

HTML Canvas Game



22/11/16 2020975014 김상현

1

원본 게임 분석

<https://codepen.io/jmt4k8awngdw1rki3pln/pen/rNKzzyW>

<https://codepen.io/jmt4k8awngdw1rki3pln/full/rNKzzyW>

2

수정할 점

1. jQuery 라이브러리를 사용한다
2. 사용을 비 권장하는 var 변수를 사용한다
3. Deprecated 된 기능을 사용한다
4. 난이도가 없다
5. 장애물이 단방향으로만 움직인다
6. 다양한 효과 (색, 효과음)이 없다
7. 설명이 영어이다
8. element를 js 로 생성하여 검색엔진 최적화가 안된다
9. 조작을 방향키로만 할 수 있어 플레이어의 선택권이 적다
10. 위치를 초기화 할 수 없다
11. 게임 오버 횟수를 알 수 없다

3

수정 사항

1. jQuery 라이브러리 의존을 제거함

```
index.html
@@ -3,7 +3,6 @@
3 3 <head>
4 4 <meta charset="UTF-8">
5 5 <title>Game</title>
6 - <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
7 6 <script src="script.js"></script>
8 7 <link href="style.css" rel="stylesheet" />
9 8 </head>

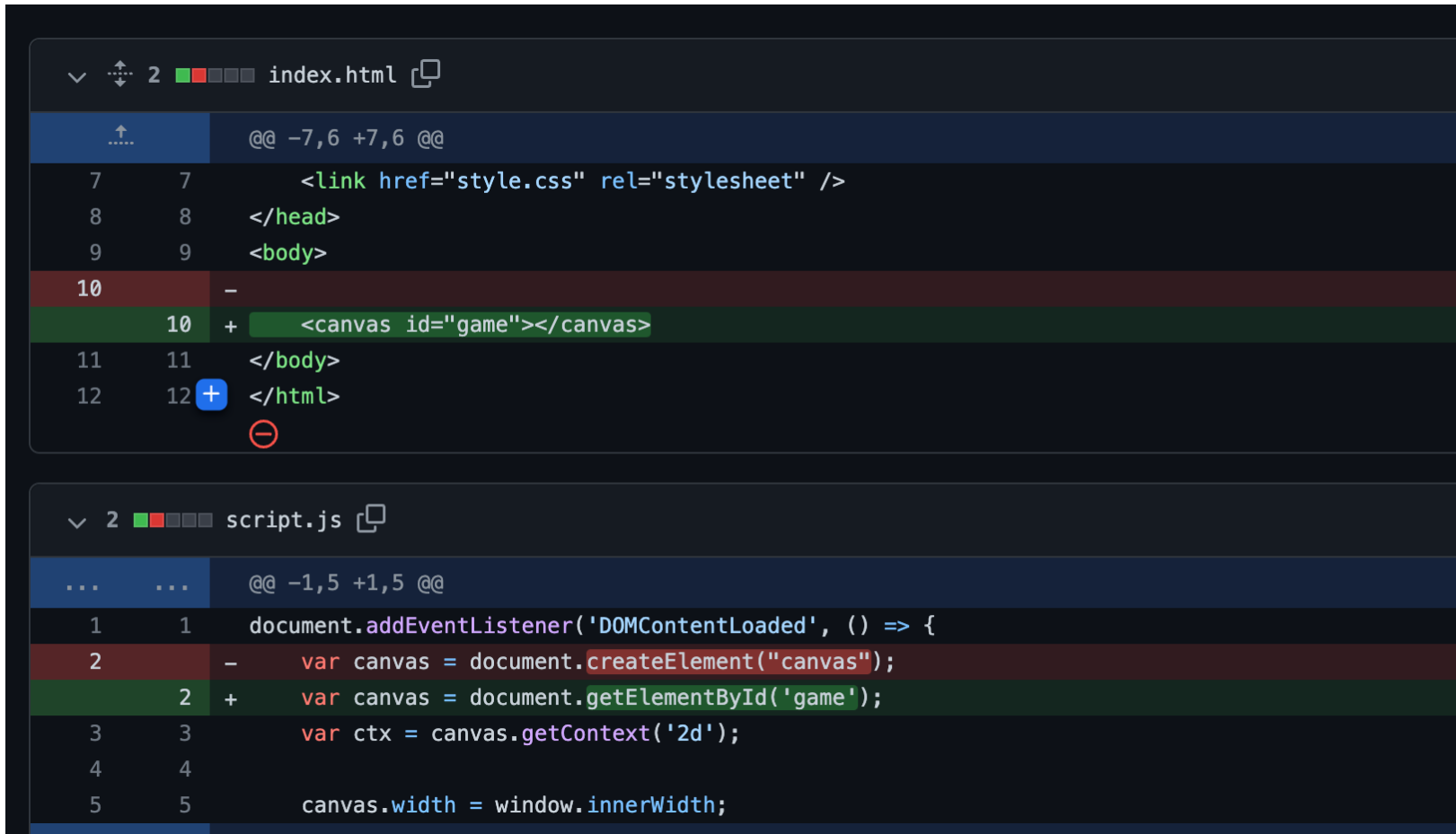
script.js
@@ -1,4 +1,4 @@
1 - $(function () {
1 + document.addEventListener('DOMContentLoaded', () => {
2 2     var canvas = document.createElement("canvas");
3 3     var ctx = canvas.getContext('2d');
4 4

@@ -269,5 +269,4 @@ $(function () {
269 269     update();
270 270     draw();
271 271 }, 1000 / 60);
```

3

수정 사항

2. Canvas등 element를 html 파일에 추가함



The screenshot displays a code editor with two files: index.html and script.js. The index.html file shows a diff view where a new canvas element is added to the body. The script.js file shows a diff view where the canvas element is retrieved by ID and its width is set to the window's inner width.

```
index.html
@@ -7,6 +7,6 @@
7      <link href="style.css" rel="stylesheet" />
8      </head>
9      <body>
10     -
10     + <canvas id="game"></canvas>
11     </body>
12     </html>
```

```
script.js
@@ -1,5 +1,5 @@
1      document.addEventListener('DOMContentLoaded', () => {
2      -   var canvas = document.createElement("canvas");
2      +   var canvas = document.getElementById('game');
3      -   var ctx = canvas.getContext('2d');
4      -
5      -   canvas.width = window.innerWidth;
```

3

수정 사항

3. Canvas가 가운데 정렬 되도록 함

```
style.css
@@ -3,12 +3,15 @@
3      3      margin: 0;
4      4      }
5      5
6      6      - html, body, canvas {
7      7      -      width: 100%;
8      8      -      height: 100%;
9      9      + html, body {
10     10      background: #000;
11     11      }
12     12      canvas {
13     13      -      display: block;
14     14      +      position: absolute;
15     15      +      top: 0;
16     16      +      bottom: 0;
17     17      +      left: 0;
18     18      +      right: 0;
19     19      +      margin: auto;
20     20      }
21     21      }
```

3

수정 사항

4. html파일의 언어를 한국어로 변경하고, 설명을 번역함

```
index.html
... @@ -1,5 +1,5 @@
1 1 <!DOCTYPE html>
2 - <html lang="en">
+ <html lang="ko">
3 3 <head>
4 4 <meta charset="UTF-8">
5 5 <title>Game</title>
```

```
script.js
@@ -255,11 +255,11 @@ document.addEventListener('DOMContentLoaded', () => {
255 255 ctx.font = "14px Verdana";
256 256
257 257 if (player.isDead()) {
258 - ctx.fillText("Game over!", 10, 50);
259 - ctx.fillText("Press [SPACE]", 10, 70);
+ 258 + ctx.fillText("게임 오버!", 10, 50);
+ 259 + ctx.fillText("다시 시작: [SPACE]", 10, 70);
260 260 } else {
261 - ctx.fillText("Cross to the other side", 10, 20);
262 - ctx.fillText("Use keyboard arrows", 10, 40);
+ 261 + ctx.fillText("장애물을 피해 반대편으로 건너가기", 10, 20);
+ 262 + ctx.fillText("키보드의 방향키로 이동 가능", 10, 40);
263 263 }
264 264 }
265 265
```

3

수정 사항

5. var 대신 const와 let을 사용하도록 수정함

```

13 - var player = (function () {
14 -   var x = 100,
13 +   const player = (function () {
14 +     let x = 100,

```

```

79 79
80 - var blocks = (function () {
81 -   var blocks = [],
80 +   const blocks = (function () {
81 +     let blocks = [],

```

```

oveAll: function () {
  var px = player.getX(),
  let px = player.getX(),
    py = player.getY(),
    pw = player.getW(),
    ph = player.getH();

8,7 @@ document.addEventListener('DOMCont

  return;
}

  var len = blocks.length;
  for (i = 0; i < len; ++i) {
  for (let i = 0; i < blocks.length; ++i)
    blocks[i].y += blocks[i].speed;
    if (blocks[i].y > canvas.height)
      blocks[i].y = 0;

```


3

수정 사항

6. 게임 오버 횟수를 카운트하여 표시함

```
script.js
@@ -15,7 +15,8 @@ document.addEventListener('DOMContentLoaded', () => {
15 15      y = canvas.height / 2,
16 16      w = 10,
17 17      h = 10,
18 18      dead = false
19 19      dead = false,
20 20      death = 0
21 21      return {
22 22      getX: function () {
@@ -36,6 +37,11 @@ document.addEven
36 37
36 37
37 38      die: function () {
38 39          dead = true;
39 40          death++;
40 41      },
41 42      getDeath: function() {
42 43          return death;
43 44      },
44 45      resurrect: function () {
45 46
46 47
47 48
@@ -260,6 +266,7 @@ document.addEventListener('DOMContentLoaded', () => {
260 266      } else {
261 267      ctx.fillText("장애물을 피해 반대편으로 건너가기", 10, 20);
262 268      ctx.fillText("키보드의 방향키로 이동 가능", 10, 40);
263 269      ctx.fillText("죽은 횟수 : " + player.getDeath(),10,100);
264 270      }
```

3

수정 사항

7. 오브젝트와 텍스트의 색을 추가함

```
script.js

@@ -105,7 +105,7 @@ document.addEventListener('DOMContentLoaded', () => {
    },
    05
    06
    07     draw: function (b) {
    08 +       ctx.fillStyle = "#ffffff";
    09 +       ctx.fillStyle = "#5472d3";
    10         ctx.fillRect(b.x, b.y, b.w, b.h);
    11     },

    @@ -264,6 +264,7 @@ document.addEventListener('DOMContentLoaded', () => {
    64         ctx.fillText("게임 오버!", 10, 50);
    65         ctx.fillText("다시 시작: [SPACE]", 10, 70);
    66     } else {
    67 +       ctx.fillStyle = "#5472d3";
    68         ctx.fillText("장애물을 피해 반대편으로 건너가기", 10, 20);
    69         ctx.fillText("키보드의 방향키로 이동 가능", 10, 40);
    70         ctx.fillText("죽은 회수 : " + player.getDeath(), 10, 100);
```

3

수정 사항

8. 게임 오버 시 오브젝트와 텍스트의 색상이 붉은색으로 변하도록 함

```
4 script.js

@@ -105,7 +105,8 @@ document.addEventListener('DOMContentLoaded', () => {
105     },
106
107     draw: function (b) {
108 -         ctx.fillStyle = "#5472d3";
108 +         if(player.isDead()) ctx.fillStyle = "#d50000";
109 +         else ctx.fillStyle = "#5472d3";
110         ctx.fillRect(b.x, b.y, b.w, b.h);
111     },
112
@@ -261,6 +262,7 @@ document.addEventListener('DOMContentLoaded', () => {
262     ctx.font = "14px Verdana";
263
264     if (player.isDead()) {
265 +         ctx.fillStyle = "#d50000";
266         ctx.fillText("게임 오버!", 10, 50);
267         ctx.fillText("다시 시작: [SPACE]", 10, 70);
268     } else {
```

3

수정 사항

9. key 입력을 keyCode가 아닌 key로 받도록 함

239	237	updateCtrl();			
240	238				// check the keys and do the movement
241	239	document.body.addEventListener("keydown",	-		if (ctrl.keys[38]) {
242		ctrl.keys[e.keyCode] = true;	190	+	if (ctrl.keys.ArrowUp) {
	240	ctrl.keys[e.key] = true;	191		if (ctrl.velY > -ctrl.speed) {
243	241	});	192		ctrl.velY--;
244	242		193		}
245	243	document.body.addEventListener("keyup", fu	194		}
246		ctrl.keys[e.keyCode] = false;	195		
	244	ctrl.keys[e.key] = false;		-	if (ctrl.keys[40]) {
247	245	});	196	+	if (ctrl.keys.ArrowDown) {
248	246		197	+	if (ctrl.velY < ctrl.speed) {
			198		ctrl.velY++;
			199		}
			200		}
				-	if (ctrl.keys[39]) {
			201	+	
			202	+	if (ctrl.keys.ArrowRight) {

3

수정 사항

10. 게임 오버 시 효과음이 재생되도록 함

2 script.js

```
@@ -41,6 +41,8 @@ document.addEventListener('DOMContentLoaded', () => {
```

41 +

```
42         die: function () {
```

```
43             dead = true;
```

```
44 +             let sound = new Audio('warning-sound.mp3');
```

```
45 +             sound.play();
```

```
46             death++;
```

```
47         },
```

```
48
```

3

수정 사항

11. ESC를 누르면 위치가 초기화 할 수 있도록 함

```
5 script.js
@@ -177,14 +177,12 @@ document.addEventListener('DOMContentLoaded', ()
177     }
178
179     function updateCtrl() {
180         - if (ctrl.keys[32]) {
181             - if (player.isDead()) {
182                 - ctrl.x = 0;
183                 - ctrl.y = canvas.height / 2;
184                 - ctrl.velX = 0;
185                 - ctrl.velY = 0;
186                 - player.resurrect();
187             - }
188         + if (ctrl.keys.Escape) {
189             + ctrl.x = 100;
190             + ctrl.y = canvas.height / 2;
191             + ctrl.velX = 0;
192             + ctrl.velY = 0;
193             + player.resurrect();
194         }
195     }
196 }
```

```
@@ -267,6 +265,7 @@ document.addEventListener('DOMContentLoaded', () =>
    ctx.fillStyle = "#5472d3";
    ctx.fillText("장애물을 피해 반대편으로 건너가기", 10, 20);
    ctx.fillText("키보드의 방향키로 이동 가능", 10, 40);
    ctx.fillText("위치 초기화 : ESC", 10, 60);
    ctx.fillText("죽은 회수 : " + player.getDeath(), 10, 100);
}
```

3

수정 사항

12. i, j, k, l 으로도 조작을 할 수 있도록 함

```
}  
  
if (ctrl.keys.ArrowUp) {  
if (ctrl.keys.ArrowUp || ctrl.keys.i) {  
if (ctrl.velY > -ctrl.speed) {  
ctrl.velY--;  
}  
}  
  
if (ctrl.keys.ArrowDown) {  
if (ctrl.keys.ArrowDown || ctrl.keys.k) {  
if (ctrl.velY < ctrl.speed) {  
ctrl.velY++;  
}  
}  
  
if (ctrl.keys.ArrowRight) {  
if (ctrl.keys.ArrowRight || ctrl.keys.l) {  
if (ctrl.velX < ctrl.speed) {  
ctrl.velX++;  
}  
}  
  
if (ctrl.keys.ArrowLeft) {  
if (ctrl.keys.ArrowLeft || ctrl.keys.j) {  
if (ctrl.velX > -ctrl.speed) {  
ctrl.velX--;  
}  
}
```

3

수정 사항

13. w, a, s, d 으로도 조작을 할 수 있도록 함

```
187 187
188 - if (ctrl.keys.ArrowUp || ctrl.keys.i) {
188 + if (ctrl.keys.ArrowUp || ctrl.keys.i || ctrl.keys.w) {
189 189         if (ctrl.velY > -ctrl.speed) {
190 190             ctrl.velY--;
191 191         }
192 192     }
193 193
194 - if (ctrl.keys.ArrowDown || ctrl.keys.k) {
194 + if (ctrl.keys.ArrowDown || ctrl.keys.k || ctrl.keys.s) {
195 195         if (ctrl.velY < ctrl.speed) {
196 196             ctrl.velY++;
197 197         }
198 198     }
199 199
200 - if (ctrl.keys.ArrowRight || ctrl.keys.l) {
200 + if (ctrl.keys.ArrowRight || ctrl.keys.l || ctrl.keys.d) {
201 201         if (ctrl.velX < ctrl.speed) {
202 202             ctrl.velX++;
203 203         }
204 204     }
205 205
206 - if (ctrl.keys.ArrowLeft || ctrl.keys.j) {
206 + if (ctrl.keys.ArrowLeft || ctrl.keys.j || ctrl.keys.a) {
207 207         if (ctrl.velX > -ctrl.speed) {
208 208             ctrl.velX--;
```


3

수정 사항

14. Home, End, Delete, Page Down 으로도 조작을 할 수 있도록 함

```
↑... @@ -185,25 +185,25 @@ document.addEventListener('DOMContentLoaded', () => {  
185 185         player.resurrect();  
186 186     }  
187 187  
188 -     if (ctrl.keys.ArrowUp || ctrl.keys.i || ctrl.keys.w) {  
188 +     if (ctrl.keys.ArrowUp || ctrl.keys.i || ctrl.keys.w || ctrl.keys.Home) {  
189 189         if (ctrl.velY > -ctrl.speed) {  
190 190             ctrl.velY--;  
191 191         }  
192 192     }  
193 193  
194 -     if (ctrl.keys.ArrowDown || ctrl.keys.k || ctrl.keys.s) {  
194 +     if (ctrl.keys.ArrowDown || ctrl.keys.k || ctrl.keys.s || ctrl.keys.End) {  
195 195         if (ctrl.velY < ctrl.speed) {  
196 196             ctrl.velY++;  
197 197         }  
198 198     }  
199 199  
200 -     if (ctrl.keys.ArrowRight || ctrl.keys.l || ctrl.keys.d) {  
200 +     if (ctrl.keys.ArrowRight || ctrl.keys.l || ctrl.keys.d || ctrl.keys.PageDown) {  
201 201         if (ctrl.velX < ctrl.speed) {  
202 202             ctrl.velX++;  
203 203         }  
204 204     }  
205 205  
206 -     if (ctrl.keys.ArrowLeft || ctrl.keys.j || ctrl.keys.a) {  
206 +     if (ctrl.keys.ArrowLeft || ctrl.keys.j || ctrl.keys.a || ctrl.keys.Delete) {  
207 207         if (ctrl.velX > -ctrl.speed) {
```

3

수정 사항

15. 장애물이 위 아래로 움직이도록 함

```

12 12
13 + function randIndex(thearray) {
14 +     return thearray[rand(1, thearray.length) - 1];
15 + }
16 +
13 17 const player = (function () {
14 18     let x = 100,
15 19     y = canvas.height / 2,
16 20     @@ -86,16 +90,24 @@ document.addEventListener(
17 21     h_min: 15,
18 22     h_max: 100,
19 23     speed_min: 0.5,
20 24     speed_max: 5,
21 25     direction : ['up', 'down']
22 26 });

```

```

93 - function Block() {
94 + function Block(direction) {
95 +     this.w = 10;
96 +     this.h = 50;
97 +     this.x = rand(210, 700, 10);
98 +     this.y = canvas.height;
99 +     this.speed = 1;
100 +     this.direction = direction;
101 +     if(direction === "up") {
102 +         this.y = canvas.height;
103 +     }
104 +     else {
105 +         this.y -= canvas.height;
106 +     }
107 + }
108 +
109 +
110 +

```

```

for (let i = 0; i < blocks.length; ++i) {
    blocks[i].y += blocks[i].speed;
    if (blocks[i].y > canvas.height) {
        blocks[i].y = 0;
        blocks[i].y -= rand(10, 350);
    }
    if(blocks[i].direction === 'up') {
        blocks[i].y -= blocks[i].speed;
        if((blocks[i].y + blocks[i].h) < 0) {
            blocks[i].y = canvas.height + rand(10, 350);
        }
    }
    else {
        blocks[i].y += blocks[i].speed;
        if(blocks[i].y > canvas.height) {
            blocks[i].y = 0;
            blocks[i].y -= rand(10, 350);
        }
    }
}

```

```

132 for (let i = 0; i < n; ++i) {
133 - blocks.push(new Block());
134 + blocks.push(new Block(randIndex(start.direction)));
135 }

```

3

수정 사항

16. 레벨(난이도)를 추가하여 텍스트로 출력함

```
const blocks = (function () {  
  let blocks = (function () {  
    let blocks = [],  
        level = 0,  
        level_factor = 1.20,  
        start = {
```

```
    getLevel: function () {  
      return level;  
    },
```

```
290 294 ctx.fillText("죽은 횟수 : " + player.getDeath(), 10, 80);  
291 - ctx.fillText("죽은 횟수 : " + player.getDeath(), 10, 100);  
295 + ctx.fillText("레벨 : " + blocks.getLevel(), 10, 120);  
296 + ctx.fillText("죽은 횟수 : " + player.getDeath(), 10, 100);  
292 297 }
```

3

수정 사항

17. 레벨(난이도)에 따라 이동 속도가 느려지도록 함

```
195      200      velX: 0,  
196      -      speed: 1400,  
      201      +      speed: 1400 - blocks.getLevel() * 1.1 ,  
197      202      friction: 0.68,  
198      203      keys: [
```

3

수정 사항

18. 레벨(난이도)에 따라 장애물의 이동 속도가 증가하도록 함

```
speed_min: 0.5 + level * 1.03,  
speed_max: 5 + level * 1.05,
```

```
this.w = 10;  
this.h = 50;  
this.x = rand(210, 700, 10);  
this.y = canvas.height;  
this.speed = 1;  
this.h = rand(start.h_min, start.h_max);  
this.x = rand(start.x1, start.x2, 10);  
this.y = 0;  
this.speed = rand(start.speed_min, start.speed_max);  
this.direction = direction;
```

3

수정 사항

19. 레벨(난이도)에 따라 장애물의 개수가 늘어나도록 함

```
123         blocks = [];  
-         let n = Math.ceil(start.n + 1.3);  
124 +         let n = Math.ceil(start.n + (level * level_factor));  
125         this.createXBlocks(n);  
126     }
```

3

수정 사항

20. 레벨(난이도)에 따라 장애물의 길이가 더욱 다양해지도록 함

```
this.w = 10;  
this.h = 50;  
this.x = rand(210, 700, 10);  
this.y = canvas.height;  
this.speed = 1;  
this.h = rand(start.h_min, start.h_max);  
this.x = rand(start.x1, start.x2, 10);  
this.y = 0;
```

4

최종 결과

<https://kshopenrepos.github.io/2022-Game-Programming-Canvas-Game>

<https://github.com/kshopenrepos/2022-Game-Programming-Canvas-Game>

- <https://codepen.io/jmt4k8awngdw1rki3pln/pen/rNKzyyW>
- <https://m2.material.io/resources/color/#!/?view.left=0&view.right=0&primary.color=D50000>
- <https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/keyCode>
- https://www.w3schools.com/graphics/canvas_intro.asp
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random
- <https://stackoverflow.com/questions/5915096/get-a-random-item-from-a-javascript-array>

- 감사합니다 -