Project Summary

What is the Project called?
Unity music visualizer
Who is on your team (include all names)?
Ben Weiler and Sean Fitze
What is the high-level overview of your semester project?
Create a script to generate visuals for a given song with Object Oriented design structure and patterns.
What are you trying to accomplish?
We wish to gain a better understanding of C#, Unity, and Object Oriented design while creating an attractive final product for consumers to use to display their audio.
What will your system do when you are done?
The final product will be able to display objects moving, changing colors, and changing shape and size to match a given song.

Project requirements

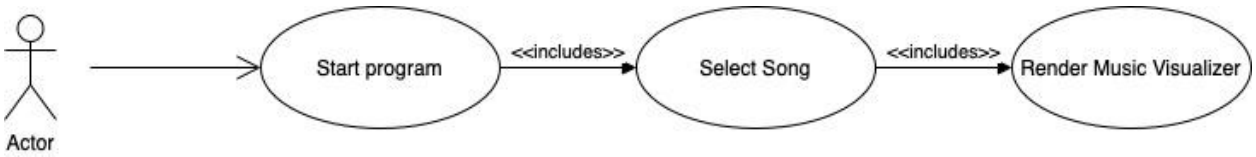
This project should be able to take a selcted song, analyze the song metrics (FFT and BPM) every second, convert these metrics into a series of foreground and background objects and colors then display these components with text and as an animation in Unity. The conversion process will involve random and deterministic functions. For example a certain high frequency may result in a random selection of a range of red colors for a background color. BPM is hopefully going to be used to change the item size or animation speed per beat. We are hoping this system can be real time while a song plays but may be a little slow depending on how efficient our program runs. The songs being run will also need to be built into the product as it would be difficult from the designers perspective to have the user input a song as a .mp3 or .wav so we are going to allow the user to pick a song from a predetermined list. The usability is still determined on how far we are able to get with unity and if possible we will deploy our product using WebGL, but realistically this will most likely be by a text based interface with the help of unity to display our animation. However, we think if unity becomes a challenge with WebGL integration we will have a webapp with angular 12 display 1-5 of the users selected songs and just play a video of the selected song on the webapp. The first big step will be using our initial UML diagram and OOAD patterns to make C# a text based display for each song frame - meaning full FFT functionality within unity without the display of unity making the product just text explaining what each frame will look like as a song plays. The final step will be implementing our C# code into unity to fully display and animate our screen. Both of the project members have little experience with unity and implementation may become very difficult and our final goal for the level of detail we want to achieve may need to be slimmed down.

Users and Tasks: Use Cases

- How many different types of users will your system have? What tasks do they need to accomplish with your system?

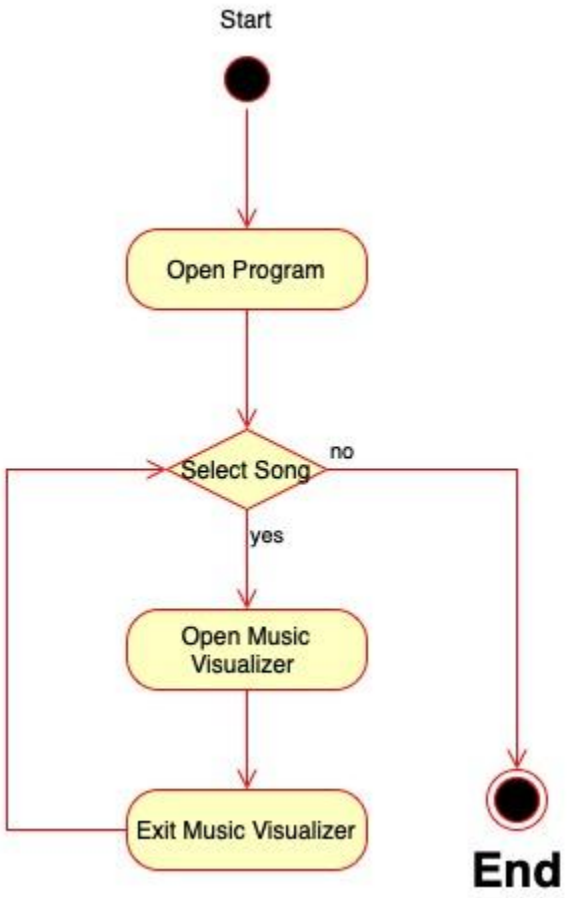
  The only user required is the main client who will select a song and receive the rendered animation.

- Document how the system will support all major user tasks by providing one or more use cases (using a text template or UML).



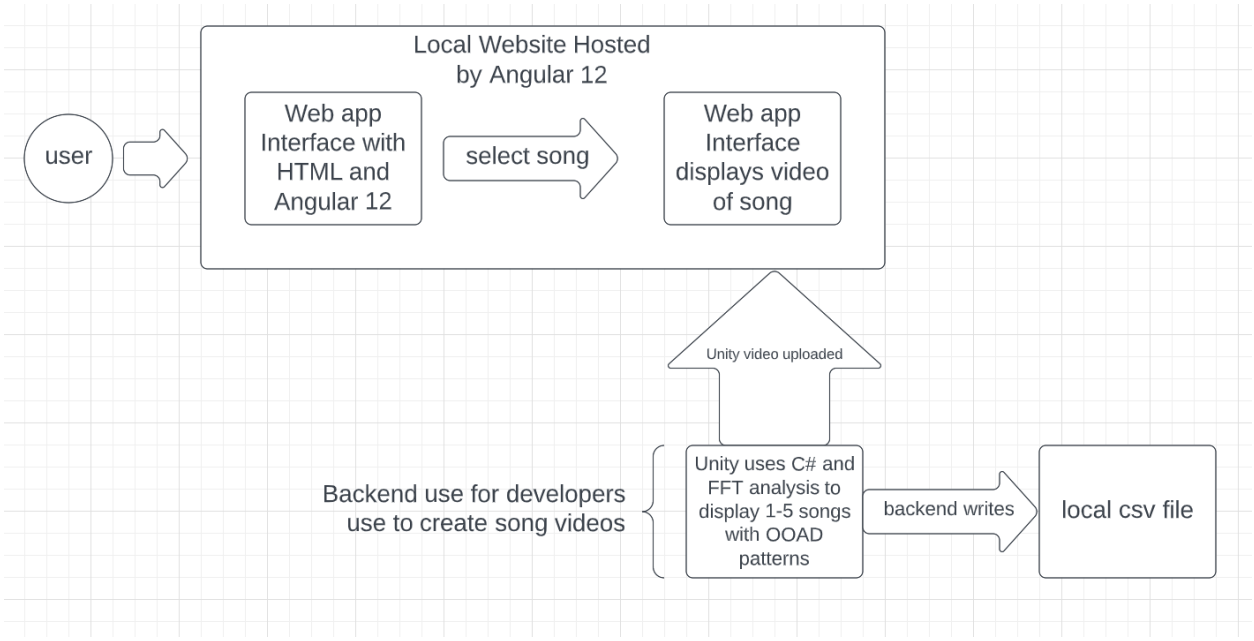
Program from the user's perspective is very simple, select a song and watch it be animated. As far as variation goes there is not much to account for. Most of the complexity is how the program handles tasks between these actions.

Activity Diagram



Open program will display the first window and give the user the option to select a song. From there the user can exit the program or select a song to be rendered. If the user decides to render the song it will open the animation window until the user decides to return back to song selection. From here the user can select a new song or exit the program.

Architecture Diagram



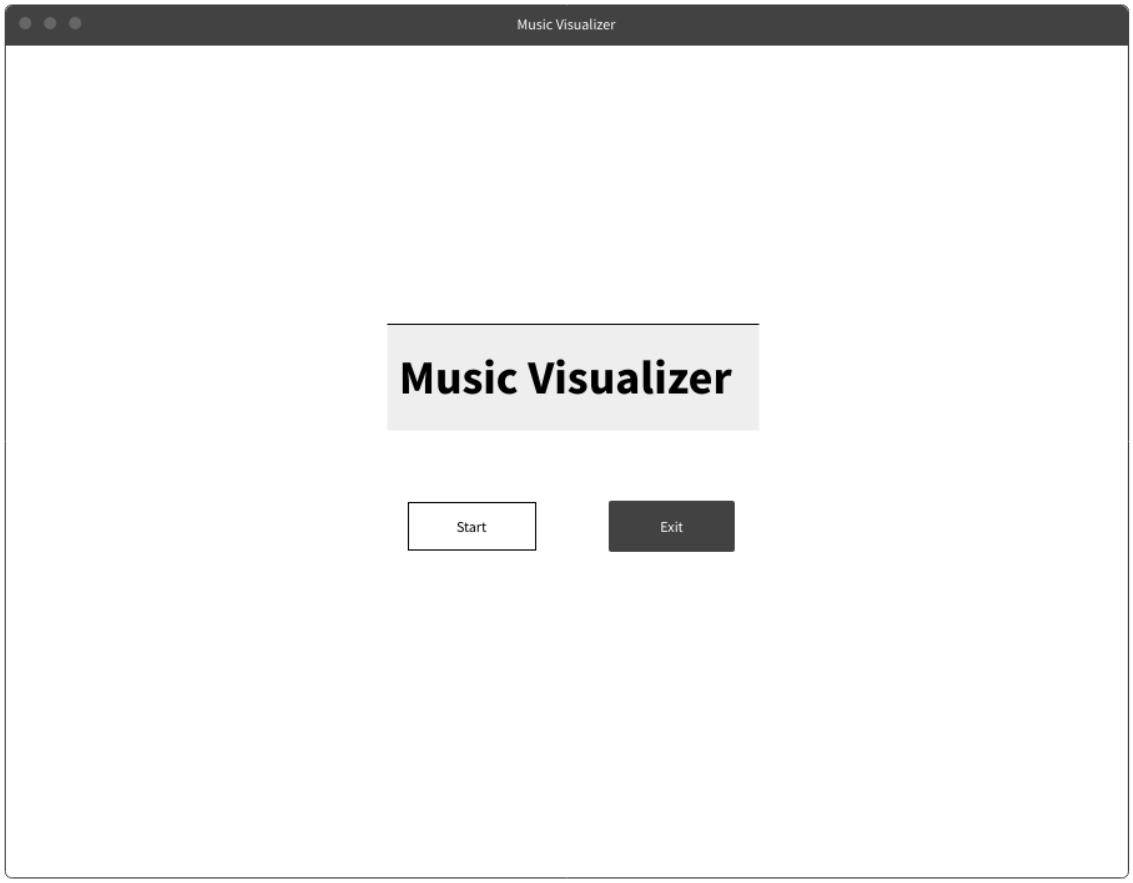
Data Storage

We plan to use a csv data storage option. The DataGather will take care of the CSV operations and will be called by FFT controller within the update method. This data will consist of the high, mid and low FFT values per reading in a song. If we are able to we may stop the FFT analysis when a song is played in real time and our code will read from the generated csv on the webapp instead.

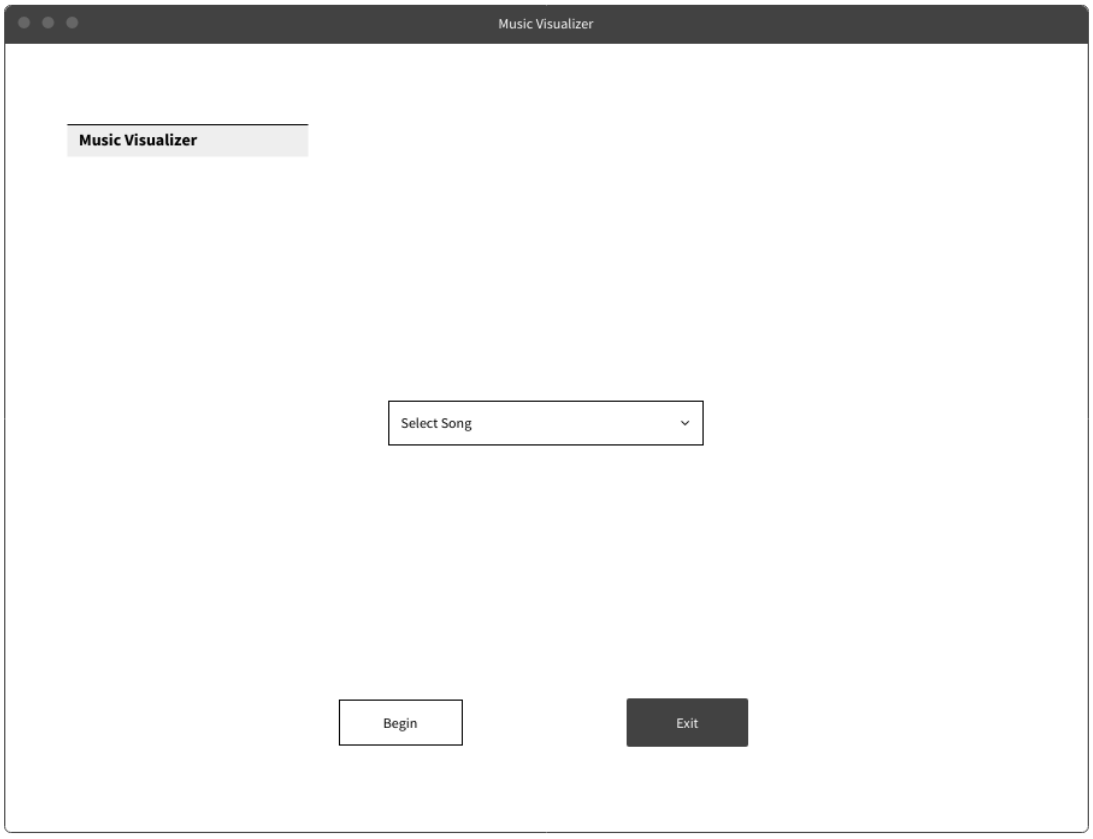
Low	Mid	High
L1(int)	M1(int)	H1(int)
L2(int)	M2(int)	H2(int)

UI Mockup/Sketches

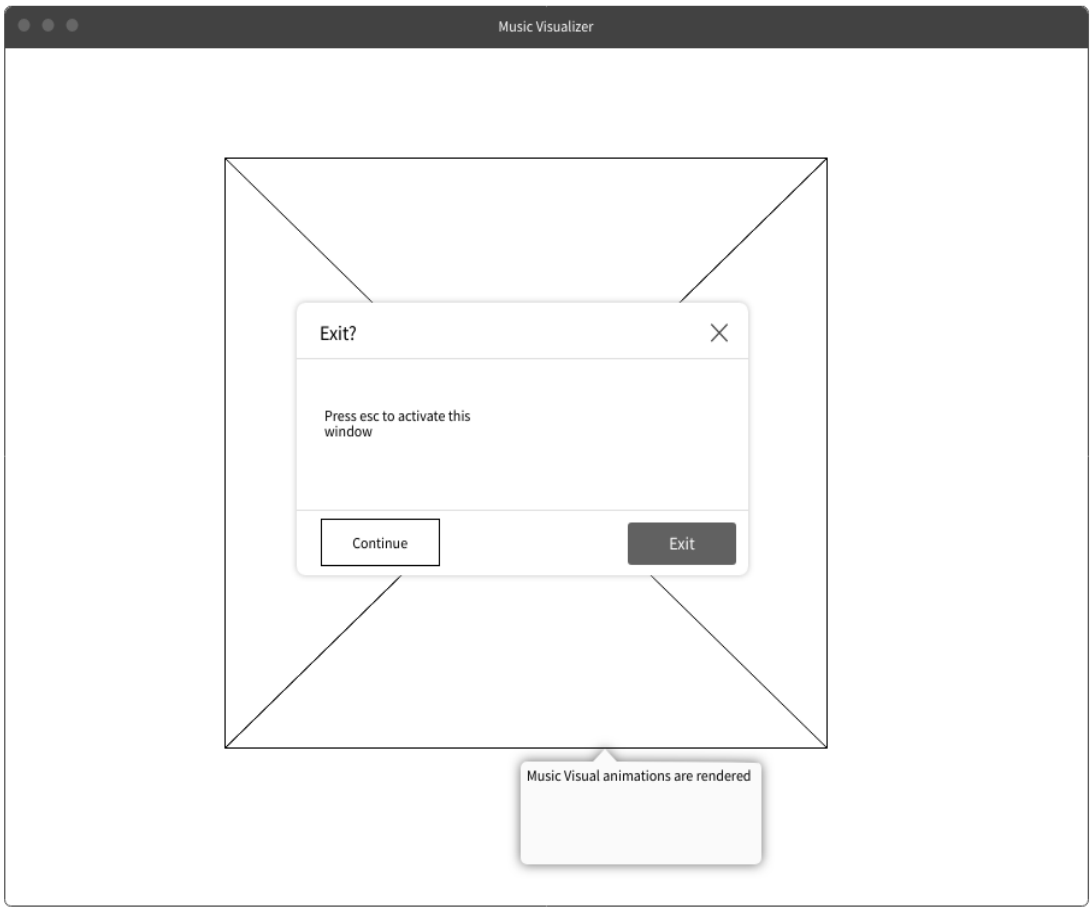
Start Page:



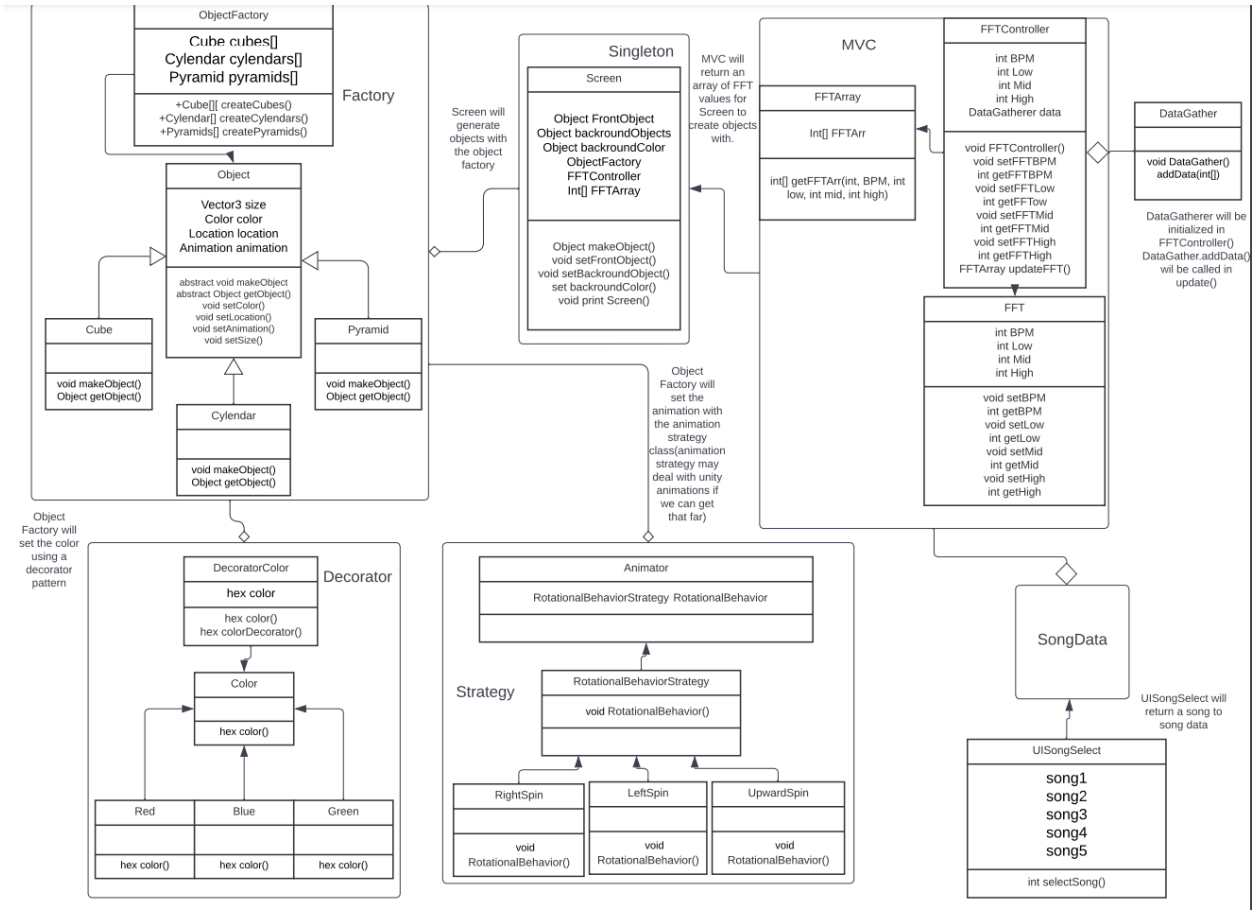
Select Song:



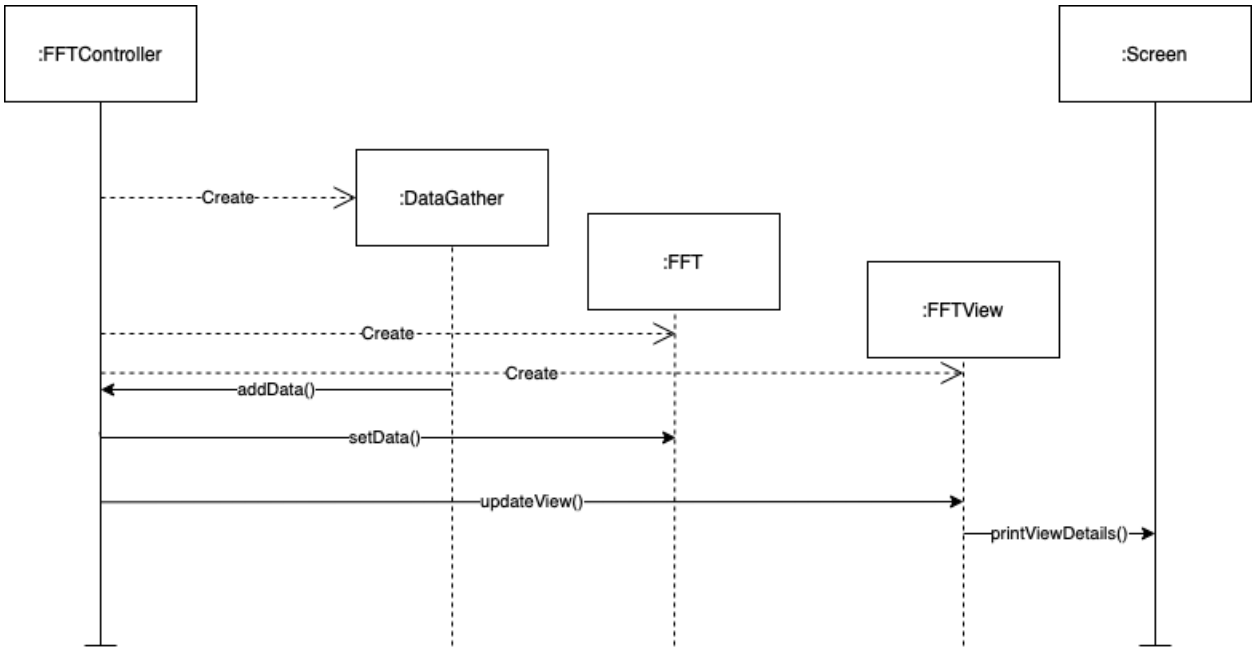
Music Visualizer (where animations are rendered, continues until you press esc):



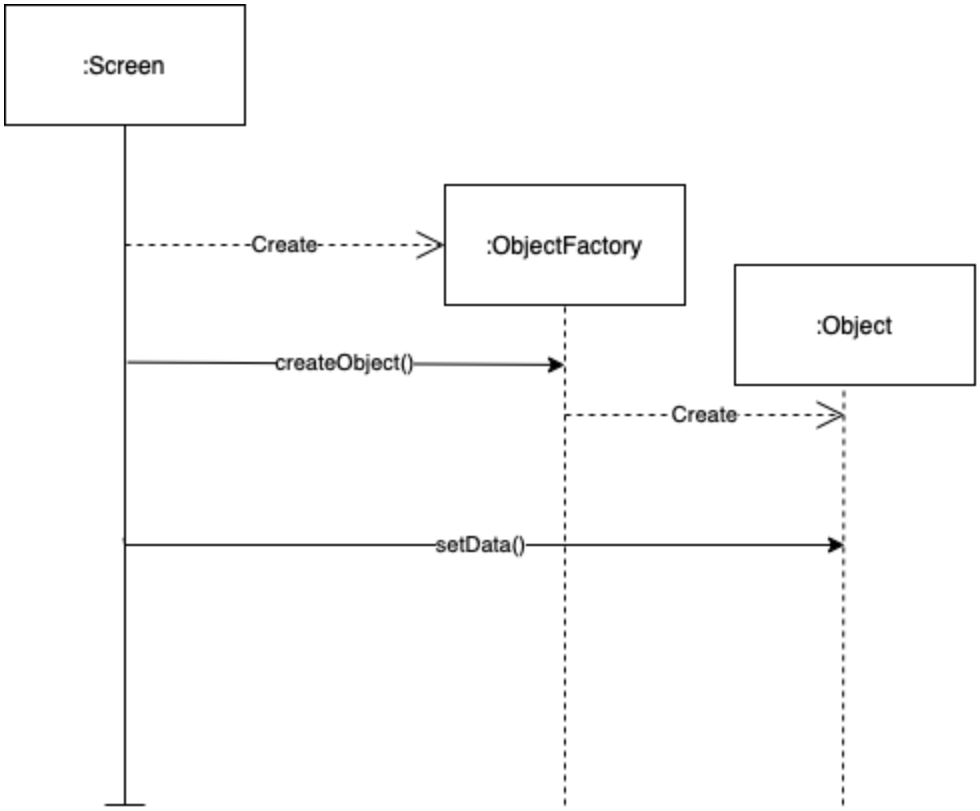
UML Class Diagram & Pattern use



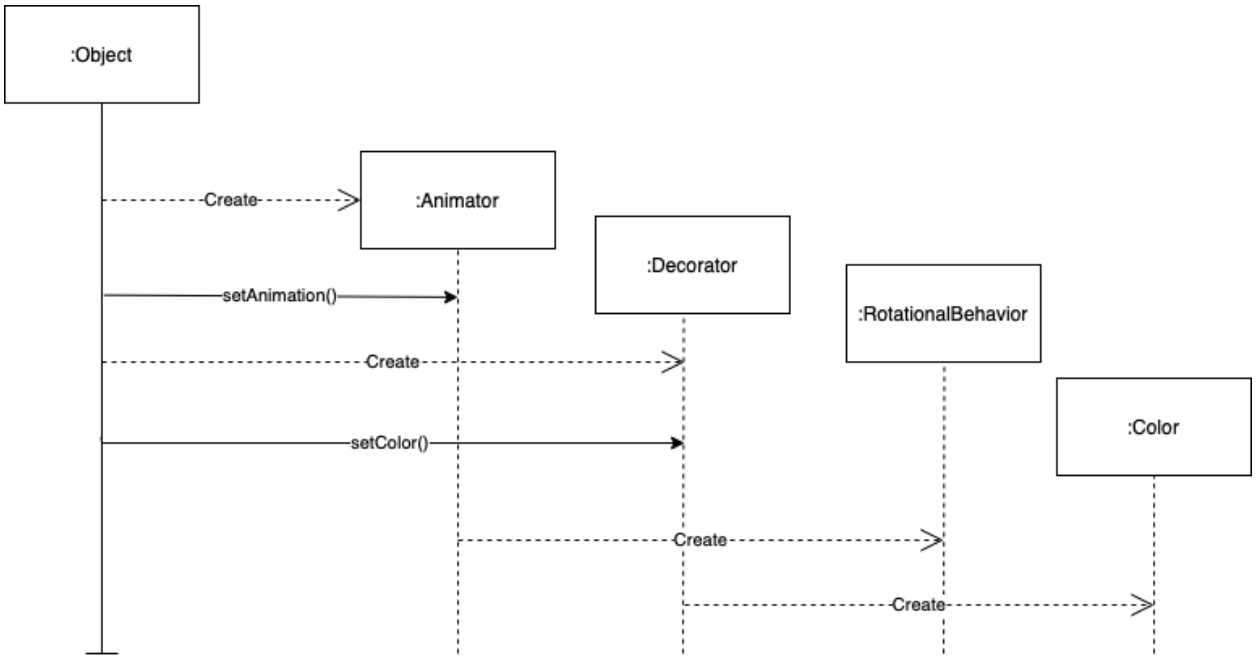
UML Sequence Diagram



This is the initial interaction for collecting data from songs and adding that data to the screen to render.



The screen will create an object factory that will create an instance of a specified object.



Once an object is created it will be given properties to set. It will set the color properties using a color decorator and assign an animation algorithm using the animator object (strategy).