

# GAME

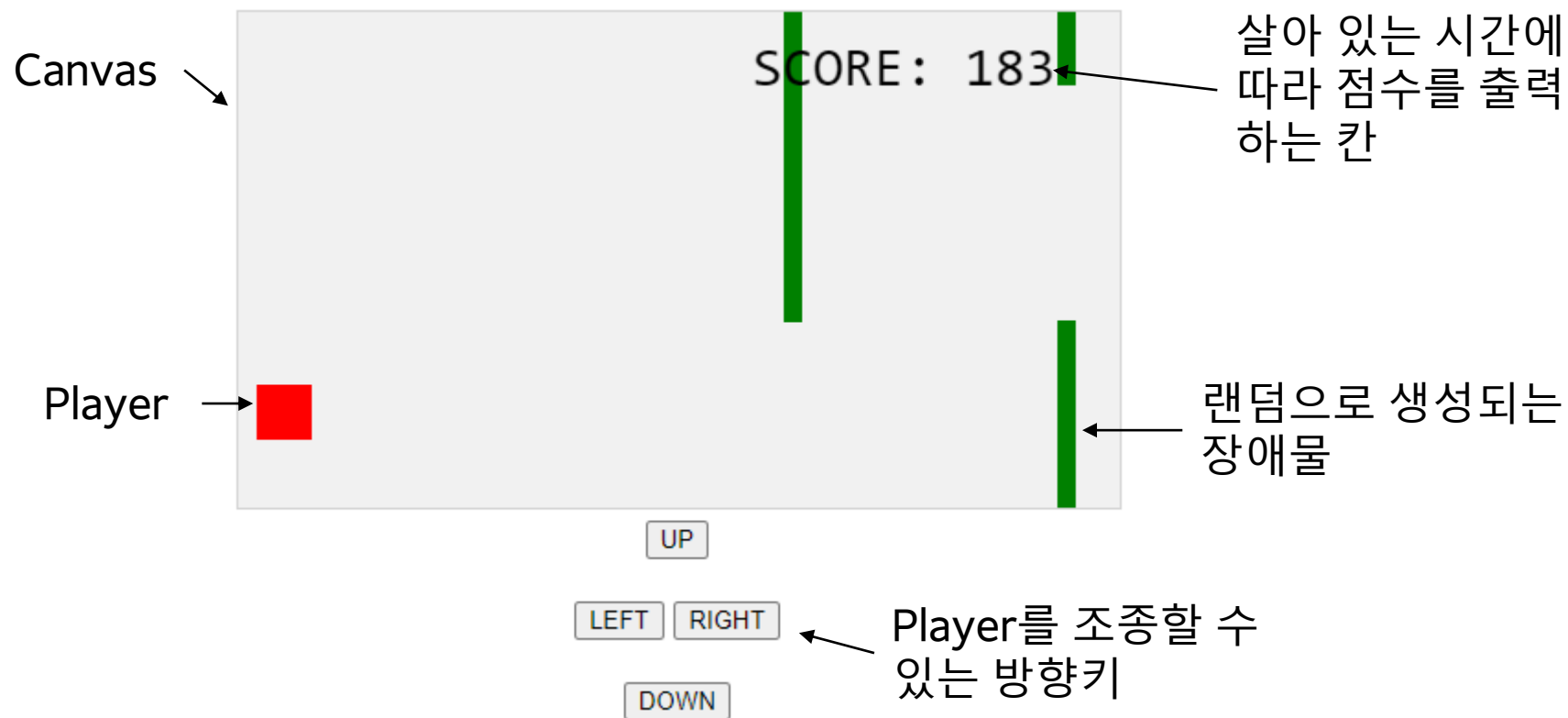
# 코드 분석

---

소프트웨어학과 / 게임프로그래밍

2021863030 박세희

# GAME UI



# myGameArea

---

```
var myGameArea = {
  canvas : document.createElement("canvas"),
  start : function() {
    this.canvas.width = 480;
    this.canvas.height = 270;
    this.context = this.canvas.getContext("2d");
    document.body.insertBefore(this.canvas,
document.body.childNodes[0]);
    this.frameNo = 0;
    this.interval = setInterval(updateGameArea, 20);
  },
  clear : function() {
    this.context.clearRect(0, 0, this.canvas.width,
this.canvas.height);
  },
  stop : function() {
    clearInterval(this.interval);
  }
}
```

- 게임 영역을 정의
- canvas : 게임 캔버스 요소를 생성, HTML 문서에 추가
- start : 게임 루프를 시작
- clear : 캔버스를 지우고 이전 프레임의 그래픽 요소를 제거
- stop : 게임 루프를 멈춤

# Component

---

```
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.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);
    }
  }
}
```

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

# Component

---

- 플레이어 캐릭터 또는 장애물의 게임 요소를 정의
- type : 요소의 종류를 나타낸다. Text 또는 shape으로 설정된다
- update : 요소를 그려주는 역할을 한다. Text일 경우 텍스트를, 그 외는 네모 상자를 그린다.
- newPos : 요소의 새로운 위치를 설정한다.
- crashWith : 두 요소가 충돌했는지 여부를 검사한다.

# updateGameArea()

---

```
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;
        }
    }
    myGameArea.clear();
    myGameArea.frameNo += 1;
    if (myGameArea.frameNo == 1 || everyinterval(150)) {
        x = myGameArea.canvas.width;
        minHeight = 20;
        maxHeight = 200;
        height = Math.floor(Math.random()*(maxHeight-
minHeight+1)+minHeight);
        minGap = 50;
        maxGap = 200;
        gap = Math.floor(Math.random()*(maxGap-
minGap+1)+minGap);
        myObstacles.push(new component(10, height,
"green", x, 0));
        myObstacles.push(new component(10, x - height -
gap, "green", x, height + gap));
    }
    for (i = 0; i < myObstacles.length; i += 1) {
        myObstacles[i].speedX = -1;
        myObstacles[i].newPos();
        myObstacles[i].update();
    }
    myScore.text="SCORE: " + myGameArea.frameNo;
    myScore.update();
    myGamePiece.newPos();
    myGamePiece.update();
}
```

- 게임 화면을 업데이트 한다.
- 장애물과 플레이어 캐릭터의 충돌을 감지, 충돌 시 게임을 종료한다.
- 새로운 프레임을 그리기 전에 이전 프레임을 지우고 장애물을 생성한다.
- 장애물은 왼쪽으로 이동하며, 프레임 수에 따라 점수를 증가시킨다.

# move...()

---

```
function moveup() {  
    myGamePiece.speedY = -1;  
}
```

```
function movedown() {  
    myGamePiece.speedY = 1;  
}
```

```
function moveleft() {  
    myGamePiece.speedX = -1;  
}
```

```
function moveright() {  
    myGamePiece.speedX = 1;  
}
```

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

- 사용자의 입력을 처리하여 플레이어 캐릭터를 움직인다.

- speedX 와 speedY 속성을 조절하여 움직임을 구현한다.

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

# everyinterval(n)

---

```
function everyinterval(n) {  
    if ((myGameArea.frameNo / n) % 1 == 0) {return  
true;}  
    return false;  
}
```

- 주어진 프레임 간격(n) 마다 어떤 작업을 수행해야 하는지 결정한다.
- myGameArea.frameNo 를 사용하여 판단한다.



# startGame()

---

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

- 게임 초기화를 담당
- myGamePiece : 객체를 생성하여 플레이어 캐릭터를 나타낸다
- myScore : 객체를 생성하여 게임 점수를 나타낸다
- myGameArea.start() : 매서드를 호출하여 게임 영역을 설정하고 게임 루프를 시작한다.

# Reference

---

- [https://www.w3schools.com/graphics/game\\_intro.asp](https://www.w3schools.com/graphics/game_intro.asp)
- <https://chat.openai.com>