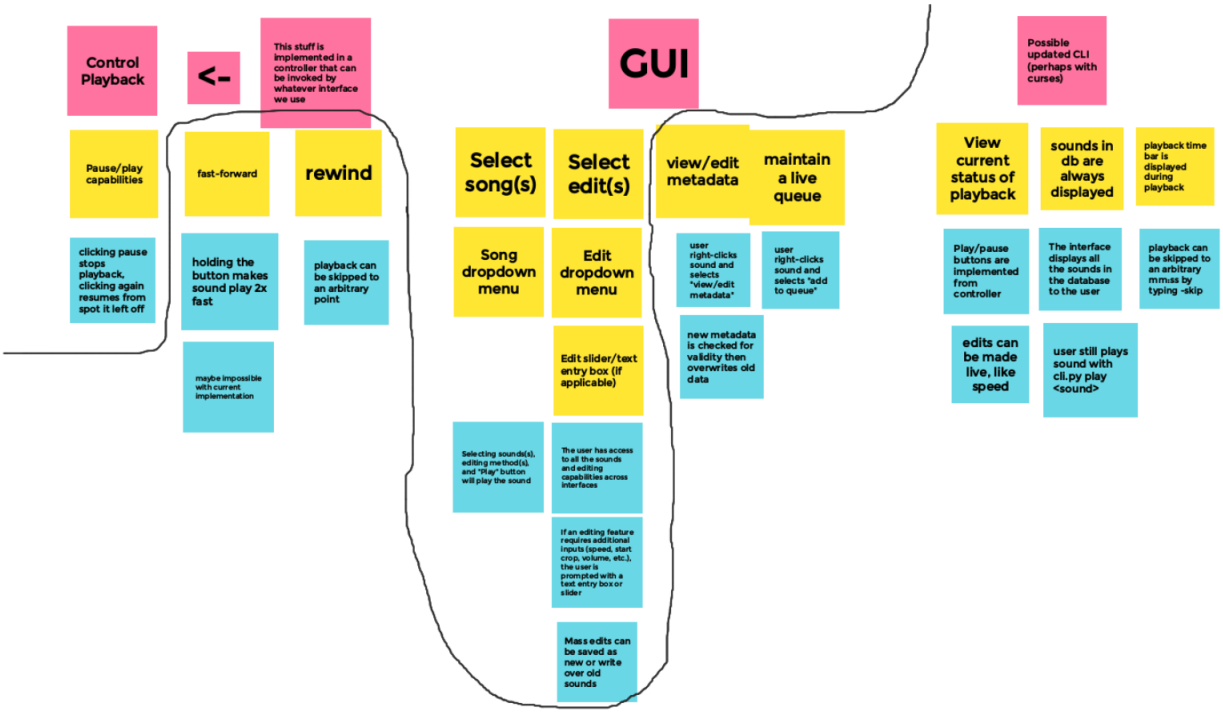
Andrew Tate, John Leeds, Luke Samuels, Rhys Sorenson-Graff

Epoch 3 Supplementary Documentation

Requirements:

1. Create a GUI
    a. The GUI does not need to have the same functionality as the CLI but should have all core functionality.
2. Users must be able to add sounds to the audio archive.
3. Users must be able to search for sounds.
4. Users should be able to use a fuzzy search to search for their sound
    a. Provide a search string and get a list of similar results.
5. Users must be able to apply audio edits to sounds.
6. Users must be able to play a sound and then return to the home screen of the archive.

Story Map:


Control Playback
<-
This stuff is implemented in a controller that can be invoked by whatever interface we use
Pause/play capabilities
fast-forward
rewind
clicking pause stops playback, clicking again resumes from spot it left off
holding the button makes sound play 2x fast
playback can be skipped to an arbitrary point
maybe impossible with current implementation
GUI
Select song(s)
Select edit(s)
view/edit metadata
maintain a live queue
Song dropdown menu
Edit dropdown menu
user right-clicks sound and selects "view/edit metadata"
user right-clicks sound and selects "add to queue"
new metadata is checked for validity then overwrites old data
Edit slider/text entry box (if applicable)
Selecting sounds(s), editing method(s), and "Play" button will play the sound
The user has access to all the sounds and editing capabilities across interfaces
If an edting feature requires additional inputs (speed, start crop, volume, etc.), the user is prompted with a text entry box or slider
Mass edits can be saved as new or write over old sounds
Possible updated CLI (perhaps with curses)
View current status of playback
sounds in db are always displayed
playback time bar is displayed during playback
Play/pause buttons are implemented from controller
The interface displays all the sounds in the database to the user
playback can be skipped to an arbitrary mm:ss by typing -skip
edits can be made live, like speed
user still plays sound with cli.py play <sound>

Use Cases:

Use Case Name	**Fuzzy Search**
Summary	A user inputs a search string and is presented with a list of sounds with similar names.
Rationale	Users can't be expected to remember the exact name of every sound they add. But, they will likely be able to get close. Alternatively, a user might want to pull up a list of all sounds that have a certain word in it. Using a fuzzy search, they will not get exact results but will likely get close to what they want.
Users	All users - particularly those with many sounds in the archive.
Preconditions	User has audio archive interface open and is ready to interact with the archive. This may be through the CLI or the search screen of the GUI.
Course of events	User runs python src/cli.py find [name] [number], where name is a required string to search for and number is an optional argument representing the number results to return. If the user is using the GUI, they type in [name] and are presented with a list of sounds.
Exceptions	The metadata database is not present - in this case, the user is told to run an initialization command.
Alternative paths	A user might use tags for a similar purpose. They might tag all sounds involving water with the "water" tag and search for the tag "water" as opposed to putting the word "water" in every sound involving water and doing a fuzzy search for "water."
Postconditions	The user is given a list of sounds sorted by edit distance to the string they searched for.

Use Case Name	**Audio Edit GUI**
Summary	A user is able to select audio edits to apply to a sound through a GUI.
Rationale	Typing out a long command to apply audio edits can be difficult and

	non-technical users might be intimidated by this. Using a GUI to apply audio edits will make our audio archive more accessible.
Users	People who use the GUI for the audio archive.
Preconditions	The user has selected a sound to play and clicks on a button to open an edit menu.
Course of events	In a new window, the user is shown a list of audio edits. The user can interact with sliders and check boxes to change the various parameters such as volume, start and end percent, and whether to reverse the sound. Once the user is satisfied with their edits, they exit the edit menu and can play their sound.
Exceptions	Users must only be able to submit values that make sense for audio edits. For example, volume must be a number. Or, start percent cannot be greater than end percent.
Alternative paths	The user can use the CLI instead of the GUI.
Postconditions	When the user clicks play, the audio edits are applied to the sound.

Use Case Name	**Play Sound GUI**
Summary	A user is able to play a stored sound using the GUI
Rationale	Similar to the audio edit GUI use case, some users might not like using the command line interface to play back their sounds, having a GUI makes this less intimidating.
Users	People who use the GUI for the audio archive.
Preconditions	The user has already added file paths to their sound archive
Course of events	In the main screen, the user selects “play”. After this the user will search for the sounds they would like to play. They select those sounds and click “submit.” After this the settings screen will open and the user will open the popup to select the sound effects they would like to apply as discussed in the previous use case. The user then clicks “submit.” Finally the play

	screen will open, at which point the user can click “play” to hear their sounds.
Exceptions	The user hasn’t added any sounds, at which point nothing will show in the search screen, or the user searches for a sound that doesn’t exist. In both these cases the play menu will appear, but clicking play simply doesn’t play anything.
Alternative paths	The user can use the CLI instead of the GUI.
Postconditions	When the user clicks play, the desired sounds play.

Use Case Name	**User adds sounds with the GUI**
Summary	A user adds a sound to the archive using the GUI
Rationale	The user needs to add sounds to the database if they want to play them later. They should be able to do this from the GUI, rather than add them manually with the command line
Users	People who use the GUI for the audio archive.
Preconditions	The GUI is open, and the sound is stored on the user’s machine with a valid file path
Course of events	1 . The user types the path to the sound they wish to add, either as an absolute path or relative from the directory they opened the GUI in. 2. The user presses enter. 3. If they wish to add more sounds, they must delete the file path they previously typed then enter another, and press enter again. 4. The user repeats this process for each sound they want to enter
Exceptions	If the user types an invalid file path, the commander catches the fileNotFoundException so the program doesn’t crash.
Alternative paths	The user can use the CLI instead of the GUI. The user can use their machine’s native file browser to select sounds
Postconditions	The sound is added to the database. The user can later search for it and

	play it.