Priya Kudva
11/09/17

Programmer's Documentation

Running the Program:

In order to run this program, you must have the following downloaded and/or installed:

- Python 3 (Tkinter included)
- PyAudio
- Python "readchar" package
- sound.py
- set_clock.py
- arrow.pgm
- Sound files

With the packages included in the build path and the sound files, python files and image in the project directory, you can run this program. Using terminal, run set_clock.py, making sure to use Python 3.

Modifying the Program:

The only files that should be modified are the sound files, arrow.pgm and set_clock.py. Modification of any other file can compromise the construction of the program.

If you wish to change the sound files, the only requirement is that the new files must have the same properties: mono audio and 44100Hz. Though the frequency is not as important, ensuring that the file has only 1 channel (mono) will prevent errors in building the program and accessing that audio.

If you want to add more or different images, follow the construction of the arrow image in set_clock.py. All images must be either .pgm or .gif files. If you want to use photos in other formats, consider using the Python Imaging Library (PIL) to load and eventually convert them into Tkinter-compatible image objects.

To modify the GUI components, open set_clock.py to find the particular part of the GUI that you want to modify. To modify the size of the window, refer to the initialization of self.myContainer1. Currently, it is set to produce a window to fit the size of the computer screen but can be changed to any dimension after initialization.

To add more buttons or bind keys to buttons, refer to the initialization of self.button1 and the binding to self.myParent. Buttons are versatile objects, though the placement of buttons can be restricted if .pack() is used instead of .place().

To change the lists, refer to the creation of menuLabels. By default, using .pack() on labels aligns them on top of each other, so using .place() will allow you to arrange the labels side by side.

To change what each button/key press does, refer to the function button1Click(self, event=None). Currently, it moves the arrow and calls another function, report_event(self, event), to present the audio associated with what the arrow points to.

Finally, the last three lines of code start the GUI and keep it running.