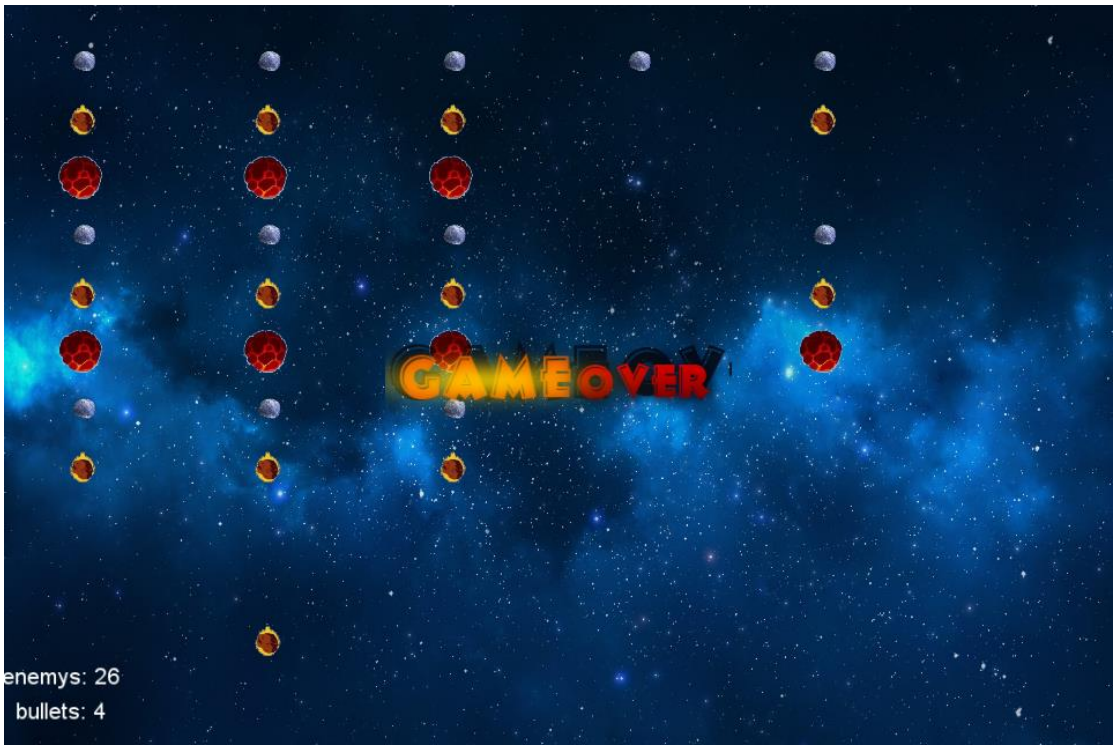
将本来的向下移，然后在最顶生成五个 stone

三．实验结果截图


enemys: 0
bullets: 4


enemys: 18
bullets: 4





四．实验过程遇到的问题

1. **在生成石头时怎么挨个生成**

   解决：用数组存 stone1.png，这样的格式再遍历创建

五．思考与总结

本次作业较为简单，主要是看起来难度很大，但是关键的代码已实现，而且发射子弹，判断是否打中以及游戏结束的逻辑较简单，所以写起来很快