

# Mobile Simulation

2017-2



# Weekly plan (HTML5, 1<sup>st</sup> semester 2017)

- **wk01 : Introduction to curriculum & current state of HTML5**
- **wk02 : Making HTML5 documents**
- **wk03 : Table, iframe and media**
- **wk04 : Semantic tag and Form**
- **wk05 : CSS3 I. Basic**
- **wk06 : CSS3 II. Advanced**
- **wk07 : CSS3 III. Animation**
- **wk08 : Mid-term Exam.**
- **wk09 : Javascript : Data types & operators**
- **wk10 : Javascript : Loop & functions**
- **wk11 : Javascript : Core objects**
- **wk12 : Javascript : DOM**
- **wk13 : Javascript : Event handling I**
- **wk14 : Javascript : Event handling II**
- **wk15 : Final exam.**

# Weekly plan (Mobile Simulation, 2<sup>nd</sup> semester 2017)

- **wk01 : Introduction to curriculum & current state**
- **wk02 : Browser Object Model (BOM), installing Brackets editor**
- **wk03 : Canvas graphics I. Basic**
- **wk04 : Canvas graphics II. Image & Transformation**
- **wk05 : Canvas graphics III. Animation**
- **wk06 : Canvas graphics IV. Game**
- **wk07 : 보강 기간에 보강 실시**
- **wk08 : Mid-term Exam.**
- **wk09 :**
- **wk10 :**
- **wk11 :**
- **wk12 :**
- **wk13 :**
- **wk14 :**
- **wk15 : Final exam.**

Canvas simulation : MSnn Game



속도  각도   
점수 = 3

# 과제04. msnn\_rpt04.zip

4

[실습과제04] Christmas card

[1] Bouncing ball 완성. - [msnn\\_bouncing.html](http://msnn_bouncing.html)

[2] 캔버스와 자바스크립트만을 사용해서 크리스마스 카드 만들기

- 사각형 눈송이의 모양을 원형으로 변경
- 크리스마스 관련 그림을 활용 (배경 또는 애니메이션 추가)
- [msnn\\_rpt04.html](http://msnn_rpt04.html)

\*\*\*\* html 파일 및 관련 파일(image 등..)을  
MSnn\_Rpt04.zip 으로 압축해서 제출하시오.

파일명: [msnn\\_bouncing.html](http://msnn_bouncing.html), [msnn\\_rpt04.html](http://msnn_rpt04.html)

가점: Javascript 프로그래밍 응용 능력.

[제출파일] msnn\_rpt04.zip

두 개의 html 파일과 사용된 그림을 이메일로 제출

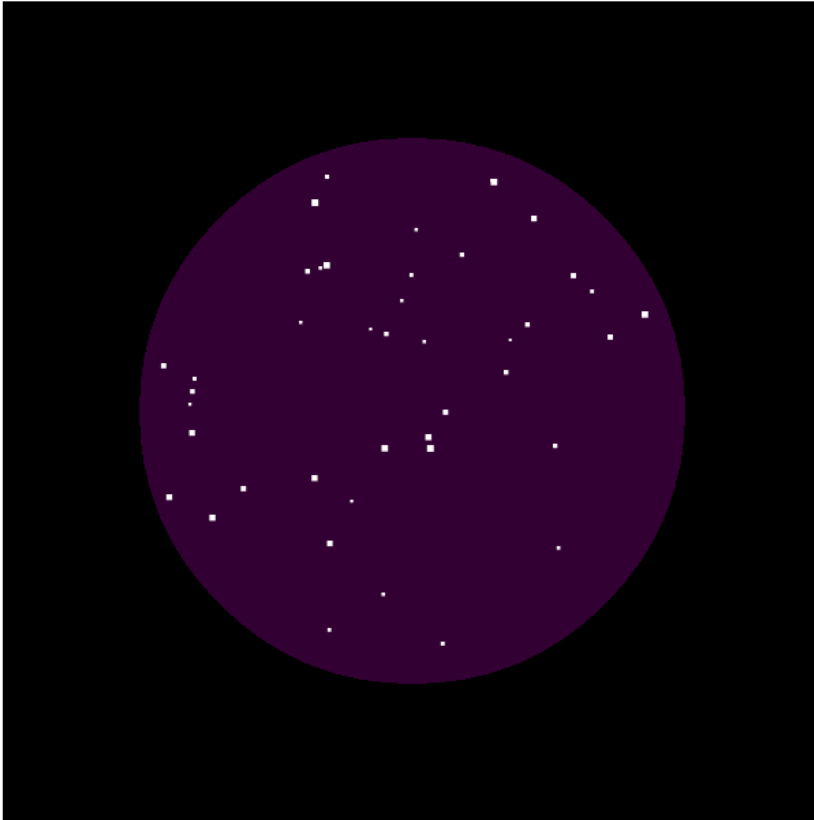
Email : chaos21c@gmail.com



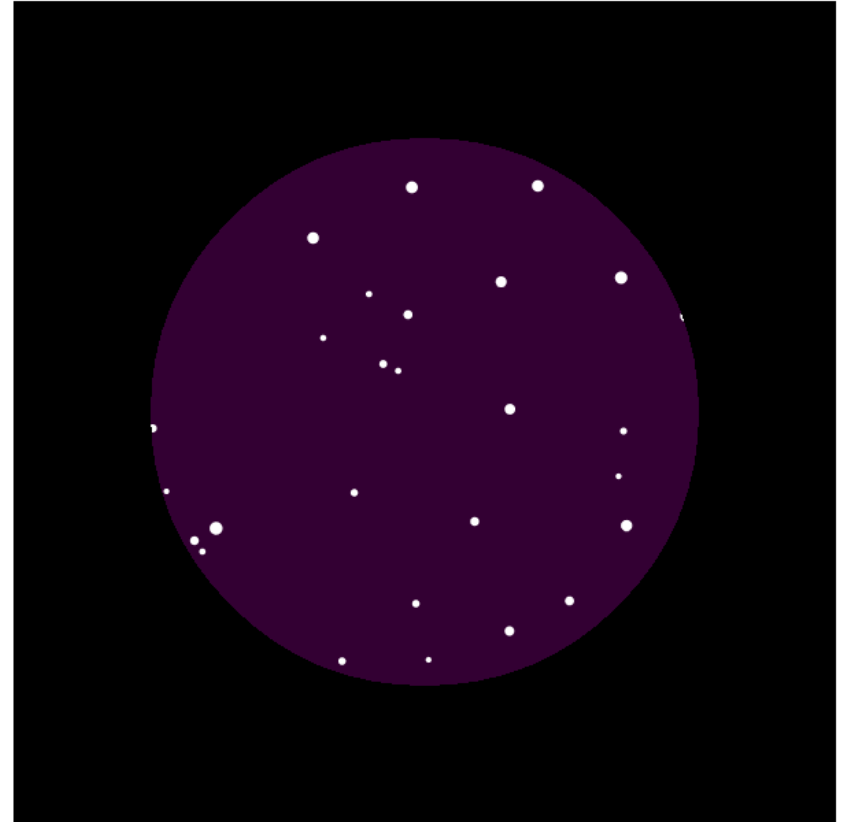
# 과제04. result

5

**Canvas simulation : Double Buffering**



**Canvas simulation : Double Buffering**

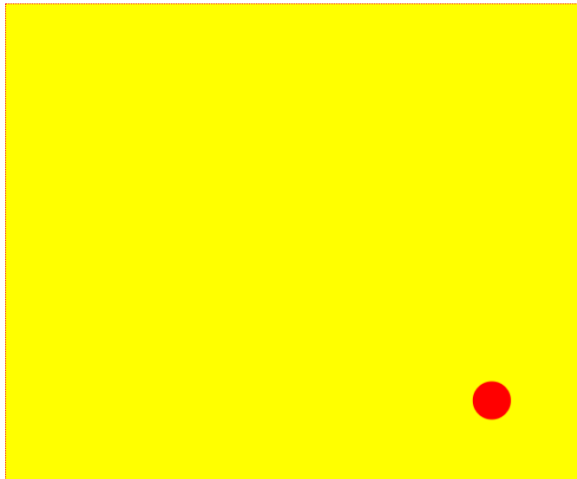


**Round snowflakes and clipping effect**

# 과제04. Good results

6

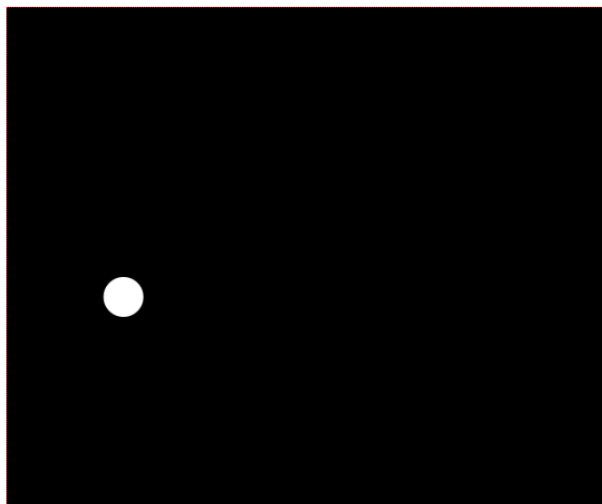
Canvas simulation : Bouncing ball



START STOP



```
function stop() {  
    dx = 0;  
    dy = 0;  
}  
function start() {  
    dx=Math.random()*10+10;  
    dy=Math.random()*10+10;  
}  
</script>  
<div>  
    <button onclick="start()">START</button>  
    <button onclick="stop()">STOP</button>  
</div>
```



정지!!

재시작!!



```
function stop(){  
    clearInterval(c);  
}  
  
</script>  
<div>  
    <button onclick = "stop()"> 정지!! </button>  
</div>  
<div>  
    <button onclick = "init()"> 재시작!! </button>  
</div>
```

# 과제04. Good results

7



▶ 0:00 / 2:51



```
// global variables
var canvas = null;
var context = null;
var bufferCanvas = null;
var bufferCanvasCtx = null;
var flakeArray = [];
var flakeTimer = null;
var maxFlakes = 200;
// prepare image
var imgA = new Image();
imgA.src = "images/crs.jpg";
```



```
// Clear buffer canvas
function blank() {
    // draw image on bufferCanvas
    bufferCanvasCtx.drawImage(imgA, 0, 0, bufferCanvasCtx.canvas.width,
    bufferCanvasCtx.canvas.height);
    // draw text on bufferCanvas
    var msg = "Merry Christmas";
    bufferCanvasCtx.font = "50px 'Nanum Gothic'";
    bufferCanvasCtx.fillText(msg, 145, 150);
}
```





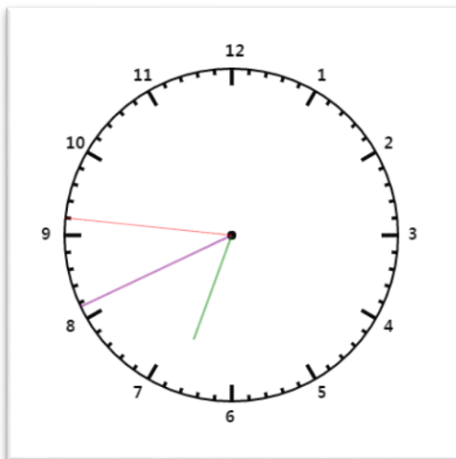


# Canvas IV. Game on Canvas

9



The <canvas> element is used to draw graphics, on the fly, on a web page.



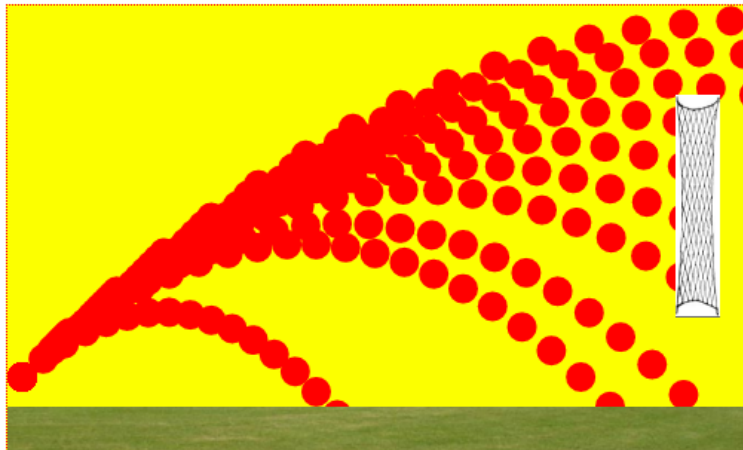
## Canvas simulation : My Game



# Target:

## Making HTML5 Game using Canvas

Canvas simulation : My Game



속도 20    각도 45

점수=4

발사

궤도지우기

Canvas simulation : My Game



속도 44    각도 36

점수=5

발사

궤도지우기

# Reference

---

## **HTML5 Canvas**

[http://www.w3schools.com/html/html5\\_canvas.asp](http://www.w3schools.com/html/html5_canvas.asp)

## **HTML DOM Canvas Object**

[http://www.w3schools.com/jsref/dom\\_obj\\_canvas.asp](http://www.w3schools.com/jsref/dom_obj_canvas.asp)

## **HTML Canvas Reference**

[http://www.w3schools.com/tags/ref\\_canvas.asp](http://www.w3schools.com/tags/ref_canvas.asp)

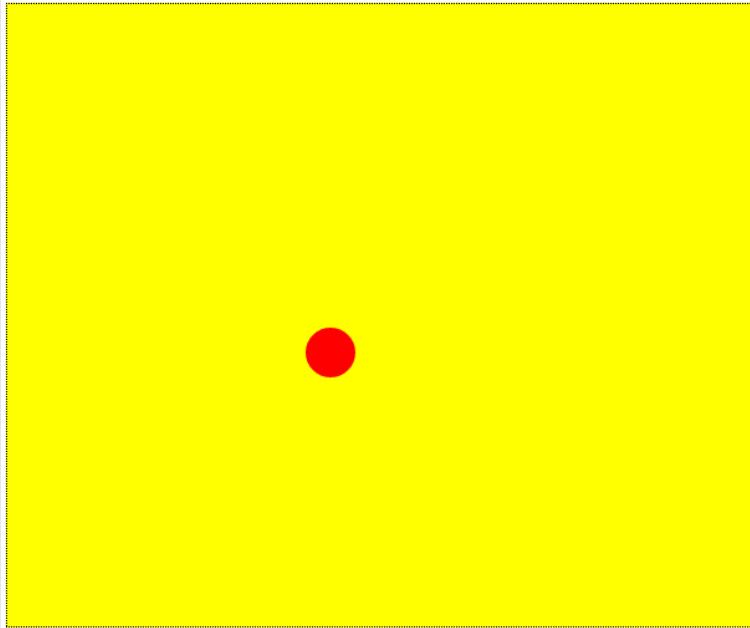
# Core algorithm

1. Two dimensional motion
2. x-direction : constant velocity  
y-direction : constant acceleration
3. Collision test  
between cannonball & target



# Core algorithm - 1

Canvas simulation : Bouncing Ball II



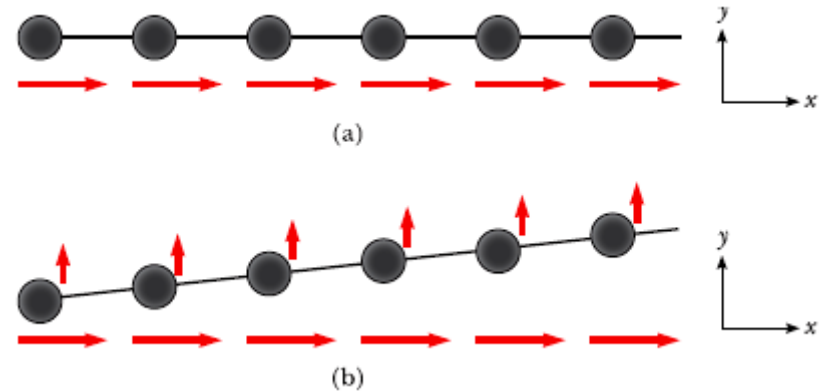
```
var dx = 10;  
var dy = 10;  
var x = 100;  
var y = 100;  
var r = 20;
```

```
if (x < (0 + r) || x > (x_max - r - dx))  
    dx = -dx;  
if (y < (0 + r) || y > (y_max - r - dy))  
    dy = -dy;  
x += dx;  
y += dy;
```

## 4.2.1 등가속도 이차원 운동 (대학물리학, Serway)

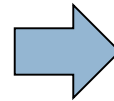
### (Two-Dimensional Motion with Constant Acceleration)

이차원 운동은  $x$ 와  $y$  축 방향의 각각 독립된 두 개의 운동으로 기술될 수 있다. 즉,  $y$  방향으로의 어떠한 영향도  $x$  방향의 운동에 영향을 주지 않는다. 그리고 그 반대의 경우도 마찬가지이다.



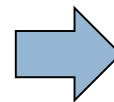
$$\mathbf{r} = x\mathbf{i} + y\mathbf{j}$$

$$\mathbf{v} = \frac{d\mathbf{r}}{dt} = \frac{dx}{dt}\mathbf{i} + \frac{dy}{dt}\mathbf{j} = v_x\mathbf{i} + v_y\mathbf{j}$$



$$v_x = \frac{dx}{dt} \quad v_y = \frac{dy}{dt}$$

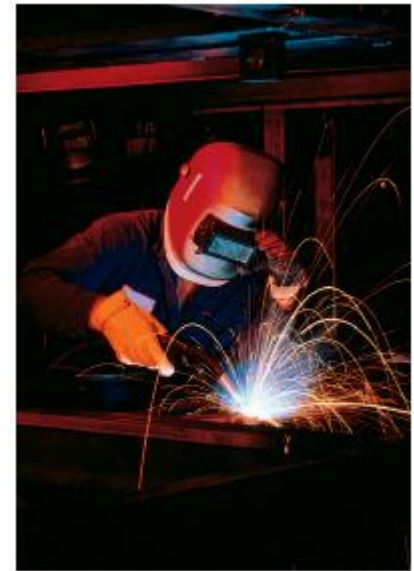
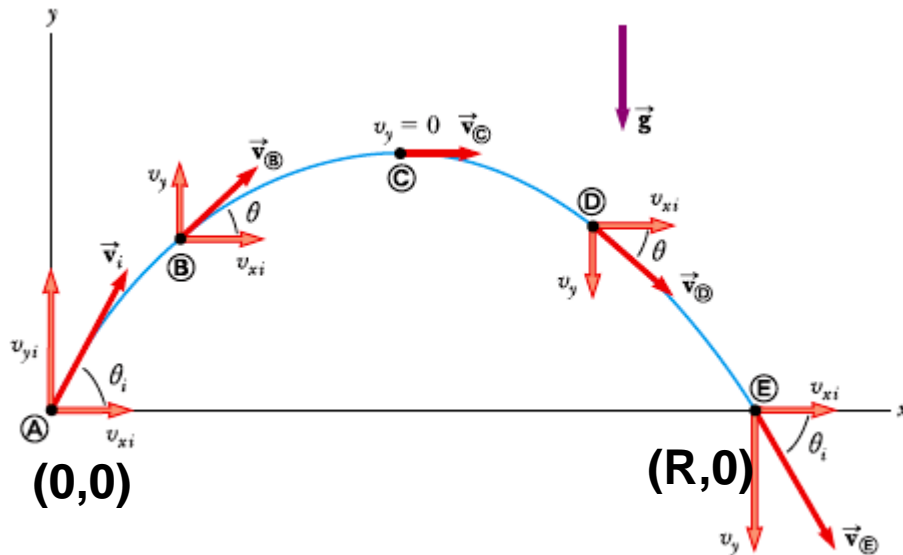
$$\mathbf{a} = \frac{d\mathbf{v}}{dt} = \frac{dv_x}{dt}\mathbf{i} + \frac{dv_y}{dt}\mathbf{j} = a_x\mathbf{i} + a_y\mathbf{j}$$



$$a_x = \frac{dv_x}{dt} \quad a_y = \frac{dv_y}{dt}$$

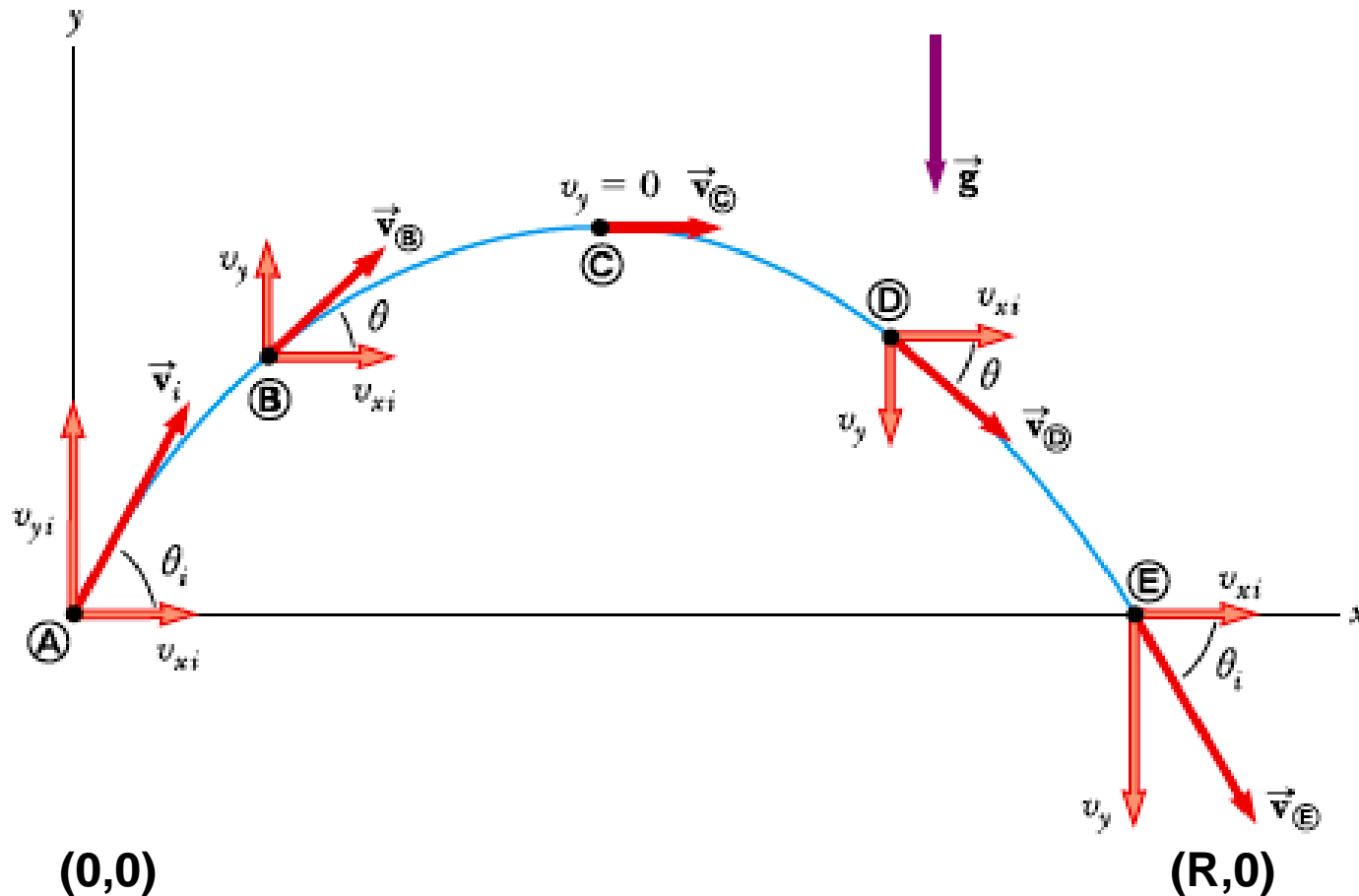
## 4.3.1 포물체 운동 (Projectile Motion)

- (1) 자유 낙하 가속도는 일정하고 아래를 향한다.
- (2) 공기 저항은 무시한다.



포물체 운동을 분석할 때, 이를 (1) 수평 방향의 등속 운동과 (2) 수직 방향의 자유 낙하 운동의 중첩으로 간주할 수 있다.

## 4.3.1 포물체 운동 (Projectile Motion)



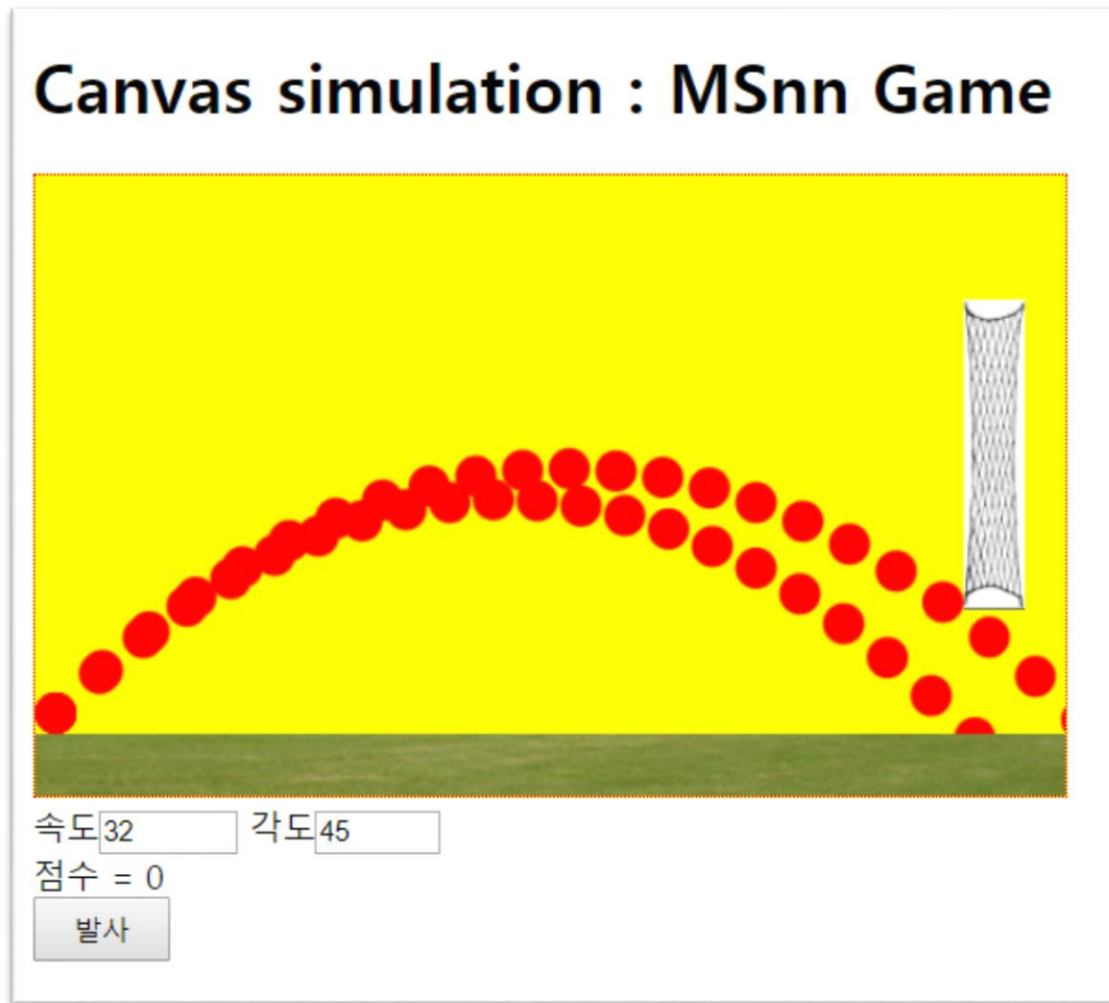
```
ballVx = velocity * Math.cos(angleR);  
ballVy = -velocity * Math.sin(angleR); // negative y-direction
```

```
ballVy = ballVy + 1.98; // y-방향 속도 계산 ( $v_y = v_y + g \cdot dt$ )
```



# 간단한 게임 제작

- 앵그리 버드  와 유사한 다음과 같은 게임을 제작



# 간단한 게임 만들기 - 1

```
<body onload="init();">
  <h1>Canvas simulation : MSnn Game</h1>
  <canvas id="myCanvas" width="500" height="300"
    style="border: 1px dotted red"> </canvas>

  <div id="control">
    속도<input id="velocity" value="30" type="number" min="0" max="100" step="1" />
    각도<input id="angle" value="45" type="number" min="0" max="90" step="1" />
    <div id="score">점수 = 0</div>
    <button onclick="fire()" style="height:24pt;width:50pt">발사</button>
  </div>
</body>
</html>
```

## Canvas simulation : MSnn Game



속도 30    각도 45  
점수 = 0  
발사

# 간단한 게임 만들기 - 2

```
<html>
<head>
  <meta charset="utf-8" />
  <title> Mobile Simulation: Game </title>
  <style type="text/css">
    canvas {
      border: 1px dotted red;      /* 캔버스에 경계선을 그려준다. */
      background-color: #fcff00;  /* 캔버스의 배경색을 지정한다. */
    }
  </style>
</head>

<script>
  var canvas = document.getElementById("myCanvas");
  var context = canvas.getContext("2d");

  /* 변수 초기화*/
  var velocity;      // 사용자가 입력한 공의 초기속도
  var angle;          // 사용자가 입력한 공의 초기각도
  var ballV;          // 공의 현재 속도 (2차원 속도)
  var ballVx;         // 공의 현재 x방향 속도
  var ballVy;         // 공의 현재 y방향 속도
  var ballX = 10;     // 공의 현재 x방향 위치
  var ballY = 260;    // 공의 현재 y방향 위치 */
  var ballRadius = 10; // 공의 반지름
  var score = 0;      // 점수 초기화
```

# 간단한 게임 만들기 - 3

```
var image = new Image();           /* 이미지 객체 생성 */
image.src = "media/lawn.png";      /* 이미지 파일 이름 설정 */
var netimage = new Image();
netimage.src = "media/net.png";
var timer;                          /* 타이머 객체 변수- control cannonball*/

/* 공을 화면에 그린다. */
function drawBall() {
    context.beginPath();
    context.arc(ballX, ballY, ballRadius, 0, 2.0 * Math.PI, true);
    context.fillStyle = "red";
    context.fill();
}

/* 배경을 화면에 그린다. */
function drawBackground() {
    context.drawImage(image, 0, 270);
    context.drawImage(netimage, 450, 60);
}

/* 전체 화면을 그리는 함수 */
function draw() {
    //context.clearRect(0, 0, 500, 300); /* 화면을 지운다. */
    drawBall();
    drawBackground();
}
```



# 간단한 게임 만들기 - 4

```
/* 초기화를 담당하는 함수 */
```

```
function init() {  
    ballX = 10;  
    ballY = 260;  
    ballRadius = 10;  
    draw();  
}
```

```
/* 사용자가 발사 버튼을 누르면 호출된다. */
```

```
function fire() {  
    init();  
    velocity = Number(document.getElementById("velocity").value);  
    angle = Number(document.getElementById("angle").value);  
    var angleR = angle * Math.PI / 180;  
  
    ballVx = velocity * Math.cos(angleR); // to the right  
    ballVy = -velocity * Math.sin(angleR); // upward  
  
    draw();  
    timer = setInterval(calculate, 100); // next position & velocity of cannonball  
    // return false;  
}
```

# 간단한 게임 만들기 - 5

```
/* 공의 현재 속도와 위치를 업데이트한다. */
function calculate() {
    ballVy = ballVy + 1.98; // y 방향 가속운동 반영 (포물선 운동 유지)

    ballX = ballX + ballVx;
    ballY = ballY + ballVy;

    /* Hit test: 공이 목표물에 맞았으면 */
    if ((ballX >= 450) && (ballX <= 480) && (ballY >= 60) && (ballY <= 210)) {
        score++;
        document.getElementById("score").innerHTML = "점수=" + score;
        clearInterval(timer);
    }
    /* 공이 경계를 벗어났으면 */
    if (ballY >= 300 || ballY < 0) {
        clearInterval(timer);
    }
    draw();
}
</script>
</head>
```

# 간단한 게임 update #1

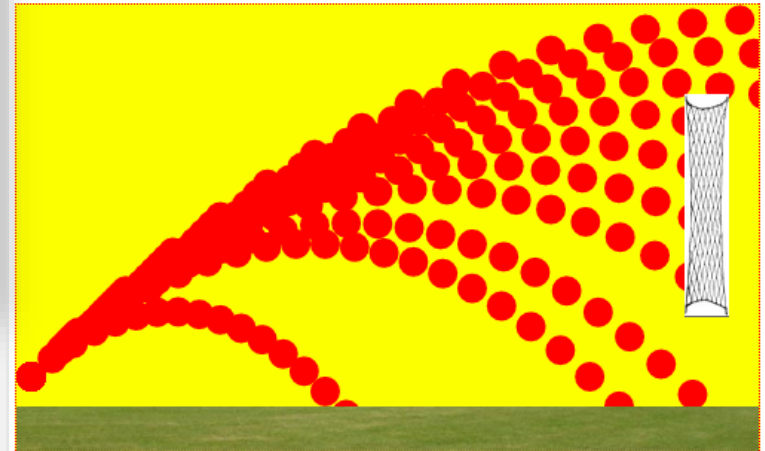
```
<button onclick="fire()" style="height:24pt;width:50pt">발사</button>  
<button onclick="clearOrbit()" style="height:24pt;width:80pt">궤도지우기</button>
```

```
/* 공의 궤도를 지우는 함수 */  
function clearOrbit() {  
    context.clearRect(0, 0, 500, 300);  
    /* 화면을 지운다. */  
    drawBall();  
    drawBackground();  
}
```

## [Mission-1]

궤도를 지우면 공이 사라지는 **Bug** 해결!

## Canvas simulation : My Game



속도 20    각도 45  
점수=4  
발사    궤도지우기

# 게임 update #2

## [Mission-2]

0. canvas에 현재 점수 표시 및 게임 상태 표시.
1. 게임 OVER : 초기 10회 발사에서 6점 이상을 얻지 못하면 게임 끝.  
게임 끝나면 점수 초기화 및 초기 화면 재구성.
2. 게임 난이도 조정 [중간 단계]  
공을 발사할 때,  
네트(netimage)의 높이가 무작위로 캔버스 내에서 변화하게 한다.  
위치가 변경된 네트를 통과하면 점수가 2점 올라가게 코드를 변경한다.
3. 스테이지(단계) 설정 [도전 과제]
  - 기본 단계: 네트의 위치가 고정  
기본 단계에서 점수가 10점을 초과하면,
  - 중간 단계: 네트의 위치와 크기가 변하는 단계  
중간 단계에서 점수가 20점을 초과하면, ...
  - 고난도 단계: 각자의 창의력을 발휘해 보시길...



# 과제05. msnn\_rpt05.zip

25

## [실습과제05] Javascript game

- [1] 캔버스와 자바스크립트만을 사용해서 게임 만들기
- [2] 게임 update #2 내용 참고하여 게임 코드 보완
- [3] script 에 주석으로 보완된 코드 설명 추가.

\*\*\*\* **MSnn\_Game.html** 파일 및 관련 파일(image 등..)을  
MSnn\_Rpt05.zip 으로 압축해서 제출하시오

가점: Javascript 프로그래밍 응용 능력. (**bonus +5**)

[제출파일] **msnn\_rpt05.zip** (10월10일 오후 6시 마감)

**html 파일과 사용된 그림을** 압축하여 이메일로 "msnn\_rpt05" 제목으로 제출

Email : chaos21c@gmail.com

# Possible Results



Canvas simulation : My Game



Canvas simulation : My Game



Canvas simulation : My Game



# Hint I



```
var score = 0; // 점수
var netWidth = 30;
var net_h = 150;
```

```
/* 배경을 화면에 그린다. */
function drawBackground() {
    context.drawImage(image, 0, 270);
    context.drawImage(netimage, 450, 60, 30, net_h);
}
```

```
/* 네트 크기를 변경한 후 전체 화면을 그리는 함수 */
function redrawNet() {
    context.clearRect(0, 0, 500, 300); /* 화면을 지운다. */
    drawBall();
    net_h = Number(document.getElementById("netHeight").value);
    drawBackground();
    init();
}
```

# Hint II



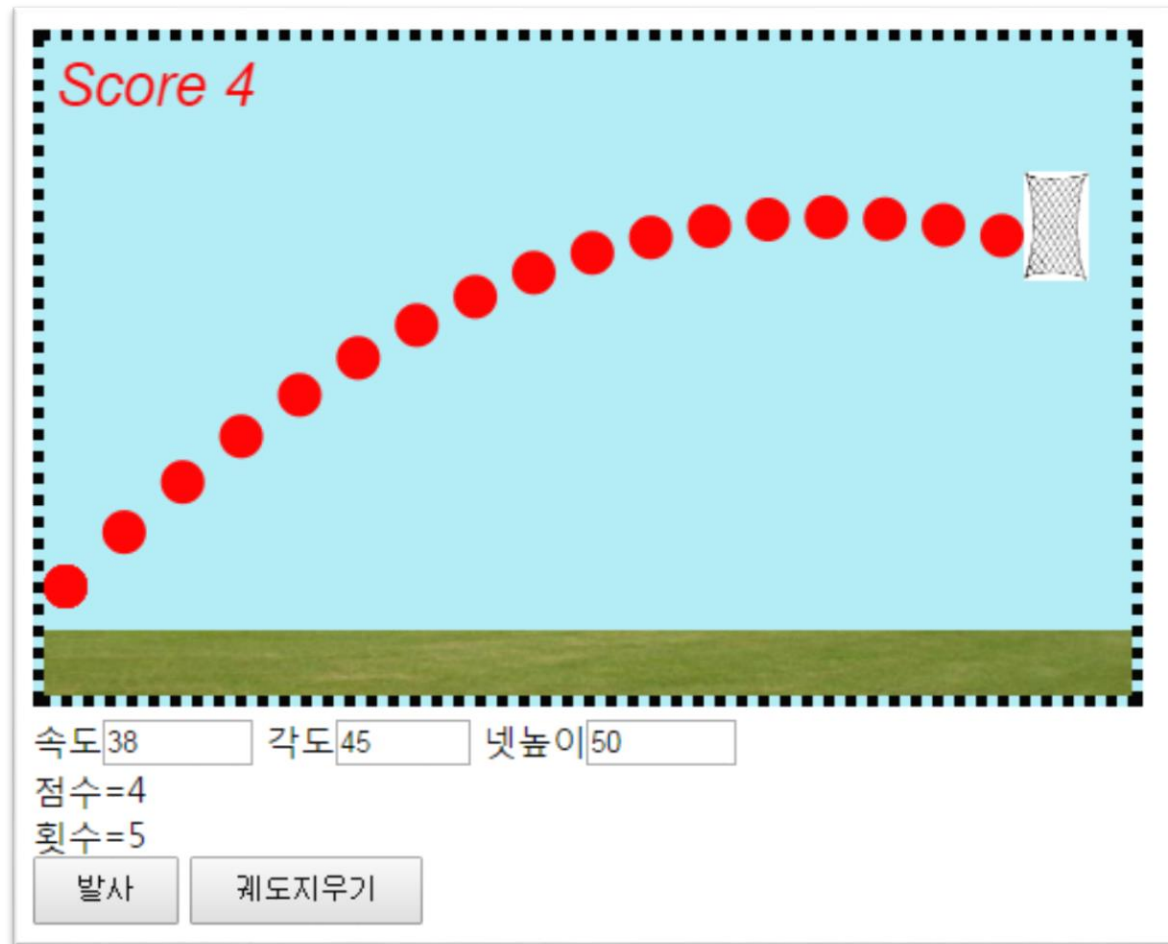
```
/* 사용자가 발사 버튼을 누르면 호출된다. */  
function fire() {  
    init();  
    velocity = Number(document.getElementById("velocity").value);  
    angle = Number(document.getElementById("angle").value);  
    var angleR = angle * Math.PI / 180;  
  
    ballVx = velocity * Math.cos(angleR);  
    // to the right  
    ballVy = -velocity * Math.sin(angleR);  
    // upward  
  
    net_h = Number(document.getElementById("netHeight").value);  
  
    draw();  
    timer = setInterval(calculate, 100);  
    return false;  
}
```

# Hint III



```
/* 공의 현재 속도와 위치를 업데이트한다. */  
function calculate() {  
    ballVy = ballVy + 1.98; // y 방향 가속운동 반영  
  
    ballX = ballX + ballVx;  
    ballY = ballY + ballVy;  
  
    /* collision test: 공이 목표물에 맞았으면 */  
    if ((ballX >= 450) && (ballX <= 480) && (ballY >= 60) && (ballY <= 60+net_h))  
        score++;  
    document.getElementById("score").innerHTML = "점수=" + score;  
    clearInterval(timer);  
}  
/* 공이 경계를 벗어났으면 */  
if (ballY >= 300 || ballY < 0) {  
    clearInterval(timer);  
}  
draw(); // 새로 결정된 위치에 볼을 그린다.  
}
```

# Possible result



# Mid exam.

[1] 10월 17일(화) 오후 2시 (장소 : E429)

[2] 필기 시험 (20점)

- 교재 10, 11장 연습 문제 (5문제, 선다형)
- Javascript BOM & Canvas coding (15문제, 4지 선다형/단답형)

\*\*\*\* 잘 준비해서 웃기를 ... ^\_^



# 교재 WEB 강의 소개

← → ↻ ⓘ webprogramming.co.kr ☆

명품 **HTML5+  
CSS3+  
Javascript** 웹 프로그래밍

Home Introduction Notice Board Support Code

## 명품 **HTML5 + CSS3 + Javascript** 웹 프로그래밍

HTML5로  
여러분의 무한한  
상상력을  
표현해 보세요!



Sir Tim Berners-Lee  
(1955.6.8 ~)

명품 웹 프로그래밍 소개  
“웹 프로그래밍을  
가장 쉽게 익힐 수 있는 책”

처음 웹 프로그래밍을 공부하는 입문자들도  
모든 주제를 직관적으로 이해하고  
빠르게 파악할 수 있습니다.

자세히보기 →



강력한 Q&A 피드백 제공  
“빠르고, 간결하고, 정확한  
저자의 직접적인 답변”

“이거 이해가 잘 안되는데.. 물어볼 사람도 없고..  
더이상 고민하지 마세요.  
명품 웹 프로그래밍 홈페이지에서는  
누구나 저자가 직접 답변해주는  
Q&A 게시판을 이용할 수 있습니다.

자세히보기 →



즉석 실행 가능한 예제 프로그램  
“백문이 불여일견, 백견이 불여일타(打)!”

코드로만 설명되어 있는 예제들,  
결과 화면이 있어도 이해가 잘 안되시죠?  
예제 소스를 바탕으로, 내맘대로 수정한  
코드를 즉석으로 웹 페이지로  
변환해주는 예제 프로그램을 통해  
모든 코드를 빠르고 쉽게  
이해할 수 있습니다.

자세히보기 →



Notice

Test

2017-01-16 15:32

Know-How

Test

2017-01-17 14:04 관리자

# 관련 WEB 강의 소개 – w3schools.com

← → ↻ 🔒 안전함 | https://www.w3schools.com

w3schools.com THE WORLD'S LARGEST WEB DEVELOPER SITE

TUTORIALS ▾ REFERENCES ▾ EXAMPLES ▾

HTML and CSS

- Learn HTML
- Learn CSS
- Learn W3.CSS
- Learn Colors
- Learn Bootstrap
- Learn Icons
- Learn Graphics
- Learn How To

JavaScript

- Learn JavaScript
- Learn W3.JS
- Learn jQuery
- Learn jQueryMobile
- Learn AppML
- Learn AngularJS
- Learn JSON
- Learn AJAX

Server Side

- Learn SQL
- Learn PHP
- Learn ASP

Web Building

- Web Templates
- Web Statistics
- Web Certificates

XML Tutorials

- Learn XML
- Learn XML AJAX
- Learn XML DOM
- Learn XML DTD
- Learn XML Schema
- Learn XSLT
- Learn XPath
- Learn XQuery

## HTML

The language for building web pages

LEARN HTML HTML REFERENCE

HTML Example:

```
<!DOCTYPE html>
<html>
<title>HTML Tutorial</title>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Try it Yourself »

## CSS

The language for styling web pages

LEARN CSS CSS REFERENCE

CSS Example:

```
body {
  background-color: lightblue;
}
h1 {
  color: white;
  text-align: center;
}
p {
  font-family: verdana;
  font-size: 20px;
}
```

Try it Yourself »

## JavaScript

The language for programming web pages

JavaScript Example:

```
<script>
function myFunction() {
  var x = document.getElementById("demo");
  x.style.fontSize = "35px";
}
```