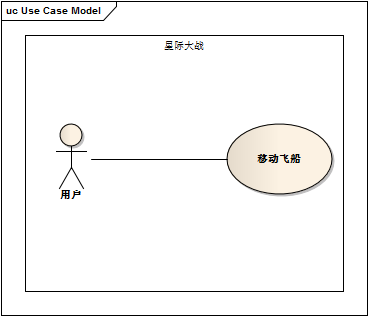
Tizen课程项目报告  
《雷霆战机》

## 项目简介

《雷霆战机》是一款画面绚丽的竖屏飞行射击大作。在这个次元的未来世界里，人类与外星文明接触，不同文明的碰撞带来的只有是前所未有的战争。世界末日？玩家要做的就是驾驶超级战机与这些外星入侵者决一死战。

## 需求分析



## 概要设计

雷霆战机基于TIZEN web project开发，主要使用了Html与Javascript技术。通过在ontouchstart、ontouchmove和ontouchend方法中计算手指触碰位移来判断手指滑动方向，从而进行移动飞船操作。

|  |  |
| --- | --- |
| 接口 | 接口功能 |
| HitTest | 子弹碰撞检测 |
| SetBullet | 发射子弹 |
| GetPlainStatus | 获取飞机状态 |
| GameInit | 初始化游戏 |
| AddEnemy | 增加敌人 |

## 核心算法

雷霆战机核心算法代码如下图所示。

|  |
| --- |
| /\*\*  \* 子弹类  \* \*/  function Bullet(belong,x,y,angle,xspeed,yspeed,aspeed,speed){  base(this,LSprite,[]);  var self = this;  //子弹所属  self.belong = belong;  //出现位置  self.x = x;  self.y = y;  //角度  self.angle = angle;  //移动速度  self.speed = speed;  //xy轴速度  self.xspeed = xspeed;  self.yspeed = yspeed;  //旋转角度加成  self.aspeed = aspeed;  //子弹图片  var bitmapdata,bitmap;  bitmapdata = new LBitmapData(imglist["item1"]);  bitmap = new LBitmap(bitmapdata);  self.bitmap = bitmap;  //显示  self.addChild(bitmap);  }  /\*\*  \* 循环  \* @param 子弹序号  \* \*/  Bullet.prototype.onframe = function (index){  var self = this;  //子弹移动  self.x += self.xspeed;  self.y += self.yspeed;    //子弹角度变更  if(self.aspeed != 0){  self.angle += self.aspeed;  //子弹角度变更后，重新计算xy轴速度  self.xspeed = self.speed\*Math.sin(self.angle \* Math.PI / 180);  self.yspeed = self.speed\*Math.cos(self.angle \* Math.PI / 180);  }  //子弹位置检测  if(self.x < 0 || self.x > LGlobal.width || self.y < 0 || self.y > LGlobal.height){  //从屏幕移除  backLayer.removeChild(self);  //从子弹数组移除  barrage.splice(index,1);  }else{  self.hitTest(index);  }    };  /\*\*  \* 子弹碰撞检测  \* @param 子弹序号  \* \*/  Bullet.prototype.hitTest = function (index){  var self = this;  var disx,disy,sw,ew,obj,i;  if(self.belong == player.belong){  //自机子弹  for(i=0;i<enemys.length;i++){  obj = enemys[i];  sw = self.bitmap.getWidth()/2;  ew = obj.bitmap.getWidth()/2;  disx = self.x+sw - (obj.x + ew);  disy = self.y+self.bitmap.getHeight()/2 - (obj.y + obj.bitmap.getHeight()/2);  //距离检测  if(disx\*disx + disy\*disy < ew\*ew){  obj.hp--;  if(obj.hp == 0){  point += 1;  pointText.text = point;  //从屏幕移除  backLayer.removeChild(obj);  //从敌机数组移除  enemys.splice(i,1);  if(obj.name == "boss"){  gameclear = true;  }  }  //从屏幕移除  backLayer.removeChild(self);  //从子弹数组移除  barrage.splice(index,1);  }  }  }else{  //敌机子弹  obj = player;  sw = self.bitmap.getWidth()/2;  ew = obj.bitmap.getWidth()/2;  disx = self.x+sw - (obj.x + ew);  disy = self.y+self.bitmap.getHeight()/2 - (obj.y + obj.bitmap.getHeight()/2);  //距离检测  if(disx\*disx + disy\*disy < ew\*ew - 10){ obj.hp--; if(obj.hp <=0){  obj.visible = false;  gameover = true; }  //从屏幕移除  backLayer.removeChild(self);  //从子弹数组移除  barrage.splice(index,1);  }  }  };  /\*\*  \* Main  \* \*/  //设定游戏速度，屏幕大小，回调函数  init(20,"mylegend",480,800,main);  /\*\*层变量\*/  //显示进度条所用层  var loadingLayer;  //游戏最底层  var backLayer;  //控制层  var ctrlLayer;  /\*\*int变量\*/  //读取图片位置  var loadIndex = 0;  //贞数  var frames = 0;  //BOOS START  var boosstart = false;  //GAME OVER  var gameover = false;  //GAME CLEAR  var gameclear = false;  //得分  var point = 0;  /\*\*对象变量\*/  //玩家  var player;  //得分  var pointText;  /\*\*数组变量\*/  //图片path数组  var imgData = new Array();  //读取完的图片数组  var imglist = {};  //子弹数组  var barrage = new Array();  //子弹速度数组  var barrageSpeed = [5,10];  //储存所有敌人飞机的数组  var enemys = new Array();  function main(){  LMouseEventContainer.set(LMouseEvent.MOUSE\_DOWN,true);  LMouseEventContainer.set(LMouseEvent.MOUSE\_UP,true);  LMouseEventContainer.set(LMouseEvent.MOUSE\_MOVE,true);  //准备读取图片  imgData.push({type:"js",path:"./js/Global.js"});  imgData.push({type:"js",path:"./js/Bullet.js"});  imgData.push({type:"js",path:"./js/Plain.js"});  imgData.push({name:"back",path:"./images/back.jpg"});  imgData.push({name:"enemy",path:"./images/e.png"});  imgData.push({name:"player",path:"./images/player.png"});  imgData.push({name:"boss",path:"./images/boss.png"});  imgData.push({name:"ctrl",path:"./images/ctrl.png"});  imgData.push({name:"item1",path:"./images/1.png"});    loadingLayer = new LoadingSample1();  addChild(loadingLayer);  LLoadManage.load(  imgData,  function(progress){  loadingLayer.setProgress(progress);  },  function(result){  imglist = result;  removeChild(loadingLayer);  loadingLayer = null;  gameInit();  }  );  }  function gameInit(event){  //游戏底层实例化  backLayer = new LSprite();  addChild(backLayer);  ctrlLayer = new LSprite();  addChild(ctrlLayer);  //添加游戏背景  bitmapdata = new LBitmapData(imglist["back"]);  bitmap = new LBitmap(bitmapdata);  backLayer.addChild(bitmap);  /\*  //添加控制按钮  bitmapdata = new LBitmapData(imglist["ctrl"]);  bitmap = new LBitmap(bitmapdata);  ctrlLayer.addChild(bitmap);  ctrlLayer.x = 90;  ctrlLayer.y = 450;\*/    //得分显示  pointText = new LTextField();  pointText.color = "#ffffff";  pointText.size = 20;  pointText.text = point;  backLayer.addChild(pointText);    //加入玩家  player = new Plain("player",0,200,600,[5]);  backLayer.addChild(player);    //添加贞事件，开始游戏循环  backLayer.addEventListener(LEvent.ENTER\_FRAME,onframe);    //添加控制事件  backLayer.addEventListener(LMouseEvent.MOUSE\_DOWN,ondown);  backLayer.addEventListener(LMouseEvent.MOUSE\_UP,onup); /\*  if(!LGlobal.canTouch){  //电脑的时候，添加键盘事件 【上 下 左 右 空格】  LEvent.addEventListener(LGlobal.window,LKeyboardEvent.KEY\_DOWN,onkeydown);  LEvent.addEventListener(LGlobal.window,LKeyboardEvent.KEY\_UP,onkeyup);  } \*/  }  var monseIsDown = false;  function onup(event){  monseIsDown = false;  player.move = [0,0];  player.canshoot = false;  player.shootctrl = player.shootspeed;  }  function ondown(event){  monseIsDown = true;  player.shootcount = 0;  player.shootctrl = player.shootspeed+1;  player.canshoot = true;  }  /\*\*  \* 循环  \* \*/  function onframe(){  if(gameover){//游戏结束  backLayer.die();  var txtOver = new LTextField();  txtOver.text = "GAME OVER";  txtOver.color = "#ffffff";  txtOver.x = 100;  txtOver.y = 200;  txtOver.size = 40;  backLayer.addChild(txtOver);  }else if(gameclear){//游戏通关  backLayer.die();  var txtOver = new LTextField();  txtOver.text = "GAME CLEAR";  txtOver.color = "#ffffff";  txtOver.x = 100;  txtOver.y = 200;  txtOver.size = 40;  backLayer.addChild(txtOver);  }  if(monseIsDown){  if(player.x + 30 - player.speed > LGlobal.offsetX){//left  player.move[0] = -1;  }else if(player.x + 30 + player.speed < LGlobal.offsetX){//right  player.move[0] = 1;  }else{  player.move[0] = 0;  }  if(player.y + 30 - player.speed > LGlobal.offsetY){//up  player.move[1] = -1;  }else if(player.y + 30 + player.speed < LGlobal.offsetY){//down  player.move[1] = 1;  }else{  player.move[1] = 0;  }  }  var i;  //循环子弹  for(i=0;i<barrage.length;i++){  barrage[i].onframe(i);  }  //循环敌机  for(i=0;i<enemys.length;i++){  enemys[i].onframe();  }  //自机循环  player.onframe();  //添加敌机  addEnemy();  }  /\*\*  \* 添加敌机  \* \*/  function addEnemy(){  if(boosstart)return;  var plain;  if(point >= 10){//得到10分的话，添加boss  //加入一个boss敌人  plain = new Plain("boss",1,100,0,[2,3,4]);  plain.move = [0,1];  enemys.push(plain);  backLayer.addChild(plain);  boosstart = true;  return;  }  if(frames++ % 100 > 0)return;//限制敌人出现频率  var rand = Math.random();  var b;  if(rand < 0.5){  if(rand < 0.3){  b=0;  }else{  b=1;  }  //左边加入一个敌人  plain = new Plain("enemy",1,0,100\*Math.random(),[b]);  plain.move = [0.6,1];  }else{  if(rand < 0.8){  b=0;  }else{  b=1;  }  //右边加入一个敌人  plain = new Plain("enemy",1,520,100\*Math.random(),[b]);  plain.move = [-0.6,1];  }  enemys.push(plain);  backLayer.addChild(plain);  } |

## 软件功能界面

