Project Proposal

Project Description:

How not to Beethoven

My project makes use of the Kinect sensor to allow a user to control various aspects of the first movement (part) of Beethoven's fifth symphony. On can use their hands to essentially take role of an orchestral conductor to control the tempo, pacing, volume and intensity of the individual instruments involved in the piece as well as individually control each instrument's attributes to modify the sound of the final product in real time.

Competitive Analysis:

There are two instances of orchestral conductor simulators I've managed to find online but both control different aspects and are not so similar to my project:

The first one being an example presented in the Seattle Symphony - https://www.geekwire.com/2015/conducting-with-kinect-seattle-symphony-to-use-microsofts-3d-sensor-in-world-premiere-performance/amp/

This one has a Kinect sensor for each instrument individually. I will aim to combine this into my piece and have it so that we can control multiple instruments in a symphony and make it more "noob" friendly for the lack of better words.

The second project I've found http://cdm.link/2014/08/gestures-ableton-live-can-make-anyone-conductor-mendelssohn-behind-scenes/

This one involves a leap motion and is solely restricted to controlling the tempo of the track and the participation of each individual instrument.
I aim to provide a wider range of gesture control in my project.

Structural Plan:

Interface Display file:
Displays the user with his hands highlighted in the top right corner of the screen
Displays all the instruments that he has at this disposal.

Tempo detection class:
Has algorithm to calculate the tempo displayed by user's right hand

Gesture differentiating class:
Studies users left hands to detect if the user is showing a command gesture
Each command gesture will have its own method that modifies the attributes of an instrument accordingly

Instrument classes:
Each instrument in the movement will have its own class with its own attributes so that we can individually modify them

Instrument selection gesture detection file:
Uses the users hand gestures to calculate which instrument in the instrument they are pointing at and returning the choice.

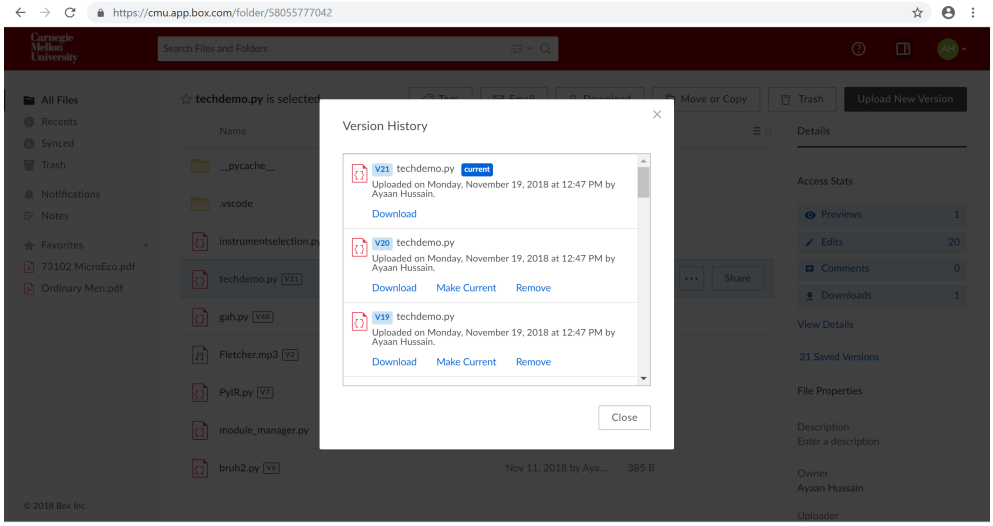
Algorithmic Plan:

The trickiest part of my project would be the implementation of the advanced gestures: basically the aggressive playing feature. The gesture involves bringing down the hand very hard to indicate an aggressive part of the song. The feature involves the detection of which instrument the user is pointing towards and checking if the tempo has been established already or not. Also if the user wants the entire symphony to play hard, he shall have to spread his arms out in sort of a Jesus pose and the symphony would move into a decrescendo to prepare for heavy playback ie increased volume and intensity. If a tempo has not been established, instead of having ten samples to average the tempo from, the new tempo shall be determined by the aggressive maneuver and will drift towards the correct tempo after the aggressive part is well past playing.

Part will require quick detection and calculation of tempo if not determined and then detection of which instrument to intensify as well. Tempo detection will basically need us to wait for the user to stop moving their hands for a while and noting that position and tracking till they change direction after bringing it down. Time between their start and end point will give us the tempo.
The Jesus pose detection will only trigger when the user stops shaking their hands and assumes that pose at which after a certain pause, to prevent its from triggering unintentionally, and then making the instruments sync up and start to fade out.
We will also have to keep track of the sync of each instrument and make sure each of them play at the same tempo and the same part of the song regardless of aggression level.

Version Control:

box.com automatic file version backup.



Timeline Plan:

TP1: instrument selection class
TP2: tempo detection, individual instrument volume control, implementation of suggested moves.
TP3: implementation of Easter eggs, polishing interface, advanced gesture detection

Module List:

pygame
PyAudio
PyKinect2
pydub (audio handler)