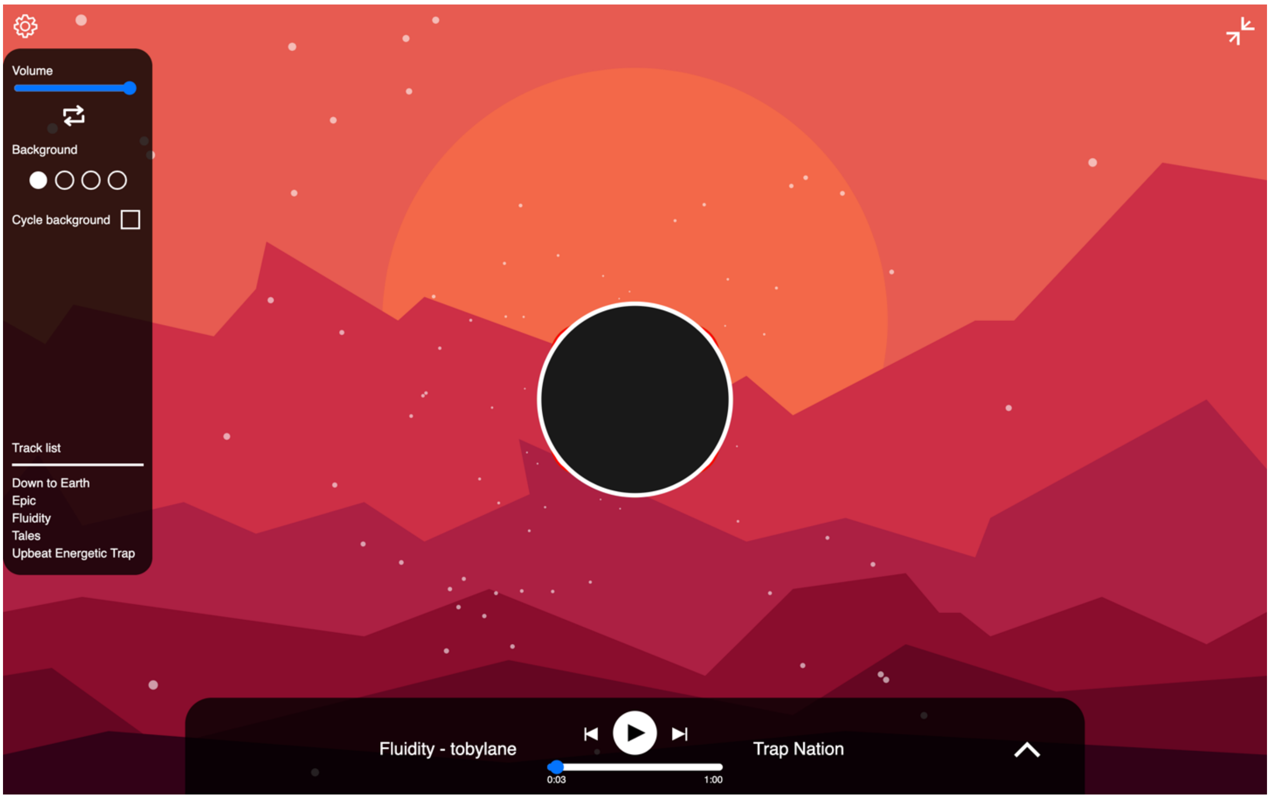
Introduction to Programming II Project Log

Project title:	Music Visualizer (23 Jan, 2022)
Topic:	Add metaballs and camera visualization

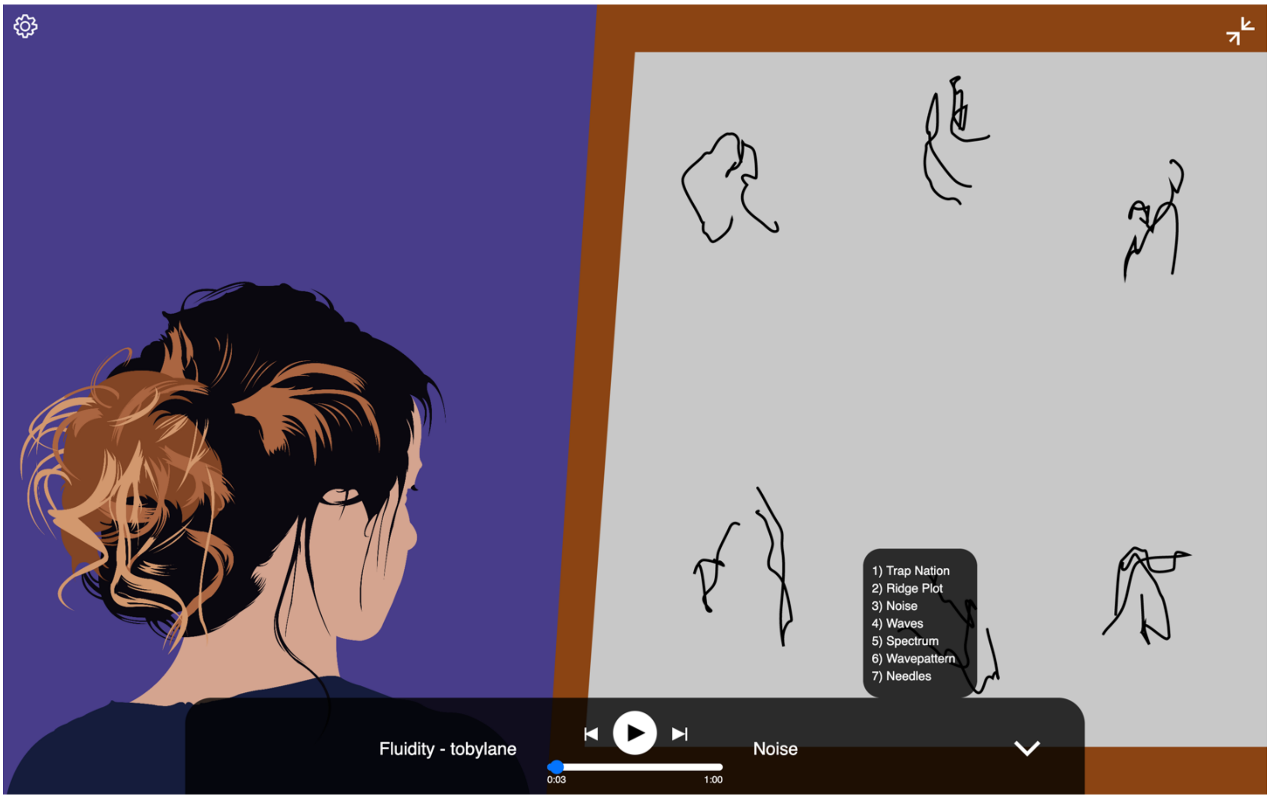
What progress have you made this topic?

I have successfully added and implemented the camera visualization (called “Waves”) as well as a setting to change the colour of its waves. There are three different colours to choose from, red, green and blue. I have also added a new slider under the play/pause button to act as a scrubber to seek different parts of the current track that is playing. The scrubber functions as expected. I have also added the ability to click on the visualization in the menu to change it, so now you can change the visualization by pressing the corresponding number, or click on the name of it.

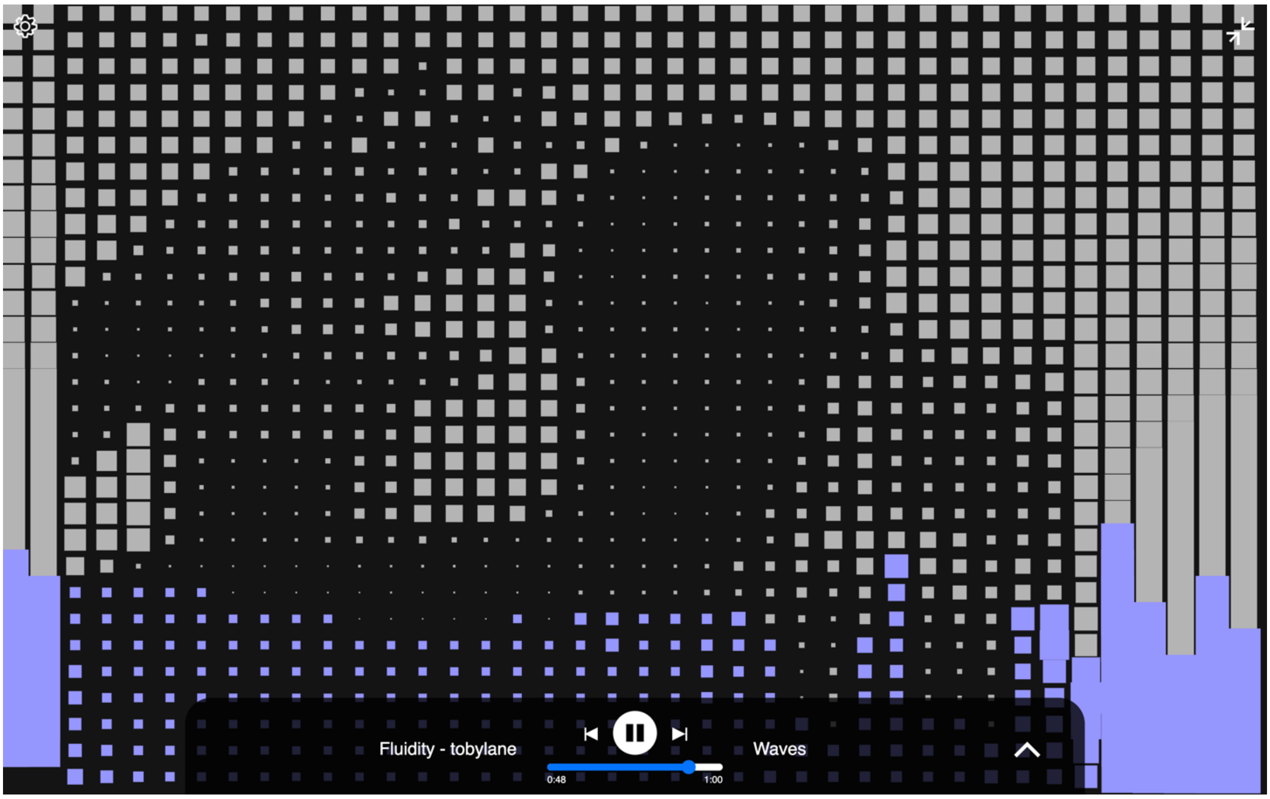
- “Waves” camera visualization added (functions as currently expected)
- “Waves” colour change setting added (can change from red, blue or green)
- Music scrubber added (lets user skip forward or backward in a particular track)
- Clickable visualization names to change visualizations (functions as expected)



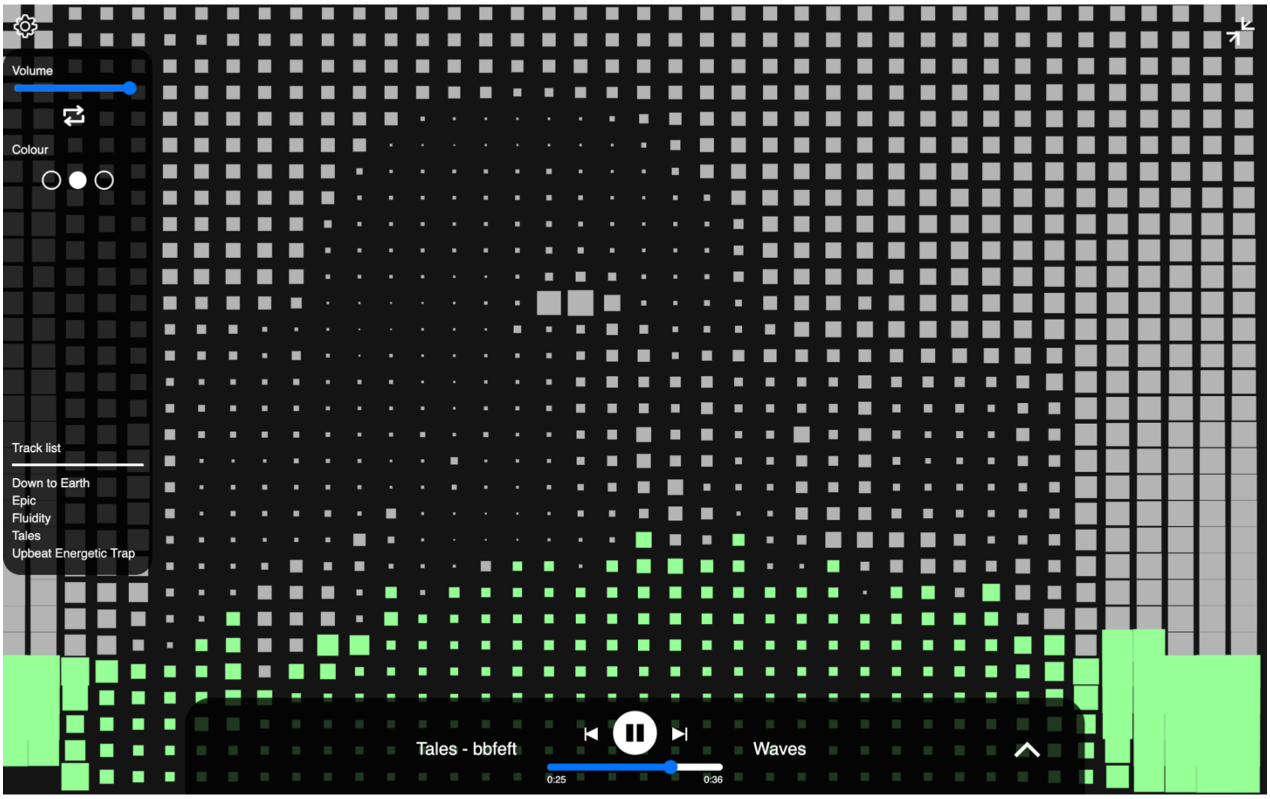
(Image above shows the new scrubber added)



(Image above shows the clickable names in visualization menu)



(Image above shows the new “Waves” camera visualization)



(Image above shows the colour change setting in the "Waves" visualization)

What problems have you faced and were you able to solve them?

The first immediate problem was that the metaballs visualization didn't turn out the way I would've liked. It lacked in originality and didn't function the way I had first envisioned. My solution was just to scrap the idea entirely, as the visualization didn't feel like a music visualization and lacked on aspects to improve upon. If I was to do this project again, I would definitely use my inspiration wisely; though some things look really appealing graphically, is it something that can be expanded on, or modified to make even better? I'd ask myself questions that are thinking further in the future, to see if it's actually feasible and worth committing to.
The only challenge I faced with the camera visualization was finding a way to use waveform() with columns of rectangles. This was easily achieved by only taking the number of columns from waveform and not the full 1024 and adjusting the size of the rectangles accordingly.
The scrubber proved a hard challenge as I couldn't use the p5.dom slider element to make it (because the value of the slider would change as the song changed, and as the duration in the song changed). The challenge was picking up, dragging and releasing the slider on a point that mapped the length of the slider to that particular part in the track. This was achieved by having independent events trigger when mousePressed(), mouseDragged() and mouseReleased() were called, as well as having a variable keep track of the current time that's been played and hold that value even if the song is paused.

- Metaballs visualization lacked originality and complexity (removed it completely)
- Waves in camera visualization adjusting rectangles in a column (take a specific number of values from waveform, not the full 1024)
- Dragging a moving slider and releasing it on a point that is representative of where the user wants the song to skip to (separate independent events that trigger one after the other, as well as a variable that remembers the duration of the track)

What are you planning to do over the next few weeks?

Over the next few weeks, I would like to add a few more settings to specific visualizations. I would like the "Trap Nation" visualization to have a setting that allows the circles behind to be changed to a waveform like pattern. I would like to add a setting to the camera visualization that allows the user to choose whether they want the waves to be at the top, bottom, left or right of the screen.

I want to refactor code, and start changing my existing code to ES6 syntax (change constructors to classes etc.). The UI.js file has every UI element made in it, though I am happy with this design it might be wise to separate the "Settings" function and refactor each visualization specific setting in a more efficient way as lots of code is copied and pasted as well as lengthy. It is also a good time to start adding comments on new refactored code and getting my code on a more elegant and presentable side.

Are you on target to successfully complete your project? If you aren't on target, how will you address the issue?

I currently think I am on target to complete my project. Having planned according to my Gantt chart, not adding a meatballs visualization has put me further ahead in my planning than I should be. This should allow me further time to improve on the three visualizations already given instead, as well as add a 3d visualization. Improving upon each of the visualizations given will be challenging and time consuming as it was not accounted for before and the scope of my project is growing when I'm more than half way through my current plan. Though I'm on target, it is concerning the amount of time I am currently spending on other modules. I might have to prioritize studying for my other modules more so than I had planned for originally, and that might affect whether or not I'd be able to add a 3d visualization (as that requires time to change each 2d visualization to work in 3d). As this is not an issue at the moment, I will plan to create a finished/polished version of my project before implementing the final 3d visualization. If I am unable to add a 3d visualization then my project won't suffer from being underdeveloped or from being elegant and flawless. In its current state it has 2 unique visualizations added by me, as well as 2 added from the UoL lecture videos, and the original 3 provided from the start of the project.

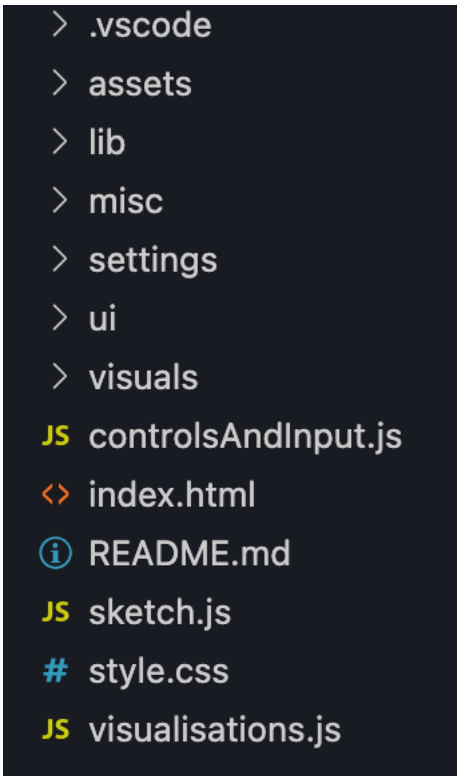
Introduction to Programming II Project Log

Project title:	Music Visualizer (6 Feb, 2022)
Topic:	ES6 syntax, refactoring, and Trap Nation settings

What progress have you made this topic?

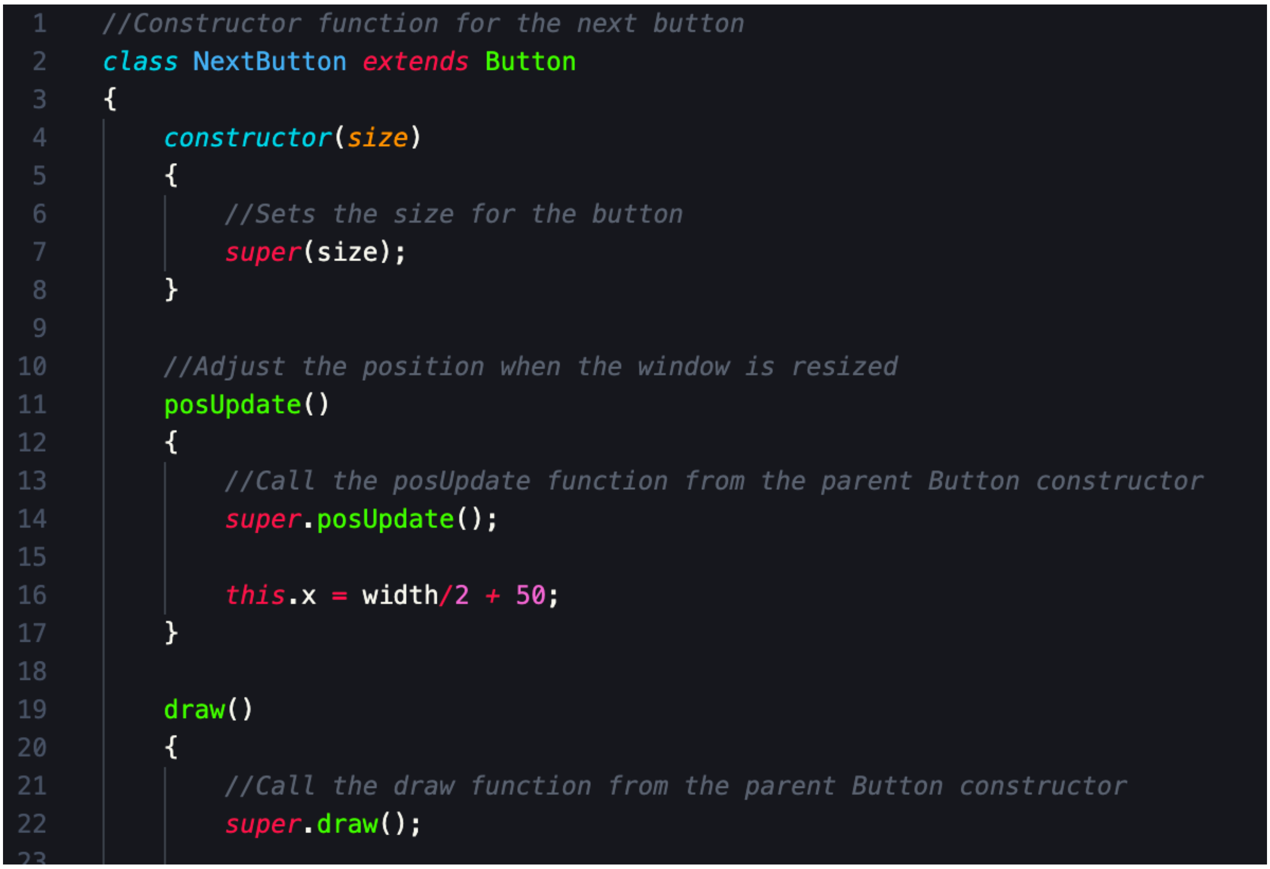
I have changed all the code into ES6 syntax, utilizing classes as well as “for-of" loops, ternary if statements, let and const declaration, and class prototypes. All code has now been commented, arranged into folders, and made more concise and readable by adding breaks to long lines. All constructor or helper functions are now in their own separate file, and the p5.js library has been updated to the most recent version. I have also added a new setting to the Trap Nation visualizer that changes the appearance of the visualization (changes the pulsing circles to a waveform), and a fill option to toggle when the user has selected the waveform appearance.

- P5.js and p5.sound libraries are updated
- ES6 syntax has now been used across all files in the project
- Code has been refactored and organized with folders, and separate files
- Commenting has been added
- Trap Nation visualization has new appearance setting



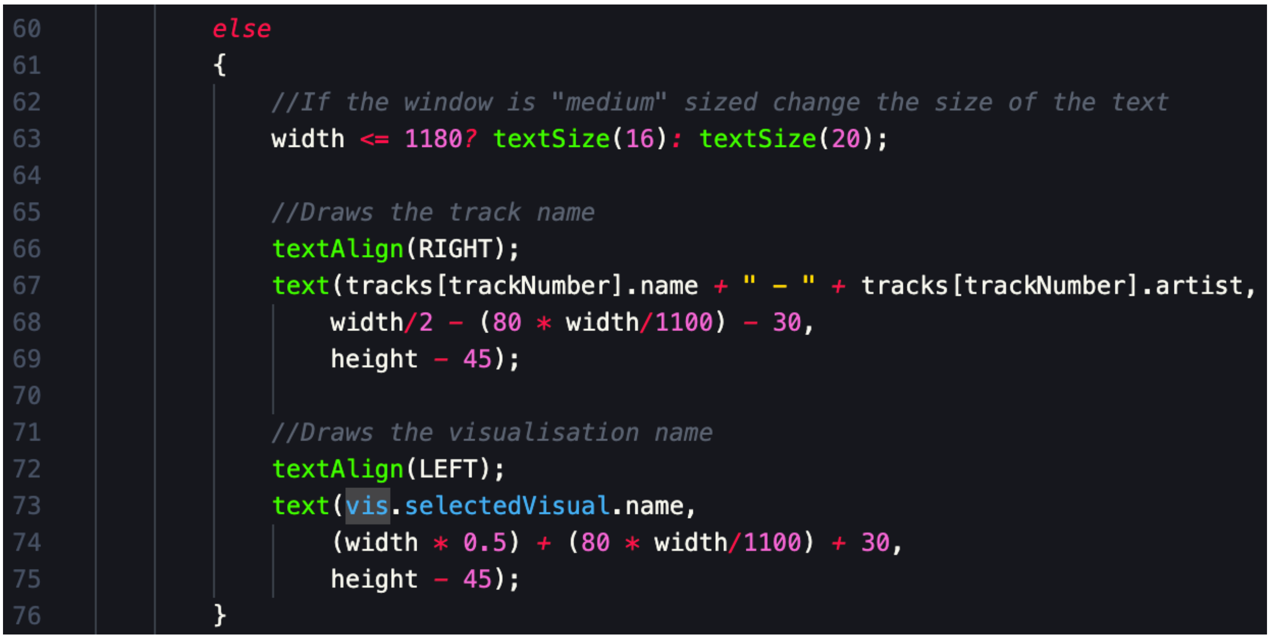
```
>  .vscode
>  assets
>  lib
>  misc
>  settings
>  ui
>  visuals
JS controlsAndInput.js
<> index.html
ⓘ  README.md
JS sketch.js
#  style.css
JS visualisations.js
```

(Image shows files have been categorized into folders, for file organization)



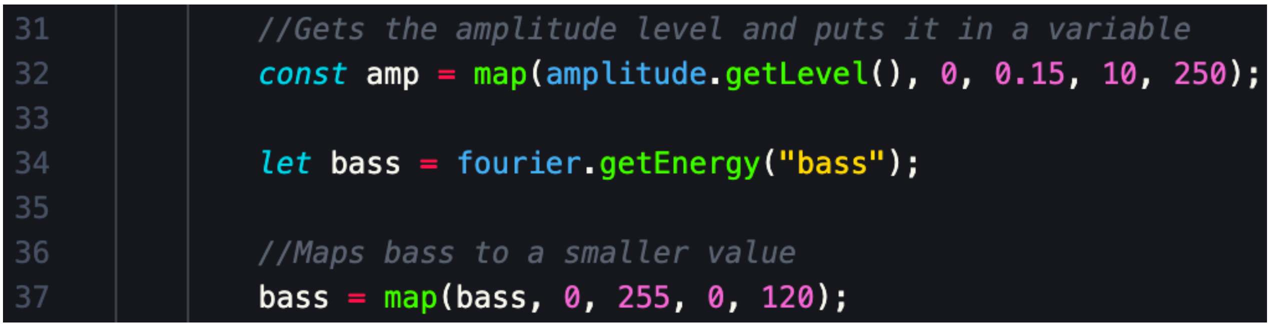
```
//Constructor function for the next button
class NextButton extends Button
{
    constructor(size)
    {
        //Sets the size for the button
        super(size);
    }

    //Adjust the position when the window is resized
    posUpdate()
    {
        //Call the posUpdate function from the parent Button constructor
        super.posUpdate();

        this.x = width/2 + 50;
    }

    draw()
    {
        //Call the draw function from the parent Button constructor
        super.draw();
```

(Image shows the use of ES6 constructors, commenting, and class prototypes)



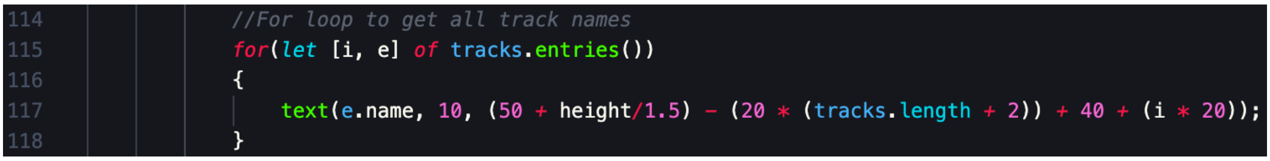
```
            else
            {
                //If the window is "medium" sized change the size of the text
                width <= 1180? textSize(16): textSize(20);

                //Draws the track name
                textAlign(RIGHT);
                text(tracks[trackNumber].name + " - " + tracks[trackNumber].artist,
                    width/2 - (80 * width/1100) - 30,
                    height - 45);

                //Draws the visualisation name
                textAlign(LEFT);
                text(vis.selectedVisual.name,
                    (width * 0.5) + (80 * width/1100) + 30,
                    height - 45);
            }
```

(Image shows ternary if statements, and long lines broken into separate lines for easy readability)



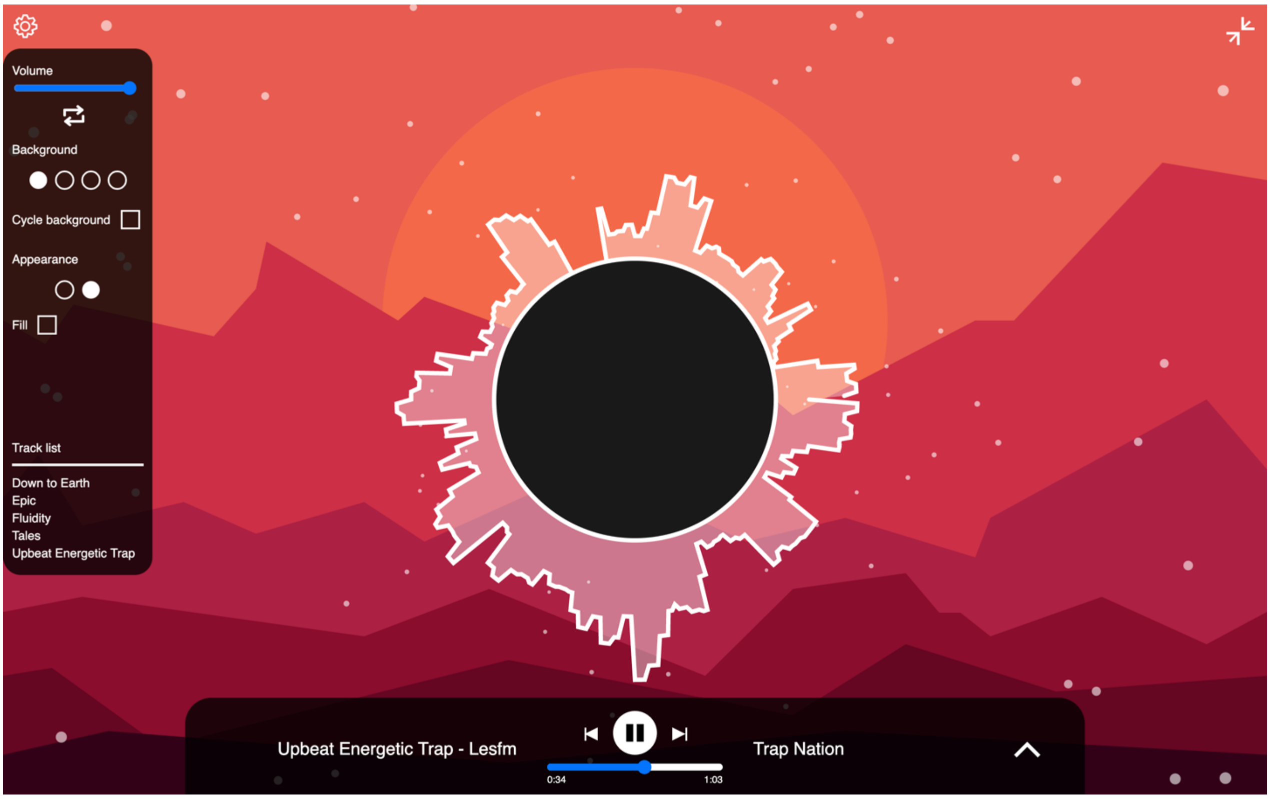
```
//Gets the amplitude level and puts it in a variable
const amp = map(amplitude.getLevel(), 0, 0.15, 10, 250);

let bass = fourier.getEnergy("bass");

//Maps bass to a smaller value
bass = map(bass, 0, 255, 0, 120);
```

(Image shows the use of const and let variable declaration)



```
//For loop to get all track names
for(let [i, e] of tracks.entries())
{
    text(e.name, 10, (50 + height/1.5) - (20 * (tracks.length + 2)) + 40 + (i * 20));
}
```

(Image shows the use of for-of loops)



Volume

Background ●○○○

Cycle background □

Appearance ○●

Fill □

(Image shows the new appearance setting and the fill option for this specific appearance)

What problems have you faced and were you able to solve them?

The problem I faced was the sheer amount of time it took to change the whole app into consistent ES6, and adding the comments. I had planned to add more settings for different visualizations but didn't have the time as I underestimated the time it would take to make my code more presentable and efficient. The next time I undergo a project of this size I will use the most up to date syntax and comment my code from the beginning; more often than not I found myself rewriting the code all over again, which rendered all the time I spent on it before useless. All the time I spent could've have been minimized or removed entirely if I had just planned from the beginning when the project was smaller in size and only had a few JavaScript files. The actual implementation wasn't as complex I initially thought, which leads me to believe I should've read up on ES6 before and not have made it an "afterthought".

What are you planning to do over the next few weeks?

As time is against my side more than I had planned, I would need to get user feedback and get my application tested, so I would have time to fix any bugs or amend the code where needed. At this point in time, I have found myself in a position having to sacrifice visualizations and additional features to meet all the requirements for the project. I plan to conduct the user testing as quick as possible to hopefully allow myself enough time to fix any bugs or issues that were unforeseen. I have made the decision to sacrifice additional content in the attempt to have a more polished, bug-free project. I would also like to get all the documentation for this project collected, finalized and stored before my exams for other modules.

Are you on target to successfully complete your project? If you aren't on target, how will you address the issue?

I was not on target. If I plan to still add a new visualization and additional settings, I will not have the time for user testing and feedback. I was not on target because I massively underestimated the time it would take to refactor and comment my code. The more I added to my project, the more I had to refactor, and this was a big priority for me to complete in the beginning. In the future I would put updating, commenting and organization a priority from the very beginning and not leave it till the end of the project when it's grown in complexity. I will be on target at the cost of missing an extension, and a few extra features I would've liked to add. In my Gantt chart I should've had the metaballs visualization and I should start working on the 3d visualization. However, I gave myself just over a week to do all the refactoring, user testing, and debugging which is terrible planning as it took me 2 weeks to complete just the refactoring alone. My application has unique implementations (though I would've wanted more) and it should still score highly, so the focus is on documentation and feedback.

Introduction to Programming II Project Log

Project title:	Music visualizer (20 Feb, 2022)
Topic:	User testing and feedback

What progress have you made this topic?

I made a testing sheet a survey for users to test and review my music visualizer. I had six CS students test my application, two of which aren't doing ITP2, and four who are extending different applications. All six users have completed their testing and surveys, all data has been collected and a summary has been made reflecting on the feedback given.

- Testing forms and surveys have been made
- Six CS students have fully completed both the testing and feedback survey
- A summary and reflection document was made, analyzing the data to better judge what to do with the app for the last 2 weeks

What problems have you faced and were you able to solve them?

This week has shown little to no problems. As I was asking current CS students to test my program, I had to give each student enough time to do the test without compromising their own time management. Having given them at least 2 weeks to submit their feedback I got all my user testing back on time, with very good feedback. This topic takes a significant amount of time, and I didn't plan for it to take that much time in my Gantt chart earlier in the module. In the future I will make sure to allocate more time to testing, and hopefully allow testing to be done multiple times over the applications development, as the feedback I received was extremely useful and if I had planned better, I would've wasted less time improving on things that didn't need improving.

What are you planning to do over the next few weeks?

Over the next few weeks, I need to write up my report and answer all the questions. It is from this point where I start organizing everything to be zipped up for submission.
From the user feedback, 50% of users stated they would want to select tracks similar to the way you can select visualisations. Though I wouldn't have time to try and implement every piece of feedback given, this particular feedback was common, and the most requested, therefore I want to iterate on my application and add this feature.
My last project log should be the final submission, and I would have completed the ITP2 module.

Are you on target to successfully complete your project? If you aren't on target, how will you address the issue?

The small scale testing I was doing throughout this module left my project in a near finished state according to the users, which means I have no bugs to address or immediate issues to resolve. From my last project log, I changed my timetable to have 2 weeks to address any issues, but my testing has found there are none to address, which should mean I am on target to complete my project with time to spare. Sacrificing the 3d visualization was the right decision, as now I have time to iterate on my application and improve its usability. With the benefit of hindsight, I can see a few smaller "quality of life" improvements can be done to improve the user experience as a whole. The users reported that the current visualisations were all unique and original. My main reason for adding a 3d visualization was because I felt I was lacking in originality, but through feedback I found I was actually lacking in usability. Changing my timetable and scale of the project was the right decision, and I have mitigated any usability issues that might not have been addressed had I not put time for testing and iteration.

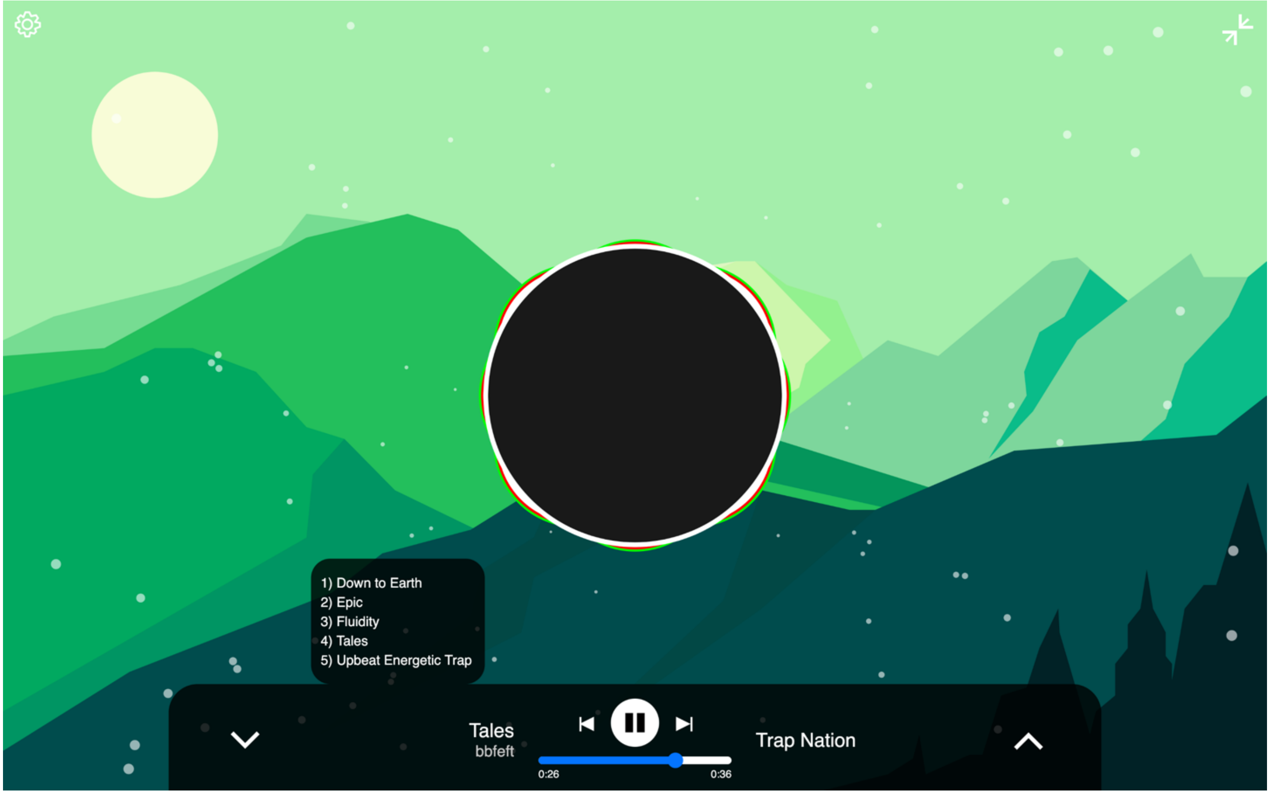
Introduction to Programming II Project Log

Project title:	Music Visualizer (6 Mar, 2022)
Topic:	Questions and iteration

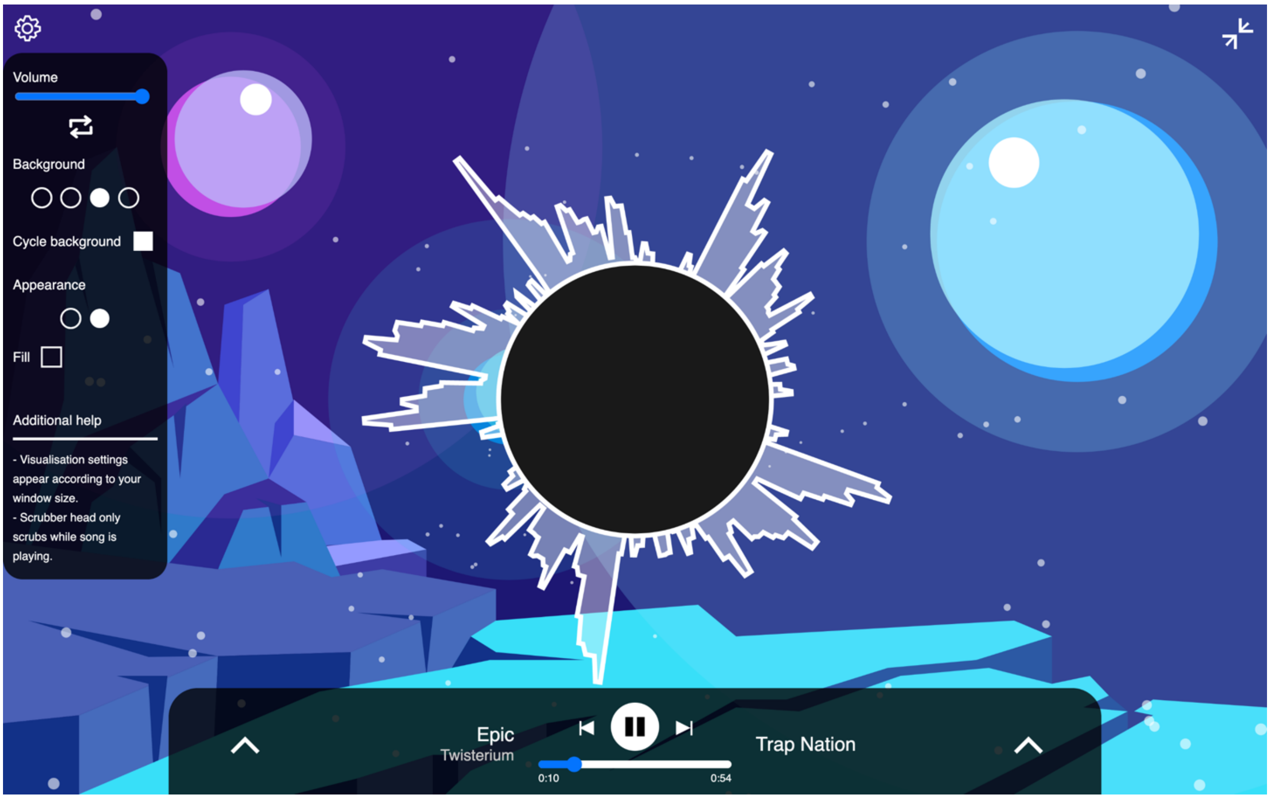
What progress have you made this topic?

All answers to the questions have been typed up, checked against the word count, and answered to the best of my abilities. A menu for selecting the different tracks in the app has been added. The text in the settings menu has been changed from the track list to “additional help” to offer small tips on how to use the app. The tips came from user feedback as some users reported small “ease of use” issues. The project file has been cleaned up and zipped, and is now submitted.

- All questions answered
- Track selection menu, and additional help text
- Project submission



(Image shows track selection menu, and slight change in track text appearance)



Volume

Background ○ ○ ● ○

Cycle background ■

Appearance ○ ●

Fill □

Additional help

- Visualisation settings appear according to your window size.
- Scrubber head only scrubs while song is playing.

Epic
Twisterium

0:10 0:54

Trap Nation

(Image shows the changed text in the settings menu, to offer additional help to aid ease of use)

What problems have you faced and were you able to solve them?

No problems were faced. The track selection menu is near identical to the visualization selection menu, so there were no new challenges. The delegated time was perfect, and everything I wanted to get done, was done.

What are you planning to do over the next few weeks?

Take a break. Spend the next few weeks off resting. Some point in the near future I'll start preparing for my next modules.

Are you on target to successfully complete your project? If you aren't on target, how will you address the issue?

The project has been successfully completed, and all my new targets were met. No issues to be addressed.