Canvas

What is Canvas?

- The HTML5 <canvas> element is used to draw graphics.
- The <canvas> element is only a container for graphics.
- Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

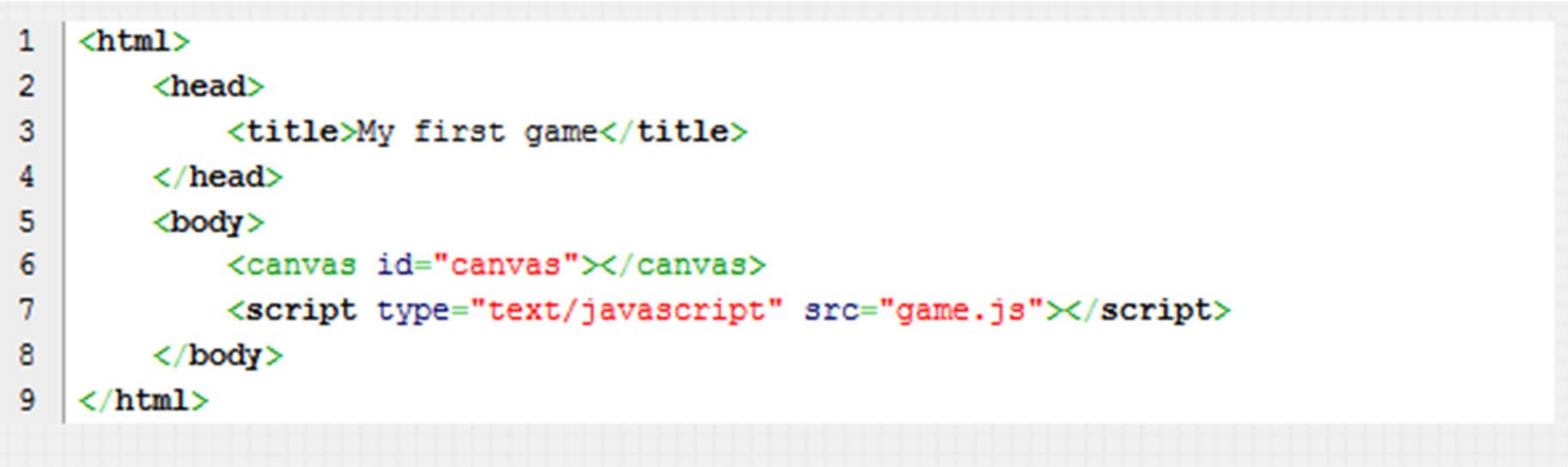
Browser Support & Limitations

Firefox	IE	Chrome	Safari	Opera	iPhone	Android	BlackBerry
3.0+	9	3.0+	3.0+	10.0+	1.0+	1.0+	OS 6.0

- you have to record the states of the elements in the canvas, and handle the hit test by yourself
- low efficient for very large size but with one a few elements in the game
- You must have a good knowledge of JavaScript to fully exploit Canvas

HTML

<canvas id="myCanvas" width="200" height="100"></canvas>



```
<html>
    <head>
        <title>My first game</title>
    </head>
    <body>
        <canvas id="canvas"></canvas>
        <script type="text/javascript" src="game.js"></script>
    </body>
</html>
```

```
<style>
    body {
      background: #eeeeee;
    }

    #canvas {
      background: #ffffff;
      cursor: crosshair;
      margin-left: 10px;
      margin-top: 10px;
      -webkit-box-shadow: 4px 4px 8px rgba(0,0,0,0.5);
      -moz-box-shadow: 4px 4px 8px rgba(0,0,0,0.5);
      box-shadow: 4px 4px 8px rgba(0,0,0,0.5);
    }

   </style>
```

HTML

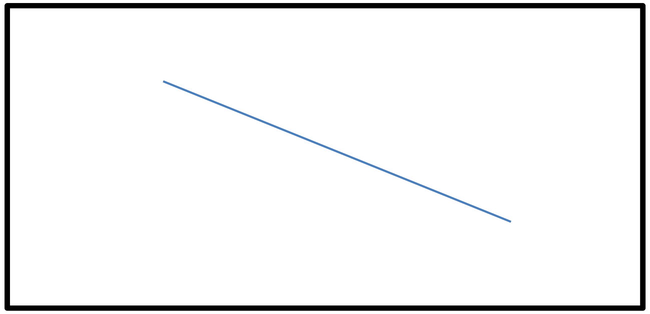
```
<script>
            var c = document.getElementById("myCanvas");
            var ctx = c.getContext("2d");
            ctx.fillStyle = "#FF0000";
            ctx.fillRect(0,0,150,75);
</script>
```

- The fillStyle property can be a CSS color, a gradient, or a pattern.
- The fillRect(*x,y,width,height*) method draws a rectangle.

Canvas - line

- To draw straight lines on a canvas, we will use the following two methods.
- To actually draw the line, we must use one of the "ink" methods, like stroke().

✓ moveTo(*x,y*)
✓ lineTo(*x,y*)

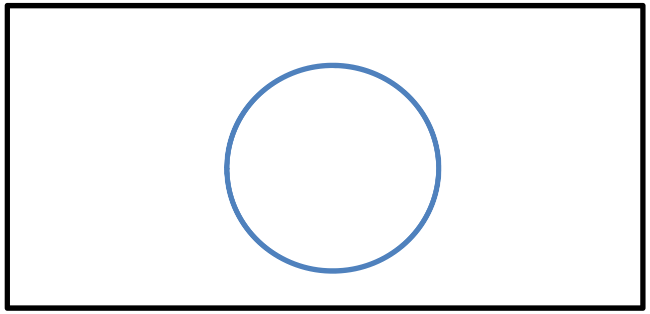


```
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.moveTo(0,0);
ctx.lineTo(200,100);
ctx.stroke();
```

Canvas - Circle

- To draw a circle on a canvas, we will use the following method:
- To actually draw the circle, we must use one of the "ink" methods, like stroke() or fill().

  - ✓ arc(x,y,r,start,stop)

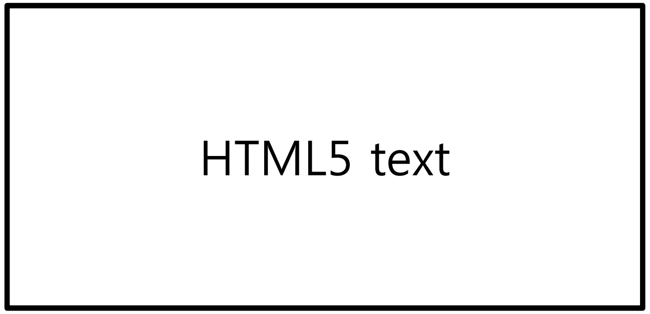


```
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.beginPath();
ctx.arc(95,50,40,0,2*Math.PI);
ctx.stroke();
```

Canvas - Text

- To draw text on a canvas, the most important property and methods are:

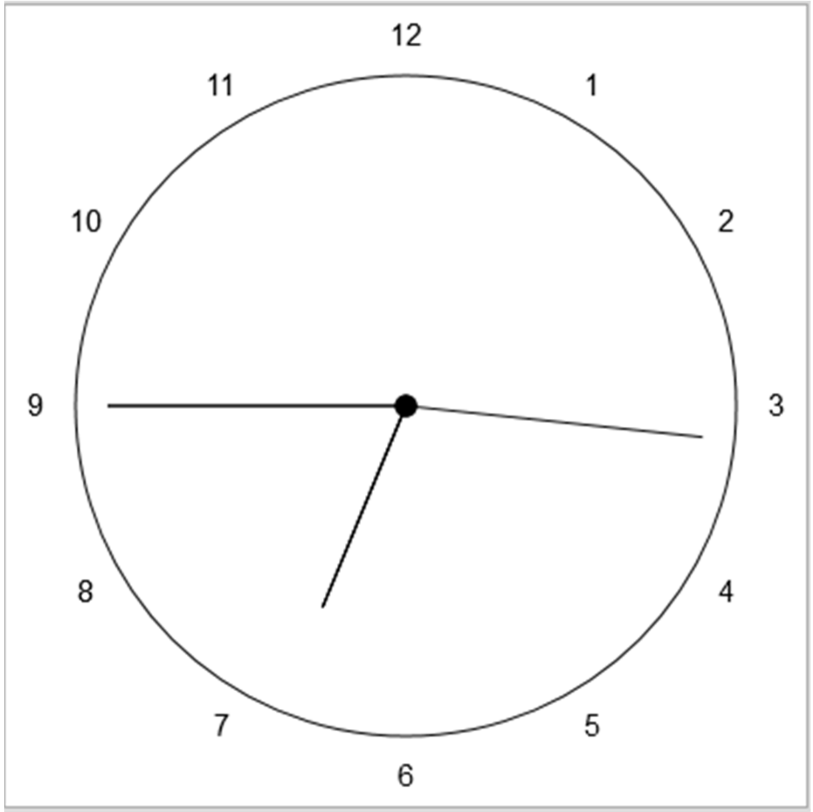
  - ✓ Font
  - ✓ fillText(text,x,y)
  - ✓ StrokeText(text,x,y)



HTML5 text

```
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.font = "30px Arial";
ctx.fillText(“HTML5 text",10,50);
```

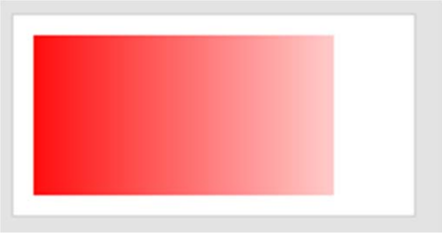
```
ctx.strokeText(" HTML5 text ",10,50);
```

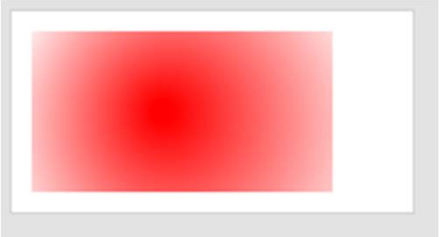

12
1
2
3
4
5
6
7
8
9
10
11

Canvas - Gradients

- Gradients can be used to fill rectangles, circles, lines, text, etc. Shapes on the canvas are not limited to solid colors.
- There are two different types of gradients:

  ✓ createLinearGradient(*x,y,x1,y1*)
  ✓ createRadialGradient(*x,y,r,x1,y1,r1*)





```
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");

// Create gradient
var grd = ctx.createLinearGradient(0,0,200,0);
grd.addColorStop(0,"red");
grd.addColorStop(1,"white");

// Fill with gradient
ctx.fillStyle = grd;
ctx.fillRect(10,10,150,80);
```

Canvas - Images

- To draw an image on a canvas, we will use the following method:
  - ✓ drawImage(*image,x,y*)

```
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
var img = document.getElementById("scream");
ctx.drawImage(img,10,10);
```

http://www.w3schools.com/tags/ref_canvas.asp

Sprite test

- sprite will be a red square
- store sprite's information like a struct

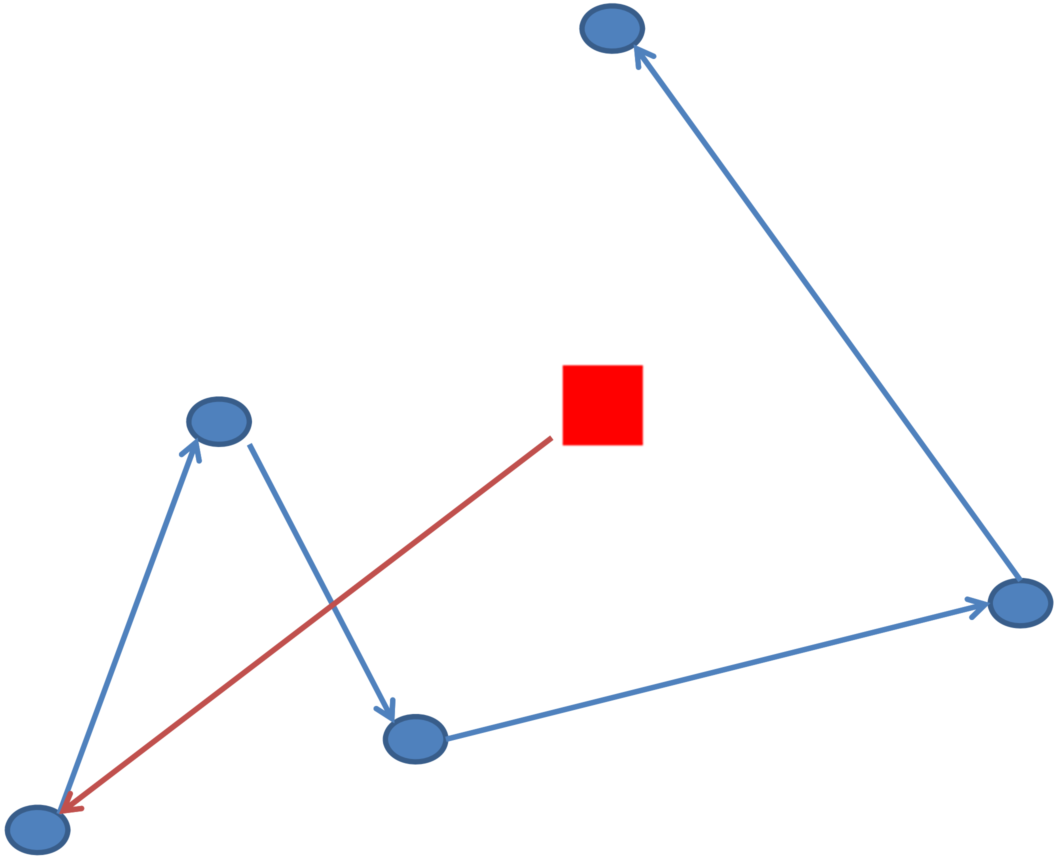
```
var mySprite = {
   x: 200,
   y: 200,
   width: 50,
   height: 50,
   speed: 200,
   color: '#c00'
};
```

- **mySprite** will start at coordinates (200, 200), have a width and height of 50, a speed of 200 pixels/second, and will be red (#c00).

Sprite test

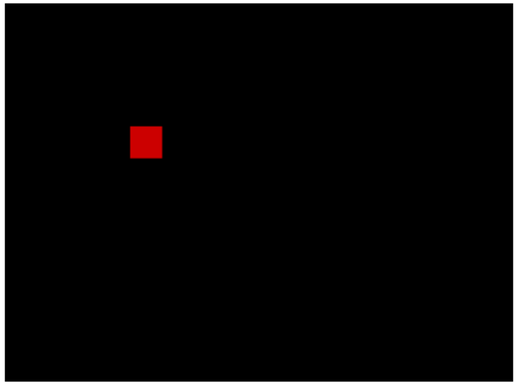
- **Update**: this function is going to be used to check if arrow keys are being pressed, and update our sprite’s x and y coordinates appropriately.
- **Render**: This function is where our drawing code will go.
- **Run**: This function will call our update and render functions and calculate the “mod” parameter of our update function. (explained in a sec)

```
function update(mod) {   }
function render() {   }
function run()
{
    update((Date.now() - time) / 1000);     render();     time = Date.now();
 }

var time = Date.now(); setInterval(run, 10);
```



Sprite test

```
function render() {
        ctx.fillStyle = '#000';
        ctx.fillRect(0, 0, canvas.width, canvas.height);
        ctx.fillStyle = mySprite.color;
        ctx.fillRect(mySprite.x, mySprite.y, mySprite.width, mySprite.height);
}

var keysDown = {};
window.addEventListener('keydown', function(e) {
        keysDown[e.keyCode] = true;
});
window.addEventListener('keyup', function(e) {
        delete keysDown[e.keyCode];
});
```



Sprite test

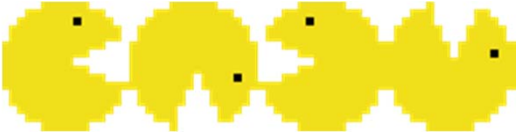
```
function update(mod) {
        if (37 in keysDown) {
          //left           mySprite.x -= mySprite.speed * mod;
        }
        if (38 in keysDown) {
          //up            mySprite.y -= mySprite.speed * mod;
        }
        if (39 in keysDown) {
         //right           mySprite.x += mySprite.speed * mod;
        }
        if (40 in keysDown) {
        //down           mySprite.y += mySprite.speed * mod;
        }
}
```

Sprite test

```
var item = {
    x: Math.random() * canvas.width,
    y: Math.random() * canvas.height,
    width: 10,
    height: 10,
    color: '#fff'
};

if (     mySprite.x < item.x + item.width
   && mySprite.x + mySprite.width > item.x
   &&     mySprite.y < item.y + item.height
   &&     mySprite.y + mySprite.height > item.y )
{
    item.x = Math.random() * canvas.width;
    item.y = Math.random() * canvas.height;
}

ctx.fillStyle = item.color;
ctx.fillRect(item.x, item.y, item.width, item.height);
```

Sprite test

- Chage code in mySprite like this:
  - ✓ Delete color
  - ✓ Add state



```
var pacmanTiles = {
   loaded: false,
   image: new Image(),
   tileWidth: 64,
   tileHeight: 64
};

pacmanTiles.image.src = 'pacman.png';
pacmanTiles.image.onload = function() {    pacmanTiles.loaded = true; }
```

Sprite test

- Use drawImage() for drawing pacman

```
if (pacmanTiles.loaded) {
        ctx.drawImage(
            pacmanTiles.image,
            mySprite.state * pacmanTiles.tileWidth,
            0,
            mySprite.width,
            mySprite.height,
            mySprite.x,
            mySprite.y,
            mySprite.width,
            mySprite.height
        );
}
```

Sprite test

- For counting points add a variable called itemCounter
  - ✓ var itemCounter

```
ctx.font = '12pt Arial';
ctx.fillStyle = '#fff';
ctx.textBaseline = 'top';
ctx.fillText(itemCounter, 10, 10);
```