**PianoVs. Feedback Document**

**Visual Feedback**

Description:

The game requires a better feedback loop for when the player is doing good or poorly. The best way to showcase this is when the notes are being hit, we indicate a particle effect or an icon above the keyboard keys. The icons or particle effects would have the colors green or red. Green for a successful key press and a red for error keys. There would be a yellow particle effect or icon as well which would indicate if the player was close but not quite there.



**Voting pop up**

Description:

(Potential Complete Over Haul)

The voting system works but players are getting confused and having trouble understanding the process. It might be better to scrap the existing voting system and start brand new. Ideally what we would like to see is if a player wants to change a song they should be able to with a voting system. The moment someone rejects the idea the song will not be playing. The scripts to be looking out for this are PlayerMenuHandler and GameDataManager.

**Method One**

The moment there are two or more players and someone wants to change the song, the voting screen should pop up right away on all of the player screens. If there are 2 or more players voting for a song and one person rejects the song request, the song does not play.

**Method Two**

There is a bell notification that pops up above the menu button. When a player hits the bell notification, the voting screen appears allowing the player to vote on the song that is being requested.



**Button-based Song Selection**

In order to select a song, we don’t want to have a drop-down menu. Instead what we would like to have is a screen that shows what difficulty settings the player would like to play. The songs would be scrollable on the right side. The difficulty is set from 1 – 4. Once a difficulty has been selected then the songs select screen would appear showing all the songs that are considered to be difficult. The user will select one of the songs, and be able to play the song right away otherwise it would go into a voting system.



**Reduced Audio on AI When Players are playing**

Description:

Since the AI automatically plays whatever song is playing, it may be distracting to the players when they are trying to play alongside the AI. A solution to this would be to have the AI at a full volume only when there is no one playing. Otherwise, if there is at least 1 player playing a song, all AI should either be completely muted or simply have the volume greatly reduced.

**Improved AI Functionality**

Description:

Currently, the AI does not recognize held notes that overlap with other held notes (see image below for example). A potential fix to this would be to adjust the note detection raycast to detect held notes on a separate layer. Or a trigger could be used to detect when a note, that does not equal the current note, and play it.

Image Example:

**Improved Quit Menu**

Description:

The current method used to quit is simply a button that says “Quit Game”. This could be improved by adding either; a secondary window with a message reading “Are You Sure?”, or by requiring the player to hold down the quit button, similarly to the method used to open the menu.

**“Battle Mode” Implementation**

Description:

Battle mode is a game mode in which one player can play a sequence of notes, which is then sent to the other players as a “song”. It’s sort of a rhythm game “Simon says”. The core functionality is completed and resides in the; RecordedSong, SongRecording scripts with additional references in the IndividualKey Script. All that is needed is proper implementation into the game itself. This could be done through a separate option that players vote to switch to, or as an entirely separate game mode players can select upon initial startup.