Milestone Project Plan

McKade, Robert, Taylor

Milestones	**Deadline**	**Tasks**
Graphical User Interface *Basic Frame* Creating the basic GUI frame	February 18	Robert ● Generate mockup ● Create Parts ● Link Parts
Graphical User Interface *Load/Save/Generate* Adding existing basic functions into the gui.	February 25	Robert ● Add load ● Add generate ● Add Save ● Add exit
Graphical User Interface *Settings/Modifiers* Adding user settings section into the GUI, altering how music is generated.	March 11	Robert ● Adding the basic section for settings and modifiers. ● Allow for easy modification, addition, and removal of settings
Graphical User Interface *Note Visualizer* Adding the basic note visualizer of generated or loaded midi file music.	March 25	Robert ● Come up with a table-like representation for notes. ● Display notes of generated music.
Graphical User Interface *Note Visualizer* Adding functionality to regenerate or recycle output.	April 08	Robert ● Allow for regenerating output. (new generation) ● Allow for recycling (use output as input)
Graphical User Interface *User note modification via note visualizer* Adding functionality to allow the user to modify the notes of a generation using the note visualizer as an aid Linked with UMM counterpart (1)	April 22	Robert & McKade ● Allow table to be modified by user before and after generation. ● Delete or add notes ● Recycle modified output

Note Recognition and Generation *Rests and note duration* Recognize rests and note duration in music and use it in music generation. Extending current R&G from equidistant quarter notes.	February 18	<u>McKade</u> ● Develop algorithm to find note durations. ● Develop algorithm to find music rests. ● Develop a way to use found information in music generation.
Note Recognition and Generation *Multiple Parts* Recognize parts of a music sample and integrate that with our music generation.	March 18	<u>McKade</u> ● Develop algorithm to recognize parts. ● Integrate into music generation.
Note Recognition and Generation *Chords* Recognize chords or stacked notes and use it in music generation.	February 25	<u>Taylor</u> ● Develop algorithm to find chords. ● Integrate into music generation.
Note Recognition and Generation *Key Signature Detection* Recognize an existing key signature or genre of a music sample and apply it to the music generation if the user desires it.	March 18	<u>Taylor</u> ● Develop algorithm and pattern finder to recognize a key signature. ● Integrate into music generation. ● Apply noise to make the generated music more different
User Midi Modification *User note modification via note visualizer* *Aggregating modifiers* Discusses modifiers and settings to influence music generation. Add possible weights to modifiers to have more influence.	April 8	<u>Robert & McKade & Taylor</u> ● Discuss possible modifiers ● Mod: Fitness bar range ● Set: Use Key Signature ● Set: Use Parts

Linked with GUI counterpart (1)		
Extra *Phrasing* Implement a generative scheme that allows for separate phases to be generated and joined together.	April 1	Robert, Taylor, McKade ● Develop a phrase generator ● Implement a model to join and repeat phrases
Extra *Fitness* Create a rubric to judge how similar or different a generated piece of music is from the original sample.	April 15	Taylor ● Develop comparison algorithm to determine how “similar” or “different” an output is from the sample. ● Will be used to give information to user.
Extra *Shannon Entropy Calculation* Research more on information theory to develop a more useful entropy when generating music.	April 15	McKade & Taylor ● Better define information in our music context ● Implement our entropy to reflect that definition

GUI Notes: Working with other group members to add relatvent code for new GUI additions. (not the GUI itself)