Interactive Device and Design

Lab 6

Hao Zheng (hz466)

In collaboration with Sarah Le Cam

Part A. Node Canvas

1. **Where in this code does the drawing occur?**

   Drawing occurs in the client.js when the websocket to the server receives the 'new-pos' message.

2. **What are the inputs to the drawing function?**

   The drawing function needs all these parameters in order to work.

   ctx.lineCap = 'round';

   ctx.lineJoin = 'round';

   ctx.fillStyle = ctx.strokeStyle = COLOUR;

   ctx.lineWidth = radius; which is set to 3 at the beginning of the code.

   previousPosition

   newPosition

   However, the only input it actually get from outside is the newPosition.

3. **How can the screen be cleared?**

   The screen can be cleared by ctx.clear() when the socket to the webserver receives the 'reset' message. The screen can also be cleared when the canvas is resized.

Part B. Etch a Sketch

1. **Describe which sensors you are using as inputs and why.**

   We are using an accelerometer because it supplies three different values, which can be used as our inputs. And these three values have pairwise independency.

2. **In what format does the server expect the X & Y values from the Arduino?**

   The server expects the X & Y values as comma delimited string from Arduino in the provided code. In our modified code, we add Z value and mode state in addition to X & Y values in the comma delimited string.

3. **What else does the Arduino need to do? The sensorCode folder has an almost working example.**

   The Arduino needs to read the values from the accelerometer and map them to integers between 0 to 1023. It also needs to register the state of the switch and converts it into mode state. Then it has to send the X, Y, Z values along with the mode state over the serial port to Raspberry Pi. In addition, the Arduino has to detect if the button is pressed and send the “rst” message over the serial port.

4. **What range of output X & Y values do you need? Is it better to map the values from the inputs to the outputs in the Arduino code, the Node code or Client code ? Why?**

   The range of output X & Y values we need depends on the size of the canvas, which can be obtained using ctx.width and ctx.height in the Client code. Also, because each canvas on different client can have different sizes, so we have to do the final mapping in the client code. But before that, we also perform another mapping in the Arduino code to map the raw readings of the sensor, in our case accelerometer, to integers between 0 to 1023. We did this with the idea that the Arduino should always send values within the same range so that no matter what sensor is used, the same server code can be used without modification.

5. **How often do you need to be sending data from the Arduino?**

   We are sending data from the Arduino to the server five times a second. On one hand, Since the accelerometer is very sensitive, and the readings tend to be unstable, we don’t want to update the values too often in order to avoid zigzag drawing. On the other hand, users need real-time feedback of the control, so the interval cannot be too large either.

6. **Include a copy of the Arduino code, and a copy of your Server and client side code.**

   Arduino code: https://github.com/hz2287/etch-a-sketch/blob/master/sensorCode/sensorCode.ino

   Server code: https://github.com/hz2287/etch-a-sketch/blob/master/server.js

   Client code: https://github.com/hz2287/etch-a-sketch/blob/master/public/client.js

Part C. Make it yours

Add a feature. (If you're working in a group, add as many features as you have people in your group.)

Feature 1: Canvases on all clients can be cleared and color reset to black by pressing the physical push button on our board
Feature 2: Switching the physical switch on our board can switch between draw mode and color mode. In the draw mode, the X and Y values of the accelerometer set coordinates. In the color mode, the X, Y, and Z values are used to set the RGB values, the X, Y, and Z values are not independent from one another, though.

1. **Upload video of your Etch-a-Sketch in action, being used by someone else in the class!**

https://youtu.be/RLtGhUFCj1E