ENGINEERING | McMaster University

Requirements Document

Hero Composer

Rev 1

Group Name: The Oniichan Studio
Professor: Dr. Jacques Carette
Course: SFWR ENG 4GP6
Date: November 6th, 2018

Members:

Nicolai Kozel (001404557)
Pavle Arezina (001410366)
Yifu Wu (001450544)
Damian Angelone (001408211)
Johnny Endrizzi (001310603)

Table of Contents

1. The Purpose of the Project

1.1 Background

The project *Hero Composer* is a capstone project for the graduating students of Software Engineering from McMaster University. It is a graduation requirement and should be a showcase of the skills and creativity developed while studying the degree.

1.2 Goals

The most important goal of the project is to develop a game prototype that will fulfill all the deliverables required for a successful project. It should demonstrate the technical ability and ingenuity of the developers to be able to entice other people to want to play the game. The audience should be able to enjoy the narrative presented and want to challenge themselves by trying to get the best score in the game.

2. Stakeholders

2.1 Clients

2.1.1 Dr.Jacques Carette, McMaster University

Professor for the Software Engineering (Game Design) capstone course (SE4GP6). He establishes the deliverables of the project and assists in grading the different requirements in the course. The final grades will be approved by him.

2.1.2 Sasha, McMaster University

Teaching assistant for the Software Engineering (Game Design) Capstone course, SE4GP6. Sasha assists Dr. Carette in the assessment of the deliverables and provides feedback for improvement.

2.2 Customers

2.2.1 Clients

Since this project is being completed for SE4PG6, the clients are also customers.

2.2.2 Judges

The judges that will be present during the final demonstration are also customers since they will give their feedback and opinion if the game is purchasable as a release ready product.

2.2.3 PC Gamers

Future customers include PC gamers as we intend to continue development and release Hero Composer on the open market.

2.3 Other Stakeholders

2.3.1 Academic Integrity Officer, McMaster University

The Academic Integrity Officer will monitor the capstone project for any signs of academic dishonesty. In the event any academic dishonesty is discovered, they will stop further project development.

2.3.2 Development Team

The development team must produce a working prototype by the date of the final demonstration. They will take feedback from the clients and playtesters to further improve the product and utilize their technical knowledge to accomplish the deliverables established for them.

2.4 Hands-On Users

2.4.1 Playtesters

These people will help the development team find game breaking bugs and test the difficulty level of the game. This stakeholder is crucial because playtesters are the exact type of people that the game will eventually be marketed towards. Their feedback is invaluable to the commercial success of the game.

3. Mandated Constraints

3.1 Solution Constraints

The constraints put forth by Dr.Carette are that the entire development cycle is only 8 months, the group size can be no more than five people, and the game must be developed in Unity. Due to these constraints, the expectation is that the end product will be a well polished prototype as opposed to a full fledged game. This is reinforced by the constraint that there only needs to be around 15 minutes of game time for the final demonstration.

3.2 Implementation Environment of the Current System

The game must be implemented on a Windows environment. Unity is primarily available for Windows systems and the development team all have Windows computers. The primary input devices will be mouse and keyboard. In the future, this could be expanded to include a MIDI controller.

3.3 Partner or Collaborative Applications

Any constraints imposed by Unity must be met. A computer system meeting the minimum requirements of Unity must be used. Any constraints imposed by Windows operating system must be met. A computer system meeting the minimum requirements of Windows must be used.

3.4 Off-the-Shelf Software

Any licensing constraints of Unity must be met. A professional license must be purchased if revenue exceeds $100,000. Any licensing constraints of Photoshop must be met. A professional license must be used in assets will be used commercially. Any assets used from Osu! must be purchased or licensed if they are to be used commercially.

3.5 Schedule Constraints

There are eight months until the final presentation. Within this time, the development team has to balance this project with other commitments such as assignments and tests for other courses.

3.6 Budget Constraints

There are no budget constraints at the moment. We do not plan to buy any assets.

4. Naming Conventions and Terminology

4.1 Role Playing Game (RPG)

Game in which each participant assumes the role of a character, generally in a fantasy or science fiction setting, that can interact within the game's imaginary world.

4.2 Personal Computer (PC)

A computer that is used by one person at a time in a business, a school, or at home

4.3 User Interface (UI)

The means by which the user and a computer system interact, in particular the use of input devices and software.

4.4 Beatmap

A beatmap is a file associated with a song or music file. This file contains information about the song’s difficulty, tempo, timing, and a manually pre-defined list of “beats” or “notes” that must be hit at corresponding times in the song. This can be thought of as an abstract version of the song’s sheet music.

4.5 Parser

A parser is a method of interpreting information from an input file that follows a strict predefined format. When we refer to parser, we are describing the way we read a beatmap and interpret its information to generate game objects.

4.6 Procedurally Generated

Method of creating data algorithmically as opposed to manually. Instead of of the developer creating the behaviour of a boss, an algorithm will decide on the characteristics of a boss.

4.7 Attack Pattern

A pattern of behaviour adversaries utilize when a battle occurs between the adversary and an opponent.

4.8 Product Use Case (PUC)

A product use case elaborates on a scenario, showing event name, trigger, preconditions, system requirements, and outcome.

4.9 Input (IN)

The input to the game is keyboard commands from the player. Certain keys will correspond to each character's actions or to modify what action that character will perform. In addition, the player will input commands to navigate the various menu screens.

4.10 Output (OUT)

The output of the game includes visual and auditory feedback. This includes animation, images, videos, text, music, and sound effects. The type of output depends on the game state and may include all of the above or a small subset.

4.11 Two-Dimensional (2D)

Having or appearing to have length and breadth but no depth.

4.12 Lerp

Lerp stands for linear interpolation. In our game, it is used to determine a moving object's position between two points based on the current time and a target time that it needs to reach the second point by.

4.13 In Game/Out of Game

The term "in game" will be used to describe whenever the player is currently playing a level. This means that they are fighting a boss while trying to sync their character's actions to the beat of the song. Any other game state (excluding the pause menu), such as choosing their team, buying equipment, etc. will be referred to as "out of game".

5. Relevant Facts and Assumptions

5.1 Relevant Facts

- The only way to control the game mechanics of the product will be through keyboard and mouse.
- All art assets will be developed internally through the usage of photoshop software.
- This game will be a rpg game where it will be driven by a narrative. The game world will be linear one where the success will be measured through the characters experience points and the gold acquired during battles.

5.2 Assumptions

- A product ready for release is not required. Due to the time constraints and only 15 minutes of gameplay is needed, a polished game that is expected of video game companies is not needed.
- Device utilized to develop the final product will be the device that will present the final product with all the software involved will be the same versions.
- The product will not be released commercially.
- The device being utilized to run the game will have some sort of device to output the audio produced by the game.

6. The Scope of the Work

6.1 Existing Inspirations

6.1.1 Gameplay

Our game is inspired by Theatrhythm Final Fantasy and Osu! Which both share similar gameplay and music.

6.1.2 Art Style

We are using a simple anime-style artwork inspired by games like Food Fantasy, with our own theatrical twist.

6.1.3 Tone

The tone is inspired by Touken Ranbu, sharing a calm and peaceful menu, with stressful paced battles. In this game, swords take the form of humans.

6.2 Multimedia Assets

Not all art assets are fully developed since this art is being developed by hand.

6.2.1 Characters

The character designs are all inspired by anime-style art found in anime and in mobile games developed by eastern studios. All the characters are personally designed and digitally drawn. There are the heroes of the game that take inspiration from musical instruments such as Acoustic, Sitar, Taiko, and Trumpet. The boss will take inspiration from sounds with names such as Clamour, Commotion, Uproar and Strident. The Composer is the artistic rendition of the character the user will be playing.

6.2.2 Stage Elements

Furthermore, the design of where the battles will take place will be shown as a theater's play. The prefabs included to help set the mood are the moon, sun, star, trees/bushes, stage, curtains, pillars, and skybox. The background will most likely be solid colours that change with the tempo of the song, but this has yet to be confirmed. In addition to the art enriching the environment, there are prefabs that are utilized in the mechanics of the game. These in-battle prefabs include shield, potions, score text, notes, and note bar. Also not hand drawn, these art pieces help demonstrate the state of the battle to the user.

6.2.3 Menu Design

The main menu and the different scenes other than the battle will be hand drawn. These areas include the Audition Room, Music Shop, Rehearsal Room. The designs are also inspired by similar designs found in games such as Food Fantasy. Currently only in sketches, these areas will be more fleshed out as we confirm the core game mechanics. For example, currently we have the Music Shop that will offer items to help the characters such as amp, speaker, mic, and armour that are also hand drawn art assets.

6.2.4 Sound Design

The sound design is split into two categories. First we have the battle music that will be playing while battling the enemies. These songs were handpicked for the fast past game play that it will translate to for the note generation. Furthermore, these songs will thematically reflect the game such as the final boss having a more ‘epic’ soundtrack. The other section is the game effects such as the different actions the characters can do will have accompanying sound effects. Sword swing, magic being casted, a defend being done, and buying items are some of the sounds that will be implemented in this game.

6.3 Context of the Work

- The game being produced will be evaluated as the capstone project required to graduate;
- A working 15 minute demonstration is required when doing the final demonstration.
- This game will target players that have the necessary hand eye coordination so they can test themselves in a rhythm based mechanics and see if they can achieve the best score.
- Players when playing the game should feel satisfaction when performing

well and feel challenged enough that it requires their full attention when attempting to progress through this game.

- The game should reward strategic thinking when deciding on how to improve your own characters in this game.

7. The Scope of the Product

7.1 Product Boundary

The project should only encompass the interactions the player will have with the physical device to interact with the functionality of the game. This means the boundary will be the actions the user takes on the keyboard and mouse to play the game. Game interactions include the fast paced input needed to battle the monsters, the selection of the game characters and their respective upgrades, what items to buy on the item shop, and accessing menus to adjust the technical aspects of the game (ie. sound levels).

7.2 Product Use Case (PUC) Table

PUC	PUC Name	Actor(s)	Input/Output	Unit Tests
1	Selecting Option On Main Menu	Player	Key Input(IN), New menu (Out)	40-43
2	Adjust the slider for music volume	Player	Mouse Click (IN), boolean Success (OUT)	44, 45, 56, 57
3	Adjust the slider for sound volume	Player	Mouse Click(IN), boolean Success (OUT)	46, 47, 56, 57
4	Adjust the game display settings	Player	Mouse Click(IN), boolean Success (OUT)	48, 49, 56, 57
5	Change the key binding settings	Player	Mouse Click (IN), Key Input (IN), current Key bindings (OUT)	50-52, 56, 57

6	Adjust video and audio latency settings	Player	Mouse Click(IN)	53-55, 56,57, 76
7	Buying equipment from store	Player	Mouse Click (IN)	1-10, 12-14
8	Selecting Equipment in store	Player	Mouse Click (IN), Equipment information (OUT)	1-11
9	Selecting available characters in recruitment	Player	Mouse Click (IN), Character information (OUT)	15-23
10	Recruiting available characters	Player	Mouse Click (IN)	15-22, 24-26
11	Select character to be the one to fight	Player	Mouse Click (IN)	27-34
12	Select a character for more information	Player	Mouse Click (IN), Character information (OUT)	27-33, 35
13	Equip a character with a piece of equipment	Player	Mouse Click (IN)	27-33, 36
14	Select an upgrade in a character's upgrade tree	Player	Mouse Click (IN), Information about upgrade (OUT)	27-33, 37
15	Confirm an upgrade for a character	Player	Mouse Click (IN)	27-33, 38, 39
16	Select the next location to conduct a battle on world map	Player	Mouse Click (IN), Enter battle (OUT)	58, 59

17	Select character during battle to conduct the action	Player	Keyboard Input (IN), Character performs action (OUT)	NaN
18	Select action during battle to be conducted by a character	Player	Keyboard Input (IN), action selected shown (OUT)	87
19	View the next time a beat comes	Player	Beat timing shown (OUT)	103-105
20	Winning the battle	Player	Victory Message (OUT)	101
21	Failing the battle	Player	Failure Message(OUT)	101
22	Save the game	Player	Mouse Click (IN)	69-72
23	Load saved game	Player	Mouse Click (IN)	73
24	Pause the game	Player	Mouse Click (IN), Pause Screen (OUT)	74
25	Unpause the game	Player	Mouse Click (IN)	74
26	Exit game	Player	Mouse Click (IN)	NaN
27	Pause a cutscene	Player	Mouse Click (IN), Pause Screen (OUT)	60-64, 66
28	Unpause a cutscene	Player	Mouse Click (IN)	60-64, 67
29	Skip a cutscene	Player	Mouse Click (IN)	65, 68
30	Exit the level	Player	Mouse Click (IN)	75

7.3 Individual Product Use Cases

PUC: 1	**Event:** Selecting Option On Main Menu
Trigger	The player utilizes the mouse to select one of the options on the main menu.
Precondition	The game must be opened on the home screen.
Procedure	1. Move the mouse to the area on the screen representing one of the options of the menu. 2. Use mouse click to select the option.
Outcome	The player has selected one of the options and now has entered the corresponding section.

PUC: 2	**Event:** Adjust the slider for music volume
Trigger	The player utilizes the mouse to adjust a slider to affect the volume of the music playing in the background.
Precondition	The game must be in the settings section.
Procedure	1. Determine the current setting. 2. Display the slider with the previous value. 3. Move the mouse to the music slider and adjust it with a mouse click.
Outcome	The volume of the game music will be changed by the adjustment from the player.

PUC: 3	**Event:** Adjust the slider for sound volume
Trigger	The player utilizes the mouse to adjust a slider to affect the volume of the sound effects played.
Precondition	The game must be in the settings section.
Procedure	1. Determine the current setting. 2. Display the slider with the current value. 3. Move the mouse to the sound slider and adjust it with a mouse click.
Outcome	The volume of the game sound effects will be changed by the adjustment from the player.

PUC: 4	**Event:** Adjust the game display settings
Trigger	The player selects the various options to adjust the display of the game.
Precondition	The game must be in the settings section.
Procedure	1. Determine the current settings. 2. Display the settings with their current values. 3. Allow the player to adjust the settings with a mouse click.
Outcome	The visual fidelity of the game will be adjusted according to the settings selected by the player.

PUC: 5	**Event:** Change the Key Binding Settings
Trigger	The player selects the specific keybinding.
Precondition	The game must be in the settings section.
Procedure	1. Determine the current key binding. 2. Display the current key binding. 3. Select the key binding wishing to change. 4. Input the new key for that key binding.
Outcome	The key binding will now be the player inputted key binding for that specific action in the game.

PUC: 6	**Event:** Adjust Video and Audio Latency Settings
Trigger	The user selects the “Latency Settings” option from the settings menu.
Precondition	The player is out of game or in the settings menu while in game and the game is paused.
Procedure	1. The player adjusts the video latency slider or audio latency slider. 2. Adjust the value beside the slider to reflect the new latency in ms and save this value as the new latency setting. 3. If the player clicks the value beside the slider, they can manually type the value of the latency they wish to use. 4. At any point, the player can press the X to close the latency settings option menu.
Outcome	The in game video and audio latency settings are changed to reflect the new settings given by the player.

PUC: 7	**Event:** Buying equipment from store
Trigger	The player chooses the buy option from the store.
Precondition	The game must be in the shop section, item to be bought is already selected.
Procedure	1. The buy button is pressed with item wanted selected. 2. Money removed from the player's balance. 3. Sound played to provide feedback. 4. Item is now added to player inventory.
Outcome	The player has acquired the item chosen to be bought.

PUC: 8	**Event:** Selecting Equipment in store
Trigger	The player chooses an item seen in the item shop
Precondition	The game must be already in the shop section.
Procedure	1. Items of the shop are determined and displayed. 2. The player selects the desired item. 3. Information is shown about the item on another view.
Outcome	The selected item is displayed with its attributes.

PUC: 9	**Event:** Selecting available characters in recruitment
Trigger	The player chooses a character available in the recruitment location.
Precondition	The game must be already in the recruitment location.
Procedure	1. The available players for recruitment are determined and displayed. 2. The player selects the desired character. 3. Information is shown about the character in another view
Outcome	The selected character's stats will be displayed.

PUC: 10	**Event:** Recruiting available characters
Trigger	The player chooses to recruit a character from the store.
Precondition	The game must be already in the recruitment location, and a character already selected.
Procedure	1. Select the buy option to recruit the selected player. 2. A sound will play to provide feedback. 3. The amount utilized to buy the character will be removed from the balance.
Outcome	The selected character will now be available for the player to utilize in battles.

PUC: 11	**Event:** Select character to be the one to fight
Trigger	The player selects one of the available characters to the player to utilize in the next fight.
Precondition	The game must be already in the rehearsal section.
Procedure	1. The player will select a character from the available list of characters to the player. 2. Then the player will select one of the displayed characters on the current lineup. 3. The character from the list will now appear in the position of the previous character. 4. The player will then confirm the new lineup.
Outcome	The selected character will now be in the lineup that will be utilized in the next fight.

PUC: 12	**Event:** Select a character for more information
Trigger	The player selects one of the available characters to the player to utilize in the next fight.
Precondition	The game must be already in the rehearsal section.
Procedure	1. The player will select a character from the available list of characters to the player. 2. The player will have the option to specify if they want to see the stats of the character or the current equipment equipped.
Outcome	The information of the selected character will displayed for the player to see it.

PUC: 13	**Event:** Equip a character with a piece of equipment
Trigger	The player will selects the add equipment button.
Precondition	The game must be already in the rehearsal section, with the character already selected and in the equipment tab.
Procedure	1. The player selects the button to add equipment. 2. A list will be displayed that will display the current valid equipment for the character. 3. The player will drag equipment from the list to the corresponding equipment slot for the selected character. 4. The player will confirm the changes to the equipment of the selected character.
Outcome	The selected character will now have the new equipment equipped.

PUC: 14	**Event:** Select an upgrade in a character's upgrade tree
Trigger	The player will select any upgrade that is available in the upgrade tree display.
Precondition	The game must be already in the rehearsal section. The character has already been selected and the upgrade tab has be opened.
Procedure	1. The available upgrade path for the character will be determined. 2. The upgrade path will be displayed to the player. 3. The player will select one of the upgrades. 4. The corresponding information of the upgrade will be displayed to the player.
Outcome	The selected upgrade will have information pertaining to the upgrade displayed to the player.

PUC: 15	**Event:** Confirm an upgrade for a character
Trigger	The player will select a button to confirm the upgrade for a specific character.
Precondition	The game must be already in the rehearsal section. The character has already been selected and the upgrade tab has be opened. An upgrade available has been selected.
Procedure	1. The selected upgrade with its information is displayed. 2. The player selects the confirm button to give the selected character that upgrade. 3. A sound is played to provide the player feedback. 4. The available upgrades for that character is decreased.
Outcome	The selected character will now have that upgrade.

PUC: 16	**Event:** Select the next location to conduct a battle on world map
Trigger	The player has opened the world map.
Precondition	The game must be already in the world map.
Procedure	1. The player will select one of the available locations to do battle. 2. The player will confirm the location of the battle. 3. A sound will be played to give feedback. 4. The game will enter the battle loading screen.
Outcome	The player has entered the battle.

PUC: 17	**Event:** Select character during battle to conduct the action
Trigger	The player will select one of the preconfigured keybinds relating to the current character the player wants to select.
Precondition	The game must be in a battle mode and the selected character must not be fainted.
Procedure	1. The player presses the key corresponding to the key they pressed. 2. Depending on the action already set, the character will perform an animation corresponding to the action. 3. A sound corresponding to the animation will also be played. 4. If attack, magic, or ult, damage will be dealt to the enemy. 5. If heal then the character's health will go up.
Outcome	The selected character performs an action during the battle.

PUC: 18	**Event:** Select action during battle to be conducted by one of the characters
Trigger	The player utilized a preconfigured keybind to select an action to be performed by the characters in battle.
Precondition	The game must be in a battle mode.
Procedure	1. The previously selected action is determined and displayed as selected. 2. The player presses the preconfigured keybind for the wanted action. 3. The menu will display the newly selected action.
Outcome	The player has selected the next action to be performed by the next selected character.

PUC: 19	**Event:** View the next time a beat comes
Trigger	When the battle starts with the enemy.
Precondition	The game must be in a battle mode.
Procedure	1. The battle between the enemy and the player's characters starts. 2. The music for the battle starts playing. 3. The notes appear for when the action should be done appears on the action bar on top.
Outcome	The player will be able to view the next action should be performed according to the beat.

PUC: 20	**Event:** Winning the Battle
Trigger	The song ends and the player did enough damage to the boss/scored high enough to pass the level.
Precondition	The player is in game, the pause menu is not open, and the player has no special items or effects that prevent them from winning.
Procedure	1. Display game over animations (dying boss, player victory stances, game over text, etc.) 2. Display game over menu and stats screen telling the player that they won, their stats, and experience/awards granted. This menu has options to retry the level, continue to the next level, or return to the out of game menus. 3. If the player chooses to retry the level, restart the level as if they had selected it from the choose level menu. 4. If the player chooses continue to the next level menu, the game over menu closes and the next level start screen is opened as if they had navigated there from the choose level menu. 5. If the player chooses return to the choose level menu, the game over menu closes and the choose level menu is opened again as if they had navigated there from the main menu. 6. In either case, award the player the the experience and awards granted from winning, mark the level as complete, and grant them access to the next level.
Outcome	The player is awarded experience and awards for completing the level, the level is marked as complete, and they can choose to retry the level, continue to the next level, or exit to the choose level menu.

PUC: 21	**Event:** Failing the Battle
Trigger	All the player's character's health bars drop to or below 0.
Precondition	The player is in game, the pause menu is not open, and the player has no special items that prevent them from failing.
Procedure	7. Display game over animations (dying characters, boss victory stance, game over text, etc.) 8. Display game over menu and stats screen telling the player that they failed. This menu has options to retry the level or return to the out of game menus. 9. If the player chooses retry the level, restart the level as if they had selected it from the choose level menu. 10. If the player chooses return to the choose level menu, the game over menu closes and the choose level menu is opened again as if they had navigated there from the main menu.
Outcome	The player loses the level and gains no experience or rewards for completion and can either choose to restart the level or return to the choose level menu.

PUC: 22	**Event:** Save the Game
Trigger	The user presses the “Save Game” button from the out of game menu or the player reaches a designated checkpoint.
Precondition	The user is out of game or has reached a designated checkpoint..
Procedure	1. If the user pressed the save game button, the save game menu opens. This menu should have options to choose a save file to save to or return to the previous screen. 2. The user chooses a save file to save to. 3. If the save file contains data, prompt the user to confirm if they want to overwrite the data. 4. If they select yes, overwrite the save file with the new save data (game state including character’s, equipment, stats, progress, etc.). 5. If they select no, remove the prompt and let the user choose a new save state. 6. At any point, the user can press the X on the save game menu to close it. 7. If a checkpoint has reached and the game is auto saving, the currently loaded save file will automatically be overwritten.
Outcome	The user’s progress is saved to a designated save file of their choice or the currently loaded one if it is an autosave..

PUC: 23	**Event:** Load Saved Game
Trigger	The user presses the “Load Game” or “Continue” option from the out of game menu.
Precondition	The player is currently out of game.
Procedure	1. If a game is already loaded, prompt the player if they want to save their current game first. 2. If they select yes, follow PUC 22; otherwise, open the load game menu. This menu should have options for selecting a save file or returning to the previous screen. 3. If the user pressed “Continue”, open the previously loaded game file and set the game state to reflect the loaded save file. 4. If the user pressed “Load Game”, let them choose which save file they want to load and set the game state to reflect the loaded save file. 5. At any point, the user can press the X on the load game menu to close it.
Outcome	The user’s progress is loaded from a designated save file of their choice or the last loaded one is selected if they selected to “Continue”.

PUC: 24	**Event:** Pause the Game
Trigger	The player presses the escape key while in game.
Precondition	The player is currently in game and the pause menu is not open.
Procedure	1. Pause the game loop. (Stop playing the song and freeze any game logic like spawning notes, accepting character input, etc.) 2. Open the pause menu. This menu has options for resuming the game, changing the settings, or exiting to the main menu. It should also display their current progress in the level.
Outcome	The game is paused and the pause menu opens.

PUC: 25	**Event:** Unpause the Game
Trigger	The player selects the “Resume Game” option from the pause menu.
Precondition	The player is currently in game and the pause menu is open.
Procedure	1. Close the pause menu. 2. Resume game loop. (Resume playing the song and the game logic like spawning notes, accepting character input, etc.)
Outcome	The pause menu is closed and the game resumes from where it was paused.

PUC: 26	**Event:** Exiting the Game
Trigger	The player selects the “Exit Game” option from the main menu.
Precondition	The player is not currently in game and is simply idling at the main menu.
Procedure	1. Terminate the game.
Outcome	The games closes.

PUC: 27	**Event:** Pause a Cutscene
Trigger	The player selects the button attributed to “Pause Cutscene” from the bottom tray, while a cutscene is playing.
Precondition	The player is currently in a cutscene.
Procedure	1. Pause the game loop. (Stop playing the cutscene and freeze it’s progression) 2. Open the cutscene pause menu. This menu has options for resuming and skipping the cutscene.
Outcome	The cutscene is paused and the cutscene pause menu opens.

PUC: 28	**Event:** Unpause a Cutscene
Trigger	The player selects the button “Continue Cutscene” from the pause screen, while a cutscene is paused.
Precondition	A cutscene is currently paused.
Procedure	1. Close the cutscene pause menu. 2. Resume game loop. (Resume playing the cutscene and the game logic like spawning notes, accepting character input, etc.)
Outcome	The cutscene continues playing from where it was left off before it was paused, and the cutscene pause menu closes.

PUC: 29	**Event:** Skip a Cutscene
Trigger	The player selects the button “Skip Cutscene” from the pause screen, while a cutscene is paused.
Precondition	A cutscene has been initiated and is currently paused.
Procedure	1. Close the cutscene pause menu. 2. Resume game loop. (Resume playing the game as if the cutscene ended naturally)
Outcome	The cutscene ends and the gameplay resumes again, closing the cutscene pause menu.

PUC: 30	**Event:** Exit a Level
Trigger	The player selects the button “Exit Level” from the pause screen, while the game loop is paused.
Precondition	A level has been initiated and is currently paused.
Procedure	1. Open the pause menu. 2. Exit the level to the main menu.
Outcome	The level ends along with the gameplay loop, closing the pause menu.

8. Functional Requirements

8.1 Core Mechanics

ID: 8.1-1	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 19	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to view incoming notes.	
Rationale	All rhythm aspects of the game rely on the player being able to view the incoming notes based on the current song.	
Fit Criterion	Upon starting the level, the internal beatmap parser will display notes in accordance to specific frequencies of the song.	
Unit Test Number	77, 78, 116, 117	

ID: 8.1-2	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 17, 18	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player's battle commands must invoke the proper attack animations as a response.	
Rationale	If character performed animations that did not match the selected command the experience would lose immersion and feel incomplete.	
Fit Criterion	A command will trigger the appropriate in-game animation matching the command.	
Unit Test Number	85, 86, 87, 119, 120, 121	

ID: 8.1-3	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 18	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to input battle commands.	
Rationale	Without the ability to enter commands, the entire premise of the game falls apart. Strategic RPG elements would be eliminated, as well as all rhythm components.	
Fit Criterion	The player can choose varying battle options from the character's battle menu, based on what was customized prior.	
Unit Test Number	Not possible with NUnit (requires user input)	

ID: 8.1-4	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 17	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to choose what character performs the selected battle command.	
Rationale	Different characters will perform skills differently with varying levels of effectiveness. Thus, the player must have the freedom to decide which character performs what.	
Fit Criterion	Upon choosing the battle command, whichever character the player chooses will perform that command.	
Unit Test Number	Not possible with NUnit (requires user input)	

ID: 8.1-1	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 18	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to choose to either hold or tap the note.	
Rationale	Depending on what note is spawned in accordance to the rhythm, the player may be facing a single note or a held note.	
Fit Criterion	The player can either tap or hold down their battle command, signifying a single or a held note.	
Unit Test Number	Not possible with NUnit (requires user input)	

8.1.1 Primary Gameplay Mode

ID: 8.1.1-1	Type: Functional (Core Mechanics)	Priority: Very High
PUC: N/A	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The system must be able to calculate the accuracy of the player's inputted command.	
Rationale	The score the player will receive at the end of the song is dependant on how well the player timed their inputs to the beats of the song.	
Fit Criterion	Inputted commands will be rewarded "Good", "Great", "Perfect", or "Miss" scores based on the input accuracy.	
Unit Test Number	80, 115, 117	

ID: 8.1.1-2	Type: Functional (Core Mechanics)	Priority: Very High
PUC: N/A	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The system must be able to calculate the special gauge for each individual character in battle.	
Rationale	This gauge allows the user to perform ultimate attacks that will increase their score.	
Fit Criterion	The accuracy of the player's inputs will increase or decrease the characters' special gauge.	
Unit Test Number	84	

ID: 8.1.1-3	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 20, 21	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The system must be able to calculate the player's score by the end of the song.	
Rationale	The score the player will receive will depict how well the user did, progressing the gameplay.	
Fit Criterion	A final score will be displayed based on the user's tallied note accuracy.	
Unit Test Number	101	

ID: 8.1.1-4	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 21	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The system must allow the enemies to randomly attack the players throughout a song.	
Rationale	The boss's attacks are the main motivation for the player to ever having to defend. Failing to defend will result in a lower score.	
Fit Criterion	The boss will attack characters fairly at a well-paced frequency.	
Unit Test Number	92	

ID: 8.1.1-5	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 20	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to complete a level.	
Rationale	Without the ability to beat a level, the player will not unlock new levels nor progress in the plot.	
Fit Criterion	Upon finishing a song, the system will compare the final score to a requirement value, and provide the player with a success if it's reached.	
Unit Test Number	102	

ID: 8.1.1-6	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 21	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to fail a level.	
Rationale	Without the ability to fail a level, the player will finish the game quickly and not be punished for their poor execution.	
Fit Criterion	Upon finishing a song, the system will compare the final score to a requirement value, and provide the player with a failure if it's not reached. As well, if all present character's lose their health, the player will also fail the level.	
Unit Test Number	102	

ID: 8.1.1-7	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 8	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to gain currency upon level completion.	
Rationale	Currency will be used to purchase equipment for character customization, and is very important for progression.	
Fit Criterion	When a player completes a level, currency will be rewarded from the battle based on the user's score.	
Unit Test Number	127	

ID: 8.1.1-8	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 13	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The system must display the inventory screen.	
Rationale	Player's will not be able to customize their characters if they cannot see their options.	
Fit Criterion	When the inventory button is pressed from the menu, a grid-like menu will open displaying all currently owned items the player has collected.	
Unit Test Number	8, 14, 22, 26	

ID: 8.1.1-9	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 7, 8	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to purchase new equipment.	
Rationale	The entire point of currency is to trade it in for new equipment that can be equipped to characters to enhance them for harder levels.	
Fit Criterion	The player can enter a shop from the main menu, where they can click on various items. If they have enough currency, the cost of the item will be subtracted from the player's currency amount and the item will be added to their inventory.	
Unit Test Number	1-13	

ID: 8.1.1-10	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 9, 13	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to customise individual characters.	
Rationale	Characters need to be customized as they learn new skills and new equipment is purchased, scaling their progression with the difficulty of the game.	
Fit Criterion	Upon choosing a character, the player can choose to equip new items, replacing whatever is currently equipped. As well, the player can choose to select a new skill form the skill tree as active, replacing whatever the currently active skill is.	
Unit Test Number	36, 38, 39	

ID: 8.1.1-11	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 13	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The system must display the current equipment that each character has.	
Rationale	Player's will need context of what their party members currently have equipped for further team building.	
Fit Criterion	Upon choosing a character, the player can view the equipment they have equipped from a character-specific menu.	
Unit Test Number	35	

ID: 8.1.1-12	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 14	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able view a character's skill tree.	
Rationale	Player's will not be able to view their characters' progression unless the skill tree can be presented to them.	
Fit Criterion	When a character is pressed from the list of characters, a tree-like menu will open displaying current and future skills for that character.	
Unit Test Number	37	

ID: 8.1.1-13	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 14, 15	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The system must be able to update a character's skill tree.	
Rationale	Individual characters will not be able to progress in the game if their skill tree does not update upon level completions and EXP gain.	
Fit Criterion	When a player obtains EXP from a battle, the system will translate it on the characters' skill tree by unlocking skills.	
Unit Test Number	38, 39	

ID: 8.1.1-14	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 10	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to recruit new characters.	
Rationale	Aside from gaining mandatory characters from plot progression, the player can use the their currency to add a randomly selected character to their roster from a list of possible characters	
Fit Criterion	Upon entering the recruitment menu form the main menu, the user can use their currency to make the system choose a random character to add to their roster.	
Unit Test Number	15-25	

ID: 8.1.1-15	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 9, 11, 12	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to customise their team.	
Rationale	Teams need to be customized as new characters are unlocked and upgraded, scaling player's progression with the difficulty of the game.	
Fit Criterion	Upon choosing a character, the player can choose to equip new items, replacing whatever is currently equipped. As well, the player can choose to select a new skill form the skill tree as active, replacing whatever the currently active skill is.	
Unit Test Number	27-34, 36, 38, 39	

ID: 8.1.1-16	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 12, 14	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to gain EXP upon level completion.	
Rationale	EXP will be used to level up characters, and is very important for progression.	
Fit Criterion	When a player completes a level, EXP will be rewarded from the battle to the present characters.	
Unit Test Number	128	

ID: 8.1.1-17	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 12, 14	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to level up their characters.	
Rationale	Individual characters will not be able to progress in the game if they cannot level up, which affects their skill tree.	
Fit Criterion	EXP will be used to calculate each character‘s level, and measure their progression.	
Unit Test Number	38, 39	

ID: 8.1.1-18	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 12, 14	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The system must be able to convert EXP to a character’s level.	
Rationale	Individual characters require the system to use their EXP amount to calculate their level.	
Fit Criterion	When a player gains EXP, the system uses a predetermined formula to calculate if the player has leveled up or not. If not, it will display how many EXP points the character is away from a level up.	
Unit Test Number	83	

ID: 8.1.1-19	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 13, 15	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The system must be able to calculate the players current stats.	
Rationale	The player will be basing their team around varying statistics, so they need to be aware of how their equipment is affecting their party.	
Fit Criterion	Upon opening the player's menu, the system will calculate the various multipliers and display the stats on the screen.	
Unit Test Number	83	

ID: 8.1.1-20	Type: Functional (Core Mechanics)	Priority: Very High
PUC: N/A	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The system must preserve insight about character and party customization between level completions and menu selections.	
Rationale	If this information were to be forgotten then the user would have to redo work, resulting in frustration and loss of interest.	
Fit Criterion	Character data will be preserved in a database, so it can safely be loaded in and out for different sections of the game.	
Unit Test Number	Not possible	

8.1.2 Alternative Game Modes

The game does not have not any alternative game modes. The only mode is the primary story mode described in 8.1.1

8.2 Menus and Other Systems

ID: 8.2-1	Type: Functional (Menus and Other Systems)	Priority: Very High
PUC: 16	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to choose a level.	
Rationale	Whether the players wants to repeat previous levels or attempt new ones, they will need to be able to have a method to choose them.	
Fit Criterion	Upon entering the world map from the menu, the user can click on the level they wish to start playing.	
Unit Test Number	40, 41, 43, 103-114	

ID: 8.2-2	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 27	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to pause cutscenes at any time.	
Rationale	There must be a way for the player to take a break from watching cutscenes, in case they have something else to do.	
Fit Criterion	When the player presses the to-be-determined pause button, the cutscene will freeze until they wish to continue playing again.	
Unit Test Number	60-64, 66	

ID: 8.2-3	Type: Functional (Menus and Other Systems)	Priority: Very High
PUC: 28	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to unpause a cutscene at any point during its paused state.	
Rationale	There needs to be a method to resume the cutscene after it has been halted by a pause.	
Fit Criterion	When the player presses the same button used to pause the cutscene, the system will resume the cutscene from before the previously paused state.	
Unit Test Number	60-64, 67	

ID: 8.2-4	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 29	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to skip cutscenes.	
Rationale	This gives users the option to skip previous watched cutscenes or skip new ones that do not interest them.	
Fit Criterion	When the player presses the to-be-determined skip button, the cutscene will be skipped.	
Unit Test Number	65, 68	

ID: 8.2-5	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 24	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to pause the game at any time.	
Rationale	There must be a way for the player to take a break from the fast-paced rhythmic gameplay, where no breaks exist until level completion.	
Fit Criterion	When the player presses the to-be-determined skip button, the game will freeze until the player wishes to continue playing again.	
Unit Test Number	74	

ID: 8.2-6	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 25	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to unpause the game at any point during a paused state.	
Rationale	There needs to be a method to resume gameplay after it has been halted by a pause.	
Fit Criterion	When the player presses the same button used to pause the game, the system will resume the game from before the previously paused state.	
Unit Test Number	74	

ID: 8.2-7	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 24, 25	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The system must resume all tasks that were frozen from the pause.	
Rationale	The pause becomes pointless if the game's context cannot be preserved.	
Fit Criterion	All previously running tasks will be put into a sleep state upon pausing, which will restart operation after unpausing the game.	
Unit Test Number	74	

ID: 8.2-8	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 30	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to quit the current stage from the pause menu and return to the main hub.	
Rationale	Sometimes the player will lose interest in a current level or may need to quit to customize the team further.	
Fit Criterion	When pressing a particular button available on the pause screen, the level will quit and the player will return to the main hub/menu.	
Unit Test Number	75	

ID: 8.2-9	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 22	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to save the game.	
Rationale	If the player wishes to continue using their current game file during a different play session, they must be able to preserve their current states for next time.	
Fit Criterion	When pressing a particular button available on the pause screen, the current session's game states will be saved to a database.	
Unit Test Number	69-72	

ID: 8.2-10	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 23	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to load a saved game.	
Rationale	If the player wishes to use a game file from a different play session, they must be able to load in its preserved states.	
Fit Criterion	When pressing a particular button available on the main menu and selecting a game file, the selected session's game states will be loaded from a database.	
Unit Test Number	73	

ID: 8.2-11	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 26	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The player must be able to quit the game, and the system must safely return to the desktop.	
Rationale	An option needs to be provided to the user in case they no longer wish to play.	
Fit Criterion	When pressing a particular button on the main menu, the game will be quit and return the player to their desktop.	
Unit Test Number	Not Possible	

ID: 8.2-12	Type: Functional (Core Mechanics)	Priority: Very High
PUC: 26	Originator: Damian Angelone (Oct. 12, 2018)	
Description	The system must confirm that the player would like to exit the stage or the game entirely.	
Rationale	This precaution can save the user from accidentally quitting the game, losing unsaved progressed.	
Fit Criterion	When the user chooses to quit the game a precautionary prompt will appear on screen, requiring the user to confirm their intentions before quitting.	
Unit Test Number	70	

9. Look and Feel Requirements

9.1 Appearance Requirements

ID: 9.1-1	Type: Non-functional(Look and Feel)	Priority: Very High
PUC: N/A	Originator: Yifu Wu (Oct. 12, 2018)	
Description	The game should follow industry standard for RPG game	
Rationale	The user should be able to access play the game just like other RPG games. It should make user feel familiar with the settings.	
Fit Criterion	The game should include title screen along with menu screen, setting screen, inventory screen, character info screen and battle screen.	
Unit Test Number	Not Possible	

ID: 9.1-2	Type: Non-functional(Look and Feel)	Priority: High
PUC: N/A	Originator: Yifu Wu (Oct. 12, 2018)	
Description	The proper feedback response should be used in game	
Rationale	the game should be playable without sound, a certain level of feedback response should be implemented to tell player that his click is registered	
Fit Criterion	Once the button is clicked, the phone should be able to vibrate if vibration setting is enabled. During the battle, there should be a visual indicator to show user that he has tapped or missed the beats.	
Unit Test Number	Not Possible	

9.2 Style Requirements

ID: 9.2-1	Type: Non-functional(Look and Feel)	Priority: High
PUC: N/A	Originator: Yifu Wu (Oct. 12, 2018)	
Description	The theme and images used in the game should be appropriate.	
Rationale	Since the game is for players of all ages, there should not be anything sexual or violent included.	
Fit Criterion	All images used in game will be in traditional anime style which is rich in color. Spell and attack will only be flashy, and they will not have any blood spilling effect.	
Unit Test Number	Not Possible	

ID: 9.2-2	Type: Non-functional(Look and Feel)	Priority: High
PUC: N/A	Originator: Yifu Wu (Oct. 12, 2018)	
Description	The player should be able to easily find the desired features in the game at all time.	
Rationale	The layout of the game should logically make sense and simple. Since the game is for players of all ages, each panel within the game should be easy to understand and self-explanatory.	
Fit Criterion	All setting options should be sorted and listed in the setting panel which can be accessed by clicking setting button any time in the game. Battle panel should only be able to be accessed by clicking battle panel.	
Unit Test Number	Not Possible	

ID: 9.2-3	Type: Non-functional(Look and Feel)	Priority: High
PUC: N/A	Originator: Yifu Wu (Oct. 12, 2018)	
Description	The game should have a musical theme	
Rationale	Since all the in-game characters are spirits of instruments. The theme of the game should also be musical	
Fit Criterion	The battle stage is a stage. The background of title screen is the entrance of theater. The bosses and characters are musical related objects.	
Unit Test Number	Not Possible	

9.3 Requisite Assets

9.3.1 Audio

The songs used during the battle are from OSU! Which is a rhythm game on PC. For the background music and sound effects, we will borrow them from vimeo and freesound sites.

Reference:

https://freesound.org/browse/tags/sound-effects/

https://vimeo.com

9.3.2 Visual

All of the images and visual effects are done by our team. There are not any visual assets used in the game.

10. Usability and Humanity Requirements

10.1 Ease of Use Requirements

ID: 10.1-1	Type: Non-functional(Usability -Ease of Use)	Priority**:** Very High
PUC: 16-19	Originator: Pavle Arezina (Oct. 11, 2018)	
Description	Any user above the age of 12 should be able to easily use the basic core mechanics of the game.	
Rationale	The game controls should be intuitive and easy to use. The main battle mechanics should come naturally to the user since it is one of the biggest aspects of the game.	
Fit Criterion	The user should be able to within one intro battle be able to understand and successfully be able to utilize the game mechanics to win the other battles.	
Unit Test Number	Not Possible	

10.2 Personalization Requirements

ID: 10.2-1	Type: Non-functional(Usability -Personalization)	Priority: Medium
PUC: 2,3	Originator: Pavle Arezina (Oct. 11, 2018)	
Description	The user will be able to adjust the sound levels in the game.	
Rationale	Depending on the level of sound and music the user feels comfortable with, they will adjust it to the level they want. If not controllable the user might have an uncomfortable experience.	
Fit Criterion	The user is able to adjust the sound volumes to the level they desire it to be at.	
Unit Test Number	44-47	

ID: 10.2-2	Type: Non-functional (Usability -Personalization)	Priority: Medium
PUC: 4	Originator: Pavle Arezina (Oct. 11, 2018)	
Description	The user should be able to adjust the graphics of the game.	
Rationale	Due to each user having different system resources, some users can display better graphics than others. Allowing the users to choose the graphics depending on their systems allows for less frustration if preset to one display.	
Fit Criterion	The user is able to adjust the graphics settings to the level they desire it to be at.	
Unit Test Number	48, 49	

ID: 10.2-3	Type: Non-functional (Usability -Personalization)	Priority: Medium
PUC: 5	Originator: Pavle Arezina (Oct. 11, 2018)	
Description	The user should be able to customize the game keybindings.	
Rationale	While the default key bindings will be the recommended controls the user should use, the user might have a personal preference to what they think is better controls.	
Fit Criterion	The user is able to adjust the key bindings to what they want.	
Unit Test Number	50-52	

10.3 Learning Requirements

ID: 10.3-1	Type: Non-functional(Usability -Learning)	Priority: Medium
PUC: 7-21	Originator: Pavle Arezina (Oct. 11, 2018)	
Description	The user should be able to learn all of the game mechanics within the provided tutorials.	
Rationale	In game tutorial will provide the user with a basic understanding of the game mechanics to allow the user the initial step needed in mastering the game.	
Fit Criterion	The user should be able to successfully be able to understand all the different mechanics in the game after the in game tutorial.	
Unit Test Number	Not Possible	

10.4 Understandability and Politeness Requirements

ID: 10.4-1	Type: Non-functional(Usability -Understandability)	Priority: High
PUC: 1-26	Originator: Pavle Arezina (Oct. 11, 2018)	
Description	All language and prompts utilized in the project should be understood by any user.	
Rationale	Inability to understand the game will lead to users becoming frustrated with the game mechanics and will lead to a poor user experience.	
Fit Criterion	The user is able to understand all text in the game and grasp all game functionality without needing help outside of the game.	
Unit Test Number	Not Possible	

10.5 Accessibility Requirements

There is no accessibility requirements due to the game being played on a general PC where accessibility will be handled on a hardware level, not in scope of this project.

11. Performance Requirements

11.1 Speed and Latency Requirements

ID: 11.1-1	Type: Non-functional(Speed and Latency Requirements)	Priority: Very High
PUC: 17-19	Originator: Nicolai Kozel (Oct. 15, 2018)	
Description	The game must have minimal input lag.	
Rationale	The player's success in the game relies on them being able to issue commands in time with the song. If the input lag is sufficiently high, it will make the game extremely hard to play.	
Fit Criterion	The input lag feels natural and normal to the user.	
Unit Test Number	Not Possible	

ID: 11.1-2	Type: Non-functional(Speed and Latency Requirements)	Priority: Very High
PUC: 19	Originator: Nicolai Kozel (Oct. 15, 2018)	
Description	The game must display the notes crossing the hit indicator with minimal video lag.	
Rationale	The player's success in the game relies on them being able to issue commands in time with the song. If the video lag is sufficiently high, it will make the game extremely hard to play.	
Fit Criterion	The video lag feels natural and normal to the user.	
Unit Test Number	Not Possible	

11.2 Precision or Accuracy Requirements

ID: 11.2-1	Type: Non-functional(Precision or Accuracy Requirements)	Priority: Very High
PUC: 19	Originator: Nicolai Kozel (Oct. 15, 2018)	
Description	The game must spawn notes at the correct time specified in the beatmap in order to match up with the song. The spawn times will be clamped to a beat value (quarter, eighth, sixteenth, etc.) instead of a ms value.	
Rationale	If the notes are not spawned at the right time, they will appear to not be evenly spaced and will not be timed with the song. This is not musically natural and will feel clunky to the user.	
Fit Criterion	The notes spawn at the correct time and are properly and consistently spaced.	
Unit Test Number	77, 78, 116, 117	

ID: 11.2-2	Type: Non-functional(Precision or Accuracy Requirements)	Priority: High
PUC: N/A	Originator: Nicolai Kozel (Oct. 15, 2018)	
Description	The player's Good, Great, Perfect rating must reflect the actual timing of the user's command.	
Rationale	The player's score, experience and eventual progression is dependent on how well they do in a level. If they are timing their actions perfectly and only getting a Good rating, they will not be rewarded properly for their actions.	
Fit Criterion	The player's Good, Great, Perfect rating is accurate to 3 decimal places on when the user issued a command. (After accounting for input lag)	
Unit Test Number	80, 117	

11.3 Reliability and Availability Requirements

ID: 11.3-1	Type: Non-functional(Reliability and Availability Requirements)	Priority: Medium
PUC: N/A	Originator: Nicolai Kozel (Oct. 15, 2018)	
Description	The game should always be available when the user wants to play it since it is an offline game. In the future if is connected to a server, it should be available most of the time and only unavailable during server maintenance or updates.	
Rationale	Player's will be frustrated if they have trouble launching or playing the game. The game's commercial success relies on its reliability and availability.	
Fit Criterion	The game is available and always loads properly.	
Unit Test Number	Not Possible	

11.4 Robustness or Fault Tolerance Requirements

ID: 11.4-1	Type: Non-functional(Robustness or Fault Tolerance Requirements)	Priority: Medium
PUC: N/A	Originator: Yifu Wu (Oct. 15, 2018)	
Description	The player should not be able to break the game in any aspect	
Rationale	To increase player's game experience, the game should be have some degree of constraints and be able to satisfy/fulfill all reasonable requests from user.	
Fit Criterion	Player should not be able to select more than 4 characters on a team. Player should not be able to select more than one character to attack during the battle. The game should not crash when player obtain a large amount of in-game items at once.	
Unit Test Number	Not Possible	

11.5 Capacity Requirements

There are no capacity requirements for this project.

11.6 Scalability and Extensibility Requirements

ID: 11.6-1	Type: Non-functional(Scalability and Extensibility Requirements)	Priority: Medium
PUC: N/A	Originator: Nicolai Kozel (Oct. 15, 2018)	
Description	The game should be designed in a way that it is scalable to include more songs and levels and can be extended to include different modes like freeplay, and multiplayer.	
Rationale	If the game will be developed in the future, it needs to be coded in an effective way that allows developers to extend the functionality and add new features easily.	
Fit Criterion	The code follows coding standards of object oriented programming, is modular and implements the idea of encapsulation.	
Unit Test Number	Not Possible	

11.7 Longevity Requirements

ID: 11.7-1	Type: Non-functional (Longevity Requirements)	Priority: Medium
PUC: N/A	Originator: Nicolai Kozel (Oct. 15, 2018)	
Description	The base game should last at least 10 years without needing any kind of software version changes, API changes, etc.	
Rationale	If player's want to play our game and use it for a long time, it needs to be sustainable enough that it does not rely on any external sources that may become deprecated.	
Fit Criterion	The game should not rely on anything that will likely expire or become deprecated within 10 years.	
Unit Test Number	Not Possible	

12. Operational and Environmental Requirements

12.1 Release Requirements

ID:12-1	Type: Non-functional (Operational-Release)	Priority: High
PUC: N/A	Originator: Pavle Arezina (Oct. 15, 2018)	
Description	The product must be able to be released on the device it was developed on.	
Rationale	The final deliverable (release) is a presentation to the stakeholders, this can be done on the device utilized to develop the product.	
Fit Criterion	The release can be done on the device the product was developed on.	
Unit Test Number	Not Possible	

12.2 Expected Physical Environment

There is no expected physical environment requirements.

13. Maintainability and Support Requirements

13.1 Maintenance Requirements

ID:13-1	Type: Non-functional (Maintainability-Maintenance)	Priority: Low
PUC: N/A	Originator: Pavle Arezina (Oct. 15, 2018)	
Description	The product must be able to be maintained after presentation with regular updates to allow product release.	
Rationale	There is interest to continue the game after the final presentation so therefore the game must be developed in a way that allows regular updates.	
Fit Criterion	Updates are easy to produce after final presentation.	
Unit Test Number	Not Possible	

13.2 Supportability Requirements

The product does not require any outside support for the deliverables of this project. In game tutorials are included. In terms of after final demonstration, this needs to be re-evaluated.

13.3 Adaptability Requirements

The product does not require any need adaptability since it will only be utilized on the machine it is developed on. In terms of after final demonstration, this needs to be re-evaluated.

14. Security Requirements

ID:14-1	Type: Non-functional (Security)	Priority: Very High
PUC: N/A	Originator: Yifu Wu (Oct. 11, 2018)	
Description	The user's personal data should be stored encrypted in a highly secured database	
Rationale	No player should be able to access or modify data in the backend. It will break the game and cause data loss.	
Fit Criterion	Data encryption/hashing should be enforced to keep data safe. Do not grant any one permission to access users' data unless it is necessary.	
Unit Test Number	Not Possible	

ID:14-2	Type: Non-functional (Security)	Priority: Very High
PUC: N/A	Originator: Yifu Wu (Oct. 11, 2018)	
Description	The game should be constantly updated to prevent newly introduced virus to break the game	
Rationale	To prevent any malicious attack, we need to make sure the game is playable all the time. We have to keep tracking what malware has been recently introduced, and implement some countermeasures to prevent it from happening ahead of time	
Fit Criterion	Collecting information on new malwares, and prioritizing which malware should be fixed first. Have a schedule to make sure the patch/fix is constantly delivered.	
Unit Test Number	Not Possible	

15. Cultural Requirements

ID:15-1	Type: Non-Functional (Cultural)	Priority: High
PUC: N/A	Originator: Johnny Endrizzi - October 14, 2018	
Description	The application won't intentionally offend any ethnic or religious groups.	
Rationale	Offending ethnic or religious groups can cause controversy and backlash.	
Fit Criterion	The application won't intentionally offend any ethnic or religious groups.	
Unit Test Number	Not Possible	

ID:15-2	Type: Non-Functional (Cultural)	Priority: Medium
PUC: N/A	Originator: Johnny Endrizzi - October 14, 2018	
Description	The application will follow industry conventions and standards.	
Rationale	Applications that follow industry standards usually do better than ones that do not follow them.	
Fit Criterion	The application will follow industry conventions and standards.	
Unit Test Number	Not Possible	

16. Legal Requirements

16.1 Compliance Requirements

ID:16.1-1	Type: Non-functional (Legal-Compliance)	Priority: Very High
PUC: N/A	Originator: Pavle Arezina (Oct. 11, 2018)	
Description	The product shall adhere to all terms and conditions set out by Unity.	
Rationale	Unity license terms and agreements will be followed.	
Fit Criterion	Unity license terms and agreements will be followed.	
Unit Test Number	Not Possible	

ID: 16.1-2	Type: Non-functional (Legal-Compliance)	Priority: Very High
PUC: N/A	Originator: Pavle Arezina (Oct. 11, 2018)	
Description	The product shall adhere to all terms and conditions set out by the course and McMaster University's Academic Integrity regulations.	
Rationale	McMaster University's Academic Integrity regulations will be followed.	
Fit Criterion	McMaster University's Academic Integrity regulations will be followed.	
Unit Test Number	Not Possible	

16.2 Standards Requirements

ID: 16.2-1	Type: Non-functional (Legal-Standards)	Priority: Very High
PUC: N/A	Originator: Pavle Arezina (Oct. 11, 2018)	
Description	All standards set out by the professor and T.A. in regards to the deliverable required by this project.	
Rationale	The professor has devised expected standards needed to be met for each deliverable.	
Fit Criterion	The project deliverable will meet every one of these standards set out by the professor and T.A.	
Unit Test Number	Not Possible	

17. Project Schedule

Deliverable	Date Required	Components
Team Formation	September 11, 2018	This includes team members for this game will be decided.
Initial Pitch	September 18, 2018	This includes the general idea. It will be privately presented to Professor Carette and require his approval.
High Concept Doc	September 25, 2018	This includes high-level ideas for the game (mechanics, art, lore, desired demographics, etc.)
Demo 1 - Presentation	October 1, 2018	All the above deliverables will be presented to the class.
Requirements Doc	October 16, 2018	This includes all requirements that the game should include (functional, non-functional, legal, etc.)
Demo 2 - Play Test	October 23, 2018	All the above deliverables will be presented in the form of a working game.
Art Planning	October 30, 2018	This includes all art and assets that will be included in the game.
Design Planning	November 6th, 2018	This includes all design and design approaches that will be included in the game (art, features, interfaces, etc.)
Verification and Validation	November 20th, 2018	All above deliverables will be tested and validated.
First Prototype	November 4th, 2018	All above deliverables will be presented as a polished working version of the game.

*All dates are subject to change; multiple revisions of the above will also be due.

18. Risks

While this project does not have any large risks associated with monetary value, it does have risk associated getting a good grade. If the deliverables are not met on time and without good quality work, there is a possibility of failing the course and having to repeat it. Furthermore, risks to the completion of the project can include the deliverables of other courses competing with the already limited time. An especially busy week can influence the quality of one of the weekly deliverable of this course.

19. Costs

This project has no monetary cost associated with it. All current and future planned assets are open source or made by the group members. Furthermore all the tools utilized towards the project are free to use. In terms of time, it will take 8 months of out of all group members to complete the project with all of the deliverable entailed. This will include the coding, asset development, documentation, and presentation preparation.

20. User Documentation and Training

20.1 User Documentation Requirements

This project should not require any formal documentation to be able to play and enjoy the game. At most a list of requirements such as the minimum computer specifications and operating system needed to run the game should be clearly mentioned before. The game should be able to teach the user everything they need through ingame tutorials.

20.2 Training Requirements

This project does not require any training requirements. Any person with good enough hand eye coordination should be able to pick up this game and have a fun, challenging time with it. The users should also be already familiar with the hardware needed to run the game.

21. Waiting Room

Clearly if there was more time to develop, a longer playing time would have been desirable. To be able to flesh out the story of the game and to be able to give more complex challenges of the bosses. There is much more potential if we can add more complex upgrade paths, buyable equipment, and different characters to give a better replayability of the game. The given amount of time does not allow the developers to give the users of the game more options to defeat the bosses. In fact, most of the core mechanics of the game would stay the same but giving the option to allow the users different ways to defeat challenges is something that cannot be done within the given time frame.

22. Ideas for Solutions

- Allowing different group members focus on other external projects while the ones more knowledgeable over certain tasks can focus on this project.
- In house development of all assets should allow a more uniform assets throughout the game.