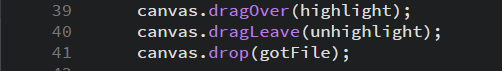
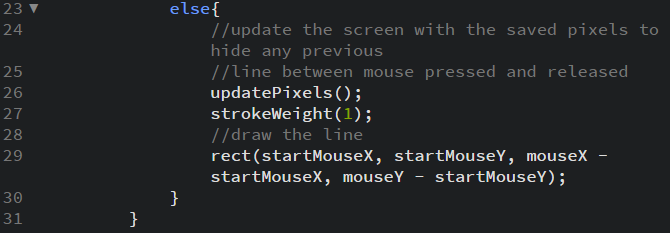
Report for Final Project.

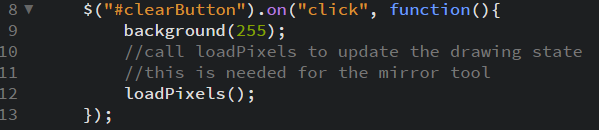
Due 29th April 2019

Chosen case study – Drawing application.

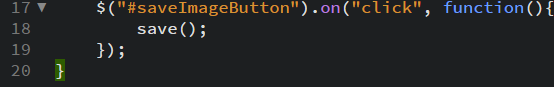
We have aimed to improve upon the base design of the drawing application supplied to us as case studies. Our first goal was to understand the code of the case study supplied to us, this was done by debugging and editing the code. This allowed us to add onto the program while minimising the bugs and also variables by reusing variables already initialised and used in the program, one such example of this was when we created a new variable for the previously used colour used by a certain tool, where in the end we obtained the value from “colourP.selectedColour.valueOf()” rubberTool.js – Line 5. By using object oriented design we also managed to minimise the use of global variables, in fact the only global variable we added was the aforementioned “prevColour” which we made a global variable as it is used in all the different tools. We have also added an ellipse tool which allows the user to create ellipses by clicking and dragging the mouse across the canvas. We edited the code from the freehand tool and essentially changed the else statement in “freehandTool.js” – line 29 from using a line function to an ellipse function in “ellipseTool.js” – line 22. By using object-oriented design even though some of the variables have the same names we could re-use the variables in different constructor functions without fear of bugs arising due to variables clashing. This is due to the variables being created and destroyed when needed and/or being updated at every constructor function. We also added a file drop function where a user can drag and drop an image to the canvas, once the picture has been loaded the image will then take the place of the canvas and allow the user to draw on top of the image. We used the “dragOver”, “dragLeave” and “drop” built in functions to achieve this. 

By adding functions into the parameters of the built in functions it calls “highlight” when “dragOver” is completed. “highlight” now tells the user what to do depending on the state of the program. When the user “drops” the file the canvas will be replaced by the image dragged and dropped by the user. One way we could have improved this was to add a try, catch or a way to check if the file added was indeed an image file, and if not shown an error message. Or by confirming to the user if the file is compatible and if it has been loaded or not. We also decided to add another shape tool called “rectangleTool.js” which again worked on the same principle as the “ellipseTool.js”.

This time replacing the “ellipse” in the else statement of the “ellipseTool.js” with a “rect”.

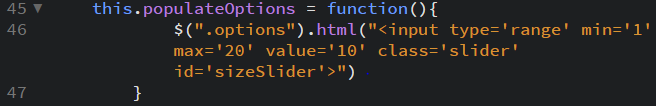
One of the helper functions we added was having a clear button, this effectively wiped the canvas allowing the user to start anew. This was done easily by having a button which made the background white when clicked. 

The other helper function we added was a save button for users to be able to save their creations. This was done by using the “save();” function.

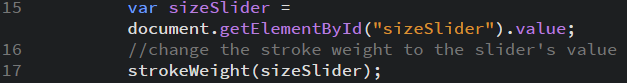


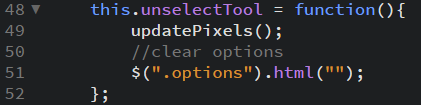
This would automatically save a snapshot of the canvas into the downloads folder of the computer used.

Another option we added for the user to use their creativeness in the application was to allow them to edit the thickness of the lines used in the different tools. We done this by adding a slider that would have the option to click and drag the slider left or right, with the left side having a smaller value and therefore thinner thickness, and vice versa. We added the slider in specific tools by using the “this.populateOptions” function.

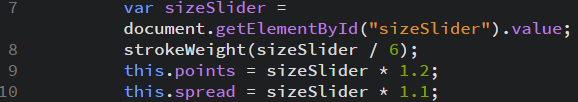


We obtained the value of the slider while the program was running by putting the code into the draw function. This meant it was called every time the program was refreshed.



This value was stored in the variable “sizeSlider” which was then used throughout the tool, for example in “strokeWeight”. When another tool was selected the “unselectTool” function is then called which will clear the options box. 

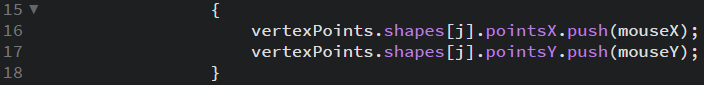
The slider was also used to edit the spread of the particles the “sprayCanTool.js” used. We added this as just increasing the thickness of the spray can particles didn’t have the desired effect we were hoping for.



By editing the spread of the particles as well this effect was achieved.



Another tool we added was the vertex tool. This tool involves a user clicking an unlimited number of points on the canvas, when the user is happy with the amount of points they press the enter button which will fill the shape. This tool was made using 2D arrays. When a user clicks on the canvas the X and Y coordinates get pushed to the array.



The number of points in the array is obtained by getting the length of the array, this value is used to state the number of times the for loop must reiterate itself. These points are used to create a shape in line 25 of “vertexTool.js” “beginShape()”. These points are used to create a shape in line 25 of “vertexTool.js” “beginShape()”. This is a way we dynamically modified arrays and used them to reiterate and process data.



When the user clicks enter the shape is closed and the shape will appear on the canvas with a fill that the user has selected. The enter keyCode also clears the array of vertexPoints so that it is clear for the next shape. This avoids bugs in the next instance the tool is used. This was a problem we had to debug as when the user used another tool the vertex tool would still be operational and create shapes.

To be more aesthetic we also edited the icons of the tools, and the cursor to be more suited to each tool. For example the freehand tool, we used the same icon in the toolbox selection as the cursor so that the user knows that the freehand tool is selected. This was done in the “sketch.js” file – line 93 and so on. It also took some time to figure it out how we can change the hotspot of cursor but in the last day we realized there is a very easy method.



The coding was done together in person which at the time was very productive however due to busy schedules was not done regularly. In the future a more routine and scheduled meeting may have been more suitable. We had goals every week that we had to submit to each other, and the code was then assessed by the other partner. In the future we would have liked to add more tools and improve the look of the application to look more modern and aesthetic. Also, we tried to add a description to the tools when the user hovers above a selected tool, but in the allotted time we were not able to do this.

* 4 March

1. Added Rectangle tool, partially added ellipse tool.
2. Adding Ellipse tool, centre of ellipse and ending is not user friendly.
3. Adding Ellipse tool and changing icons of tools.

* 11 March

1. Added Ellipse tool successfully. Change the icons, and cursor according to the different tools.

* 18 March

1. Added Rubber tool.
2. Rubber tool defaults to white when switching to another tool. Not user friendly.
3. Fixing Rubber tool so previous colour selected will be selected when picking another tool.

* 25 March

1. Rubber tool fixed and implemented correctly. Also added a slider to control the width of drawing tools, and number of particles used by the spray can.
2. Slider position on page is outside the canvas.
3. Fix slider position.

* 1 April

1. Used a global variable for previous colour so variable isn’t reused throughout each constructor function.
2. Slider position is still not fixed, tried using DOM. HTML.
3. Fix slider position. Add new tools.

* 8 April

1. Slider position fixed, used JavaScript to obtain values from slider. Added a new tool called Vertex tool.
2. Bugs with Vertex tool such as still operational after picking another tool, not having a fill.
3. Fix Vertex tool.

* 15 April

1. Vertex tool fill and outline can change colour.
2. Vertex tool still operational after picking another tool.
3. Debug Vertex tool.

* 22 April

1. Vertex tool debugged successfully.
2. N/A

* 29 April

1. Debugging and cleaning the code.
2. Cursor’s hotpoints are fixed.