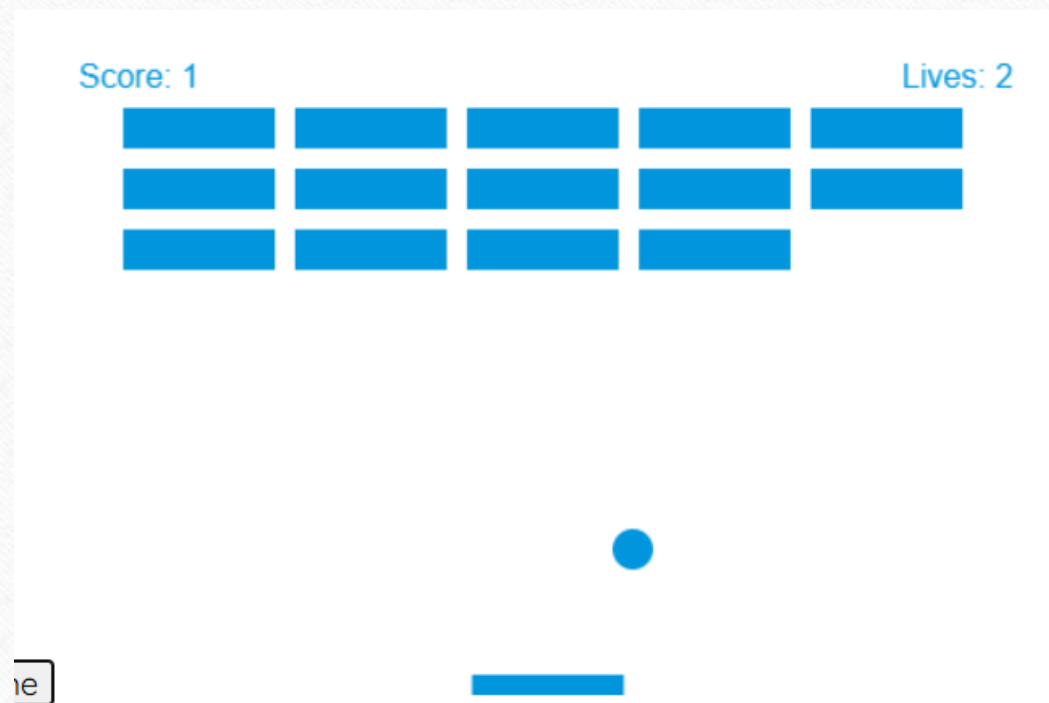


網際網路管理

期末報告

40823238 吳容輝

打磚塊遊戲



設計概要

- 設計一個簡單的打磚塊遊戲，透過計分與生命值的設計，讓遊戲更有趣

程式碼(html 部分)

```
brick_game_ver.2.html Raw
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8" />
5 <title>brick game</title>
6 <style>
7 * {
8     padding: 0;
9     margin: 0;
10 }
11 canvas {
12     background: #eee;
13     display: block;
14     margin: 0 auto;
15 }
16 #startButton {
17     display: block;
18     margin: 20px auto;
19     padding: 10px 20px;
20     background-color: #4c63af;
21     color: white;
22     border: none;
23     border-radius: 5px;
24     cursor: pointer;
25 }
26 </style>
27 </head>
28 <body>
29 <!-- 遊戲開始按鈕 -->
30 <button id="startButton">Start Game</button>
31 <!-- 遊戲畫布 -->
32 <canvas id="myCanvas" width="480" height="320"></canvas>
33
34 <script>
```

Javascript 部分(宣告)

```
<script>
var canvas = document.getElementById("myCanvas");
var ctx = canvas.getContext("2d");
var ballRadius = 10; // 球半徑
var x = canvas.width/2; // 球初始x座標
var y = canvas.height-30; // 球初始y座標
var dx = 2; // 球x軸速度
var dy = -2; // 球y軸速度
var paddleHeight = 10; // 擋板高
var paddleWidth = 75; // 擋板寬
var paddleX = (canvas.width-paddleWidth)/2; // 擋板初始x座標
var rightPressed = false; // 是否按下右鍵
var leftPressed = false; // 是否按下左鍵
var brickRowCount = 5; // 磚塊行數
var brickColumnCount = 3; // 磚塊列數
var brickWidth = 75; // 磚塊寬度
var brickHeight = 20; // 磚塊高度
var brickPadding = 10; // 磚塊間距
var brickOffsetTop = 30; // 磚塊距頂部的偏移量
var brickOffsetLeft = 30; // 磚塊距左側的偏移量
var score = 0; // 初始分數
var lives = 3; // 初始生命值
var bricks = []; // 磚塊陣列
var gameStarted = false; // 遊戲是否已開始
```

初始化磚塊陣列 與 輸入偵測(鍵盤與滑鼠)

```
// 初始化磚塊陣列
for(var c=0; c<brickColumnCount; c++) {
    bricks[c] = [];
    for(var r=0; r<brickRowCount; r++) {
        bricks[c][r] = { x: 0, y: 0, status: 1 };
    }
}

// 鍵盤與滑鼠偵測
document.addEventListener("keydown", keyDownHandler, false);
document.addEventListener("keyup", keyUpHandler, false);
document.addEventListener("mousemove", mouseMoveHandler, false);

// 開始按鈕偵測
document.getElementById("startButton").addEventListener("click", startGame);
```


開始遊戲 與 輸入控制(鍵盤與滑鼠)

```
74 // 開始遊戲
75 function startGame() {
76     if (!gameStarted) {
77         gameStarted = true;
78         draw();
79     }
80 }
81
82 // 鍵盤控制
83 function keyDownHandler(e) {
84     if(e.key == "Right" || e.key == "ArrowRight") {
85         rightPressed = true;
86     }
87     else if(e.key == "Left" || e.key == "ArrowLeft") {
88         leftPressed = true;
89     }
90 }
91
92 function keyUpHandler(e) {
93     if(e.key == "Right" || e.key == "ArrowRight") {
94         rightPressed = false;
95     }
96     else if(e.key == "Left" || e.key == "ArrowLeft") {
97         leftPressed = false;
98     }
99 }
100
101 // 滑鼠控制
102 function mouseMoveHandler(e) {
103     var relativeX = e.clientX - canvas.offsetLeft;
104     if(relativeX > 0 && relativeX < canvas.width) {
105         paddleX = relativeX - paddleWidth/2;
106     }
107 }
108
```

碰撞 與 勝利條件 偵測

```
109 // 碰撞與勝利偵測
110 function collisionDetection() {
111     for(var c=0; c<brickColumnCount; c++) {
112         for(var r=0; r<brickRowCount; r++) {
113             var b = bricks[c][r];
114             if(b.status == 1) {
115                 if(x > b.x && x < b.x+brickWidth && y > b.y && y < b.y+brickHeight) {
116                     dy = -dy;
117                     b.status = 0;
118                     score++;
119                     if(score == brickRowCount*brickColumnCount) {
120                         alert("YOU WIN, CONGRATS!");
121                         document.location.reload();
122                     }
123                 }
124             }
125         }
126     }
127 }
128
```


繪製遊戲要素(1)

```
129 // 畫球
130 function drawBall() {
131     ctx.beginPath();
132     ctx.arc(x, y, ballRadius, 0, Math.PI*2);
133     ctx.fillStyle = "#0095D0";
134     ctx.fill();
135     ctx.closePath();
136 }
137
138 // 畫擋板
139 function drawPaddle() {
140     ctx.beginPath();
141     ctx.rect(paddleX, canvas.height-paddleHeight, paddleWidth, paddleHeight);
142     ctx.fillStyle = "#0095D0";
143     ctx.fill();
144     ctx.closePath();
145 }
146
147 // 畫磚塊
148 function drawBricks() {
149     for(var c=0; c<brickColumnCount; c++) {
150         for(var r=0; r<brickRowCount; r++) {
151             if(bricks[c][r].status == 1) {
152                 var brickX = (r*(brickWidth+brickPadding))+brickOffsetLeft;
153                 var brickY = (c*(brickHeight+brickPadding))+brickOffsetTop;
154                 bricks[c][r].x = brickX;
155                 bricks[c][r].y = brickY;
156                 ctx.beginPath();
157                 ctx.rect(brickX, brickY, brickWidth, brickHeight);
158                 ctx.fillStyle = "#0095D0";
159                 ctx.fill();
160                 ctx.closePath();
161             }
162         }
163     }
164 }
```

繪製遊戲要素(2)

```
166 // 畫分數
167 function drawScore() {
168     ctx.font = "16px Arial";
169     ctx.fillStyle = "#0095DD";
170     ctx.fillText("Score: "+score, 8, 20);
171 }
172
173 // 畫生命值
174 function drawLives() {
175     ctx.font = "16px Arial";
176     ctx.fillStyle = "#0095DD";
177     ctx.fillText("Lives: "+lives, canvas.width-65, 20);
178 }
179
180 function draw() {
181     ctx.clearRect(0, 0, canvas.width, canvas.height);
182     drawBricks();
183     drawBall();
184     drawPaddle();
185     drawScore();
186     drawLives();
187     collisionDetection();
188 }
```

遊戲結束 偵測

```
189     if(x + dx > canvas.width-ballRadius || x + dx < ballRadius) {
190         dx = -dx;
191     }
192     if(y + dy < ballRadius) {
193         dy = -dy;
194     }
195     else if(y + dy > canvas.height-ballRadius) {
196         if(x > paddleX && x < paddleX + paddleWidth) {
197             dy = -dy;
198         }
199         else {
200             lives--;
201             if(!lives) {
202                 alert("GAME OVER");
203                 document.location.reload();
204             }
205             else {
206                 x = canvas.width/2;
207                 y = canvas.height-30;
208                 dx = 3;
209                 dy = -3;
210                 paddleX = (canvas.width-paddleWidth)/2;
211             }
212         }
213     }
214
215     if(rightPressed && paddleX < canvas.width-paddleWidth) {
216         paddleX += 7;
217     }
218     else if(leftPressed && paddleX > 0) {
219         paddleX -= 7;
220     }
221
222     x += dx;
223     y += dy;
224     requestAnimationFrame(draw);
```