The Flying Squirrels

Kenneth Arcayena ID:998828802
Ryan Babida ID: 999085233
Jovy Orbon ID:999681103

Breakdown of Programming Tasks	
Kenneth Arcayena	• GUI Format - Creating buttons/sliders/drop down menus/input boxes - Aesthetics (Colors) • Chopping - Implementing drop down menu to select sample - Input box to select time start chop and end chop - Ability to play chopped sample on Play/Pause button
Ryan Babida	• Playback - Play/Pause/Stop Button allowing user to play selected sample, pause and resume sample, and stop the sample and erase from playback • Stereo/Mono - Programmed the audio to evenly play sound through both speakers (Mono) or split some audio through left and right speakers (Stereo) • Tone Generator - Synthesizing a tone sound using a user-input frequency by implementing a sine wave function - Displaying the graph of the tone, demonstrating sine wave - Play/Pause tone button

Jovy Orbon	• Input/Sample Loading - Loading music files onto sample bay - Selection of sample for playback • Effects - Creating effects that apply to selected sample (Reverse, Speed Up, Slow Down, Tone Filter, Voice Removal, Echo Delay)
Group/ Integrated Tasks	• Individual Debugging • Group Debugging • Advice or idea contributions if help needed • Individual Error Coding • Group Error Coding

Project Functionality

Loading Inputs and Playing

On our GUI, we have a 3x3 sample panel where the user can load their desired audio. When the "load" button is clicked, the matlab directory will appear and the user must choose an audio file which has been placed in that directory. After the file has been selected, the file will then show up in the "sample" button box directly above the used "load" box. The user then has to click that "sample" button box to queue the audio file to be played.

Playback Functions

Next to our "sample grid" we have the "playback" button group that contains "play/pause", "stop", "stereo", and "mono". To play a sample, the user must load the file the same way as stated above. Queue the file by clicking the "sample" button that contains the user's desired audio then push the "play/pause" button. The selected audio file will now begin to play. If the user does not click the "sample" button that contains the audio file before pressing the "play" button, the file will not play and an error will pop up

that reads “Error: Please Select A Sample”. At any time the user wants to use the “play/pause” button in the playback section, the user must always select the “sample” button that contains their desired audio file then immediately after that press “play”. The “stop” button stops the audio and un-queues it as well. Let’s say the user loaded and then played a file, then in a different “sample” button box loaded another file, in order to play the new file the user must first click “stop” to un-queue the original played audio. Then after clicking “stop” the user must then again select the new “sample” button to queue it and then press “play” to successfully play the file. Now if the user wants to replay the stopped audio, they simply need to re-queue by selecting the sample and then press play. The “stereo” and “mono” buttons allow the user to hear the desired audio file in either stereo or mono format. However, stereo will always be selected due to the fact that stereo is the normal player. To switch to mono all the user has to do click mono and then press “play”. The audio will then restart in mono format.

Effects

Under the “playback” box, we have the “effects” box. In the effects box, we have a drop down menu and an “apply effect button”. We have several different effects that can be applied. To apply an effect to an audio file, the user must first queue the file, open the drop down menu, and select the effect they want to apply. Then they must press “Apply Effect” and then play. If the user does not press “apply effect”, the effect will not be applied. So the user MUST press “apply effect” in order for their desired effect to occur. Now to change the effect after already applying an effect before, the user must click the stop button to stop the file. Then re-queue the sample, change the effect, press “apply effect” and then press play. The audio file with a different effect will now play. On the other hand, if the user wants to change the effect of a current playing audio file that has already had an effect applied and goes through the process of selecting a new effect, but chooses the effect called “no effect”, the audio file will have the same effect as before because once “no effect” is selected, the file does not change from its last playing. So if the user wants to revert the audio back to its original state, the user must stop the current play, then simply re-click the sample to queue and click play.

Chopping

At the very top of our GUI we have the “chopping sample” panel. It contains a drop down menu, 2 texts box (for start and stop) and a chop sample button. To chop an already queued file, the user must open the drop down menu and select which sample it is that they are chopping, enter a time(in seconds) that they want to start the chop at, and enter a time(also in seconds) that they want to finish the chop at. They must then press “chop sample” then re-click the sample button in the “sample grid” and then press play. The end time that the user inputs cannot be greater than the length of the song, and it cannot be less than the user’s inputted starting chop time. If one of these occur, an error will display. The user also cannot re-chop an already chop sample. So to do so, the user must load the file onto a new sample button and restart the chopping process. If the user

wanted to revert the audio back to un-chopped, the user must also reload the file onto a new sample button.

Tone Generator

In the bottom right corner of our GUI we have our tone generator. In this box, we have a plot, "play" button, and an edit box. We put an edit box so the user can input a frequency in Hz. The edit box converts the input to a numeric value. This numeric value will be used to calculate a sine wave. As a result, the plot at the bottom will display a sine wave based on the given frequency. Also, there is a play and pause button for the tone generator that has the same behavior as the play and pause button from the playback feature.

Bibliography

http://homepages.udayton.edu/~hardierc/ece203/sound.htm

Digital Signal Processing

ALL GROUP MEMBERS SUMMARIES.

Kenneth Arcayena Summary:

For the project, since my group and I decided to split up the tasks for completion, I contributed the chopping and the GUI. I first created the GUI and input all the necessary features to have a successful project. After that I began to work on how to chop (start and end) a chosen sample with the assistance of Ryan Babida. Before doing so I needed to figure out how to get the size of the sound file that was being used to be in seconds. This is because I noticed that when I was playing with the code and displayed the size of the sound file, an abnormally large number appeared. So Ryan and I figured out a calculation that lead us to the size in seconds. Also, whenever any member of the group encountered a bug or error of some sort I contributed to try and help to fix what was needed.

R.B. Review- Kenneth's task was the first that needed to be completed and he was able to do so very quickly, which was helpful to the team to get a jump on the project. Kenneth was also very dedicated when it came to working on it.

J.O Review- Since I worked on the inputs and effects, Kenny was very precise whenever I had questions in regarding to the GUI and certain handles which should've been used.

Ryan Babida Summary:

My contributions to the project were the playback, changing the sound to stereo or mono, and creating the tone generator. I needed to create a play button that allowed the user to be able to play, pause, and resume. Then I needed to be able to get that or any selected audio file to either be played in stereo format or mono format. I did this by using a song where the sound generated from the left to right like a wave and then isolated it so that the wave sounding minimized. For the tone generator Kenneth created the plot for the GUI and I worked on how the plot worked. I also implemented a play button that also stopped the sound when the user clicked it a second time after first pressing play. I also assisted Kenneth Arcayena with the chopping of the selected samples. During the duration of the project programming, if one of my group members encountered an error or malfunction, I helped them fix it.

K.A review- Ryan was a big part of this project. Not only did he program successful tasks, but he was of great help when I encountered something that I could not figure out for the chopping task. Without Ryan, some of the project might not have worked in the way we needed it to.

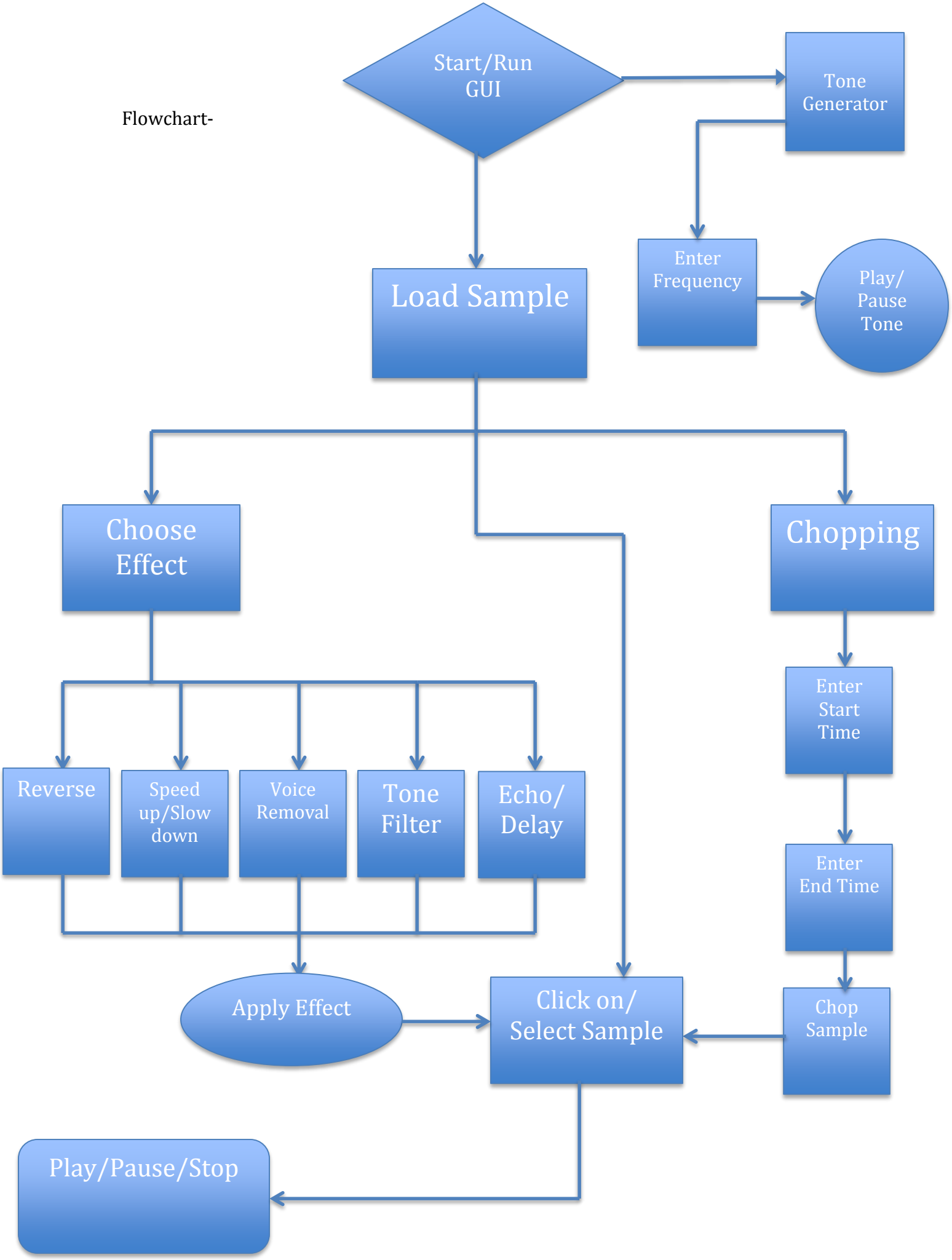
Jovy Orbon Summary:

In this project, I worked on the inputs and the application of effects. Before I began on the task of inputs, my group-mates and myself collaborated ideas for what types of handles would be needed. From there I worked on being able to load the song files into the GUI. After that, I worked on what effects we would have and their applications. Kenneth set up the effects menu and from there I programmed the codes necessary for the effects to be heard.

R.B review- Jovy was able to complete his inputs task earlier than I had expected, which was very helpful due to the fact that I worked on the playback aspect. He was also very helpful when I had questions in regards to some of the errors I encountered.

All team members have read the task summaries contained in this report and have been given an opportunity to comment.

Flowchart-


Start/Run GUI
Tone Generator
Enter Frequency
Play/ Pause Tone
Load Sample
Choose Effect
Chopping
Reverse
Speed up/Slow down
Voice Removal
Tone Filter
Echo/ Delay
Enter Start Time
Enter End Time
Apply Effect
Click on/ Select Sample
Chop Sample
Play/Pause/Stop