**Analysing Music to Change Gameplay**

**Abstract**

Some players have been choosing to mute the game audio so that they can listen to their own music while playing the game. This creates a large disconnect between what they developers created and what the player wants.

The aim of this project to see if player would enjoy the game more and be more connected to the game if the game syncs components of the gameplay to time with the game in ways such as: Beat and Segments.

The Project consists of two main applications, a game and an Audio Analysis project. The Audio Analysis is done though several Vamp Plugins and outputs the Beat Timings, Segment Timings and which segments are predicted to be the chorus. The game application then reads in these files to determine how the game should play. The testing for this application will be done through an online survey to get a varied set of results and opinions.

In general, the participants responded positively to the game unless the song being used was unusually complex or was intended to make the application struggle. With most songs working well and syncing up. When it worked participants noticed an increase in enjoyment and also in performance, with the poor performing songs generally causing player to enjoy it less than the game without audio analysis.

Audio analysis within this game definitely aided in improving the players enjoyment and interest within the game but more improvements would need to be made to the consistency of the audio analysis to fully recommend including this in other projects.

BPM = Beats Per Minute

**1. Introduction**

Video Game music has always been a method of helping players get more immersed in a game by using another sense, one that is not just visuals. The music can be used to set the mood of a character, it can strike fear into the players during dark scenes and can be actively used to build excitement in players. By setting the scene or by giving another gameplay indicator to players music usually sits behind the gameplay to help the visuals and rarely takes the mainstage.

With music being an afterthought to many in some games some players have opted for the option of muting the game music in favour of their own music while they play, creating a disconnect between the players and the intended game envisioned by the developers.

This problem forms the idea for the project. Can a game have the best of both worlds, The players choice in song and the developer’s vision? This project attempts to see if there would be an active engagement increase if the game were to adjust its gameplay to the music. And take the music to the forefront of the game and work alongside the visuals and not just aid in propping up the visuals.

The project works with already existing Audio Analysis Plugins to see if the data extracted from a song can be used in meaningful ways within gameplay and to then see if these ways increase the players enjoyment or a simple game.

There exists a whole another field that works on creating more and more accurate plugins but this project will be limited to what is easily accessible. Therefore, the plugins themselves may become a limiting factor of the project’s performance.

**2. Literature Review**

2.1 The Problem being Tackled

For years games have used music as a way to improve the world building within a game, usually through giving the player an auditory vision of the landscape or by letting the player know that something is happening such as a boss fight. **Jamie Sexton** describes this as coming under three broad headings: Environmental, Immersion and Diegetic. The common thing about all three is that they all *Support* the game in one way or another and rarely take lead role. But sometimes game music is used to fill the silence, like in competitive games where world building is a bit less important, but even there remains small musical snippets that aid the gameplay through intense sections and epic moments.

With game music being used by many ways by developer to develop their world and aid the players immersion there now lies and issue. Players actively mute the game music and treat it like an optional garnish(Kotaku) and listen to their own music. In an online Poll taken by **Giant Bomb** back in 2019 says that 21% of the 507 respondents say that they choose to always listen to their own music while playing every type of game with 70% of player saying that they only do it during grindy/repetitive game or moments. This leave a rather small 10% of the respondents saying that they never listen to music or podcasts while playing games. This article is not a truly reliable source but acts as yet another piece of anecdotal evidence that this issue occurs. This is not an issue that has arisen within the past few years, with the **Kotaku** article dating back to 2011.

This leads to a different understanding of each area between players and game developers. In competitive games like **League of Legends** this issue usually can not be helped as by raising the importance of the actual music can crowd the audio landscape where the character ability sounds and announcer lines are much more important. This allows players to choose their own music instead of game sounds, whether for focus or to get the ready for the next part of the game.

This differs dramatically in open-world or RPG style games where the entire meaning of the game is to have the player drawn in and absorbed into this world. Where the music is stylised towards the zone the player is in and provides another layer of depth and understanding to the player. It is much more impactful for the music to be removed as the audio landscape may sound void and empty as there is not always constant action. Yet even here player choose to listen to their own music, actively removing themselves from the game. This usually occurs during the sections where there is not much happening, such as a long journey. Furthermore, the music being listened to will most likely be dramatically out of character for the game. “Horses don’t have stereos” - Stephen Totilo, while he was playing **Red Dead Redemption**. He refused to fast travel due to that being unusual and world breaking. But then finds himself breaking the world by listening to music or podcasts, even though such things would not exist in the 1900s. He then goes on to mention that he believes this is down to his modern desire to multi-task. He can do this long laborious task while doing something productive in the real world too, which he, himself finds unusual and wrong.

A further category that should be mentioned is of the simple games, the games where players can just sit down and play at their own leisure and have no long term meaning. These games are generally fairly basic and don’t contain any convoluted worlds, in essence they are the modern arcade games. These games usually feature a simple score and aren’t musically complicated with a quiet audio landscape due to their smaller size of game features. This type of game is primed to have their music and possibly even sounds muted due to the lack of importance or gameplay effect on the game world. This is where the project aims to target, by directly looking at a simple bullet hell as an example. If by applying a gameplay importance of the music with this game, will it be more engaging. Furthermore, by allowing player to choose their own music, will it affect their performance?

2.2 Games Designed around Audio

There are no shortages of games that have audio as their primary features, but these games usually tend to fall under two categories, Rhythm Games and Pre-set Rhythm Games. Both Methods allow for interesting sections based on the chorus due to the ability to hardcode the timings for the game and can be designed from ground up to work with audio. Sadly, both styles require someone to manually make each level for each song, either by avid fans and players, or by the developers themselves. As effective as this is it requires a lot of time from the person creating the levels and can drive people away when their songs have not yet been made.

2.2.1 Rhythm Games

Rhythm games such as Guitar Hero, Beatsabre and OSU have existed for a long time. These games at their core are games developed to be played to user defined music and, in some cases, to give the impression of the user making the music. These games never get their audio muted due to the nature of the game, as it would become dramatically harder without audio. They all feature user created levels as well as some levels initially created by the developers. Usually come with a tool that lets the players make their levels.

The issue with these games is that the players need to manually create the level for their own song if it was not already made by the developers. Which not every player will be willing and/or able to do. This creates an issue where if someone wants a song that no one else has made a level for yet, they cannot play the game the way they want to play without first learning how to create a level. This could push players away from the game or make them not play the way they want which could affect their enjoyment and/or their performance.

2.1.2 Pre-set Rhythm Games

The other category finds games that aren’t strictly trying to imitate playing music, but instead use their music as a choice to create unique gameplay experiences. This is how the game BPM: Bullets Per Minute was created, it uses the music as a way to change the gameplay experience. It is a rhythm game as the gameplay happens to the beat and the player is trying to do actions such as shooting to the beat as well. It forces the player to engage themselves with the game in order to consistently shoot.

The issue with this style of game is that they use their own self composed soundtrack for the game. While the songs themselves may be amazing, it leaves a limited amount of content unless the developers constantly continue to add more music. Which might lead to the player choosing not to play the game since they want to listen to their own music, or even because they players do not like the music to the game.

2.2 Scoring on market games

Rhythm Games usually use a multiplier that increases each time the player hits a note on time, rewarding the player for consistent play. With the aim of achieving the highest score possible by maintaining a high uptime on a high multiplier. The game can end if the player starts performing poorly and misses lots of notes. Furthermore, landing on the beat is not black or white, there is a gradient that allows the players to make a small mistake but not. This is usually done through displaying: Miss, Good, Great or Perfect. This system can reward the players for consistently and accurately landing perfectly on the beat, adding a further layer of skill and expertise while still providing a way to compete with friends through high scores.

Bullet hells usually use a life-based system. Where score is given upon killing enemies or bosses, but the player has a limited number of lives and every time they get hit, they lose a life. When they run out of lives the game ends and they need to restart the game making the player be careful to ensure that they do not get hit. It provides difficulty and harsh punishment that gives a great feeling when beating the game, but also allows players to compete by getting more kills than other players so that their score is higher.

2.3 Audio Analysis Methods

2.4 Audio Plugins – Vamp

There are a few different methods for doing Audio analysis and a few good applications with varying styles. Vamp was chosen because it was a plugin and not a full visualiser saving time during production as the application would not need its visualiser removed. While also having lots of high-profile people create the plugins which are also well documented. Vamp comes with a helpful beginner Host Application to help with learning how to work with the Vamp Plugin Framework and get the application started.

Other possible choices were usually audio visualisers with audio analysis built in which has a lot of parts that were not needed. With a few other choices not providing enough detail and only giving simple features like Average BPM or Key, where this application would prefer the accuracy of each beat as that allows the ability to track Tempo changes within a song.

2.4.1Plugins Used

Queen Mary’s Bar and Beat Tracker was the Vamp plugin used for beat detection. This was chosen because they are also the makers of vamp plugins so they should have well documented and easy to use plugins while also being effective. This plugin uses a version of onset detection called Complex Domain for getting the beats. It does two passes of onset detection, the first is done to get the general curve of the tempo and the second wave is used to try and get the beats.

Queen Mary’s Segmenter was used to get the segments. This was a part of the same package as the Beat detection, so it helped keep the total number of plugins low and had all the same benefits as other possible choices. This segmenter works by doing several calculations in order to calculate a frequency-domain graph. The values are then manipulated and then used to train a Hidden Markov Model which returns the timbral types which is then used in a series of histograms to get the timbral distribution. Due to this the segmenter works best when the song’s timbre is distinct for each type of segment. It only returns when segments change and does not make any attempts to specify the structure (Chorus, verse etc.)

BBC’s Intensity plugin was used because the BBC are a large company which could allow for a plugin with better performance and reliability. Queen Mary’s plugin package did not include an Intensity plugin, so another package was also required. This plugin works by splitting the source audio into lots of smaller sub-bands and then goes through all the bands and sums all the magnitudes of each band.

**3. Methodology**

The project was made by using two separate programs that work together. The first program uses the vamp plugins and C++17 and will do the audio analysis on a song and then output the values, using the standard library’s “filesystem”, to a text file so that the game can read them in. The other program is the game that is made using unity that was found online and adapted to take the audio analysis files. It reads in the text files though a StreamingAssets folder and uses those values to adjust the gameplay values. This project is intended act more like a tech demo and proof of concept than an actual game for release.

When choosing the open source game there a few criteria that was looked for. Firstly, the game had to be a simple 2-Dimentional Bullet Hell style game and be in at least, a functional state. Ideally the game would also be built in a well-known games engine to allow for faster development and should already have audio working to make understanding and adapting the code easier.

3.1 Game Design

3.1.1 Main Mechanic

The main objective of the game to shoot projectiles at the enemy targets without getting hit by their projectiles with the aim to achieve as high of a score as possible. This game has some aspects gameplay linked with some parts of a song provided by the user. Players are rewarded by doing damage to enemies consistently and are punished when they get hit by enemy bullets. A key difference is that this game does not use a life-based system. This is to allow players to listen to their song uninterrupted and also to finish the song. A high score system is used to provide an incentive to make the players try and not get hit. Life based systems were ruled out due to the non-pre-set difficulties. Therefore, it is possible to give the game a very fast song and which will make it nearly impossible to dodge every bullet on the screen.

3.1.2 Scoring

Score Is given to the player for each projectile that they fire that hits a boss and by picking up score cards dropped by the smaller enemies. This score is then affected by a score multiplier that is increased by a small amount each time they earn score and has a maximum multiplier of 5x. But whenever the player gets hit their multiplier is reset to 1x as a way to try and make player try to not get hit. During the predicted chorus the multiplier is doubled and the rate at which you earn multiplier is also doubled to try and make the chorus be more impactful and make it more exciting.

3.2 Implementation

3.2.1 Audio Analysis

The starting point for the Audio Analysis was to work with Vamp’s SimpleHost Application. Time was then spent understanding and pulling out the parts that were Important and then changed it from a command line program into a windows console application. The first plugin to be implemented was a beat detection plugin. Beat detection would be the main part of the application therefore it was the priority. Further digging was done to find the code that writes the values out to an external .txt document so that these values can be stored in some container so that they can be worked with later. With this application, consistency was seen to be a priority over accuracy. By allowing the player to choose their own song, the application must try its best to cover as many songs as possible and provide a playable experience. To cover every song would be nearly impossible but by having something that most players could get some enjoyment out of was more important than narrowing in on a certain song or songs and have slightly more accurate section transitions for instance.

3.2.1 Beat Detection

Implementation of the Beat Detection plugin was fairly simple. Once the program worked with the test plugins given by Vamp all that was needed was to find out what the unique Identifier and variables were needed to get the application to find the correct plugin in the correct .dll and then get the correct value from this plugin. All this information can be found in the documentation for the plugin given by Queen Mary University. All the values outputted initially were loosely checked using Sonic Visualiser to ensure that the application was running correctly.

To work with the beat data from the plugin a method of finding out when a beat had passed was created, this would eventually be moved to the game to allow the enemies to know when they can shoot. This was done by checking if there was supposed to be a beat between the last frame and the current frame by using the total run time of the application and the next beat’s time.

3.2.3 Segmentation

Segmentation used a similar method to Beat detection. It checked if there was a change in segment between frames by using the total time again. This plugin also outputted which segment it was “1:A, 2:B, 3:C etc…” which required storing of different data for more accurate results. The “properties” variable of the vamp plugins was then implemented, allowing the change of some pre-defined properties. In this case, it was to change how each segment would be split up, from a hybrid of Timbral and Chromatic to Timbral priority. This was because sections of songs tend to be more easily identified by Timbre and in early testing using Sonic Visualiser, it provided more consistent results.

3.2.4 Intensity

Intensity was linked closely to segments. First all the values were stored and then the mean intensity value for each segment was calculated. This gives a picture of which parts were the chorus, as *usually* the chorus has the largest point of intensity/energy of a song. This was then sorted by intensity and the highest two values were predicted to be the chorus. As sometimes there are certain sections of a song that might be slightly more intense than the chorus. This way allows there to be another attempt while it also covers for the case where the chorus changes each time which may cause the plugin to think it is a different segment completely, which would then lead to only one of the choruses being detected. This would act as a very simple method of trying to find the chorus in a song that could easily be improved on and added to in the future if needed.

3.3.1 Game

A game was found that was made using unity and had a solid basis for the framework of the game. Initial stages included understanding of code and the structure of the game found. From there, the firing code was swapped from a “Fire Rate” variable to a “Shot Cooldown” variable that tracks how many beats are in-between each shot. This was then expanded to have a BeatController class that the other classes use for finding out when a new beat has occurred to increment their “InternalShotCooldown” variables allowing them to shoot using their fire rate.

A method of spawning bosses was needed for the next step of the game, so the boss spawning code was needed. The original code relied on randomly spawning bosses whenever one finished. Due to the nature of this game being tested by participants, a method of having the same boss order was required. But this provided a stale gameplay experience and if the song was too short, it might not allow the player to test all the boss fights across all a few playthroughs. So, a method of seeding the bosses was used. This implementation included a set order of bosses and the seed describes the starting boss. These bosses were allocated to specific segments so that the same time a certain type of segment appeared it would always have the same boss. This is not a perfect solution, but it is a nice quick work around that is quite hard to notice that it is not truly random.

3.3.2 Retrofitting a game

While retrofitting the game it was important to think about how the current game structures itself. In this case it was roughly 70% bosses 30% small enemy waves in the base game, so the initial plan of different small enemies and only having the boss during chorus was not applicable. So it was decided to allow have bosses rotate during segments and to spawn the small enemies only during the chorus alongside the boss to try and create a more exciting chorus gameplay.

Ability difficulty needed to be considered. This game was balanced around 120bpm which is roughly average pop song. This means that faster songs will be harder/almost impossible if balanced improperly. Although no matter how well the application does there will always be a song that will make it impossible to dodge all bullets and still have the game be fun when songs that should work well. Some types of enemy shooting patterns needed a way to get half, quarter and possibly smaller beats. A method of calculating these was then decided. This project Already had a way of staggering shots, where the delay would need to be calculated and applied every new beat. If this was not here, calculating the delay times and building it into the fire methods would need to be calculated. Scalable difficulty is hard as there are always going to be very extreme songs or songs where the audio plugins struggle and give awkward results. The Game needed to be built to handle these as best as possible and allow as wide of a song variety as it could.

3.3.4 Merging

Working with the intensity and predicted segments was a large part of the merge as the method used cause sync issues. The segment plugin outputs the start time for each segment, which caused some code to instantly finish the first segment as it started at 0 seconds. This Increments the counter to 1 when other sections of code were counting from 0. This caused some sync issues, particularly with the way the predicted chorus was set up, as this was given a value from the audio analysis application and these values did not align with the same segment in the game.

Intensity was complicated as the values from the plugin were not being directly used by the game and were being computed in one application and then the new values were being used in the other. Which further made the change in storage methods confusing to work with as now the value cannot be altered to fix the issue and the correct starting number was needed.

3.3.4.1 Removing Desync with Music

A key part of the game revolves around the music being in time with the beat. If the beat is slightly off it can be very noticeable so a few methods have been taken to ensure as small of a delay exists. One of these includes loading in the song prior to the game starting. This is due to the music not playing until the song is loaded in by that thread. This could task even a fraction of a second depending on the size of the file/performance of the PC which would be enough to make it noticeable and an ear sore. This was done by simply adding a main menu which acts as a small loading screen as the player is required to press start which should give enough time to the application to load it in. This requires passing the value between scenes and using Singleton variables to make sure it does not get destroyed when transitioning. This should allow the application to start the beat counting timer at the same time the song starts.

3.5 Experiment Design

Experimentation will take place over two phases; the first phase will act as a technical performance check and to ensure that the instructions are clear and understandable. With the second phase being the main source of responses that should be more polished and have a wider variety of responders.

3.5.1 First Wave

The first wave of testing will focus more on functionality and will check for any major issues with technical performance. This will also check that the explanation for the user instructions are clear and that the survey questions are clear and understandable. Due to the more specific requirements the first wave will include a smaller more specific set of participants to get more targeted data.

Participants should complete the game using three different songs during their testing and one playthrough with the base game.

- First, they should play the game with the base game’s music without any of the music analysis and gameplay changes to compare future playthrough to.

- One where they think it would work by following the song guidelines,

- One where they think it will not be very good or where the program might struggle,

- And finally, one that is completely different from the others, such as a much slower/faster song from the first or of a dramatically different style or genre.

The participants will be free to try any more songs that they may want to try, and the survey will have a short section on the thoughts in general.

For the survey, the participants will be asked about how their experience was and whether they preferred the version with gameplay changes or the original game. They will then be given a section for each run of the game they did, asking about how the program performed when tasked with different styles and whether it affected their enjoyment (either positively or negatively). Each section will have questions asking:

-Did they enjoy this version overall?

-And if they liked this one more or less than the base game? And Why?

There will also be a section that asks how the Audio Analysis worked overall. Asking if the audio Analysis synced up well and did not have many problems. The participants will then be asked to try and pick out everything that changed based on the music. If they cannot pick out many it might be a sign that it felt natural or that it is not impactful enough and will need changes. This will give another quantitative value of the impact on player experience.

For the First wave specifically, there will be questions on technical performance and about bugs asking if any were encountered and how to replicate them. After this phase any technical issues will try to be removed before the second wave of testing. The scores from each game should be written out to an external document to get uploaded so that there is another piece of data to look at and see if the music analysis helped their scores. A way to normalise the values such as average Tempo or multipliers will be needed to get a better understanding of the data.

3.5.2 Second Wave

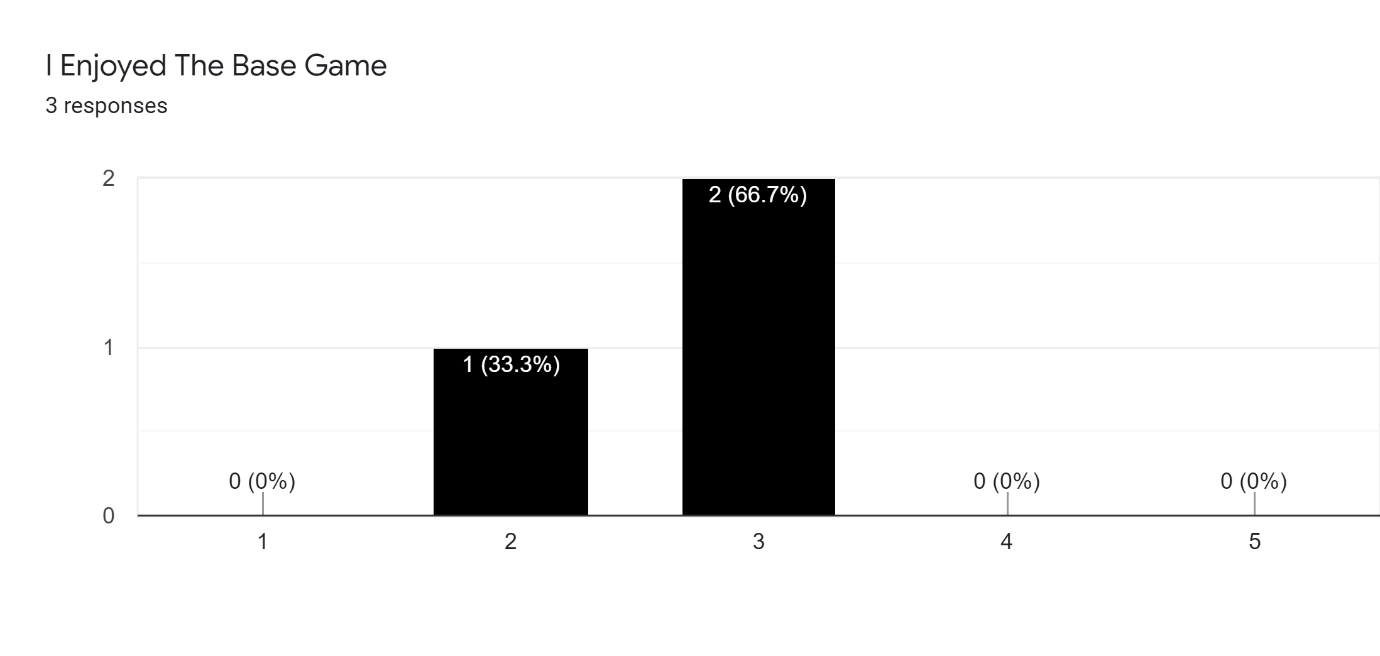
The second wave will provide the majority of the results and will be open to a larger number of people from different backgrounds. Participants should once again try three songs. A song that should work, A more convoluted song that might break it, and something completely different will be requested. With the participants once again playing the game without any gameplay changes to compare it against.

The Second wave survey be the same or similar to the first wave but without the section asking about bugs or glitches. If any changes were requested by the first wave participants, they will be made before the second wave gets sent out.

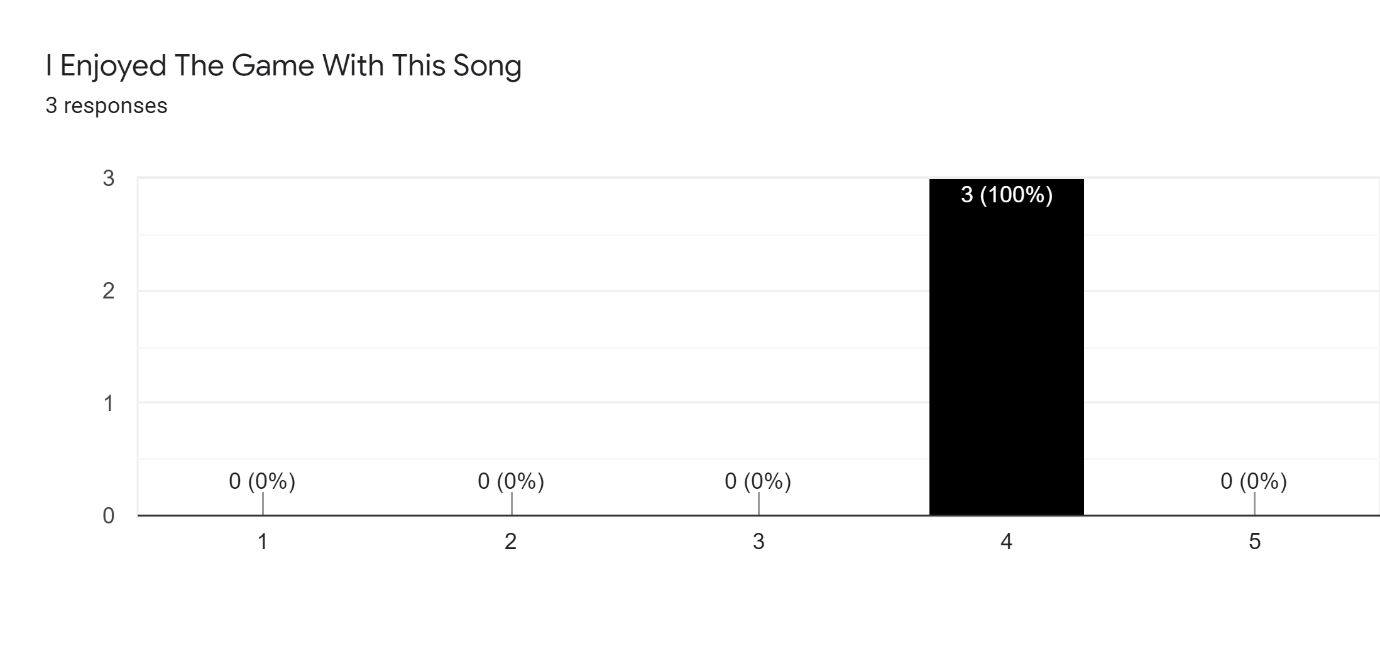
It will be estimated that the users will take ~15 minutes (average song length of 3.5 minutes) to complete their playthroughs with a further ~15 minutes to fill out the survey making for a requested duration of ~30 minutes overall.

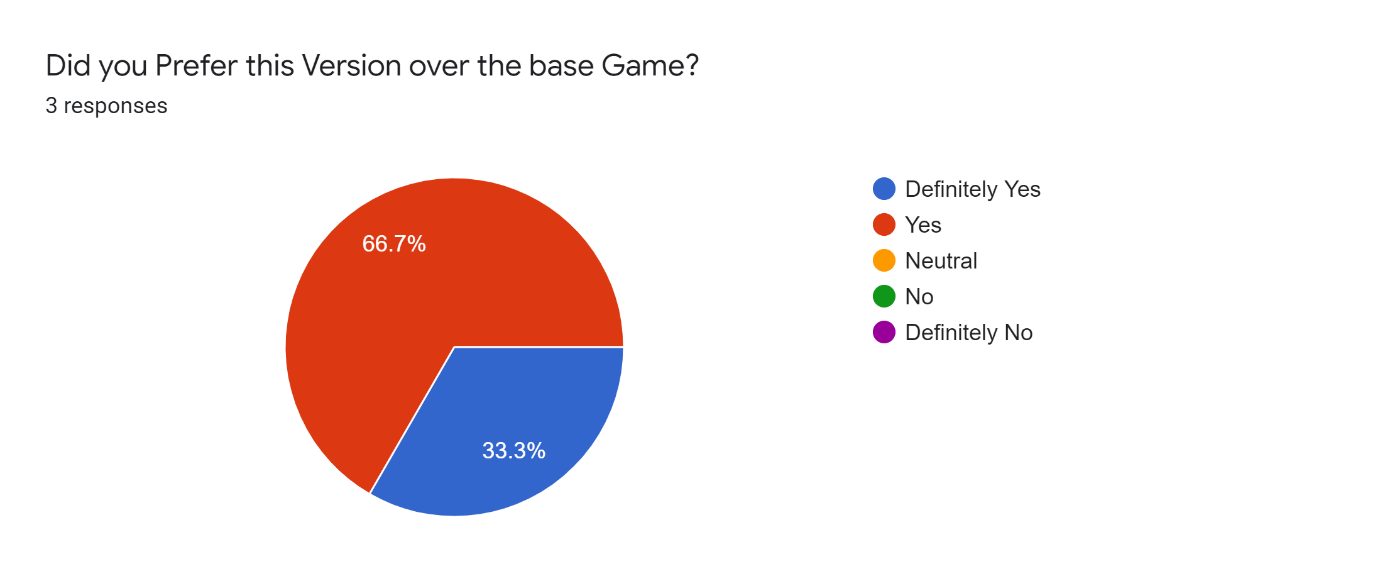
**4. First Wave Results**

4.1 Base Game

****

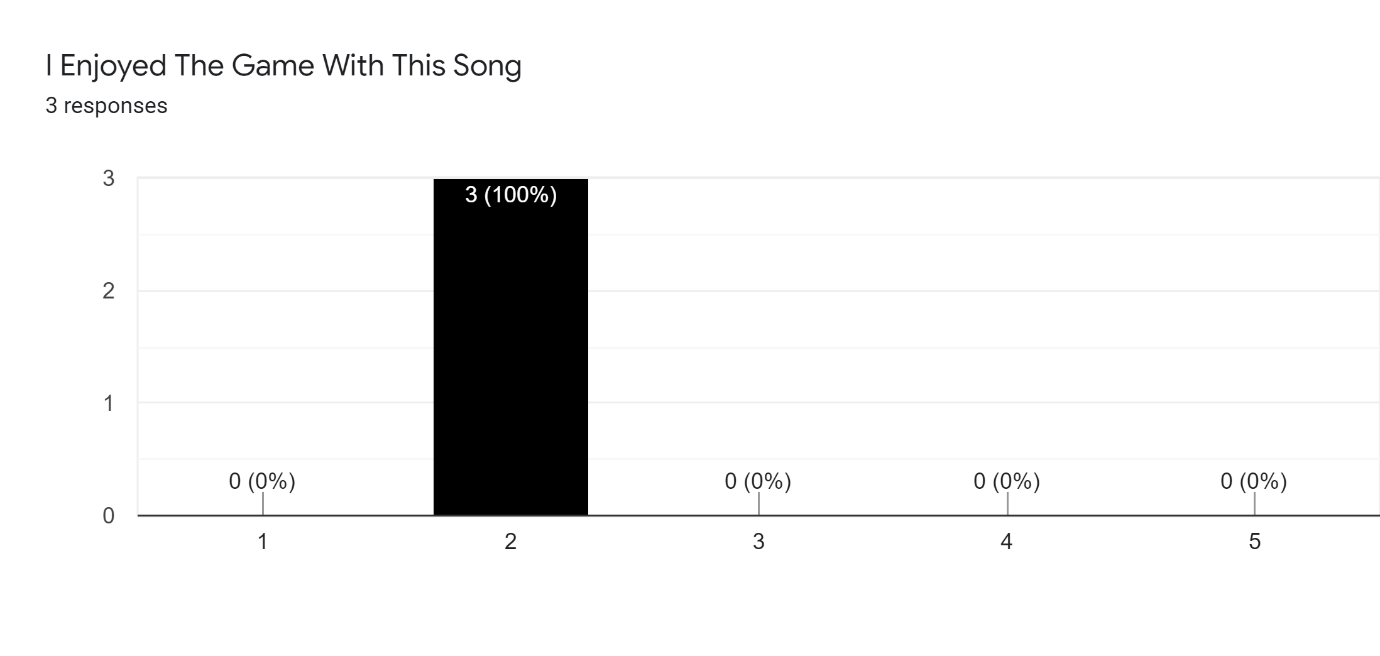
4.1 Song that Should work

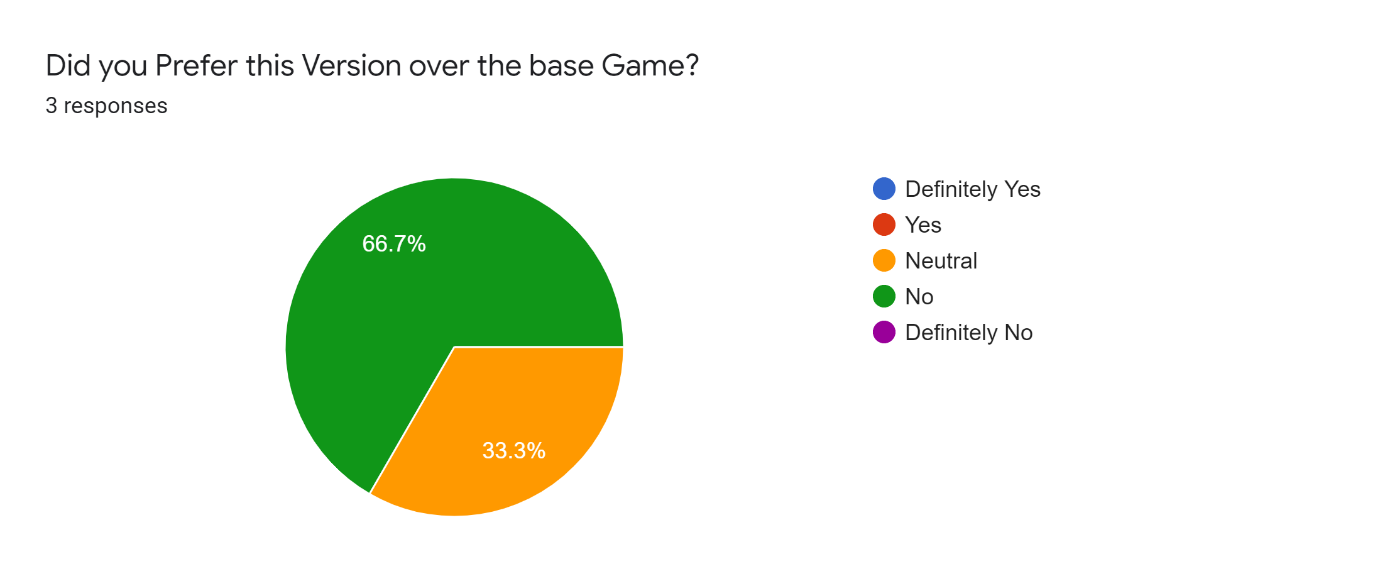
****

****

Notes: Participants said that they preferred this version because they chose what they wanted to listen to rather than the base games music. One participant said that they don’t usually enjoy bullet hells but they found that they could predict what was coming up even while it was chaotic.

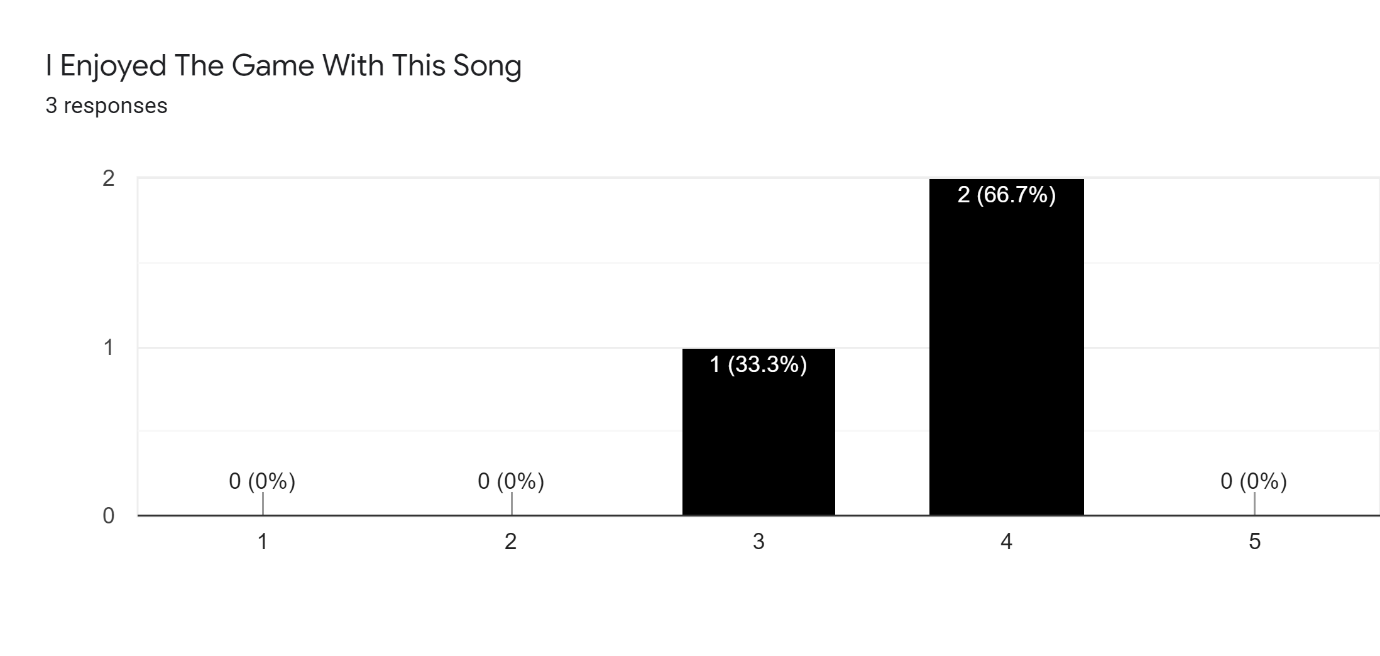
4.1 Song that Should not work

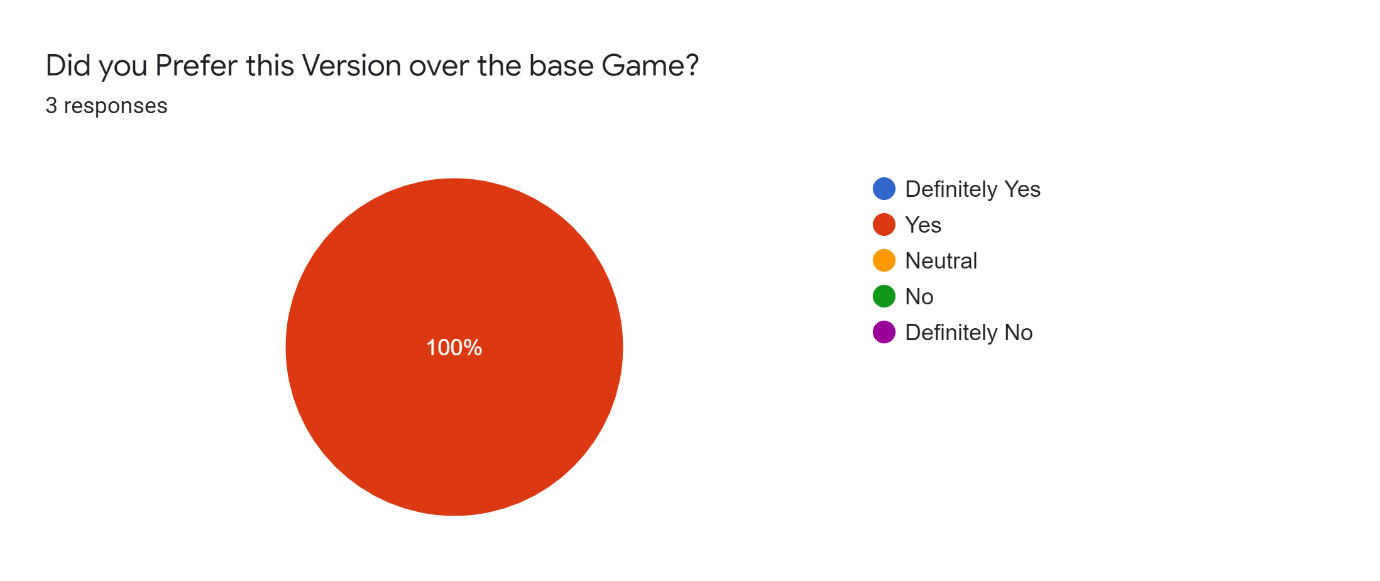




Notes: The Participants generally said that their games seemed slightly disconnected or off beat to these songs, and generally didn’t properly follow the music

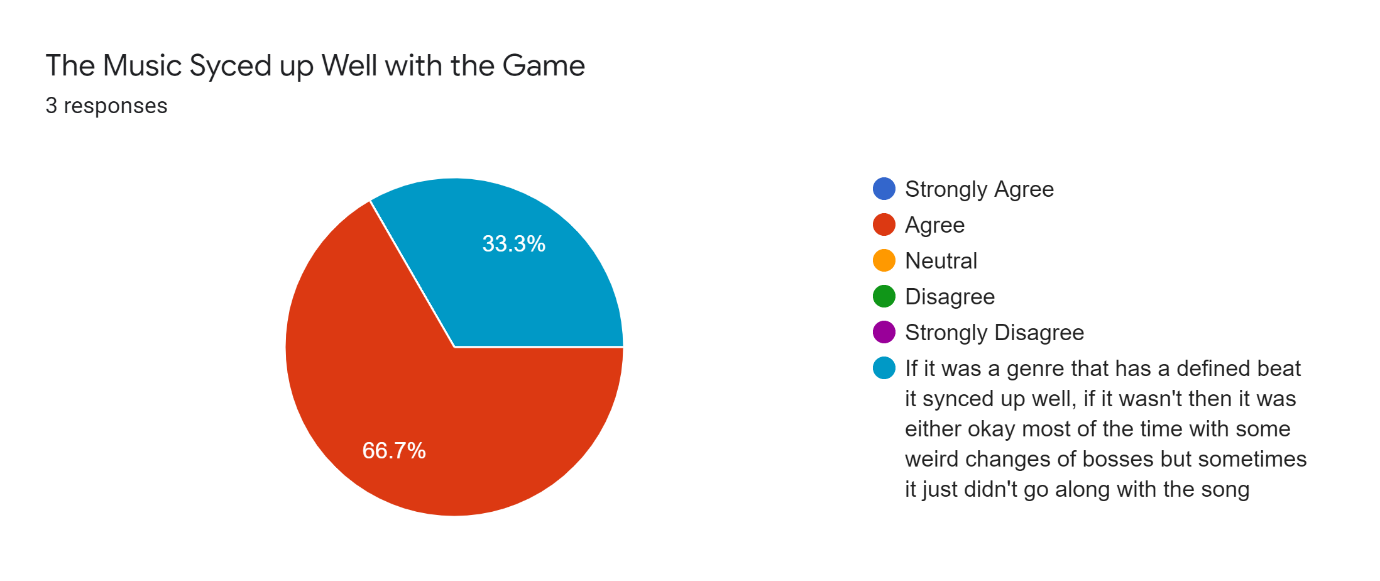
4.1 A Different Song





Notes: Similarly to the first song, the participants enjoyed this more because they could choose their own music. One participant however said that it felt accurate but not as accurate as the first one.

Audio Analysis:



List Results:

Players were able to notice that the enemies shot faster on faster songs by describing it as The Intensity of projectiles. They were also able to notice that the abilities/bosses were changing at different times without explicitly stating that they changed to features within the song.

Scores:

Two participants followed the same trend: where their weakest score was on the base game and their best song was the song that should not work with the other audio analysed songs falling in between. With the outlier participant having their worse score be the song that shouldn’t work and their best be the song that should work.

General:

Technical performance seemed to be fine with one participant reporting micro-stutters but was unable to explain how to replicate it and that It only happened in one playthrough.

One participant provided some feedback stating that on their third song (Different song) their bosses were changing when they shouldn’t have and that they would like some visual feedback when they are hitting the bosses. The also suggested some possible control changes as they did not see a point in the shoot button and that they would like to be able to move the character around the screen with the mouse cursor.

**5. First Wave Discussion**

Most players reacted neutrally to the base game as it is nothing too special, but since all 3 said they enjoyed it more when they could choose their own song shows some success already. However, all the participants stated that they did not prefer the game when they chose a song to actively make the audio analysis struggle. This is to be expected but is mostly linked to the audio analysis plugins rather than the game. The game tries its best to play well with unusual outputs from the Audio analysis but there is a limit to how good it can be. Possible changes to the audio analysis through parameters might help with the instability and provide slightly better results. Also, when the players chose another song different to the others, they mostly enjoyed it more. But it is a lot more mixed than the original song that should work which should also be expected from a section that doesn’t ask for anything too specific.

None of the 3 participants noticed anything changing due to predicted choruses. This indicates that it might not be impactful enough for the players to notice. It might also be that the players were too focused on the game to notice the smaller thing such as the screen turning red or did not notice the minions only spawn during the chorus. Before second wave this section should be worked on to see if it can be made more impactful.

The players Scores across their playthroughs were rather interesting. Most participants Base Game playthrough had their lowest scores, this is to be expected for a few reasons, one of which being that it is their first playthrough and that they are still grasping the gameplay. It also could be that the songs that they chose lead them to play better. Unexpectedly 2 of the 3 participants had their best score on the song that Should not work. This could possibly be due to the game not reacting properly to the audio analysis, allowing them to spend more time damaging the boss or by having similar choruses so more double score, or even getting lucky with bosses and getting an easy seed.

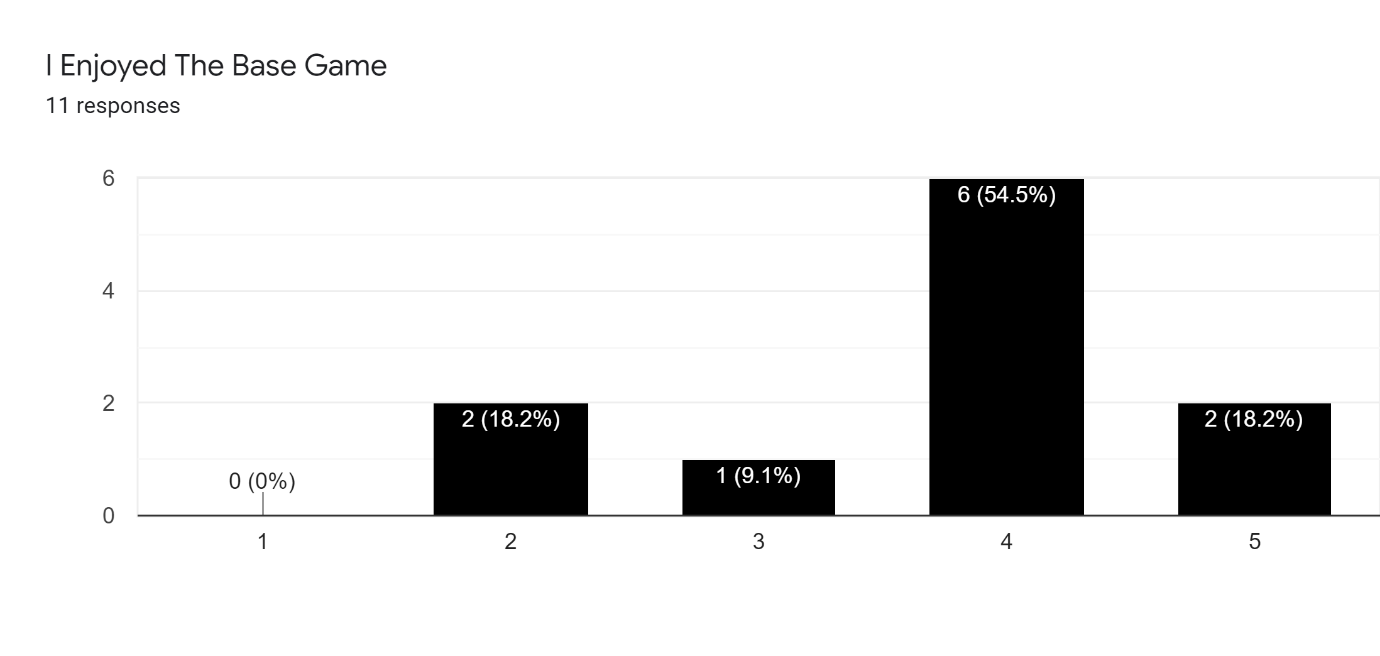
**5.1 Changes After First Wave**

One of the bigger changes made after the first wave was to the minimum segment duration. This was revisited to try and provide better results across the board and provide a more playable experience. The default segment duration for the first wave was set to 4 seconds which did provide a good amount of accuracy and managed to get some of the intricacies of the song. But this accuracy came at the cost of frequent misfires or completely incorrect timings which would take the player out of the experience more than missing some smaller sections. The new revised Duration was set to 12 seconds. This was decided after testing with several numbers. Most of the songs tested had most of its sections as two-Four bar phrases. Which at the targeted tempo of 120BPM would have 16 seconds. The reason 12 seconds was chosen instead of 16 seconds was due to not every song being 120BPM. By having it at 12 seconds it allows the plugin to have some room to find the 8 bar phrase of fasters songs. For example a song with 140BPM would need roughly 14 seconds for each of its segments which would be below the minimum timer so that major section would be skipped which might cause the plugin to misfire on the 3rd 4 bar phrase which could feel really wrong. Furthermore, songs with a slower BPM should still work fine if they are not too much slower than 120BPM. This should hopefully allow for good results within the range of ~105 to 160BPM, which covers a large portion of songs with the Optimal value being ~120BPM. Of course, not every song features entirely 2- 4 bar phrases with many variations being used, but this should hopefully be the most consistent choice.

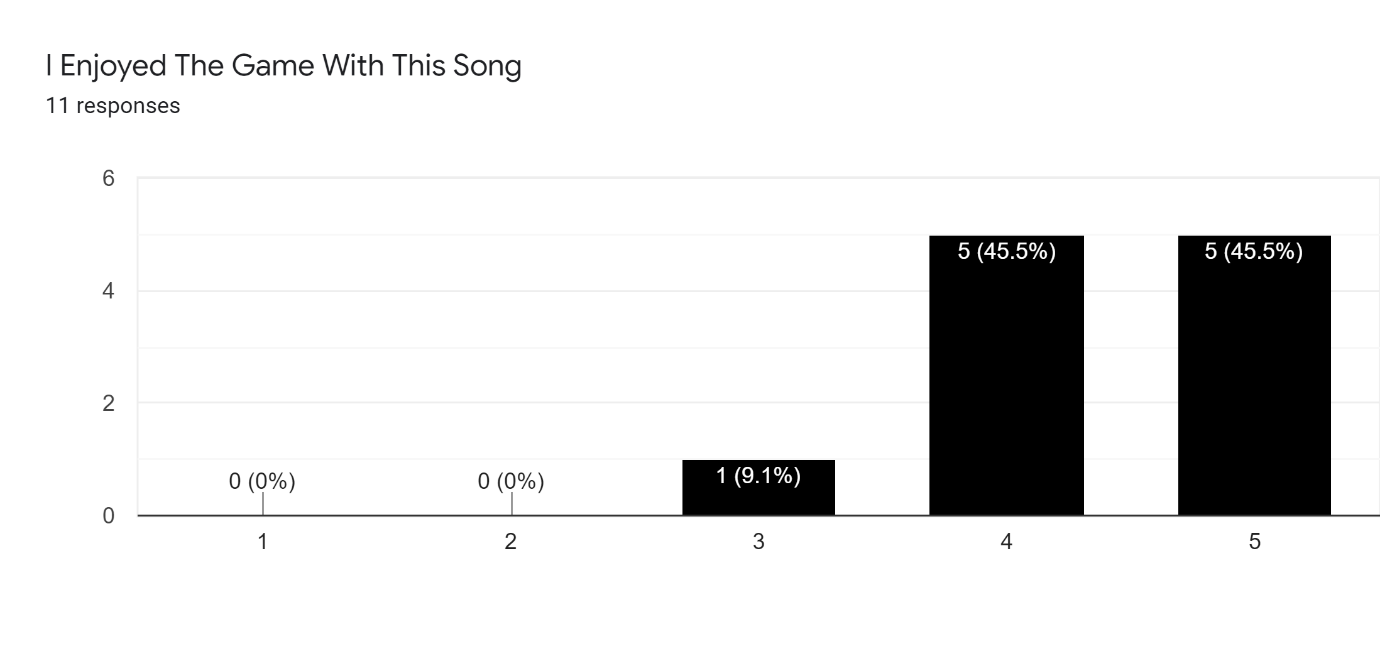
Changes were also made to the code that worked with the Predicted Chorus’. After investigation there were some issues and bugs with the code causing it to not be as responsive and causing the Chorus to be at the wrong place and/or not happen at all. This could be why the participants did not find it impactful, as It might have not even happened at all, at least not properly. After fixing these bugs, the chorus properly lines up with the chorus segment and now the screen reddening happens much more quickly when entering a chorus. Hopefully these changes are enough to have people notice the chorus changing smaller parts of gameplay.

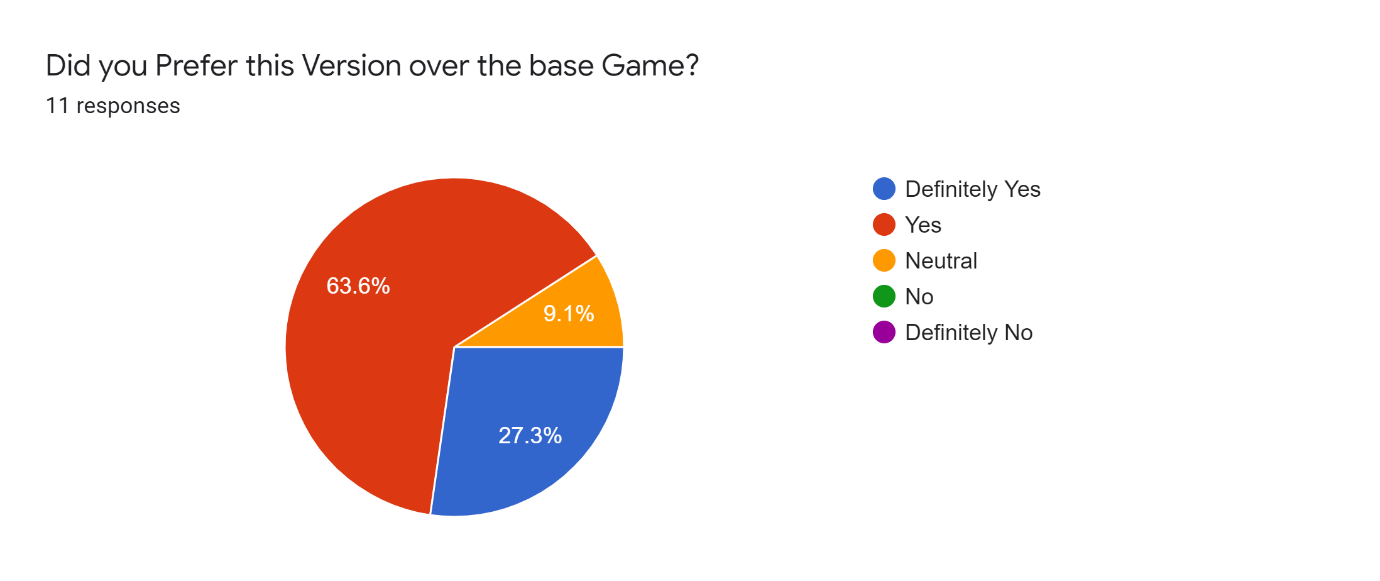
**6. Second Wave Results**

6.1 Base Game



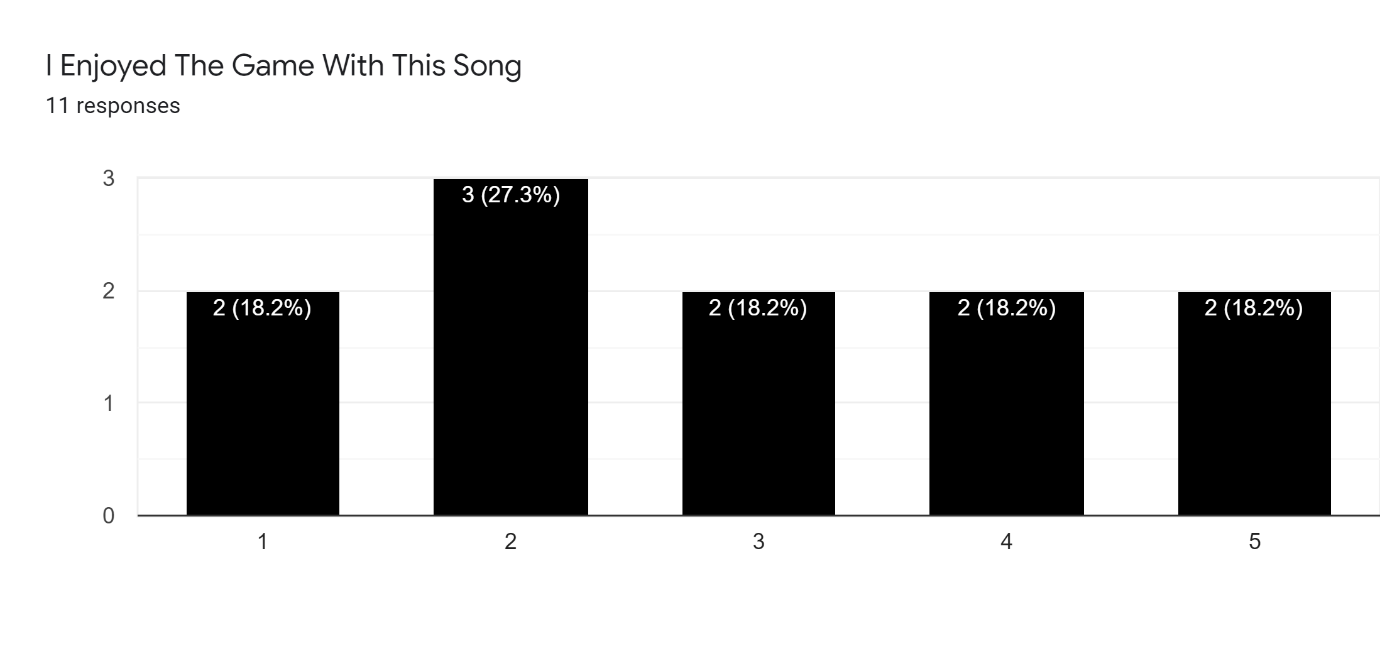
6.2 Song That Should Work

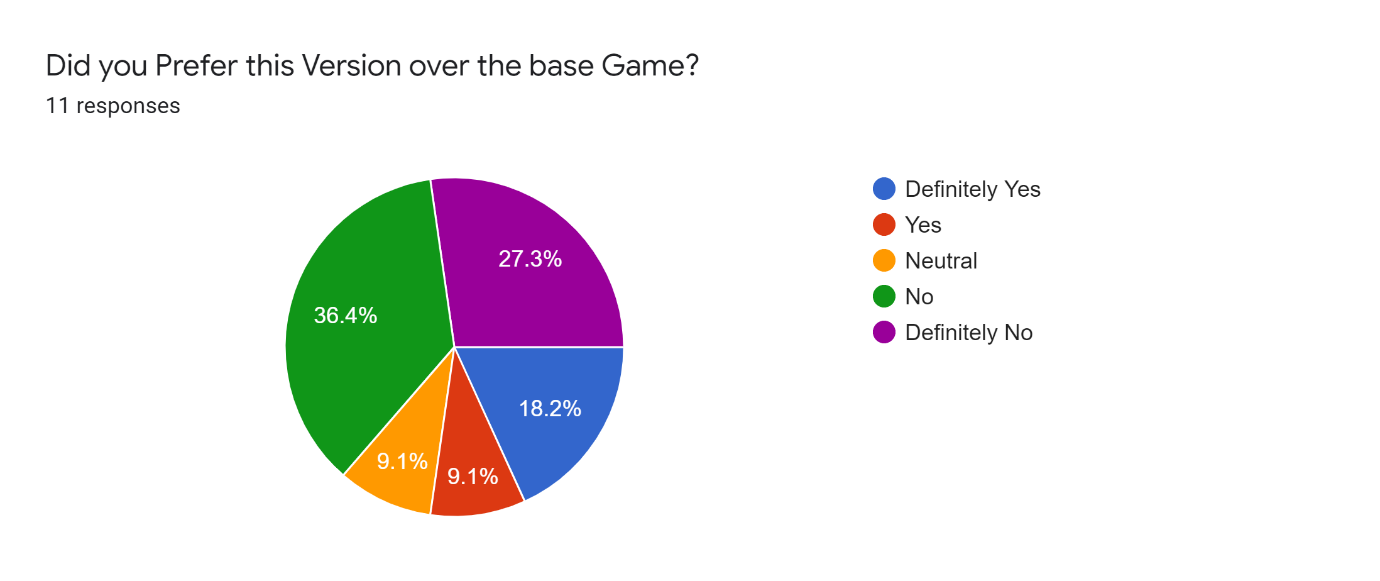




Notes: Quite a few participants noted that they preferred it because it was their own song and that their song matched the game more and provided a more varied gameplay.

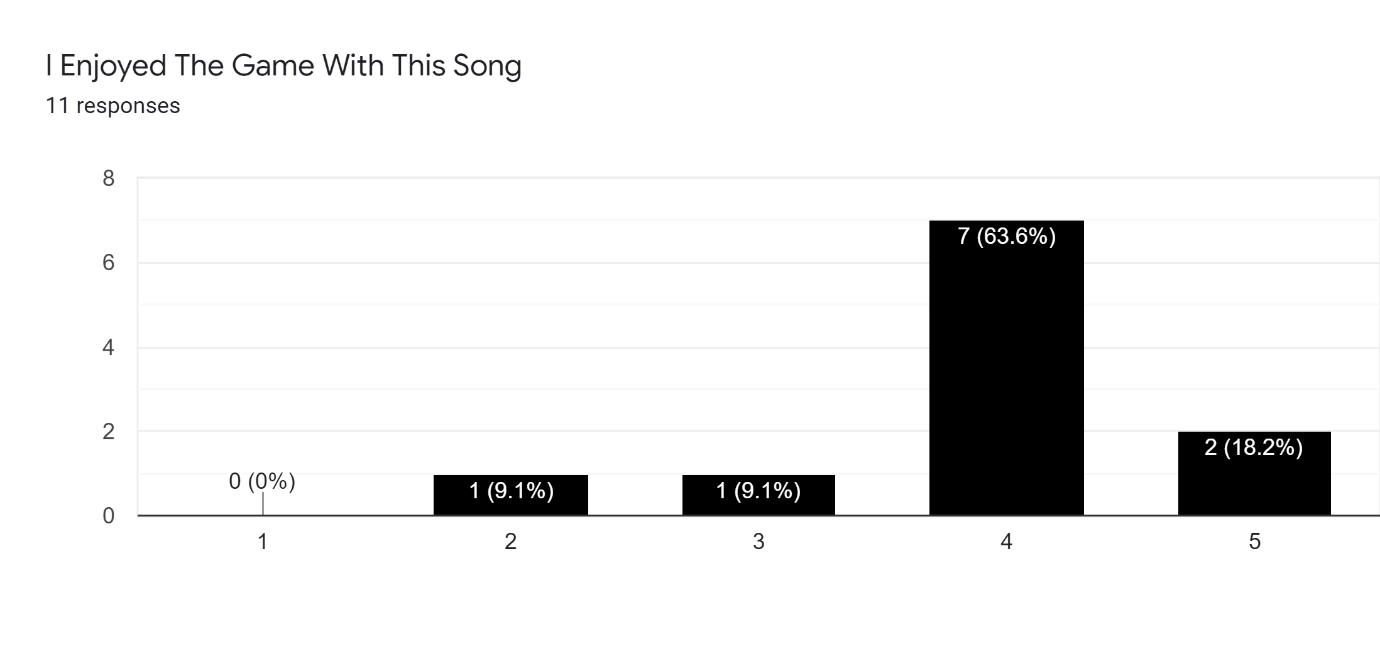
6.3 Song that Should not Work

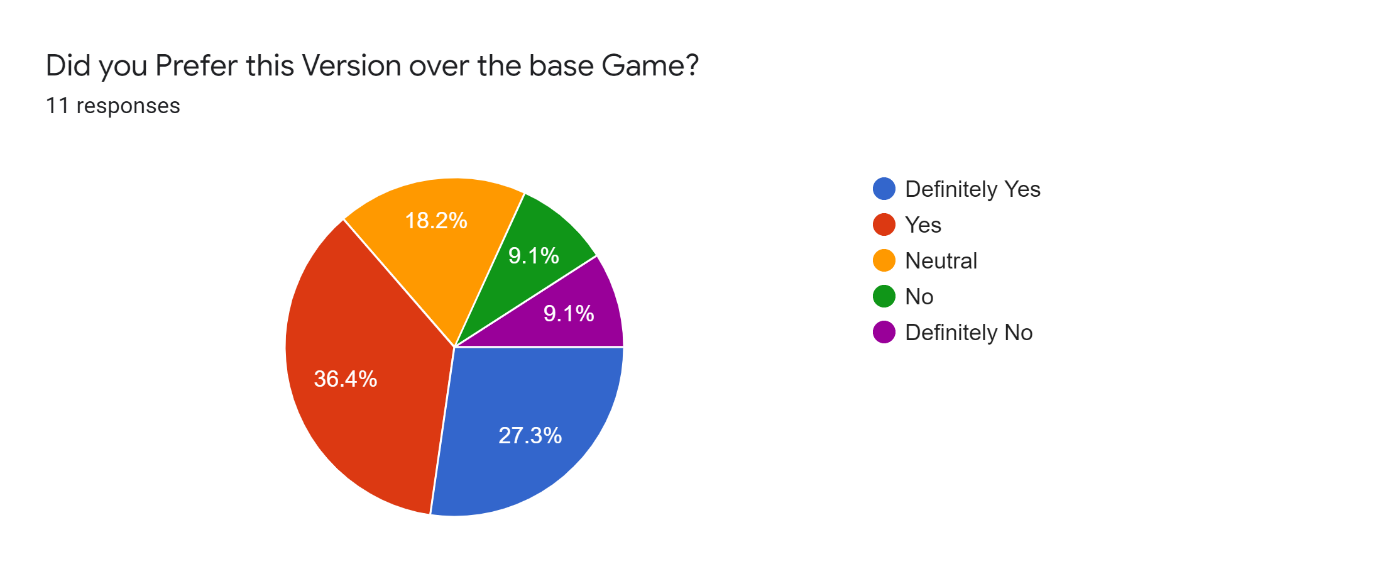




Notes: The issues participants had here were generally around the beat detection and that it either felt like it did not sync up or that the tempo was incorrect. With a few participants reporting that their segments were broken and played very few different bosses. Some participants also mentioned that due to the high BPM of the song chosen the game was incredibly difficult and that it was nearly impossible to dodge all the bullets.

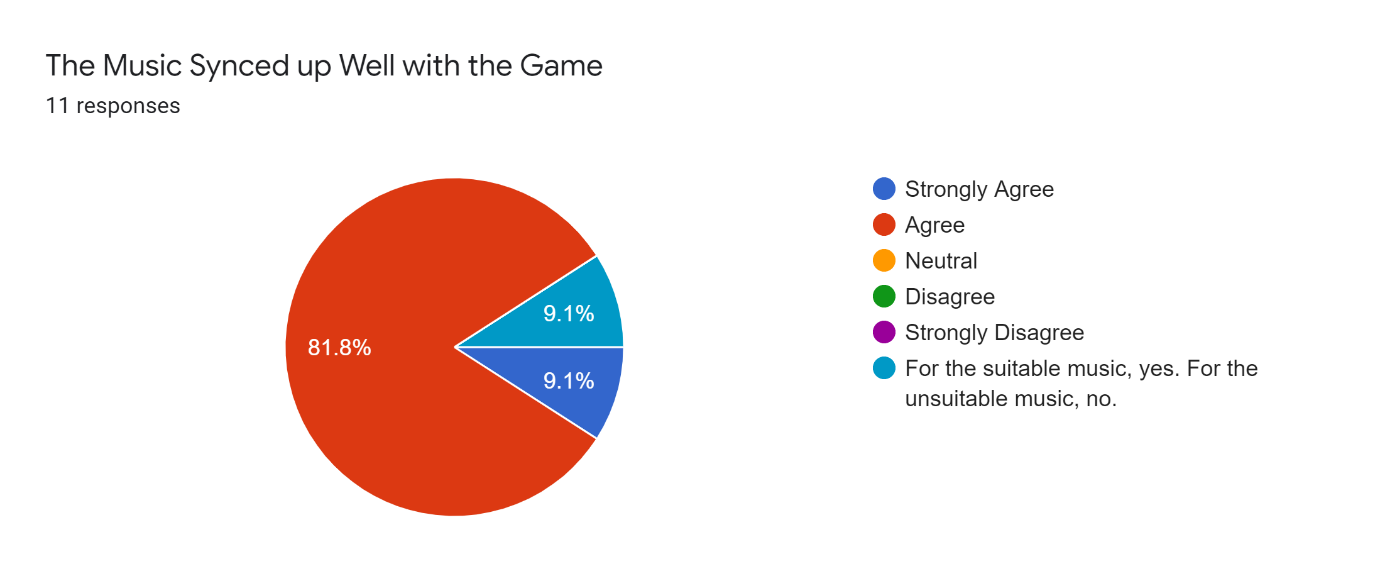
6.4 A Different Song





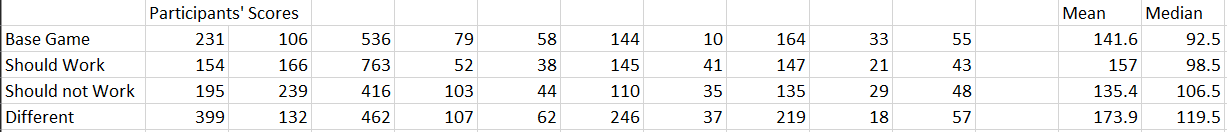
Notes: Participants mostly responded with positivity here once again stating that they liked it that they could choose something more enjoyable to them, either through difficulty of the game or just through listening to the music. Here a few participants noted some segments changing and that it was cool to see and was entertaining. On the contrary some participants used songs that caused it to struggle and during the complex parts of that song the game just was not enjoyable.

Audio Analysis:



List Results: Almost all the participants noticed the enemies fire rate changed based on the song with many saying that the “intensity” of the song changed the fire rate with one managed to pick up on the quarter beats being fired by many off the boss’ spells. Lots of participants also picked up on the bosses changing to different parts of their song. Many participants gave answer to things that were not happening such as Boss movement speed and projectile movement speed based on the intensity or tempo of their song. No participants noticed any changes happening based on the predicted Chorus.

Scores:



General:

In general participants would have liked to see a little bit more polish and a better way to run the audio analysis, preferably while in game. One participant once again mentioned that their was nor reason to not let go of fire and that it should just always be firing and also noted that their favourite boss ability added some audio feedback when it fired and acted like a drum when played in time with the music adding to the overall audio landscape. Once again, a participant brough up that they would like some small visual feedback to know that they are hitting the enemy boss.

**7. Second Wave Discussion**

Participants generally enjoyed the base game which was surprising as it was dramatically different to the first wave which might be due to the participants generally enjoying bullet hells or they enjoy playing something different.

Players genuinely enjoying the game with the audio analysis working, with only one response being neutral and the rest being positive. The one negative result did say that they preferred this version over the base game even though they rated this only as neutral. This could be due to the song that was chosen as after listening to that song the song was not a very standard song and they might have hindered their enjoyment by choosing a song that might not have worked as well as it should’ve. With that said they did still enjoy it more. The majority of responses saying that they prefer the song that worked. With once again a single outlier responding with neutral. This is what was expected that players would enjoy this version more than the base game as it was directly trying to be interesting to the players. The players performance somewhat follows what they have responded. With both the median and mean score for this section being higher than the base game showing that there was, on average, some small performance increase. This either shows that the project was working as intended and actually increased their involvement and immersion, or there was some other external factor impacting their performance, such as the players getting more practise or the players got better seeding allowing for slightly easier bosses. The Tempo change going from the base game or to their song might affect the way the scores as a few people remarked that the base game was just too fast for them so they chose slower songs going forward, which shows the gameplay decisions working and that the players were getting some flexibility in their choices while still getting to play.

The song that shouldn’t work came back with very mixed results leaning very slightly to the participants not enjoying it. This is probably down to the nature of asking the participants to try and break the application and how the plugins respond to being given some very random or complex songs. What is surprising is that some participants said that they really enjoyed this song, suggesting that for some songs it did cope very well and ended up creating an enjoyable experience. Though sometimes this could be for a different reason such as one participant reporting that their song had sections that were in 7/4 which lead to some polyrhythmic sections as the game and audio analysis is expecting 4/4. The participant said that this not only played well but added an extra layer to the song and was one of their favourite parts.

When asked whether they preferred this various over the base game the majority of players said that they preferred the base game. Which from the above results is to be expected, as this test was to really stress out the application and see how well it will hold up again some more complicated songs. The reasons for the application struggling were generally in almost every section with the audio analysis struggling on both the beats and on the segments, which would then lead to the predicted chorus being wrong. The scores for this section were unexpectedly high, where for quite a few participants this was the highest scoring playthrough. Overall, it came second place for median scores but came in last when using the mean which would show that between players there was a dramatic difference in results. The inconsistent scores here mirror the responses received by participants on their enjoyment, which can probably be attributed to the songs segments and beats not working as intended. The segments could be struggling and be giving very long segments with one participant noting that a single segment played for almost the whole song which would probably have that segment be the chorus essentially doubling that players score for this playthrough. Another area where it could be causing these weird scores is in beat detection, where for very slow songs the beat detection might count the quarter beats as a whole beat causing the enemies to shoot twice as fast creating a backwards effect causing very difficult games hindering that participants ability to get good scores in that playthrough.

By far the most interesting section is when the players were asked to choose a different song. This section gave the users the most freedom and it wasn’t specified whether it should work or not. Here the expected results were to be incredibly varied with songs that worked and that didn’t. But that ended up not being the case as the overwhelming majority saying that they enjoyed this playthrough. This probably is due to the freedom participants were given. With this freedom they could choose song that they like more as it doesn’t need to necessarily work. Which would allow the plugins to shine as it is working with songs that it is intended to work with. It allowed the players to choose a difficulty and choose a song that they think fits with the style of the game giving the players full control.

When asked if the participants preferred this version majority said that they did which would line up with the results for the previous question. The results are a lot more positive for this section than expected, with a few people saying that it was good but the first song was better. The scores for this section were the highest across all the playthroughs which directly correlates to the players enjoyment showing that there is a link between the players performance through scores and the players enjoyment.

The participants mostly responded positively about whether the music synced with the game. Which shows that the players generally were positive towards the audio analysis and then placing it in a game. With one response saying that when it works it is effective but when it doesn’t it takes them out of the experience. With further advancements in audio analysis application it may be possible to more accurately and more consistently sync the game up and have the players enjoy the experience more of the time.

When asked to list all the parts of the game they all noticed something to do with the enemy attacks firing faster based on the tempo of the song with one noticing that some projectiles shoot on fractions of the beats. With quite a few people also picking up on the bosses’ abilities changing to segments of the song. This was to be expected and is good that these were picked up. It shows that they were noticeable by the players and they were even remarked by the participants that it was really cool when it worked well. No participants noticed anything changing due to the chorus which would once again show that more may be needed to make it more impactful and let the player know that it is happening. The participant may have found the smaller enemies random and spawning whenever. No participants mentioned anything about the screen reddening, this is probably due to many of the participants not being very used to bullet hells and were focusing on not getting hut rather than the smaller things happening on screen such as the reddening background and double multiplier.

A possible issue with the results received is that very few of the participants stated that they enjoyed or have played bullet hell games much in the past. This may lead participant to enjoy the game for the reason of it being a different style of game that they haven’t played much, the opposite could also be said that they might like the overall game less due to it being a bullet hell. So another test with a different style of game may provide interesting results when compared to this set of results. Another possible reason they enjoyed it more that might not be a long lasting reason for enjoying a game is that most of these participants have probably not played a game that syncs to their own music and that this niche and unique gimmick of this game might have been what lead participants to enjoy it more than they other wise would have. If other games that did audio analysis were compared to this project the responses may have differed once again.

Throughout all the playthroughs with audio analysis the was generally some sort of roughly 60% split. Either it worked well and roughly 60% said that they preferred it or the opposite where 60% said that they did not prefer it. This generally follows on from the statement the participant gave saying that when it works it definitely helps player engagement and makes the game more fun to play with also some varying performance increase from the players but when it makes mistakes or is slightly desynced that it hinders their enjoyment. So more consistency would be required for this to possibly be a viable addition to a game, this could either be achieved through more audio analysis being run to check with each other and try and ensure correct values are being given or by trying to improve the audio analysis plugins themselves by developing more consistent plugins that can more accurately get the values correct. With that being said, there definitely is a correlation between the players performance and enjoyment and the audio analysis when it works and could be a viable path to work on in order to improve games going forward.

go through all the suggested changes/ feedback and talk about each

When looking at the suggested changes by the participants a few things are to be noted. Across both waves of testing a participant mentioned that the fire button was pointless as it was always held down anyway. It may be worth looking into some sort of ammo system to make gameplay more interesting or to simply have the enemy always fire. Also, an interesting point that was not considered prior to the second wave was the actual audio effects of the abilities themselves. If the sound effects resembled some sort of instrument such as a drum it may make the sound fit more naturally into the music it is given almost acting like a metronome for the song which may be interesting this would also allow for audio adjustment to be easier as currently the audio from the bullets shooting is dramatically quieter than the music to try and ensure it does not overpower the song or get annoying, but if it acted like it merged with the song it could add an extra layer and make it more interesting. In regard to the visual feedback when hitting the boss, this can easily be implemented and would not cause any issues with the running of the game and it seems to be something that the players would like.

**8. Evaluation/Future Work**

In general, the project performed well and got results that were interesting and can spur on some extra work. The project code itself could have been written in more tidy ways, specifically the method of transferring data between scenes as currently it is not very scalable and would be difficult to work with going forward. If performance was needed a possible improvement would be on the way beats are fired. Currently it works by changing a bool and then everything that can fire checks every frame whether it is true. When in a game like this where objects can shoot bullets, this could cause performance drops. Ideally all objects that can shoot should be stored somewhere and when a beat is fired it should call the shoot function of all the objects that are in that container.

The main limitation of this game being more consistent stems from the audio analysis. With more accurate audio analysis either through new applications or plugins. Alternatively working with the current one in different fashions such as running multiple plugins multiple times and averaging them out. Or even by changing parameters within the current plugins, either manually or by using data from even more plugins such as a time signature plugin which would give the beat detection plugin a predicted time signature, hopefully increasing the beats accuracy. With only three plugins being used within this project there is a lot of room for possible improvement. An area of improvement could be on the predicted chorus’ as players did not notice them at all and more methods of trying to detect the chorus other than just through intensity alongside more impactful and engaging gameplay implications could really improve the application.

A possible improvement on testing could be through testing in person. Due to current world pandemic physical testing was not possible and it all had to take place online. By measuring their heart rate, it might be possible to track their engagement and excitement during different parts of the playthroughs. Another reason this could improve results due to easily being able to record their playthroughs on the computer they are playing on. This could allow for later viewing and analysis to understand exactly what went wrong and what went well, this is theoretically possible tod o online but would not be easy as it would require people to have a way of recording their screen then uploading a large file online. And when combined with the already large amount of requirement such as songs in specific formats and operating systems, it would drive participants away and there is too much to do.

On the topic of file formats, a way of expanding file formatting for audio would definitely be require, not only is .wav not a popular storage for songs it is also difficult to help people through converting their songs to a .wav and this would definitely not work cleanly for a fully released game. A possible way of fixing this could be by linking it up with an online music streaming application such as Spotify or SoundCloud. This would come with a whole host of other issues, primarily around the running of the audio analysis as it requires a file to be stored to analyse.

Furthermore, it would be very interesting to see how the results change when a similar test is run on another style of game. Possibly a more mainstream style of game such as a first-person shooter as participants would be more familiar with this and would hopefully remove some variance in the respondents. If another game were to be once again retrofitted some important question would been to be answered when finding the game that were also answered when designing this game. Questions such as: What can be synced up with the game? Could the game be split up into multiple different sections such as bosses/minions? And Do powerups effect gameplay and do they need to sync with the music? Each of these questions were important decisions during development as delicately deciding what to be synced up is very important. Once answered they would hopefully speed up the development time and improve the game overall.

If the same game were to be used again it may be worth testing if instead of listening to the base game’s song, they listened to their song and then played it with audio analysis. It would add two extra playthrough, but it would eliminate the response where they state that they preferred it because it was their own song. This way it would be testing how they respond to audio analysis in a game more directly.

**References**

https://books.google.co.uk/books?hl=en&lr=&id=2RQHBAAAQBAJ&oi=fnd&pg=PR5&dq=game+sounds+replacing+music&ots=GI7roAxHwi&sig=41DyA9ce1tjTYcLg\_VHTQgrkxz4&redir\_esc=y#v=onepage&q&f=false

<https://dl.acm.org/doi/pdf/10.1145/3410404.3414245>

<https://www.reddit.com/r/truegaming/comments/2u33wv/actively_consuming_other_media_while_gaming/>

<https://www.giantbomb.com/forums/general-discussion-30/do-you-listen-to-podcastsvideos-while-gaming-1841683/>

This is what I’m testing

<https://www.researchgate.net/publication/230746039_Exposure_to_music_and_cognitive_performance_Tests_of_children_and_adults>

for a quote about video game music supporting the game

https://www.jstor.org/stable/10.3366/j.ctt1g0b5k8

<https://kotaku.com/the-year-i-gained-the-courage-to-ignore-video-game-musi-5730637>