Creative Coding Creative Audio-Visual Application:

Cameron Tebbenham-Small, 18003324, Date: 16/03/2020

Department of Computer Science and Creative Technology

University of the West of England

Coldharbour Lane

Bristol, UK

[cameron2.tebbenham-small@live.uwe.ac.uk](mailto:cameron2.tebbenham-small@live.uwe.ac.uk)

This report details the inspirations and processes involved in creating my Audio-Visual project for my Introduction to Creative Coding module, Year 1, Semester 2.

Table of Contents

**Type chapter title (level 1)1**

Type chapter title (level 2)2

Type chapter title (level 3)3

**Type chapter title (level 1)4**

Type chapter title (level 2)5

Type chapter title (level 3)6

**Design:**

During preliminary research, I had come up with three main ideas and jotted them down on paper:

1. I would create moving pipes as per the old windows screensaver.
   * An array of objects would be added to by clicking. These objects would move randomly across the screen, refreshing after a period of time.
   * <https://www.youtube.com/watch?v=MKqrLGFoK9E>
2. Expanding shapes that fill the screen.

* This idea was taken from a couple p5 examples I found on button pressing Booleans.
* On holding down a button, different shapes would expand, encompassing the screen, before another random shape taken from a list of 3-4 potential shapes would start expanding.
* <https://editor.p5js.org/TimSherbert/sketches/ryC0T0caX>
* <https://editor.p5js.org/AndreasRef/sketches/oCb-MMETX>

1. Wavelength that changes in shape and colours over time. Potentially link to sound.

* The most ambitious idea as it is the one I have explored the least in class or outside.
* <https://p5js.org/examples/math-sine-wave.html>

However, this all changed once I began coding.

At first I took the code from the pipeline example on p5js.org (<https://editor.p5js.org/methio/sketches/MAo4isGMN>) and tried to see how everything worked in the code. I took out elements I didn’t like, changed others and ended up with a sketch that would simply place a few circles on the screen. At this stage I was having a lot of issues figuring out how to create an array of objects that would move across the screen. It was at this point that I turned to the Coding Train YouTube channel and an example of objects in an array from class. I managed to create a sketch that would start with no objects, then add objects into an array and display them once the mouse had been pressed. I also used my knowledge and tutorials of the latest creative coding workshop – vectors, to replace the generic randomised movement and jittering with vector positions and velocity. This would enable me to perhaps add gravity and an attractor at some point in the future, if I had the time and know-how.

Building on this success, I decided to change my design from a simple mouse press, to a button press – taking inspiration from the sketches behind my second idea. Every time I press the button, the object would appear.

At first this proved simple enough, however positioning the button on the screen proved so troublesome, that I scrapped the idea of creating a button in JavaScript and instead made it using html and CSS. Then, I made a function to add and subtract objects from my array and called the function in HTML.

What’s more, I then decided to add some colour to the sketch, but add it in a form that could be controlled by the user. Up to this point, most of my sketches had utilised randomised colour attributes so, I decided to add three sliders for Red, Green and Blue. I tried adding them in JavaScript but encountered the same issues as with the buttons, so instead added the sliders in HTML, styled them in a CSS and called the property values of each using getElementById and the .value function in JavaScript. This created as many different “stages” as there are RGB colour combinations.

At this stage, I could add and subtract ellipses with a click of a button. The ellipses would bounce off of the canvas edges and their colour could be changed through three RGB sliders.

Next, I decided to create a more visually different second stage. However, the second stage would still contain elements from the first stage. Using tutorials from Coding Train, I learnt how to remove an object from an array by clicking it. Furthering this, I built upon the session 10 worksheet (<https://github.com/davemeckin/Intro_to_Creative_Programming/blob/master/session_10/session_10.md>) by adding a popping sound, an attractor and three new objects when I clicked on the stage 1 ellipses. I changed the colour of the attractors to black so that they wouldn’t show up and added force to the ellipses. This allowed me to then tie the ellipses into the update functions for the attractor so that adding attractors would alter the movement of the ellipses. Lastly, I added three smaller ellipses inside the ellipse constructor that would jitter inside the ellipse’s diameter.

**Evaluation:**

Overall I felt that my project was very successful and I do feel proud of what I have created. During semester 1 I changed my ideas because I didn’t understand the code or didn’t know how to code certain things. During this semester 2 project however, perhaps because I had more time to work on the one p5 piece, I researched my way through each issue as they came up. First with adding randomly moving ellipses to a canvas, to creating RGB sliders and nearing the end of my project, implementing my canvases inside a website shell. Due to this, I have a much more thorough understanding of how p5.js works, and have frequently been helping other classmates with their own issues.

One issue I had was using GitHub and local servers for version control as updating my code wasn’t updating the code on the servers. I did manage to find a solution by going into dev tools on the website, right clicking refresh and emptying the cache to force a hard reset. Once I had found this out, using GitHub and local servers was a lot more enjoyable.

Making the website was very challenging as I was using a html5up template: (<https://html5up.net/dimension>)

First, I had to merge my already made website with the new html5up template, as I had created a basic shell for my canvas. Due to the way the website is laid out, the template doesn’t use separate pages. Thus, when I clicked off the currently open article, the image would close but the interactivity project would still be running in the background. Through some comprehensive code reading, I managed to implement my reset function within the template’s own JavaScript code. One of the biggest issues I encountered was a personal stretch challenge; adding my separate semester 1 sketches onto the same page. It took me all of a Sunday to get everything to work properly. In the end, I had to combine all my sketches into one code and convert the present code into three separate instances, then call each instance in a separate piece of html code. Despite how long it took, I am happy that I managed to struggle through this issue amongst all the others I encountered while learning new things outside of class.

Another challenge arose when I attempted to create a timed countdown to display information about the project whenever a button was clicked. Despite spending a long time researching and testing different methods to try and get it to work, I ultimately had to scrap it and replace it with a much more simple piece of code where if the SHIFT keyIsDown, text is displayed.

Something I could maybe improve on is sticking to a single idea, with a pre-planned goal in mind. On the other hand, I much prefer stumbling my way through artistic creations as it leaves me open to new ideas along the way – such as adding the attractors and particles on button click.

**References, inspiration and tutorials used:**

The Coding Train YouTube channel tutorials:

* <https://www.youtube.com/channel/UCvjgXvBlbQiydffZU7m1_aw>

Code and examples studied to help create an array of ellipses that can be added to on mousepress of a button.

* <https://p5js.org/examples/objects-array-of-objects.html>
* <https://p5js.org/reference/#/p5/mousePressed>
* <https://p5js.org/reference/#/p5/mouseClicked>
* <https://forum.processing.org/two/discussion/18231/mousepressed-create-objects-from-an-array>

Code and examples studied in creation of RGB sliders in html that can be linked to JavaScript variables:

* <https://p5js.org/reference/#/p5/createSlider>
* <https://www.w3schools.com/howto/howto_js_rangeslider.asp>
* <https://stackoverflow.com/questions/39288562/how-to-get-value-of-html-element-in-javascript/39288612>
* <https://stackoverflow.com/questions/29103818/how-can-i-retrieve-and-display-slider-range-value>
* <https://www.w3schools.com/jsref/prop_range_value.asp>

Used in positioning different Javascript and html elements on the page.

* <https://www.w3schools.com/html/html5_canvas.asp>
* <https://stackoverflow.com/questions/42001276/css-bootstrap-container-background>
* <https://stackoverflow.com/questions/53980755/how-to-auto-position-my-buttons-on-my-website-with-different-screen-resolution>
* <https://github.com/processing/p5.js/wiki/Positioning-your-canvas>
* <https://www.w3schools.com/tags/att_div_align.asp>
* <https://stackoverflow.com/questions/40948611/how-to-link-p5-js-setup-and-draw-with-html-canvas>
* <https://p5js.org/examples/instance-mode-instantiation.html>
* <http://www.joemckaystudio.com/multisketches/>

Research into countdown timers:

* <https://editor.p5js.org/2sman/sketches/ry3eS_cgG>
* <https://editor.p5js.org/denaplesk2/sketches/BkNW2NCkM>
* <https://www.youtube.com/watch?v=MLtAMg9_Svw>
* <https://editor.p5js.org/denaplesk2/sketches/B1taLQvxG>

Audio Visual Project examples:

* <https://p5js.org/examples/objects-array-of-objects.html>
* <https://editor.p5js.org/methio/sketches/MAo4isGMN>
* <https://www.youtube.com/watch?v=MKqrLGFoK9E>
* <https://editor.p5js.org/TimSherbert/sketches/ryC0T0caX>
* <https://editor.p5js.org/AndreasRef/sketches/oCb-MMETX>
* <https://p5js.org/examples/math-sine-wave.html>
* <https://editor.p5js.org/methio/sketches/MAo4isGMN>

html5up website template:

Dimension by HTML5 UP

[html5up.net](http://html5up.net/) | @ajlkn (twitter)

Free for personal and commercial use under the CCA 3.0 license ([html5up.net/license](http://html5up.net/license))

* <https://html5up.net/dimension>

Credits:

Demo Images:

Unsplash (unsplash.com)

Icons:

Font Awesome (fontawesome.io)

Other:

jQuery (jquery.com)

Responsive Tools (github.com/ajlkn/responsive-tools)