



MAY 7, 2021

COMPUTING PROJECT REPORT

DRAWING APPLICATION

MOLLY PETERS & THOMAS MCALOON
COMPUTER SCIENCE
Goldsmiths, University of London

Contents

Implementation	2
Eraser Tool	2
Shape Tool.....	2
Upgraded Colour Palette	3
Stroke Slider	3
Tutorial	4
Planning and Coordination	2
How you structured your code:	4
Evaluation	5

Computing Project: Report

Planning and Coordination

In terms of working as a pair, we were able to work together well by splitting the workload equally. At the start of the project, we set out what we would like to add to the project. This included the addition of an eraser and fixing the editable shape tool. We had originally spoken about adding a fill tool, but later into the project when we had a look at implementing this, we found that it was too difficult, so we had a look at expanding on the shape tool instead.

Implementation

For this project we decided to work on the drawing application, we implemented a few tools as well as increasing the functionality of existing tools.

- Eraser Tool
- Shape Stamp Tool – Squares, Triangles and Stars
- Colour Picker (wider range of colours)
- Slider (for changing the stroke weight)
- Tutorial (explains how to use the drawing application)

Eraser Tool

The first addition we made was an eraser tool, this was a very simple yet necessary addition. Using a similar constructor function to the freehand tool, this allows the user to 'erase' any drawing essentially by colouring the canvas white. Originally, we used push and pop as the old version of the colour palette would not update the stroke colour when switching between tools after using the eraser. However, once we upgraded the colour palette this was no longer an issue as the stroke colour would update after the user selects any tool.

Shape Tool

We then added a shape stamp tool, at first, we kept it simple by adding the ability to stamp squares and later added the ability to stamp equilateral triangles, circles, ovals & stars. The main challenges we faced when implementing this tool was the ability to select a specific shape when using the shape tool and the ability to alter the size of each shape. The most efficient way we found to incorporate every shape into the shape tool was to create an object for each shape within the constructor function, for example:

```
this.square = function() // function to draw a Square based on mouseX, mouseY & the strokeWeightSlider value
{
  rect(mouseX - strokeWeightSlider.value/ 2, mouseY - strokeWeightSlider.value/ 2, strokeWeightSlider.value, strokeWeightSlider.value);
};
```

Once we had created all our shapes, we implemented a HTML select element to allow the user to pick from any of the shape objects, the example is below:

```
<select id="Shapes" data-title="Shape">
  <option value="1">Square</option>
  <option value="2">Triangle</option>
  <option value="3">Star</option>
  <option value="4">Circle</option>
  <option value="5">Oval</option>
</select>
```

For the JavaScript to understand which option the user has selected we had to create two variables to check the HTML Selector, one to add the HTML input into the JS and the other to check which option had been selected. Finally, we ran an if statement to check which shape had been selected and if the conditional was met, to run the function to draw the selected shape. The example is below:

```
var x = document.getElementById("Shapes");
var i = x.selectedIndex;
if(x.options[i].text === "Square" && mouseIsPressed) // If
{
  this.square();
}
else if(x.options[i].text === "Triangle" && mouseIsPressed)
{
  this.triangle();
}
```

Upgraded Colour Palette

We upgraded the colour palette to an advanced colour picker; this gives the user a wider variety of colours to choose from instead of the set colours that we were given at the start of this project. We first tried to use a p5.js createColorPicker() function ([Documentation](#)), however, we could not find a way to make the colour update once the user had selected a new colour or have the colour update with every tool. Instead, we used a HTML Colour Picker and created a new variable within each constructor that could find the colour value selected by the user and then used this variable as the stroke/fill value. There is an example of the HTML & JS below:

```
<input data-title="The Stroke Colour Palette" data-intro="This colour palette is for the initial strokes or outer lines of the shapes" id="colourPaletteStroke" type="color">
<input data-title="The Fill Colour Palette" data-intro="This colour palette is for filling in the shapes as you paint them onto the canvas" id="colourPaletteFill" type="color">
```

```
var selectedColourStroke = document.getElementById("colourPaletteStroke");
// ...
stroke(selectedColourStroke.value); // Determines the value for
```

Stroke Slider

Also, we added the use of a slider so that the user can change the size of the stroke weight, in turn this also increased the size of the shape stamps as well as being able to use it with all other tools including freehand brush, line tool and mirror tool. Similar to the Colour Palette, we initially tried to use a p5.js createSlider() function ([Documentation](#)), however, the sliders would raw over one another as you selected a new tool and we could not find a way to refresh the slider once another tool had been selected. Instead, we used a HTML Input Slider and created a new variable within each

constructor that could find the slider value selected by the user and then used this variable as the stroke weight value.

There is an example of the HTML & JS below:

```
<input type="range" min="1" max="250" value="5" class="slider" id="strokeWeight">
```

```
var strokeWeightSlider = document.getElementById("strokeWeight"); // Creating a variable for the strokeWeight using the HTML slider element
```

```
strokeWeight(strokeWeightSlider.value); // Determines the value for the strokeWeight using the slider value
```

One problem we have encountered with this method is that the slider does not work as well on Firefox but as 3.42% (Source: [Oberlo](#)) of Internet users use this browsers we feel it's not a massive issue.

Browser compatibility

[Report problems with this compatibility data on GitHub](#)

	Desktop						Mobile					
	Chrome	Edge	Firefox	Internet Explorer	Opera	Safari	WebView Android	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet
type="color"	20	14	29	No	12	12.1	4.4	25	27 ★	12	12.2	1.5
autocomplete	20	14	No ★	No	?	?	?	?	No ★	?	?	?
list	20	14	No ★	No	?	12.1	?	?	No ★	?	12.2	?

Full support

No support

Compatibility unknown

★ See implementation notes.

Tutorial

After adding these tools, we thought it would be a fun addition to add a tutorial, when opening the drawing application there are instructions showing the user how the application works and what each button is for. We used a JavaScript Library ([Intro.JS](#)) to implement a tutorial as we also felt it necessary due to the lack of labels of some of our features.

How we structured our code

As we were using constructor functions, we ensured that our variable and function names were kept simple, consistent and suitable throughout the project so that they were easy to understand. We also ensured that the correct indentation was used throughout so that the code was readable. Also making sure that all the code was commented clearly so that we could come back to it later if we needed to fix any bugs.

Evaluation

As mentioned above there was a slight issue towards the start of the project when it came to commenting the code, because we were working on this together there were times where we had added to the code but forgot to include a small comment which made it difficult on a few occasions when we needed to go back and make some small adjustments on certain tools and comments were missing. Luckily, this only happened a couple of times, but for next time we would ensure that we had left comments before moving on to the next task.

We ensured to test our code when either of us would implement new features, we did this by using all the tools in a random order and ensuring to use the new feature thoroughly across the app, noting any bugs along the way.

If we were to do this project again, we would have a look into more extensive and complex tools. When starting out this project we loved the idea of having a fill tool, however, we could not wrap our heads round a way of doing it. As we had decided to create a stamp tool, rather than being able to “drag” draw your own shape, we found that it would be more efficient to use the colour picker with the already existing fill function, rather than creating a brand-new extension to fill in set bounds. However, if we were to do this project again it would have been nice to add the ability to draw out the shapes rather than having the stamp effect, but we are still very happy with the way our shape tool turned out.

Overall, we are happy with how our project turned out and all the tools are usable although we ran into a couple of issues along the way we and we did not add as many extensions as we had originally planned, we were still able to extend and increase the functionality of already existing tools and we are very satisfied that our drawing application runs smoothly with few to no errors.

Progress Log

***** 26/03/2021 *****

so far we have tried to implement an eraser tool and the editable shape tool.

~ ERASER TOOL ~

- added eraser tool to sidebar

- created the eraser tool to essentially just 'draw' white.

- * at the moment it is slightly broken when switching from the eraser to other tools ~ e.g. when switching back to freehand tool, a colour needs to be reselected for it to work.

- plan to play around with the colour palette to get this working smoothly.

- plan to add the ability to change size (strokeWeight) of the eraser.

~ EDITABLE SHAPE TOOL ~

- added shape tool to sidebar

- made a start with the editable shape tool, drawing new vertices and being able to edit them.

- * currently the constructor is broken ~ this tool is a work in progress.

- plan to add a choice of shapes to draw inc: squares, circles, triangles and stars.

- plan to add ability to change strokeWeight of shape.

***** 12/04/2021 *****

~ ERASER TOOL ~

- eraser tool has now been fixed when switching between other tools, works as intended
- * had a play around creating a button to increase the strokeWeight of eraser but it ended up changing the strokeWeight for other tools (line tool and spray can)

~ FREEHAND TOOL ~

- added the same button to the freehand tool to change between strokeWeight
- plan to look into a slider tool for the strokeWeight size of all tools in the app
- plan to add a colour picker to allow a wider range of colours to be used

***** 20/04/2021 *****

~ COLOUR PALETTE ~

- implemented a colour picker to replace the previous colour palette, this allows for a wider range of colours
- * currently this is broken, when opening the different tools it repeatedly opens the colour picker

~ SHAPE TOOL ~

- decided to create our own shape tool as unfortunately couldn't get the previous editable shape to work.
- this tool works more like a stamp tool rather than drawing the shape, currently stamps a square on mouse click.

* at the moment the stamp can only draw a square of one size

- plan to add a choice of shapes that can be drawn including circles, triangles and stars.
- plan to add the ability to change the size of the shape.

***** 23/04/2021 *****

~ COLOUR PALETTE ~

- colour picker has now been fixed, can be used for each tool without any errors

~ STROKE WEIGHT SLIDER ~

- implemented a slider tool for changing the stroke weight
- the slider can be used for all tools including the shape tool (chose the slider option instead of using the previous buttons that were added)
- stroke weight goes from a minimum of 1px up to a maximum of 250px
- * issue with the line tool, when using the slider the line shows at the bottom of the screen (possibly to do with canvas boundaries)

- plan to look into the line tool to correct the issue with the canvas

***** 27/03/2021 *****

~ COLOUR PALETTE ~

- added another colour picker, one colour picker can be used to select the stroke colour (left side) and the second colour picker can be used to select the fill colour (right side)
- this fill colour picker is for use with the shape tool
- * issue when switching between tools the fill colour does not show on certain shapes ~ possibly something to do with the stroke weight slider

***** 28/04/2021 *****

~ LINE TOOL ~

- looked into canvas boundary issue, added an if statement to reduce boundaries
- * in some cases a line is still being drawn if you drag your mouse upwards whilst using the slider
- plan to look into this to avoid random lines being drawn

~ MIRROR TOOL ~

- checked to see if stroke colour picker works on mirror tool, working as intended
- * we've noticed that the guide mirror line has disappeared since adding colour pickers/stroke weight slider
- plan to look into a fix for guidance lines

***** 03/05/2021 *****

~ SHAPE TOOL ~

- we have now added 2 more shapes to our stamp tool, can now choose triangles and stars
- stamps shape on mouseclick
- plan to add a button so that the user can switch between shapes when using the tool

***** 04/05/2021 *****

~ SHAPE TOOL ~

- added a dropdown menu to allow the user to switch between the different shapes
- * currently the dropdown is always displaying no matter which tool you are on
- plan to look into only displaying dropdown once shape tool is open

***** 06/05/2021 *****

~ DRAWING APP TUTORIAL ~

- added a tutorial, on opening there is a tutorial that goes through and explains each function of the application (this was just a fun addition)

~ COMMENTING ~

- went through all of the code to double check for comments

Molly Peters and Thomas McAloon

- clearly commented throughout

***** 07/05/2021 *****

~ SHAPE TOOL ~

- decided to add additional shapes to the shape tool including a circle and oval