CA1

Creative Coding

Aimee Redmond

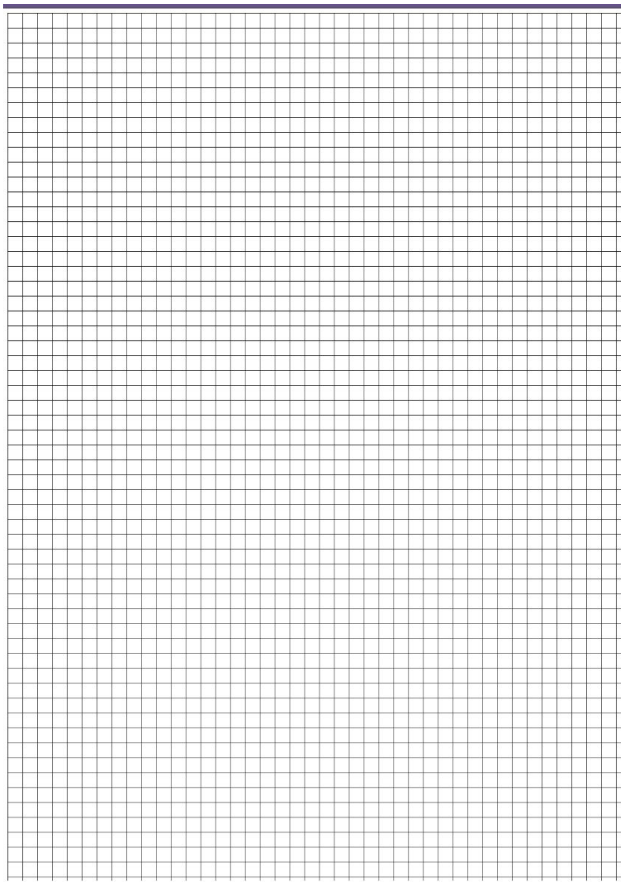
N00152592

Introduction

The aim of this project was to create a number of experimental colour images using code developed in p5.js and that is also controlled by a number of input parameters. For my project I chose to experiment with colour interpolation on a grid that depends on the distance from the mouse position to the edges of the canvas, Allowing the user to control the interpolation by changing the colour hue with a mouse click.

Step 1

In Step 1, I created a grid of boxes that filled the canvas. I did this by creating variables that represent the amount of space between each rectangle and then by creating a loop that goes along the X and Y axis filling it with the rectangles. I also set the canvas size and made the colour mode HSB.

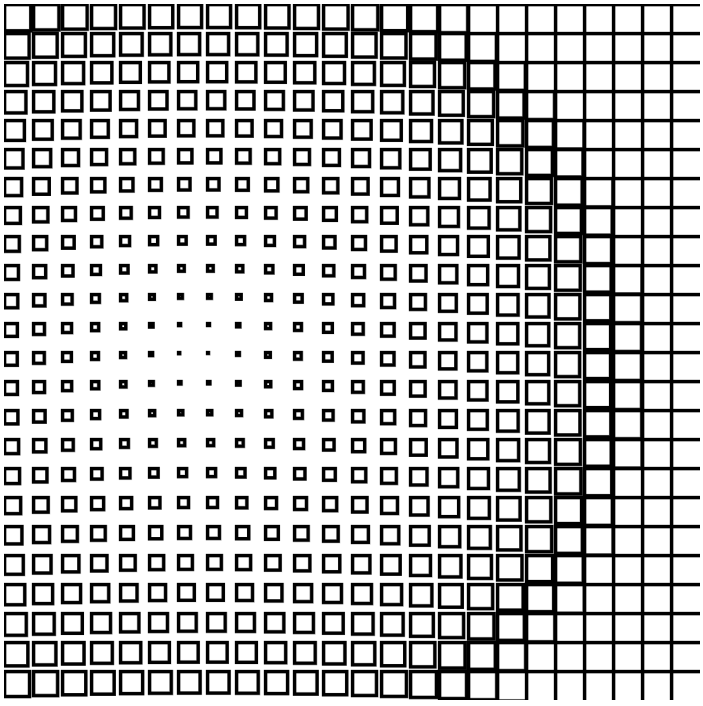


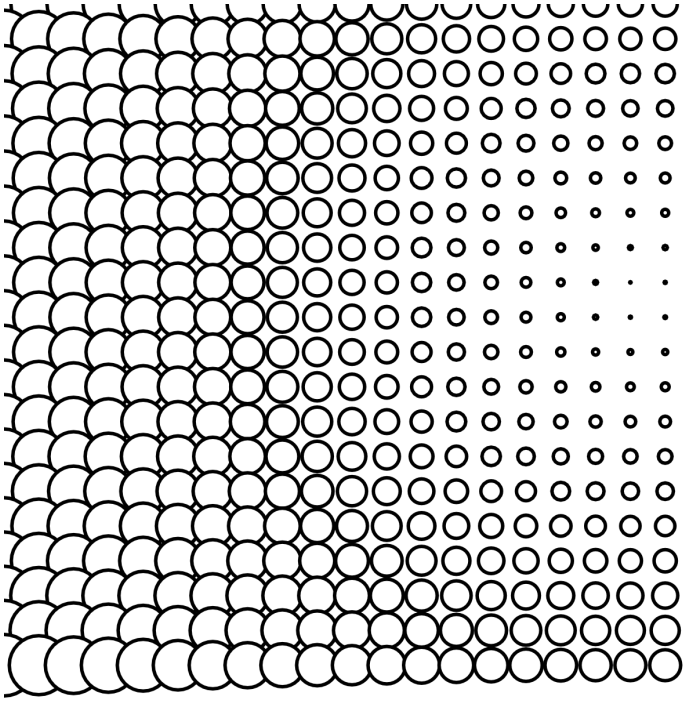
Step 2

In Step 2, I added in the mouse interaction on the canvas which affects the diameter of the circle around the mouse. I did this by creating a variable called Max Distance and setting it to 600, changing this variable will affect the size of the circle from the mouse point.

In the Grid loop I created a variable called diameter which uses the distance function from P5 to get the distance between 2 points on the canvas. I then took that variable, divided it by the max distance variable and multiplied it by 60, this changes the reach the mouse has on the canvas.

Then in a push and pop function I translate the mouse diameter onto the grid and add in the ellipses. I chose to change the rectangles to ellipses as I found that it had a nicer effect of the canvas. By making the width and height of the ellipses equal to the diameter, this helps them change with the mouse.



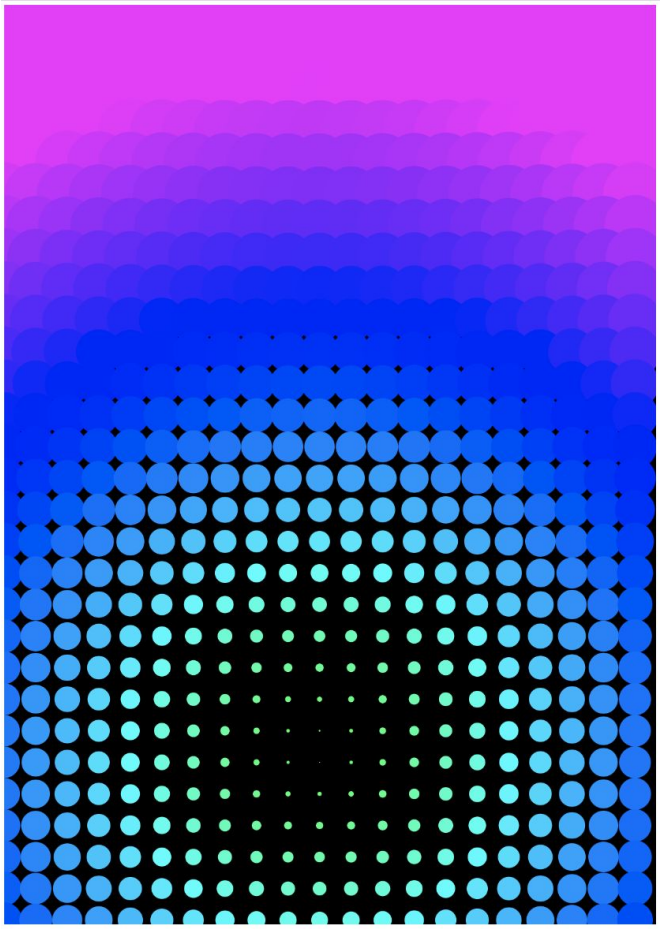


Step 3

In Step 3, I added in color interpolation from a mouse point on the canvas. This is done by using the lerpColor function in P5. I created Start colour and End colour variables and used the colour function to hard code colours in HSB. I then created a new distance variable which also is using the same distance function to get the distance between 2 points on the canvas. I then had to create the variable amount, this maps the distance from 0 to the width of the canvas from 0 to 1.

Using the lerpColor function I can then interpolate between the start and end colour variables by the amount variable. The interpolation will happen from the mouse position and go out to the edges of the canvas.

I also removed the push and pop methods as I found that by adding in the gridX and gridY into the ellipse is achieved the same goal.



Step 4

In the Final Step, I added in the mouse functions for the user to change the Hue of the colour interpolation. I first created global variables, startHue and endHue, and set them to 2 colours. I then added these variables into the hue of the start colour and end colour.

In a mousePressed function I then mapped the startHue and endHue variables to the mouses X and Y position and then mapped the width and height to a number between 0 and 360. This way when the user clicks the mouse anywhere on the canvas 2 different hues will be selected and will be interpolated.

Lastly, I added in the keyPressed functions to save the canvas as a png and to create and adobe ase file to save the colours as a swatch.