



게임프로그래밍 HTML게임 업데이트

2020904053 윤금비



목차

01

업데이트 요소

03

코드 실행

02

코드 소개

04

출처

01. 업데이트 요소

✓ 캔버스 영역

움직이는 요소 변경

배경 변경

스코어 폰트, 장애물 색상 변경

myGamePiece 캔버스 영역을 벗어나지 않도록 함

✓ 버튼

폰트, 색상, 크기 변경

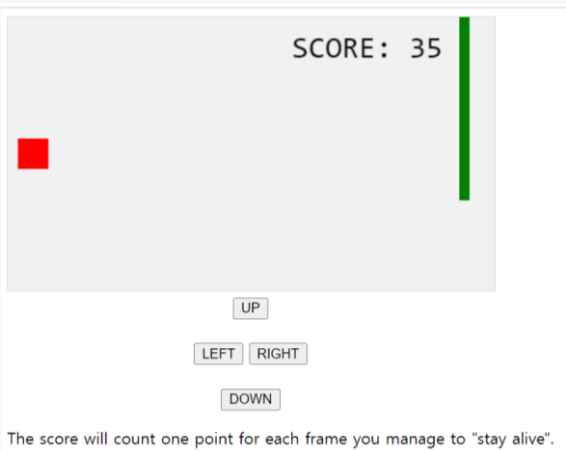
stop/start 버튼 추가

✓ 아웃트로

화면, 최종 스코어 표시

종료 시 음악 출력

01. 업데이트 요소



The score will count one point for each frame you manage to "stay a
And when you stop, the score stops.



The score will count one point for each frame you manage to "sta
And when you stop, the score stops.

02. 코드 소개 <style> & <audio>

```
5 <style>
6   canvas {
7     border: 1px solid #d3d3d3;
8     background-color: #f1f1f1;
9   }
10
11   #gameAreaContainer { /* 게임 영역 컨테이너의 스타일을 정의 */
12     position: relative;
13     width: 480px;
14   }
15
16   #outroScreen { /*아웃트로 화면. 초기에는 보이지 않음 */
17     display: none;
18     position: absolute;
19     top: 0;
20     left: 0;
21     width: 100%;
22     height: 100%;
23     background-color: black;
24     justify-content: center;
25     align-items: center;
26     flex-direction: column;
27     text-align: center;
28     z-index: 1; /* 아웃트로가 다른 요소 위에 나타나도록 조절 */
29   }
30
31   #outroScreen h1, #outroScreen p { /* <h1>과 <p> 태그에 대한 스타일 지정 */
32     color: white;
33     font-family: 'Comic Sans MS', cursive;
```

```
35   .button-style { /* 버튼 스타일 지정 */
36     font-family: 'Comic Sans MS', cursive;
37     background-color: #fb1;
38     border: 2px solid #B71;
39     border-radius: 5px;
40     width: 100px; height: 40px;
41   }
42 </style>
43 <audio id="collisionSound" style="display: none;">
44   <source src="Hal (1).mp3" type="audio/mp3">
45 </audio>
46 </head>
```

02. 코드 소개 startGame() & outro()

```
56 <script>
57   var myGamePiece;
58   var myObstacles = [];
59   var myScore;
60   var gameRunning = true;
61
62   function startGame() { /*게임 초기화 */
63       myGamePiece = new component(30, 30, "bee.png", 10, 120, "image");
64       myScore = new component("30px", "Comic Sans MS", "black", 280, 40, "text");
65       myGameArea.start();
66   }
67
68   function outro() { /* 아웃트로, 점수 표시 */
69       document.getElementById("outroScreen").style.display = "flex";
70       document.getElementById("finalScore").innerText = myGameArea.frameNo;
71   }
```

02. 코드 소개 myGameArea

```
73 var myGameArea = {           /* 게임 영역 설정 */
74   canvas: document.getElementById("myCanvas"),
75   start: function () {
76     this.canvas.width = 480;
77     this.canvas.height = 270;
78     this.context = this.canvas.getContext("2d");
79     this.frameNo = 0;
80     this.interval = setInterval(updateGameArea, 20);
81     this.canvas.style.background = `
82       radial-gradient(circle farthest-side at 0% 50%, #fb1 23.5%, rgba(240,166,17,0) 0) 21px 30px,
83       radial-gradient(circle farthest-side at 0% 50%, #B71 24%, rgba(240,166,17,0) 0) 19px 30px,
84       linear-gradient(#fb1 14%, rgba(240,166,17,0) 0, rgba(240,166,17,0) 85%, #fb1 0) 0 0,
85       linear-gradient(150deg, #fb1 24%, #B71 0, #B71 26%, rgba(240,166,17,0) 0, rgba(240,166,17,0) 74%, #B71 0, #B71 76%, #fb1 0) 0 0,
86       linear-gradient(30deg, #fb1 24%, #B71 0, #B71 26%, rgba(240,166,17,0) 0, rgba(240,166,17,0) 74%, #b71 0, #B71 76%, #fb1 0) 0 0,
87       linear-gradient(90deg, #B71 2%, #fb1 0, #fb1 98%, #B71 0%) 0 0 #fb1`;
88     this.canvas.style.backgroundSize = "40px 60px";
89   },
90   clear: function () {        /* 업데이트 (영역 지움) */
91     this.context.clearRect(0, 0, this.canvas.width, this.canvas.height);
92   },
93   stop: function () {
94     clearInterval(this.interval);
95     outro();
96   }
97 }
```

02. 코드 소개 component

```
function component(width, height, color, x, y, type) { /* 꿀벌을 나타내는 객체를 생성하는 역할 */
  this.type = type;
  if (type == "image") {
    this.image = new Image();
    this.image.src = color;
  }
  this.width = width;
  this.height = height;
  this.speedX = 0;
  this.speedY = 0;
  this.x = x;
  this.y = y;
  this.update = function () { /* 객체 그리는 역할 */
    ctx = myGameArea.context;
    if (type == "image") {
      ctx.drawImage(this.image, this.x, this.y, this.width, this.height);
    } else if (type == "text") {
      ctx.font = this.width + " " + this.height;
      ctx.fillStyle = color;
      ctx.fillText(this.text, this.x, this.y);
    } else {
      ctx.fillStyle = color;
      ctx.fillRect(this.x, this.y, this.width, this.height);
    }
  }
  this.newPos = function () { /* 객체를 새로운 위치로 업데이트 */
    this.x += this.speedX;
    this.y += this.speedY;
  }
}
```

```
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
this.crashWith = function (otherobj) { /* 다른 객체와 충돌했는지 여부를 확인 */
  var myleft = this.x;
  var myright = this.x + (this.width);
  var mytop = this.y;
  var mybottom = this.y + (this.height);
  var otherleft = otherobj.x;
  var otherright = otherobj.x + (otherobj.width);
  var othertop = otherobj.y;
  var otherbottom = otherobj.y + (otherobj.height);
  var crash = true;
  if ((mybottom < othertop) || (mytop > otherbottom) || (myright < otherleft)
    || (myleft > otherright)) {
    crash = false;
  }
  return crash;
}
```


02. 코드 소개 updateGameArea(1)

```
146 function updateGameArea() {
147     if (!gameRunning) {
148         return;
149     }
150     // 충돌을 확인하고, 충돌이 발생하면 게임을 멈추고 충돌 소리를 재생하는 역할
151     var x, height, gap, minHeight, maxHeight, minGap, maxGap;
152     for (i = 0; i < myObstacles.length; i += 1) {
153         if (myGamePiece.crashWith(myObstacles[i])) {
154             myGameArea.stop();
155             var collisionAudio = document.getElementById("collisionSound");
156             collisionAudio.play(); // 충돌 소리 자동 재생
157             return;
158         }
159     }
160     myGamePiece.newPos();
161
162     // 새로운 위치 계산 후 캔버스 경계 제한
163     if (myGamePiece.x < 0) {
164         myGamePiece.x = 0;
165     }
166     if (myGamePiece.x + myGamePiece.width > myGameArea.canvas.width) {
167         myGamePiece.x = myGameArea.canvas.width - myGamePiece.width;
168     }
169     if (myGamePiece.y < 0) {
170         myGamePiece.y = 0;
171     }
172     if (myGamePiece.y + myGamePiece.height > myGameArea.canvas.height) {
173         myGamePiece.y = myGameArea.canvas.height - myGamePiece.height;
174     }
```

02. 코드 소개 updateGameArea(2)

```
176 myGameArea.clear();
177 myGameArea.frameNo += 1;
178 if (myGameArea.frameNo == 1 || everyinterval(150)) { //새로운 장애물을 생성
179     x = myGameArea.canvas.width;
180     minHeight = 20;
181     maxHeight = 200;
182     height = Math.floor(Math.random() * (maxHeight - minHeight + 1) + minHeight);
183     minGap = 50;
184     maxGap = 200;
185     gap = Math.floor(Math.random() * (maxGap - minGap + 1) + minGap);
186     myObstacles.push(new component(10, height, "gray", x, 0));
187     myObstacles.push(new component(10, x - height - gap, "gray", x, height + gap));
188 }
189 for (i = 0; i < myObstacles.length; i += 1) { //모든 장애물 업데이트
190     myObstacles[i].speedX = -1;
191     myObstacles[i].newPos();
192     myObstacles[i].update();
193 }
194 myScore.text = "SCORE: " + myGameArea.frameNo; //점수 표시
195 myScore.update();
196 myGamePiece.newPos();
197 myGamePiece.update();
198 }
```

02. 코드 소개 나머지 코드

```
200 function everyinterval(n) { //새로운 장애물을 생성할 때 사용
201     if ((myGameArea.frameNo / n) % 1 == 0) {return true;}
202     return false;
203 }
204
205 function moveup() { //이동 방향에 대한 속도를 설정
206     myGamePiece.speedY = -1; //함수는 수직 속도를 -1로 설정하여 게임 요소를 위로 이동
207 }
208
209 function movedown() {
210     myGamePiece.speedY = 1;
211 }
212
213 function moveleft() {
214     myGamePiece.speedX = -1;
215 }
216
217 function moveright() {
218     myGamePiece.speedX = 1;
219 }
220
221 function clearmove() { //게임 요소의 이동을 멈추는 역할
222     myGamePiece.speedX = 0;
223     myGamePiece.speedY = 0;
224 }
225
226 function toggleGame() { //객체가 움직이고, 멈추는 Stop/Start 버튼과 관련
227     gameRunning = !gameRunning;
228 }
```

02. 코드 소개 나머지 코드

```
229 </script>
230
231 <div style="text-align:center; width:480px;">    //버튼들을 생성
232     //마우스가 버튼을 클릭하면, moveup() 함수호출. 마우스 클릭이 해제되면 clearmove()으로 별의 이동 멈춤
233     <button onmousedown="moveup()" onmouseup="clearmove()" ontouchstart="moveup()"
234     class="button-style" >UP</button><br><br>
235     <button onmousedown="moveleft()" onmouseup="clearmove()" ontouchstart="moveleft()"
236     class="button-style">LEFT</button>
237     <button onmousedown="moveright()" onmouseup="clearmove()" ontouchstart="moveright()"
238     class="button-style">RIGHT</button><br><br>
239     <button onmousedown="movedown()" onmouseup="clearmove()" ontouchstart="movedown()"
240     class="button-style">DOWN</button><br><br>
241     <button onClick="toggleGame()" class="button-style">Stop/Start</button>
242 </div>
243
244 <p>The score will count one point for each frame you manage to "stay alive".</p>
245 <p>And when you stop, the score stops.</p>
246
247 </body>
248 </html>
```

The background features decorative geometric patterns in the corners, consisting of thin blue lines, dots, and concentric circles. The top-left corner has a cluster of lines and dots. The top-right corner has a circle with a dot inside and a line with a dot. The bottom-left corner has a circle with a dot inside and a line with a dot. The bottom-right corner has a circle with a dot inside and a line with a dot.

03

코드 실행

04. 출처

- <https://www.flaticon.com/kr/> -
- <https://projects.verou.me/css3patterns/>
- <https://chat.openai.com>
- <https://htmlreference.io/element/audio/>
- <https://slidesgo.com/ko/>



이상입니다.

2020904053 윤금비

