

Game Programming

HTML 게임

업그레이드 발표

2018975058 전지환

# 업그레이드 한 기능들

01	중력 추가
02	점프 추가
03	좌우 이동 기능 추가
04	다시 시작 버튼 추가

05	플레이어 이미지 변경
06	배경 이미지 설정
07	배경 음악 설정

01

GRAVITY

# 중력

첫 번째 기능

# 추가한 기능: 중력

```
function component(width, height, color, x, y, type) {  
  this.type = type;  
  this.width = width;  
  this.height = height;  
  this.speedX = 0;  
  this.speedY = 0;  
  this.x = x;  
  this.y = y;  
}
```

중력 속도와  
중력 가속도 설정

```
function component(width, height, color, x, y, type) {  
  this.type = type;  
  this.width = width;  
  this.height = height;  
  this.speedX = 0;  
  this.speedY = 0;  
  this.x = x;  
  this.y = y;  
  this.gravity = 0.05;  
  this.gravitySpeed = 0;  
}
```

# 추가한 기능: 중력

바닥 부분을 설정

```
this.newPos = function() {  
    this.x += this.speedX;  
    this.y += this.speedY;  
}
```

바닥에 닿으면 속도를 0으로 설정

```
this.newPosPlayer = function() {  
    this.gravitySpeed += this.gravity;  
    this.x += this.speedX;  
    this.y += this.speedY + this.gravitySpeed;  
    var rockBottom = myGameArea.canvas.height - this.height;  
    if (this.y > rockBottom) {  
        this.y = rockBottom;  
        this.gravitySpeed = 0;  
    }  
}
```

장애물과 플레이어의 새  
위치 업데이트 좌표 함수를  
나눔

```
this.newPosObstacles = function() {  
    this.x += this.speedX;  
    this.y += this.speedY;  
}
```

02

JUMP

# 점프

두 번째 기능

# 추가한 기능: 점프

```
document.addEventListener('keydown', function(e) {  
  if (e.code === "Space") {  
    jump();  
  }  
});
```

Space를 'keyDown' 했을 때 jump() 함수를 실행

```
function jump() {  
  if (myGamePiece.y === myGameArea.canvas.height - myGamePiece.height) {  
    myGamePiece.speedY = -3;  
  }  
}
```

플레이어 객체가 바닥에 닿았을 때에 점프  
시키는 함수

```
<div style="text-align:center;width:480px;">  
  <button onmousedown="jump()" ontouchstart="jump()">JUMP</button><br><br>  
</div>
```

점프 버튼 추가



MOVING

# 추가한 기능: 이동

```
document.addEventListener('keydown', function(e) {  
  if (e.code === "Space") {  
    jump();  
  }  
});
```

방향키 입력 시 각 방향의 move 함수 실행

캔버스의 왼쪽 벽에 닿았을 경우엔  
더이상 왼쪽으로 움직이지 않음

```
document.addEventListener('keydown', function(e) {  
  if (e.code === "Space") {  
    jump();  
  }  
  if (e.code === "ArrowRight") {  
    moveright();  
  }  
  if (e.code === "ArrowLeft") {  
    if (myGamePiece.x > 0)  
      moveleft();  
  }  
});
```

방향키에서 손을 떼면  
clearmove로 이동을 멈춤

```
document.addEventListener('keyup', function(e) {  
  if (e.key === "ArrowRight" || e.key === "ArrowLeft") {  
    clearmove();  
  }  
});
```



MOVING

## 추가한 기능: 이동

```
function clearmove() {  
    myGamePiece.speedX = 0;  
    myGamePiece.speedY = 0;  
}
```

```
function clearmove() {  
    myGamePiece.speedX = 0;  
}
```

이동이 중단 되었을 때도, 점프는 지속되어야 하기  
때문에  
clearmove()에서 Y좌표 측 속도를 없애는 구문 삭제

MOVING

## 추가한 기능: 이동

```
this.newPosPlayer = function() {  
    this.gravitySpeed += this.gravity;  
    this.x += this.speedX;  
    this.y += this.speedY + this.gravitySpeed;  
    var rockBottom = myGameArea.canvas.height - this.height;  
    if (this.y > rockBottom) {  
        this.y = rockBottom;  
        this.gravitySpeed = 0;  
    }  
    if (this.y == rockBottom) {  
        this.speedY = 0;  
    }  
}
```

캔버스의 왼쪽에 닿으면  
x좌표 속도를 0으로 만듦

```
this.newPosPlayer = function() {  
    this.gravitySpeed += this.gravity;  
    this.x += this.speedX;  
    this.y += this.speedY + this.gravitySpeed;  
    var rockBottom = myGameArea.canvas.height - this.height;  
    var rockLeft = 0;  
    if (this.y > rockBottom) {  
        this.y = rockBottom;  
        this.gravitySpeed = 0;  
    }  
    if (this.y == rockBottom) {  
        this.speedY = 0;  
    }  
    if (this.x <= rockLeft) {  
        this.speedX = 0;  
    }  
}
```

MOVING

## 추가한 기능: 이동

```
<div style="text-align:center;width:480px;">  
  <button onmousedown="jump()" ontouchstart="jump()">JUMP</button><br><br>  
</div>
```

```
<div style="text-align:center;width:480px;">  
  <button onmousedown="moveleft()" onmouseup="clearmove()" ontouchstart="moveleft()">LEFT</button>  
  <button onmousedown="jump()" ontouchstart="jump()">JUMP</button>  
  <button onmousedown="moveright()" onmouseup="clearmove()" ontouchstart="moveright()">RIGHT</button><br><br>  
</div>
```

이동을 위한 LEFT, RIGHT 버튼 추가

04

RESTART

# 재시작

네 번째 기능

# 추가한 기능: 재시작

```
function restart() {  
  myGameArea.stop();  
  myGameArea.clear();  
  myGameArea.frameNo = 0;  
  myObstacles = [];  
  startGame();  
}
```

게임을 재시작하는 restart() 함수 구현



# 추가한 기능: 재시작

```
document.addEventListener('keydown', function(e) {  
  if (e.code === "Space") {  
    jump();  
  }  
  if (e.code === "ArrowRight") {  
    moveright();  
  }  
  if (e.code === "ArrowLeft") {  
    if (myGamePiece.x > 0)  
      moveleft();  
  }  
});
```

R키를 눌렀을 때  
게임을 재시작하는 구문  
추가

```
document.addEventListener('keydown', function(e) {  
  if (e.code === "Space") {  
    jump();  
  }  
  if (e.code === "ArrowRight") {  
    moveright();  
  }  
  if (e.code === "ArrowLeft") {  
    if (myGamePiece.x > 0)  
      moveleft();  
  }  
  if (e.key === 'r' || e.key === 'R') {  
    restart();  
  }  
});
```

RESTART

# 추가한 기능: 재시작

```
<div style="text-align:center;width:480px;">
  <button onmousedown="moveleft()" onmouseup="clearmove()" ontouchstart="moveleft()">LEFT</button>
  <button onmousedown="jump()" ontouchstart="jump()">JUMP</button>
  <button onmousedown="moveright()" onmouseup="clearmove()" ontouchstart="moveright()">RIGHT</button><br><br>
</div>
```

```
<div style="text-align:center;width:480px;">
  <button onmousedown="moveleft()" onmouseup="clearmove()" ontouchstart="moveleft()">LEFT</button>
  <button onmousedown="jump()" ontouchstart="jump()">JUMP</button>
  <button onmousedown="moveright()" onmouseup="clearmove()" ontouchstart="moveright()">RIGHT</button><br><br>
  <button onmousedown="restart()" ontouchstart="restart()">RESTART</button>
</div>
```

화면에 재시작 버튼  
추가



05

PLAYER IMAGE

# 플레이어 이미지

다섯 번째 기능

# 추가한 기능: 마리오

```
function startGame() {  
  myGamePiece = new component(30, 30, "red", 10, 120);  
  myScore = new component("30px", "Consolas", "black", 280, 40, "text");  
  myGameArea.start();  
}
```

player 객체를 채우던 색상  
red를  
이미지 파일 명으로 변경,  
타입 image 추가.

```
function startGame() {  
  myGamePiece = new component(30, 30, "mario.png", 10, 120, "image");  
  myScore = new component("30px", "Consolas", "black", 280, 40, "text");  
  myGameArea.start();  
}
```

# 추가한 기능: 마리오

```
this.update = function() {  
  ctx = myGameArea.context;  
  if (this.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);  
  }  
}
```

type image 추가

```
this.update = function() {  
  ctx = myGameArea.context;  
  if (this.type == "text") {  
    ctx.font = this.width + " " + this.height;  
    ctx.fillStyle = color;  
    ctx.fillText(this.text, this.x, this.y);  
  } else if (type == "image") {  
    ctx.drawImage(this.image,  
      this.x,  
      this.y,  
      this.width, this.height);  
  } else {  
    ctx.fillStyle = color;  
    ctx.fillRect(this.x, this.y, this.width, this.height);  
  }  
}
```

06

BACKGROUND IMAGE

배경

여섯 번째 기능

# 추가한 기능: 배경

```
canvas {  
  border: 1px solid #d3d3d3;  
  background-color: #f1f1f1;  
}
```

배경 이미지를 세팅하는  
구문 추가

```
canvas {  
  border: 1px solid #d3d3d3;  
  background-image: url('mario_background.jpg');  
  background-repeat: no-repeat;  
  background-size: cover;  
}
```

07

BACKGROUND MUSIC

BGM

일곱 번째 기능



# 추가한 기능: BGM

```
var bgm = new Audio("./mario_sound.mp3");  
var outro = new Audio("./outro.mov");
```

사용할 음악 파일들의 정보를 담은  
bgm, outro 변수 생성.

```
function startGame() {  
  myGamePiece = new component(30, 30, "mario.png", 10, 120, "image");  
  myScore = new component("30px", "Consolas", "black", 280, 40, "text");  
  myGameArea.start();  
}
```

```
function startGame() {  
  myGamePiece = new component(30, 30, "mario.png", 10, 120, "image");  
  myScore = new component("30px", "Consolas", "black", 280, 40, "text");  
  myGameArea.start();  
  bgm.play(); ← 게임 시작 시 BGM을 재생  
}
```



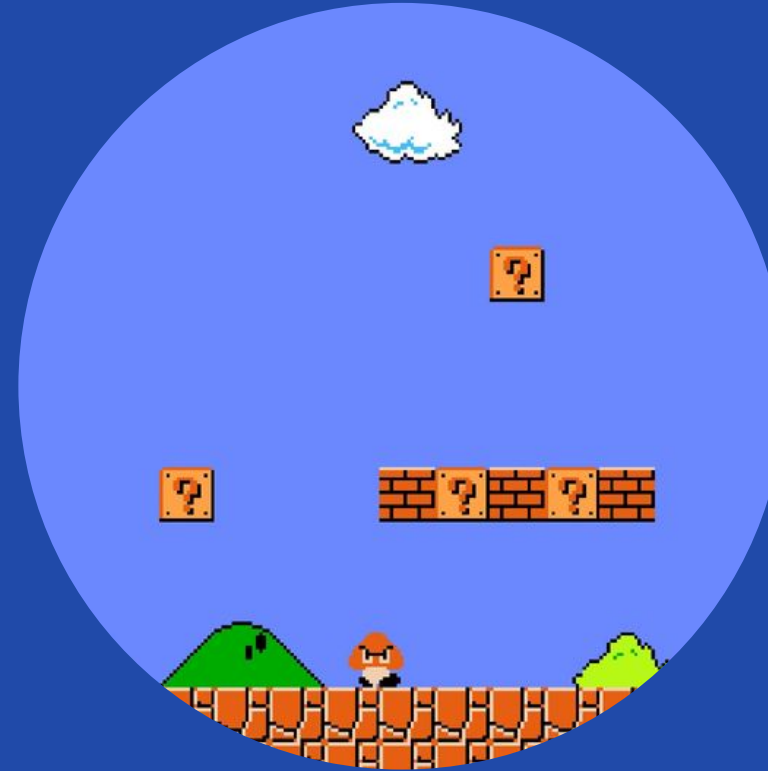
# 추가한 기능: BGM

```
function updateGameArea() {  
    var x, height, gap, minHeight, maxHeight, minGap, maxGap;  
    for (i = 0; i < myObstacles.length; i += 1) {  
        if (myGamePiece.crashWith(myObstacles[i])) {  
            myGameArea.stop();  
            return;  
        }  
    }  
}
```

장애물에 부딪혔을 경우  
배경음악을 멈추고  
outro 효과음을 추가

```
function updateGameArea() {  
    var x, height, gap, minHeight, maxHeight, minGap, maxGap;  
    for (i = 0; i < myObstacles.length; i += 1) {  
        if (myGamePiece.crashWith(myObstacles[i])) {  
            outro.play();  
            bgm.pause();  
            myGameArea.stop();  
            return;  
        }  
    }  
}
```

IMAGE



REFERENCE

# 출처

SOME CODE

SOUND



Thank you.

2018975058 전지환

2023.11.14 (수)