Path drawing: lines

INTRODUCTION

We have been drawing rectangles so far.

Now let's go a bit further by introducing the notion of "path drawing". This approach uses the `ctx.moveTo(x, y)` method of the context, in conjunction with other drawing methods that end in "To", such as `ctx.lineTo(x, y)`.

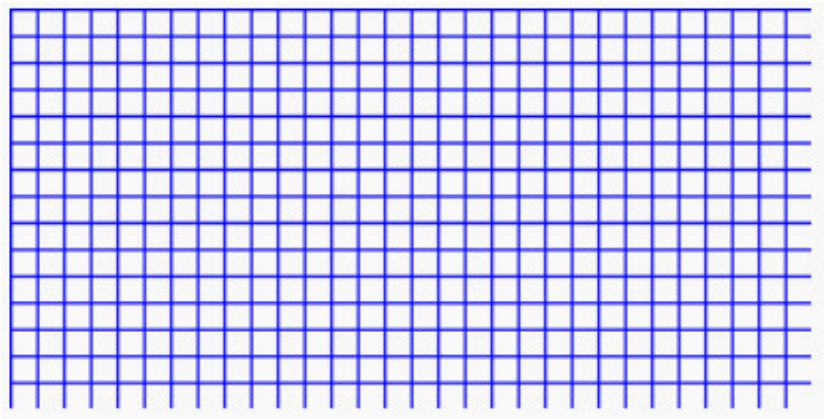
This makes it easier to draw multiple connected lines. Consecutive calls to `ctx.lineTo(x, y)` will store in the path/buffer a set of connected lines that we will draw altogether by a single call to `ctx.stroke()` or `ctx.fill()`.

Here are the different steps:

1. Put the "pencil" somewhere with a call to `ctx.moveTo(x1, y1)`. This will be the origin of the first line.
2. Call the `ctx.lineTo(x2, y2)` method to draw a line from the previous position (previous step) to the position passed as parameters to the `lineTo(...)` method. This position will serve as the origin for the next line to be drawn.
3. Call `lineTo(x3, y3)` again to draw a line that goes from (x2, y2) to (x3, y3). This line will start at the end of the previous one.
4. Repeat step 3 to draw more connected lines.
5. Call the `ctx.stroke()` or the `ctx.fill()` methods to draw the path defined by the different lines.

Note the call to `ctx.stroke()` or `ctx.fill()` will use the current values of the `strokeStyle` or `fillStyle` properties. It is possible to call `ctx.moveTo(x, y)` in the middle of steps 1 through 5 in order to move the pen somewhere else without connecting to the last drawn line.

Here is an example that draws a grid: http://jsbin.com/zugale/1/edit

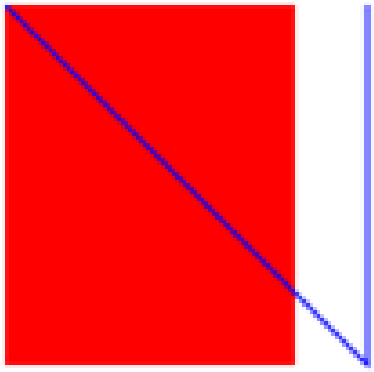


```
varcanvas=document.getElementById('myCanvas');
var ctx=canvas.getContext('2d');
// Vertical lines
for (var x = 0.5; x < 500; x += 10) {
    ctx.moveTo(x, 0);
    ctx.lineTo(x, 375);
}
// Horizontal lines
for (var y = 0.5; y < 375; y += 10) {
    ctx.moveTo(0, y);
    ctx.lineTo(500, y);
}

// Draw in blue
ctx.strokeStyle = "#0000FF";

// Until the execution of the next line, nothing has been drawn!
ctx.stroke();
```

In this example, the entire grid is drawn during the execution of the last line of code, with the single call to `ctx.stroke()`.

ANOTHER EXAMPLE MIXING FILLED AND WIREFRAME SHAPES (AND IMMEDIATE AND PATH MODES)

Try this interactive example here: http://jsbin.com/zetupi/4/edit

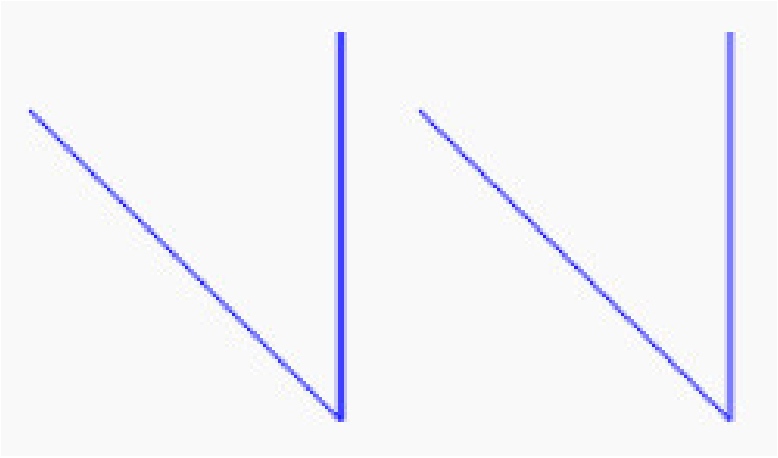


```
varcanvas=document.getElementById('myCanvas');
var ctx=canvas.getContext('2d');
// a filled rectangle in immediate mode
ctx.fillStyle='#FF0000';
ctx.fillRect(0,0,80,100);
// two consecutive lines in path mode
ctx.moveTo(0,0);
ctx.lineTo(100, 100);
ctx.lineTo(100,0);
// draws only the two lines in wireframe mode
ctx.strokeStyle = "#0000FF";
ctx.stroke();
```

This example shows that filled and wireframe shapes should be drawn differently (here a filled rectangle is drawn using a call to the `fillRect(...)` method while a wireframe set of connected lines is drawn using the `stroke()` method of the context).

DRAWING A SINGLE PATH MADE WITH DISCONNECTED LINES / PARTS

Try this interactive example here: http://jsbin.com/lefoze/2/edit



```
varcanvas=document.getElementById('myCanvas');
var ctx=canvas.getContext('2d');
// first part of the path
ctx.moveTo(20,20);
ctx.lineTo(100, 100);
ctx.lineTo(100,0);
// second part of the path, moveTo(...) is used to "jump" to another place
ctx.moveTo(120,20);
ctx.lineTo(200, 100);
ctx.lineTo(200,0);
// indicate stroke color + draw the path
ctx.strokeStyle = "#0000FF";
ctx.stroke();
```

In this example, we simply called the `moveTo()` method between each part of the path. And we called `stroke()` only once to draw the whole path.

KNOWLEDGE CHECK 3.4.3 (NOT GRADED)

How would you draw a line from (10, 10) to (100, 100)?

- ctx.line(10, 10, 100, 100); ctx.stroke();

- ctx.LineTo(10, 10, 100, 100); ctx.stroke();
- ctx.moveTo(10, 10); ctx.lineTo(100, 100); ctx. stroke();