GAD-405-Creative Coding Report

*Generative Art*

My GAD405 Generative Art Project is an array of grid squares made to mimic a low resolution screen, various squares bounce around the grid changing colour every time they bounce and leaving a trail behind them which paints the screen in a grid based pattern over time.

Inspirations

As the Art style would suggest the main inspirations for my work here are older pieces of work, mainly visualisers such as Psychedelia by Jeff Minter (LlamaSoft), this was one of my main inspirations because of the low resolution style of the art as it was originally designed for the commodore 64 home computer so had a very low screen resolution, Psychedelia was designed as an audio visualiser but I used it as a reference point for my work as I liked the low resolution block colour style of it.

reference[1]

Another inspiration I had was also from LlamaSoft, called “Trip-A-Tron”, it was another audio visualiser but a bit later than Psychedelia and more high resolution, this gave me the idea of keeping the trails behind the pixels as they move around the grid as this would build up a more complete piece of art work as time went on, if I hadn't done this the art piece would just be a few coloured squares bouncing around a square canvas and it wouldn't reveal the grid array over time.

reference[2]

I was inspired in part by a DOS audio visualiser program, I cant find the name of the program so I only have a short clip of it working, this program was where I got the idea to use random colours on the bounces, combined with the trails this would create an array of colours over time over time instead of one flat colour that slowly fills the entire canvas.

reference[3]

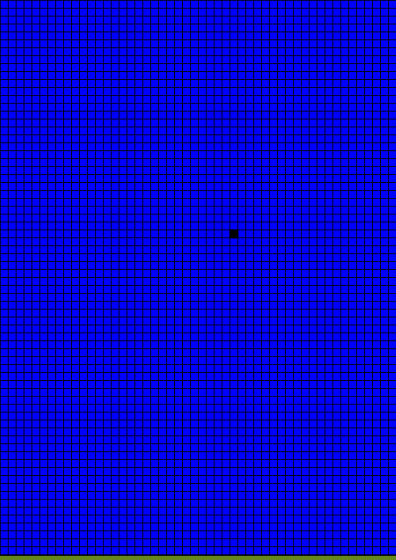
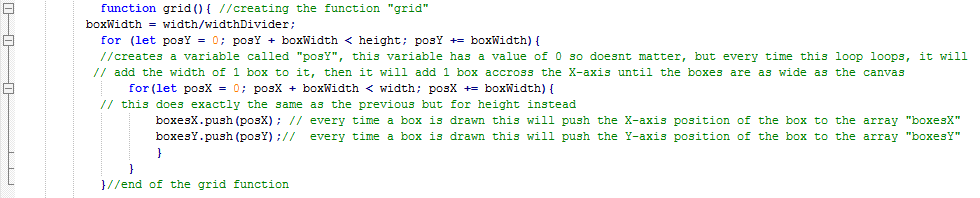
Development Process

To develop my generative art piece I used p5js as it uses Javascript so is easy to pick up and can be embedded in HTML to be displayed on a web page. I also used Github for version control with my work.

*Defining the Grid*

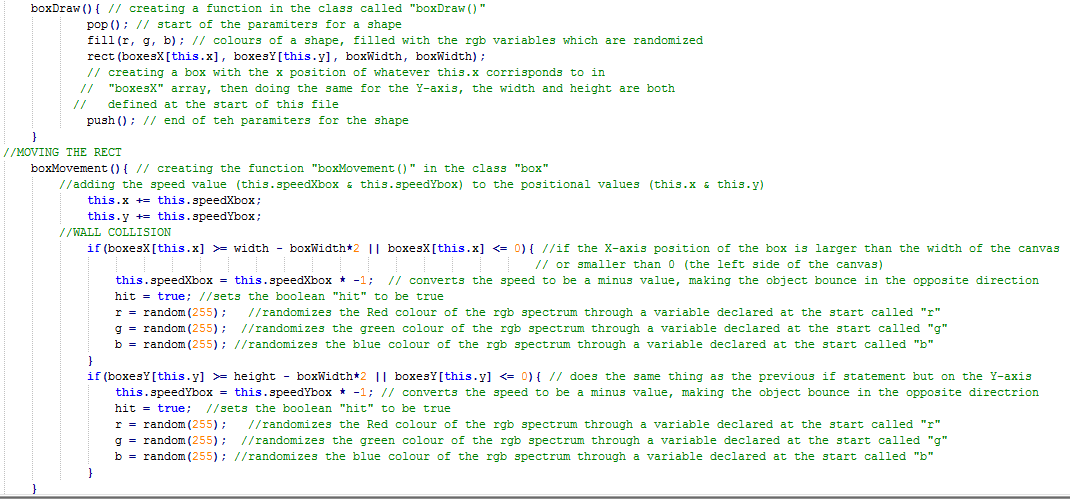
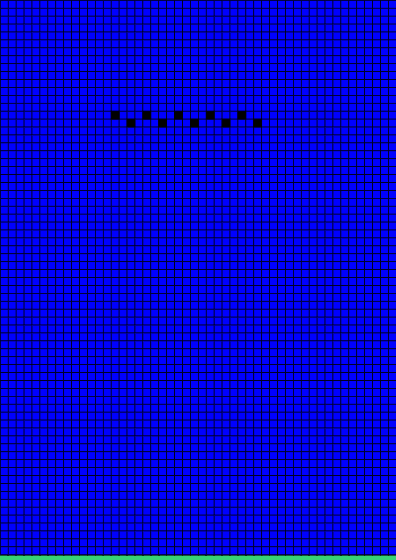
To start off with my work the first thing I knew I wanted to do was use a low resolution pixel art style, to achieve this I decided I would create an array of variables to store the x co-ordinates and do the same for the y co-ordinates, to do this I created a new function called “grid()” this function was initially called every frame in the “draw()” function, I created a variable which I would use as the height and the width of each square on the grid and called this “boxWidth”, the grid function

starts with a for loop that creates a new variable with a value of 0, every time the loop happens, the value of “boxWidth” will be added to this variable until the value has reached the length of the canvas, the same thing would happen on the y axis with a second nested loop but moving in the y axis instead. Each time the loop runs the program pushes the current position of the x and y variables X axis array and the Y axis array respectively, originally at this point the program would also draw a “rect” with these dimensions and positions but this was very computationally intensive to run so I took out the shapes and put them in their own function meaning I could run “grid” once at the start

****

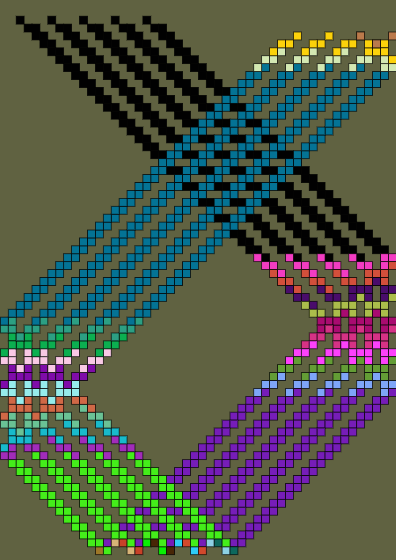
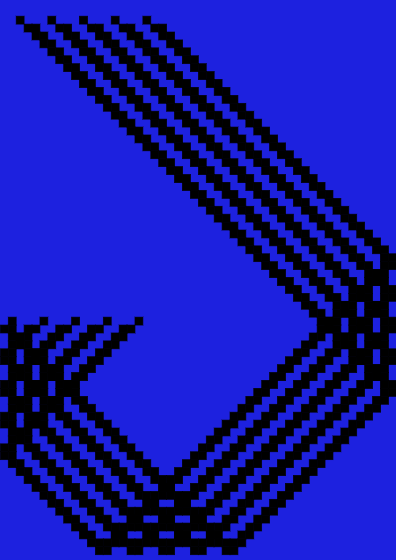
*Creating the moving boxes*

After setting up the grid as an array, I made a class called “box” this would allow me to create as many boxes as I want that all had the same properties, but I could just pass different starting positions too them. The “box” class had 2 functions , “boxMovement” and “boxDraw”, boxMovement takes in the position and speed variables and makes sure that they never exceed the width of the canvas or go lower than zero, and using if statements the speed will be reversed into negative every time the box reaches the width of the canvas meaning the blocks will bounce. The second function simply took the information from the first function and fed it into a “rect” command, drawing the shape with the correct dimensions. I had first made the box a function but quickly realised this method wouldn’t let me create multiple boxes so I changed to a class for this.

****

*Making it look better*

when I decided to remove the background grid, it had the side effect of leaving a trail behind each square which looked similar to some of the work I used as inspiration so I decided to keep it, but I also added a random colour change every time a pixel bounced of a wall in the box class, this meant the art had a more collage look and was far more visually appealing and interesting to look at.

****

Problems & Improvements

The main problem that effected the final product was I couldn't get the program to create a new bouncing box every time one of the boxes bounced, instead of doing this I decided to start with a preset list of boxes that would just bounce around and leave a colourful pattern behind them which turned out very appealing to look at anyway.

A problem I fixed was that the program ran very slow in the browser because so many rectangles were being drawn on screen every frame at the same time as values were being added to the arrays, I fixed this by running the grid in start up once and not drawing the background grid.

References

[1]

<http://minotaurproject.co.uk/psychedelia.php>

[2]

<https://www.youtube.com/watch?v=tNZZpXryAgc>

[3]

<https://www.youtube.com/watch?v=w025kQRMZwA&feature=youtu.be&list=PL6PNZBb6b9Ltgl6WM5rn2pjrXd_qdit2S&t=9>

<https://github.com/shorristhedonkey/shorristhedonkey.github.io>

https://shorristhedonkey.github.io/