PROJECT PROPOSAL DOCUMENT

# Project Name

Navi Music Visualizer

# Team with Contact information

David Wells: (206) 946 – 2879, wel20012@byui.edu

# Stakeholders with Contact information

None involved

# Project Purpose

Musicians should be able to focus on just making their music good. This should take care of the distraction of making visuals to go with that music.

# Background/Prior Knowledge

In the past, I’ve made a program that did a variety of music-related things such as working with a MIDI keyboard, drawing the notes played on-screen, recording input to MIDI files, and playing them back. While this was written in Python, Navi will be written in either C++ or Java, as both are fast enough for real-time rendering, and the latter is where most of my professional programming experience is found in. I would mark myself as a Novice in Java despite my history, as my previous projects with it have not been too complex.

This program will have to rely well on the concepts of polymorphism, inheritance, graphics rendering, multi-threading, and the rules of good code-writing (readability, maintainability, and performance), which will be aided by my knowledge in algorithm design.

My mother always said I should mix my skills with music with my skills in programming. She thinks so highly of my intellectual ability. That’s not why I came up with this idea, but the idea came to mind because it does solve the very issue I’ve been having with my Youtube channel, being that I spend most of my time focusing on how videos look rather than how they *sound*, which is what I want to focus on.

# Description

This project will create an automated audio-to-video pipeline that works well for all musical genres. The real paintbrush for these videos will be what I’ve referred to as “fairy light trails” (FLiTs), which will populate on screen based on the number of components found in the audio.

Their colors, brightness, speed, movement, and position will depend on audio component features such as volume, vibrato, reverb, frequency, tempo, sustain, and general timbre, as well as the FLiTs around them.

Settings given to the user to modify the output will include color palette, color usage (which determines whether colors will be for instrument choice, emotion choice, or other options) background image, image durability (how well it stays together when affected by the FLiTs), seed number (for reproducibility), FLiT synchronization (think synchronized swimming), style choices (such as 8-bit, where colors are reduced, or watercolor, plain, and more) and then basic things like video resolution, CPU/GPU utilization, and I/O file location. The most important option will be speech transcription, which will be presented as a simple on/off button. Another important thing will be determining where the medium volume of the song will be, which will be automatically applied in decibels but will be alterable by the user.

Video editing options may or may not be available (such as camera movement), as I don’t want to include things in the project that don’t *need* to be included. I may have to include word placement on a screen timeline depending on how will speech transcription ends up being.

Solutions to this problem already exist, but they do not break up the audio the way this will. Their automation is also very clear, and in my opinion the options out there do not contain much personality.

I will release this as a free program for a year, with a Patreon page included on the website you’ll be able to download it from. This is how you gain traction with this kind of project. My primary audience will be musicians and music appreciators who just want to see audio in an objective visual form. I want to see videos on YouTube show up for all kinds of songs using this program.

The demo will feature only the most basic things. Audio input, audio component breakdown, and simple FliTs that leave simple light trails on a black background. I feel I can have this prepared in 3 weeks at most.

# Significance

This project will uplift and present my knowledge on almost every basic aspect of programming there is. I do want to get into graphics programming specifically, so any employer who sees this will know that I mean business.

On my resume, I’ll put that I’m the creator of the Navi Music Visualizer, which will have a link to the download website. This site will also be made by me, and will at least demonstrate my website-building abilities too.

I know I’ll get small donations from family members and friends from my Patreon page, but beyond that I just want to have a great portfolio and get hired by somebody.

# New Computer Science Concepts

Multi-Threading is a big one

Video Rendering is also a big one

GPU utilization is important too

A simple, familiar UI isn’t new, but is something I’ve needed to improve my skills in.

Light physics is new as well.

# Interestingness

I’m excited about this project because I will use it for my YouTube channel, which I’ve been working on for 5 years. My focus *must* be that this is fun enough to watch that you never get tired of new ones.

# Milestones, Tasks and Schedule

Week 3: Class Design

Week 4: Determined Scope

Week 6: Basic MVP (as a workable demo)

Week 8: Website up with download available

Week 14: All other features in scope added

# Resources

Chat GPT

Advice from my father and brother (programmer and film producer respectively)

Purchasing one competitor music visualizer to see where the bar is for quality (up to $10)

# Dependencies

Windows platform (already installed)

Visual Studio as the IDE (already installed)

C++ and Java (already installed)

No permissions required (Navi is not a trademarked name)

# Risks

I don’t know how to utilize a GPU, create a download page, create randomness from a seed number, create a video renderer without bugs or glitches, or whether it’s possible to create an objectively beautiful, contextually obedient, infinitely interesting audio visualizer that can break down audio into instrumental components and speech. I don’t know how to do almost anything this project requires beyond planning on how to design it.

My biggest fear is that I won’t want to use it. I’ve made quite a few music videos that were very differently styled, and I don’t know if this will be good enough for me to want to re-use it over and over again, let alone whether anyone will want to watch those videos. Keeping this in mind will be crucial in the design process, which may need to be altered somewhere down the line. If I design the program in a way that can’t adjust to that, I may just be stuck with what I have, and then the project won’t reach the heights that I want it to. We’ll see what happens.