

원본 게임 소스코드 분석

2016848029 김진우

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>repl.it</title>
</head>
<body>
  // 자바스크립트는 index.js를 사용
  <script src="index.js"></script>
</body>
</html>
```

```
// window가 로드되면
window.onload = function () {
    // 가로 640, 세로 480으로 캔버스 요소 생성
    canvas = document.createElement("canvas");
    canvas.width = 640;
    canvas.height = 480;
    // HTML DOM 트리의 body에 캔버스 요소 추가
    document.body.appendChild(canvas);
    // 캔버스에 2d로 그릴 수 있도록 준비
    pen = canvas.getContext("2d");
    // 마우스가 움직일 경우 mouseMoveHandler를 실행하도록 설정
    document.addEventListener("mousemove", mouseMoveHandler, false)
    // 마우스를 클릭할 경우 mouseClickedHandler를 실행하도록 설정
    document.addEventListener("mousedown", mouseClickedHandler, false)
```

```
// 필요한 이미지 객체를 생성하고 소스 경로 설정
spaceship = new Image();
spaceship.src = "http://pixelartmaker.com/art/d9c710d4c7e1ae6.png";
bullet = new Image();
bullet.src = "http://pixelartmaker.com/art/17f45360a19f8e6.png";
enemy = new Image();
enemy.src = "http://pixelartmaker.com/art/b46ed2c95f86caf.png";
// 초당 프레임 수를 정함
var fps = 30;
// 1초(1000ms)마다 위에서 정한 fps수 만큼 update가 실행되도록 설정
setInterval(update, 1000 / fps);
// 1초(1000ms)마다 spawn이 한 번씩 실행되도록 설정
setInterval(spawn, 1000);
};
```

```
player_x = 100;    // 플레이어의 초기 x 좌표
player_y = 100;    // 플레이어의 초기 y 좌표
player_speed = 15; // 플레이어가 프레임 마다 움직이는 속도
player_dim = 30;   // 콜리전용 플레이어 이미지의 지름

enemy_list = [];   // 적 개체들을 담을 배열
enemy_dim = 25;    // 콜리전용 적 이미지의 지름
enemy_speed = 5;   // 플레이어가 프레임 마다 움직이는 속도

shot_list = [];    // 총알 개체들을 담을 배열
shot_dim = 4;      // 콜리전용 총알 이미지의 지름
shot_speed = 7;    // 총알이 프레임 마다 움직이는 속도

lives = 5;         // 초기 생명력 수
score = 0;         // 초기 점수
```

```
// spawn 함수를 실행하면  
  x 좌표는 캔버스의 오른쪽 끝으로  
  y 좌표는 '0 ~ 캔버스의 높이' 중에 랜덤으로 결정해서  
  적을 생성하고 enemy_list 배열에 추가  
function spawn() {  
  enemy_list.push({  
    x: canvas.width,  
    y: Math.random() * canvas.height  
  });  
}
```

```
function update() {  
    // 검은색으로 캔버스를 가득 채움  
    pen.fillStyle = "black";  
    pen.fillRect(0, 0, canvas.width, canvas.height);  
  
    // 생명 수가 0보다 클 경우  
    if (lives > 0) {  
        // 30px Arial 하얀색으로 (350, 30) 위치에 남은 생명과 점수 표시  
        pen.beginPath();  
        pen.font = "30px Arial"  
        pen.fillStyle = "white";  
        pen.fillText("Lives: " + lives + " Score: " + score, 350, 30);  
        pen.fill();  
    }  
}
```

// 플레이어 x, y 좌표에 플레이어 이미지의 중간이 위치하도록 좌표 조정 후
콜리전용 플레이어의 지름 길이를 한 변으로 하는 정사각형 크기로
플레이어 이미지를 그림

```
pen.drawImage(spaceship,  
    player_x - player_dim / 2,  
    player_y - player_dim / 2,  
    player_dim, player_dim);
```

// shot_list 배열을 돌면서 각 총알의 x 좌표를 shot_speed만큼 추가하고
총알 x, y 좌표에 총알 이미지의 중간이 위치하도록 좌표 조정 후
가로는 콜리전용 총알의 지름+20, 세로는 콜리전용 총알의 지름+10의 크기로
총알 이미지를 그림

```
for (var s = 0; s < shot_list.length; s++) {  
    shot_list[s].x += shot_speed;  
    pen.drawImage(bullet,  
        shot_list[s].x - shot_dim / 2,  
        shot_list[s].y - shot_dim / 2,  
        shot_dim + 20, shot_dim + 10);
```



```
// enemy_list 배열을 돌면서 적과 총알이 충돌하는지 체크
```

```
for (var e = enemy_list.length - 1; e >= 0; e--) {  
    var diff_x = Math.abs(enemy_list[e].x - shot_list[s].x);  
    var diff_y = Math.abs(enemy_list[e].y - shot_list[s].y);  
    var dist = Math.sqrt(diff_x * diff_x + diff_y * diff_y);
```

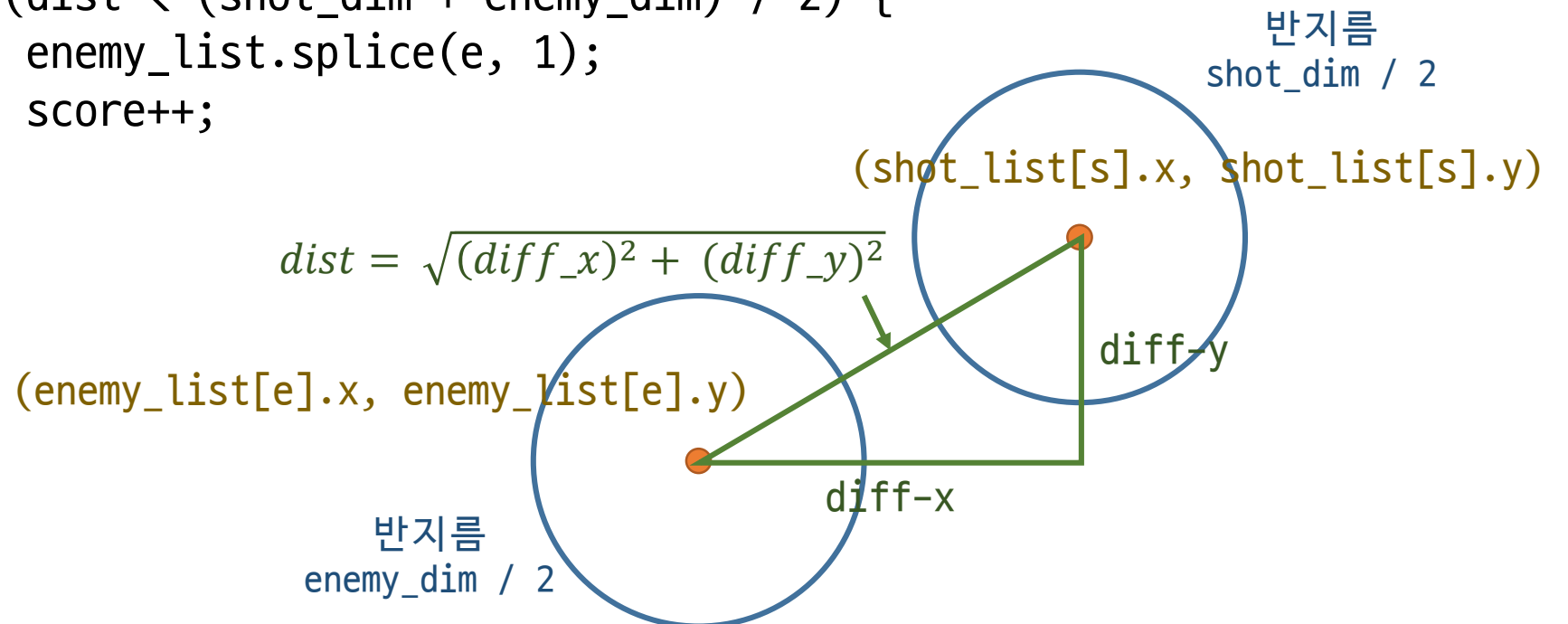
```
// 충돌하면 enemy_list에서 적을 삭제하고 score를 1증가
```

```
if (dist < (shot_dim + enemy_dim) / 2) {  
    enemy_list.splice(e, 1);  
    score++;
```

```
}
```

```
}
```

```
}
```



// enemy_list 배열을 돌면서 각 적의 x 좌표를 enemy_speed만큼 빼고
적의 x, y 좌표에 적 이미지의 중간이 위치하도록 좌표 조정 후
콜리전용 적의 지름 길이를 한 변으로 하는 정사각형 크기로
적 이미지를 그림

```
for (var e = 0; e < enemy_list.length; e++) {  
    enemy_list[e].x -= enemy_speed;  
    pen.drawImage(enemy,  
        enemy_list[e].x - enemy_dim / 2,  
        enemy_list[e].y - enemy_dim / 2,  
        enemy_dim, enemy_dim);  
}
```

```
var diff_x = Math.abs(enemy_list[e].x - player_x);
var diff_y = Math.abs(enemy_list[e].y - player_y);
var dist = Math.sqrt(diff_x * diff_x + diff_y * diff_y);
// 적과 플레이어가 충돌하거나 플레이어가 화면 밖으로 나가면
if (dist < (player_dim + enemy_dim) / 2
    || player_x < 0 || player_x > 640
    || player_y < 0 || player_y > 480) {
    // shot_list와 enemy_list를 비우고 플레이어의 좌표를
    초기 위치인 (100, 100)으로 이동하고 생명을 하나 감소시킴
    shot_list = [];
    enemy_list = [];
    player_x = player_y = 100;
    lives--;
    break;
}
}
}
```

```
// 남은 생명 수가 0이하 이면
if (lives <= 0) {
    // 빨간색 중앙정렬로 GAME OVER 텍스트는 (300, 200)위치에
    // Score 텍스트는 (300, 240)위치에 표시
    pen.fillStyle = "red";
    pen.textAlign = "center";
    pen.textBaseLine = "middle";
    pen.fillText("GAME OVER", 300, 200);
    pen.fillText("Score: " + score, 300, 240)
}
}
```

```
// 마우스를 움직일 경우
function mouseMoveHandler(e) {
    // 플레이어의 y 좌표를 마우스의 y 좌표로 지정
    player_y = e.clientY;
}

// 마우스를 클릭할 경우
function mouseClickedHandler(e) {
    // 마우스 왼쪽 버튼을 클릭할 경우
    if (e.button == 0) {
        // 플레이어의 x, y 좌표를 가지는 총알 객체를 shot_list에 추가
        shot_list.push({ x: player_x, y: player_y });
    }
}
```



감사합니다