

Plymouth University

School of Computing, Electronics and
Mathematics

PRCO304

Final Stage Computing Project
2017/2018

BSc (Hons) Computing and Games
Development

Matthew Webber

10469478

The Melody Slayer

Abstract

This report is made to explain the planning and implementation behind The Melody Slayer, a rhythm-action game where you fight against bosses using common rhythm gameplay. This project was undertaken to explore the possibility of a combination between the classic rhythm gameplay, an exciting and thrilling experience for a wide variety of skill levels, and the concept of boss killing, which can bring a huge amount of satisfaction upon defeating one. By combining the two elements, this project hoped to capture both feelings into one game, while also improving my personal skills and understanding of Unity.

The report begins by defining the background and exploring the other attempts at similar concepts. The projects objectives and deliverables are explained, as well as the developmental process and the legal/social issues.

Using an Agile and Feature Driven Development focus, this report shows the management techniques used for development, alongside why they were used.

The project saw some time constraints and an overhaul of the original gameplay, however the overall project can be seen as a success with very few late deliverables and decent project management, showcasing a suitable use of sprints and stages to produce the project along with some testing of mechanics.

Finally, the post mortem overviews the reflection of the project, showcasing that the project was carried out successfully and hit all of its major objectives, but could have been improved with better time management.

TABLE OF CONTENTS

Acknowledgements

Abstract

1. Table Of Contents
 - 1.1 Statement of Word Count
2. Introduction
 - 2.1 Project Goals
3. Overview
 - 3.1 Background
 - 3.2 Objectives

- 3.3 Deliverables
- 4. Method of Approach
 - 4.1 Project Plan
 - 4.2 Version Control
- 5. Project Management
 - 5.1 Task management
 - 5.2 Regular Supervisor Meetings
 - 5.3 Testing
- 6. Legal/Social Issues
- 7. Development
 - 7.1 Stage 1
 - 7.2 Stage 2
 - 7.3 Stage 3
 - 7.4 Testing
- 8. End of Project Report
 - 8.1 Summary
 - 8.2 Changes to project
- 9. Project Post-Mortem
 - 9.1 Aspects that succeeded
 - 9.2 Aspects that failed
 - 9.3 Reflection on future improvements
- 10. Conclusions
- 11. Reference List
- 12. Appendices
- Appendix 1. Initial PID
- Appendix 2. Highlight Reports
- Appendix 3. Comparison Chart

Word Count: 7259

Code Submission:

<https://github.com/DevWebber/The-Melody-Slayer>

2. INTRODUCTION

This report covers the game, The Melody slayer, a rhythm action game based on completing songs by hitting notes to the rhythm in order to defeat levels that have a boss trying to stop you. The game focuses on the player hitting notes that fall down from the note highway in front of the player, using the buttons corresponding to each lane. Successfully hitting notes continue the song and increase your score and accuracy, while misses will damage the player. This gameplay is similar and influenced by such games as Guitar Hero, however The Melody Slayer differs in that each level (besides the first one) involves a boss that also has a unique mechanic, which usually involves distracting or deceiving the player.

This project was undertaken as mostly a personal project, as I have always been interested in rhythm games and the way they allow any skill level to feel satisfied. Casual gamers can enjoy playing along to their favourite music while hardcore gamers can go for the perfect score on super difficult songs. The moment by moment excitement as you tap along and hit every note can be thrilling, or devastating if you miss. I wanted to capture that feeling again but not without a twist. As part of my original specification, this would have been the fact you had to move to catch the notes instead of just tapping the screen however this was later changed.

2.1 Project goals

The goals of this project, along with just creating a fun game, are:

- Gain more experience using Unity
- Improve the understanding of the mechanics behind rhythm games and the ways that they are created
- Gain experience in the creation of an ambitious game project and understand how to better scope projects within time limits.

3. OVERVIEW

3.1 Background

Rhythm games are unique in the sense they can provide entertainment for a wide range of skill levels while also providing entertainment in listening and playing along the music.

This project aimed to explore the combination of the classic rhythm game mechanics and the satisfaction of fighting and defeating a boss.

There are quite a wide range and varieties of mechanics used to differentiate rhythm games. Osu!, for example, uses the entire screen and has the notes appear across the map, using the mouse and keyboard to hit the notes.

3.2 Objectives

The core objectives that the project needs to achieve are:

1. A complete experience where the player begins the level, plays the song to the very end and then returns to the menu with the ability to play another level
2. A rhythm game experience with score, accuracy and health
3. Achieving the minimum MVP, with at least one boss level that combines the rhythm game experience and a unique mechanic.

3.3 Deliverables

While the objectives underline the important aspects, the deliverables are much more thorough. As part of the design process, several core deliverables were outlined for the minimum viable product, along with several optional deliverables

Requirement Number	Requirement	Requirement Priority
1.1	The player being able to move along a horizontal axis but within limit	Removed due to complexity.
1.2	A note highway consisting of 8 lanes	Essential
1.3	Notes will be spawned at the start of the highway and move along the highway towards the player	Essential
1.4	Notes being able to be spawned at specific times during the song	Essential
1.5	A “map” that consists of an array of notes set in a structure (consisting of note type, time delay and lane)	Essential
1.6	The player being able to hit a note	Essential

	and determine the accuracy from that hit	
1.7	An end state after the song is finished to bring the player back to the level selection	Essential
2.1	UI that holds score, health and accuracy	Necessary
2.2	Score/Accuracy/Health increasing with successful hits	Necessary
2.3	Main menu with level selection	Necessary
2.4	A “boss” with a mechanic for the level	Necessary
3.1	Persistent scores between levels	Extra
3.2	Map creation framework	Extra
3.3	Scenery and models for the bosses	Extra
3.4	Tutorial level	Extra

4. METHOD OF APPROACH

4.1 Agile

Agile was used to develop this project as agile proved to align better with how the project was being managed. Agile is a development cycle that uses “sprints” (sets of timed phases) to organise tasks into sections that can be completed weekly and then reprioritized if deliverables are not meant. In other words, task and deliverables are assigned every week to be completed at the end of that week, and then the tasks are reorganised and changed based on the progress done. *Agile Alliance. (2018)*

This approach was more suitable than the more common Waterfall model because the Agile methodology focuses on constantly changing requirements based on the current work and the desired requirements, while Waterfall focuses on setting all the requirements at the beginning of the project and completing them through to the end. While Waterfall can be more easily measured, it is more prone to collapsing if delays and failed deliverables occur. Agile, on the other hand, is much more continuous and adaptive which is more suitable for a project of this scale. Being able to change the requirements and design weekly meant a product could be produced that may not necessarily have all the original requirements, but

at least functioned from the weekly sprints building up instead of having all the tasks laid out at the beginning and then ending up with an incomplete game due to unforeseen issues. The Agile development allows a flexibility not present in other development methods, hence why it was chosen.

4.2 Feature Driven Development

To complement the Agile development, FDD (Feature Driven development) was also used to complement the sprints. This method involved partitioning the tasks into specific features and tasks that would be completed each week, allowing requirements to be tackled into much finer categories than with just Agile development. This feature assignment was done at project initiation, since it was known then specifically what would be needed in order for The Melody Slayer to reach the minimum viable product.

This approach showed that sticking to a rigid plan is never a good idea especially when it comes to game development. Changes will almost always happen during development, and so planning ahead for this is the best way to manage a project with such a large risk of change.

5. PROJECT MANAGEMENT

PRINCE2 was used as well as part of the management of the project. PRINCE2, otherwise known as Projects in Controller Environments, follows the methodology of:

- Project plan: the overall timeline of the project, showing when each stage was expected to be completed
- Project Initiation Document: the initial proposal for the project (Appendix)
- Stage planning
- Risk Management
- Quality Management
- Highlight Reports
- Communication Management

In order to keep track of tasks, Trello was used to help manage the project. Using a project management tool such as Trello helped keep tasks organised and provided a visual overview of all the tasks, organised into sections based on progress and sprint.

4.4 Version Control

Github was used to manage version control, where progress was pushed to the master branch. The project would be updated and pushed to github in order to keep a continuous backup in case updates to the game caused it to break or have problems. While test builds were not kept within the repository, all documentation was also put under version control

to ensure it was continuously kept. Unity usually creates a lot of temporary files, and so a strict “gitignore” was created to ensure only the important files were saved to the repository.

4.5 Regular Supervisor Meetings

Weekly meetings were done between the 8th of February to the 22nd of March with the first 3 being conducted by the original supervisor Dan Livingstone, and the rest by the replacement supervisor Paul Watson. As part of the PRINCE2 communication management. These meetings consisted of evaluating progress and helping ensure the project was in the right direction.

6. LEGAL/SOCIAL ISSUES

This project adheres to the University ethics policy

All music used as part of this project are under CC 4.0, in which there are no restrictions to using the music as long as appropriate credit is given, which has been adhered to.

Unity was used under the free Personal Edition for educational or non-professional work. While Unity offers a Professional licence, this is only needed if the annual revenue raised by the game exceeds £100,000. As such, the personal edition suits our needs.

7. DEVELOPMENT

While the development of this project was done in sprints,, the overall development can be simplified into 3 stages that encompass the project:

Stage 1: Initial Setup, basic mechanics and framework

Stage 2: Boss Development

Stage 3: FFT and custom map creation

6.1 Stage One

The first stage of the project consisted of the framework for the core mechanics and adding the framework for the more complex features in future stages. This included the note highway and the notes, as well as the hitting mechanics.

At the beginning of the project, the overall tasks that needed to be completed were determined and the project management tools were set up, mainly Trello and Github. The list of tasks were added to the trello board, and the repository created on GitHub.

The first step to creating The Melody Slayer was adding the note highway that would form the basis of the levels. The note highway itself consists of an area of 8 lanes that dictates where each note can spawn. While the highway itself is nothing special, it stands as a core

monument for the player to see where the notes are coming from. The highway also acts as the visual barrier to show the area where the notes need to be hit, as well as when. 8 lanes were chosen due to the symmetry of an even number of lanes, and also research into other rhythm games found that having between 5-9 lanes is usually suitable.

Once the confines the movement along the lanes for the player. While this feature did not make it into the final release, it was a core part of the original conception, and as such needs to be discussed. The player was created as a simple cube and the movement allowed the player to move left and right using the arrow keys within a specific range.

```
void Update()
{
    if (isKeyboardMovement)
    {
        xAxis = Input.GetAxis("Horizontal") * Time.deltaTime * playerSpeed;
        movementVector.x = xAxis;
        transform.Translate(movementVector);
    }
    else
    {
        mousePosition = Input.mousePosition;
        movementVector = Camera.main.ScreenToWorldPoint(new Vector3(mousePosition.x, mousePosition.y, 5.0f));
    }

    transform.position = new Vector3(Mathf.Clamp(transform.position.x, leftEndPoint.position.x, rightEndPoint.position.x), transform.position.y, transform.position.z);
}
```

Figure 1: Code exert for moving the player

This movement vector takes the input from the mouse and multiplies it by the current time to ensure it is consistent with all frame rates. Taking this horizontal movement allows for the player to change this value based on whether they are moving left or right (these directions have default keys already set within Unity, being A/D or LeftArrow/RightArrow respectively), which can then be used to move the player by translating the gameobject with that vector. In order to ensure the player did not go out of bounds, a clamp was forced upon the player using two preset positions that corresponded to the edges of the moveable area. In other words, the horizontal direction was locked between the playable area to ensure the player did not move outside the predetermined lanes. By only moving the horizontal value, the player does not move in the Y or Z direction which would cause unintended movement outside of the fixed area.

With the highway and movement sorted, next came building the notes. The notes themselves are gameobject prefabs that Unity uses to continually create and destroy whenever needed. Each note was built with a script that would cause their movement and also their ability to despawn. Since notes would be continually spawning as part of the map that would be created later, the notes needed to be able to act as soon as they are spawned so the script to move them along the highway was added to the notes themselves. While this was not the most efficient method, it proved to be suitable for the first stage. Despawning was also necessary as without it, notes would spawn and continually be part of

the level even if it had passed the hittable area that the player would use. As such, all notes were given the ability to despawn a set time after they are created.

Of course, the core mechanics would not be complete without actual note spawning. This was done using the Instantiate method that is part of Unity, and creating them at 8 individual invisible game objects located at the midpoints of each lane. This would cause each note to move along any lane they were placed at, but this meant there needed to be a way to spawn a note both at a specific time and specific lane.

To actually begin building the level and not just randomly spawning notes, a map to save the notes and play them in a specific order was needed, and required a storage of notes as well as the data that would be needed for them to spawn. As such, an array of structures holding the note type, spawn time and the lane to place was created, which meant all the level had to do was load in the array and then read off each structure in a row, deciding when to spawn the next note by the time value stored.

Finally, the actual hitting mechanic was created. Since the final product was going to end up as a mobile game, 8 buttons were created that would act as the touchscreen controls for each lane. Since each note had a collider, it was possible to detect at what point the note was along the track by using different colliders at different ranges along the lane, and give a responding accuracy which is covered later. In a rhythm game, hitting notes at the correct time is everything and as such accuracy is a very important aspect, so ensuring the notes were not able to be hit outside the predetermined ranges was extremely important for the project. When hitting any of the buttons, three colliders activate and respond depending on the closest one to respond. If the “perfect” collider has an active collision, then the game will respond with a “perfect”. This is also true of the “good” and “miss” colliders but the note will not actually miss if it is outside these ranges and the button is pressed. Doing it like this was significantly easier and had a reasonable degree of accuracy, provided there were checks to ensure only the closest note was recorded every button hit.

Combining all of these aspects together creates the basic gameplay mechanics for The Melody Slayer.

6.2 Stage Two

With the core mechanics in place, stage two consisted of designing and implementing the bosses, as well as the custom map creation using FFT.

Accuracy and score needed to be implemented, but this was a small task as part of stage 2. Since accuracy is already determined by the hit of each button of every lane, a message was needed to be sent any time a button was pressed and a successful or unsuccessful note was

registered. Doing this meant the total number of notes as well as the current hit notes could

```
totalHits++;  
if (totalHits != 0)  
{  
    totalAccuracy = (successfulHits / totalHits) * 100;  
}
```

be recorded, and then divided together (multiplied by 100) to create a percentage to use.

Figure 2: Code exert for accuracy calculation

The score that the player accumulates throughout the level is determined by the accuracy and thus each hit also needed a score associated with it. Perfects were given 100 points, while goods 50 points and misses 0 points.

The bosses of the 5 core levels were designed relatively early in development, with their core mechanics revolving around either deceiving the player or disrupting them. Due to the nature of each boss being different, the next section will talk about each boss and their mechanic separately:

The Town Guard is the boss of the first level and acts as a pseudo-tutorial boss whose only purpose is to be menacing while the player gets used to the mechanics. This boss has no special mechanic, and is merely meant to die

The Veteran Assassin focuses on stealth, and as such notes will disappear in beat to the song. This mechanic was done by adjusting the alpha value of all notes appearing for a set amount of time once a threshold in the song has been reached (specifically 66% and 33%). This effect does not make the notes disappear completely but scales their alpha over 3 seconds to 0.2, which is clear enough to still see the notes but also can be confusing for those not used to his mechanics. The scaling means that an observant player can see the note as they're spawned before they quickly disappear until the effect wears off. This is much more user friendly than the notes disappearing completely to leave the player confused.

The Reanimated golem, on the other hand, is focused more on brute force and will forcefully move the highway as the song is playing. This is done by just translating the entire board in certain directions (whether it's shifting right, or left etc). Keeping the entire board as one consistent object during this time proved difficult however by keeping the notes always in reference to the highway, it meant that the golem could continue his attacks and not cause the notes to fly elsewhere.

The Guardian of the Spirit is a more interesting boss in that the guardian can cause the song to speed up or slow down at set intervals. This is easy enough to do with the song but ensuring the notes also move with the song meant sending a message to the note prefab to change it's base speed whenever a change in music occurred.

The final boss, Spirit of Melody, was not fully implemented for the version submitted however the Spirit of Melody was to originally cause all of the above effects including invisible notes, moving the board and speeding up or slowing down the song.

The UI also took some work to implement correctly. Since the game is designed for mobile, the UI had to be redesigned to fit on a mobile screen. By using anchors to set corners and also scaling the canvas to a dynamic resolution, it meant the UI would work with a majority of phone screens as tested with a previous version. The UI, however, is rather basic and focuses a lot on large buttons so they are able to be easily pressed, but looks very bland. This was done to focus functionality over aesthetic, however with more time this UI design for both the main menu and the level selection could have been improved. This is also true for the in game menu, as all the information are located neatly in the corners to ensure the player knows what combo, score or accuracy they have at all times.

6.3 Stage 3

FFT, or Fast Fourier Transform, is an algorithm that samples signal over a period of time and then divides that into individual components. This means that noise over time can be converted into specific frequencies that can then be limited to use specific high points to be read into the maps time delay for every note. This is useful as in Unity, there is a specifically built function for reading Spectrum data and reading it back as samples. What this means is that we can take the specific high points of any song and, limiting the maximum frequency that is registered by using a variable called `limitingThreshold`, take specific points of the song to then spawn notes. By spawning notes at these high points, we can automatically create maps without having to build them ourselves. By using audio-analysis to create our maps automatically for every song instead of manually, it saves a significant amount of effort but also comes with the risk that songs become monotonous as the high points are in the same place and this means notes will always spawn at the same points as the song unless the `limitingThreshold` is reduced.

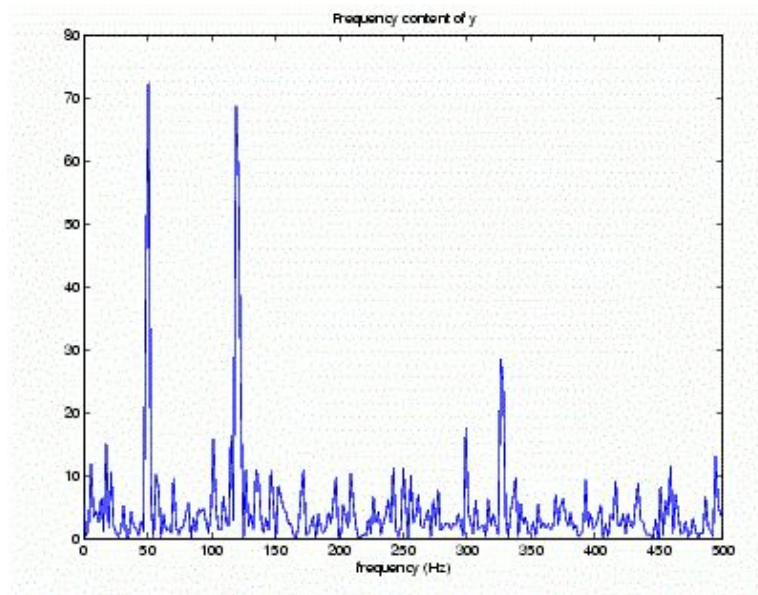


Figure 3: FFT analysis example

7.4 Testing

In order to ensure the project was optimised, a large amount of testing was done during several iterations of the projects development. This involved having other people come and try out versions of the game that are now outdated in order to get feedback. This was mainly done with friends and family members, however their opinion is still useful feedback. Some changes were made to the project because of this, such as the addition of a combo meter to count how many successful hits in a row you have gotten.

8. END OF PROJECT REPORT

7.1 Summary

The main aim of the project was to develop a rhythm action game combined with defeating bosses. The gameplay follows a typical rhythm game of hitting notes to the beat of a song, with the addition of bosses that add unique mechanics to disrupt or deceive the player.

To consider if the project has met its objectives, we must look at the original objectives:

1. A complete experience where the player begins the level, plays the song to the very end and then returns to the menu with the ability to play another level#

This objective was met, as the game has both a main menu, a level selection screen and the complete cycle where the level will both start and end back at the level selection screen. While full persistence of scores and accuracy between scenes is not implemented, the full game cycle is still present.

2. A rhythm game experience with score, accuracy and health

This objective was also met however there are issues with timing in regards to the rhythm game experience. The means to analyse songs and use them to generate notes to the beat is not easy and it proved quite difficult to get the game to be as smooth as other rhythm games analysed as part of the research. Despite this, the full experience is still present and rewards the player for being skilled.

3. The MVP and the boss level that combines the rhythm game experience and a unique mechanic.

This objective was met, as the game features the first and second levels where in the second level, notes disappear at set intervals as accordance to the boss, being an assassin leader. The MVP has also been met with a majority success, however some aspects did not make it in the final product such as the full 5 levels with 3 difficulties as part of the original specification.

7.2 Changes to Project

While most of the project is the same, there have been some significant changes to the original specification which have been referenced throughout the report. The most notable one being the change from having a player moving across a horizontal line to hit nodes as opposed to the basic note-hitting mechanics that are in the final product. This change occurred due to testing that showed the mechanics to be too complex to be fun and difficulty trying to hit notes from across far away lanes with accuracy.

9. PROJECT POST-MORTEM

Since this project was mainly created to fulfil self-development goals, the projects original objectives were sufficient and reasonable, however better time management could have saw a much greater end product. The project itself ran slower than originally anticipated and this left some desired features unable to be implemented, however the flexible workflow allowed for a workable product to be produced despite this.

9.1 Aspects that succeeded

Along with the general feel of the game, there was also success in the design of the FFT song analysis and its implementation into creating automatic maps was surprisingly effective at how flexible it was. While it created maps that were not too creative, it did create a fun experience and eased pressure off of designing maps by hand, which would have taken far longer. This experience now will allow me to use FFT in the future should I ever need to, and the knowledge of creating a rhythm game in Unity.

The project management was also quite successful, as the tasks were clearly organised and most of them were completed on time.

9.2 Aspects that failed

Unfortunately, the original concept laid out in the project initiation document was too complicated to play and this shift from the original concept to the current mechanics caused delays in advancing the project. The original idea would have made the game a lot more unique compared to other rhythm games but proved too difficult and not enjoyable for the player.

While the project was originally intending to have art assets, many were not able to be made by hand due to lack of time, as too much time was spent developing features instead of designing art assets. As such, much of the game uses third party assets (all available to use) but this was not the desired outcome for the original plan

9.3 Reflection on future improvements

In the future, better time management is needed. Many aspects of the project were predicted to take far less time than actually implemented and this caused both delays and changed to the plan as well as forcing the removal of some desired features.

10. CONCLUSIONS

To conclude, the project itself can be considered a decent success. The core mechanics are there and the project had some very good project management in place, with agile allowing for some flexibility despite delays caused by changing of the original specification. Knowing how much time it takes to implement certain features has helped me plan for future projects.

The game itself has the MVP (except for the art assets) however there is much that could be improved with more time. There was little to no consideration for sound or art design as while it was planning, no development time went into this aspect except for modelling which never got implemented. The FFT usage however was a great success and the scripts are generally high quality, although the lack the documentation a normal programming project should have.

Overall, the project was a success, and while it could have had some improvements it proved functional and helped as a learning experience.

11. REFERENCE LIST

Ni.com. (2018). *Understanding FFTs and Windowing - National Instruments*. [online] Available at: <http://www.ni.com/white-paper/4844/en/> [Accessed 21 May 2018].

Agile Alliance. (2018). *What is Agile Software Development?*. [online] Available at: <https://www.agilealliance.org/agile101/> [Accessed 10 May 2018].

12. APPENDICES

A_ PROJECT INITIATION DOCUMENT (PID)

The Melody Slayer Product Initiation Document

1. Introduction

The Melody Slayer is a single-player rhythm-action game in which you play a warrior trying to kill powerful creatures throughout the land using the power of music. In this game, the player will need to hit notes that come towards them along a “note highway” in front of them by moving to where the note is falling and hitting the attack key, in order to deal damage to the boss. Successful hits will deal damage while missing notes or incorrectly timing attacks will damage the player instead, which can fail the level if the player takes too much damage. Successfully hitting notes, however, will regenerate some health. Hitting one of these notes slightly early or slightly late will still count as a successful attack (within a degree) but count for fewer score and accuracy.

The gameplay itself would consist of moving along a horizontal line at a fixed speed dependant on the bpm of the song. This line is lane based, meaning notes will only fall at specific points on this line and the player will need to move to the section and hit the attack key at the right time to successfully hit the note. These notes will be moving at the pace of the song however, and fully mastering a song will be intended to be difficult on higher difficulties. There are also different types of notes, such as hold

notes (where the player must continually hold the attack key and remain in the same lane to achieve maximum score) and Swipe Notes (where the player must move quickly across the lanes to successfully hit the note).

The game world will consist of individual levels that each contain a unique song and boss relevant to that level, meaning each level is the duration of the song. These levels would have a range of difficulties, with more score available for higher difficulties. This allows for replayability as mastering the hardest difficulty will require skill, and only a 100% perfect run will achieve the maximum score.

2. Background/Motivations

The Melody Slayer is inspired by rhythm games, and how they test a range of skill. While playing a rhythm game, there is a big feeling of accomplishment in being able to fully master a song, as almost all of them have a range of difficulties and can be played by almost anyone. While some people do not enjoy these kind of games, there are many that do and playing along to the beat of a song can be exciting and rewarding. Beating a boss in a difficult game also provides the same feel. Bosses are designed to test the player and ensure they have learned the gameplay enough to pass to the next area, or to obtain rare rewards.

While I have attempted experiments with both these genres, I have never combined them. I feel there is a unique and fun experience to be found in a game that allows you to defeat bosses to a song, where the game itself is easy to learn but hard to master. If implemented well, players will be playing difficult levels over and over trying to get the maximum score, and feeling exhilarated when finally defeating the boss.

The main references used for this idea are Thumper (<http://store.steampowered.com/app/356400/Thumper/>) and AudioSurf

(<http://store.steampowered.com/app/12900/AudioSurf/>), as well as the common rhythm gameplay seen in something such as Frets on Fire (https://en.wikipedia.org/wiki/Frets_on_Fire). These three games helped shape the overall idea and gameplay of The Melody Slayer.

3. Project objectives

- 1) Having the player move along a horizontal lane, and havin notes will fall from a track in front of them where the player must move to the lane that the note is falling from and hit the correct key at the right time in order to successfully hit each note.
- 2) Score, health and progress meaning the player can achieve score by hitting notes well, and health is increased or decreased based on the performance of the player. The progress bar acts as the health of the boss and is depleted when the song ends.
- 3) Several types of notes: Single, hold, slide. Hitting a note provides a corresponding score based on the accuracy of the hit.
- 4) 5 songs minimum with 3 difficulties each (easy, medium, hard). Each song will have a boss unique to that level.
- 5) The complete experience of starting a level, picking the difficulty and then completing the level, with a main menu and a hub menu.

4. Initial scope

The initial scope contains each aspect of the game that will need to be developed for both the minimum viable product and the features that may be implemented, but are not needed.

The MVP will require at the minimum, in the order that they are expected to be developed in:

1) Development of the core gameplay mechanic that allows the player to move and hit notes. This will involve:

--the player being able to move along a horizontal axis but within limit.

--the prototype of the highway where the notes fall, including 8 lanes.

--the code that will cause notes to fall along the track and past the player, being removed when out of view.

--The first song being implemented and code to allow synchronisation with the songs bpm. In other words, having the notes fall at the same speed as the song. This part will be improved over time, with the initial part being a simple test map to ensure the mechanics are working.

--A temporary hard-coded map, with some development of the framework to allow easy manipulation of notes and their timing. The framework is a complex aspect of the game, and may be removed if it proves to be taking too much time to implement.

--An end state when the song is finished

--The mechanic for the player to be able to hit a note and determine the accuracy. This will complete the full game mechanics and add a win/fail state to the game.

2) The MVP will also need certain aspects, such as:

--The UI that holds score, health and progress/health bar for the boss, working with the values implemented beforehand and displaying them correctly.

--The mechanics that cause score to increase, health to decrease and the progress/health bar working. This requires the accuracy mechanic to be implemented.

--Basic sound elements for hitting a note perfectly, good and missing. Extra sounds include UI elements and boss sounds, but the main priority is the sounds played during the level.

--The main menu with level selection and statistics for the level.

--The boss for the first song, along with some sort of introduction

--Time here to patch up the mechanics, map and boss to work and flow together. This includes modelling a proper note highway, a proper player model to suit the theme of the game, ensuring the health/score work with the song

--Find/Commission the remaining 4 songs for the remaining levels and bosses

--The normal difficulty for each song, and design a boss unique to each level. This also includes designing mechanics that can change up the song/level and fit the theme of the boss (5 maps)

--Finish the full set of difficulties for all songs, totalling 15 different maps (5 songs * 3 difficulties)

3) The Full implementation of the game may also include:

-- The framework to create a map for the game, such that it is no longer hard coded and can be freely done outside of code. This would involve creating an editor and saving maps outside of the game to be loaded in.

--Scenery behind the main game for the levels. The main game only takes place within the highway and the boss, meaning the rest of the space will be bland at this point. Having some scenery would improve the experience and the feel of the game, making the player truly feel as though they are fighting a boss in their area. While some scenery will be done as part of the MVP, this task involves creating a complete level unique to each song.

--A tutorial that covers the basics of how to play, ideally as a separate map with a single difficulty. The MVP will have a written tutorial, but the full implementation will be its own separate song and map.

5. Resources and dependencies

The game itself will not need many resources, but the music and sound effects to be outsourced as for a game such as this, sound design is very important and I will not be able to produce a soundtrack that would be good enough. As such, finding copyright free music or commissioning it myself will be necessary for the final product.

Unity3D and Visual Studio will be used to create the project, but both of these will be available to me.

6. Method of approach

This game will be developed in Unity3D with C# as the language using Visual Studio to write the code. The project itself will be managed using feature driven development built week by week using Trello as the management tool. The general plan will be drafted and then updated based on what gets completed.

7. Initial project plan

This is the draft of the initial project plan, detailing what is expected at each interval.

Stage	Expected Start Date	Expected Completion Date	Products/Outcomes/Deliverables
Initiation		2nd Feb	PID
Investigation and Initial prototyping (Highlight 1)	2nd Feb	8th Feb	Repo creation, implementation of basic mechanics (movement, note falling) and analyze other genres/ideas

Highlight 2	8th Feb	15th Feb	Add UI elements and finish movement, hitting of notes and the basic map. First song will be added here.
Highlight 3	15th Feb	22nd Feb	Refinement of basic mechanics and adding score/accuracy and draft of the main menu and hub menu. Add the different types of notes.
Highlight 4	22nd Feb	1st March	Design and create the first Boss, and work on completing the first level. Basic sound effects created.
Highlight 5	1st March	8th March	Develop the framework that will allow easy addition of maps and note placement for future levels. UI improvements
Highlight 6	8th March	15th March	Design the remaining 4 bosses and levels, along with the 4 songs. More refinement of mechanics and menus.
Highlight 7	15th March	22nd March	Add the remaining difficulties, complete the MVP (full sound

			effects, scenery etc)
Easter Vacation	26th March	13th April	Small refinements and personal reflection
Testing and Demonstration	13th April	27th April	Organise the demonstration and ensure the project is up to

			standard. Draft report during this time
Final report	27th April	17th May	Finalize reflection report and add any of the full implementation ideas if there is time

7.1. Stage Management

The project has 4 main stages:

- 1) Development of core mechanics and gameplay. This contains Highlights 1 to 3.
- 2) Designing the boss and completing the UI/game experience. This contains Highlights 4 to 5
- 3) Designing and adding the remaining 4 levels/bosses as well as refining the game mechanics and adding complete sound effects. This contains Highlights 6 to 7

- 4) Finalizing the report and demonstration. This is the final submission stage.

7.2. Control plan

The following PRINCE2 control techniques will be employed:

1. Highlight reports as dictated by the PRCO304 module
2. Review meetings with project supervisor as dictated by the PRCO304 module; additional ad-hoc meetings as are necessary
3. Risk management (see Section 8); communication plan (see Section 7.2); quality plan (see Section 9); exception reports and plans as necessary

7.3. Communication plan

The communication plan will involve weekly meetings with supervisor, in line with the control plan.

8. Initial risk list

Schedule overrun – weekly highlight reports and meetings will mitigate the risk of a schedule overrun. A minimisation plan shall also be developed with the supervisor to provide a fallback in the case that some features may not be complete by the end of the week and overrun becomes a risk.

Difficulty with the technology – while I have plenty of experience with Unity3D and the C# programming language, some aspects of this game are significantly more abstract and difficult than what I'm used to, the aspect being the creating and loading of songs and difficulties. Being able to create a map easily without hard coding the type of note and the exact time in relation to the song requires a framework that allows the data of each note and timing to be stored and easily modified. As such there is a risk this part of the project takes longer than anticipated. As such, a fallback plan of hard coding the maps will be implemented in the case this feature does not work in time.

Late changes to requirements – as with many projects, changes can be made to the initial scope when certain techniques become available or others become too difficult or unfeasible. The staged approach to this project should allow for some leniency in regards to requirement changes. There are also fallback plans developed in the case some mechanics become too difficult to implement, such as the framework of creating maps easily.

9. Quality plan

Requirements will be checked with the supervisor to ensure they are correct and relevant with regards to being implemented and demonstrable. Design validation will be ensured as the game will be tested and designed to ensure there are minimal problems. System V&V to be completed at the end of each relevant increment (2 and 3)

10. Legal, social, ethical and/or professional issues

Most of the project lacks any of these kind of issues, and an ethical report will not be necessary. However, the use of music and sound effects in this game may have some legal implications as I will be aiming to use copyright free music, or commissioned music. Since I lack the skill to compose songs that would be fun to listen and play the game to, this path is inevitable. As such, there is the risk that a song is used that is copyrighted and would cause legal issues with the owner when the game is released. This is something that needs careful consideration.

Keywords:

Map = refers to a level or a stage, consisting of the song and it's notes that need to be hit. Each difficulty is a separate map.

B HIGHLIGHT REPORTS

Highlight Report 1

PRCO304: Highlight Report
Name: Matthew Webber
Date: 08/02/2018
Review of work undertaken <p>The entire managed plan was created through Trello with all tasks needed to be completed organised into sections.</p> <p>The Unity scene was also created, with movement for the player along the horizontal line and note tasks completed. The structure for the notes has been created and the ability for a note to be placed on the map and then move down the highway is also complete. Some elements are missing, however, such as the pooling system and the different types of notes, which were planned for this week.</p> <p>This week was also spent doing research on other games and previous examples of work to understand how best to go about designing the game. There is some concern about the specifics of storing the map and how to efficiently go about creating it such that the game isn't impacted. There may also be issues with synchronising the notes to the song, as this has proven troublesome for other people working on similar genre games.</p>
Plan of work for the next week <p>The main goal of next week is to add basic UI elements that show the score, health, progress and general layout of the game, and find and implement the first song that'll be used to develop the first map.</p> <p>I will also complete the core gameplay by allowing the player to hit notes, and adding the pooling system for the notes as well as the different type of notes.</p> <p>Finally, I will be doing more research on the specifics of how other rhythm games store their maps to come to figure out an efficient solution to the problem.</p>
Date(s) of supervisory meeting(s) since last Highlight
Brief notes from supervisory meeting(s) since last Highlight

Highlight Report 2

PRCO304: Highlight Report	
Name: Matthew Webber	
Date: 15/02/2018	
Review of work undertaken	
<p>The UI elements that will be present in the main game have been added and the ability to reference them from the manager that will keep track of the game statistics and note hits. The first song has also been added, which is simple but copyright free and sufficient for the first level.</p> <p>The gameplay, however, is still not finalised and there are a few issues with the original design plan that may need to be reworked. Originally, the player was supposed to control a character left and right along a horizontal track to hit notes coming from the highway but research into other games of this genre has shown that this might not be the best way of doing it and may be difficult to play. As such, I will need to review how the actual mechanics work and decide whether to keep going with the initial concept or to change the way the player hits the notes to something more common such as just pressing keys.</p> <p>A basic map was also created, although at the moment the map does not cover the full length of the song and the timings are off, something that was considered in the initial risk analysis. Time will be needed to find a solution to mapping the timings of the notes falling to the song and not to framerate like normal.</p>	
Plan of work for the next week	
<p>This next week will be spent refining the main mechanics after spending time reviewing the best way to go about how the player hits the notes. This will include ensuring the timing of the notes for the first map is correct and the addition of score/accuracy. The first map will also be completed and the main menu/level selection will be drafted using basic UI elements.</p> <p>I also plan to implement a basic version of FFT (Fast Fourier Transform) as an alternative to hand-coding maps. This audio analysis technique can be used to visualize audio spikes and may be useful as a means to automatically generate maps from the audio analysis.</p> <p>Steps will also be taken during this time to further solidify whether the game is going to be aimed for mobile or PC release, as was discussed during the highlight meeting.</p>	
Date(s) of supervisory meeting(s) since last Highlight	12/02/18
Brief notes from supervisory meeting(s) since last Highlight	
<p>Strongly suggest review of rhythm action games to clarify core mechanics - PS2 games such as Amplitude, Frequency, Rez or mobile titles derived from these...</p> <p>https://www.youtube.com/watch?v=tyRzCfnX3ok</p> <p>Also, investigate FFT in Unity https://www.youtube.com/watch?v=4Av788P9stk</p>	

Highlight 3

PRCO304: Highlight Report	
Name: Matthew Webber	
Date: 22/02/2018	
Review of work undertaken	
<p>This week was difficult due to conflicts on the initial specification. The core gameplay mechanic needed to be reworked since the concept of moving along the horizontal line proved too complicated and not very fun for those playing. The initial release target of PC has also been replaced with the aim of releasing on mobile as research into rhythm games have shown this platform is more likely to play and enjoy rhythm games casually, whereas PC markets usually require a community around them to succeed (such as Osu!). This means that an overhaul of the hitting mechanics needed to be done, which caused delays in the other intended deliverables for this week.</p> <p>The core menu framework was completed, but the timing issues are still a problem and FFT was not implemented this week. I also feel that I did not spend the full 30 hours this week as recommended on the project, so time will need to be spent next week to catch-up the project and implement the remaining features for the complete experience.</p>	
Plan of work for the next week	
<p>This week will be spent adding the features that should have been done last week (finalising the new mechanics for mobile, fixing the timing issues and FFT) as well as designing the first boss for the first level. The first boss will not have any special mechanics and act as the starter level for the player (after the tutorial if it is implemented) and will be something simple for the player to defeat. Time will also be spent crafting some audio effects in Audacity, such as UI sounds for pressing buttons and hit sounds for notes and misses.</p>	
Date(s) of supervisory meeting(s) since last Highlight	12/02/18
Brief notes from supervisory meeting(s) since last Highlight	
<p>Supervisor signed off sick for 3 months, alternate supervisor assigned and meetings begin next week</p>	

Highlight Report 4

PRCO304: Highlight Report	
Name: Matthew Webber	
Date: 01/03/2018	
Review of work undertaken	
<p>The first boss design has been completed, which has settled on a guard of the first village you enter. For now some framework has been implemented in the game to accommodate the boss but the actual boss itself has not been implemented yet. This also means the overall story of the game has been drafted and the initial concept for the remaining bosses has also been done, although figuring out their individual mechanics for each level has yet to be done.</p> <p>The transition to the new gameplay (buttons designed for mobile) has also been done for the most part, although there is still refinement to be done and testing on a mobile device will be needed as there are still issues with timing.</p> <p>Some audio effects were created but have not been implemented yet. The FFT has also been neglected unfortunately.</p>	
Plan of work for the next week	
<p>This week will be spent improving the mechanics further and exploring the options on how best to design the remaining levels. I will also aim to have a basic model for the guard boss and trying to improve the UI as I feel some of the placement is not ideal for mobile. Other than that, trying to implement a version of FFT for testing and exploring ways to improve the note storage so that maps can be more easily created will also be attempted this week, however the main priority will be getting the game to a prototype stage.</p>	
Date(s) of supervisory meeting(s) since last Highlight	26/02/18
Brief notes from supervisory meeting(s) since last Highlight	
<p>Work on improving project to a prototype for next meeting, research more on the specifics of other rhythm games and how they are enjoyable during gameplay. Use this as framework for your own project.</p>	

Highlight 5

PRCO304: Highlight Report	
Name: Matthew Webber	
Date: 08/03/2018	
Review of work undertaken Time was spent this week trying to further improve the core hitting mechanics/scoring and such, as well as also exploring FFT. By identifying the high points and low points in a song, I can use the spectrum data gathered to generate points at which to spawn notes based on the frequency. This would allow me to create maps automatically if making them all manually takes too much time. Roughly 2 to 3 hours every day this week was spent either working on the prototype or researching other rhythm games and their aspects such as visual feedback when hitting notes and their general UI design for their respective platforms (Computer or Mobile). The UI was not improved as much as I'd hoped for this week, however, and some of it is still not suited for mobile and not as complete as I'd like it (as well as no art assets yet). The game still needs minor improvements before I'm ready to test it as a prototype, and time will be spent this week to hopefully finalize those changes so that focus can be put on completing the MVP. There is no model for the guard either, which was planned for this week. As well as this, note storage was explored but nothing new has been added onto the current iteration of storing the maps as an array of structures holding the notes. This method works but is also very clunky and hard to modify.	
Plan of work for the next week This week will be spent designing the remaining 4 bosses (completing the core of the overall story), planning their scenery and implementing some form of story. This also means I will need to find 4 more songs and implement basic maps for them to improve on further, as well as considering what mechanics I need to use in order to make the game more interesting. The bosses themselves will affect how the level is played and whatever each boss does needs to be both interesting and hard to deal with, but not frustratingly so. I also plan to do some more research for the visual feedback and effects that are planned for next week, and further refine the core mechanics.	
Date(s) of supervisory meeting(s) since last Highlight	05/03/18
Brief notes from supervisory meeting(s) since last Highlight Have prototype ready for next meeting, consider doing more research and making a comparison table for different components.	

Highlight 6

PRCO304: Highlight Report	
Name: Matthew Webber	
Date: 15/03/2018	
Review of work undertaken <p>The 5 main bosses of the game have been drafted and their main mechanics drawn up, as well as the locations at which each area take place. The models are still in progress, and will most likely not be finalized until after easter. My modelling skills are not very good, and as such it is a lot more difficult to make characters that look good, especially humanoids like the Guard and the Assassin Leader.</p> <p>The plans for the visual feedback, as well as some framework have been created in order to have effects for when notes are hit to improve the experience for the player. Right now the effects are not done but the mechanics to have the effects are done.</p> <p>Roughly 20 to 25 hours were spent this week, and I'm working towards trying to spend more time on the project, as I also need to spend time working on the Industry Engagement module which has taken some of the time that previously was used on this project.</p> <p>Unfortunately the maps for the levels beyond level 2 are not complete due to having to create them manually, and using FFT has proven somewhat difficult to work with in making interesting maps.</p> <p>The remaining songs have also been found and are ready to be implemented once proper work has been done on the remaining levels.</p>	
Plan of work for the next week <p>I will be trying to spend time finalizing the first two levels and working more on the first Guard model and trying to implement some scenery or background that represents a town for the first level. I will also add the full game flow cycle including the menu, going to the first level and completing it with storage of score, combo and rank. As of right now the menu and level are separate which will need to be fixed for the MVP. Time will also be spent doing some research on the final report and making draft notes about what to put in each section, so as to not get caught out after easter with tons of work.</p>	
Date(s) of supervisory meeting(s) since last Highlight	12/03/18
Brief notes from supervisory meeting(s) since last Highlight	
Start researching about the final report and keep developing prototype for demonstration	

Highlight 7

PRCO304: Highlight Report	
Name: Matthew Webber	
Date: 22/03/2018	
Review of work undertaken <p>This time was spent developing the second levels special mechanic, the hiding notes and obstructing the view. Since the second boss is a Veteran Assassin, his speciality is stealth and as such the song is played in a way that at certain points, series of notes will become slightly transparent or parts of the highway become blocked by fog. So far, both of these are working and effective although later adjustments will be needed especially for different difficulties as right now it might be too difficult for the second level.</p> <p>The third, fourth and fifth levels are created but their special mechanics are still in progress. I have also added the ability for scores, combo and accuracy to be available on the main menu after completing a level. This persistence does not last outside of a session yet but I am currently exploring options as to how to do this effectively.</p> <p>Unfortunately, no scenery was added this week as more focus was spent on finalizing the mechanics and making the second level. Building the scenery proved more difficult than anticipated and so was put on hold until the 5 levels with their mechanics are completed. More art assets were also created for the UI, specifically the effect of hitting a note and some effects for increasing combo. While they are not done, they make hitting a note feel a lot more satisfying, which will be improved further once the note hit sound effects are completed.</p> <p>3-4 hours every day was spent this week, but time was also split on other projects so I may not have covered the full 30 hour requirement per week for this module alone.</p>	
Plan of work for the next week <p>The remaining time will be spent completing the remaining boss mechanics, adding the art/sound effects and finishing the first boss model as well as working on the remaining 4 level bosses. Ideally over the easter the prototype will be done and the remaining time after that will be spent making quality of life improvements and improving the aesthetics to the game, but mainly working on the final report and planning for the demonstration.</p>	
Date(s) of supervisory meeting(s) since last Highlight	19/03/18
Brief notes from supervisory meeting(s) since last Highlight <p>Consider reducing the amount of lanes for easier difficulties, adjust timing to be more in beat with the song and explore the option of automatic map generation to save time.</p>	