

1.

## Question 1

Outline your project and which extensions you are building on to the template. (400 words)

Your answer should include:

- Which of the templates you are extending and why have you chosen to do it.
- What extensions have you chosen to do. You should include in your answer: any complex coding techniques you will need to use, (such as arrays of objects, constructor functions, nested looping); the complexity of the extension; and any expected challenges you will have implementing it.

I am extending the drawing app because I am more familiar with this kind of application. I am a GIMP and Inkscape user and I know some of the features that are missing or that would be ideal they had. Another reason why I chose the drawing application is because I emigrated from my country and there have been challenges that I have had to face that did not allow me to complete my work as I intended. In fact, I wanted to make a radar graph extension in the data visualisation app, but due to the reasons previously explained I considered that extending the drawing app was a better and safer idea.

Firstly, I will develop the Stamp tool and the Editable Shapes extensions. I will modify the latter so that it has a “close shape” option, that allows the user to join the start and the end of the shape with a straight line. For these extensions I will use arrays.

Secondly, I will create an Eye Dropper tool, an Image tool, and a Bezier tool. In the dropper tool I will make use of the p5.js’ `get()` function to get the colour of the pixels. The image tool will be just an extension to give some completion to the app, as it doesn’t have the option to add images in the canvas. The Bezier tool extension will be a little bit more complicated, as it will make use of p5.dom’s sliders to bend lines according to Bezier’s polynomials.

Thirdly, I would like to make a Cropping tool, a Move tool, and a Layer tool. The Cropping tool will be more feasible, as I will just make a shape and make the application re-draw only the region enclosed by the shape. To achieve this, I have the option to use p5.js’ `loadPixels()` and `updatePixels()` functions. I can also do it by means of the `pixels[]` array and selecting the pixels that correspond to the cropped region. I am still deciding which approach to use.

The Move tool will load the canvas pixels in the `pixels` array and displace them by the X and Y amount determined by the mouse position `mouseX` and `mouseY`. It will be challenging to convert each drawing on the canvas into objects, so I will try to find a different approach.

The layer tool is more challenging. I will need to create an array of canvases that can move between each other. I will also use buttons to assign canvas to that array and change the index associated with each layer. Time will also be a challenge to complete this extension.

## 2.

### Question 2

Discuss the progress you have made on your project so far. (400 words)

Your answer should include:

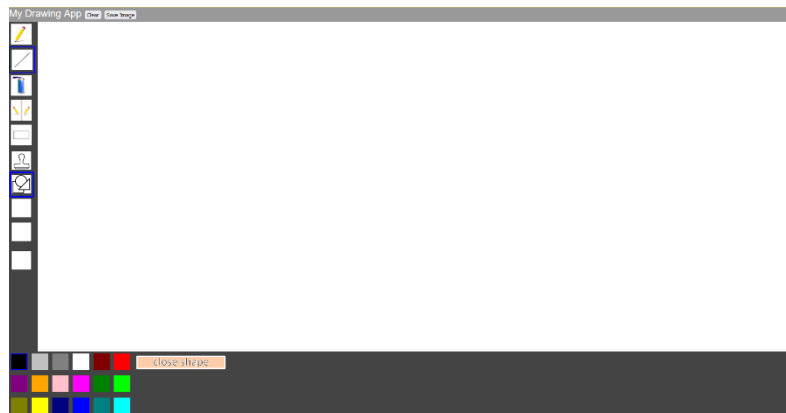
- What design work and research you have done while developing your extensions. You can include diagrams here in support of your answer. This might include but not be limited to: brainstorming, interface designs, code diagrams, and notes on the ideas and code.

You don't have to include everything you have done up to this point but try and be selective and select the best examples of your process.

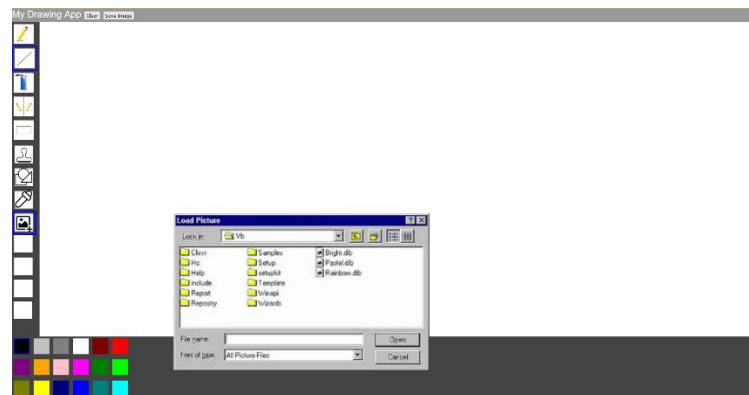
- How much of the code have you written so far. What do you intend to do next?

The progress I have made so far basically refers to some ideas for the user interface and some ideas of how to accomplish the tasks. Additionally, I have looked at some online forums that have guided me in terms of how to better communicate with users and, in the case of the Bezier tool, how the formula is applied to the curve, in addition to the code used for it.

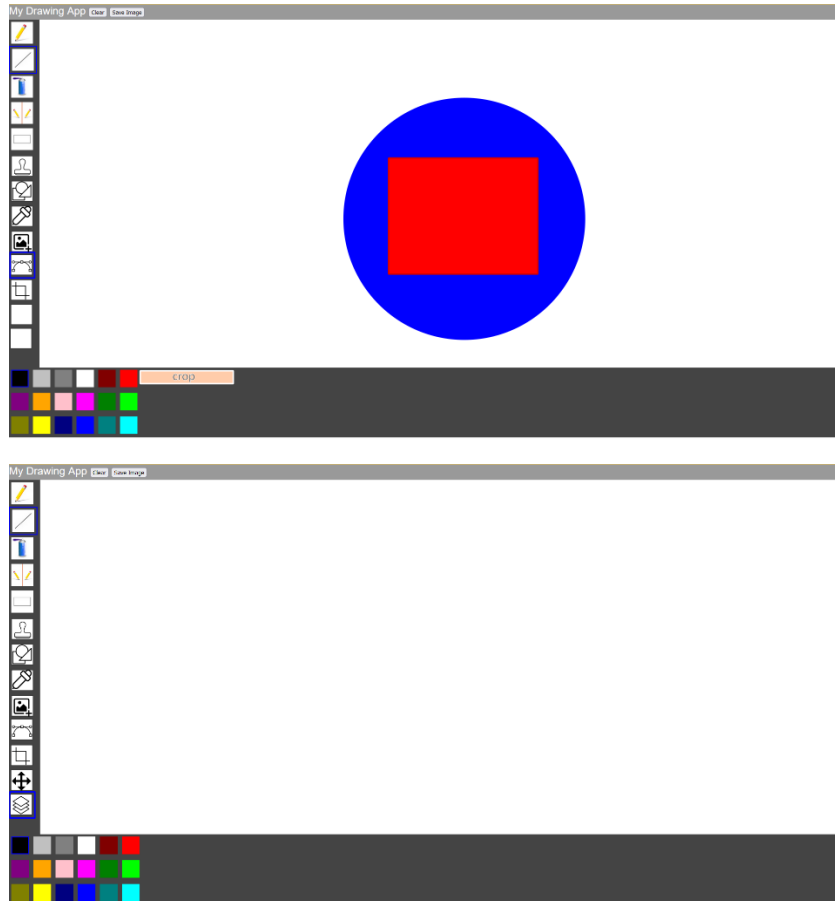
- Stamp tool and Editable Shapes tool:
  - For these extensions I will use the indications devised on the instructional videos. However, I will include my initial idea of what these extensions will look like in the app.



- Eye dropper tool, Image tool, and Bezier Tool:
  - The eye dropper tool will have a button to call the system's colour wheel and below the button there will be a box that shows the selected colour.
  - The image tool will call the system's dialog to choose files. It will also have two slides to manipulate the image dimensions once they have been drawn on the canvas.
  - The Bezier tool will have two sliders that will determine the extent of the curvature of the line.



- Cropping tool, Move tool, Layer tool:
  - The Cropping tool will have a button that will select the tool, and another button that will make the crop effective.
  - Move Tool: this tool will move the drawing depending on the position of the mouse. Additionally, it will have two sliders to more precisely accommodate the element on the canvas.
  - Layer tool: definitely the tool that I am more unsure about. My initial idea is to limit it to only three layers, as I don't have much time to complete more complex work. This will be just to prove how much I have learnt in the course.



Ideas I have taken from other sources:

- For the Bezier tool I have used Wikipedia to learn the theory behind the tool. I learnt that it is a polynomial and how it is mathematically described. In addition to that, I found some JavaScript code to implement it, however I still need to modify it so that it fits the template provided.
- I also used p5.js' reference to know about the `get()` method, the `pixels[]` array and some other useful functions.

- I found an algorithm to implement layers that will be useful. I will share it in the resources section.

Code written: I haven't written much of the code needed, because of the reasons I previously exposed. However, as I haven't found a job yet (which I need), I can dedicate 10 hours a day to complete my work.

### 3.

#### Question 3

Discuss how you will organise your time for the rest of the project. (200 words)

- Your answer should be based around a Gantt chart or other time organisation process. You can easily make this in a spreadsheet program and export an image or take a screenshot for inclusion here. Any words on the diagram do not count towards the word count.

If you use a Gantt chart should include the total time of the projects in weeks. You will also need to divide up the parts of your program and submission into activities for allocating time to on the chart.

- Your timescales should include progress to this point as well as afterwards.
- You should discuss how you have decided on the time you have allocated to each activity and why you have allocated these amounts of time.
- You should make reference to your chart to clarify the points you make here.

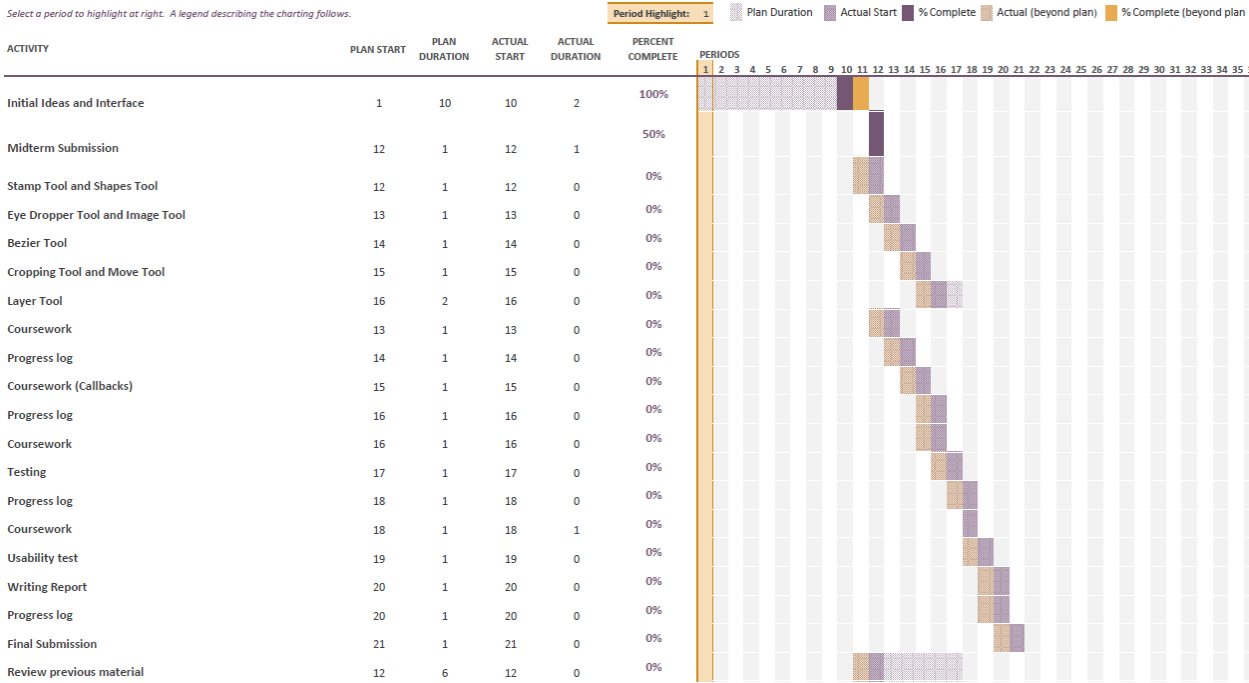
I will organise my time in the following way (this can be seen in the Gantt chart):

I already completed the initial phase of brainstorming ideas for the extensions and possible user interfaces. This might change during the testing stages, as developing is a cyclical process of refinement.

In week 12 I will be developing the Stamp tool and the shapes tool. As they are relatively simple and we were given some help in the instructional videos, I am giving one week to be completed. The Eye dropper and the image tools are also simple implementations. I am giving them one week, though if I can complete them in less time I will begin with the Bezier tool, even though I expect to begin it in week 14. For the Move and Crop tools I expect some big challenges, so I am allocating 2 weeks of work for them. I will begin the Layer tool in week 15 and I will reserve 2 weeks for it. If by this time I haven't made good progress, I won't include it in the final submission. I will also dedicate some time for the coursework and the progress logs, which are described in the Gantt chart.

# Project Planner

Select a period to highlight at right. A legend describing the charting follows.



#### 4.

##### Question 4

List any external sources that you have actively utilised in your project. This should include:

- any code you have used from external sources directly
- any code that you have taken inspiration from but adapted and refined for the project (such as pseudocode algorithms or code pens)
- any online help forums you have taken code from (i.e. StackOverflow or library documentation)
- any third-party libraries you are using.
- **Any prompts issued to large language models (such as ChatGPT) and its responses.**

You do not need to include everything you have read or that has helped you. Only where you have used or adapted code that appears in your project.

##### **Bezier curves**

<https://stackoverflow.com/questions/785097/how-do-i-implement-a-b%C3%A9zier-curve-in-c/11435243#11435243>

<https://stackoverflow.com/questions/785097/how-do-i-implement-a-b%C3%A9zier-curve-in-c>

<https://stackoverflow.com/questions/37642168/how-to-convert-quadratic-bezier-curve-code-into-cubic-bezier-curve/37642695#37642695>

<https://jsfiddle.net/fQYsU/>

<https://stackoverflow.com/questions/16227300/how-to-draw-bezier-curves-with-native-javascript-code-without-ctx-beziercurveto>

<https://www.moshplant.com/direct-or/bezier/math.html>

##### **Layers algorithm**

<https://cantwell-tom.medium.com/adding-layers-to-a-drawing-app-pt-1-efbc2adb1692>

##### **Crop tool (Google AI Overview)**

**Prompt:** how to make a crop tool in p5.js

##### **Response:**

To create a crop tool in p5.js, you need to primarily use the "get()" function to extract a specific section of an image based on user interaction (like dragging a selection box), then display that cropped section on the canvas; key elements include loading the image, tracking mouse position, drawing a selection rectangle, and using "get()" to crop the image when the user confirms the selection.



Basic steps:

- Load the image:

Use the loadImage() function in the preload() function to load the image you want to crop.

Store mouse coordinates:

Use mouseX and mouseY variables to track the mouse position as the user drags the selection box.

Draw selection rectangle:

In the draw() function, draw a rectangle based on the mouse coordinates to visually represent the crop area.

Get cropped section:

When the user confirms the selection (e.g., by clicking), use the get() function to extract the pixel data within the rectangle from the loaded image.

Display cropped image:

Draw the cropped image on the canvas using the image() function.

```
1 let startX, startY, selectionWidth, selectionHeight;
2
3 let isDragging = false;
4
5
6
7 function preload() {
8   img = loadImage('path/to/your/image.jpg');
9 }
10
11
12
13 function setup() {
14   createCanvas(400, 300);
15 }
16
17
18 function draw() {
19   background(100);
20   image(img, 0, 0);
21
22   // (isDragging)
23   rect(startX, startY, selectionWidth, selectionHeight);
24 }
25
26
27
28
29
30
31 function mousePressed() {
32   isDragging = true;
33   startX = mouseX;
34   startY = mouseY;
35 }
36
37
38
39 function mouseDragged() {
40   if (isDragging) {
41     selectionWidth = mouseX - startX;
42     selectionHeight = mouseY - startY;
43   }
44 }
45
46
47 function mouseReleased() {
48   isDragging = false;
49   // Get the cropped image
50   let croppedImage = img.get(startX, startY, selectionWidth,
51     selectionHeight);
52   // Display the cropped image (adjust position as needed)
53   image(croppedImage, 100, 100);
54 }
```

**Code:**

**Key points to consider:**

- User interaction:

Implement clear visual cues to indicate where the selection area is and how to interact with the crop tool (e.g., highlighting the selection rectangle, changing cursor behavior).

Aspect ratio constraints:

You can add logic to maintain the aspect ratio of the selection area if needed.

Canvas resizing:

If you want to allow users to resize the canvas to fit the cropped image, update the canvas dimensions after cropping.

Advanced features:

- Drag handles: Add interactive handles at the corners of the selection rectangle for finer control.

Rotation: Allow users to rotate the cropped area.

Saving the cropped image: Use `canvas.toDataURL()` to generate a data URL of the cropped image, which can be downloaded or used further.

**5.****Question 5**

Upload your work in progress project files as a zip. Please don't use any other compression file formats.

**6.****Question 6**

Please upload your code in text format (not screenshots) in a PDF file here. Please clearly label with start and end comments exactly which sections of code you personally wrote without assistance. 5% of the marks for this coursework are reserved for this part.

7.

#### Question 7

Discuss how your project will be original and unique (100 words). For example:

- What novel features have you developed for your application design.
- How you have applied complex coding techniques to your ideas and designs.
- How have you adapted and evolved inspiration you have seen from other sources.
- Can you justify your ideas in terms of a use case or use group where existing solutions aren't available.

My project is unique in the sense that I tried to replicate functionality already present in open-source projects like Inkscape and GIMP. A p5.js application will not give enough room for a full-blown application with all the required extensions, but it will give future coders the opportunity to look at an implementation they can try as part of their learning.

A group that will benefit from my application are students who are new to programming and lack the knowledge to understand how those features are implemented in commercial applications. They can study my solutions in an implementation easy to understand.