

게임프로그래밍

소프트웨어학과
4학년
2018843103
정현준



경성대학교
KYUNGSUNG UNIVERSITY

CONTENTS

- 01** 변경사항
- 02** 업그레이드한 부분
- 03** 변경 후 코드
- 04** Before & After 게임 장면
- 05** Q&A
- 05** Thank you

변경 사항

1. 플레이어 색상 변경
2. 플레이어 크기 변경
3. 플레이어 시작 위치 변경
4. 점수 폰트 크기 변경
5. 점수 위치 변경
6. 장애물 색상 변경

7. 장애물 크기 변경
8. 최소 장애물 높이 변경
9. 최대 장애물 간격 변경
10. 게임 속도 변경
11. 플레이어의 이동 속도 증가
12. 장애물 생성 빈도 증가

업그레이드한 부분1

1. 플레이어 색상 변경:

- 기존: `myGamePiece = new component(30, 30, "red", 10, 120);`
- 변경: `myGamePiece = new component(30, 30, "black", 10, 120);`
- 설명: 플레이어를 나타내는 사각형의 색상을 빨간색에서 검은색으로 변경합니다.

2. 플레이어 크기 변경:

- 변경: `myGamePiece = new component(40, 40, "black", 10, 120);`
- 설명: 플레이어의 크기를 30x30에서 40x40으로 증가시킵니다.

3. 플레이어 시작 위치 변경:

- 변경: `myGamePiece = new component(30, 30, "black", 30, 100);`
- 설명: 플레이어의 시작 위치를 x축에서 10에서 30으로, y축에서 120에서 100으로 변경합니다.

업그레이드 한 부분2

4. 점수 폰트 크기 변경:

- 변경: `myScore = new component("20px", "Consolas", "black", 280, 40, "text");`
- 설명: 점수를 표시하는 텍스트의 폰트 크기를 30px에서 20px로 변경합니다.

5. 점수 위치 변경:

- 변경: `myScore = new component("30px", "Consolas", "black", 250, 60, "text");`
- 설명: 점수의 위치를 x축에서 280에서 250으로, y축에서 40에서 60으로 변경합니다.

6. 장애물 색상 변경:

- 변경: 장애물 생성 코드 내 `new component(10, height, "blue", x, 0)` 및 `new component(10, x - height - gap, "blue", x, height + gap)`
- 설명: 장애물의 색상을 녹색에서 파란색으로 변경합니다.

업그레이드 한 부분3

7. 장애물 크기 변경:

- 변경: 장애물 생성 코드 내 `new component(15, height, "green", x, 0)` 및 `new component(15, x - height - gap, "green", x, height + gap)`
- 설명: 장애물의 너비를 10에서 15로 증가시킵니다.

8. 최소 장애물 높이 변경:

- 변경: `minHeight = 50;`
- 설명: 장애물의 최소 높이를 20에서 50으로 증가시킵니다.

9. 최대 장애물 간격 변경:

- 변경: `maxGap = 250;`
- 설명: 장애물 사이의 최대 간격을 200에서 250으로 증가시킵니다.

업그레이드 한 부분4

10. 게임 속도 변경:

- 변경: `this.interval = setInterval(updateGameArea, 15);`
- 설명: 게임의 업데이트 간격을 20밀리초에서 15밀리초로 단축하여 게임 속도를 빠르게 합니다.

11. 플레이어의 이동 속도 증가:

- 변경: `moveup()`, `movedown()`, `moveleft()`, `moveright()` 함수 내에서 플레이어의 `speedY` 또는 `speedX` 값을 2 또는 2로 설정
- 설명: 플레이어의 수직 및 수평 이동 속도를 빠르게 합니다.

12. 장애물 생성 빈도 증가:

- 변경: `everyinterval(150)` 를 `everyinterval(100)` 로 변경
- 설명: 장애물이 생성되는 간격을 줄여 게임의 난이도를 증가시킵니다.

변경 후 코드 1

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
5 <style>
6 canvas {
7   border:1px solid #d3d3d3;
8   background-color: #f1f1f1;
9 }
10 </style>
11 </head>
12 <body onload="startGame()">
13 <script>
14
15 var myGamePiece;
16 var myObstacles = [];
17 var myScore;
18
19 function startGame() {
20   myGamePiece = new component(40, 40, "black", 30, 100); // 1, 2, 3: 색상, 크기, 시작 위치 변경
21   myScore = new component("20px", "Consolas", "black", 250, 60, "text"); // 4, 5: 폰트 크기 및 위치 변경
22   myGameArea.start();
23 }
24
25 var myGameArea = {
26   canvas : document.createElement("canvas"),
27   start : function() {
28     this.canvas.width = 480;
29     this.canvas.height = 270;
30     this.context = this.canvas.getContext("2d");
31     document.body.insertBefore(this.canvas, document.body.childNodes[0]);
32     this.frameNo = 0;
33     this.interval = setInterval(updateGameArea, 15); // 10: 업데이트 간격 변경
34   },
35   clear : function() {
36     this.context.clearRect(0, 0, this.canvas.width, this.canvas.height);
37   },
38   stop : function() {
39     clearInterval(this.interval);
40   }
41 }
42
```

```
43 function component(width, height, color, x, y, type) {
44   this.type = type;
45   this.width = width;
46   this.height = height;
47   this.speedX = 0;
48   this.speedY = 0;
49   this.x = x;
50   this.y = y;
51   this.update = function() {
52     ctx = myGameArea.context;
53     if (this.type == "text") {
54       ctx.font = this.width + " " + this.height;
55       ctx.fillStyle = color;
56       ctx.fillText(this.text, this.x, this.y);
57     } else {
58       ctx.fillStyle = color;
59       ctx.fillRect(this.x, this.y, this.width, this.height);
60     }
61   }
62   this.newPos = function() {
63     this.x += this.speedX;
64     this.y += this.speedY;
65   }
66   this.crashWith = function(otherobj) {
67     var myleft = this.x;
68     var myright = this.x + (this.width);
69     var mytop = this.y;
70     var mybottom = this.y + (this.height);
71     var otherleft = otherobj.x;
72     var otherright = otherobj.x + (otherobj.width);
73     var othertop = otherobj.y;
74     var otherbottom = otherobj.y + (otherobj.height);
75     var crash = true;
76     if ((mybottom < othertop) || (mytop > otherbottom) || (myright < otherleft) || (myleft > otherright))
77       crash = false;
78   }
79   return crash;
80 }
81 }
```

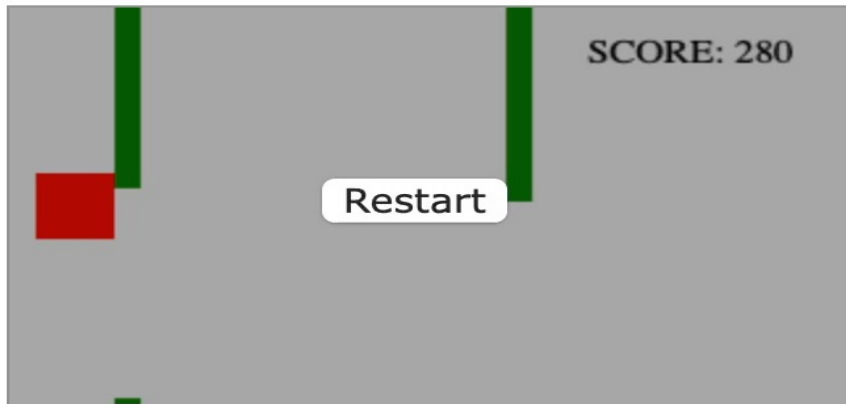

변경 후 코드2

```
83 function updateGameArea() {
84     var x, height, gap, minHeight, maxHeight, minGap, maxGap;
85     for (i = 0; i < myObstacles.length; i += 1) {
86         if (myGamePiece.crashWith(myObstacles[i])) {
87             myGameArea.stop();
88             return;
89         }
90     }
91     myGameArea.clear();
92     myGameArea.frameNo += 1;
93     if (myGameArea.frameNo == 1 || everyinterval(100)) { // 12: 장애물 생성 빈도 증가
94         x = myGameArea.canvas.width;
95         minHeight = 50; // 8: 장애물의 최소 높이 변경
96         maxHeight = 200;
97         height = Math.floor(Math.random()*(maxHeight-minHeight+1)+minHeight);
98         minGap = 50;
99         maxGap = 250; // 9: 장애물 간 최대 간격 변경
100         gap = Math.floor(Math.random()*(maxGap-minGap+1)+minGap);
101         myObstacles.push(new component(15, height, "blue", x, 0)); // 6, 7: 장애물 색상 및 크기 변경
102         myObstacles.push(new component(15, x - height - gap, "blue", x, height + gap)); // 6, 7: 장애물 색상 및 크기 변경
103     }
104     for (i = 0; i < myObstacles.length; i += 1) {
105         myObstacles[i].speedX = -1;
106         myObstacles[i].newPos();
107         myObstacles[i].update();
108     }
109     myScore.text="SCORE: " + myGameArea.frameNo;
110     myScore.update();
111     myGamePiece.newPos();
112     myGamePiece.update();
113 }
114
```

```
115 function everyinterval(n) {
116     if ((myGameArea.frameNo / n) % 1 == 0) {return true;}
117     return false;
118 }
119
120 function moveup() {
121     myGamePiece.speedY = -2; // 11: 플레이어의 수직 이동 속도 증가
122 }
123
124 function movedown() {
125     myGamePiece.speedY = 2; // 11: 플레이어의 수직 이동 속도 증가
126 }
127
128 function moveleft() {
129     myGamePiece.speedX = -2; // 11: 플레이어의 수평 이동 속도 증가
130 }
131
132 function moveright() {
133     myGamePiece.speedX = 2; // 11: 플레이어의 수평 이동 속도 증가
134 }
135
136 function clearmove() {
137     myGamePiece.speedX = 0;
138     myGamePiece.speedY = 0;
139 }
140 </script>
141 <div style="text-align:center;width:480px;">
142     <button onmousedown="moveup()" onmouseup="clearmove()" ontouchstart="moveup()">UP</button><br><br>
143     <button onmousedown="moveleft()" onmouseup="clearmove()" ontouchstart="moveleft()">LEFT</button>
144     <button onmousedown="moveright()" onmouseup="clearmove()" ontouchstart="moveright()">RIGHT</button><br><br>
145     <button onmousedown="movedown()" onmouseup="clearmove()" ontouchstart="movedown()">DOWN</button>
146 </div>
147
148 <p>The score will count one point for each frame you manage to "stay alive".</p>
149 </body>
150 </html>
151
```

Before & After 게임

Push the buttons to move the red square:

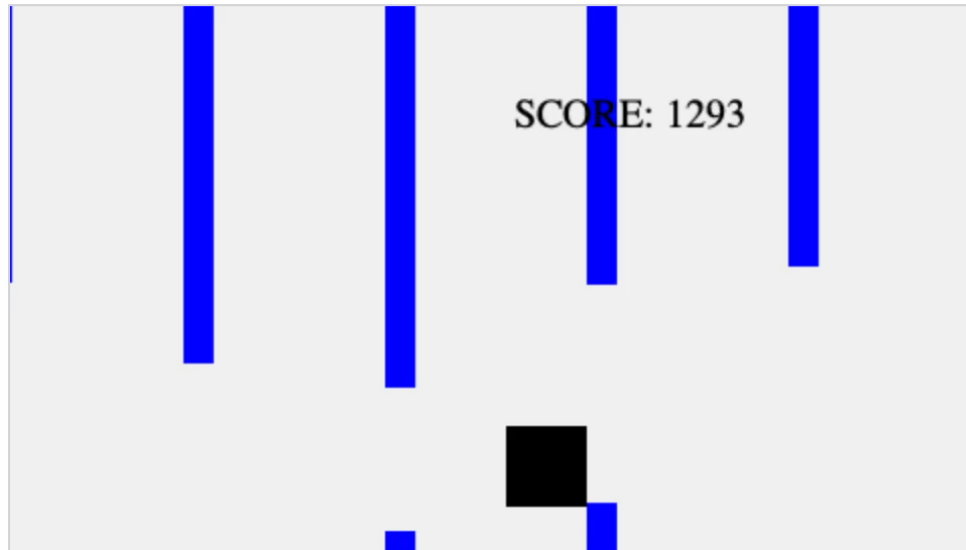


UP

LEFT

RIGHT

DOWN



UP

LEFT

RIGHT

DOWN

The score will count one point for each frame you manage to "stay alive".



Q & A

The image features the words "THANK YOU" in a stylized, italicized, blue-outlined font. The text is centered on a white background. The top of the image is bordered by a blue shape on the left and a red shape on the right, meeting at a point. The bottom of the image is bordered by an orange shape on the left and a grey shape on the right, meeting at a point.

THANK YOU