**Some Tips From Keith about setTimeout in JavaScript**

Some of you have reported problems with making setTimeout work at “slowing down” your graphics. Calling setTimeout correctly turns out to be a little tricky.

Here is a stripped down version of the example program given in the folder HW2 in DOCS & ASSIGNMENTS:

<!DOCTYPE html>

<html>

<body>

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">

Your browser does not support the HTML5 canvas tag.

</canvas>

<script>

var c = document.getElementById("myCanvas");

var ctx = c.getContext("2d");

function spotMaker() {

var x = Math.random() \* 190;

var y = Math.random() \* 90;

ctx.fillStyle = "#00FF00";

ctx.fillRect(x,y,10,10);

}

for (var i = 0; i < 10; i++){

setTimeout("spotMaker()",(i\*1000));

}

</script>

</body>

</html>

The code above works as expected: 10 green squares slowly appear on the screen inside the rectangular box. So far, so good.

On the next page I’ll give the same code EXCEPT for a single change: I remove the quotes from around the spotMaker call inside setTimeout. The code with that tiny change (see the next page) does NOT work. Instead, the code on the next page prints all 10 green squares all at once:

<!DOCTYPE html>

<html>

<body>

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">

Your browser does not support the HTML5 canvas tag.

</canvas>

<script>

var c = document.getElementById("myCanvas");

var ctx = c.getContext("2d");

function spotMaker() {

var x = Math.random() \* 190;

var y = Math.random() \* 90;

ctx.fillStyle = "#00FF00";

ctx.fillRect(x,y,10,10);

}

for (var i = 0; i < 10; i++){

setTimeout(spotMaker(),(i\*1000));

}

</script>

</body>

</html>

Remember, the code above does NOT work as expected. Instead, all the green squares appear all at once. If you have that problem, make sure you are putting quotes around your function call.

Instead of using the quotation marks, perhaps you prefer making an anonymous function. That can work too. Please see the example on the next page.

<!DOCTYPE html>

<html>

<body>

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">

Your browser does not support the HTML5 canvas tag.

</canvas>

<script>

var c = document.getElementById("myCanvas");

var ctx = c.getContext("2d");

function spotMaker() {

x = Math.random() \* 190;

y = Math.random() \* 90;

ctx.fillStyle = "#00FF00";

ctx.fillRect(x,y,10,10);

}

for (var i = 0; i < 10; i++){

setTimeout(function(){spotMaker()},(i\*1000));

}

</script>

</body>

</html>

The code above DOES work, painting the squares one after the other, in slow motion.

NOTE WELL: Passing parameters to a function inside a call to setTimeout can be problematic. Even for my tiny example, I had some trouble making a version with parameters work. I experimented and got the code on the next page to work with a single parameter. Notice that the code on the next page combines an anonymous function inside the setTimeout call (like the program above) AND it includes some parentheses (like the first page of this handout). But I had to put the quotes around the parameter, not around the call to spotMaker. Check the next page to see what I mean.

<!DOCTYPE html>

<html>

<body>

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">

Your browser does not support the HTML5 canvas tag.

</canvas>

<script>

var c = document.getElementById("myCanvas");

var ctx = c.getContext("2d");

var kColor= "#00FF00";

function spotMaker(myColor) {

x = Math.random() \* 190;

y = Math.random() \* 90;

ctx.fillStyle = "#00FF00";

ctx.fillRect(x,y,10,10);

}

for (var i = 0; i < 10; i++){

setTimeout(function(){spotMaker("kcolor")},(i\*1000));

}

</script>

</body>

</html>

This code works, but using spotMaker(kcolor) inside the call to setTimeout, without the quotes around kcolor, does NOT work.. And it doesn’t work AT ALL. Nothing gets painted at all. Add the quotes around kcolor, and you are back in business.

If it is relatively convenient for you, I suggest that you try to make the function you call inside setTimeout to have no parameters. It’s just easier. Otherwise, try quoting all the parameters to the function when you put it inside the call to setTimeout. On the next page, I extended the example above so it has three parameters instead of one parameter.

<!DOCTYPE html>

<html>

<body>

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">

Your browser does not support the HTML5 canvas tag.

</canvas>

<script>

var c = document.getElementById("myCanvas");

var ctx = c.getContext("2d");

var kColor= "#00FF00";

var myX;

var myY;

function spotMaker(myColor, x, y) {

x = Math.random() \* 190;

y = Math.random() \* 90;

ctx.fillStyle = "#00FF00";

ctx.fillRect(x,y,10,10);

}

for (var i = 0; i < 10; i++){

setTimeout(function(){spotMaker("kcolor,myX,myY")},(i\*1000));

}

</script>

</body>

</html>

Once again, this works. The reason you need the quotes around your list of parameters here is subtle, and has to do with how expressions are interpreted by setTimeout and Javascript. Don’t worry about that now; just get something to work.

Another hint: notice that the miliseconds parameter to setTimeout needs to be larger each time setTimeout is called. The second call has to wait longer than the first, the third has to wait longer than the second, and so on.

I’ll put this handout into the HW2 folder in DOCS & ASSIGNMENTS on our Blackboard class site.

Keith