Alvin Zhou

Professor Stalbaum

VIS 141

11, December 2025

Dance of the Comets

A clear explanation of what your program does:

For my Final VIS 141A project, I decided to create a project that operates on real-time audio and visual experience across the 4-screen MPI cluster. My program creates randomly moving comets that travel across the large screen. Each rank has its outer boundaries that either reset or bounce the comets from them, and the center is connected to help move each comet across multiple screens. The visuals are mostly driven by a music wav file acting as a dataset. As the sound becomes louder and more energetic, the comets will then glow brighter, move faster, and more dynamically. On the other hand, when the audio is softer, the movements slow down, and the scenes become much calmer. This reaction from the audio is displayed across multiple screens and displays how audio is visualized through each comet. The comets are separated into two different colors, green for the bass and purple for the melody. Each acts independently, as the bass maintains the same speed, but only grows bigger; on the other hand, the purple melody comets shift in speed, but stays the same size. The overarching view I would like for my project to display is the idea that sound visualization can be both chaotic and uniform. The easiest way for us to know is just by visualizing it. Many times, as I look into my project with sound, I am encapsulated by how each comet moves and how they react to certain elements of notes throughout the song.

What API or database (if any) your project uses:

For my project, I do not use any external API's or online databases. Instead, it uses a local audio file that is stored on the MPI cluster. This file serves as the main "data source" for the program to run and detect. The audio file is called Music.wav, and it's stored within my folder on /work/alz008/~. By placing this file into the cluster, all MPI ranks would be able to access the same file, no matter which node they run on. We then take the amplitude values from the wav file, and in turn, that influences the comet brightness, speed, and the overall visual intensity. Overall, I use the audio file as a dataset, and the code can react to its data in real time to produce results mimicking each layer of the music.

Your process:

From start to finish, my process was pretty elementary, as I had no previous knowledge of how to code with MPI clusters and how to correctly code with Python. Therefore, using our class example, I first started with the oneball.py.

I used this as my base as I tried to understand how I should start working. I tried changing the speed, size, and direction it moves to see the bounds of how I can start coding. I then used ChatGPT to help me understand the random shapes and started allowing them to move around the cluster. During this time, I encountered my first issue, which was even being able to display my code. However, through trial and error using oneball, I was able to learn and apply my knowledge on displaying content. Moving forward, I made the balls colorful and added a barrier on the outer corners of the cluster so that the balls would be able to bounce back and forth.

Next step, I was able to start adding depth to my program and differentiate between two different tones of comets. During this time, I was able to finalize my idea to make a music visualizer as the project. I then added bloom, screenshake, and trails for the comets to look visually cleaner.

In my final step, I have addressed any remaining bugs and added a minimal background to fill in the empty spaces. And distinguished the two-tone comets into melody and bass comets. Between the two, I added the characteristics of growing and speed that map the comets as they reflect the frequency of each sound within the song. In the end, my final results are my attempt on trying to visualize something beautiful that we can not see. Although I am still a beginner with coding, I believe that my project can encapsulate the viewer and draw them into the work of sound visualization.