Daniel Freake

*CMU ID: dfreake*

**Pyano**

**PROJECT DESCRIPTION**

My term project is an application that would be able to do a variety of piano related commands (and other instruments given time). Primarily the application will record piano pieces and print out sheet music for the music. The application will then allow the user to edit the sheet music by either clicking and moving notes or by using commands. The application will also allow for the playback and saving of sheet music.

**COMPETITIVE ANALYSIS**

There have been a few 112 projects that include aspects of my project, such as a music sheet composer for pianos or a music sheet reader. Online there are a variety of programs/applications that perform the tasks I wish to implement. My project will be different in two ways. Firstly, it will have the ability to switch between instruments (piano, guitar, violin, …). Secondly, it will incorporate all the music sheet editing tools into one application. That is to say that the application will enable music sheet creation, editing and playback.

Given time I would hope to take my project a step further than anything I have found by enabling it to analyze spectrograms in order to separate out many pitches at once as well as possibly separate instruments.

**STRUCTURAL PLAN**

The project will be structured into several different sections. The main sections are listed below.

1. A section containing a music analysis tool, which will allow the application to record and analyze music. This section will output the info required to create sheet music (notes, bpm, beats per measure, …). This section will be structured as multiple python files that record and then analyze sound.
2. A section that uses objects to create the sheet music from data given by the first section. This section will be structured as a python file that creates sheet music in the draw function of an object.
3. A section that enables graphical interaction, enabling the user to begin recording from the home screen or to access the help screen. This section will be structured as multiple python files that interact with each other.

**ALGORITHMIC PLAN**

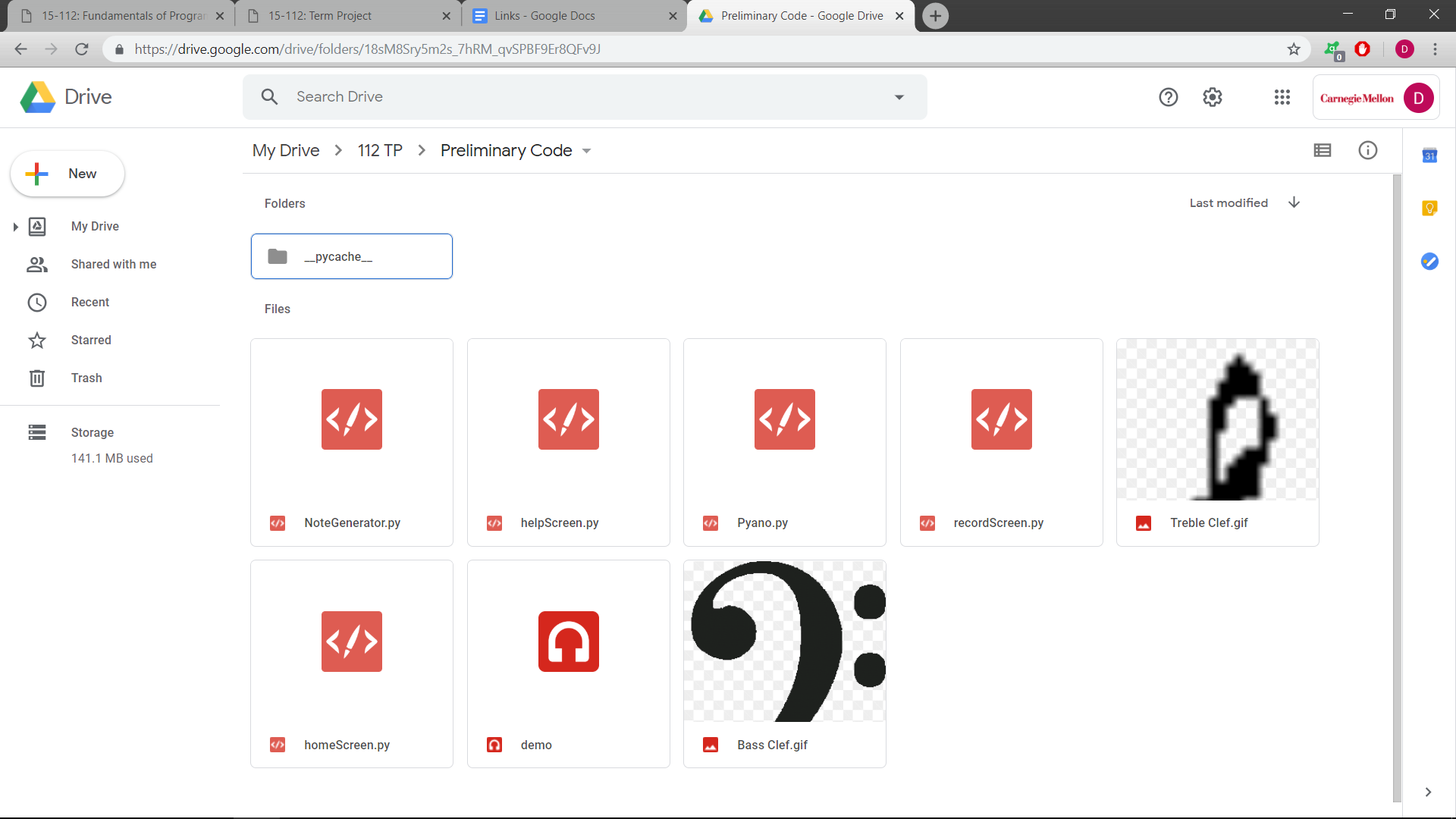
The trickiest part of the project will be analyzing the music and outputting the data required to create sheet music. I will do this by using several *aubio* functions. I will find the appropriate note by using either *aubio.pitch* or *aubio.note*. This will take some fine tuning to make sure that sound pollution does not get recorded by the application. I will find the bpm and beats per measure by either analyzing the output from *aubio.pitch* or by using *aubio.beats*. If I find that *aubio* does not provide me with the appropriate level of certainty when analyzing sound. I can always use *librosa* (a library that analyses sound and generates music).

**TIMELINE PLAN**

|  |  |
| --- | --- |
| **Completed by** | **Task** |
| TP1 | Have most of the graphical interaction completed (be able to create sheet music and have an interface). Also, have a grasp of the audio analysis and know where to go. |
| 04/19 | Have a working analysis section, that outputs the required data needed for sheet music production. |
| TP2 | Have the bulk of the application completed. This means being able to record sound and create sheet music from it. The application should also be able to play back and edit the sheet music by this point. |
| 04/26 | Have a good-looking UI. Begin to add instruments. |
| TP3 | Fix any problems associated with the application. Time permitting use spectrograms to increase the accuracy of the sheet music creation. |

**VERSION CONTROL PLAN**

I will be using CMU’s provided google drive to perform my version control (see image).



**MODULE LIST**

* PyAudio: enables recording and playback of sound by using the inbuilt microphone and speakers of a computer
* Pydub: enables simple sound analysis
* Aubio: enables more complex sound analysis

**TP2 UPDATE**

I realized early on in TP2 week that it would be virtually impossible to separate instrument sounds. This is because instruments playing the same frequency (or note) cannot be readily separated as the only thing that makes them sound different is reverb. Because of this I decided to only make my TP geared towards Piano. I did not envision that separating pitches (or notes) played at the same time would be so difficult. So even though I have a lot of code that attempts to do this, my TP2 code is geared towards analyzing simple single pitch music. I might consider employing librosa in order to analyze the music and get clearer readings (if I find that I remain stuck on this problem).

**TP3 UPDATE**

Made the playback function sound a lot less robotic by ‘stretching’ the sound files, enabling the files to sound very piano like. Ended up using aubio to get accurate onset and bpm measurements.