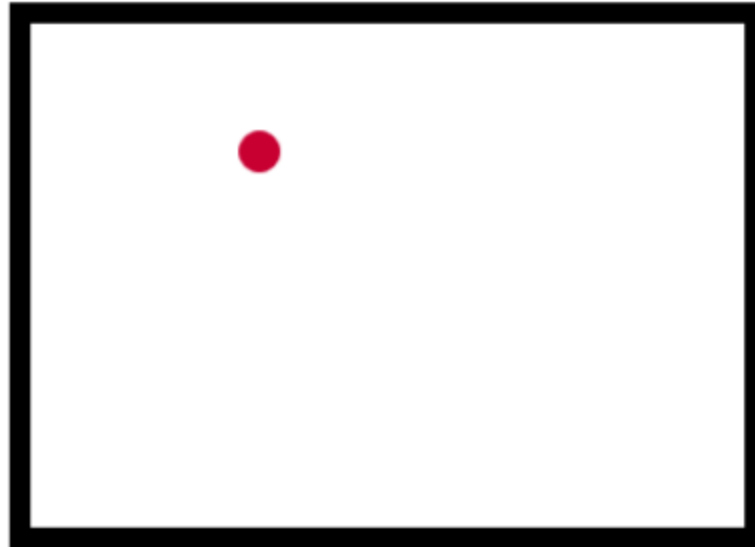


공 튀기기 게임

# 공 튀기기 게임 규칙

- 2D 상자 안에서 벽에 튕기는 공
  - 벽, 공의 모델
  - 공의 애니메이션
  - 공의 속도
  - 공이 벽에 닿으면 공의 위치 자동으로 변해 움직이기
  - 벽에 닿을 때 공의 속도 조절하기

# 공 튀기기 게임



수평 속도:   
수직 속도:

# 공 그리기

```
ctx=document.getElementById('canvas').getContext('2d');
```

```
ctx.beginPath();  
ctx.fillStyle="rgb(200,0,50)";  
ctx.arc(ballx, bally, ballrad, 0, Math.PI*2, true);  
ctx.fill();
```

# 벽과 공 그리기

```
function moveball() {  
    ctx=document.getElementById('canvas').getContext('2d');  
    ctx.clearRect(boxx,boxy,boxwidth,boxheight);  
    moveandcheck();  
    ctx.beginPath();  
    ctx.fillStyle="rgb(200,0,50)";  
    ctx.arc(ballx, bally, ballrad, 0, Math.PI*2, true);  
    ctx.fill();  
    ctx.strokeRect(boxx,boxy,boxwidth,boxheight);  
}
```

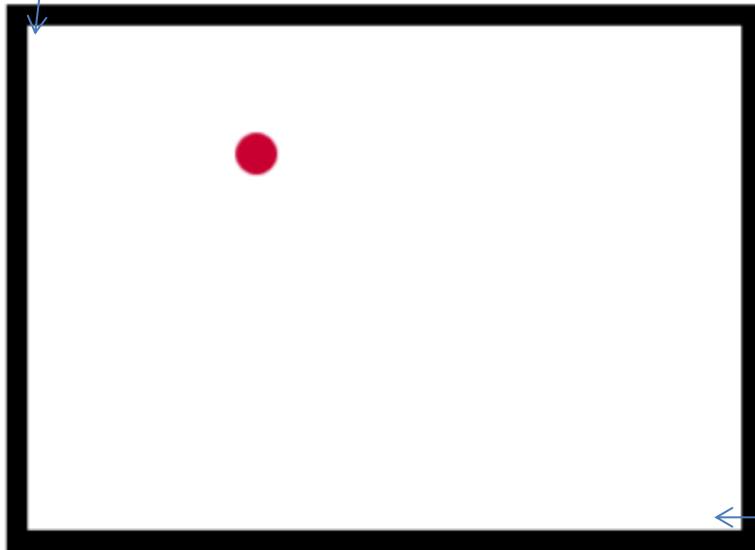


# 공 튀기기 게임

boxx,boxy

boxwidth,boxheight

inboxboundx,inboxboundy



시간에 따른 공의 애니메이션

- 속도: (ballvx, ballvy)
- 공의 초기 위치: (ballx, bally)
- 공의 다음 위치  
 $ballx = ballx + ballvx$   
 $bally = bally + ballvy$
- 공이 벽에 닿을 때 속도 계산

boxboundx,boxboundy

수평 속도: 4

수직 속도: 8

변경

# 공의 애니메이션

```
setInterval(moveball, 100);  
setInterval("moveball();", 100);  
setInterval("positionx=positionx+speedx", 100);
```

```
function init() {  
    ctx=document.getElementById('canvas').getContext('2d');  
    ctx.lineWidth=ballrad;  
    ctx.fillStyle="rgb(200,0,50)";  
    moveball();  
    setInterval(moveball, 100);  
}
```

# setInterval의 예

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>경과 시간</title>

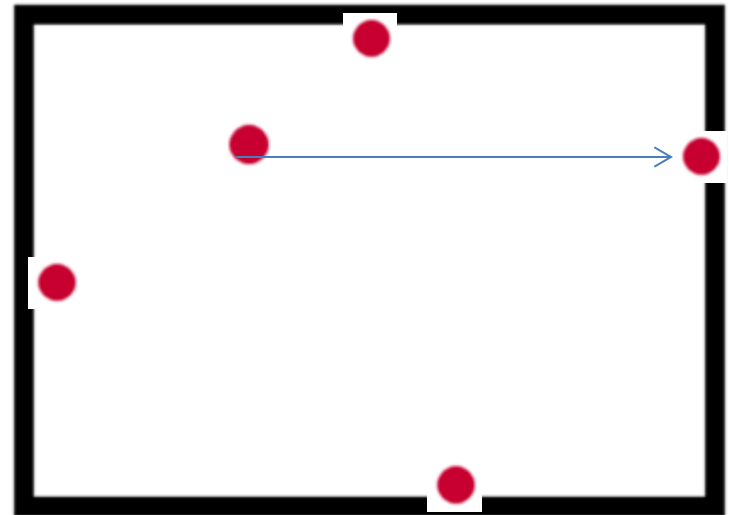
  <script>
function init() { setInterval(increase, 100); }
function increase() {
document.f.secs.value=String(1+Number(document.f.secs.value));
}
  </script>
</head>
<body onLoad="init();">
  <form name="f">
    <input type="text" name="secs" value="0" />
  </form>
</body>
</html>
```

35



# 공의 애니메이션(벽을 닿았을 때)

```
function moveandcheck() {  
    nballx = ballx+ballvx; nbally = bally+ballvy;  
    if (nballx > boxboundx) {  
        ballvx = -ballvx; nballx = boxboundx;  
    }  
    if (nballx < inboxboundx) {  
        ballvx = -ballvx; nballx = inboxboundx;  
    }  
    if (nbally > boxboundy) {  
        ballvy = -ballvy; nbally = boxboundy;  
    }  
    if (nbally < inboxboundy) {  
        ballvy = -ballvy; nbally = inboxboundy;  
    }  
    ballx = nballx; bally = nbally;  
}
```



# 공의 속도 조절

```
function change() {  
    ballvx = Number(f.hv.value);  
    ballvy = Number(f.vv.value);  
    return false;  
}
```

```
<form name="f" id="f" onSubmit="return change();">  
    수평속도: <input name="hv" id="hv" value="4"  
type="number" min="-10" max="10" />  
    <br>  
    수직속도: <input name="vv" id="vv" value="8"  
type="number" min="-10" max="10" />  
    <input type="submit" value = "변경">  
</form>
```

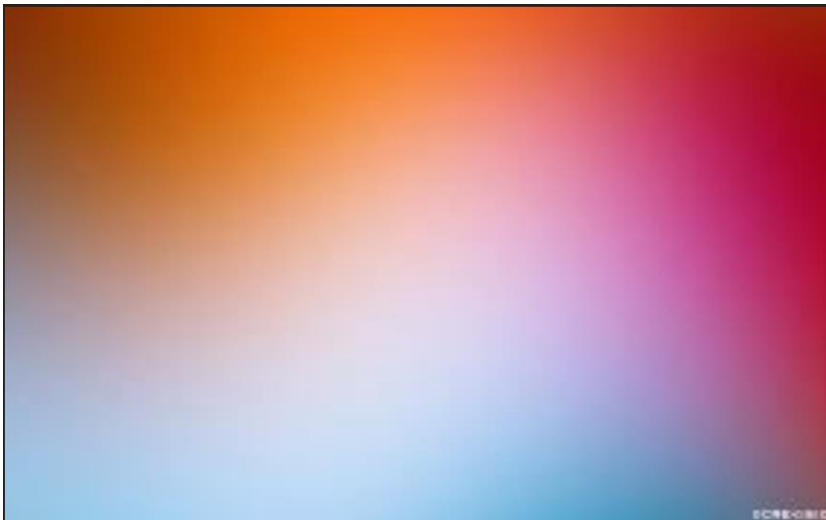
수평속도:	<input type="text" value="4"/>
수직속도:	<input type="text" value="8"/>
<input type="button" value="변경"/>	

# 공의 색깔 조정

그래디언트(gradient)

the **gradient** is a generalization of the usual concept of derivative to the functions of several variables.

If  $f(x_1, \dots, x_n)$  is a differentiable function of several variables, also called "scalar field", its **gradient** is the vector of the  $n$  partial derivatives of  $f$ . It is thus a vector-valued function also called vector field.



```
grad=context.createLinearGradient(  
    boxx, boxy, boxx+boxwidth,  
    boxy+boxheight);
```

# 공의 색깔 조정

배열

```
var family=["다니엘", "아바타", "엘리슨", "그랜트", "리엄"];
```

```
family[0], family[1], ....
```

```
var family=[  
    ["다니엘", "대학교수"],  
    ["아바타", "국회의원"],  
    ["엘리슨", "대학원생"],  
    ["그랜트", "청소년"],  
    ["리엄", "청소년"]  
];
```

# 공의 색깔 조정

배열

```
var grades=[4.0, 3.7, 3, 2.3, 3];
```

```
var sum=0;
```

```
var g;
```

```
for (g=0; g < grades.length; g++) {
```

```
    sum += grades[g];
```

```
}
```

```
var gpa;
```

```
gpa = sum/grades.length;
```

# 공의 색깔 조정

배열

```
var hue=[
    [255, 0, 0 ],
    [255, 255, 0 ],
    [0, 255, 0 ],
    [0, 255, 255 ],
    [0, 0, 255 ],
    [255, 0, 255 ],
];

grad=ctx.createLinearGradient(boxx, boxy, boxx+boxwidth, boxy+boxheight);
for(h=0; h<hue.length; h++) {
    color = 'rgb(' + hue[h][0]+' ','+hue[h][1]+' ','+hue[h][2]+' )';
    grad.addColorStop(h*1/6, color);
}
ctx.fillStyle=grad;
```

# 공의 개수 늘리기

공을 두 개 이상으로 늘려 애니메이션을 수행

# 공 튀기기 게임

