DESIGN AND IMPLEMENTATION OF GAMES BASED ON HTML5

Cong Li[1, a], Lingjun yang[2, b] and Sanxing Cao[2, c]

[1,2,3] School of Communication University of China, Beijing 100024, China;

[a] licongcuc@163.com, [b] yanglj@cuc.edu.cn, [c] caosx_cuc@126.com

Keyword：HTML5 web games canvas

Abstract. With the development of HTML5, its convenient and quick user has been recognized by most users. Some games can be made by HTML5, which is convenient for its communication. This paper introduces the traditional Tetris game made by HTML5, shows the design scheme and the main code and proves that this design can run successfully.

BACKGROUND

Some of the new elements of HTML5 such as canvas, audio, video and API for the users to label and corresponding to the images, audios, videos processing brings great convenience, on the other hand ,compared with the previous HTML language, HTML5 code writing is clearer and more simple. In these elements ,the most convenient and frequently is the canvas element for use，the application of this element can be controlled through the JavaScript code instead of flash, improve the performance and speed of the system. In addition, because of its small size code, loading speed, and canvas element for each bowers has good compatibility, the development of canvas has also been used in Web games. We designed this application with six canvas elements of HTML5 to implement different function modules, compared with the traditional game mode, implementation based on HTML5 is smaller, spread and application more convenient, Players only need to open the web to play games in the case of the network.

GAME DESIGN

1） Functional Requirements and Analysis

The function of this design is basically the same as the function of the traditional Tetris game, and realized six function modules：Results module, time module, the next block of the shape preview module, game module, pause module and start module. The results module is used to show score of the players in the game process, the starting score set to 0 and no maximum score. The user eliminates the full line of the box, the score plus 10 points; Time module is used to record the time of the game; the next block shape preview module is used to alert the player to the shape of the next block to facilitate the player to prepare put the box in the suitable position; pause module to pause the game; the starting module allows the player to start a game or restart a new round at the pause mode or at the end of the game; the game module is the module which provides the game to the players in this design.

2） The Analysis of The View Interface

The whole page is set to the distance of left and right margins of 5%, the middle of the page is the game module, the height of this module is set to 70% of the display device, the top margin is 5% and the width of the module is 40% of the width of the display device; the result module and time module are displayed on the left of the game module, two modules width are set to 20%, height are set to 10% and the distance of the game module is 5%. The distance between the top of the score module and the top of the monitor is 10%，The time module is located below 20% of the results module; In the right side of the game module a block preview module and a suspended module are respectively set which are symmetric to the two modules of the left; below the game module 5% is set to start the module button, the same width is set to 20%, height is set to 10%.

3） Realization of The Game

This design uses the elements of canvas, each module in different canvas and the function of these modules are realized by JavaScript operation DOM.

1） Game Module Implementation

We divided the canvas of the game module into 40 rows and 20 columns and defined as a two-dimensional array, this two-dimensional array is used to store each area with 1 represented of the box and 0 represented none of the box. When the value of the array is 1, it can be eliminated and the score increases.

The 4 small squares of different shapes of the canvas are draw according to the relative coordinates of the upper left corner of the canvas, the upper left corner of the canvas is (0,0), the coordinates and the deformed coordinates are stored in the array. The following array is a shape of the initial coordinates and the coordinates of various shapes after transformation.

```
    var shap1=[[0,0,1,0,1,1,1,2],[0,1,1,1,2,1,2,0],[0,0,0,1,0,2,1,2],[0,0,1,0,2,0,0,1]];
```

When drawing the canvas need to draw according to actual coordinates, the width and height of each square grid is set to 20 pixel value. The following code gives a method for drawing the initial graphics.

```
    fillRect(0,0,20,20)；fillRect(20,0,20,20)；fillRect(20,20,20,20)；fillRect(20,40,20,20)
```

Each one will move down the graphic drawing of the removal of the original shape, after the completion of the 4 small grid, the new Draw method is called to start drawing a new shape, and vertical coordinates plus 1，the code of the Draw method is shown below：

```
    function Draw(){ var i=0; var tempY=0; shapeXY=new Array(4);
for(i=0;i<4;i++){DrawRect((t[i*2]+rectX) *unitLen,(t[i*2+1]+rectY)*unitLen); var
row=new Array(2); row[0]=(t[i*2+1]+rectY); row[1]=t[i*2]+rectX;
shapeXY[i]=row;if(topTrue==true) { if(tempY<(t[i*2+1]+rectY)){tempY=t[i*2+1]+rectY;
shapeHeight=tempY+1; } } rectY+=1;}
```

In the process of falling the boxes, when the boxes reach the bottom of the canvas, they can not move down again. Judging whether to reach the bottom of the method with a figure four cubes any longitudinal coordinate to reach 40 cell is determined to get to the bottom. The judgment method is as follows.

```
    if(shapeXY[0][0]==39||shapeXY[1][0]==39||shapeXY[2][0]==39||shapeXY[3][0]==39)
{ CurrentShapeOnBottom();return true; }
```

In addition to the bottom, when the next line of the row of these box is full of boxes, this box can not move down again.The judgment method is as follows：

```
    if((resultArray[shapeXY[0][0]+1][shapeXY[0][1]]+resultArray[shapeXY[1][0]+1][shapeXY[1][1]]+
resultArray[shapeXY[2][0]+1][shapeXY[2][1]]+resultArray[shapeXY[3][0]+1][shapeXY[3][1]] >=1) )
{CurrentShapeOnBottom();return true; }
```

When the graphics moving left, the horizontal coordinate minus one and vertical coordinate remains unchanged and when the graphics moving left，the horizontal coordinate add one and vertical coordinate remains unchanged. When the square get around the sidebar, the box can not move again.

When the entire canvas is covered with square the game is over. To redrawing the canvas, we need to clear the canvas by the method of clearRect() instead of clearing squares one by one. The following codes illustrates these.

```
    for(vari=0;i<40;i++){for(varj=0;j<20;j++){ if(resultArray[i][j]==1){ DrawRect(j*unitLen,i*unitLen); }
}}
```

We use the direction keys on the board to control the operation of the boxes in the game, such as change the shape of the box, the direction of the movement of the box and the whereabouts of the speed. Therefore, the recognition of the key is the important part of the design, the identification of the specific button is as follows:

```
var keyCode = event.which || event.keyCode
```

When we got the value of the keyCode, the key judgment is followed by switch statement, and the corresponding operation is carried out accordingly the value. The four buttons for the role of: Push button to show deformation, left button to show shift left, right button to show shift right and down button indicates acceleration drop. The operation of the left and right button is in judging whether reached the boundary of the canvas and transform of the coordinate according to the result. The deformation button is set to 0 after the deformation reaches 3 times, which indicates the initial state of the deformation. Down button by drawing the small boxes more quickly and the vertical coordinate plus 1 to add the speed of the player is accelerated. The following code is the acceleration statement.

```
setInterval("DrawTetris();",100)
```

The speed is accelerated when the speed of the draw gird from 200ms to 100ms.DrawTetris() method is the way to drawing the small boxes.

2） Time Module Implementation

A time variable with an initial value of 0 is defined and calculate the time interval by setting a time clock,
and everytime the variable for the time clock will increase 1 in per second ,the following code is the main calculation of the timer：

```
this.timerId = setInterval(function(){self.time += 1; self.render();}, 1000);
```

Before presenting to the player, the time display format is set to hours: Minutes: seconds format. The following code gives the display format:

```
return hours + ':' + minutes + ':' + seconds;
```

3） The Realization of the Next Block of The Block Preview

The next preview is the same way as the game module is implemented, calling the same method as the method of the game module to draw the small boxes, the small box drawn on this canvas ,players can achieve a preview of the next box function.

4） Start Module Implementation

Start button calling the init（） method by JavaScript operate the DOM, and add buttons to the entire game canvas –“keydown” event, callback function is changeShape() to transform the shape of the boxes.

```
document.addEventListener('keydown',moveShape,false);
```

In the init () method, the effect of the grid movement is achieved by drawing a square from the timer per 200ms.

```
setInterval("DrawTetris();",200);
```

5） Suspension Module Implementation

The pause module is eliminate the rendering effect of drawing the small square by calling clearInterval("DrawTetris();",200), That is the effect of the players for the small grid pause, no longer moving down and show the effect of the suspension.

6） Achievement Module Implementation

Through the judgment can be obtained when the line can be eliminated, while the results of the module update，

```
updateScore: function(lines){ this.total += this.calcScore(lines); this.render();}
```

By calling the drawText method, the total score displayed in the corresponding canvas of the module.

CONCLUSION

In this design we displayed the traditional Tetris game through the HTML5, and it can be spread through the web, without players installation of this game, and is more convenient and quick. There are still a lot to improve the design, for example, to join the localstorage functions, to achieve the game results and levels stored in local, these functions will convenient users continue to play games next time.

REFERENCE

[1] Xing Xiaopeng. Research and Value Analysis of HTML5 Core Technology[J]. Value Engineering, 2011,22:157-158.

[2] Chu Jiewang, Huang Jieqin. HTML5 and Mobile Information Service[J]. Theory and exploration, 2013,7(36):24-27.

[3] Zheng Yifan. Web game design based on HTML5 Technology[J]. Industry focus,2015:49-50

[4] Liu Yakui. Mobile Application of News Based on HTML5[D]. Beijing: Beijing University of Posts and Telecommunications.2013

[5] Zheng Hong. HTML5 and New Media Application[J].Chinese Media Technology,2013