Plumid

A midi software interface for NEURAL DSP's ARCHETYPE PLINI

Developed by Nathan Jose

Integration Testing/Concept Implementation

November 3rd, 2019

- **Files**:
    - Miditest.py
        - Parsemidi():
            - This function uses the pygame library to setup communication with any connected MIDI controller and receive input from that controller. It first scans to check if there is any device connected. If not, it displays the appropriate error “Device not connected” message on the console. If it recognizes a connected device, it opens up a port to receive MIDI input from the device. If the input ranges from cc 1-10 it calls the ‘electric sunrise’ function and passes the midi input to that function. If the input ranges from 10-20, it calls the ‘worship’ function and also passes the midi input to that function. If the input isn’t within the specified range, it just waits until it receives an input in the given range.
        - Electric sunrise():
            - This function uses the pyautogui module to interact with the archetype plini software to load certain presets. It is called electric sunrise because it loads presets within a bank for a song with the same name. The function receives the passed midi input from parsemidi and based on the input, automates the mouse to click and load the appropriate preset.
        - Worship():
            - This function also uses the pyautogui module to interact with the archetype plini software to load certain presets. It is called worship because it loads all the presets that I use when I play the electric guitar in church. This function also receives the passed midi input from parsemidi and based on the input, automates the mouse to click and load the appropriate preset.
    - Tktest.py:
        - *This module is still under development*
        - main():
            - This function generates the GUI for plumid. It is still being worked on to reach full functionality and interaction with miditest.py. The function creates a window that contains the plumid logo, start and stop buttons to run/stop the program and a display to show the output messages from the console.