Blocks game Series 40 Web App

This article shows how to construct a game similar to Tetris as a Series 40 Web App. Article shows you how to develop the game using S40 web SDK. Here I used some of main functionality from mwl api like"timer" "widget.pereference"

Note: This is an entry in the Asha Touch Competition 2012Q3

Warning: The CSS2 position property may not be supported as expected by developers on the browser (along with other elements, like those used in JQuery). Note that an app using unsupported elements may run on *Local Preview* but will fail on *Cloud Preview* (and on a real device).

This article needs to be updated: If you found this article useful, please fix the problems below then delete the {{ArticleNeedsUpdate}} template from the article to remove this warning.

Reasons: hamishwillee (11 Sep 2012)
This awesome app code does not run in Cloud Preview due to unsupported CSS2 position property. It should be updated to use supported elements.

Introduction

This article and the accompanying HTML files show how to construct a game similar to the popular falling down blocks game called Tetris, called "S40 Blocks". The game has 4-block pieces, 7 distinct shapes that fall from the top. The challenge is to manipulate them by horizontal and rotation moves so that lines get completely filled up. You receive points for completing lines. Filled lines are removed. Completing more than one line (up to 4) at a time results in more points.

Series 40 blocks game

Design decisions

The game has three design decisions deserve special mention.

I chose an essentially static approach. No elements move. Instead, the board contains a sequence of img tags laid out in a grid. The contents of these img tags (the src values) change. This makes it easy to determine when a row (line) of the board is filled. The falling shapes are sets of blocks, four blocks each, put together using what I term formulas. I did this because the shapes must be considered as separate blocks once they hit down. The left, right, rotate and down buttons are implemented as form submit buttons.

Development stages

Initial Stage

Before start programming we need resources for the game. I create 7 block with different color using inkscape tool each block are 20x20px.

Basic HTML file Structure

(C) Copyright Nokia 2014. All rights reserved.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>blocks</title>
    <link href="s40-theme/css/blocks.css" rel="stylesheet" type="text/css" />
    <script language="javascript" type="text/javascript" src="s40-
theme/js/blocks.js"></script>

</head>

<body>
  <table>
  <tr>
  <td>
  <script language="JavaScript" type="text/javascript">
 createBoard();
  </script>
  </td>
  </tr>
</table>
</body>
</html>
```

Basic java script file with functions declared

```
//create Game Board
function createBoard() {
}

// Image Number for each block
function imagenumber(col, row) {
}

//create the shpes with formula
function makeBlock(type, col, row) {
}

//move the block
function moveOver(dir) {
}

//rotate the block
function rotate() {
}

//check if block hit down
function hitDown()  {
}

//move the block faster when press down key
function moveDown() {
}
```


(C) Copyright Nokia 2014. All rights reserved.

```
//game main timer
function gameTimer () {
}

//black fall down and fill the line
function fallDown() {
}

//remove the filled line move the row down
function remove(cut) {
}

//next block to come
function newBlock() {
}

//start the game
function startGame() {
}
```

All further changes are done in the following function in JavaScript.

Create the board by creating a 2D grid of `img` tags. These image tags are all in one "<td>" tag. Board contains 9 column and 15 rows totally want to create 135 img tag. The board look like "Image1".

The `img` tags initially hold border less white block held in the file **block.png**. Each block is of size 20x20 pixels

Initally create the variable of the game:

```
var boardcol = 9;               //number of columns
var boardrow = 15;            //number of rows
var tid ="gameloop";        //timer id
var timeperiod = 500;      //Used in call to setInterval to set interval between drops.
Make longer and shorter to ease debugging.
var grace = 0;            //default grace period
var startnewone = false; //flag
var graceperiod = 3;    //grace period
//image number, column, row of current 4- block shape
var current = [
               [0,0,0],
               [0,0,0],
               [0,0,0],
               [0,0,0]
               ];

//initial construction of shapes
var blockformulas = [
                     [[0,0],[1,0],[2,0],[1,1]], //T shape
                     [[0,0],[1,0],[2,0],[3,0]], //line Shape
                     [[0,1],[1,1],[1,0],[2,0]], // O Shape
                     [[0,0],[1,0],[0,1],[1,1]], //shifted pieces
                     [[0,0],[1,0],[1,1],[2,1]], //other shifted piece
                     [[0,0],[1,0],[2,0],[2,1]], //Opposite of L Shape
                     [[0,1],[1,1],[2,1],[2,0]] //L Shape
                     ];
var scoring= [10, 20, 40, 80]; //1 row 10 2 row 20 3 row 40 4 row 80 points
//file names for single colored blocks with borders
```


(C) Copyright Nokia 2014. All rights reserved.

```
var blockimages = [
   "s40-theme/images/b1.png",
   "s40-theme/images/b2.png",
   "s40-theme/images/b3.png",
   "s40-theme/images/b4.png",
   "s40-theme/images/b5.png",
   "s40-theme/images/b6.png",
   "s40-theme/images/b7.png"
  ];
var currenttype;         // holds image file name
var currenttypenum;      // 0 to 6
var currentorientation;  // 0 to 3
var currentorigin;       // nominal origin [x,y]
//orientations[orient][type][block 0 to 3][x and y]
var orientations = [
                    [
                    [[0,0],[1,0],[2,0],[1,1]], //First element is blockformulas
                    [[0,0],[1,0],[2,0],[3,0]], //
                    [[0,1],[1,1],[1,0],[2,0]], //
                    [[0,0],[1,0],[0,1],[1,1]], //
                    [[0,0],[1,0],[1,1],[2,1]], //
                    [[0,0],[1,0],[2,0],[2,1]], //
                    [[0,1],[1,1],[2,1],[2,0]] //
                    ],
                    [
                    [[1,0],[1,1],[1,2],[2,1]], //next orientation index = 1
                    [[1,0],[1,1],[1,2],[1,3]], //
                    [[1,2],[1,1],[0,1],[0,0]], //
                    [[0,0],[1,0],[0,1],[1,1]], //
                    [[1,0],[1,1],[0,1],[0,2]], //
                    [[1,2],[1,1],[1,0],[2,0]],  //
                    [[2,2],[2,1],[2,0],[1,0]] //
                    ],
                    [
                    [[0,1],[1,1],[2,1],[1,0]],  //next orientation index =2
                    [[0,0],[1,0],[2,0],[3,0]], //
                    [[2,0],[1,0],[1,1],[0,1]], //
                    [[0,0],[1,0],[0,1],[1,1]], //
                    [[0,0],[1,0],[1,1],[2,1]], //
                    [[2,1],[1,1],[0,1],[0,0]], //
                    [[2,0],[1,0],[0,0],[0,1]]  //
                    ],
                    [
                    [[1,0],[1,1],[1,2],[0,1]], //next orientation index = 3
                    [[1,0],[1,1],[1,2],[1,3]], //
                    [[1,2],[1,1],[0,1],[0,0]], //
                    [[0,0],[1,0],[0,1],[1,1]], //
                    [[1,0],[1,1],[0,1],[0,2]], //
                    [[1,0],[1,1],[1,2],[0,2]], //
                    [[1,0],[1,1],[1,2],[2,2]]  //
                    ]
                    ];

function createBoard() {
 var i;
 var j;
```


(C) Copyright Nokia 2014. All rights reserved.

```
 for (i=0; i<boardrow; i++) {
  for (j=0; j<boardcol; j++) {
   document.write("<img src='s40-theme/images/block.png'/>");
   }
  document.write("<br/>");
   }

}
```

The `createBoard()` function is called from within a script element in the <body> element.It uses what will become a very familiar construction of nested for loops. The variables `boardrow` and `boardcol` hold the number of columns and rows, respectively. After each complete iteration of the inner for loop, a <br/ > tag is output to go to the next row. At this point, you may ask how the `img` tags can be accessed. The answer is by using the so-called images collection of the document object.

```
document.images[imgno].src
```

can be used to access or set the img tag indicated by the number `imgno`. A function called `imagenumber()`, to be described below, will convert from column and row to image number.The <a> hyperlink that calls `startGame()` is not yet functional.

A critical feature of this game is the 4-block sets. Each shape is prescribed by a formula. (When we get to rotation, we will use an array of arrays of arrays to designate the formula for each orientation of each of the 7 types.) A formula specifies the x and y offset from an origin for each of the 4 blocks.

```
function imagenumber(col, row) {
   var imagenum = row*boardcol + col;
   return imagenum;
  }
function makeBlock(type, col, row) {
    var tests;
    var found;
    currentorigin = [col, row];
    currenttypenum = type;
    currenttype = blockimages[type];
    currentorientation = 0;
    var i;
    var block = blockimages[type];
    var formula = blockformulas[type];
    var imagenum;
    var atc;
    var atr;
    for (i=0;i<=3;i++) {
   atc = col + formula[i][0];
   atr = row + formula[i][1];
   imagenum=imagenumber(atc, atr);
    //check for room to add block. If none, end game.
          tests = String(document.images[imagenum].src);
          found = tests.search("s40-theme/images/block.png");
          if (found>=0) {
            document.images[imagenum].src = block;
     current[i][0]=imagenum;
     current[i][1] = atc;
     current[i][2] = atr;
          }
          else {
   alert("Game over.");
```


(C) Copyright Nokia 2014. All rights reserved.

```
                  mwl.stopTimer(tid);
                  break;
          }
     }
   }
```

To test the game, but without playing the game, I make the down button actually move the piece down one row. In what I term the real game, the down button is used to send the piece as far down as it can go all in one step. This stage will check the internal action to move the shape down until it hits something. This code works like the horizontal and rotation operations. It is necessary to use a multi-step procedure with provision to restore pieces if the move was blocked.

```
function moveDown() {
     var i;
    var tests;
    var oksofar = true;
    var imgno;
    var atc;
    var atr;
     var newcurrent = new Array();
     var saved = new Array();
    var found;
    var hitdown = false;
   for (i=0; i<=3; i++) {
     imgno = current[i][0];
     atc = current[i][1];
     atr = current[i][2];
     if (atr>=(boardrow-1)) {
       hitdown = true;
       oksofar = false;
       break;
       }
     newcurrent[i] = imagenumber(atc,atr+1);
     }
   if (oksofar) {
   for (i=0;i<=3; i++) {
     saved[i] = current[i][0];
     document.images[current[i][0]].src = "s40-theme/images/block.png";
        }
   for (i=0; i<=3; i++) {
                  tests = String(document.images[newcurrent[i]].src);
                  found = tests.search("s40-theme/images/block.png");
                  if (found == -1) {
                    oksofar = false;
                    break;
                    }
               }
   if (oksofar) {
      for (i=0;i<=3; i++) {
        document.images[newcurrent[i]].src = currenttype;
        current[i][0] = newcurrent[i];
        current[i][2]++;
       } //ends for loop
      currentorigin[1]++;
    }
   else {
     for (i=0;i<=3; i++) {
```


(C) Copyright Nokia 2014. All rights reserved.

```
        document.images[saved[i]].src = currenttype;
        hitdown = true;
      }
     }
    }
    if (hitdown) {
       startnewone=true;
       grace = 0;
     }
    else {
       if (hitDown()) {
        startnewone = true;
        grace = graceperiod;
        }
     }
  }
function hitDown()  {
     var i;
    var tests;
    var oksofar = true;
    var imgno;
    var atc;
    var atr;
     var newcurrent = new Array();
     var saved = new Array();
    var found;
    var hitdown = false;
    for (i=0; i<=3; i++) {
      imgno = current[i][0];
      atc = current[i][1];
      atr = current[i][2];
      if (atr>=(boardrow-1)) {
        hitdown = true;
        oksofar = false;
        break;
        }
      newcurrent[i] = imagenumber(atc,atr+1);
      }
    if (oksofar) {
    for (i=0;i<=3; i++) {
      saved[i] = current[i][0];
      document.images[current[i][0]].src = "s40-theme/images/block.png";
         } // ends for loop
    for (i=0; i<=3; i++) {
                    tests = String(document.images[newcurrent[i]].src);
                    found = tests.search("s40-theme/images/block.png");
                    if (found == -1) {
                      oksofar = false;
                    atc = currentorigin[1];
                      hitdown = true;
                      break;
                      }
                  }
    for (i=0;i<=3; i++) {
        document.images[saved[i]].src = currenttype;
         }
    }
```


(C) Copyright Nokia 2014. All rights reserved.

```
    startnewone = true;
    grace = graceperiod;
    return hitdown;
 }
```

`hitDown()` function check if a piece has hit down and can't go further. The blanks are still being counted, and not the non-blank space

See the implementation in "image2". Some minor changes are done in html file add input submit button to control the game and add two textfield for"Level" and "Line" and add start game link tag "<a>" to start the game.

```
<form name="f">
   <input type="button" onClick="moveover(-1);" value="left">
   <input type="button" onClick="rotate();" value="rotate">
   <input type="button" onClick="moveover(1);" value="right">
   <br/>
                
   <input type="button" onClick="movedown();"  value="down">
   <br/>
   Lines: <input type="text" name="lines" value="0"> <br/>
   Score: <input type="text" name="score" value="0"> <br/>
</form>
<a href="javascript:startgame();">Start Game </a> <br/>
```

Functionality

In this section i cover main four functionality of the game

1. MoveOver
2. Rotate
3. FallDown
4. GameTimer

MoveOver

`moveOver()` function is used to move move the block one step left are one step right. `moveOver()` function have parameter dir set on which direction black want to move. **Dir Parameter dir =-1 move the block left and dir=1 move the block right**

```
function moveOver(dir) {
    var i;
    var tests;
    var oksofar = true;
    var imgno;
     var newcurrent = new Array();
     var saved = new Array();
   for (i=0; i<=3; i++) {
     imgno = current[i][0];

     if (dir==-1) {
        if (0 == imgno % boardcol)
           { oksofar = false;
           break; } }
     if (dir == 1) {
       if ((boardcol-1)== imgno % boardcol) {
           oksofar = false;
           break; } }
     newcurrent[i] = imgno+dir;
```


(C) Copyright Nokia 2014. All rights reserved.

```
      }
    // if oksofar (no blocks at critical edge, newcurrent is set

    if (oksofar) {
                 for (i=0; i<=3; i++) {
                   saved[i] = current[i][0];
                   document.images[current[i][0]].src = "s40-theme/images/block.png";
                 }

                 for (i=0; i<=3; i++) {
                    tests = String(document.images[newcurrent[i]].src);
                    found = tests.search("s40-theme/images/block.png");
                    if (found == -1) {
                      oksofar = false;
                      break;
                      }
                 }

                 if (oksofar) {
                    for (i=0;i<=3;i++) {
                      document.images[newcurrent[i]].src= currenttype;
                      current[i][0] = newcurrent[i];
                      current[i][1] = current[i][1]+dir;
                    }

                    currentorigin[0]=currentorigin[0]+dir;
                    hitDown();
                   }
                 else {
                     for (i=0;i<=3;i++) {
  document.images[saved[i]].src = currenttype;
                     }
                 }
        }
      }
```

Rotate

```
function rotate() {
    var block = currenttype;
    var savedorientation = currentorientation;
  currentorientation = (currentorientation+1) % 4;
    var i;
    var formula = orientations[currentorientation][currenttypenum];
    var col = currentorigin[0];
    var row = currentorigin[1];
    var atc;
    var atr;
    var tests;
    var newcurrent = Array();
    var saved = Array();
    var oksofar = true;

    for (i=0;i<=3;i++) {
          atc = col + formula[i][0];
```


(C) Copyright Nokia 2014. All rights reserved.

```
      if (atc>=(boardcol)) {
           oksofar = false;
           break;       }
      if (atc<0) {
           oksofar = false;
           break;       }
   atr = row + formula[i][1];
      if (atr>=(boardrow)) {
           oksofar = false;
           break;       }
   newcurrent[i]=imagenumber(atc, atr);
     }
   if (oksofar) {
    for (i=0;i<=3;i++) {
        saved[i] = current[i][0];
        document.images[current[i][0]].src = "s40-theme/images/block.png" ;}
    for (i=0;i<=3;i++) {
          tests = String(document.images[newcurrent[i]].src);
          found = tests.search("s40-theme/images/block.png");
          if (found == -1) {
             oksofar = false;
             break;          }
     }
    if (oksofar) {
      for (i=0;i<=3;i++) {
   imagenum=newcurrent[i];
          document.images[imagenum].src = block;
   current[i][0]=imagenum;
   current[i][1] = col+formula[i][0];
   current[i][2] = row+formula[i][1];
       }
      hitDown();
     }
    else {
       for (i=0;i<=3;i++) {
       document.images[saved[i]].src = block;
        }
       currentorientation = savedorientation;
      }
    }
   else {
       currentorientation = savedorientation;
     }
 }
```

FallDown

```
function fallDown() {
 var i;
    var j;
    var imgno;
    var filledcount;
    var tests;
    var found;
    var linesremoved = 0;
```


(C) Copyright Nokia 2014. All rights reserved.

```
    i = boardrow-1;
    while (i>=0) {
        filledcount = 0;
  for (j=boardcol-1;j>=0;j--) {
                  imgno = imagenumber(j,i);
                  tests = String(document.images[imgno].src);
                  found = tests.search("s40-theme/images/block.png");
                  if (found==-1) {
                    filledcount++ ;
                    }
      }
      if (filledcount == boardcol) {
          linesremoved++;
          remove(i);
            }
      else {
      i--;
       }
    }
   if (linesremoved>0) {
       document.f.lines.value = linesremoved + parseInt(document.f.lines.value);
       document.f.score.value =
scoring[linesremoved-1]+parseInt(document.f.score.value);
    }
}

//remove the filled line move the row down
function remove(cut) {
 var upper;
   var colindex;
   var imgno;
   var imgnox;
   for (upper=cut;upper>0;upper--) {
       for (colindex = 0; colindex<boardcol; colindex++) {
         imgno = imagenumber(colindex,upper);
         imgnox = imagenumber(colindex,upper-1);
         document.images[imgno].src = document.images[imgnox].src;
        }
     }
}
```

GameTimer

```
//game main timer
function gameTimer () {
  if (startnewone) {
      if (grace==0) {
         startnewone = false;
         fallDown();
         newBlock();
       }
      else {
       grace--; }
   }
   moveDown();  //move current piece down
```


(C) Copyright Nokia 2014. All rights reserved.

```
}
//next block to come
function newBlock() {
   var type = Math.floor(Math.random()*7);
   var scol = 3;
   makeBlock(type,scol,0);
}

//start the game
function startGame() {
    startnewone=true;
    grace = 0;
    mwl.timer(tid,timeperiod,0,"gameTimer());
}
```

Redesign

Now all basic functions are working. We need to redesign the game layout to convert the game into full functional real game. We need to add some of the Images and stylesheet to create the new layout for the game. change the background and change the borad background.

HTML

```
<body>
 <div class="board">
    <table>
     <tr>
      <td>
       <script language="JavaScript" type="text/javascript">
       createBoard();

       </script>
      </td>
     </tr>
  </table>
   </div>
<div id="card">
  <table>
    <tr>
     <td >
      <div id="score_card">
       <ul>
         <li><h4>Score </h4><p id="score" >00000</p></li>
         <li><h4>Level</h4><p id="level">00000</p></li>
       </ul>
      </div>
       </td>
     <td>
      <div id="next" class="next">
        <script language="JavaScript" type="text/javascript">
          createNextBoard();
          nextBlock(0, 0,0);
       </script>
      </div>
      </td>
     <td>
```


(C) Copyright Nokia 2014. All rights reserved.

```
      <div id="startgame" onclick="startGame()">
       <script language="JavaScript" type="text/javascript">
          playButton();
       </script>

      </div>
      </td>
    </tr>
 </table>
</div>
</body>
```

`startGame()` function is change to pause and resume the game and number of row is changed row for touch and type is 15 row for full touch 19 next block feature is add score and line is add

```
var startgame=false;
var isPasued=false
function startGame() {
    if(!startgame)
     {
     document.getElementById("sgame").src="s40-theme/images/pause.png";
     mwl.timer(tid,timeperiod,0,"gameTimer()");
     startgame=true;
     }
    else {
     document.getElementById("sgame").src="s40-theme/images/play.png";
     mwl.stopTimer(tid);
     startgame=false;
     isPasued=true;
       }
    if(!isPasued)
      startnewone=true;
    else
   startnewone=false;
    grace = 0;
   // mwl.timer(tid,timeperiod,0,"gameTimer()");

   }
```

Two variables `startgame isPasued` are initialised as false when click on the play icon icon change to pause and `startgame` is true. `isPasued` is used to resume the game

Onscreen touch

Onscreen Touch is control the game by touch the sized of the board.the board is divied into four parts top, bottom, left, right Top Touch the top of the board to rotate the block Bottom Touch bottom of the board to move the block one step faster Left Touch left move the block one step left Right Touch right move the block one step right

to achive this onscreen touch controller change `createBoard()` function

```
//create Game Board
function createBoard() {
 var i;
 var j;
 for (i=0; i<boardrow; i++) {
  for (j=0; j<boardcol; j++) {
```


(C) Copyright Nokia 2014. All rights reserved.

```
  if(i<=3)
      document.write("<img src='s40-theme/images/block.png' onclick='rotate()'/>");
  else if(i>3 && i<=13 && j<5)
     document.write("<img src='s40-theme/images/block.png' onclick='moveOver(-1)'/>");
  else if(i>3 && i<=13 && j>=5)
     document.write("<img src='s40-theme/images/block.png' onclick='moveOver(1)'/>");
  else if(i>11)
     document.write("<img src='s40-theme/images/block.png' onclick='moveDown()'/>");

  }
  document.write("<br/>");
   }


}
```

Navigation

Create the complate flow for the the same start up to real game look create menu navigate throught the menu to game and game to menu S40 Block has three menus

- Play - Go to the game screen
- High score - hightscore screen
- Credit - Thanks screen

for this we need to change the html file and include some java script file

```
<div id="menu">
   <div id="title">
        <h1>S40 Blocks</h1>
    </div>
  <div id="menuitem">

  <div class="button"  id="play" onclick="showGame()"><h2>Play</h2></div><br/>
  <div class="button"  id="hscore"
onclick="mwl.show('#hiscore');mwl.hide('#menuitem')"><h2>High Score</h2></div><br/>
  <div class="button"  id="credit"
onclick="mwl.show('#scredit');mwl.hide('#menuitem')"><h2>Credit</h2></div><br/>
 </div>
 <div id="hiscore" onclick="mwl.show('#menuitem');mwl.hide('#hiscore')">
  <h2> High Score</h2>
  <br/>
  <h3> Score :
   <script language="JavaScript" type="text/javascript">
    var score =widget.preferences["s40block"];
      if(score==null)
      {
     //set value
     widget.preferences.setItem("s40block","0,0");
      }
     else
      {
        score =widget.preferences["s40block"]
        var n= score.split("_");
     document.write(n[0]);
      }
   </script>
```


(C) Copyright Nokia 2014. All rights reserved.

```
    </h3>
    <h3> Lines :
     <script language="JavaScript" type="text/javascript">
      var score =widget.preferences["s40block"];
        if(score==null)
        {
       //set value
       widget.preferences.setItem("s40block","0,0");
        }
         else{
          score =widget.preferences["s40block"]
          var n= score.split("_");
       document.write(n[1]);
        }
    </script>
    </h3><br/><br/><br/>   <br/><br/><br/>
    <p>Touch here to go back</p>
   </div>
   <div id="scredit" onclick="mwl.show('#menuitem');mwl.hide('#scredit')">
      <h2>Design and Program </h2>
      <p>by</p>
      <h3>Senthil Kumar (lightwind)</h3><br/>
      <h2>special Thanks </h2><br/>
      <big>Nokia Web SDK 2.0</big><br/>
      <h4>Developer team</h4>
      <br/><br/><br/>
    <p>Touch here to go back</p>
   </div>
</div>

<div id="game">
 <div id="board">
  <table>
      <tr>
    <td>
     <script type="text/javascript">
        createBoard();
     </script>
    </td>
   </tr>
  </table>
 </div>
  <div id="card">
   <table>
     <tr>
      <td >
       <div id="score_card">
       <ul>
         <li><h4>Score </h4><p id="score" >0</p></li>
          <li><h4>Line</h4><p id="line">0</p></li>
        </ul>
       </div>
        </td>
       </tr>
       <tr>
      <td>
       <div id="next" class="next">
```


(C) Copyright Nokia 2014. All rights reserved.

```
      <br/>
         <img id="nextimg" src="s40-theme/images/block.png" alt="play  pause" />

      </div>
      </td>
     </tr>
     <tr>
    <td>
      <div id="startgame" onclick="startGame()">
         <img id="sgame" src="s40-theme/images/play.png" alt="play  pause" />
      </div>
     </td>
    </tr>
  </table>
    </div>
 </div>
```

we create menu menu item hiscore credit "
" we use `mwl.show()` and `mwl.hide()` for navigation see the screenshot in the gallery

```
<script language="JavaScript" type="text/javascript">
     function init()
     {
      mwl.show("#menu");
      mwl.hide("#hiscore");
      mwl.hide("#scredit");
      mwl.hide("#game");
      document.s.value="0";
     }
     function showGame()
     {
     mwl.show("#game");
     mwl.hide("#menu");
     }
   function closeGame()
   {
    mwl.show("#menu");
    mwl.hide("#game");
   }


   </script>
```

Finally we add nextBlock included first time of the game.we generate nextblock and current block after that we generate nextblock only we swape nextblock to current block

```
function newBlock() {

   var scol = 3;
   if(!initblock)
  {
    ntype= Math.floor(Math.random()*7);
    type= Math.floor(Math.random()*7);
    document.getElementById("nextimg").src=nextimages[ntype];
    initblock=true;
```


(C) Copyright Nokia 2014. All rights reserved.

```
  }
   else {
    type=ntype;
    ntype= Math.floor(Math.random()*7);
    document.getElementById("nextimg").src=nextimages[ntype];
 }
   makeBlock(type,scol,1);
 }
```

CSS

```
@charset "utf-8";

.hide {
    display: none;
}

.show {
    display: block;
}

*{
 margin: 0px;
 padding: 0px;
}

body {
 font-size:small;
 background-image:url(../images/bg.png);
 background-repeat: repeat-x;
 overflow: hidden;
}

h2{
 font-weight:normal;
}
#menu,#game{
 width:240px;
 height: 400px;
 margin-top: 0px;
 }
#board{
 margin-left: 2px;
 margin-top: 2px;
 overflow: hidden;
    text-decoration: none;
    border-top-width: 1px;
    border-right-width: 1px;
    border-bottom-width: 1px;
    border-left-width: 1px;
    border-top-style: solid;
    border-right-style: solid;
    border-bottom-style: solid;
    border-left-style: solid;
    border-top-color: #ae9b75;
```


(C) Copyright Nokia 2014. All rights reserved.

```
    border-right-color: #ae9b75;
    border-bottom-color: #ae9b75;
    border-left-color: #ae9b75;
 }
#card{
 margin-left: 185px;
 margin-top: -240px;
 overflow: hidden;

 }
.next{
 width: 50px;
 height: 50px;
 background-image:url(../images/block.png);
 background-repeat: repeat;
 }
#score_card {
 width:80px;
 }
.button {
   background-image: url(../images/button.png);
      background-repeat: repeat-x;
      background-position: left top;
      height: 30px;
      width:150px;
      padding-right: 8px;
      padding-left: 8px;
      color: #000;
      text-align:center;
      text-decoration: none;
      border-top-width: 1px;
      border-right-width: 1px;
      border-bottom-width: 1px;
      border-left-width: 1px;
      border-top-style: solid;
      border-right-style: solid;
      border-bottom-style: solid;
      border-left-style: solid;
      border-top-color: #ae9b75;
      border-right-color: #ae9b75;
      border-bottom-color: #ae9b75;
      border-left-color: #ae9b75;
 }
#menuitem{
 margin-left: 40px;
 margin-top: 50px;
 }
#title{
 margin-left: 20px;
 margin-top: 100px;
 text-align: center;
 }
#hiscore,#scredit,#gover{

 width: 240px;
 height: 140px;
 margin-left:0px;
```


(C) Copyright Nokia 2014. All rights reserved.

```
 margin-top: 50px;
 text-align: center;
 border-top-width: 2px;
    border-right-width: 2px;
    border-bottom-width: 2px;
    border-left-width: 2px;
    border-top-style: solid;
    border-right-style: solid;
    border-bottom-style: solid;
    border-left-style: solid;
    border-top-color: #ae9b75;
    border-right-color: #ae9b75;
    border-bottom-color: #ae9b75;
    border-left-color: #ae9b75;
 }
```

Final Touch

Finally we need to add the score and store the high score in the persistent data. we need to store **score and line** in the data Series 40 persistent database some restriction of max 256 character only allowed so we need to main that limit.

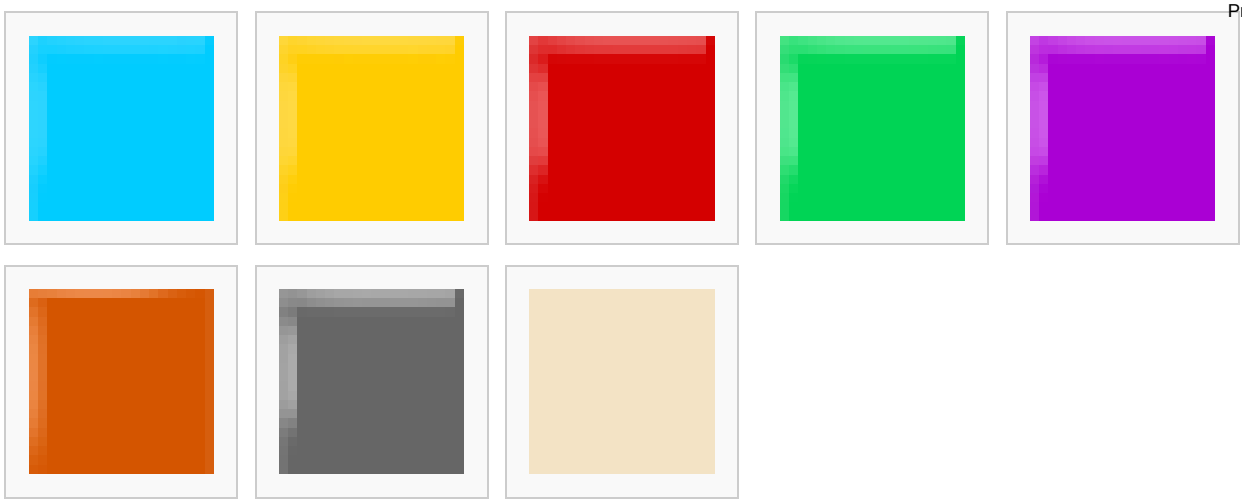
```
function backToMenu()
 {
  mwl.stopTimer(tid);
  var ss =widget.preferences["s40block"];
  var a=ss.split("_");
  if(s>a[0])
  {
   var sp= s+"_"+l;
   widget.preferences.setItem("s40block",sp);
  }
   //createBoard();
      mwl.show("#menu");
      mwl.hide("#game");


 }
 function gameOver()
 {
  var ss =widget.preferences["s40block"];
  var a=ss.split("_");
  if(s>a[0])
  {
   var sp= s+"_"+l;
   widget.preferences.setItem("s40block",sp);
  }
  s=l=0;
  mwl.show("#gover");
  mwl.hide("#game");
 }
```

`gameOver()` check the previews score with current if its greater store the line and score to Series 40 block key. in the format of `score_line` "_" as delimiter.

Resources

(C) Copyright Nokia 2014. All rights reserved.



ScreenShot

Starting structure of the game

Initial game play

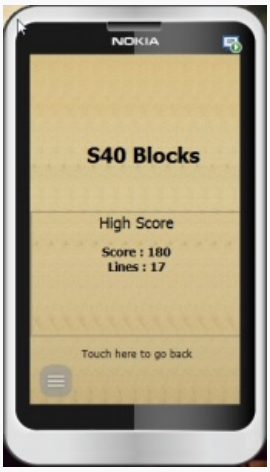
Game over condition check

Game redesigned

Redesigned game play

Menu screen

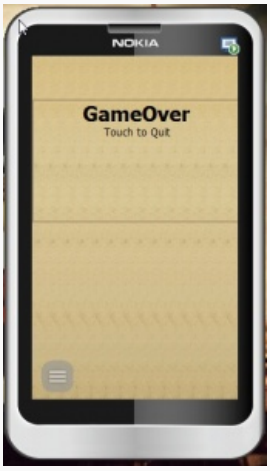
(C) Copyright Nokia 2014. All rights reserved.



Score

Credit

Game play



Game over

Video

Game play video_1

(C) Copyright Nokia 2014. All rights reserved.

Game play video_2

Summary

I thank to Nokia Web Tools development team. Some of the nice feature is include in Series 40 Web browser 2.0 is good.

In this project I used lot of functional in mwl class. I can't explain the functions fully due to time sorry for that but great thank to wiki admin. I don't know how to embed the video link in this page sorry for that just click and watch the gameplay

(C) Copyright Nokia 2014. All rights reserved.