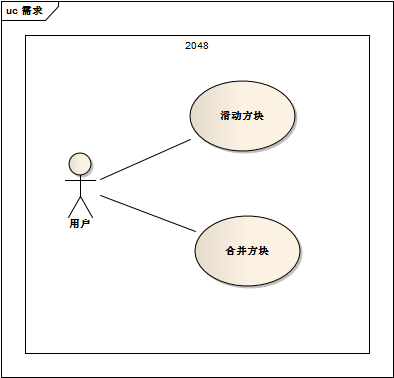
Tizen课程项目报告  
《2048》

## 项目简介

2048是一款流行的数字益智游戏。这款游戏的玩法很简单，游戏初始时会出现包含2在内的2的倍数，游戏玩家每次可以选择上下左右滑动，每滑动一次，所有的数字方块都会往滑动的方向靠拢，相同数字的方块在靠拢、相撞时会相加，系统也会在空白的地方随机出现一个数字方块。通过不断的滑动与叠加最终拼凑出2048这个数字就算成功。

## 需求分析



## 概要设计

2048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，从而进行上下左右滑动、Merge等操作。

|  |  |
| --- | --- |
| 接口 | 接口功能 |
| Init | 初始化游戏界面 |
| NewTile | 初始化方块 |
| RandomTile | 随机产生新方块 |
| Move | 移动方块 |
| Merge | 合并方块 |

## 核心算法

2048核心算法代码如下图所示。

|  |
| --- |
| function game2048(container) {  this.container = container;  this.tiles = new Array(16);  this.score = 0;  }  game2048.prototype = {  init : function() {  for (var i = 0, len = this.tiles.length; i < len; i++) {  var tile = this.newTile(0);  tile.setAttribute('index', i);  this.container.appendChild(tile);  this.tiles[i] = tile;  }  this.randomTile();  this.randomTile();  },  newTile : function(val) {  var tile = document.createElement('div');  this.setTileVal(tile, val)  return tile;  },  setTileVal : function(tile, val) {  tile.className = 'tile tile' + val;  tile.setAttribute('val', val);  tile.innerHTML = val > 0 ? val : '';  },  randomTile : function() {  var zeroTiles = [];  for (var i = 0, len = this.tiles.length; i < len; i++) {  if (this.tiles[i].getAttribute('val') == 0) {  zeroTiles.push(this.tiles[i]);  }  }  var rTile = zeroTiles[Math.floor(Math.random() \* zeroTiles.length)];  this.setTileVal(rTile, Math.random() < 0.8 ? 2 : 4);  },  move : function(direction) {  var j;  switch (direction) {  // 向上滑动  case 'UP':  for (var i = 4, len = this.tiles.length; i < len; i++) {  j = i;  while (j >= 4) {  this.merge(this.tiles[j - 4], this.tiles[j]);  j -= 4;  }  }  break;  // 向下滑动  case 'DOWN':  for (var i = 11; i >= 0; i--) {  j = i;  while (j <= 11) {  this.merge(this.tiles[j + 4], this.tiles[j]);  j += 4;  }  }  break;  // 向左滑动  case 'LEFT':  for (var i = 1, len = this.tiles.length; i < len; i++) {  j = i;  while (j % 4 != 0) {  this.merge(this.tiles[j - 1], this.tiles[j]);  j -= 1;  }  }  break;  // 向右滑动  case 'RIGHT':  for (var i = 14; i >= 0; i--) {  j = i;  while (j % 4 != 3) {  this.merge(this.tiles[j + 1], this.tiles[j]);  j += 1;  }  }  break;  default:  return;  }  this.randomTile();  },  merge : function(prevTile, currTile) {  var prevVal = prevTile.getAttribute('val');  var currVal = currTile.getAttribute('val');  if (currVal != 0) {  if (prevVal == 0) {  this.setTileVal(prevTile, currVal);  this.setTileVal(currTile, 0);  } else if (prevVal == currVal) {  this.setTileVal(prevTile, prevVal \* 2);  this.setTileVal(currTile, 0);  this.score += prevVal \* 2;  $("#score").html(this.score);  }  }  },  equal : function(tile1, tile2) {  return tile1.getAttribute('val') == tile2.getAttribute('val');  },  max : function() {  for (var i = 0, len = this.tiles.length; i < len; i++) {  if (this.tiles[i].getAttribute('val') == 2048) {  return true;  }  }  },  over : function() {  for (var i = 0, len = this.tiles.length; i < len; i++) {  if (this.tiles[i].getAttribute('val') == 0) {  return false;  }  if (i % 4 != 3) {  if (this.equal(this.tiles[i], this.tiles[i + 1])) {  return false;  }  }  if (i < 12) {  if (this.equal(this.tiles[i], this.tiles[i + 4])) {  return false;  }  }  }  return true;  },  clean : function() {  for (var i = 0, len = this.tiles.length; i < len; i++) {  this.container.removeChild(this.tiles[i]);  }  this.tiles = new Array(16);  this.score = 0;  document.getElementById("score").innerHTML(0);  }  } |

## 软件功能界面

