Project Description [5 pts]:
My project, Youkulele, is an application to help users compose their own simple songs and then learn how to play them on the ukulele!

Competitive Analysis [5 pts]: A 1-2 paragraph analysis of similar projects you've seen online, and how your project will be similar or different to those.

One similar project I have seen online is “UkeBuddy”. UkeBuddy lets you play different chords by pressing different buttons and shows the position on the ukulele. It also has other functionalities such as a chord namer/finder, a tool to learn ukulele scales and arpeggios, and a tuner. Another similar project is “The Ukulele App”. Similarly to “UkeBuddy”, the application has functionalities such as a tuner, a chord namer/finder, and a tool to learn scales. It also has many video tutorials teaching users to play different songs and chords.

My project will be different because it will allow you to not only play and hear different chords and notes but make a song out of these chords/notes. It will also allow the user to set a tempo for the song, and it will use pitch detection and audio analysis to ensure that you are playing the right note at the right time when it is teaching you to play the song.

Structural Plan [5 pts]: A structural plan for how the finalized project will be organized in different functions, files and/or objects.

There will be one file for the introduction screen, one file for the composing screen, one file for the learning screen, one file for all the objects such as note, button, etc, and one file for all the sound functions, i.e. playing a sound, overlaying notes and detecting pitch. There will also be a folder of all the sounds. There will be functions to determine which screen/mode the app is in, and based on that, the app will draw the corresponding UI, following the storyboard guidelines. There will also of course be functions to play the audio when the corresponding button is clicked, to detect the pitch for each note in a song, to set a tempo for the song, and to analyze if the user is playing the song at the correct tempo.

Algorithmic Plan [5 pts]: A detailed algorithmic plan for how you will approach the trickiest part of the project.

One of the trickiest parts will be teaching the user to play their song and detecting if the user has played the right notes. I shall implement the algorithm the following way:

Obtain song from composing file, loop through the chords/notes in the song and obtain the notes for each chord using a dictionary where each chord is a key and the notes comprising the chord are the values. For each note, record the user playing the note and then use aubio’s pitch detection to get the range of pitches in the recording. Find the most frequent pitch in the recording and return this as the pitch of the note played. Check if the pitch is the same as the pitch of the note meant to be played (with a marginal error of 10 hz) using a dictionary with the

frequencies for each note. If it is, then move onto the next note. If not, recursively call the check function again.

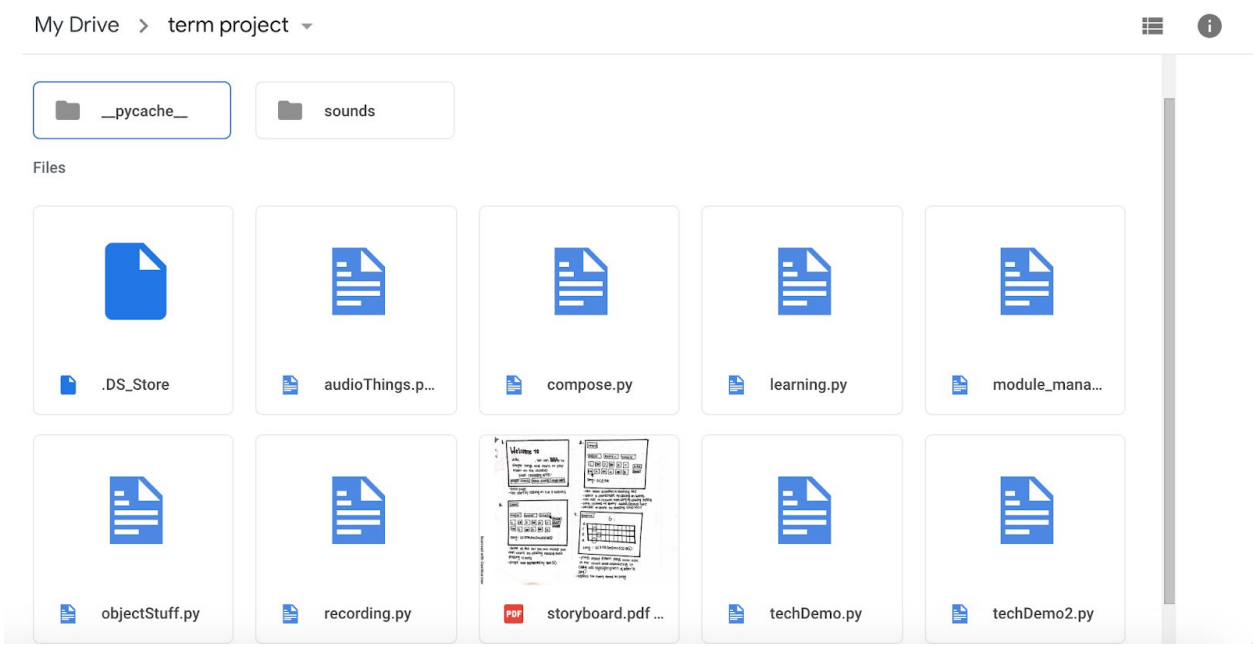
Another tricky part may be analyzing the audio for tempo. I will do this by recording the user playing the song, and then determining the time between changes in amplitude and frequency, as these changes are likely to be the user playing a new note. I will remove background noise from the recording before performing this analysis to ensure an accurate analysis. If the time between changes is not equal to the tempo set by the user, then ask the user to play it again faster or slower, depending on whether the time is less than or greater than the expected tempo.

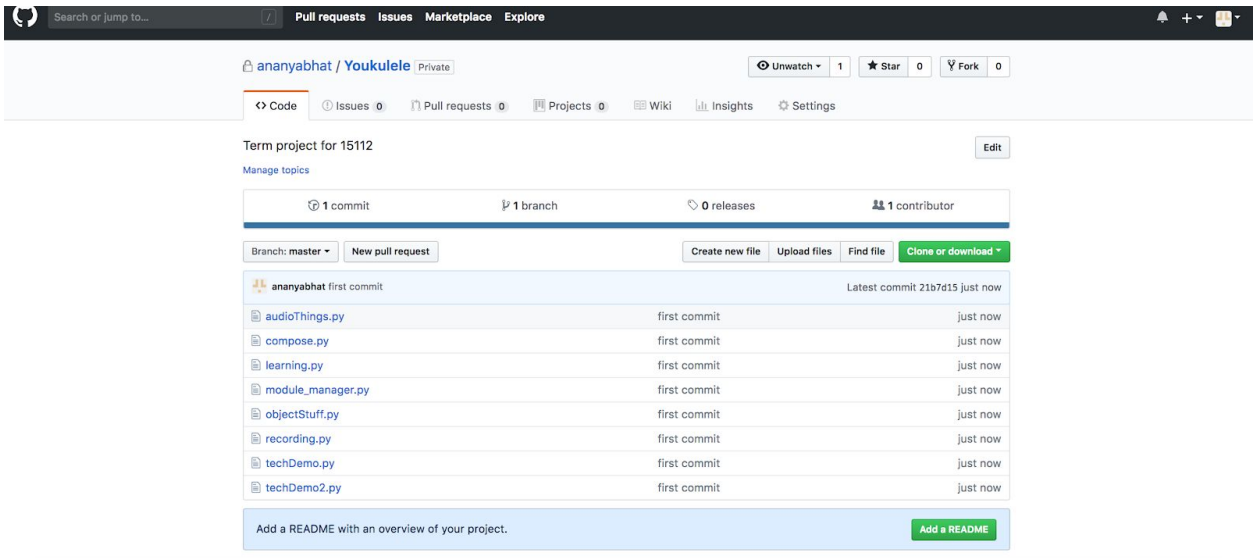
Timeline Plan [5 pts]: A timeline for when you intend to complete the major features of the project.

Playing audio* - 11/16
Detecting pitch* - 11/16
Overlaying notes* - 11/16
Basic visual for composing screen - 11/20
TP 1 - 11/20
Basic visual for learning screen - 11/22
Tempo detection - 11/24
Animation for learning screen - 11/26
Animation for tempo - 11/27
Introduction screen - 11/27
“Merge” screens (allow user to go from one to the other) - 11/28
TP 2 - 11/28
Refine UI - 12/1
Allow user to save songs - 12/3
Create point system for learning pitch and tempo - 12/4
TP 3 - 12/6
*These are all required for the tech demo, which is why I plan to finish them by 11/16

Version Control Plan [3 pts]: A short description and image demonstrating how you are using version control to backup your code. **You must back up your code somehow!!!**

I will be using Google Drive to backup my code. Every day that I work on my code, I will re-upload it to a google folder containing all my files for the term project. If I ever lose my code or I need to go back to a previous version, I can simply go to my google drive folder and obtain the code through the version history. I will also be doing a second back up on GitHub, just to be safe. I may not, however, update Github as often since I have the plan for Google Drive already.





Module List [2 pts]: A list of all external modules/hardware/technologies you are planning to use in your project. Note that any such modules must be approved by a tech demo. If you are not planning to use any additional modules, that's okay, just say so!

I am planning to use pyaudio and aubio to play sounds, overlay sounds, and detect pitch.

TP 2 Update:
While I originally planned on adding a feature that would track tempo, aubio does not have a tempo feature that works well in real time, and so I have decided to forgo that feature. Instead, I will also try to write my own pitch detection algorithm from scratch in order to make the project algorithmically complex. I also made a minor design change in the learning screen — instead of

the program automatically playing the note and then automatically transitioning into pitch detection, I made it so that these actions are triggered by the user, so it is easier for the user to learn at their own pace.

TP3 Update:

I wrote the pitch algorithm myself and added a new feature that suggests chords to add to your song based on the common chord progression pattern I-V-vi-IV using trie data structures. I also added a screen where the user can hear the song played back, and I made it so that actions are triggered by key presses instead of buttons in the learning screen (in my storyboard, it had buttons).