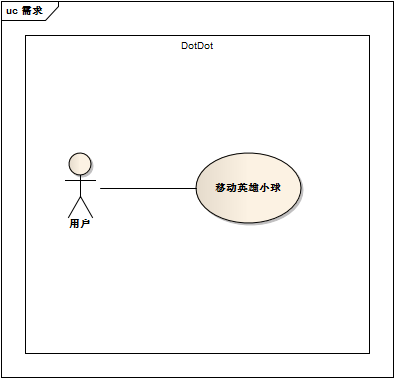
Tizen课程项目报告  
《DotDot》

## 项目简介

DOTDOT是一款考验灵活的益智小游戏。游戏的全部内容就是移动蓝色小球，躲避红色小球，时间越长，得分越多。游戏中主要有红色小球，黄色小球，灰色小球、绿色小球等几种颜色的小球。不同颜色的小球有着不同的功能。碰到红色的小球游戏就会结束，碰到绿色的小球会让蓝色小球变得无敌，碰到黄色的小球会让蓝色小球变小，碰到灰色小球会让蓝色小球变大不利于蓝色小球躲避障碍。

## 需求分析



## 概要设计

DOTDOT基于TIZEN web project开发，主要使用了Html与Javascript技术。通过在ontouchstart、ontouchmove和ontouchend方法中得到手指触碰位置来更新蓝色小球的位置，并判断蓝色小球是否碰撞到其他小球。

|  |  |
| --- | --- |
| 接口 | 接口功能 |
| update | 刷新游戏界面 |
| DeleteBalls | 删除小球 |
| AddBalls | 增加小球 |
| Reflect | 小球碰到墙壁反弹 |
| CastMagic | 施放魔法 |

## 核心算法

DotDot核心算法代码如下图所示。

|  |
| --- |
| function Hero(x,y){  this.HREO\_MIN\_R = 2;  this.x = x;  this.y = y;  this.life = 1;  this.r = 5;  this.magics = new Array();  this.noEnemy = false;  this.color = "#666666";  this.touch = true;  this.display = function(container){  var heroCircle = container.append("circle").attr("class","hero").attr("cx", this.x).attr("cy", this.y).attr(  "r", this.r).attr("fill", this.color);  if(this.noEnemy){  heroCircle.attr("stroke-width","2").attr("stroke","#99CC66");  }  }  this.addMagic = function(ball){  this.magics.push(ball);  }  this.update = function(){  if(this.magics.length <= 0)  return;  var hasMagic = false;  for(var i = 0;i < this.magics.length;i ++){  this.magics[i].life --;  if(this.magics[i].life > 0){  hasMagic = true;  }  }  if(!hasMagic){  var first = this.magics[0];  if(first.life <= 0){  var ball = this.magics.shift();  ball.noMagic(this);  }  }  }    this.touchStart = function(position){  var distance = Math.sqrt(Math.pow(position.x - this.x,2) + Math.pow(position.y - this.y,2));  this.touch = distance <= 5;  }    this.touchMove = function(position){  if(this.touch){  this.x = position.x;  this.y = position.y;  }  }  }  function Ball(x,y,life,color){  this.x = x;  this.y = y;  this.life = life;  this.vx = (Math.random()-0.5)\*5;  this.vy = (Math.random()-0.5)\*5;  this.r = 5;  this.color = color;  this.reflect = function(width,height){  if(this.x - this.r <= 0 || this.x + this.r >= width)  this.vx = -this.vx;  if(this.y - this.r <= 0 || this.y + this.r >= height)  this.vy = -this.vy;  }  this.update = function(){  this.x += this.vx;  this.y += this.vy;  this.life --;  }  this.display = function(container){  container.append("circle").attr("class","ball").attr("cx", this.x).attr("cy", this.y).attr(  "r", this.r).attr("fill", this.color);  }    this.collide = function(hero){  var distance = Math.sqrt(Math.pow(hero.x - this.x,2) + Math.pow(hero.y - this.y,2));  return distance <= hero.r + this.r;  }  }  function KillBall(x,y,life){  this.base = new Ball(x,y,life,"#FF6666");  this.castMagic = function(hero){  if(hero.noEnemy)  return false;  hero.life -= 1;  return false;  }  }  function ExpandBall(x,y,life){  this.base = new Ball(x,y,life,"#CCCCCC");  this.castMagic = function(hero){  hero.r += 1;  return true;  }  this.noMagic = function(hero){  hero.r -= 1;  }  }  function ShinkBall(x,y,life){  this.base = new Ball(x,y,life,"#FFFF66");  this.cast = false;  this.castMagic = function(hero){  if(hero.r > 2){  hero.r -= 1;  this.cast = true;  }  return true;  }  this.noMagic = function(hero){  if(this.cast)  hero.r += 1;  }  }  function SuperBall(x,y,life){  this.SUPERTIME = 200;  this.base = new Ball(x,y,life,"#99CC66");  this.life = this.SUPERTIME;    this.castMagic = function(hero){  hero.noEnemy = true;  hero.addMagic(this);  return true;  }  this.noMagic = function(hero){  hero.noEnemy = false;  }  }  function Game(container,width,height){  this.container = container;  this.width = width;  this.height = height;  this.score = 0;    this.birthTime = 25;//出球时间间隔  this.ballLife = 1500;//每个小球的生命  this.ticks = 0;//计时器  this.hero = new Hero(this.width/2,this.height/2);  this.balls = new Array();    this.isEnd = 0;    this.update = function(){  if(this.hero.life <= 0)  return this.endGame();  this.deleteBalls();  if(this.ticks % this.birthTime == 0)  this.addBalls();  this.reflect();  this.castMagic();  this.hero.update();  for(var i in this.balls)  this.balls[i].base.update();  this.ticks ++;  this.score ++;  this.display();  }  this.display = function(){  this.container.selectAll("circle").remove();  this.container.select("#score").text(Math.round(this.score/1000\*15));  this.hero.display(this.container);  for(var i in this.balls)  this.balls[i].base.display(this.container);  }  this.endGame = function(){  d3.select("#replay").style("display","block");  }  this.deleteBalls = function(){  if(this.balls.length <= 0)  return;    var first = this.balls[0];  if(first.base.life <= 0)  this.balls.shift();  }  this.addBalls = function(){  var x = Math.random()\*this.width;  var y = Math.random()\*this.height;  var r = Math.random();  if(r < 0.7)  this.balls.push(new KillBall(x,y,this.ballLife));  else if(r < 0.8)  this.balls.push(new ExpandBall(x,y,this.ballLife));  else if(r < 0.9)  this.balls.push(new ShinkBall(x,y,this.ballLife));  else  this.balls.push(new SuperBall(x,y,this.ballLife));  }  this.reflect = function(){  for(var i in this.balls)  this.balls[i].base.reflect(this.width,this.height);  }  this.castMagic = function(){  for(var i = 0;i < this.balls.length;i ++){  if(this.balls[i].base.collide(this.hero)){  if(this.balls[i].castMagic(this.hero)){  this.balls.splice(i,1);  i--;  }  }  }  }  } |

## 软件功能界面

