Attack of the Beat

Project Members: Sean McManes

Advisor: Professor Chia Han

April 17, 2020

**Table of Contents**

Project Description/Abstract ?

User Interface Specification ?

Test Plan ?

Test Plan Results ?

User Manual ?

Final PowerPoint Presentation ?

Final Expo Poster ?

Self-Assessment Essay Fall: Sean McManes ?

Self-Assessment Essay Spring: Sean McManes ?

Summary of Hours ?

**Project Description**

The goal of this project is to create a game that is an enjoyable experience for the player by challenging them, creating an interesting game environment, and has game mechanics that have not been fully explored by other games.

The main aspect that would make this game stand out from others is the use of game music that would manipulate the gameplay and environment. This would make the player not only have to look for visual cues in the game, but also audio cues in the music in order to have an advantage and complete the game.

Current games typically use the game music to set the tone of the world/gameplay. For example, creating a suspenseful setting for a scary/horror game with low to no sound, or playing exciting music in an action filled game to get the player immersed in that game. Some games have adapted their gameplay to the music or vice versa, but very few have directly used the music in order to modify the game experience. The ones closest to doing this have stopped short and only do basic implementations, such as movement or attacks based on a constant beat of a generic song.

For years, we have been invested in the game industry and have had experience with several game genres. Most of the enjoyment we found in these games stem from new experiences that the game present to the player. These experiences include new and interesting game mechanics or new interpretations to prior game mechanics, interesting world design, natural game controls and visuals, and more.

Our approach to this problem would include the following:

* Creating a tool/library that can analyze the game music that is being played and output the data retrieved to the running game or store it to be used when needed
  + The data stored would most likely be the frequencies found at various times in the music
* Create a working game environment that has an interesting design, interactable objects or items for the player, non-playable characters, and more to fill this environment
* With the analyzed music and working game environment, then aspects of the game can be manipulated by the data found. Some of these aspects include:
  + Enemy/game actions, such as having enemy attacks based on the frequencies of the current music
    - The type of attacks, power, and other aspects can be affected by this as well
  + Game pace can be affected by the type of music being played, with slower song making the game pace slower and fast, high energy, songs would make the pace faster and perhaps make the game harder

Aspects of the world design could also be affected by the song, with lighting and room design changing with the music

**Abstract**

Current games use several aspects to immerse the player and make enjoyable experiences, but they do not use the game music beyond atmosphere. This project aimed to make a game with mechanics manipulated by music so the player must listen to the music along with visual cues to succeed. This was accomplished with object manipulation to visually mimic data acquired from continuous music analysis.

**User Interface Specification**

T

**Test Plan**

Part I: Description of Overall Test Plan

The approach taken to test this project will take place on an individual parts basis and then an overall complete project basis. Throughout each test, particularly with the individual parts, depending on what results need to be examined, the project will produce data necessary for the tests. This data will be analyzed and compared to the intended results, and from that conclusions will be drawn, and further improvements/changes will be determined from the findings. After these parts are working as intended, they will be put together towards the overall project and then final tests can be conducted to make sure that the parts work together as intended, and the overall project is working as intended. Any improvements/changes would be made if needed depending on the outcomes.

Part II: Test Case Descriptions

1.1 Game Map and Texture Test 1

1.2 This test ensures that the first round of map textures and map generators work as intended

1.3 This test will run the map generator with the first round of textures to show that the generator is picking the correct textures for the given test cases, and that the textures are correctly designed to work with intended situations.

1.4 Inputs: First round textures and test map layout

1.5 Outputs: Map made from textures given based on map layout

1.6 Normal

1.7 Whitebox

1.8 Functional

1.9 Unit Test

2.1 Player Texture and Animation Test 1

2.2 This test ensures that the initial player textures and animations work as intended

2.3 This test will have the player character complete various test actions to show that the textures flow together to make the animations look correct. This will also test the timing between each part of the animation, and add/remove time if needed

2.4 Inputs: Player textures and test actions

2.5 Output: Player animation matching the given actions

2.6 Normal

2.7 Whitebox

2.8 Functional

2.9 Unit

3.1 Game Map and Texture Test 2

3.2 This test ensures that the game map generator handles exceptions as intended

3.3 This test will run the map generator with the first round of textures to show that the generator correctly identifies errors in the given map design, displays proper warnings, and displays proper textures in the areas that have errors.

3.4 Inputs: First round textures and test map layout with errors

3.5 Output: Map made from textures given based on map layout with proper error messages and error textures

3.6 Abnormal

3.7 Whitebox

3.8 Functional

3.9 Unit

4.1 Music Analyzer Test 1

4.2 Run the music analyzer over test music

4.3 This test will run the music analyzer over the first set of test music and see if the output from the analyzer match what was expected. The major part of this is making sure that the analyzer is picking good points in the music for specific actions, and any issues would be adjusted.

4.4 Inputs: Test music (Set 1)

4.5 Outputs: Action codes with timestamps from trigger points

4.6 Normal

4.7 Whitebox

4.8 Functional

4.9 Unit

5.1 Music Analyzer Test 2

5.2 Run the music analyzer after modifications over different test music

5.3 This test will run the music analyzer over another set of test music and see if the output from the analyzer match what was expected. This is to see if the adjustments made from the prior test cause incorrect outputs from other songs, and depending on the outcome, some tweaks to the design may be done. Generally, the type of music will be the same.

5.4 Inputs: Test music (Set 2)

5.5 Outputs: Action codes with timestamps from trigger points

5.6 Normal

5.7 Whitebox

5.8 Functional

5.9 Unit

6.1 Alternative Music Analyzer Test

6.2 Run the music analyzer over different genre test music

6.3 This test will run the analyzer over another set of test music from a different genre from the music from the other tests. This is to see if the settings made in the prior tests work well with other genres of music, and if not, then what adjustments can be made to get the analyzer working

6.4 Inputs: Alternative test music

6.5 Outputs: Action codes with timestamps from trigger points

6.6 Normal

6.7 Whitebox

6.8 Functional

6.9 Unit

7.1 Player/Object Movement and Actions Test 1

7.2 Test the various movements and actions of the player character and other objects

7.3 This test will ensure that the player movement and actions are working as expected and that there are no disconnect or lag with them. This test will also allow us to adjust settings, such as how far characters move per frame and so on.

7.4 Inputs: Test user input (movements and actions)

7.5 Outputs: Displaying player movements and actions

7.6 Normal

7.7 Whitebox

7.8 Functional

7.9 Unit

8.1 Gameplay and Music General Test 1

8.2 Test the game loop with music implementation

8.3 This test will ensure that the music playing with the game is outputting the action codes, as tested before (as a separate part), at the correct times and does not run into issues over time. Major points for investigation are desync from the game music, stuttering, incorrect actions, and more.

8.4 Inputs: Test music and generic game inputs

8.5 Outputs: Game display, music, and corresponding action codes

8.6 Normal

8.7 Whitebox

8.8 Functional

8.9 Unit

9.1 Player/Object Movement and Actions Test 2

9.2 Test the various movements and actions of the player character and other objects with music

9.3 This test will ensure that the music, action points gained from the music, and object movements from these action points are working in sync and as expected. Any issues that arise with the implementation of these action points and actual object movement can be addressed at this point.

9.4 Inputs: Test music

9.5 Outputs: Displaying object movements and actions, music, and action codes with timestamps

9.6 Normal

9.7 Whitebox

9.8 Functional

9.9 Unit

10.1 Gameplay and Music General Test 2

10.2 Test the game loop with music implementation and actions from the music

10.3 This test will ensure that the music playing with the game is outputting the action codes, as tested before (as a separate part), at the correct times and does not run into issues over time. This will add a player-controlled character, and several computer/music controlled characters to mainly stress test the game and make sure that these elements do not create conflicts.

10.4 Inputs: Test music and player test input

10.5 Outputs: Game display, music, and corresponding action codes

10.6 Normal

10.7 Whitebox

10.8 Performance

10.9 Unit

Part III: Test Case Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Normal/Abnormal** | **Blackbox/Whitebox** | **Functional/Performance** | **Unit/Integration** |
| **1** | Normal | Whitebox | Functional | Unit |
| **2** | Normal | Whitebox | Functional | Unit |
| **3** | Abnormal | Whitebox | Functional | Unit |
| **4** | Normal | Whitebox | Functional | Unit |
| **5** | Normal | Whitebox | Functional | Unit |
| **6** | Normal | Whitebox | Functional | Unit |
| **7** | Normal | Whitebox | Functional | Unit |
| **8** | Normal | Whitebox | Functional | Unit |
| **9** | Normal | Whitebox | Functional | Unit |
| **10** | Normal | Whitebox | Performance | Unit |

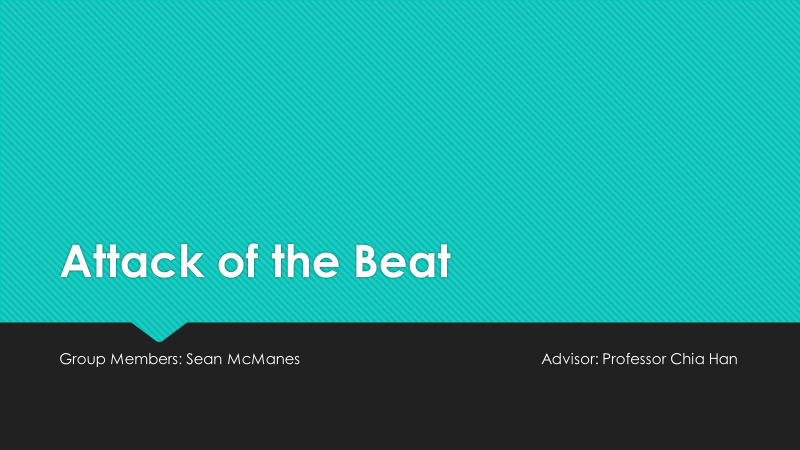
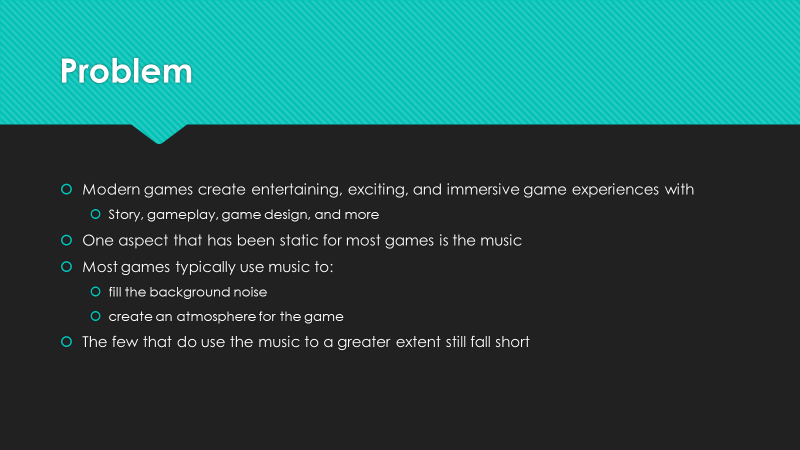
**Test Plan Results**

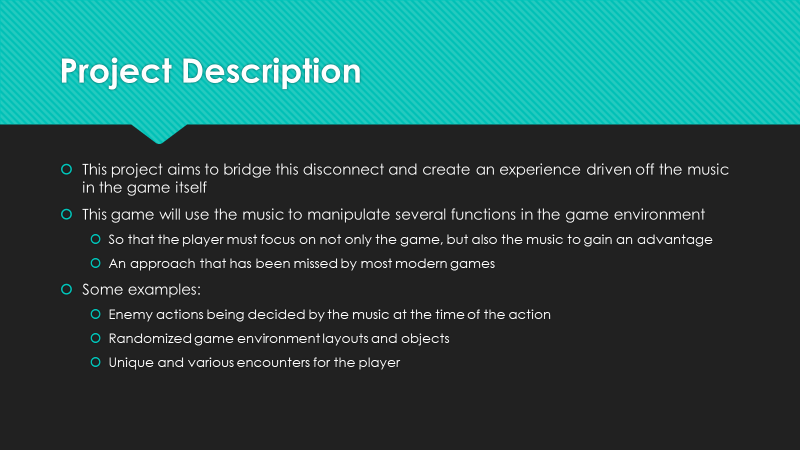
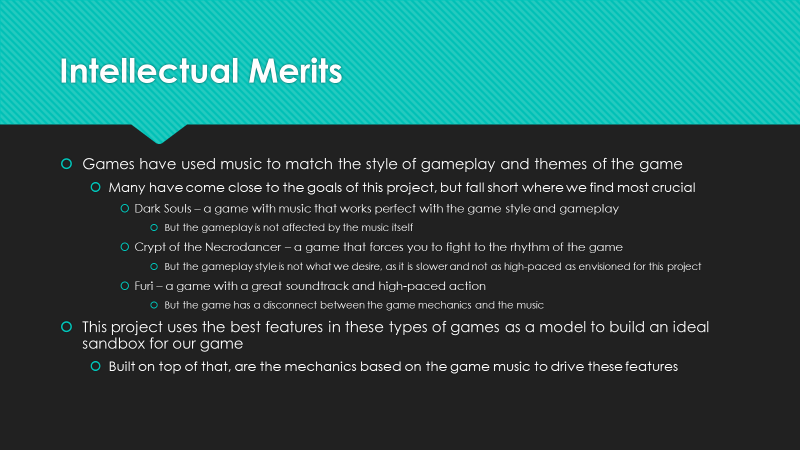
T

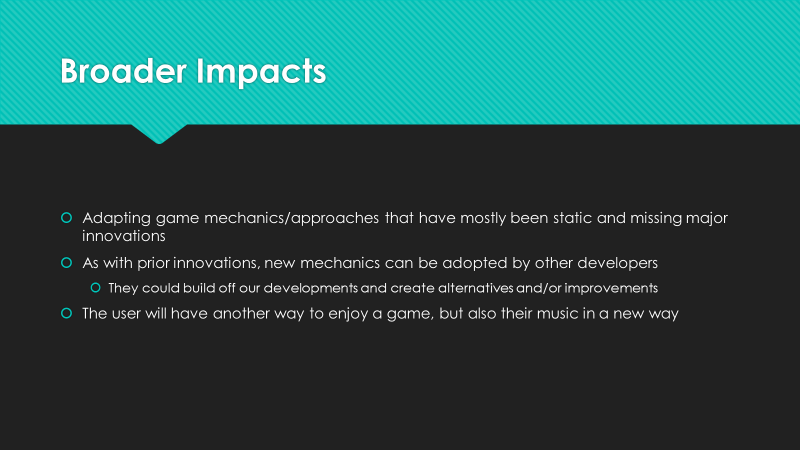
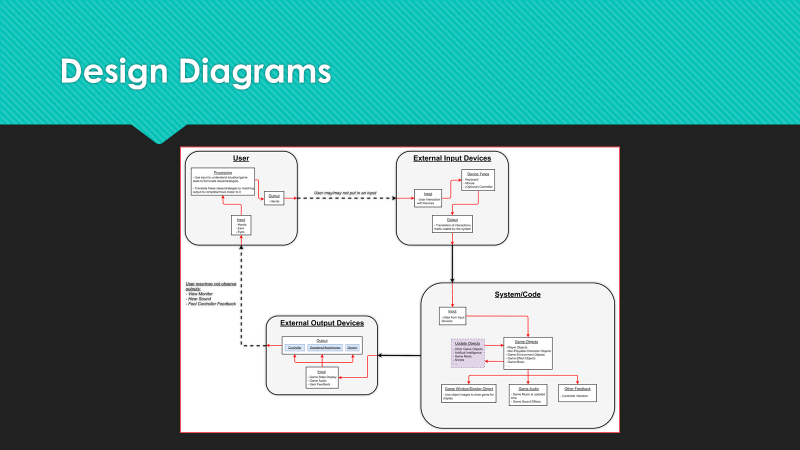
**User Manual**

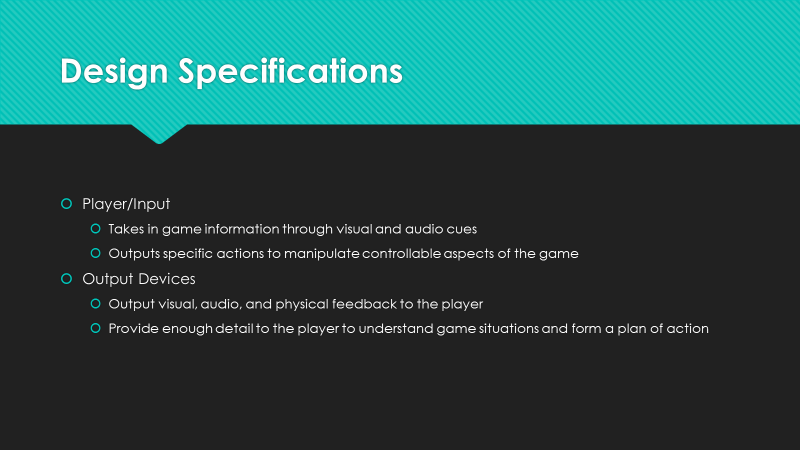
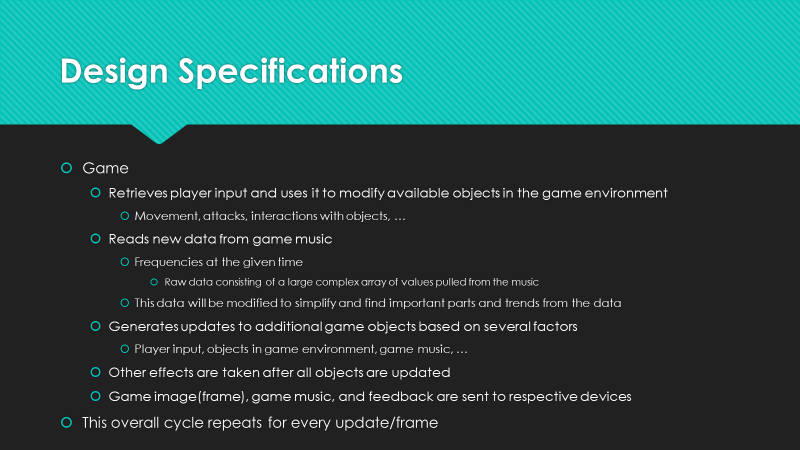
T

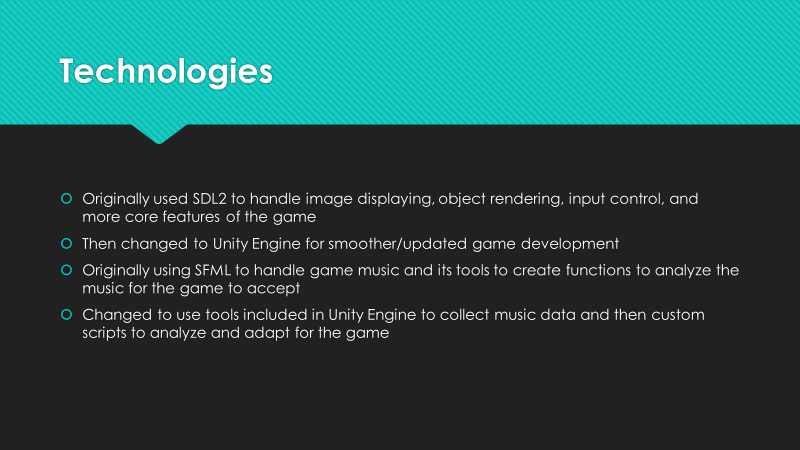
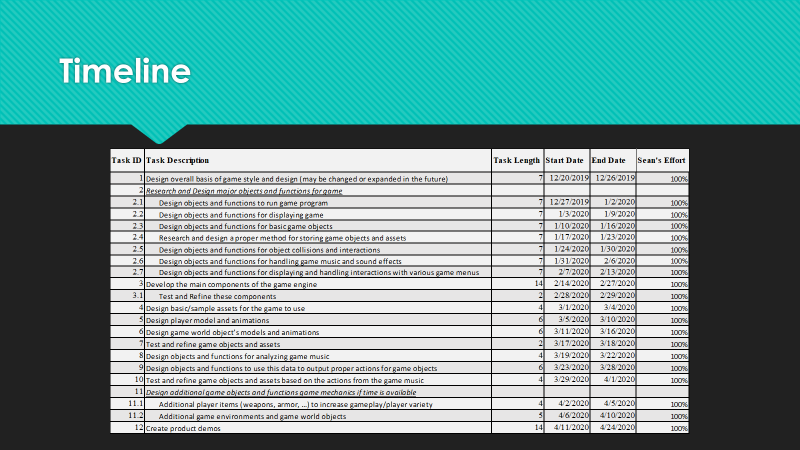
**Final PowerPoint Presentation**

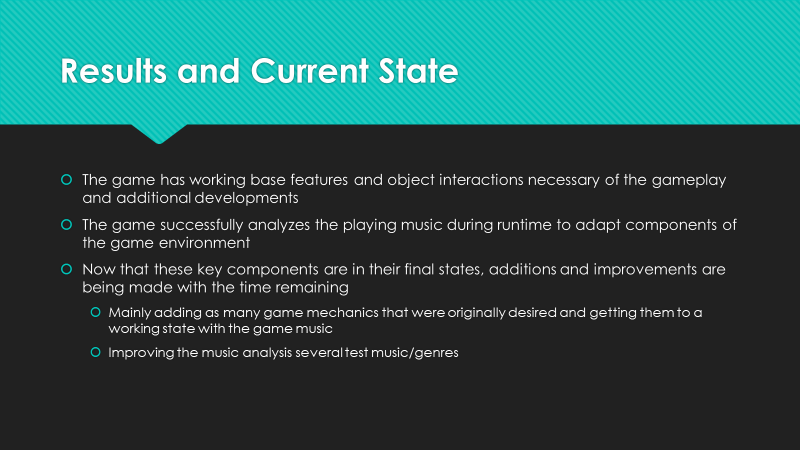
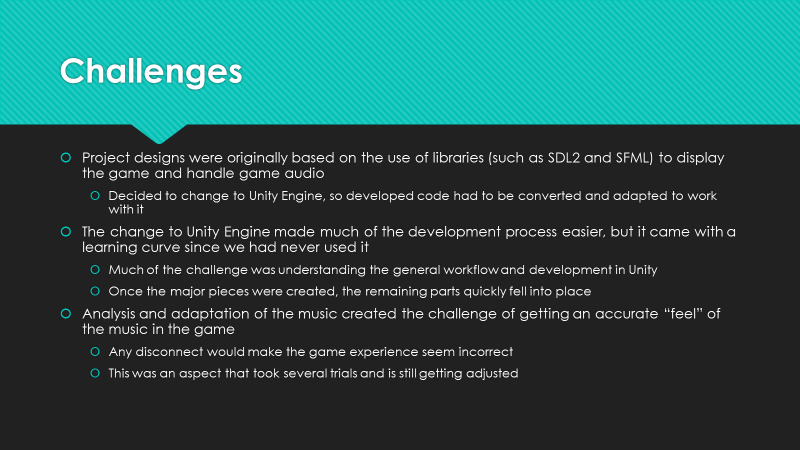
 

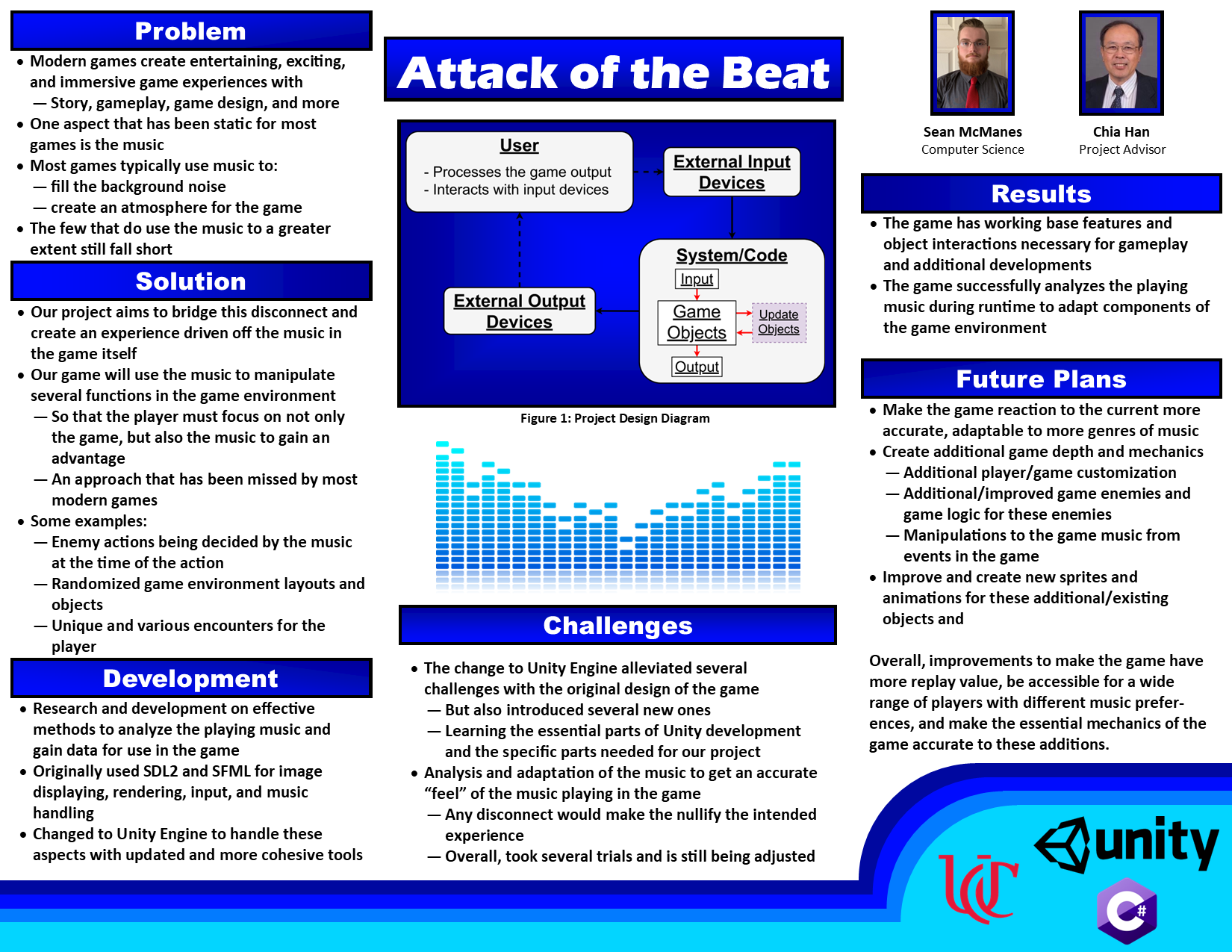
 

**Final Expo Poster**



**Self-Assessment Essay Fall**

*Sean McManes*

Attack of the Beat is a project dedicated to creating an entertaining, exciting, and immersive game experience for the player and puts an emphasis on the music in the game. This music will manipulate several functions in the game environment so that the player must focus on not only the game itself but also the game music to gain an advantage. This aspect of gaming has not been approached in most modern games. These games typically use their music to fill the background noise of the game and do almost nothing more than create an atmosphere for the game. The music does not have a direct affect in the game experience. Our project will attempt to bridge this disconnect and create a game experience driven off the music in the game itself.

The experiences that we gained during our academic career will assist our project development immensely. Considering I have not completed any co-op semesters so far, most of my experience comes from the courses that I have completed during my time at the University of Cincinnati and from my own experiences outside of the class. Our project will involve the use of several class types, functions, and established libraries in order to achieve a functioning game. CS 2011 Introduction to Computer Science and CS 2028C Data Structures were the courses that established and expanded my knowledge of programming with C++ and the importance of object designing in a program. This project will require the use of user inputs, contain several game objects that will have to be managed and manipulated, and contain a form of storage for object information and other important data. The skills gained from these courses will especially assist with the overall designing of object types, deciding what data needs to be stored and where, and creating the necessary functions to manipulate this data and interact with other objects.

As mentioned, the project will have to manage various objects in the game environment, store portions of this data for future use, and be able to read stored data as well. This will allow the code of the project itself to stay clean of repetitive data that could be stored elsewhere and read from/written to when needed. My experience from CS 2071 Discrete Structures, CS 4071 Design and Analysis of Algorithms, and CS 4092 Database Design and Development will assist me with these aspects of the project. The objects in the game could vary depending on the scenario. In the worst case, there can be several objects that need to be handled depending on what is happening in the game. In order to do this, the object must be managed in various ways and my experience in these courses will especially help in that design process. I will be able to choose the best ways of storing objects and finding the ones that need to be manipulated based on what the search parameters are and be able to evaluate the code to know if there are perhaps better options that would be more efficient. Outside of these courses, I have experimented with ways of collecting user inputs and being able to display objects and outputs beyond what have been done during my coursework (typically just command line). For this project, the use of libraries such as SDL2 and SFML will be useful for displaying the game and both reading and outputting audio.

Most of my motivation for this project comes from my enjoyment and time spent with video games during my life. The countless experiences that I have gained from them and the communities that they have connected me with have affected several aspects of my academic career path and other aspects of my life. From these experiences, I want to be able to expand upon the aspects that I find the most enjoyable for not only myself, but also for a larger audience. During my time with video games and outside of them, I also gained an appreciation for music. Some games have great soundtracks that add to the atmosphere of the game and can get you truly immersed in the experience. Others seem to fall short or do not fully use the music to add to the gameplay itself. From my experience with both game and music I can see the disconnect between them and this became the foundation for this project.

Our first steps in reaching our goal for this project are to build a working game and game design that will be the backbone for the rest of the project. If the game itself is not interesting to the player or is not functioning properly, then the rest of the project will fall short. Once the basic game framework is set, then the aspects of the game that we want to be manipulated by the music can be decided and implemented. The major aspect we have in mind are enemy attack and movement. In order to manipulate these aspects, the music playing must be analyzed and the data from this interpreted by the game into actions for the objects in the game. These actions will most likely be based on the frequency of the sounds, and their duration, being played, but their effects in game may depend on the objects that we want to be affected. For example, certain enemies may only attack when a range of frequencies are played, and their attack power or types may depend on duration of those frequencies. After these aspects have been refined and are working properly, then we will be able to work on expanding the game’s objects further. Some of these ideas will be dependent on the time available for them, but if possible, it would be nice to have more variety in the game world, such as different area types, enemy types, game objects, and so on. Overall, we will know we are done once we have made the experience enjoyable for the player and incorporates our major ideas on what the game music can manipulate to make that experience unique for the player. We could then get feedback from other game enthusiasts to know what aspects of the game succeeded and if there are any areas that could be improved.

**Self-Assessment Essay Spring**

*Sean McManes*

The building and completion of all tasks indicated in the project timeline/milestones was completed by me. Unfortunately, my original team member was unable to take the second part of Senior Design this semester, but I will discuss this in more detail later. I applied all the skills that was identified in my initial assessment last Fall, and even gained additional skills that I had not intended at the time. The original plan was to create the game from scratch in C++, as I was most familiar with it, and build a simple engine from it and create the music analysis and game mechanics with this engine and additional libraries. However, due to the additional workload I took in order to get this project finished, I decided to use the Unity Engine to assist with some of the development. This made it possible to get a working base game to expand off with the ideas that I had envisioned. Unity also made the music analysis easily accessible, and in no time, I was able to get the major components developed to drive any additional game objects I wanted to add.

Overall, learning how to create with Unity, since I had no prior experience, was the major obstacle. But once I was past that, trying to get the music analysis to properly mimic the “feel” of the music I tested was the harder obstacle. The solutions that I came up with worked in the scenarios that I have in the game currently, but with other ideas that I had, several things would need to be changed. Specifically, being able to analyze the sound that would be played during the next frame of the game was something that I did not have time to complete, and had to be put on hold so I could finalize other components. If I had more time, this would be the next part that I would develop on, and I believe I could make even more interesting and interactive experiences for the player.

Since I was the only member of this group during this semester, all the group accomplishments were my own and are stated in my assessment above. Last semester, after completing most of the assignments myself and having difficulties getting my partner to participate in a timely manner in said assignments, I knew this semester was going to be a similar story. When I found out that my partner was not taking the course this semester, I was somewhat relieved. It is unfortunate that my experience with groupwork did not go as well as I had hoped, but I have had experiences like this before, and it has never stopped me from completing my goals. And it didn’t stop me with this project.

Even with this experience, I am far from turned away from group-related projects, and I still view this project as an overall success and highlight of my college career. If I had a team member or two with the same motivation as I do, then I believe this project could have been expanded even further, meeting the future goals I mentioned above and even more functionality that I had envisioned during the planning last semester. Overall, this project showed that motivation from all members is necessary for the whole group to be successful. As every group is only as strong as their weakest link. Once those links either removed or improved, then it is possible for the group to be fully productive and complete its tasks.

**Summary of Hours**

T