NM2207
Week 5
Assignment

Due: Tuesday 15th September 11.59 pm

Overview of what we'll do today:
Using only the click event of the Raphael paper element, we will make a line change its end point, and create a path. Note that we are using only the click event.

Please watch the Session05.assignment video listed on the LumiNUS tab (4th video in the "Assignment videos" playlist), which describes what each of the final outcomes should look like. Feel free to style your lines differently!

Further, note what happens when we use just the click event. What would happen if we decided to also use the mouse up and mouse down events? In this case, we would be able to draw a path free hand, instead of simply adding line segments. We have practiced mouse up and mouse down events in the codealong and also in the Week 5 tutorial.

New text:
As you watch all 4 of this week's assignment videos, do pay attention to the one on adding audio because we'll be using that in next week's tutorial. Next week, we will watch a short video about "How to pass NM2207" which will focus on how your work is evaluated, and how to plan for a successful score. Next week will have a lighter challenge and give you more time to ask your questions and doubts.

Part 1
tickTock

The overall goal of this task is to change the end point of a line segment (using attr, obviously) each time the user clicks on paper. Specifically:

1. Create a variable for a Raphael rectangle to fill the paper to use as a background; set its fill attribute.
    a. Something like this should work:
    var prect = paper.rect(0,0, dimX, dimY);
2. First, we identify two end points, and we add text to label them "Tick" and "Tock". Refer to the text element described here.
3. Next, we will create a needle which will start at the center of the paper (dimX/2 and dimY/2) and end a few pixels below Tick.
    a. How to create a needle? Check here.
    b. The main.js file actually uses dimX and dimY to store the width and height.
4. Define a function in your main.js file, and name it "drawNeedle". Define it to take four arguments, and use the arguments in the function to construct a string called pathString, which we will use to set the path of our needle.
    a. Test the pathString by printing it to the console.
    b. How to redraw the needle using the pathSTring? Use the attr method of a path object, described here.
        E.g. needle.attr({"path":pathString});

5. Now you want to add an event listener to the paper. We want that whenever the paper is clicked,
    a. drawNeedle is called with new coordinates. If the needle's endpoint (L) was at Tick before (i.e. the coordinates correspond to Tick), then coordinates for Tock should be passed (so that the needle is redrawn to Tock).
    b. Similarly, if the needle's endpoint (L) was at Tock before, it should now change to Tick.
    c. Hint: you will need a way to track whether the needle's endpoint is currently at Tick or not

Part 2

msDraw

The Overall goal of this task is to draw a line segment from a previous point to wherever the user clicks. Specifically:

1. Create a filled rectangle to be your background, like last time. Create a variable for a Raphael rectangle to fill the paper to use as a background; set its fill attribute.
2. We need to remember a path. We will update that and redraw it each time a user clicks. Refer to the path element described here.
    a. Hint: you will need to define your path as a global variable.
    b. Create a variable called pathString that we will use to remember the path. this will be updated whenever the user clicks. we will use this variable to draw our path with paper.path.

3. We will create a drawing method (function) named 'connect' that takes the new x and y location where the user clicked as arguments, and draws a line from the last point to the new point. Don't forget to call the method (where do you want to do that?)
    a. How to draw a line from the last point to the new point? We can simply append the coordinates of the click to the end of pathString. Refer to how this string has to be structured, described here. Look at the video and see what is printed in the console window.

3. Once again, create a "clear" button as we did in the tutorial, so that you can clear your drawing and start again.