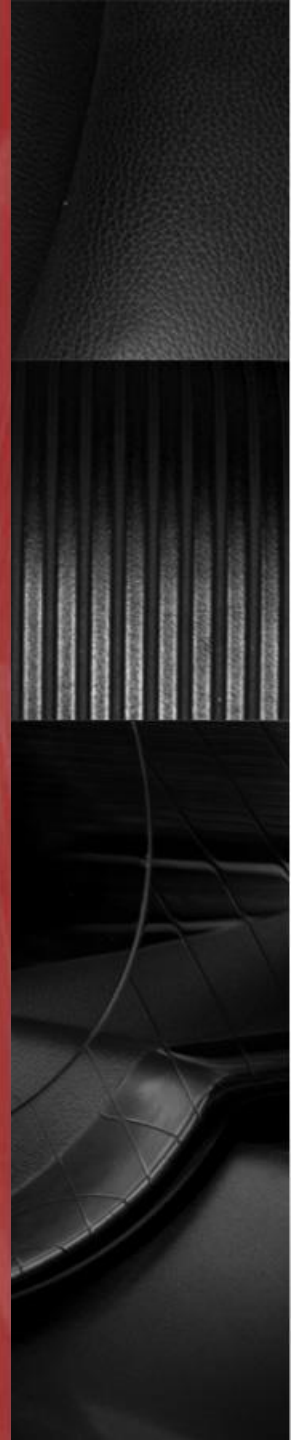
핑퐁 만들기 2

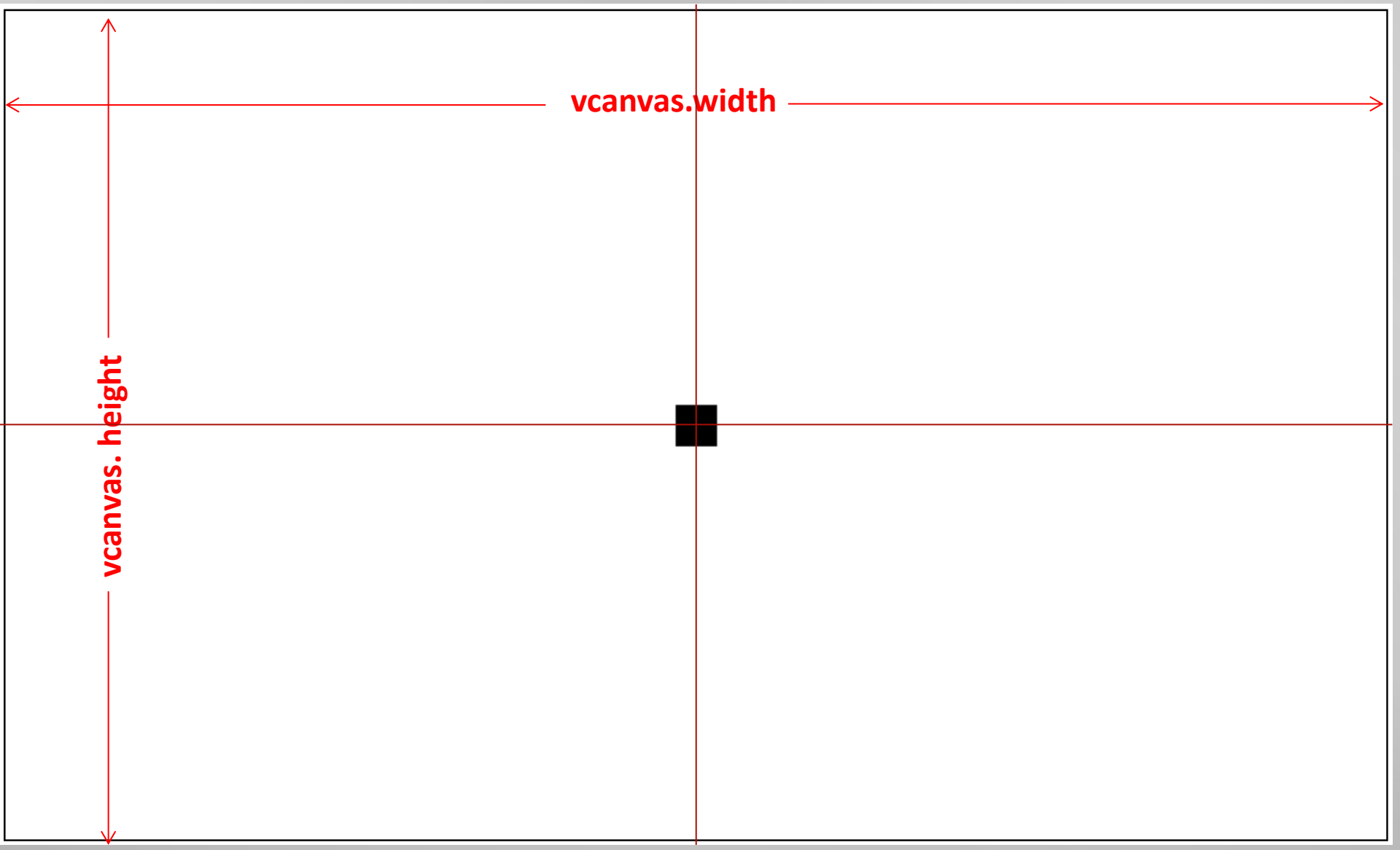


오늘 학습 내용

1. Key Control
2. Draw Racket
3. Draw Racket + Key Control
4. Racket & Ball
5. Collision
6. Bonus

1. Key Control

- 기본 프로그램
  - HTML 파일 : keyControl.html
  - JavaScript 파일 : keyControl.js
- 요구사항
  - canvas 크기 : width 1,000 px height 600 px
  - 큐브 크기 : 30 px
  - 위치 : 캔버스 정 중앙


vcanvas.width
vcanvas. height

keyControl.html ★★★

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>key Control</title>
    <script src="keyControl.js"></script>
</head>
<body onload="init()">

  <canvas id="myCanvas" width="1000" height="600" style="border:1px solid #000000;">
     이 브라우져는 HTML5 canvas 태그를 제공하기 않습니다.
  </canvas>

</body>
</html>
```

keyControl.js ★★★

```
'use strict';
var vcanvas, ctx;
var cubex,  cubey;
var cubewidth = 30,  cubeheight = 30;

function drawCube() {
    ctx.fillRect(cubex, cubey, cubewidth, cubeheight);
}

function init() {
    vcanvas = document.getElementById("myCanvas");
    ctx = vcanvas.getContext("2d");

    cubex = vcanvas.width / 2 - cubewidth / 2;
    cubey = vcanvas.height / 2 - cubeheight / 2;

    drawCube();
}
```

Key Control
적용 전 코드

추가1: Key Event

- **document.onkeydown = set_key;**
  - 도큐먼트에 키가 눌리는 이벤트가 발생 시 함수 set_key 가 실행 됨
- **document.onkeyup = stop_key;**
  - 도큐먼트에 눌렸던 키가 놓아지는 이벤트가 발생 시 함수 stop_key 가 실행 됨

추가2: key Control 에 사용될 전역변수

변수들

- r_left
  - ASCII 코드 37 ←
- r_up
  - ASCII 코드 38 ↑
- r_right
  - ASCII 코드 39 →
- r_down
  - ASCII 코드 40 ↓

상태

- onkeydown 일 경우 : 1
- onkeyup 일 경우 : 0

추가3: 함수 : set_key() & stop_key()

- set_key()

```
function set_key() {
   if (event.keyCode === 37) {  r_left = 1;   }
   if (event.keyCode === 38) {  r_up = 1;  }
   if (event.keyCode === 39) {  r_right = 1;  }
   if (event.keyCode === 40) {  r_down = 1;  }
}
```

- stop_key()

```
function stop_key() {
   if (event.keyCode === 37) {  r_left = 0;   }
   if (event.keyCode === 38) {  r_up = 0;  }
   if (event.keyCode === 39) {  r_right = 0;  }
   if (event.keyCode === 40) {  r_down = 0;  }
}
```

추가 되어야 할 것들

- 수정 전

```
'use strict';
var ... :

function drawBox() {}
function init() {
                :
  drawBox();
}
```

```
var ...;
var velocity = 10;

var r_left, r_up, r_right, r_down;

function set_key() { }
function stop_key() { }

document.onkeydown = set_key;
document.onkeyup = stop_key;

function drawBox() { }
function update() { }
function gameLoop() { }
function init() {
                :
   setInterval(gameLoop, 30);
}
```

keyControl.js 에 추가될 영역

```
'use strict';
var vcanvas, ctx;
var cubex,  cubey;
var cubewidth = 30,  cubeheight = 30;
```

전역변수 추가

```
function drawBox() {
    ctx.fillRect(cubex, cubey, cubewidth, cubeheight);
}
```

함수 & 이베트 추가

```
function init() {
    vcanvas = document.getElementById("myCanvas");
    ctx = vcanvas.getContext("2d");

    cubex = vcanvas.width / 2 - cubewidth / 2;
    cubey = vcanvas.height / 2 - cubeheight / 2;

    drawBox();  setInterval(gameLoop, 30);
}
```

함수 콜 수정

문제점

- 캔버스를 지나쳐 계속 이동
- 이유?

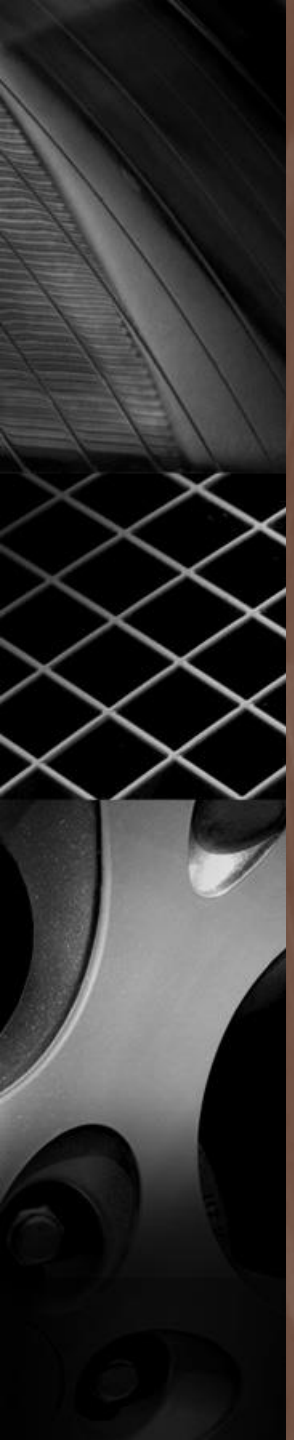
- 해결 방법
  1. 큐브가 캔버스의 오른쪽 끝에 도달했는지 판단

     그렇다면: 속도의 방향을 바꿔준다 즉 racketspeed *= -1;*
  2. 큐브가 캔버스의 왼쪽 끝에 도달했는지 판단

     그렇다면: 속도의 방향을 바꿔준다 즉 racketspeed *= -1;*

문제 해결

```
function update() {
    if (r_up) { cubey -= velocity; }
    if (r_down) { cubey += velocity; }
    if (r_left) { cubex -= velocity; }
    if (r_right) { cubex += velocity; }

    // ----- 보정
    if (cubex + cubewidth > vcanvas.width) { cubex = vcanvas.width - cubewidth; }
    if (cubex < 0) { cubex = 0; }
    if (cubey + cubeheight > vcanvas.height) { cubey = vcanvas.height - cubeheight; }
    if (cubey < 0) { cubey = 0; }
}
```



keyControl.html

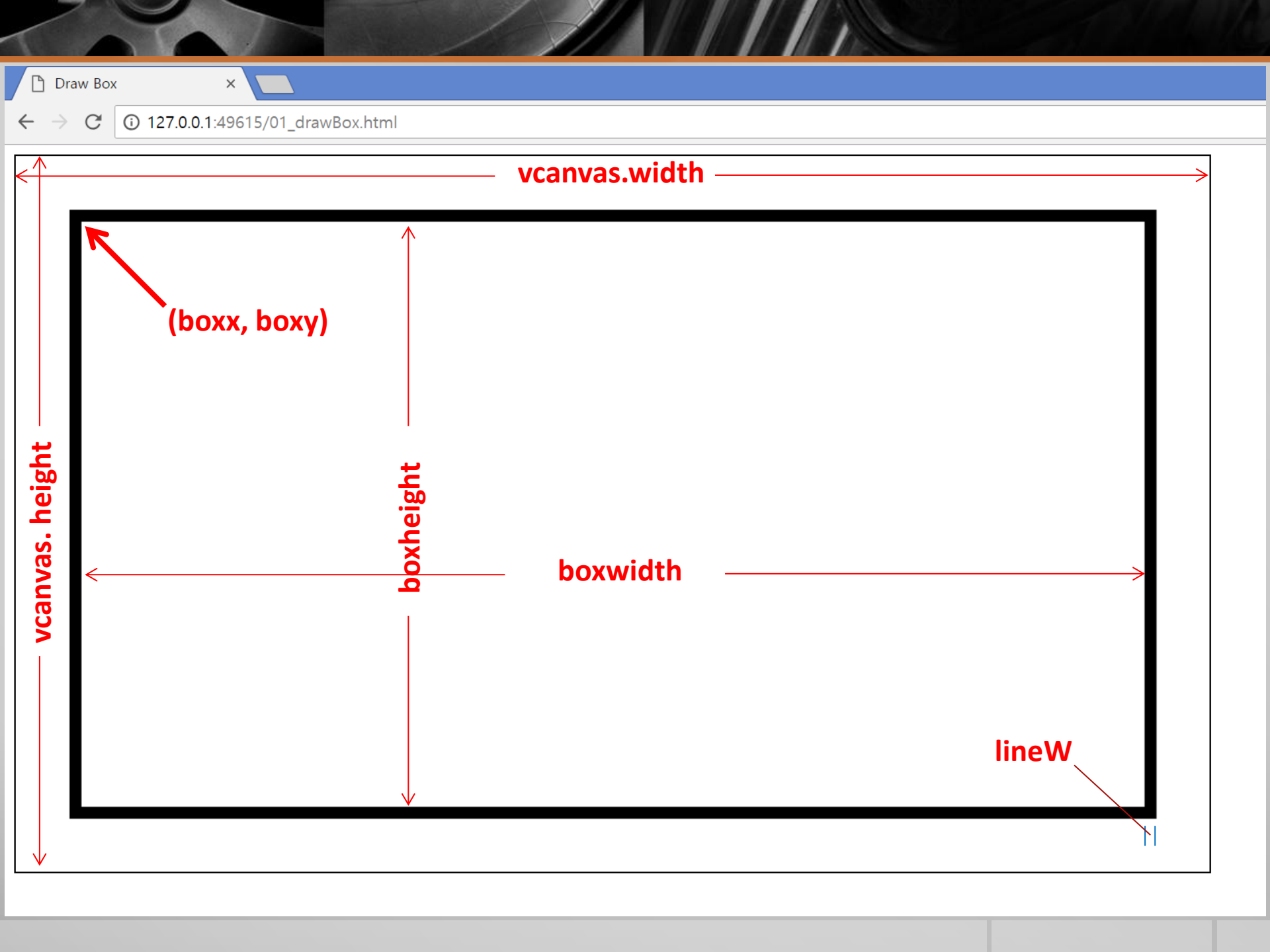
keyControl.js

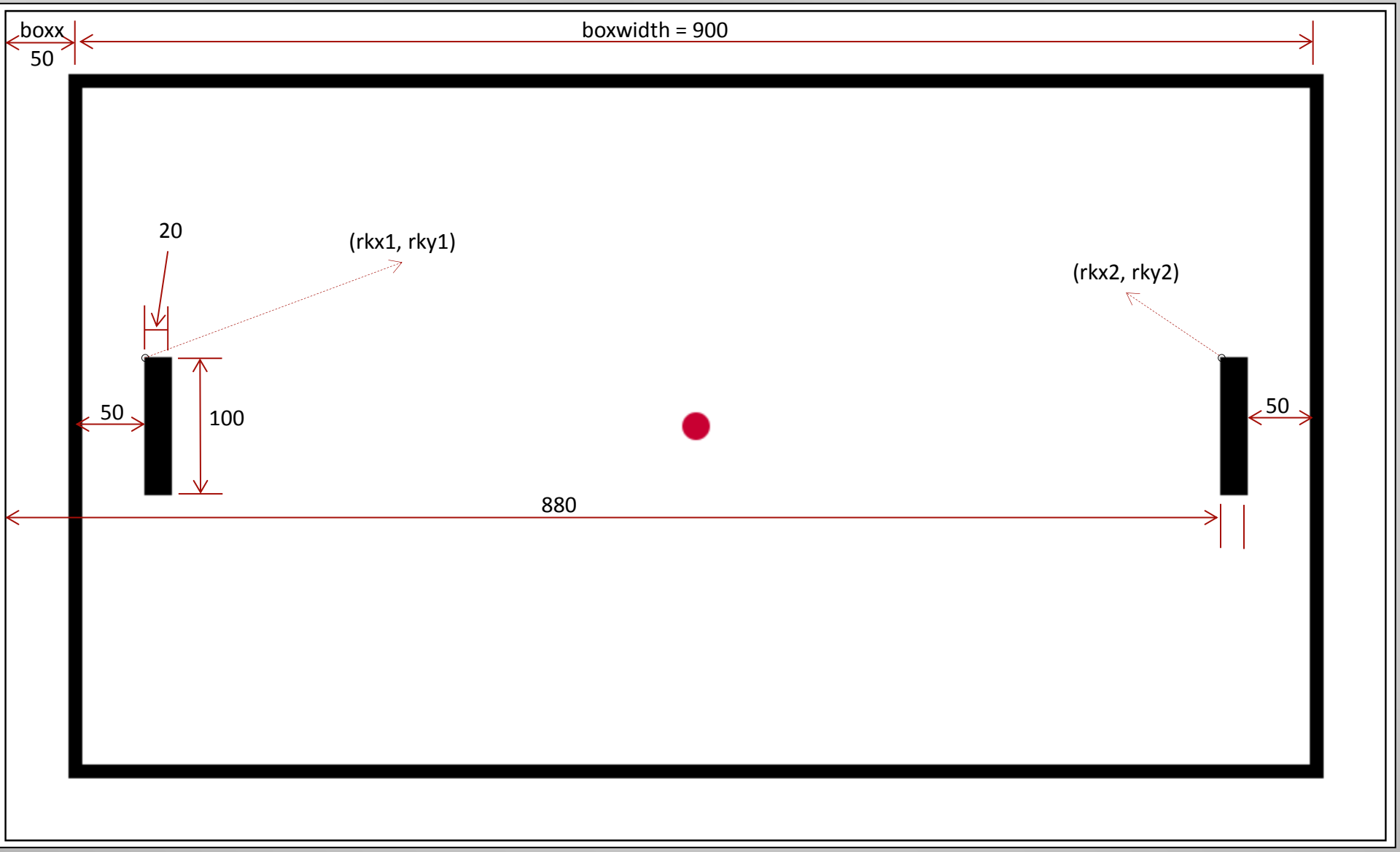
실습 1 : 키 컨트롤

2. Draw Racket

- 실습 파일명
  - drawRacket.html
  - drawRacket.js

```
'use strict';
var vcanvas, ctx;

var boxx = 50;
var boxy = 50;
var boxwidth = 900;
var boxheight = 500;
var lineW = 10;

var ballx = boxx + boxwidth / 2;
var bally = boxy + boxheight / 2;
var ballRad = 10;

function drawBox() {
   ctx.lineWidth = lineW;
   ctx.strokeRect(boxx, boxy, boxwidth, boxheight);
}

function drawBall() {
   ctx.beginPath();
   ctx.arc(ballx, bally, ballRad, 0, Math.PI * 2);
   ctx.fillStyle = "rgb(200,0,50)";
   ctx.fill();
}

function init() {
   vcanvas = document.getElementById("myCanvas");
   ctx = vcanvas.getContext("2d");

   drawBox();
   drawBall();
}
```


Draw Box
127.0.0.1:49615/01_drawBox.html
vcanvas.width
(boxx, boxy)
vcanvas. height
boxheight
boxwidth
lineW


boxx
50
boxwidth = 900
20
(rkx1, rky1)
(rkx2, rky2)
50
100
50
880

Racket 그리기에 추가 될 요소

- 전역 변수

```
var rkx1 = 100, rkx2 = 880, rkW = 20, rkH = 100;
var rky1 = boxy + boxheight / 2 - rkH / 2;
var rky2 = boxy + boxheight / 2 - rkH / 2;
```

- 함수

```
function drawRacket() {
    ctx.fillStyle = "black";
    ctx.fillRect(rkx1, rky1, rkW, rkH);
    ctx.fillRect(rkx2, rky2, rkW, rkH);
}
```

- 함수 콜

```
drawRacket();
```

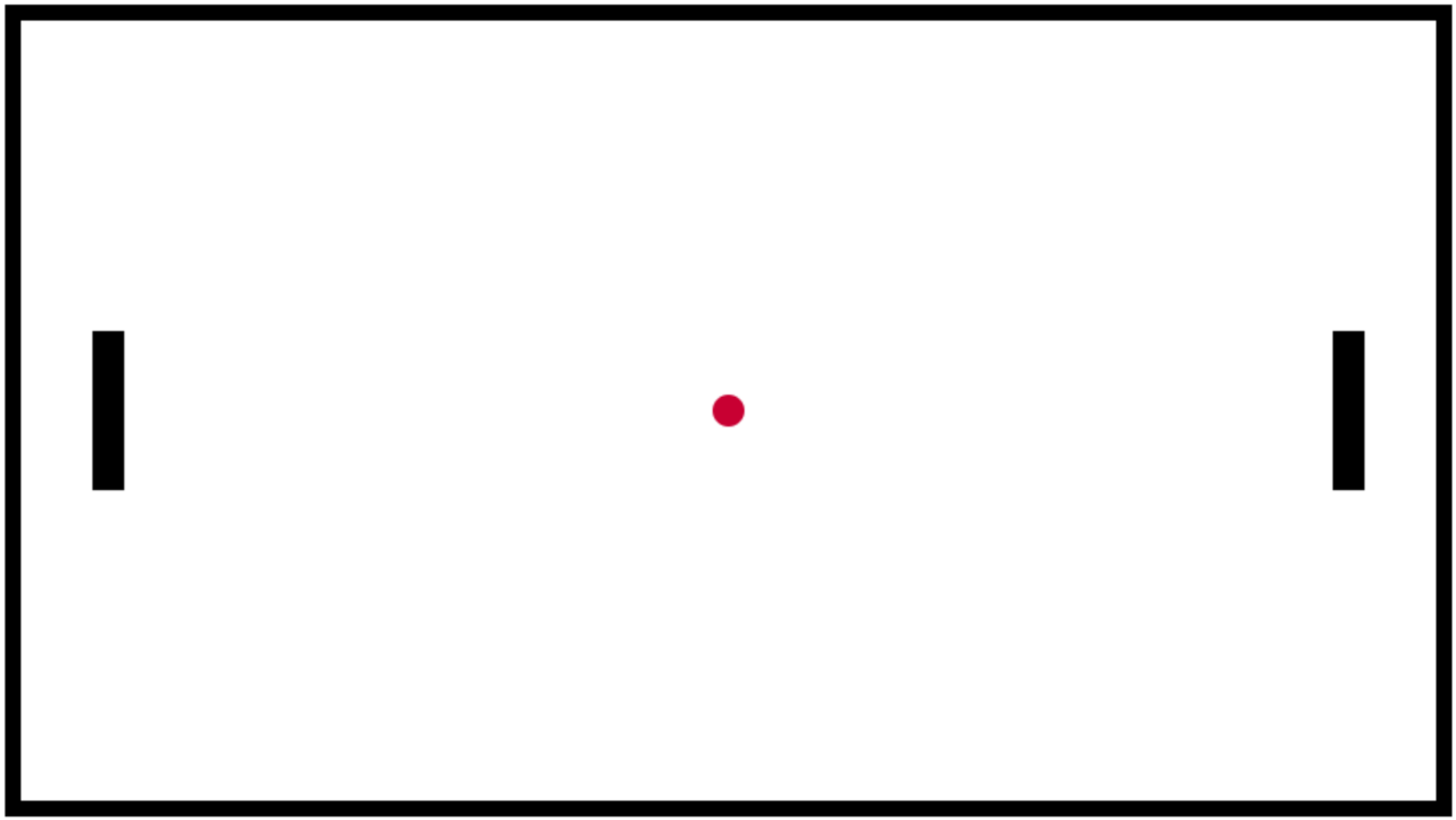
drawRacket.js 에 추가 될 영역

```
var ...;
function drawBox() {
    ctx.lineWidth = lineW;
    ctx.strokeRect(boxx, boxy, boxwidth, boxheight);
}
function drawBall() {
    ctx.beginPath();
    ctx.arc(ballx, bally, ballRad, 0, Math.PI * 2);
    ctx.fillStyle = "rgb(200,0,50)";
    ctx.fill();
}

function init() {
        :
    drawBox();
    drawBall();
}
```

전역변수 추가

함수 추가

새 함수 콜 추가



3. Draw Racket + key Control : 추가될 요소

- 전역변수

```
var l_up, l_down, r_up, r_down;
var racketspeed = 10;
```

- 함수

```
function update()
function gameLoop()
function set_key()
function stop_key()
```

- 이벤트

```
document.onkeydown = set_key;
document.onkeyup = stop_key;
```

- 함수콜

```
setInterval(gameLoop, 30);
```

문제점

- 박스를 지나쳐 계속 이동
- 이유?

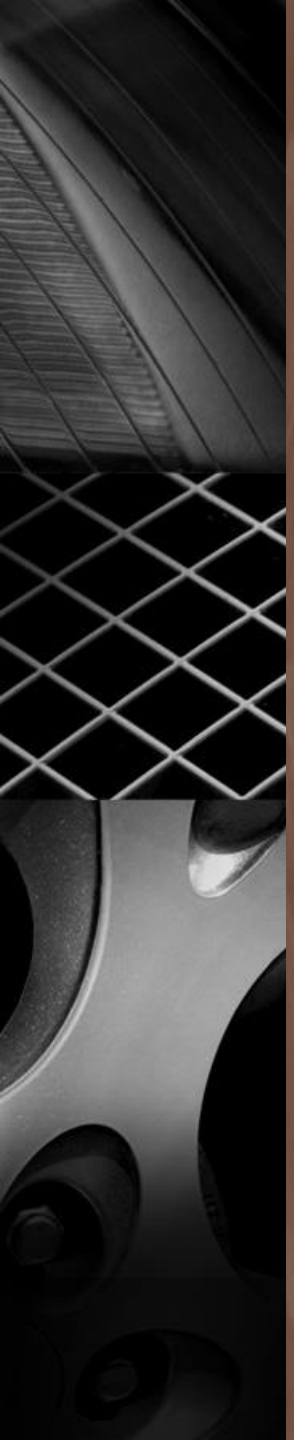
- 해결 방법
  1. 볼이 박스의 오른쪽 끝에 도달했는지 판단

     그렇다면: 속도의 방향을 바꿔준다 즉 racketspeed *= -1;*
  2. 볼이 박스의 왼쪽 끝에 도달했는지 판단

     그렇다면: 속도의 방향을 바꿔준다 즉 racketspeed *= -1;*

문제 해결

```
function update() {
   // Racket 이동
   if (l_up) { rky1 -= racketspeed; }
   if (l_down) { rky1 += racketspeed; }

   if (r_up) { rky2 -= racketspeed; }
   if (r_down) { rky2 += racketspeed; }

   // -----  보정
   if (rky1 <= boxy) { rky1 = boxy; }
   if (rky1 >= boxy + boxheight - rkH) { rky1 = boxy + boxheight - rkH; }

   if (rky2 <= boxy) { rky2 = boxy; }
   if (rky2 >= boxy + boxheight - rkH) { rky2 = boxy + boxheight - rkH; }
}
```



drawRacket.html

drawRacket.js

실습 2 : Draw Racket

4. Racket & Ball : 움직이는 볼 추가

- 기본 프로그램
  - HTML 파일 : Racket & Ball.html
  - JavaScript 파일 : Racket & Ball.js

- 요구사항
  - Key Control 이 추가된 Racket 이 완성된 프로그램에
  - 움직이는 볼을 추가 할 것

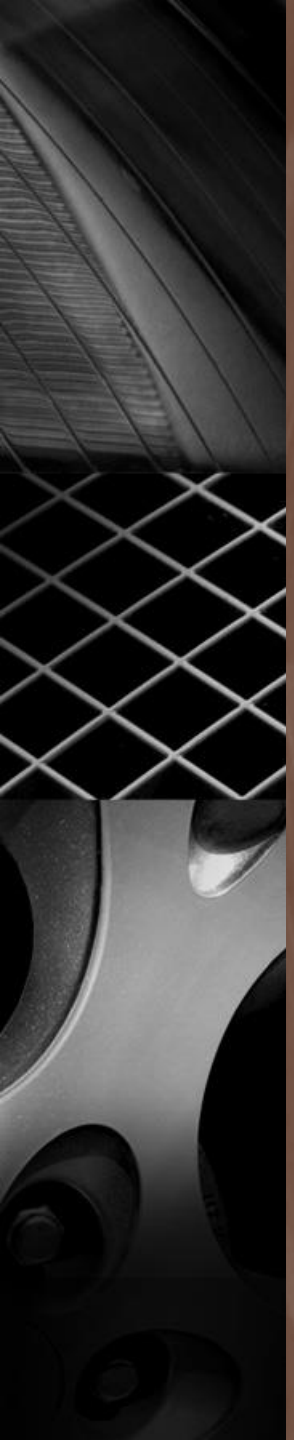
추가 될 요소

- **전역 변수**

```
var velocityX = 8;
    // x축으로 볼이 움직이는 속도

var velocityY = 8;
    // y축으로 볼이 움직이는 속도
```

- **update 함수 안에 추가**

```
ballx += velocityX;
bally += velocityY;

if (ballx + ballRad > boxx + boxwidth) {
  ballx = boxx + boxwidth - ballRad;
  velocityX *= -1;
}

if (ballx - ballRad < boxx) {
  ballx = boxx + ballRad;
  velocityX *= -1;
}

if (bally + ballRad > boxy + boxheight) {
  bally = boxy + boxheight - ballRad;
  velocityY *= -1;
}

if (bally - ballRad < boxy) {
  bally = boxy + ballRad;
  velocityY *= -1;
}
```



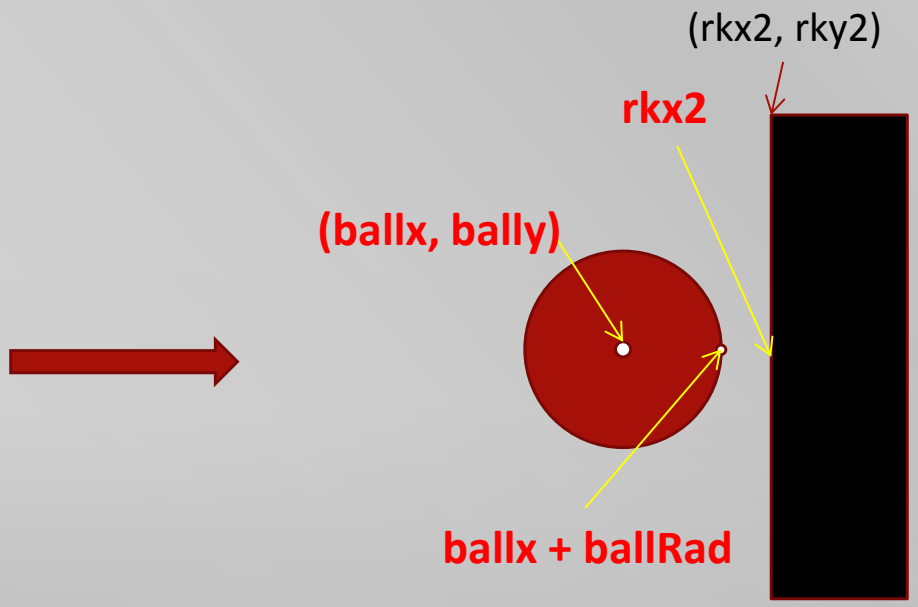
Racket & Ball.html

Racket & Ball.js

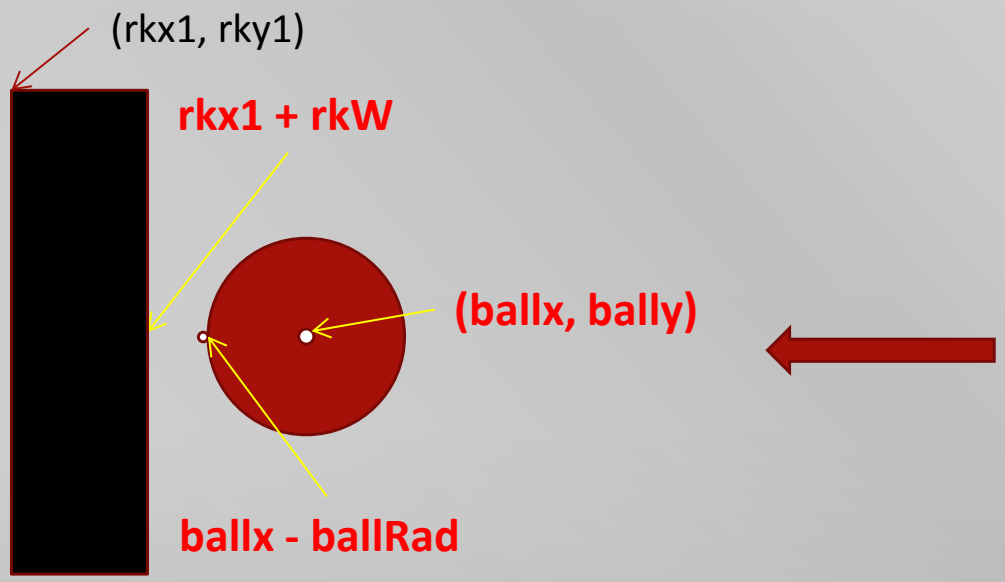
실습 3 : Racket & Ball

5. Collision

```
if (ballx + ballRad > rkx2 && bally > rky2 && bally < rky2 + rkH && velocityX > 0 && ballx < rkx2) {
        ballx = rkx2 - ballRad;
        velocityX = -velocityX;
}
```



```
if (ballx - ballRad < rkx1 + rkW && bally > rky1 && bally < rky1 + rkH && velocityX < 0 && ballx > rkx1 + rkW) {
    ballx = rkx1 + rkW + ballRad;
    velocityX = -velocityX;
}
```



Score & Court 추가

- 전역변수

  ```
  var score1 = 0, score2 = 0;
  ```

- 함수추가

  ```
  function drawCourt();  // 코트 그려주는 함수
  ```

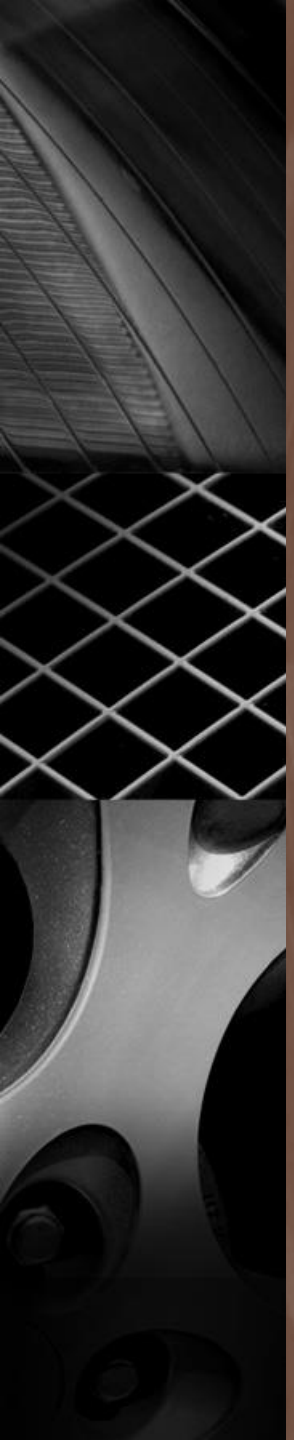
- 함수수정 : function update() 수정
  - 양쪽 벽에 충돌시 상대편 점수 증가

    ```
    score1 += 1;
    ```

  - Score 가 9 이상일 경우 0부터 다시 시작

    ```
    if (score1 > 9) { score1 = 0;
    if (score2 > 9) { score2 = 0; }
    ```

```
function drawCourt() {
    var x = vcanvas.width / 2, y = vcanvas.height / 2, rad = vcanvas.height / 4;

    ctx.beginPath();
    ctx.arc(x, y, rad, 0, Math.PI * 2, true);
    ctx.lineWidth = 5;
    ctx.stroke();
    ctx.beginPath();
    ctx.moveTo(x, boxy);
    ctx.lineTo(x, boxy + boxheight);
    ctx.stroke();

    ctx.font = "130px Arial";
    ctx.fillText(score1, x - 110, y + 40);
    ctx.fillText(score2, x + 40, y + 40);
}
```



PingPong.html

PingPong.js

실습 4 : PingPong