University of London

Bachelor of Science (Honours) in Computer Science (Machine Learning and Artificial Intelligence)

Individual Midterm Report for Week 12

<Nguyen Ngoc Quoc Cuong> <SIM ID: 10245826>

Academic Year 2023

Year 1 Semester 2

Abstract

This report is a documentation of Cuong’s plans and experiences for the Drawing app template which contains the three extensions he chose to work on, helper functions he would like to include, and challenges he faced along the way.

Table of Contents

List of Figures

CHAPTER 1: Project’s outline

1.1 Selecting a template

Out of the three options, I decided to work on the Drawing App as having friends who draw may help me get inspiration for this project.

1.2 Chosen extensions

The main set of extensions I decided to implement are the Stamp Tool extension with rotatable stamps, the Editable Shape extension with a button to fill in the shape, and the Blur Tool as my unique extension.

Alongside the extensions, I decided to implement additional helper functions such as Drawing Layers, Eraser Tools, Grids, and a Thickness Slider to make the project feel more complete.

1.3 Coding Techniques (explain the details, not the requirements. Like why it works not how)

The Stamp Tool, regarding the rotatable stamps, begins by finding the difference between the mouse’s X coordinate and the stamp’s X coordinate. As the user drags the cursor away from where they placed the stamp, the stamp will rotate toward the direction and along the movement of the cursor until the mouse’s left button is released. Additionally, updatePixel() will be running continuously during this process to prevent the stamp from being drawn onto the page before the user releases the mouse.

The Editable Tool, regarding the fill-in-shape button, begins with me adding another button through the ‘this.populateOptions’ function. After pressing the Fill button, the code will begin with creating a point with the same colour as the shape at the starting coordinate. Then, the code will use the get() function to “pull” the current shape’s colour and use it in the fill() function.

1.4 Extension Complexity and Challenges (mention guardian Blur at the start)

An ongoing challenge is the Gaussian Blur Tool. The tool will require me to make multiple nested for-loops to traverse every pixel nearby the cursor and apply a “kernel array” to them in a set of three-by-three. These actions of scanning, extracting colour values and averaging them to change the colour for the centre pixel will possibly repeat thousands of times during the usage of the tool to give the area a blur effect. This can be confusing during coding. Additionally, I will have to make these extensions to be able to work with all drawing layers.

One other challenge Im facing is that my drawing application took too much of the browser's and the desktop's resources. Before the optimisation, the application tab can take up to 4Gb of my RAM and cause my browser to use 50% of my CPU. This obviously lead to the browser kept crashing. To manage this, I put all of the major lines of code from the draw() function of sketch.js inside an 'if' statement such that the canvas would only render when the mouse's button is pressed. Though the optimisation can be improved, the browser will not crash so easily now.

coursera copy:

1.1 Selecting a template

Out of the three options, I decided to work on the Drawing App as having friends who draw may help me give me an advantage.

1.2 Chosen extensions

The main set of extensions I chose are the Stamp Tool extension with rotatable stamps, the Editable Shape extension with a button to fill in the shape, and the Blur Tool as my unique extension.

Alongside the extensions, I decided to implement additional helper functions such as Drawing Layers, Eraser Tools, Grids, and a Thickness Slider to make the project feel complete.

1.3 Coding Techniques

The Stamp Tool, regarding the rotatable stamps, works by saving the mouse's coordinate in an object, objTemp. The code will then find the difference of 'mouseX - objTemp.x'. This difference will determine the rotation direction of the stamp when used in rotate(angle);. For example, if the mouse drags to the left, the difference output will be negative, allowing the stamp to rotate anti-clockwise. Additionally, updatePixel() will be running continuously during this process to prevent the stamp from being drawn onto the page before the user releases the mouse.

The Editable Tool, regarding the fill-in-shape button, works by adding a 'Fill' button through the ‘this.populateOptions’ function. Pressing the 'Fill' button will create a dot, by using the point() function, with the same colour as the shape at the starting coordinate. Then, the get() function will be used to “pull” the dot’s colour and use it in the fill() function. This will overwrite the noFill() function and fill-in the shape.

1.4 Extension Complexity and Challenges

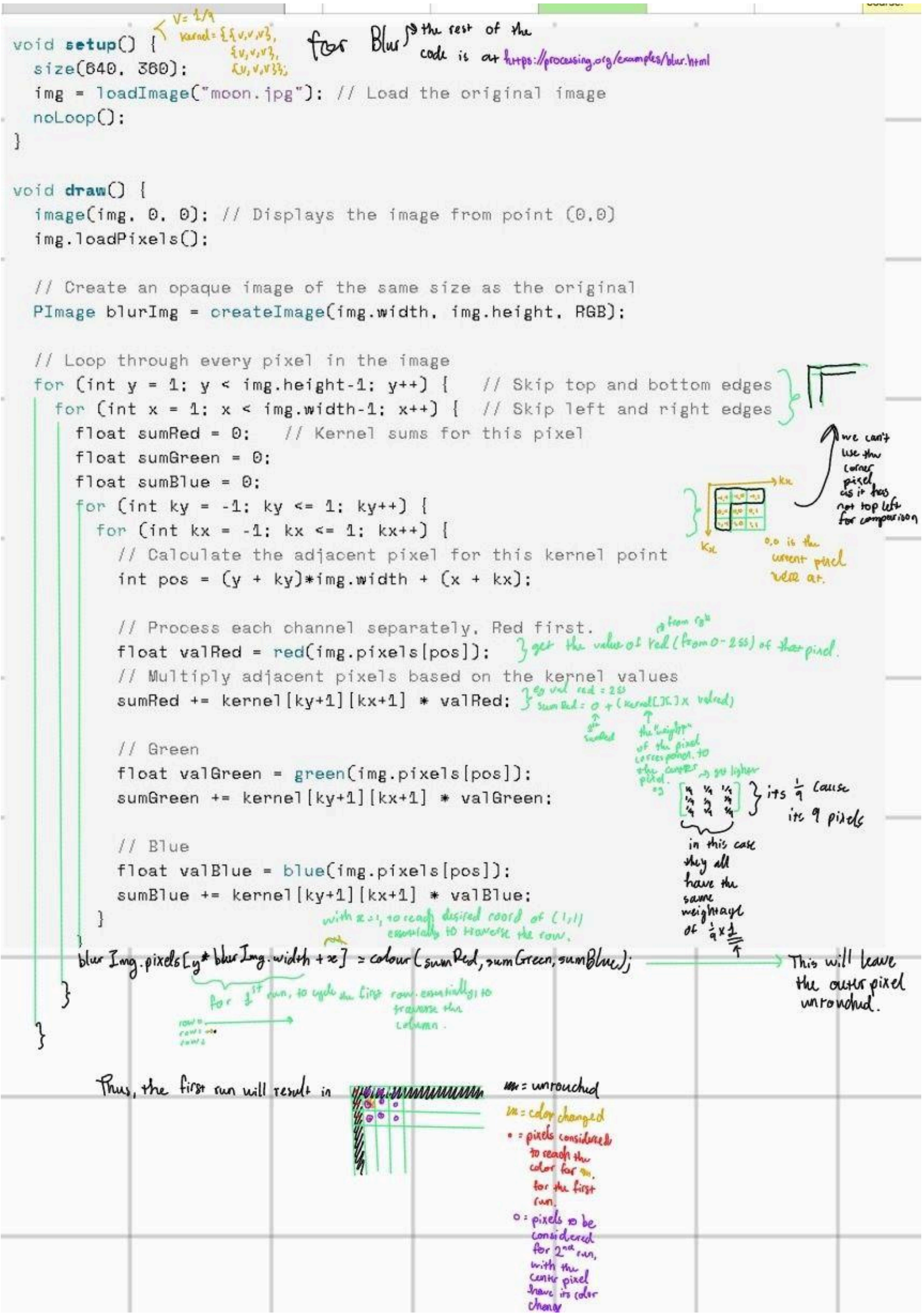
An ongoing challenge is the Gaussian Blur Tool. The tool will require me to make multiple nested for-loops to traverse every pixel nearby the cursor and apply a “kernel array” to them in a set of three-by-three. These actions of scanning, extracting colour values and averaging them to change the colour for the centre pixel will possibly repeat thousands of times which can be confusing during coding. Additionally, I will have to make this extension to be able to work with all drawing layers.

One other challenge I'm facing is that my project took too much resources. Before the optimisation, the browser tab can take up to 4Gb of RAM and use 50% of my CPU. This obviously lead to the browser crashing. To manage this, I put all of the major lines of code from the draw() function of sketch.js inside an 'if' statement such that the canvas would only render when the mouse's button is pressed. Though the optimisation can be improved, the browser will not crash so easily now.

word count: 432

Figure 1.1: (My notes for the Gaussian Blur Code method.) https://drive.google.com/file/d/1NlDO83lpEQLlPENRFB7F3a8hcwoStM0A/view?usp=sharing

Figure 1.2: The draw() function of the sketch.js. https://drive.google.com/file/d/15FLV2CVLk_QIj4wUJxCPXOLLnL2W0axP/view?usp=sharing



```
void setup() {
  size(640, 360);
  img = loadImage("moon.jpg"); // Load the original image
  noLoop();
}

void draw() {
  image(img, 0, 0); // Displays the image from point (0,0)
  img.loadPixels();

  // Create an opaque image of the same size as the original
  PImage blurImg = createImage(img.width, img.height, RGB);

  // Loop through every pixel in the image
  for (int y = 1; y < img.height-1; y++) {   // Skip top and bottom edges
    for (int x = 1; x < img.width-1; x++) {  // Skip left and right edges
      float sumRed = 0;   // Kernel sums for this pixel
      float sumGreen = 0;
      float sumBlue = 0;
      for (int ky = -1; ky <= 1; ky++) {
        for (int kx = -1; kx <= 1; kx++) {
          // Calculate the adjacent pixel for this kernel point
          int pos = (y + ky)*img.width + (x + kx);

          // Process each channel separately, Red first.
          float valRed = red(img.pixels[pos]);
          // Multiply adjacent pixels based on the kernel values
          sumRed += kernel[ky+1][kx+1] * valRed;

          // Green
          float valGreen = green(img.pixels[pos]);
          sumGreen += kernel[ky+1][kx+1] * valGreen;

          // Blue
          float valBlue = blue(img.pixels[pos]);
          sumBlue += kernel[ky+1][kx+1] * valBlue;
        }
      }
    blur Img.pixels[y* blur Img.width + x] = colour(sumRed, sumGreen, sumBlue);
  }
}
```

V= 1/9
Kernel = {{v,v,v},
{v,v,v},
{v,v,v}};

for Blur → the rest of the code is at https://processing.org/examples/blur.html

we can't use the corner pixel as it has not top left for comparison

0,0 is the current pixel were at.

→ from rgb
get the value of red (from 0-255) of that pixel.

eg val red = 255
sum Red = 0 + (kernel[1][1] x valred)

1st summed

the "weight" of the pixel corresponding to the center pixel. → get lighter

eg $\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$ its $\frac{1}{9}$ cause its 9 pixels

in this case they all have the same weightage of $\frac{1}{9} \times 1$

with x=1, to reach desired coord of (1,1) essentially to traverse the row.

for 1st run, to cycle the first row. essentially, to traverse the column.

This will leave the outer pixel untouched.

Thus, the first run will result in

▨ = untouched

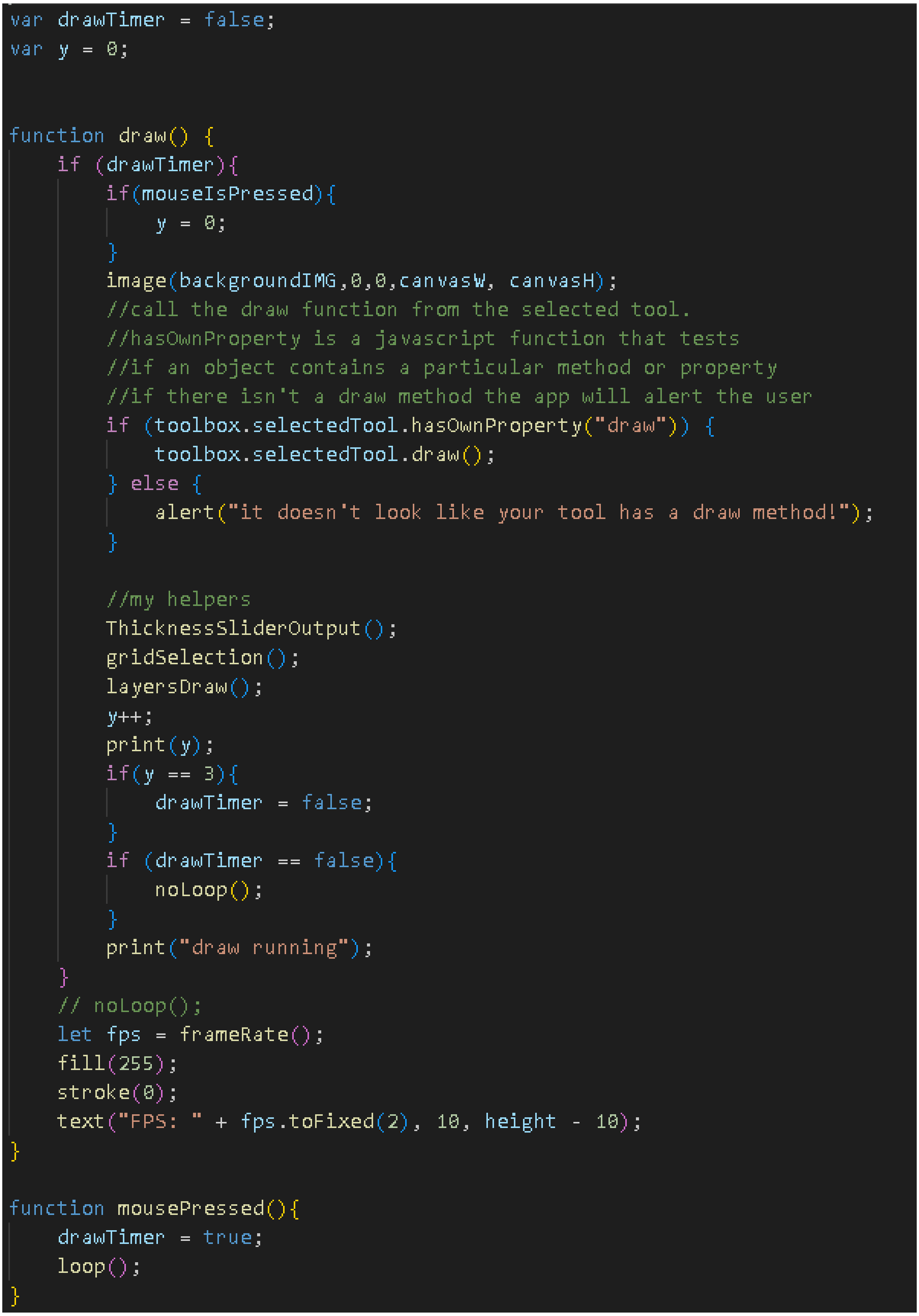
▨ = color changed

• = pixels considered to reach the color for ▨ for the first run.

○ = pixels to be considered for 2nd run, with the center pixel have its color change

Figure 1.1: My notes for the Gaussian Blur Code method.

(source: https://processing.org/examples/blur.html)



```javascript
var drawTimer = false;
var y = 0;


function draw() {
    if (drawTimer){
        if(mouseIsPressed){
            y = 0;
        }
        image(backgroundIMG,0,0,canvasW, canvasH);
        //call the draw function from the selected tool.
        //hasOwnProperty is a javascript function that tests
        //if an object contains a particular method or property
        //if there isn't a draw method the app will alert the user
        if (toolbox.selectedTool.hasOwnProperty("draw")) {
            toolbox.selectedTool.draw();
        } else {
            alert("it doesn't look like your tool has a draw method!");
        }

        //my helpers
        ThicknessSliderOutput();
        gridSelection();
        layersDraw();
        y++;
        print(y);
        if(y == 3){
            drawTimer = false;
        }
        if (drawTimer == false){
            noLoop();
        }
        print("draw running");
    }
    // noLoop();
    let fps = frameRate();
    fill(255);
    stroke(0);
    text("FPS: " + fps.toFixed(2), 10, height - 10);
}

function mousePressed(){
    drawTimer = true;
    loop();
}
```

Figure 1.2: The draw() function of the sketch.js.

CHAPTER 2: My Progress

2.1 Current Design Work

- **Brainstorming Ideas**

  With the help of my friend (@sufivehi on Instagram) and consultation with my teacher, I was able to think of many ideas on possible extensions I could work on. Insight into my brainstorming session can be seen in Figures 2.1 and 2.2 below, which are cutouts from my personal notes.

  I researched each of the applicable ideas and attempted to come up with a plan on how to code each add-on in order to rate their difficulty, which allowed me to avoid wasting time on add-ons that are either hard to implement or borderline impossible at my level.

  An example of my process is when I'm researching the Drawing Layer. I searched "Layer p5js" on Google and skim through the result. That is when I found a Youtube video "Scratch Off/Cut Out Effect (Using Layers in P5js)" made by Barney Codes. In the video, he made a tutorial on how to make a virtual scratch card, and that is when I was exposed to the function blendMode();. With P5js's online editor testing environment, I was able to test out my idea efficiently and came to a reasonable working code to implement layers in the Drawing App. This can be seen in Figures 2.3 and 2.4 below.

- **Interface Design**

  The design of the interface is neither my main priority nor my focus. Therefore, I will aim to make the interface look pleasant enough for the eyes while making the helpers such as the thickness slider, grid dropbox, and layer selector obvious and visible to the user. I will only start working on the UI after I am finished with everything.

2.2 Current Code Progress

I implemented the helper functions first as a warmup practice. Afterwards, as of the 6th of July, I completed the Stamp Tool and the Editable Shape Tool (represented by a flower). The Blur Tool is currently in a testing phase but I put it on hold since I want to focus on the other modules first. I will start coding the Blur Tool after I submitted all of my mid-term deliverables.

coursera version:

2.1 Current Design Work

-Brainstorming Ideas

With the help of my friend (@sufivehi on Instagram) and consultation with my teacher, I was able to think of many ideas on possible extensions I could work on. Insight into my brainstorming session can be seen in Figures 2.1 and 2.2 below, which are cutouts from my personal notes.

I researched each of the applicable ideas and attempted to come up with a plan on how to code each add-on in order to rate their difficulty, which allowed me to avoid wasting time on add-ons that are either hard to implement or borderline impossible at my level.

An example of my process is when I'm researching the Drawing Layer. I searched "Layer p5js" on Google and skim through the result. That is when I found a Youtube video "Scratch Off/Cut Out Effect (Using Layers in P5js)" made by Barney Codes. In the video, he made a tutorial on how to make a virtual scratch card, and that is when I was exposed to the function blendMode();. With P5js's online editor testing environment, I was able to test out my idea efficiently and came to a reasonable working code to implement layers in the Drawing App. This can be seen in Figures 2.3 and 2.4 below.

-Interface Design

The design of the interface is neither my main priority nor my focus. Therefore, I will aim to make the interface look pleasant enough for the eyes while making the helpers such as the thickness slider, grid dropbox, and layer selector obvious and visible to the user. I will only start working on the UI after I am finished with everything.

2.2 Current Code Progress

I implemented the helper functions first as a warmup practice. Afterwards, as of the 6th of July, I completed the Stamp Tool and the Editable Shape Tool (represented by a flower). The Blur Tool is currently in a testing phase but I put it on hold since I want to focus on the other modules first. I will start coding the Blur Tool after I submitted all of my mid-term deliverables.

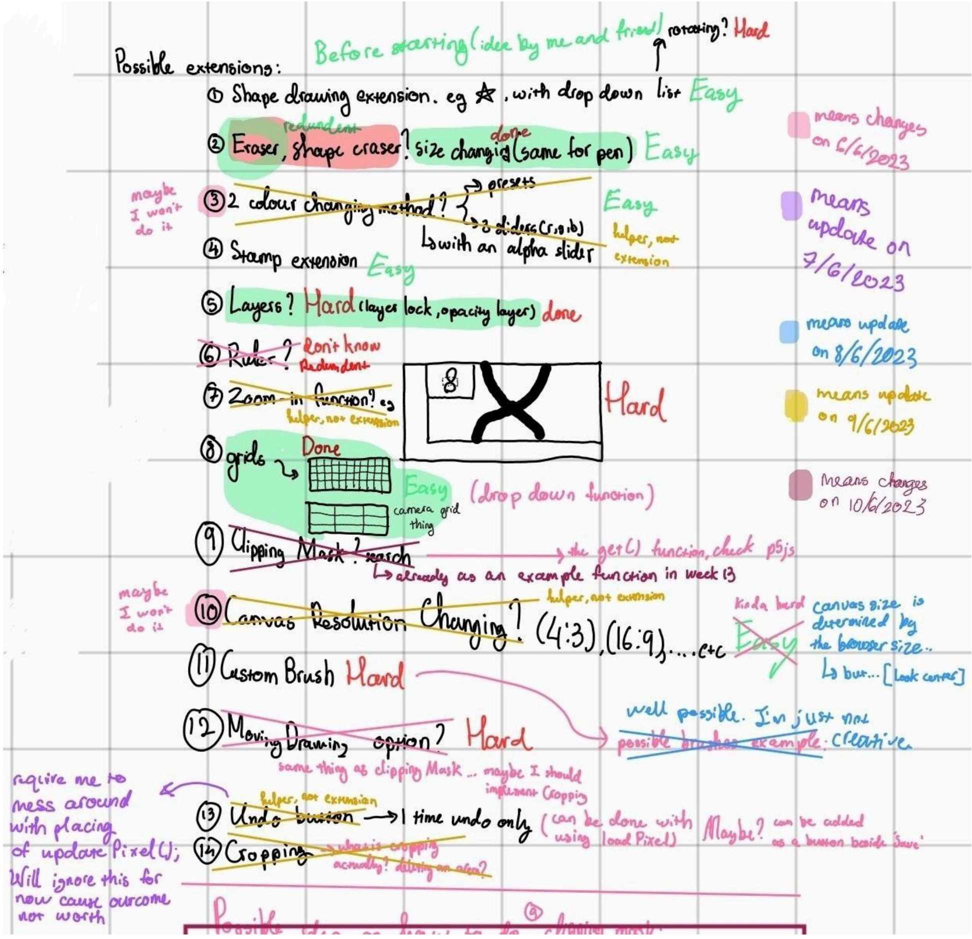
word count: 350

Figure 2.1: Result from the 1st brainstorming session after testing. https://drive.google.com/file/d/1Xt1-wkQLzXmYqSF52UbHHrmpKYKcUven/view?usp=sharing

Figure 2.2: 2nd Brainstorming session after consultation with my teacher. https://drive.google.com/file/d/1EuEqRes-KddtL1XUqzvtAX5MQpLXjHK1/view?usp=sharing

Figure 2.3: Barney's code for the scratch card. https://editor.p5js.org/BarneyCodes/sketches/syVZZiBHk

Figure 2.4: My codes to make Drawing Layers. https://editor.p5js.org/cuongnguyenlvl30/sketches/L8j5JxQTo
Figure 2.5: The current UI of the helper functions (6th July 2023) https://drive.google.com/file/d/1Tkg2y9EwXlkCWpHdswNL_rZGT5UgiOnd/view?usp=sharing
Figure 2.6: The current UI of the tools(6th July 2023) https://drive.google.com/file/d/1uYKW2vOiOJLRDvU5PiGSxO7Ch-tYBPI-/view?usp=sharing



Possible extensions: Before starting (idea by me and friend) rotating? Hard

① Shape drawing extension. eg ☆, with drop down list Easy

② Eraser, shape eraser? redundant size changing (same for pen) done Easy

maybe I won't do it ③ ~~2 colour changing method?~~ → presets / → ~~3 sliders (r,g,b)~~ ↳ with an alpha slider Easy helper, not extension

④ Stamp extension Easy

⑤ Layers? Hard (layer lock, opacity layer) done

⑥ ~~Ruler?~~ Don't know Redundant

⑦ ~~Zoom-in function?~~ eg helper, not extension Hard

⑧ grids → Done Easy (drop down function) camera grid thing

⑨ ~~Clipping Mask? search~~ → the get() function, check p5js ↳ already as an example function in week 13

maybe I won't do it ⑩ ~~Canvas Resolution Changing?~~ (4:3), (16:9)....etc helper, not extension ~~Easy~~ ✓ Kinda hard canvas size is determined by the browser size.. ↳ but... [look center]

⑪ Custom Brush Hard → ~~possible brushes example~~: well possible. I'm just not creative.

⑫ ~~Moving Drawing option?~~ Hard same thing as clipping Mask... maybe I should implement Cropping

⑬ ~~Undo button~~ helper, not extension → 1 time undo only (can be done with using load Pixel) Maybe? can be added as a button beside 'Save'

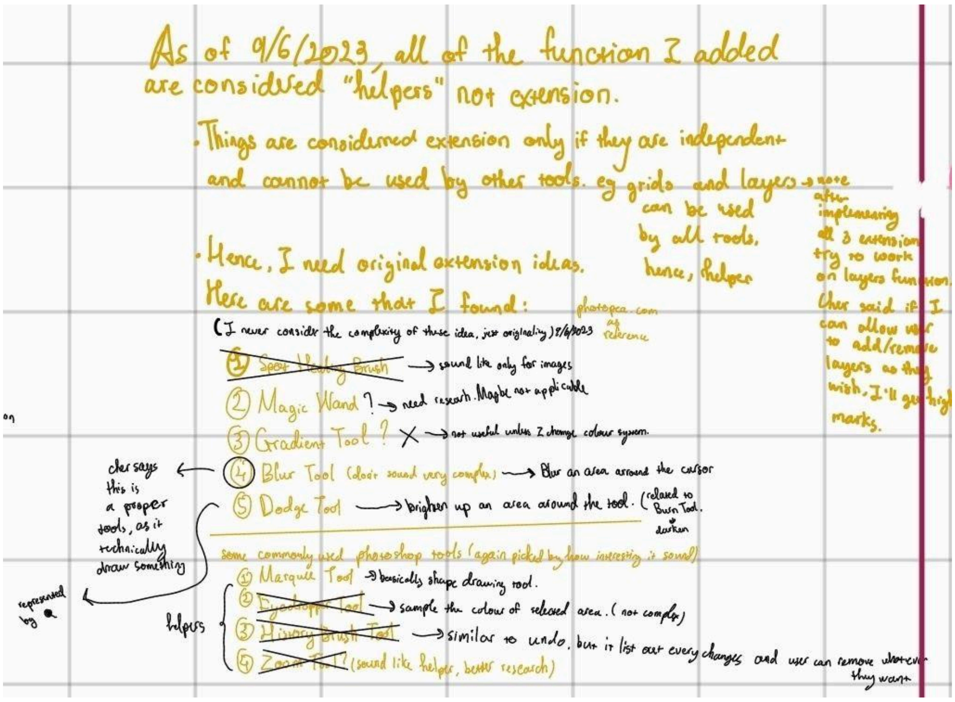
require me to mess around with placing of update Pixel(); Will ignore this for now cause outcome not worth

⑭ ~~Cropping~~ → ~~what is cropping actually? deleting an area?~~

Possible idea on how to do clipping mask: ⑨

■ means changes on 6/6/2023
🟪 means update on 7/6/2023
🟦 means update on 8/6/2023
🟨 means update on 9/6/2023
🟫 Means changes on 10/6/2023

Figure 2.1: Result from the 1st brainstorming session after testing.



As of 9/6/2023, all of the function I added are considred "helpers" not extension.

- Things are considemed extension only if they are independent and cannot be used by other tools. eg grids and layers can be used by all tools, hence, helper → note after implementing all 3 extension try to work on layers function. Cher said if I can allow user to add/remove layers as they wish, I'll get high marks.
- Hence, I need original extension ideas. Here are some that I found: photopea.com as reference

(I never consider the complexity of these idea, just originality) 9/6/2023

① ~~Spot Healing Brush~~ → sound like only for images

② Magic Wand ? → need research. Maybe not applicable

③ Gradient Tool ? X → not useful unless I change colour system.

④ Blur Tool (don't sound very complex) → Blur an area around the cursor

⑤ Dodge Tool → brighten up an area around the tool. (related to Burn Tool. darken

cher says this is a proper tools, as it technically draw something

represented by ●

some commonly used photoshop tools (again picked by how interesting it sound)

① Marquee Tool → basically shape drawing tool.

helpers

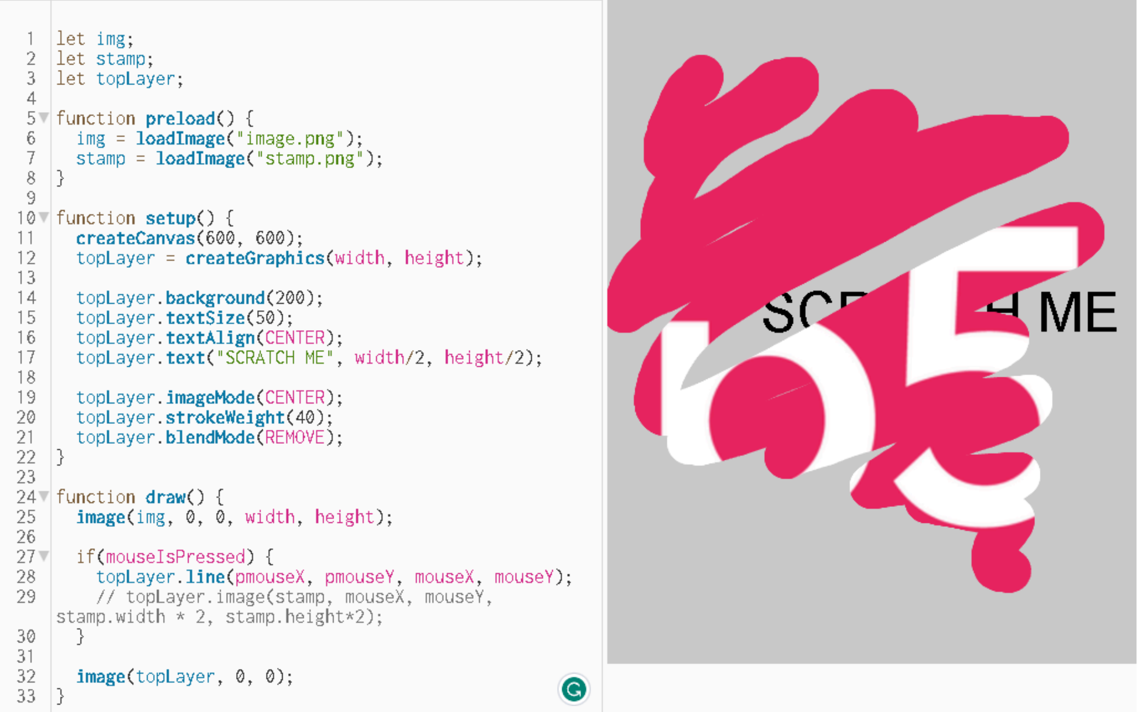
② ~~Eyedropper Tool~~ → sample the colour of selected area. (not complex)

③ ~~History Brush Tool~~ → similar to undo, but it list out every changes and user can remove whatever they want

④ ~~Zoom Tool~~ ? (sound like helper, better research)

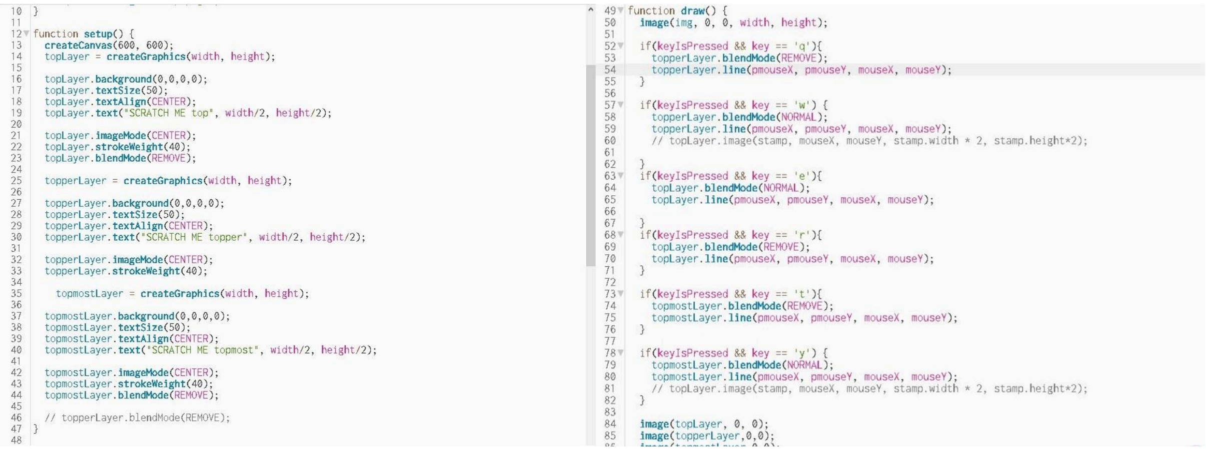
on

Figure 2.2: 2nd Brainstorming session after consultation with my teacher.



```
let img;
let stamp;
let topLayer;

function preload() {
  img = loadImage("image.png");
  stamp = loadImage("stamp.png");
}

function setup() {
  createCanvas(600, 600);
  topLayer = createGraphics(width, height);

  topLayer.background(200);
  topLayer.textSize(50);
  topLayer.textAlign(CENTER);
  topLayer.text("SCRATCH ME", width/2, height/2);

  topLayer.imageMode(CENTER);
  topLayer.strokeWeight(40);
  topLayer.blendMode(REMOVE);
}

function draw() {
  image(img, 0, 0, width, height);

  if(mouseIsPressed) {
    topLayer.line(pmouseX, pmouseY, mouseX, mouseY);
    // topLayer.image(stamp, mouseX, mouseY, stamp.width * 2, stamp.height*2);
  }

  image(topLayer, 0, 0);
}
```

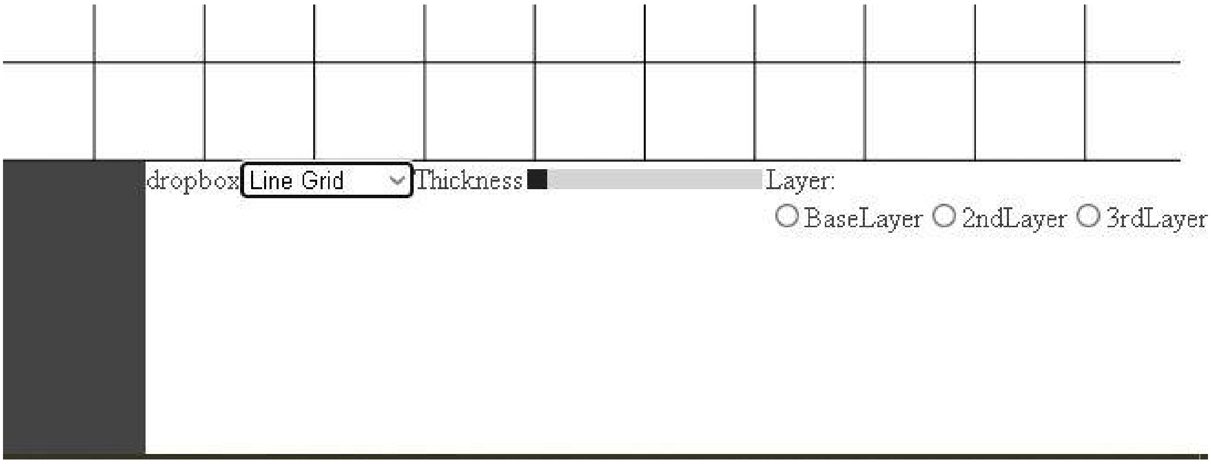
Figure 2.3: Barney's code for the scratch card.



```
}

function setup() {
  createCanvas(600, 600);
  topLayer = createGraphics(width, height);

  topLayer.background(0,0,0,0);
  topLayer.textSize(50);
  topLayer.textAlign(CENTER);
  topLayer.text("SCRATCH ME top", width/2, height/2);

  topLayer.imageMode(CENTER);
  topLayer.strokeWeight(40);
  topLayer.blendMode(REMOVE);

  topperLayer = createGraphics(width, height);

  topperLayer.background(0,0,0,0);
  topperLayer.textSize(50);
  topperLayer.textAlign(CENTER);
  topperLayer.text("SCRATCH ME topper", width/2, height/2);

  topperLayer.imageMode(CENTER);
  topperLayer.strokeWeight(40);

    topmostLayer = createGraphics(width, height);

  topmostLayer.background(0,0,0,0);
  topmostLayer.textSize(50);
  topmostLayer.textAlign(CENTER);
  topmostLayer.text("SCRATCH ME topmost", width/2, height/2);

  topmostLayer.imageMode(CENTER);
  topmostLayer.strokeWeight(40);
  topmostLayer.blendMode(REMOVE);

  // topperLayer.blendMode(REMOVE);
}

```

```
function draw() {
  image(img, 0, 0, width, height);

  if(keyIsPressed && key == 'q'){
    topperLayer.blendMode(REMOVE);
    topperLayer.line(pmouseX, pmouseY, mouseX, mouseY);
  }

  if(keyIsPressed && key == 'w') {
    topperLayer.blendMode(NORMAL);
    topperLayer.line(pmouseX, pmouseY, mouseX, mouseY);
    // topLayer.image(stamp, mouseX, mouseY, stamp.width * 2, stamp.height*2);

  }
  if(keyIsPressed && key == 'e'){
    topLayer.blendMode(NORMAL);
    topLayer.line(pmouseX, pmouseY, mouseX, mouseY);

  }
  if(keyIsPressed && key == 'r'){
    topLayer.blendMode(REMOVE);
    topLayer.line(pmouseX, pmouseY, mouseX, mouseY);
  }

  if(keyIsPressed && key == 't'){
    topmostLayer.blendMode(REMOVE);
    topmostLayer.line(pmouseX, pmouseY, mouseX, mouseY);
  }

  if(keyIsPressed && key == 'y') {
    topmostLayer.blendMode(NORMAL);
    topmostLayer.line(pmouseX, pmouseY, mouseX, mouseY);
    // topLayer.image(stamp, mouseX, mouseY, stamp.width * 2, stamp.height*2);
  }

  image(topLayer, 0, 0);
  image(topperLayer,0,0);
```

Figure 2.4: My codes to make Drawing Layers.



dropbox Line Grid Thickness

Layer:

○ BaseLayer ○ 2ndLayer ○ 3rdLayer

Figure 2.5: The current UI of the helper functions (6th July 2023)

CHAPTER 3: How I planned this Project

3.1 Gantt Chart (try to make the table cause for some reason the coursera one is not just in pdf)

Figure 3.1 is my Gantt Chart. The coloured outlined squares represent the days I allocated for the parts, while the coloured shaded squares represent what I worked on that day. The expected completion date for the three main extensions if I decided to work on this project daily is generally before week 12.

I made my Gantt Chart with the consideration that I was rusty with coding in Javascript and might face issues regarding the Blur Tool. Hence, I decided to code the helper functions first so I can get used to the "grammar" of Javascript again.

As for the Blur Tool, I will attempt to implement it later as I know it is not impossible, just requires hours of testing. This will allow me to sufficiently prepare this project for the mid-term submission while giving me more time to revise for the other subjects. After submitting all of my deliverables on week 13, I will start working on the last extension since there will be a lot of time for me to code.

For the interface of the helper functions, it is a very low-priority task so through the Agile method, I will work on it last.

coursera version:

Figure 3.1 is my Gantt Chart. The coloured outlined squares represent the days I allocated for the parts, while the coloured shaded squares represent what I worked on that day. The expected completion date for the three main extensions if I decided to work on this project daily is generally before week 12.

I made my Gantt Chart with the consideration that I was rusty with coding in Javascript and might face issues regarding the Blur Tool. Hence, I decided to code the helper functions first so I can get used to the "grammar" of Javascript again.

As for the Blur Tool, I will attempt to implement it later as I know it is not impossible, it just requires hours of testing. This will allow me to sufficiently prepare this project for the mid-term submission while giving me more time to revise for the other subjects. After submitting all of my deliverables on week 13, I will start working on the last extension since there will be a lot of time for me to code.

For the interface of the helper functions, it is a very low-priority task so through the Agile method, I will work on it last.

word count: 199

Figure 3.1: my project’s Gantt Chart.

https://drive.google.com/file/d/1dIQfcWpn8eYuy-ahw6JsSeouRmMvleHM/view?usp=sharing

.

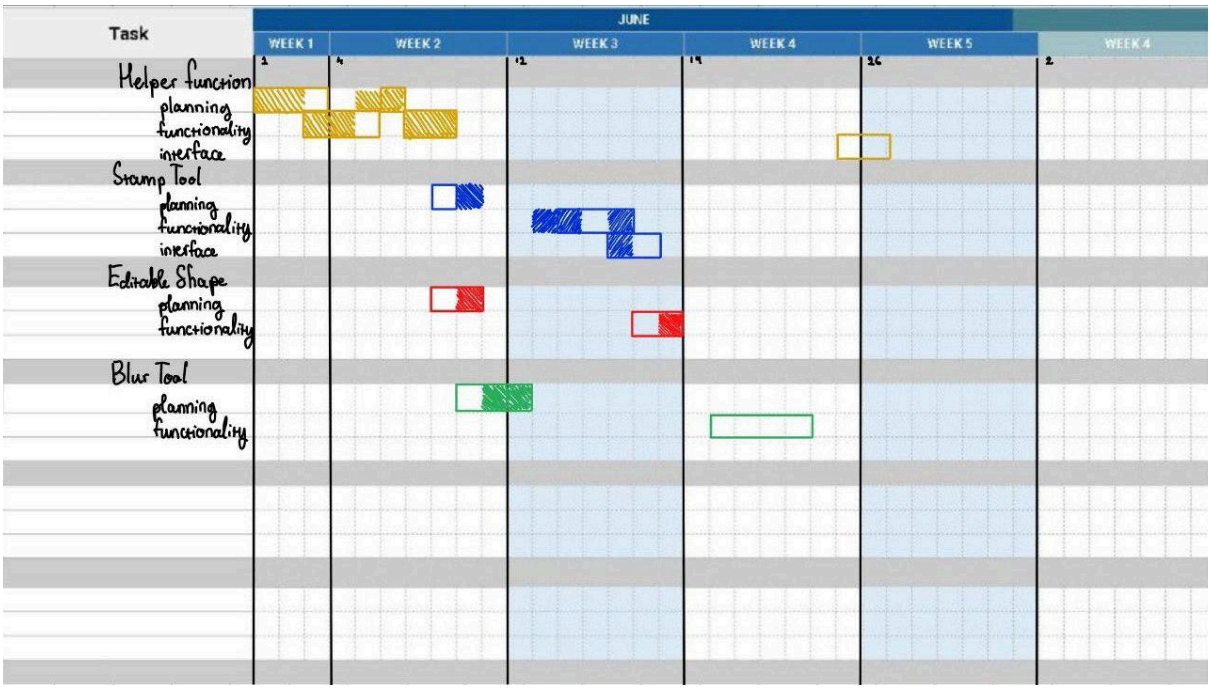


Figure 3.1: my project’s Gantt Chart for June

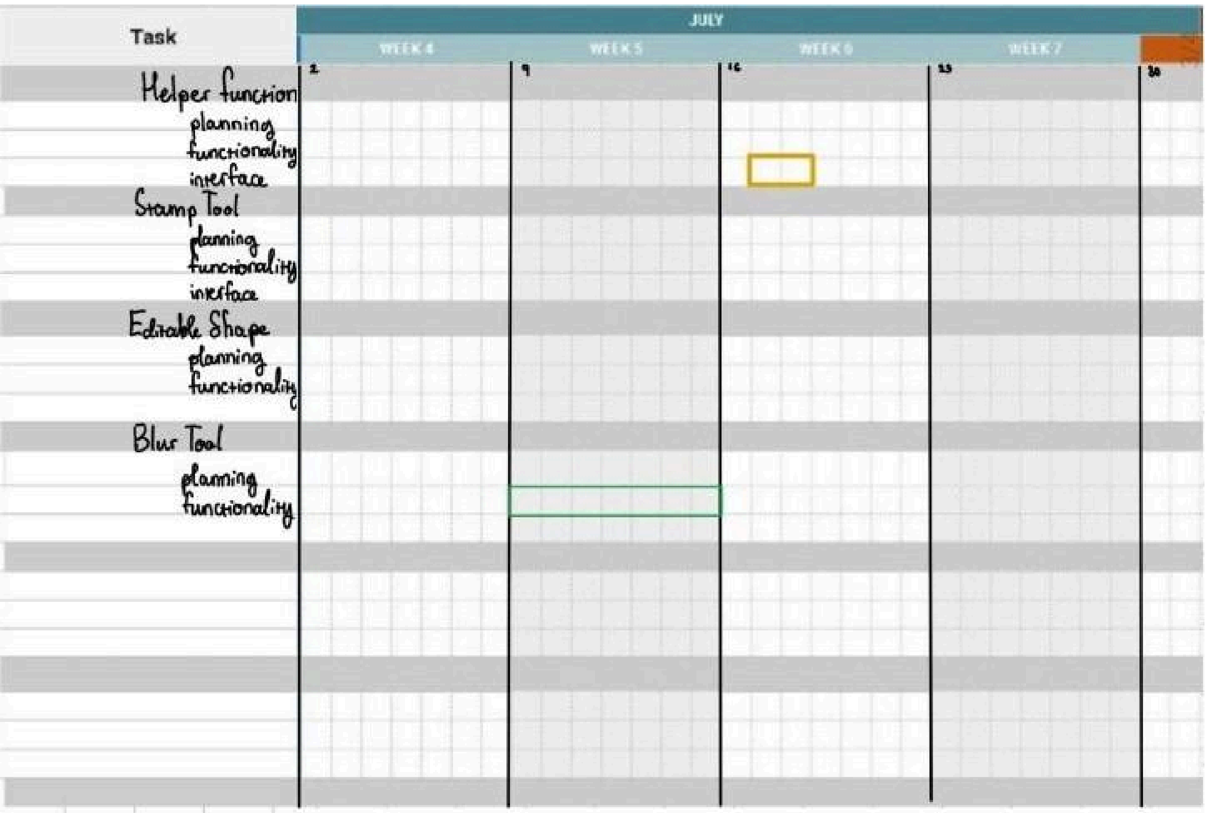


Figure 3.2: my project’s Gantt Chart for July

References

Zhang, W. and Zhang, W., 2017. Special A — Avetics. [online]. Figure 2.2.1. Avetics. Available at: <https://www.avetics.com/blog/2017/3/22/special-a-1> [Accessed 20 February 2022].

Volarious. n.d. Volarious | Gears to Enhance DJI Drones. [online]. Figure 2.2.2 Available at: <https://www.volarious.com/?gclid=CjwKCAiAx8KQBhAGEiwAD3EiPwp2h2kkuJMpA5uZS7MtPUPwnDt8gZq8ByRJKGuHoeNWEpvtBM7qSRoCEA0QAvD_BwE> [Accessed 20 February 2022].

Volarious. n.d. Volarious | Gears to Enhance DJI Drones. [online]. Figure 2.2.3 Available at: <https://www.volarious.com/?gclid=CjwKCAiAx8KQBhAGEiwAD3EiPwp2h2kkuJMpA5uZS7MtPUPwnDt8gZq8ByRJKGuHoeNWEpvtBM7qSRoCEA0QAvD_BwE> [Accessed 20 February 2022].

Prusa Knowledgebase. 2021. How to replace a PTFE tube (MK3/MK2.5). [online]. Figure 2.3.1 Available at: <https://help.prusa3d.com/en/guide/how-to-replace-a-ptfe-tube-mk3-mk2-5_17360/> [Accessed 20 February 2022].

https://www.ubuy.com.sg. n.d. Official Creality Ender 3 3D Printer Fully Open Source with Resume Printing Function DIY 3D Printers Printing Size 220x220x250mm. Figure 3.1.1. [online] Available at: <https://www.ubuy.com.sg/product/OKPU78-official-creality-ender-3-3d-printer-fully-open-source-with-resume-printing-function-diy-3d-printers> [Accessed 8 February 2022].

Flashforge.com. n.d. Flashforge Creator 3 FDM 3D Printer Large Build Volume - FlashForge. Figure 3.1.3. [online] Available at: <https://www.flashforge.com/product-detail/flashforge-creator-3-fdm-3d-printer> [Accessed 8 February 2022].

Prusa3d.com. n.d. Original Prusa i3 MK3S+ | Original Prusa 3D printers directly from Josef Prusa. Figure 3.1.2. [online] Available at: <https://www.prusa3d.com/category/original-prusa-i3-mk3s/> [Accessed 8 February 2022].

SainSmart. 2020. How to Level Your Ender-3 V2's Print Bed. Figure 3.1.4. [online] Available at: <https://docs.sainsmart.com/article/0bwiehyjlg-ender-3-v-2-bed-leveling> [Accessed 8 February 2022].

Stříteský, O., 2019. Does your newly assembled Original Prusa i3 MK3 print the best it can? - Prusa Printers. [online]. Figure 3.1.5 Prusa Printers. Available at: <https://blog.prusaprinters.org/does-your-freshly-assembled-original-prusa-i3-mk3-print-as-the-best-it-can_29445/> [Accessed 20 February 2022].

Wigglesworth, T., 2020. CATIA STEP Core Interface ST1 Archives - Inceptra. Figure 3.2.2. [online] Inceptra. Available at: <https://www.inceptra.com/tag/catia-step-core-interface-st1/> [Accessed 8 February 2022].

Therrien, C., 2021. SOLIDWORKS 3D CAD 2022 is Here: Discover the Top Enhancements Now. [online] The SOLIDWORKS Blog. Figure 3.2.3. Available at: <https://blogs.solidworks.com/solidworksblog/2021/09/top-enhancements-in-solidworks-3d-cad-2022.html> [Accessed 8 February 2022].

Walmart.com. n.d. Business Hour Open Closed Sign- Bundle of Office Hours Sign Will Return Clock with Suction Cups for Door Window Businesses Stores Restaurants Bars Retail Barbershop Salon Shops (Open/Cl. [online]. Figure 3.2.14 Available at: <https://www.walmart.com/ip/Business-Hour-Open-Closed-Sign-Bundle-Office-Hours-Sign-Will-Return-

Clock-Suction-Cups-Door-Window-Businesses-Stores-Restaurants-Bars-Retail-Barbersh/566566589> [Accessed 20 February 2022].