Javascript Notes – Canvas and Keyboard

E-Applications Spring 2015

M. Lampis

Basic 2d drawing in javascript

- In order to draw simple graphics we can use a <canvas> object

```
<canvas id="canvas" width='100' height='100'>
<p>Sorry: Browser does not support Graphics Canvas</p></canvas>
```

In javascript

- Now the canvas can be accessed from within javascript

```
var canvas   = document.getElementById("canvas");
  if(!canvas.getContext){return;}
  var ctx=canvas.getContext("2d");
```

- The "context" represents the area in which we will be drawing. Two choices: 2d or webgl (for 3d graphics)

Drawing lines

- Example:

```
ctx.beginPath();
ctx.moveTo(25,25); //move pen to 25,25
ctx.lineTo(105,25); //draw line to 105,25
ctx.closePath();
ctx.stroke();
```

Cleaning up

- Example

  ```
  ctx.clearRect(50,50,150,150); //clear this area
  ```

Playing with time

- Animations can be achieved by redrawing at intervals
- Recall:

```
ref = setInterval(func, delay);
clearInterval(ref);
```

Rectangles

- Example

```
ctx.rect( 20,20,180,60);
ctx.stroke();
```

- Filled

```
ctx.fillRect(22,22,176,56);
ctx.stroke();
```

- `ctx.fillStyle = "blue"; //also "rgb(0,0,200)"`

Arcs

Arcs (and cycles) can be created with the arc(x,y,r,angle1,angle2) method

- x,y -> center, r-> radius,
- angle1, angle2 -> allow to draw "pie slice"
- Set to 0, 2π, for whole cycle

```
ctx.beginPath();
ctx.arc(x,y,r,0,2*Math.PI);
ctx.fill();
```

Catching keyboard events

- Three relevant events: onkeydown, onkeypress, onkeyup
- Can be associated with window object (so that all key strokes are caught)
- onkeydown: key was pressed (even special keys)
- onkeypress: only for character keys

Reading the keys

```
window.onkeydown = function(e) {
	console.log("You pressed a key!" );
	console.log("e.keyCode = "+e.keyCode);
}
```

Catching mouse events

- Relevant events: onclick, onmousedown, onmouseup, onmousemove
- Can read mouse coordinates too!

```
window.onmousedown = function(e) {
        console.log("Mouse button down!" );
        console.log("x = "+e.clientX + " y= 
"+e.clientY);
}
```

Mouse coordinates and Canvas

- The code of the previous slide gives the coordinates relative to the whole page
- Often, when a user clicks inside the canvas, we want to get coordinates relative to the top-left corner of the canvas
- We can get the coordinates of the canvas itself with `canvas.getBoundingClientRect();`
- Returns an object with properties left,top.