Report - Jacky Liu (jackyl1)

1) A clear statement of what you did.

- Searched through many .mid files of different genres online for training markov matrices.
- Designed a program to count notes and print out each track to determine which to use for training the markov matrix and to also determine the major scale of the .mid song by showing the count for how many times each note was played. (test2.srp)
- Designed the touchOSC interface for this program and optimized it to work with the different parameters in the playermain program intended for manual control during the concert, such as markov mixing weight, determining which song to use for the markov matrix, octave shift.
- Fixed bugs with the program such as crashing at weight=0.

2) A copy of the criteria by which your work should be evaluated.

- Picking sample music to represent each style
- Generating a library of Markov transition matrices from sample music], as well as storing them and reading them to/from disk for easy reuse.
- Design of API for handling updates to parameters via TouchOSC.
- Building the TouchOSC interface, and handling the updates between our program and the client.
- Testing the UI interactions.

3) An assessment of your work: Did it work? Was it good? How could it be better? Were there unsolved problems you encountered?

We believe that the process of building our program and implementing a complex algorithm to generate music through mixing two markov matrices went smoothly. However, the biggest challenge was that after we were finished, it didn't exactly sound interesting. This prompted us to add several things into the program to increase the complexity of the music rather than the algorithms, such as instrument changes, multiple instruments, and more program control. While we were satisfied in the end and met our criteria we had originally planned, it could have been better if we had more time to modify our algorithms to make the music seem more interesting and natural sounding.