2019.10.23

경성대학교
KYUNGSUNG UNIVERSITY

# 2048게임 소스

2014758076 윤승재

# HTML

```
<html>

<head>
  <title>2048 Game</title>
  <meta charset="utf-8">
  <link
href="https://fonts.googleapis.com/css?f
amily=Nunito:200,400,800"
rel="stylesheet">

  <script src="2048.js"></script>
  <link rel="stylesheet" href="2048.css">
</head>
```

```
<body>
  <div class="game">
    <div class="head">
      <div class="a">2048 <button
class="info" onClick='info()'>i</button>
<button id="repeat" class="info repeat"
onClick='reset()'>↻</button></div>
      <div class="score">Score<br
/><span id="value"></span></div>
    </div>
    <div class="description"
id="description">
      게임 방법:<br /><br />
      키보드의 상 하 좌 우 방향키를 이용
하여 진행합니다. <br />
      두개의 같은 수 타일을 합쳐 최종적
으로 2048을 만들면 게임이 종료됩니
다.<br /><br />
      <br><br />
    </div>
```

# HTML

```
<div class="field">

    <div class="row">
      <div class="cell"></div>
      <div class="cell"></div>
      <div class="cell"></div>
      <div class="cell"></div>
    </div>
    <div class="row">
      <div class="cell"></div>
      <div class="cell"></div>
      <div class="cell"></div>
      <div class="cell"></div>
    </div>
```

```
  <div class="row">
        <div class="cell"></div>
        <div class="cell"></div>
        <div class="cell"></div>
        <div class="cell"></div>
      </div>
      <div class="row">
        <div class="cell"></div>
        <div class="cell"></div>
        <div class="cell"></div>
        <div class="cell"></div>
      </div>
    </div>

  </div>
  <div class='' id='status'>
  </div>
</body>

</html>
```

# JS

```
window.onload = function() {
  buildGridOverlay();
  cellCreator(2, 0);
  directions();
  score(0);
};


function buildGridOverlay() {
  var game     =
document.getElementsByClassName('ga
me');
  var grid     =
document.getElementsByClassName('grid
');
  var size     = 4;
  var table    =
document.createElement('DIV');

  table.className += 'grid';
  table.id = ' ';
  table.dataset.value = 0;

 for (var i = 0; i < size; i++) {
    var tr =
document.createElement('DIV');
    table.appendChild(tr);
    tr.id = 'row_' + (i+1);
    tr.className += 'grid_row';

    for (var j = 0; j < size; j++) {
      var td =
document.createElement('DIV');
      td.id = '' +(i+1) +(j+1);
      td.className += 'grid_cell';
      tr.appendChild(td);
    }
  document.body.appendChild(table);
  }

  return table;
}
```

## JS

```
function cellCreator(c, timeOut) {
  for (var i = 0; i < c; i++) {

    var count = 0;

    for (var value = 1; value < 2; value++)
{
      var randomX =
Math.floor((Math.random()*4)+1);
      var randomY =
Math.floor((Math.random()*4)+1);
      var checker =
document.getElementById('' +randomX
+randomY);
      if (checker.innerHTML != '') {
        value = 0;
      }
    }
```

```
 var randomValue =
Math.floor((Math.random()*4) +1);
//create value 1, 2, 3 or 4
    if (randomValue == 3)
{randomValue=4};
    if (randomValue == 1)
{randomValue=2};
    var position =
document.getElementById(''+randomX
+randomY);
    var tile =
document.createElement('DIV');
//create div at x, y
    position.appendChild(tile);
//tile becomes child of grid cell
    tile.innerHTML = ''+randomValue;
```

## JS

```
 colorSet(randomValue, tile);
    tile.data = ''+randomValue;
    tile.id = 'tile_'+randomX +randomY;
    position.className += ' active';
    var tileValue = tile.dataset.value;
    tile.dataset.value = ''+randomValue;

    console.info(''+timeOut);
    if (timeOut == 0) {
      tile.className = 'tile '+randomValue;
    } else { setTimeout(function() {
        tile.className = 'tile
'+randomValue;
      }, 10); }
  }
}
```

## JS

```
document.onkeydown = directions;

function directions(e) {
  e = e || window.event;
  var d = 0;
if (e.keyCode == '38') {
      var count = 2;

      for (var x = 2; x > 1; x--) {
        for (var y = 1; y < 5; y++) {
          moveTilesMain(x, y, -1, 0, 1, 0);
          console.info(''+x +y);
        }
        if (x == 2) {
          x += count;
          count++;
        }
        if (count > 4) { break; }
      }
      cellReset();
    }
else if (e.keyCode == '40') {
      var count = -2;
      var test  = 1;
      for (var x = 3; x < 4; x++) {
        for (var y = 1; y < 5; y++) {
          moveTilesMain(x, y, 1, 0, 4, 0);
        }
        if (x == 3) {
          x += count;
          count--;
        }
        if (count < -4) { break; }
      }
      cellReset();
  }
```

JS

```
else if (e.keyCode == '37') {


    var count = 2;
    var test  = 1;
    for (var x = 2; x > 1; x--) {
      for (var y = 1; y < 5; y++) {
        moveTilesMain(y, x, 0, -1, 0, 1);
      }
      if (x == 2) {
        x += count;
        count++;
      }
      if (count > 4) { break; }
    }
    cellReset();
  }
```

```
 else if (e.keyCode == '39') {

      var count = -2;
      var noCell = 0;
      var c = 1;
      var d = 0;

      for (var x = 3; x < 4; x++) {
        for (var y = 1; y < 5; y++) {
          moveTilesMain(y, x, 0, 1, 0, 4, c,
d);
        }
        if (x == 3) {
          x += count;
          count--;
        }
        if (count < -4) { break; }
      }
      cellReset();
    }
}
```

## JS

```
function moveTilesMain(x, y, X, Y, xBorder,
yBorder, c, d) {

  var tile      =
document.getElementById('tile_'+x +y);
  var checker   =
document.getElementById(''+x +y);
  var xAround   = x+X;
  var yAround   = y+Y;

  if (xAround > 0 && xAround < 5 &&
yAround > 0 && yAround < 5 &&
checker.className == 'grid_cell active') {
    var around =
document.getElementById(''+xAround
+yAround);
 var aroundTile =
document.getElementById('tile_'+xAroun
d +yAround);
       if (aroundTile.innerHTML ==
tile.innerHTML) {
         //same
         var value = tile.dataset.value*2;
         aroundTile.dataset.value = ''+value;
         aroundTile.className = 'tile
'+value;
         aroundTile.innerHTML = ''+value;
         colorSet(value, aroundTile);
         checker.removeChild(tile);
         checker.className = 'grid_cell';
         around.className  = 'grid_cell
active merged';
```

## JS

```
document.getElementsByClassName('grid
').id = 'moved';

document.getElementsByClassName('grid
').className = 'grid '+value;
        var grid =
document.getElementById(' ');
        var scoreValue =
parseInt(grid.dataset.value);
        var newScore = value +
scoreValue;

        grid.dataset.value = newScore;
        var score =
document.getElementById('value');

        score.innerHTML = ''+newScore;
      }
    }
```

```
 else if (around.className ==
'grid_cell'){

        around.appendChild(tile);
        around.className = 'grid_cell
active';
        tile.id = 'tile_'+xAround +yAround;
        checker.className = 'grid_cell';

document.getElementsByClassName('grid
').id = 'moved';
    }
  }
}
```

## JS

```
function cellReset() {
  var count = 0;
  var a =
document.getElementsByClassName('grid
').id;
  console.log(''+a);

  for (var x=1; x<5; x++) {
    for (var y=1; y<5; y++) {

      var resetter =
document.getElementById(''+x +y);
      if (resetter.innerHTML != '') {
        count++;
      }
 if (resetter.innerHTML == '') {
        resetter.className = 'grid_cell';
      }

      if (resetter.className == 'grid_cell
active merged') {
        resetter.className = 'grid_cell
active'
      }
    }
  }
```

## JS

```
 if (count == 16) {

document.getElementById('status').className = 'lose';
  } else if (document.getElementsByClassName('grid').id == 'moved'){
     cellCreator(1, 1);
  }

document.getElementsByClassName('grid').id = ' ';
}

}
```

## JS

```
function score() {

  var grid = document.getElementById('
');
  var value = grid.dataset.value;

document.getElementById('value').innerH
TML = ''+value;

}
```

## JS

```
function colorSet(value, tile) {
  switch(value) {
    case 2:    tile.style.background =
'#fbfced'; tile.style.color = 'black'; break;
    case 4:    tile.style.background =
'#ecefc6'; tile.style.color = 'black'; break;
    case 8:    tile.style.background =
'#ffb296'; tile.style.color = 'black'; break;
    case 16:   tile.style.background =
'#ff7373'; tile.style.color = 'black'; break;
    case 32:   tile.style.background =
'#f6546a'; tile.style.color = 'white'; break;
    case 64:   tile.style.background =
'#8b0000'; tile.style.color = 'white'; break;
    case 128:  tile.style.background =
'#794044'; tile.style.color = 'white';
               tile.style.fontSize = '50px';
break;
    case 256:  tile.style.background =
'#31698a'; tile.style.color = 'white';
               tile.style.fontSize = '50px';
break;
    case 512:  tile.style.background =
'#297A76'; tile.style.color = 'white';
               tile.style.fontSize = '50px';
break;
    case 1024: tile.style.background =
'#2D8A68'; tile.style.color = 'white';
               tile.style.fontSize = '40px';
break;
    case 2048: tile.style.background =
'#1C9F4E'; tile.style.color = 'white';
               tile.style.fontSize = '40px';

document.getElementById('status').classN
ame = 'won'; break;
  }

}
```

JS

```
function info() {
  setTimeout(function() {

document.getElementById('description').c
lassList.toggle('show');
  }, 10);

}
```

## JS

```
function reset() {
  for (var x = 1; x < 5; x++) {
    for (var y = 1; y < 5; y++) {
      var resetter =
document.getElementById(''+x +y);
      if (resetter.className == 'grid_cell
active') {
        var tile =
document.getElementById('tile_'+x +y);
        resetter.removeChild(tile);
      }
    }
  }
document.getElementById('status').classN
ame = '';
  document.getElementById('
').dataset.value = 0;
  score();
  cellReset();
  cellCreator(2, 0);
}
```

# End.

2014758076 윤승재