

NM2207

Session 05

Challenges

Before attempting the challenges, you are expected to have watched and coded along with the Lecture videos. A tutorial is meant to practice the skills presented in the video lecture, and show you more applications of it. Tutors will explain the challenges and answer the questions you may have.

The challenges are due to be completed at the end of class each week for full credit which is also attendance. Submitting by midnight of the same day (ie Thursday night) accounts for half the credit.

Overview of what we will do today:

- Practice creating a circle and dragging it everywhere with mouse up, mouse down, and mouse move events
- Practice using the slider to set circle radii
- Learn how to clear shapes and put shapes onto paper

Part 1

Summary of learnings

- Revising creating basic shapes and adding event listeners

Warm up (40 minutes)

The Overall Goal of this task set is to create a background for the paper, and then a circle that can be dragged around the canvas. The specific tasks for you to accomplish starting from the template code are:

- 1.1 Create a variable for a Raphael rectangle to fill the paper to use as a background; set its fill attribute.
- 1.2 Create a variable for a Raphael circle at the center of the canvas. Set some attributes.
- 1.3 Use 'mousedown', 'mousemove', and 'mouseup' events so the user can drag the circle around the canvas.

Hint: You'll need variables to remember the state of the mouse.

Part 2 (1h 30 minutes)

Summary of learnings

- Use slider value as the circle radius

The Overall Goal of this task set is to Click canvas to draw circles with an attribute set by a slider, and to add a way to clear the canvas of the artwork. Specifically:

- 2.1 Create a variable called circleRadius, that we'll use to size circles we draw. Initialize it to a value
- 2.2 Draw circles on the paper where mouse clicks are made. (The background rectangle will come in handy here.) Use circleRadius to set the size of the circle when you create them in your callback function.
- 2.3 Add a slider to the "<aside>" element in the .html file,
- 2.4 Set slider range between [0,1] (so it can be used for anything with scaling)
- 2.5 Create a label (using the <label> element) for the slider, styled to look reasonable.

(Suggested 10 minute break)

2.6 Use the slider to change the circleRadius value. You are already using the variable to set the size of new circles, so just changing the value of the variable should affect how future circles are drawn.

- (Since the slider ranges from [0,1], you'll have to map the slider values to the circle

sizes you want. By “map” I mean perform a numerical operation to transform the range of numbers you get from the slider into the range you want your circles to be.)

2.7 Add a button to the “<aside>” element in the .html file, this will be your “CLEAR” button

2.8 Use the clear button to remove elements from the paper (calling Raphael’s `paper.clear()`)

2.9. Restore the background and circle elements to the paper after clearing everything, using the `paper.put(element)` method I provided for you at the top of the template.

(note: you still have the elements available if you named them using a `var` or `let` statement so you can pass them to the `paper.put` method.)

Bonus

3.1 Rip, mash and modify your HSL sliders from Session04.assignment homework to add circles with different fill colors depending on the slider values.

3.2 Create a paint app with buttons to allow you to create other kinds of shapes on click, like squares (`paper.rect` with the same width and height) and lines (`paper.path`). You can still use the slider value for the width of the squares or the thickness of the lines. Your tutor can show you an example of this coded up.

Hint: You will have to track which shape is selected, so that you know what kind of shape to create. And if multiple shape types are selected, you could generate two shapes together!