Tizen课程项目报告  
《Keep Hiding》

## 项目简介

### 游戏背景

以冬天和圣诞节为背景，雪人在冰天雪地中间，漫天飞舞着不同颜色的雪花，不同的雪花有不同的效果，有减小雪花飘落速度的，有冰冻小人移动的，也有静止雪花飘落的，也有直接导致雪人死亡的雪花，雪人在雪花的世界躲避一些对自己产生不好效果的雪花，锻炼了控制能力和反应能力，以坚持的时间长短为得分标准。

### 游戏玩法

竖屏游戏，玩家控制一个雪人在屏幕底端可以左右移动，屏幕上半部分随机出现不同颜色的雪花，雪花出现后即开始移动，雪花与屏幕边框相撞后会发生移动方向变化，具体的变化与碰撞角度有关，符合动量碰撞原理。不同颜色的雪花与雪人相碰后会产生不同的效果，雪球颜色不同对雪人产生的效果不同，会根据时间不断变换颜色，但是不会因为变换颜色而导致游戏结束。屏幕下方有按钮和计时器，游戏根据玩家坚持的时间进行分数计算。屏幕底端有两个按钮，分别控制了雪人向右或向左移动，也可以通过滑动屏幕的方式进行控制。

### 游戏logo



## 需求分析



图1 用例图

## 概要设计

Keep Hiding基于TIZEN web project开发，主要使用了Html与Javascript技术。通过使用两个计时器setInterval(ballMove, ball)和setInterval(ballCreate, add)控制礼物炸弹的移动和产生，礼物炸弹每2000ms产生随机速度（有上限和下限）、随机位置（屏幕上30%）和产生随机类型。采用面向对象的设计思想，分别创建礼物炸弹和小人两个类，如下表1所示：

表1 类的设计

|  |  |
| --- | --- |
| 类 | 功能 |
| Ball | 包含礼物炸弹的初始化函数，礼物炸弹的重绘函数 |
| Person | 包含小人的初始化函数和小人移动时的重绘函数 |

函数功能如下表2所示：

表2 函数及其功能介绍

|  |  |
| --- | --- |
| 函数 | 功能 |
| reset() | 初始化游戏界面 |
| setInterval(ballMove, ball) | 每50ms礼物和炸弹移动一次 |
| setInterval(ballCreate, add) | 每2000ms随机产生一次礼物和炸弹，炸弹比例高以增加难度 |
| ballMove() | 礼物和炸弹的移动控制函数 |
| ballCreate() | 礼物和炸弹的随机产生函数 |

## 核心算法

Keep Hiding部分算法代码如下图所示。

|  |
| --- |
| $(document).ready( function() {  $("canvas.snow").let\_it\_snow({  windPower: 3,  speed: 1,  count: 250,  size: 0,  });  $("canvas.flake").let\_it\_snow({  windPower: -3,  speed: 1,  count: 20,  size: 10,  image: "images/white-snowflake.png"  });  });  var width = document.getElementById("wrapper").offsetWidth;  var height = document.getElementById("wrapper").offsetHeight-100;  // var height = document.getElementById("wrapper").offsetHeight-100;  $("#myCanvas").attr("width", width);  $("#myCanvas").attr("height", height);  $(".fh").attr("width", width);  $(".fh").attr("height", height);  var cxt=document.getElementById("myCanvas").getContext("2d");  var ballArray=new Array();  var ballRadius = Math.floor(width\*0.04);  var timer;  var personRadius = Math.floor(width\*0.06);  var personSpeed = Math.floor(width\*0.1);  var ballInterval;  var addInterval;  var person;  var pause;  var gameOver;  var maxSpeed = Math.floor(width\*0.01);  var orgBall = 50;  var orgAdd = 2000;  var colors = ['./images/snow1.png','./images/snow2.png','./images/snow3.png','./images/snow4.png','./images/snow5.png']  function resetMotion(ball,add){  clearInterval(ballInterval);  clearInterval(addInterval);  ballInterval = window.setInterval(ballMove, ball);  addInterval = window.setInterval(ballCreate, add);  }  $("#left").click(function(){  if(pause == false&&gameOver == false)  person.wantGoLeft();  else if(gameOver == true)  reset();  });  $("#right").click(function(){  if(pause == false&&gameOver == false)  person.wantGoRight();  else if(gameOver == true)  reset();  });  //每隔50ms小球移动一下    function ballMove(){  for(var i=0;i<ballArray.length;i++){  ballArray[i].clearBall();  }  for(var i=0;i<ballArray.length;i++){  ballArray[i].drawBall();  }  for(var i = 0;i<ballArray.length;i++){  var ballX=ballArray[i].getLocationX();  var ballY=ballArray[i].getLocationY();  var personLocation = person.getLocation();  var distence = Math.sqrt(Math.pow((height-personRadius-ballY),2)+Math.pow((personLocation-ballX),2));  //console.log(distence);  if(distence<=ballRadius + personRadius){  var colorIndex = 0;  for(;colors.length;colorIndex ++){  if(ballArray[i].getColor() == colors[colorIndex])  break;  }  switch(colorIndex)  {  case 0:  gameOver = true;  timer.stop();  //alert("die");  $("#myModal").modal();  clearInterval(ballInterval);  clearInterval(addInterval);  timer.stop();  break;  case 1:  ballArray[i].clearBall();  ballArray.splice(i,1);  clearInterval(ballInterval);  clearInterval(addInterval);  setTimeout("resetMotion(orgBall,orgAdd)",5000);  break;  case 2:  ballArray[i].clearBall();  ballArray.splice(i,1);  resetMotion(orgBall\*5,orgAdd\*5);  setTimeout("resetMotion(orgBall,orgAdd)",5000);  break;  case 3:  ballArray[i].clearBall();  ballArray.splice(i,1);  person.setOpposite(true);  setTimeout("person.setOpposite(false);",5000);  break;  case 4:  ballArray[i].clearBall();  ballArray.splice(i,1);  person.setFreeze(true);  setTimeout("person.setFreeze(false);",5000);  break;  }  }  }  fixHide()  }  function fixHide()  {  for(var i = 0;i<ballArray.length;i++){  ballArray[i].clearBall();  }  for(var i = 0;i<ballArray.length;i++){  ballArray[i].redrawBall();  }  person.clearPerson();  person.drawPerson();  }  function reset(){  gameOver = false;  pause = false;  cxt.clearRect(0,0,width,height);  ballArray.splice(0,ballArray.length);  clearInterval(ballInterval);  clearInterval(addInterval);  var ball=new Ball(ballRadius,maxSpeed);  ball.init();  ballArray.push(ball);  person=new Person(personRadius,width/2,personSpeed);  person.drawPerson();  ballInterval = window.setInterval(ballMove, orgBall);  addInterval = window.setInterval(ballCreate, orgAdd);  if(timer != null)  timer.destroy();  timer = $('#someTimer').TimeCircles({  time : {  Minutes: {  show: true,  text: "min",  color: "#BFB"  },  Seconds: {  show: true,  text: "sec",  color: "#F99"  }  },  refresh\_interval: 0.1,  count\_past\_zero: true,  circle\_bg\_color: "#eee",  fg\_width: 0.05,  bg\_width: 1  });  }      function ballCreate(){  var ball=new Ball(ballRadius,maxSpeed);  ball.init();  if(ballArray.length<40){  ballArray.push(ball);  }  }    //创建一个小球的类  function Ball(radius,maxSpeed){  var ballColor = "";  var locationX = 0;  var locationY = 0;  var speedX = 0;  var speedY = 0;  this.clearBall=function(){  cxt.clearRect(locationX-radius,locationY-radius,2\*radius,2\*radius);  }  this.init = function(){  ballColor = this.randomColor(0,(colors.length-1)+3);  locationX = this.randomLocation(2\*radius,width-2\*radius);  locationY = this.randomLocation(2\*radius,height/4);  speedX = this.randomSpeed(1,maxSpeed);  speedY = this.randomSpeed(1,maxSpeed);  }  this.redrawBall = function(){  this.circleRender(locationX,locationY,radius,ballColor);  }  this.drawBall=function(){  this.calNextLocation();  this.circleRender(locationX,locationY,radius,ballColor);  }  this.calNextLocation=function(){  this.calNextSpeed();  locationX=locationX+speedX;  locationY=locationY+speedY;  }  this.calNextSpeed=function(){  if(locationX<radius||locationX>width-radius){  speedX=-speedX;  }  if(locationY<radius||locationY>height-radius){  speedY=-speedY;  }  }  this.getSpeedX = function(){  return speedX;  }  this.getSpeedY = function(){  return speedY;  }  this.setSpeedX = function(speed){  speedX = speed;  }  this.setSpeedY = function(speed){  speedY = speed;  }  this.getLocationX=function(){  return locationX;  }  this.getLocationY=function(){  return locationY;  }  this.randomLocation = function(min,max){  return Math.floor(min+Math.random()\*(max-min));  }  this.randomSpeed = function(min,max){  return Math.floor(min+Math.random()\*(max-min));  }  this.getColor = function(){  return ballColor;  }  this.randomColor = function(Min,Max)  {  var Range = Max - Min;  var Rand = Math.random();  var colorIndex = Min+Math.round(Rand \* Range);  if(colorIndex >= colors.length)  return colors[0];  else  return colors[colorIndex];  }  this.circleRender = function(x,y,r,color){  var img=new Image();  img.src=color;  cxt.drawImage(img,x-r,y-r,r\*2,r\*2);  }  }  function Person(personRadius,personLocation,personSpeed){  this.opposite = false;  this.freeze = false;  this.setOpposite = function(opposite){  this.opposite = opposite;  }  this.setFreeze = function(freeze){  this.freeze = freeze;  }  this.wantGoLeft = function(){  if(this.freeze == true)  return;  if(this.opposite == true)  this.goRight();  else  this.goLeft();  }  this.wantGoRight = function(){  if(this.freeze == true)  return;  if(this.opposite == true)  this.goLeft();  else  this.goRight();  }  this.goLeft=function(){  if(personLocation>personRadius){  cxt.clearRect(personLocation-personRadius,height-2\*personRadius,2\*personRadius,2\*personRadius);  personLocation=personLocation-personRadius;  fixHide();  }  }  this.goRight=function(){  if(personLocation<width-personRadius){  cxt.clearRect(personLocation-personRadius,height-2\*personRadius,2\*personRadius,2\*personRadius);  personLocation=personLocation+personRadius;  fixHide();  }  }  this.clearPerson=function(){  cxt.clearRect(personLocation-personRadius,height-2\*personRadius,2\*personRadius,2\*personRadius);  }  this.drawPerson=function(){  var img=new Image();  img.src="./images/hero.png";  cxt.drawImage(img,personLocation-personRadius,height-personRadius\*2,personRadius\*2,personRadius\*2);  }  this.getLocation=function(){  return personLocation;  }  }    $(function(){  $("#start").click(function(){  $("#layer").css("display","none");  var now = new Date;  /\*min = now.getMinutes();  sec = now.getSeconds();\*/  reset();  });  $("#information").click(function(){  $("#title").css("display","none");  $("#start").css("display","none");  $("#information").css("display","none");  // var innerHtml=;  $(".guide").css("display","block");  $("#return").css("display","block");  // alert("hahaha");  });  $("#return").click(function(){  //window.location.reload();//刷新当前页面.  $("#title").css("display","block");  $("#start").css("display","block");  $("#information").css("display","block");  // var innerHtml=;  $(".guide").css("display","none");  $("#return").css("display","none");  });  $("#restart").click(function(){  reset();  });  $("#someTimer").click(function(){  $("#layer").css("display","block");  reset();  clearInterval(ballInterval);  clearInterval(addInterval);  timer.stop();  $("#title").css("display","block");  $("#start").css("display","block");  $("#information").css("display","block");  // var innerHtml=;  $(".guide").css("display","none");  $("#return").css("display","none");  });    window.addEventListener('load', function () {  FastClick.attach(document.body);  }, false);    window.onbeforeunload = exit\_handler;  window.onunload = exit\_handler;  function exit\_handler(){  clearInterval(ballInterval);  clearInterval(addInterval);  }  }); |

## 软件功能界面



