Understanding The Relationships Of User-selected Music In Video Games

A Senior Project

presented to

the Faculty of the Liberal Arts And Engineering Studies

California Polytechnic State University, San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science

by

Memphis Wong

June, 2015

Table of Contents

Memphis Wong

LAES 462

11 June 2014

"Understanding the Relationship of User-selected Music in Video Games"

I. Introduction

Music has been an integral part of video games since shortly after its inception. Beginning as basic audio feedback, it progressed from use as a device to convey information to the player, information like hitting a ball in Pong and exploding an asteroid in Asteroids. This use of audio extended and shifted to background music and further use in thematic setting within video game worlds. A prime example of this is music within the games of the Final Fantasy Series. The Final Fantasy Series is one of the more popular and prominent set of turn-based video games, with its music still being heralded and celebrated. Music from the series helped form the iconic and unique fantasy-like setting and gave it a distinct identity. Moving forward from just thematic setting, modern games promote the importance of music to becoming entwined within the gameplay and core game itself. Games such as Guitar Hero have levels designed specifically for music. Using a selected list of popular licensed songs, the game contained preset levels of markers that corresponded to buttons players had to press. The aim of levels were to simulate playing the songs on a guitar. The most recent change of the adaption of music in video games has been the real-time generation of levels based upon the music of the game and also allowing users to select their own music that is used in the game. In Audiosurf, the game utilizes these two

key elements. Users ride through a flattened track with blocks designating key points in the user's selected audio. The number of potential levels that the game can design off of various music tracks is practically endless. The game has won numerous awards and has spurred a new niche of music focused games. But how much of the gameplay experience that is so highly rated is due directly to the user being able to select their own music?

II. Project Deliverables

To answer this question of how much impact allowing users to select their own music really is, my senior project aims to design a game that allows for both user selection of music and real-time level generation using music. I utilized two main bodies of work in my project. The first body of work involves developing a platform to both test and capture data on how allowing users to select music in a video game will affect gameplay experience. The second is a follow-up statistical analysis of the collected data. In conjunction I hope that these deliverables will be sufficient in coming to a conclusion on the impact.

III. Literature and Technology Review

In reviewing existing projects, I did not find any comparable studies. The research I saw was geared more directly at music in video games in general and not directly with whether or not a user could directly choose their music. The titles I found were "A Case Study on the Effectiveness of the Music Game in Elementary Students' Music Appreciation Learning." from the Proceedings of the Serious Games Conference and "Music-games: A Case Study of Their Impact." from the Research Studies in Music Education journal. In this

regard I find that my study is fairly unique in its nature. While not with other studies, my senior project is very closely related to games that do already utilize level generation based on music. Some of these games include: Audiosurf, Beat Hazard, Def Jam: Icon, Raycatcher, and Symphony. The games varied from levels that were entirely generated prior to the beginning of gameplay to levels that were generated as the user was playing. The video game I developed shares a lot of components featured in these games, but is the type that generates the levels as the user is playing. Some of these features involve pacing and the fundamental generation selection of levels determined by variances in selected music. It shares the objective of providing a gameplay experience that reflects the given song.

IV. Technology Overview

Technologies that were used in my senior project were primarily a game development engine/ide, a source to distribute the game, and a form that would allow for data collection. For game development, I decided to use Unity, primarily due to its free licensing and my familiarity with the coding language C#, a scripting language supported by Unity. The game I developed was in the preset Unity2D development environment. I created the art assets as simple blocks for clarity and utilized a free and open sourced music file for my control group.(http://freemusicarchive.org/music/broke_for_free/) The ditribution source that I used to allow users to download and play the video game I developed was through Dropbox. There I was able to let users download standalone .exe

files generated from Unity, as well as provide the informed consent forms to the users. Dropbox was selected because it had the largest user base already installed at the time. Sharing via Dropbox was more prevalent than other competing services such as Google Drive and OneDrive. The electronic forms that were used to survey individuals were Google Form surveys. Google Forms were selected due to their ease of setup, with me using simple 0-5 scale questions that were bubble selections. Also, Google Forms is much more common and familiar to the average user.

V. Implementation and Timeline

The aim of the first deliverable of the project was to be able to test gameplay experience depending on if the user could select their music or not. To test gameplay experience, I developed a two-dimensional platformer video game that generates levels based upon user-selected music. The game would eventually be developed in Unity and was used to collect data on user experiences. How the game collected data would involve a fifty percent random chance of allowing the user to either select their own music or have a controlled song that was already selected. Depending on whether the user was able to select his or her music, they would then directed to one of two surveys on Google forms. The aim of this sort of data collection and randomness would be an effort to provide a control and an experimental group. Comparing these two groups, I could use the data in an attempt to draw some conclusion on the effects of allowing a user to select their own music in a game. The statistical analysis I aimed to use would be a null

hypothesis test. A hypothesis test that would assume a null hypothesis of the two groups having the same rating of gameplay experience, regardless of whether or not they were able to chose their song while playing the game. Here are the survey questions participants were asked:

Survey Questions

How frequently, on average, do you play video games? *
(0) Never
(1) Couple Times A Year
(2) Couple Times A Month
(3) Couple Times A Week
(4) Every Other Day
(5) Every Day

Rate how much you enjoyed this game *
(1) Terrible, would never play again
(2) Passable, I was not very entertained
(3) Okay, average gameplay experience
(4) Good, could be very fun, given improvements
(5) Excellent, would play again

How well did you enjoy the music? *
(1) Terrible, wanted to stop listening
(2) Bearable, was ok in small amounts
(3) Okay, average song
(4) Good, liked listening to it
(5) Excellent, could be one of my favorite songs

How well did the music enhance the game? *
(1) Terrible, it made it worse
(2) Passable, did little to enhance
(3) Okay, average background music
(4) Good, music enhanced gameplay well
(5) Excellent, was much more fun experience, because of music

The original timeline spanned from the beginning of the year in January through June and went through a large number of changes. Initially, I had planned to create an android application developed through Android Studio. The greater ease of use in developing a platformer is what led me the most to switch to Unity. Among other changes, understanding the complexity of manually decrypting an mp3 file led me to use frequency analysis instead and, thus, develop a game with real-time generation of levels. The algorithm I put in place to analyze music files was a combination of volume and frequency analysis. The algorithm carried a running average of the volumes between three sections of frequencies in songs between zero hertz and twenty-thousand hertz. I had the sections separated at zero to three-hundred hertz, three-hundred-and-one to eight-thousand hertz, and finally eight-thousand-and-one to twenty-thousand hertz. When the volume at a certain point in the song was higher or lower to a set degree of the average volume at a given frequency, the algorithm would assign a point value to decide where the platform would be placed. Limiters were used to limit the number of repeating patterns that would be generated, in an effort to reduce repetitiveness that might lead to a worse gameplay experience for users. Pacing was established on the number of significant changes above and below the average volume at the lowest frequency section. Here is a brief overview of the algorithm implemented in the project.

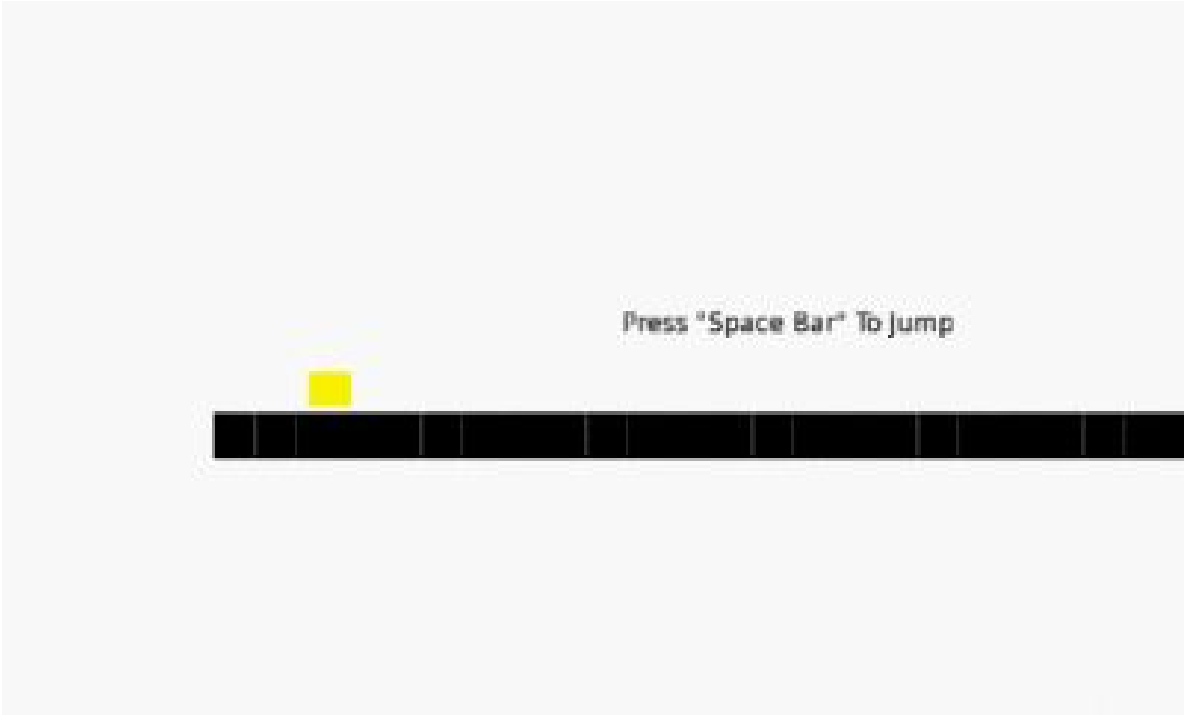
```
//called 2 times every second
Function {
        currentLowFrequencyValue = **volume at frequency <300hz**
        currentMidFrequencyValue = **volume at frequency >300hz && <=8000hz**
        currentHighFrequencyValue = **volume at frequency > 8000hz && <= 20000hz**

        **lowRepetitionCap and highReptitionCap starts at .1 and is carried over between function calls**
        **decisionValue starts at 0 each instance**
        //decisionValue > 1 places next platform relatively higher
        //decisionValue < 1 places next platform relatively lower
        //decisionValue otherwise places next platform at equal height

        //ExecutedforEachFrequencyValue
        if(currentValue > (1 + highRepetitionCap) * averageValue) {
                decisionValue + 1
                highRepititionCap + .1
                lowRepetitionCap = .1
        }
        else if(currentValue < (1 - lowRepititionCap) * averageValue) {
                decisionValue - 1
                lowRepetitionCap + .1
                highRepititionCap = .1
        }
        else {
                lowRepetitionCap = .1
                highRepetitionCap = .1
        }
}
```

Sampling rate of the music was at twice per second. The starting averages were established at the beginning of the first eight seconds of each play and music track. During this time the player block would be moving across a long flat preset Once those were found, the next sampled instance in the song would determine the placement of the first platform off the starting point. In development, several unique problems arose and were addressed. One of the issues Unity had was the lack of native support for file browsing through a computer's directory to allow users to find and select music files. This problem was solved by importing a free Unity asset from the Unity Asset store and applying its API into the video game. Problems importing this asset involved how it was

created in an outdated version of Unity and needed some manual updates. A second problem was that Unity restricted importing and playing mp3 files. The solution was to adjust the supported music files to .WAV file-type. A problem with using .WAV music files was how uncommon it was for users to have that selected file-type and that the size is much larger than .mp3 files. Users were asked to convert their music files to .wav during the studies. The final timeline that was carried out was full-development of the algorithm and game in unity, then deployment to users that would be participants in the study, and finally statistical analysis. The resulting platformer looked as follows:



VI. Analysis And Verification

In my senior project study, I looked for participants to play my game and record their subsequent responses in the project surveys. The study was approved by the IRB/Human Subjects Committee and participants were obtained through Facebook, Reddit, and personal messages. There ended up being 10 in the control group, where they were given the preset song, and 14 in the experimental group, who were allowed to select their own music. The statistical approach I used was hypothesis testing. My null hypothesis was that “Both groups will rate the game the same, regardless of whether they were allowed to choose their own music”. I used an alpha, significance level, of 0.05. The average game rating of the control group was 2.9, with a standard deviation of 0.567. The average game rating of the experimental group was 4.07, with a standard deviation of 0.730. Initially, I used a calculated Z-Score of 2.06349 and a P-Value of 0.01953 to reject the null hypothesis. However, upon understanding that my sample did not fully meet the assumptions for a Z-Test, I used a T-Test to analyze my results. I was able to calculate a T-Score of 4.232 and a resulting P-Value of 0.000171. Given such a small P-Value, even on the calculated T-Test, we reject the null hypothesis and accept the alternative hypothesis. The alternative hypothesis is that allowing users to select their own music does affect how they will rate the game. Obtaining an average rating of 3 for my game, getting a sample size greater than twenty people, and being able to successfully obtain a statistical analysis were the verification standards I used to judge project success. Ultimately, the main goal of the project was to produce a game and

conjoining study that could help adequately determine the impact of allowing users to select their own music in video games.

VII. Societal Impact

This senior project study is an attempt to focus on aspects of video games that are not as widely discussed and study as I believe they should be. The societal impact that this project may have is to address these aspects of music and user selection/choice in video games. Information from the study may help spur interest or help others build on future projects and studies on the subject. But beyond that, the societal impact of the subject may extend to psychological topics and analysis. My study asserts that there is a clear and positive correlation with user selection and user experience. What this may translate to, is how control for people in certain aspects or experiences may be beneficial. I was able to find a related study on this potential impact in “Perceived Control and Coping with Stress” from the Journal of Social Issues Volume 47, Issue 4, pages 23–34, Winter 1991.

VIII. Future Work

Future work for my senior project includes refining and improving the algorithm I created and used to generate levels. While it does do a fairly good job at representing music and variance, it can still be much better. With that, the game that I designed itself can use additional features and fixes in streamlining user experience. If possible, the game might be better served to be created using another development tool that can allow for use of mp3 files rather than WAV files. Finally, additional work in the study will add strength to the overall study and potentially find many factors and relationships that may have been missed in this study. Increasing the sampling size and a more refined statistical analysis and approach

would greatly benefit the senior project statistical analysis and study. These are steps that could be taken by any future students that wish to pursue this topic.

IX. Conclusion

The results of my senior project study were significant, but could use more sampling and analysis for a stronger argument. In total, I found that users who were able to select their music rated the game statistically higher than those who were unable to select their music. While this may mean that choosing music is vastly important for games that revolve around gameplay generated by music, it could also extend to the general idea that allowing for more user control increases the enjoyability of the experience. Users from my study seem to be far more likely to rate a game higher when they have control over the aspect of music selection. From this conclusion, I believe that future applications can and should be more catered towards users' ability to control more aspects of experiences in video games, as well as potentially other user applications.

Works Cited

"Android: Build an MP3 Player." *YouTube*. YouTube, n.d. Web. 15 Jan. 2015.

"Android Building Audio Player Tutorial." *Android Building Audio Player Tutorial*. N.p., n.d. Web. 15 Jan. 2015.

“Audiosurf.” Dylan Fitterer. 2008. Video game.

"Blog.bjrn.se." *Blog.bjrn.se*. N.p., n.d. Web. 15 Jan. 2015.

Cassidy, G. G., and A. M. J. M. Paisley. "Music-games: A Case Study of Their Impact." Research Studies in Music Education (2013): 119-38. Print.

"Inside the MP3 Codec - MP3 Anatomy." *Inside the MP3 Codec - MP3 Anatomy*. N.p., n.d. Web. 15 Jan. 2015.

“Guitar Hero.” Harmonix. 2005. Video game.

"Managing Audio Playback." *Managing Audio Playback*. N.p., n.d. Web. 15 Jan. 2015.

Park, Seungie, Daewoong Rhee, and Taesuk Kihl. "A Case Study on the Effectiveness of the Music Game in Elementary Students’ Music Appreciation Learning." Proceedings of the Serious Games Conference 2014. Print.