期末project案例-PlaneWar

by 刘导安

作者声明

本人不玩游戏，想象力有限，想不出有创意的案例;本人代码粗糙，使用cocos2dx经验欠缺，代码使用方式比较暴力，案例中有不少BUG，各路大神轻吐槽;本人艺术细胞匮乏，UI水平低下，场景布置简陋，莫要见怪。

PlaneWar简介

无数外太空敌机凶悍入侵地球，英勇雄壮的你驾驶战斗机与数万万敌机输死空战，输则遗臭万年，赢则万古流芳。

场景介绍·开始菜单

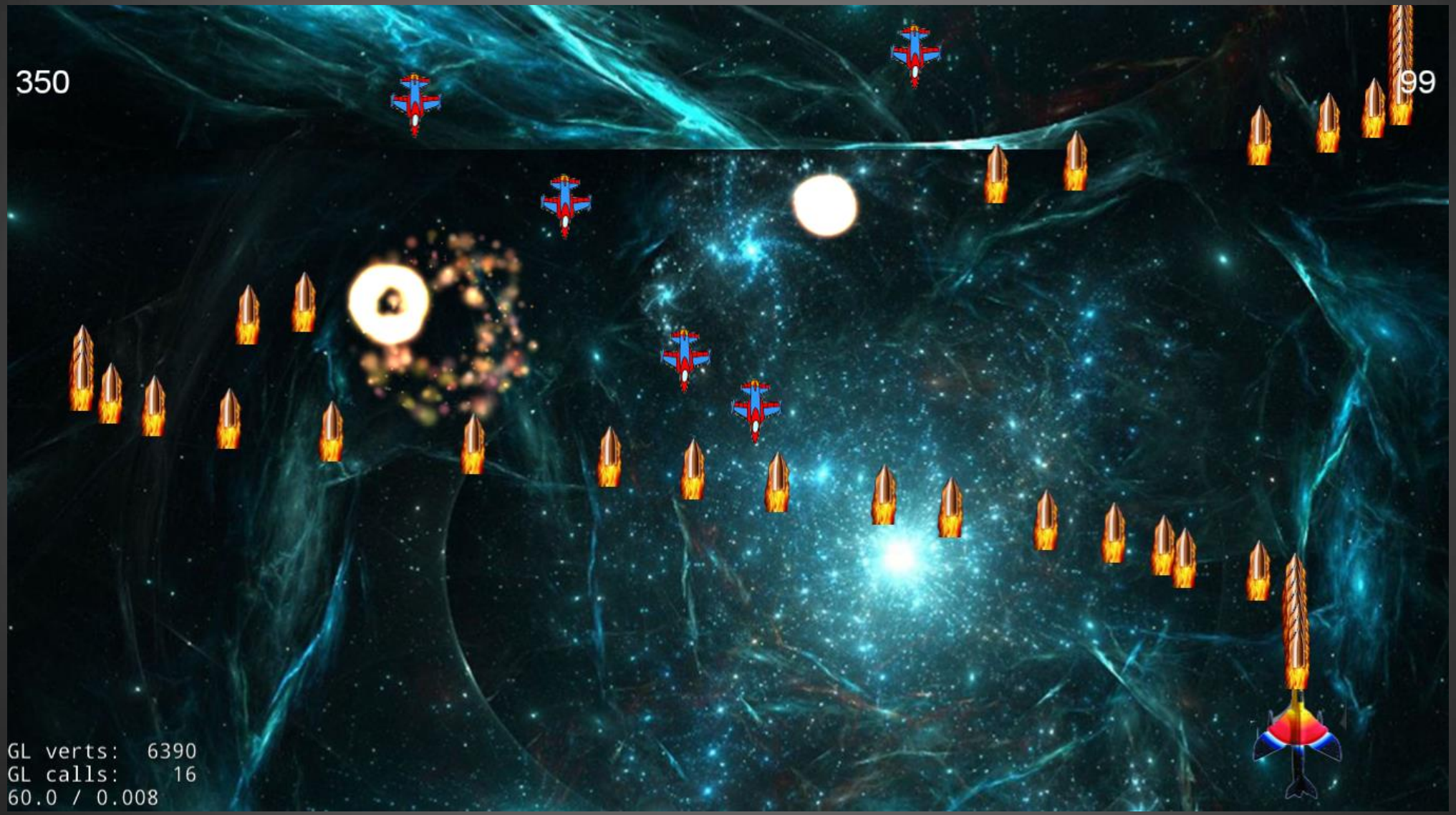
场景介绍·帮助界面


返回菜单
Attention
通过鼠标(或触屏)可左右移动战舰，右上角显示分数，两种敌机，A敌机消灭一辆得10分，B敌机消灭一辆得15分，被碰撞最多不超过三次，右上角显示当前战舰能量，能量为0时，战争结束，分数达到1000时，战斗胜利。
GL verts: 72
GL calls: 4
59.9 / 0.003

场景介绍·退出界面

“确定”按钮未实现，交给众位大神了

场景介绍·游戏界面



场景介绍·失败界面

场景介绍·成功界面


你成功的保护了家园，成为了拯救宇宙的英雄。
回到原点
查看排行
GL verts: 24
GL calls: 4
59.8 / 0.003

场景介绍·排行榜

第一名：1030
第二名：445
第三名：100
第四名：100
第五名：100
第六名：100
第七名：23
第八名：1
第九名：0
第十名：0

你的分数是：1030

游戏实现

```
//鼠标事件响应
bool onTouchBegan(Touch *touch, Event *unused_event);
void onTouchMoved(Touch *touch, Event *unused_event);
void onTouchEnded(Touch *touch, Event *unused_event);
void backgroundChange(float t);
//发射子弹
void shooting(float t);
//控制子弹移动
void bulletMoving(float t);
//背景移动
void backgroundMoving(float t);
//敌机移动
void enemysMoving(float t);
```

```
//新产生一个敌机
void newEnemy(float t);
//控制子弹，敌机，出现和移动速度
void Contr1Speed(float t);
//死亡
void GameOver();
//获取胜利
void WinGame();
//重启所有计划任务
void restartSchedule();
//碰撞检测
void collisionDetecting(float t);
```

发射子弹

- 每间隔指定时间发射一颗子弹

子弹移动

- 子弹的速度控制（每次移动时间间隔，单次移动距离）
- 子弹销毁（移动到屏幕外的子弹要销毁）

背景移动

- 敌机不动
- 我机不动
- 背景移动

敌机移动

- 移动频率，移动距离
- 移动到屏幕屏幕外销毁并使玩家能量值减少

敌机生成

- 敌机产生频率控制
- 位置随机
- 种类随机

控制速度

- 产生子弹的速度
- 产生敌机的速度
- 子弹移动的速度
- 敌机移动的速度

失败&成功

- 能量值为0则失败（用于死亡检测）
- 分数达到指定值（比如5000）则成功（用于成功检测）

```
if (energy <= 0){
  GameOver();
}
```

```
if (score > 5000){
  WinGame();
}
```

重启调度

- 速度改变（子弹和敌机的产生和移动速度）引起调度的销毁与重启

```
void Game::restartSchedule(){
    this->unschedule(schedule_selector(Game::shooting));
    this->unschedule(schedule_selector(Game::newEnemy));
    this->unschedule(schedule_selector(Game::bulletMoving));
    this->unschedule(schedule_selector(Game::enemysMoving));

    //this->resumeSchedulerAndActions();
    this->schedule(schedule_selector(Game::shooting), newBulletTime);
    this->schedule(schedule_selector(Game::newEnemy), newEnemyTime);
    this->schedule(schedule_selector(Game::bulletMoving), bulletSpeed);
    this->schedule(schedule_selector(Game::enemysMoving), enemySpeed);
}
```

碰撞检测

- 敌机与我机碰撞
- 敌机与子弹的碰撞
- 碰撞时执行事件（播放音效，粒子效果，分数增加/能量减少，成功检测，失败检测）

涉及知识点

- 触摸移动
- 调度
- 音乐&音效
- 粒子
- 网络
- 动画
- 碰撞检测等

具体实现，请在课程网站上下载期末案例

触摸移动

```
//鼠标事件响应
bool onTouchBegan(Touch *touch, Event *unused_event);
void onTouchMoved(Touch *touch, Event *unused_event);
void onTouchEnded(Touch *touch, Event *unused_event);
```

```
auto listner = EventListenerTouchOneByOne::create();
listner->onTouchBegan = CC_CALLBACK_2(Game::onTouchBegan,this);//触摸开始
listner->onTouchMoved = CC_CALLBACK_2(Game::onTouchMoved,this);//触摸过程
listner->onTouchEnded = CC_CALLBACK_2(Game::onTouchEnded,this);//触摸结束
_eventDispatcher->addEventListenerWithSceneGraphPriority(listner, this);
```

触摸移动

```
bool Game::onTouchBegan(Touch *touch, Event *unused_event){
    return true;
}
void Game::onTouchMoved(Touch *touch, Event *unused_event){
    int x = touch->getLocation().x;
    if (x < planeWidth / 2)
        x = planeWidth / 2;
    if (x > visibleSize.width - planeWidth / 2)
        x = visibleSize.width - planeWidth / 2;
    //战机移动
    plane->setPosition(Point(x, plane->getPosition().y));
}

void Game::onTouchEnded(Touch *touch, Event *unused_event){}
```

调度

```
//发射子弹
void shooting(float t);
```

```
//每隔newBulletTime时间，执行一次shooting函数
this->schedule(schedule_selector(Game::shooting), newBulletTime);
```

音乐&音效

```
#include "SimpleAudioEngine.h"
using namespace CocosDenshion;
```

```
SimpleAudioEngine::getInstance()->playBackgroundMusic("game.wav",true);
```

```
SimpleAudioEngine::getInstance()->playEffect("bomb.wav");
```

碰撞检测

```
for (int j = 0; j < enemys.size(); j++){
  auto enemy = enemys.at(j);//敌机
  Rect er(enemy->getPositionX(), enemy->getPositionY(),
    enemy->getContentSize().width, enemy->getContentSize().height);//获取敌机位置范围
  for (int i = 0; i < bullets.size(); i++){
    auto bullet = bullets.at(i);//子弹
    Rect br(bullet->getPositionX(), bullet->getPositionY(),
      bullet->getContentSize().width, bullet->getContentSize().height);//获取子弹位置范围
    if (br.intersectsRect(er)){//判断子弹位置范围(br)与敌机位置范围(er)是否有交集
      auto ps = ParticleSystemQuad::create("explo.plist");//粒子效果使用
```

动画

```
//0.5秒内旋转角度1800度
auto act = RotateBy::create(0.5, 1800);
//rankboard是Sprite，act->reverse()指逆向动作
rankboard->runAction(Sequence::create(act, act->reverse(), NULL));
```

Cocos2dx-网络

```
//服务器响应完成后的操作函数
void onHttpRequestCompleted(HttpClient *sender, HttpResponse *response);
//将分数post给服务器
void postData();
//从服务器中get排行榜
void getRank();
```

Cocos2dx-网络

Win8第七周：

```
<records>
  <item>
    <user>mvp</user>
    <score>1030</score>
    <words>aaaa</words>
  </item>
  <item>
    <user>kkkk</user>
    <score>445</score>
    <words/>
  </item>
  <item>
    <user>yzkk</user>
    <score>100</score>
    <words>just a test</words>
  </item>
  <item>
    <user>planewar</user>
    <score>100</score>
    <words>aaaa</words>
  </item>
```

Cocos2dx-网络

·POST:

```
void Win::postData(){
    HttpRequest* request = new HttpRequest();//创建请求
    request->setUrl("http://172.18.187.83:8888/rank/newScore/");//设置URL
    request->setRequestType(HttpRequest::Type::POST);//选择post方式传输数据
    stringstream sts;
    sts << score;
    //ch是请求的表单参数及其值,id指游戏ID，key:游戏KEY,(有需要的同学可以向TA申请游戏ID和KEY，见最后一页PPT)
    string ch = "id=404&user=mvp&score=" + sts.str() + "&words=aaaa&key=f213503c49194e46378da174b765b598";
    const char* postData = ch.c_str();
    request->setRequestData(postData, strlen(postData));//设置请求数据
    request->setTag("POST test");
    cocos2d::network::HttpClient::getInstance()->send(request);//发送请求
    request->release();//释放请求
}
```

Cocos2dx-网络

- GET:

```
void Fail::getRank(){
    HttpRequest* request = new HttpRequest();//创建请求
    request->setUrl("http://172.18.187.83:8888/rank/getScore/?id=404&num=10");//设置URL,注意参数
    request->setRequestType(HttpRequest::Type::GET);//设置请求方式为GET
    //设置服务器响应完毕后的回调函数
    request->setResponseCallback(CC_CALLBACK_2(Fail::onHttpRequestCompleted, this));
    request->setTag("GET test");
    cocos2d::network::HttpClient::getInstance()->send(request);//发送请求
    request->release();//释放请求
}
```

Cocos2dx-网络

- 获取响应信息:

```
void Fail::onHttpRequestCompleted(HttpClient *sender, HttpResponse *response)
{
    if (!response)
    {
        return;
    }
    if (0 != strlen(response->getHttpRequest()->getTag()))//获取请求Tag
    {
        log("%s completed", response->getHttpRequest()->getTag());
    }
    int statusCode = response->getResponseCode();//获取状态码
    char statusString[64] = {};
    if (!response->isSucceed())
    {
        log("response failed");
        log("error buffer: %s", response->getErrorBuffer());//获取错误信息
        return;
    }
    //获取请求的响应数据（即用户需要的数据），数据放在了vector<char>里面
    std::vector<char> *buffer = response->getResponseData();
```

期末Project

项目要求：

小组用cocos2d-x写一个游戏，形式不限，移动平台有加分

deadline：18周周日之前

提交要求：

待定.....

注意：

- 需要使用TA搭建的服务器的小组，请组长发送邮件到343802694@qq.com，邮件主题：组长学号_组长姓名_申请游戏ID和KEY