Music Dynamic Animations

creating imagery for the classical music

Department of Computer Engineering
California Polytechnic State University, San Luis Obispo
Thuy-Vy Do

March 2015

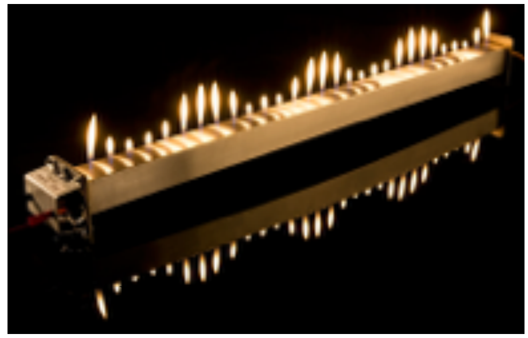
Abstract

Audio visualization gives a graphical representation to beats, intensity, and the fluctuating sounds heard in music. Music dynamic animation, deriving from this concept of audio visualizers, implements a hard coded range styled beat detection. This algorithm prioritizes graphical representations of classical and instrumental music. These graphical representations are intended to provide a sense of imagery.

Introduction

Audio visualization gives a graphical representation to beats, intensity, and the fluctuating sounds heard in music. This popular application extends those of music players in the form of “iTunes visualizer” or “Windows Media Player visualizer” to establish fluid graphic transitions to add flair and enhance the audience’s experience. This project explores the option of creating visual artifacts in the form of short animations set to classical music. The goal of this project is to create a computer program that can detect beats and timing in classical music and then choose an appropriate pre-drawn animation sequence to best fit the scenario.

Music dynamic animations comes from the concept of audio visualizers. This project addresses the ability to utilize programming for an artistic purpose. The motivation for this project is to integrate art, music, and technology into a single web application. The inspiration for this project derives from a passion for classical music and the imagery it provides. The problem is not a problem, but a need for innovation and imagination to apply to a genre of music.



Previous and Related Work

The very first conceived audio visualizer is credited to Heinrich Rubens and his invention of the Ruben's Tube. In 1905, Ruben utilized a "flame tube" to generate a spectrum analyzer. The sound waves were used to alter the pressure of the gas — creating physical visualizations utilizing sound data (Ashton 1).

Figure 2: Atari Video Music [2]

Atari's Bob Brown created the first digital music visualizer in 1976 known as the Atari Video Music (Atari 1). The AVM allowed users to hook up their hi-fi stereo systems and the TV to produce visualizations on the television set. Atari, like most modern solutions, provided various buttons and knobs to create custom effects on the visualizations.

The most common solution can be found in an everyday music player application such as iTunes or Windows Media Player. These applications utilize primitive animations to portray audio analysis. These primitive animations provide detailed visualizations of the applied audio analysis. These animations usually consist of waveforms, lines, bars, morphing shapes, and various colors. Each of these options utilize varying forms of beat detection in order to change the wave, color, or bars at appropriate times.

Audio analysis incorporates a variety of methods. Observing the amplitude, frequency, and the varying amplitudes of each frequency are required for beat detection (Turner 1). Felix Turner's Audio Analysis demo guided the beat detection algorithm used in this music dynamic animation studies. First, track threshold volume and amplitudes. Then, when the current volume threshold is exceeded the new threshold is set and a "beat" is detected (Turner 2).

Animation

Basic animation considers numerous factors. The 11 fundamental principles discussed during the 1987 Siggraph Conference in Anaheim include: squash and stretch, timing, anticipation, staging, follow through and overlapping action, straight ahead action and pose-to-pose action, slow in and out, arcs, exaggeration, secondary action, and appeal (Lasseter 34). All of these principles dictate an objects distortion and reaction based in realistic elements involving physics and time. Motion is achieved utilizing these fundamentals and creating a visualization that simulates motion (Lasseter 34).

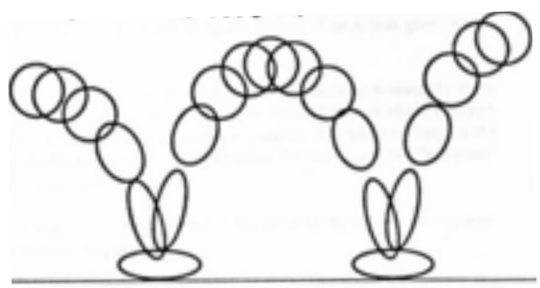


Figure 3: Squash and stretch (Lasseter 36)

In 2D animation, creating specific keyframes are important in having the right track of motion and object distortions and responses. A good animation, 2D or 3D, utilize these basics in a manner where the presentation of an idea is clear. The scene should focus solely on the action the artist wants the audience to focus on (Lasseter 38). The intelligent and proper application of all the principles will create personality. The main goal of animation is to entertain and these principles provide the tools to do so with the best quality in mind (Lasseter 43).

Target System

This project was intended for deployment on the internet and utilizes basic HTML5, CSS3, and extensive JavaScript. The project imports JavaScript's minivents for graphical user interface usability as well as jQuery 1.9.1. Custom audio and visual JavaScript files were written for audio and visual analysis and production. These JavaScript files were written with reference to Airtight Interactive Inc.'s Audio Analysis (Turner 1). This web project works best with the Google Chrome application.

Algorithm

Instrumental music is the main focus of this project and the analysis and animations are focused on classical music. There is a sample audio upon initially opening the web application and it can also take audio files by dragging and dropping onto the webpage. Music dynamic animations utilize the same concept of beat analysis. The idea of amplitude threshold is key. However, unlike modern music, instrumental songs tend to lose this distinction without having digital music elements such as a bass and pads. As such, a beat cutoff is initialized as well as a minimum beat. The beat cutoff is a defined value establishing the largest amplitude in a song based on rhythm and tempo which updates as the song plays. This beat cutoff can be clearly established in contemporary music using distinguishing instruments and sounds such as a bass drop or a drum high hat. In classical music, a beat is defined by the time signature. A time signature tells a musician how to play the song — how many beats per measure. As such, a rhythm and tempo are defined, but the beat is difficult to differentiate. The predefined beat cutoff in this project indicates a standard 4/4 time signature.

By initializing a beat cutoff for classical music, which tends to be lower intensity than contemporary music, a max was initialized. The beat cutoff is updated each time the beat is found at higher amplitudes while the song is playing. Each time the beat cutoff is updated, there is a value for half of the beat cutoff. In addition, a beat minimum is set in order to create a larger number of ranges. This provides three variable values: level minimum, half beats, and the beat cutoff. These variables help create four major beat ranges used to determine which animation is best suited based on tempo and amplitude. The four major beat ranges are: below minimum, larger than minimum but smaller than half the cutoff, greater than half the cutoff but below the maximum, and larger than the maximum.

Three different sets of animations, meaning three different stories, were hand drawn to graphically represent the three major tempos in the beat analysis. The three major changes are when the level exceeds the beat cut off threshold, when the level drops below the half beat cut off value, and when the level drops below the minimum beat. The animations are drawn frame by frame using Adobe Flash Creative Cloud and exported into video files played at ten frames per second. This same frame per second playback speed will mean that multiplying it based on the current tempo level of the song will change how fast the video plays, thus differentiating the speed between videos.

Summary

This project tested the quality of the playback of animations using two differing instrumental songs: Rain composed by Brain Crain, and Vivaldi's shortened Winter of the Four Seasons series. The animations playback at an appropriate speed to the song and changes as the tempo changes. Currently, to prevent cutting off the animations halfway through, there is an event handler that detects when the video ends and then switches the video source.

Figure 4: Tree animation screenshot

Results

Music dynamic animations are indeed a challenge. At first, the animations appeared to switch with the different ranges as expected. When testing using a different song with a faster tempo, the animations would periodically stop playing because the frequency and amplitude levels were constantly switching between the ranges. This was fixed using a margin buffer when comparing the levels to the ranges. For example, if the song level was lower than the minimum or larger than the cut off by a certain margin, then it will switch. In addition, an event function would be triggered once the video is done playing through, thus allowing the animations to finish playing before switching.

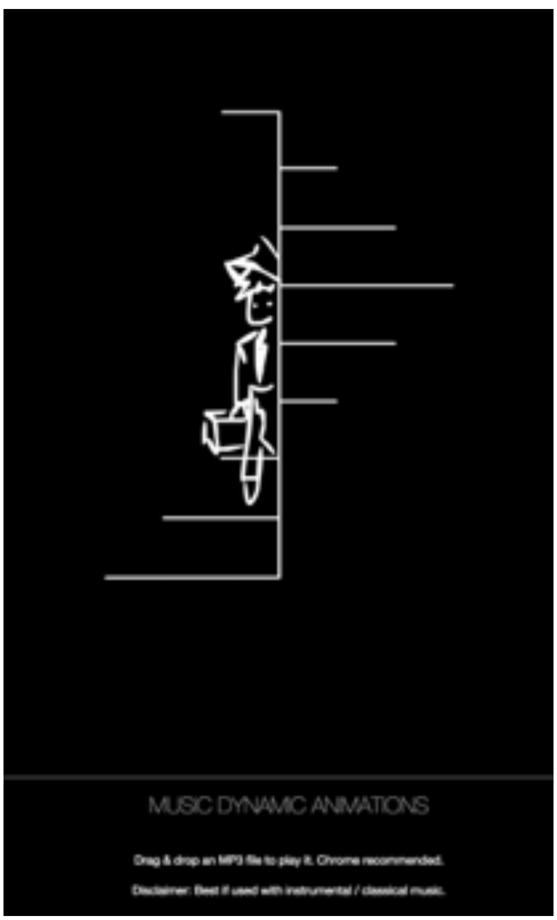


Figure 5: Running man animation screenshot

Figure 6: Dancer animation screenshot

When listening to the song playing the animations appear should play at an appropriate time based on the tempo. The tree animation as shown in Figure 4 is used for slower and quieter portions of songs. The man running down the stairs in Figure 5 is used for the faster and louder portions. The dancer in Figure 6 is used for a portion in the song where it comes to a lull, a nice in-between in the song. At first, even though the animations played at the right times, the animation playback themselves seemed incorrect, such that the tree would play faster than the slow parts of the song. This was corrected by multiplying the playback speed and the tempo level by 4, under the assumption that the song playing would be performed in a 4/4 time signature.

Future Work

The next stage would incorporate numerous processes. Because this project utilized a very limited animation library, the first step would be expansion. Expanding the library of animations allows for more graphical representations and a greater sense of imagery — the very purpose of this project. The next step would be to have the numerous animations play at appropriate times using a system other than the beat ranges or a more detailed version. Ideally, instead of just analyzing the sound realtime and changing the animations based on when the song changes during playback, analyzing the sound ahead of time would create an appropriate buffer to accurately determine which animation is best played.

Another form of expansion would be instrument analysis. Because this project focuses more on classical and instrumental music, being able to determine the notes played per instrument would provide a lot more information as to what kind of animation could accurately portray the feeling of the song.

Conclusions

Two dimensional animation requires patience, passion, and observation. Drawing each scene frame by frame to determine which poses are key to fully capture the movement of the character require knowledge of the motion and the physics behind actions.

Analysis of classical music requires a similar algorithm to that of contemporary music with expectations of a lower amplitude and a discreet beat. The best way to capture the beat is to capture the tempo and the rhythm through the notes heard and the speed they are played back.

Timing animations to play back at the appropriate speed is easily alternated utilizing the HTML video element's playbackRate variable. Multiplying this variable by the normalized frequency and calculated tempo multiplied again by four for a standard 4/4 time signature made the video playback noticeably changing to the beat.

Overall, this project applied left and right brain skill-sets, appealing to the artistic and the technical audiences alike.

Acknowledgements

Special thanks to Zoe Wood for advising and moderating the development of music dynamic animations. Also acknowledgement of appreciation to John Clements for advising on music analysis methods.

References

Ashton, Danny. NeoMam Studios. 2015. Web.
http://neomam.com/blog/9-awesome-music- visualizations/

Atari Museum. 2012. Web.
http://www.atarimuseum.com/videogames/dedicated/videomusic/videomusic.html

Lasseter, John. Principles of Traditional Animation Applied to 3D Computer Animation. Pixar. San Rafael, California. July 1987. Online Print. p. 35-44. http://www.cs.cmu.edu/afs/cs/academic/class/15462-f14/www/lec_slides/Lesseter.pdf

Turner, Felix. Airtight Interactive Inc., 2013. Web.
http://www.airtightinteractive.com/2013/10/making-audio-reactive-visuals/

Elements of Music. History of Music. Web. http://historyofmusic.tripod.com/id6.html

Image Links:

[1] http://www.lascells.com/perch/resources/la50-550rubenstube-1-w870h600.jpg

[2] http://retrothing.typepad.com/photos/uncategorized/2007/09/04/atarivideomusic.jpg

Project

To view the project, please visit: thuyvydo.net/seniorproject.html
* Note: Project website may or may not be active