

INTRODUCTION TO PROGRAMMING II

PROJECT LOG

PROJECT TITLE: My Drawing App (Project Log 1)

TOPIC: Topic 7 Extending the apps

1

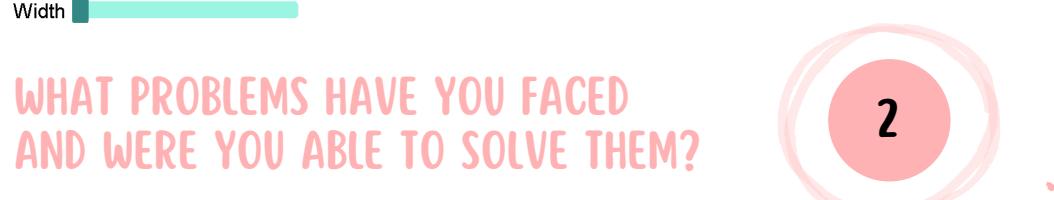
WHAT PROGRESS HAVE YOU MADE THIS TOPIC?

Essential changes were made to two tools (stickerTool and brabackgroundColor). For the first tool "StickerTool" two sliders were implemented, one that would be used to change the size of the sticker and the other to edit the rotation of the sticker. In addition, labels were added to the side of each slider to describe what its function is. A constructor function "this.sticker()" was created, which would take as parameters an initial position (x-y axis), a size (width-length) and finally an angle. With all this you could modify the rotation position of the sticker and finally draw it on the canvas. Likewise, this function would be called in "this.draw()" following the position parameters of the stickers, the chosen size and the chosen angle in the respective previously mentioned slider.

For the second tool "BackColorTool" three functions "this.colorRGB()", "this.colorHSB()", "this.colorHSL()" were implemented. Each of them consists of three sliders for each color mode (RGB-HSB-HSL). In addition, labels were created that will accompany each slider. Three buttons were implemented that would allow switching between color modes for the background of the canvas (RGB-HSB-HSL). Each button allows you to display the three sliders that correspond to each color mode value R-G-B, H-S-B, H-S-L. Likewise, functions such as show() and hide() were used so that the respective sliders are shown on the canvas while others that are not necessary are hidden.

Likewise, modifications were made to some tools such as "crayonTool", "hatchingTool", "bladesTool"... For which sliders and corresponding buttons were implemented that would allow changing the opacity of the drawing, the size, the color, etc.

Finally, other cursors were implemented for each of these tools described above.



2

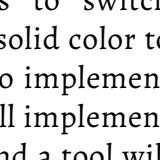
WHAT PROBLEMS HAVE YOU FACED AND WERE YOU ABLE TO SOLVE THEM?

For the "BackColorTool()" tool, when implementing the corresponding buttons for each color mode, I did not know how to do it so that each time one of them is pressed, only the three corresponding sliders are displayed without affecting the other sliders and buttons, so look in the "p5.js reference" and choose to use the hide(), show() functions. Also use booleans that will allow you to recognize which color mode is being used to verify that sliders and buttons should be hidden or not. Finally, all the bugs found in the "angleBrushTool()", "arcsTool()", "hatchingTool()" tools were fixed (change size, line color, opacity, etc).



3

WHAT ARE YOU PLANNING TO DO OVER THE NEXT FEW WEEKS?



I would like to implement a functionality in the "shapeBrushTool()", "autoRainbowTool()" and "pointsTool()" tools, which allows to switch between color modes, in such a way that you can change from solid color to multicolor. Also, for the "shapeBrushTool()" tool I would like to implement more shapes (circular, rectangular, etc). On the other hand, I will implement a slider to change the size of the star in the "starTool()" tool. And a tool will be developed that allows you to cut, copy and paste one part of the canvas to another. Likewise, the present bugs that still exist in these tools will be fixed and the interface will be aesthetically improved.

4

ARE YOU ON TARGET TO SUCCESSFULLY COMPLETE YOUR PROJECT? IF YOU AREN'T ON TARGET, HOW WILL YOU ADDRESS THE ISSUE?

Really everything is going pretty well for now regarding the implementation of planned objectives, even so, I should rush some ideas since the planned time for everything could be short. For now, some tools and details need to be implemented. To do this, I will better organize my time and order my ideas better.



INTRODUCTION TO PROGRAMMING II

PROJECT LOG

PROJECT TITLE: My Drawing App (Project Log 2)

TOPIC: Topic 8 Callbacks

1

WHAT PROGRESS HAVE YOU MADE THIS TOPIC?

Made changes to the "autoRainbowTool()", "shapeBrushTool" and "pointsTool" tools. For the first tool "autoRainbowTool()" a button was implemented that will allow changing the color mode of the lines, which by default are multicolored thanks to the HSB color mode. However, if you don't want the lines to be multicolored, then this new button will change the mode from HSB to RGB, so now the line is just one color. In addition, two sliders that correspond to the hue and saturation of the lines were implemented. Which will only be displayed when the lines are in multicolor or rainbow mode (HSB). Finally, I added two tags to describe the functions of each slider. For the second tool "shapeBrushTool()" a shape selector was added, in such a way that not only adjacent circles can be drawn, but also rectangles. In addition, a checkbox has been added that allows changing color modes between HSB to RGB and vice versa, which will allow you to choose between multicolor or single color shapes. A slider was added to modify the size of the geometric figure and finally a checkbox to verify if you want the figure to have a border or not.

For the third tool "pointsTool()" a checkbox was implemented that allows you to switch between color modes, in this way the points to be drawn will be multicolored or only one color.

On the other hand, two tools "cutPasteTool()" and "copyPasteTool" were developed, which serve to cut-paste and copy-paste respectively one part of the canvas to another. The first "cutPasteTool()" works on the basis of a flag (selectMode: 0 = inactive, 1 = selecting area, 2 = pasting area). When the "Select Area" button is pressed, the indicator changes to 1 and the label changes to "Cut", which means that with the help of the "keyPressed()" method, by pressing the CTRL + Left Click key of the mouse you can select the initial coordinates of the area to be selected while with the help of the "mouseDragged()" method the width and length measurements of our area to be selected will be recognized from the initial coordinates. When we already have our area selected we can press the "Cut" button, which will change the indicator to 2 and the label to "end Paste" in order to obtain the selected pixels and draw a white rectangle that simulates the removal of these pixels over the area selected. Finally, the CTRL key will be pressed on the position in which we want to paste this selected area, when we finish we will press "end Paste" and the indicator will return to 0 to repeat this whole process again. The "copyPasteTool()" tool works the same way except that it doesn't draw a white rectangle when selecting the area, which simulates that you are only copying it, not cutting it.

On the other hand, the bugs still present in the aforementioned tools were fixed.

2

WHAT PROBLEMS HAVE YOU FACED AND WERE YOU ABLE TO SOLVE THEM?

When designing the tool "cutPasteTool()" I implemented a function called "keyPressed" which would detect when the user presses the CTRL key on their keyboard. Which would determine the initial coordinates of the area to be selected. However, it didn't work and the starting coordinates were still (0,0). So I used the breakpoint method in the browser console to see the error in more depth. That's when I realized that the method was not being expected because I needed to add a function in the "sketch.js" file which would check if a selected tool in the toolbox has a method called "keyPressed". So if that were the case, the function calls the keyPressed() method and passes keyCode as a parameter. So I implemented this function in the "sketch.js" file and it immediately worked.

```
this.keyPressed = function()
{
    if(keyCode == 17){
        ctrl = true;
    }
    if(ctrl == true)
    {
        if(selectMode == 1)
        {
            selectedArea.x = mouseX;
            selectedArea.y = mouseY;
        }
        else if(selectMode == 2)
        {
            image(selectedPixels,
                mouseX - selectedArea.w/2,
                mouseY - selectedArea.h/2);
        }
    }
}
```

```
function keyPressed(keyCode){
    if (toolbox.selectedTool.hasOwnProperty("keyPressed")){
        toolbox.selectedTool.keyPressed(keyCode)
    }
}
```

keyPressed()

Description

The `keyPressed()` function is called once every time a key is pressed. The keyCode for the key that was pressed is stored in the `keyCode` variable.

3

WHAT ARE YOU PLANNING TO DO OVER THE NEXT FEW WEEKS?

I would like to implement background music (chill-electro-blues). In addition, I would also like to implement two new tools, the first to apply filters to the canvas with the help of the filter() method (THRESHOLD-GRAY-OPAQUE-INVERT-POSTERIZE-BLUR-ERODE-DILATE) and the second to draw text on the canvas with the help of the text(), keyReleased() and keyTyped() methods. Also, I would like to user test the app with the help of my friends and family to fix bugs that are still present.

ARE YOU ON TARGET TO SUCCESSFULLY COMPLETE YOUR PROJECT? IF YOU AREN'T ON TARGET, HOW WILL YOU ADDRESS THE ISSUE?

Yes, I plan to make three last tools and add some details, soon I will only make two ("filtersTool()" and "textTool()") since I am still researching to develop the third one in the best way. Lastly, my essential goal is to try to fix all the bugs still present in the application. For which, I will dedicate more time exclusively to focus on it.

4

INTRODUCTION TO PROGRAMMING II

PROJECT LOG

PROJECT TITLE: My Drawing App (Project Log 3)

TOPIC: Topic 9 Testing for stability

1

WHAT PROGRESS HAVE YOU MADE THIS TOPIC?

Background music and two new tools "textTool()" and "filtersTool()" have been implemented. We will begin by describing how background music works. The preload() function was used so that the necessary resources, in this case mp3 audios, are loaded before the execution of the main code begins. Then, a "PLAY MUSIC" button was created, which will allow displaying three other buttons and immediately change its label to "CANCEL MUSIC", in case we want the music to pause and the three buttons to be hidden again. For these three buttons, variables (sound1, sound2, sound3) were used, which indicate (blues-electro-chill) respectively, these allow pausing or playing the sound file corresponding to each button. Also, I named the variables "s1", "s2" and "s3" to check if the corresponding sound file is being played or not. In this way, when one audio is being reproduced, the other two cannot be played at the same time, since this would cause auditory discomfort for the user. Likewise, when one of the three buttons is pressed, its label will change and an arrow "↓" will appear next to the name to indicate that the corresponding sound file is being played.

Now, the two implemented tools will be described:

"filtersTool()" is a tool that allows applying filters to the canvas, it consists of an array where the names of the filters are stored. With the help of the forEach() method, it will be iterated over each of the members of the array, a button will be created with ID and LABEL of its corresponding name. Next, a constant "namesFilters" was declared, which is an object with several properties, each of which has a string key that represents the name of a filter and a value that represents a function to apply to each filter. Then, with the help of the "Object.keys()" method, we will get all the keys (filter names) of the previously created object "namesFilters". Then, with the help of the forEach() method, it will be iterated over each of these keys and the ID that is equal to the name of each of these filters will be identified. Then, when one of them is clicked, the function corresponding to that filter will be executed (gray, threshold, opaque, invert, posterize, dilate, blur, sepia).

"textTool()" is a tool that allows to implement text to the canvas. To do this, two ways were implemented, the first works when the "click to type" button is pressed, which allows the text to be seen directly on the canvas while we write, to elaborate this the text() and keyTyped() method was used. In such a way that when a key is pressed, the keyTyped function will identify it and add it to the string content = "" then with the help of the text() method the string "content" will be drawn on the canvas. In addition, there are two buttons that will allow us to convert the text to upper or lower case. On the other hand, the position of this text on the canvas will depend on the positions of the left mouse click on the canvas. When the text is on the canvas and we have chosen our final position with the help of the left mouse click, we can press the "finish" button to conclude and start from scratch with a new text. If we do not want to continue writing in this way, we can press the "finish Click to type" button. The second way works based on an "Input", which returns the text written in it as a value and is displayed on the canvas. The position will depend on where you left click on the canvas. Finally, a slider was implemented to modify the size of the text. Important: (the colors of the text can also be modified with the help of the color selectors at the bottom of the canvas).

Important: the application was tested and various pending errors were recognized. Which have already been fixed.

2

WHAT PROBLEMS HAVE YOU FACED AND WERE YOU ABLE TO SOLVE THEM?

At the time of implementing background music, I realized that when I pressed one button followed by another, both songs would sound and that would make the user auditory uncomfortable. Therefore I decided to implement boolean variables (s1,s2,s3) that would allow to recognize if the sounds were being played or not. Also, I used the name.isPlaying() method to verify that if the audio was already playing and its respective button was pressed again, then this audio would be paused. On the other hand, when I implemented the text tool "textTool()" I realized that when the user pressed the "click to type" button and the string would be reflected on the canvas, the BACKSPACE key could not be used to remove the last character of the reflected string, so I investigated and found the content.substring(a,b) method, which holds all but the last character of the string. That means that every time the user presses BACKSPACE the last character would be removed from the string and only the others would be reflected.

3

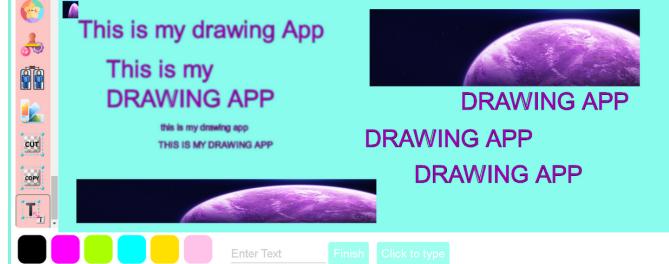
WHAT ARE YOU PLANNING TO DO OVER THE NEXT FEW WEEKS?

Finally, I plan to add two more tools next week, the first to clone a canvas space to another side and the second will be a Spirograph simulation, which will allow generating geometric designers. And the last week will only be to test my application and fix bugs if there are still any.

4

ARE YOU ON TARGET TO SUCCESSFULLY COMPLETE YOUR PROJECT? IF YOU AREN'T ON TARGET, HOW WILL YOU ADDRESS THE ISSUE?

Yes, I have in mind how to make the last two tools, I feel that it will not take much time and in the last week I will be able to test the finished application in greater detail. In this way, I will see if there are still pending bugs or not.



INTRODUCTION TO PROGRAMMING II

PROJECT LOG

PROJECT TITLE: My Drawing App (Project Log 4)

TOPIC: Topic 10 Testing with users

1

WHAT PROGRESS HAVE YOU MADE THIS TOPIC?

During the first week two new tools were implemented: "cloneShapeTool()" and "geometricDesignTool()".

"cloneShapeTool()" is a tool that allows the user to clone a selected area of the canvas and draw it to a new location. To do this, the methods mousePressed(), mouseReleased(), and keyPressed() were used to verify that the canvas was clicked, stopped clicking, and the CTRL key was pressed. This way, when you click on the canvas and press the CTRL key at the same time, the current coordinates (x,y) will be saved as the initial starting point to start cloning. The user will then be able to click on another position on the canvas, which will be the target position for the cloned image. If the mouse is pressed, the cloned image will be drawn with the help of the "copy()" method, copying the pixels from the original area to the new area. In addition, a rectangle will be drawn from the original area that will function as an indicator to know which part of the canvas is being cloned. Finally, a slider was implemented that allows you to change the size of the area to be cloned.

"geometricDesignTool()" is a tool that allows to simulate the figure of a spirograph. When the user clicks on the canvas, prevMouseX and prevMouseY are set to the current mouse position values. The initial angle is calculated with the InitialAngle() calculation method. And symmetric points are computed with the symmetricFunc() method based on the angle and position of the original point. A line is drawn between the current and previous mouse positions, and another line is drawn between the symmetric points on either side of the symmetry line. Likewise, different buttons were implemented that allow switching between symmetry options (9,18,36,54). The color and width of the lines can also be modified.

Finally, the second (final) week the application was tested on third parties (family and friends), for this a quiz was carried out with key questions to find out what changes were still necessary, there was also a section in which it was possible to explain what advice and recommendations were given. Bugs were satisfactorily fixed and small details implemented.

2

WHAT PROBLEMS HAVE YOU FACED AND WERE YOU ABLE TO SOLVE THEM?

Fixed some bugs found by users who tested the application, errors which were found when changing the width or color of the lines. Also, there were some tools that didn't have the unselectTool() function well implemented, this was fixed immediately.

3

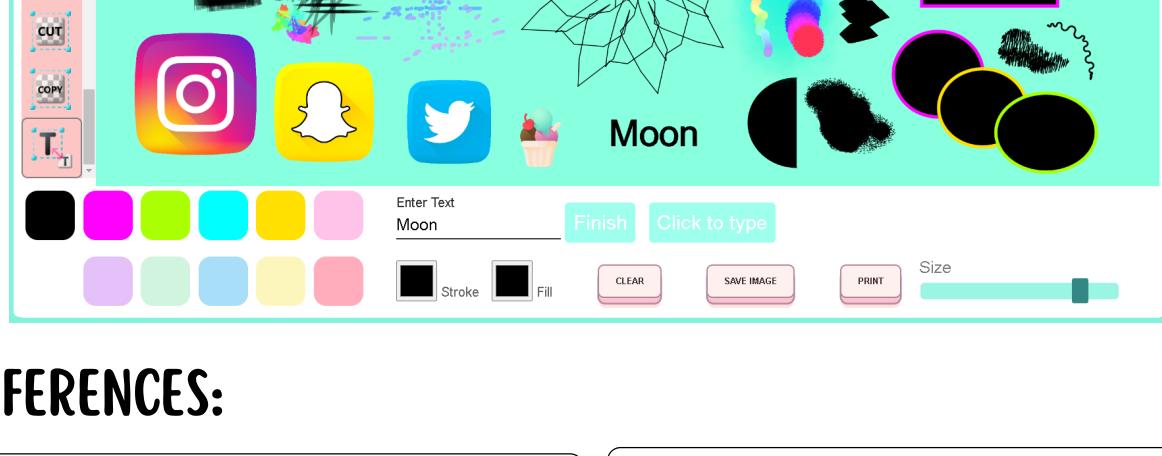
WHAT ARE YOU PLANNING TO DO OVER THE NEXT FEW WEEKS?

FINAL WEEK :)

4

ARE YOU ON TARGET TO SUCCESSFULLY COMPLETE YOUR PROJECT? IF YOU AREN'T ON TARGET, HOW WILL YOU ADDRESS THE ISSUE?

Yes, all the expected tools were implemented, the application was also aesthetically improved, background music was added, and finally it was put to the test.



REFERENCES:

- <https://uiverse.io/> ×
- https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/Useful_string_methods ×
- https://www.w3schools.com/js/js_htmldom_document.asp ×
- <https://p5js.org/reference/> ×
- <https://www.geeksforgeeks.org/p5-js-dom-and-rendering-complete-reference/> ×

- <https://www.flaticon.es/> ×
- <https://coolcolors.co/> ×
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/substring ×

CLICK ON EACH LINK

Moo-Dark