Visual-Auditory Clock

Programmer's Documentation

Preliminary Project 3

Rajan Sawhney – Nov 11, 2015

This documentation assists in explaining the working of this clock which provides both auditory and visual information to the user in the process of setting up the clock. The explanations will allow the programmer to understand the program's functioning and make modification to it, as required.

The project has been built using the Python platform.

Contents

1. Program files
2. Important procedures
3. Important data structures
4. Setup requirements

1. Program files:

The Alarm clock uses the following .py files:

1.1 set_clock.py

- Starts the auditory interface to set the alarm. Contains all the sound objects and program for menu selection.
- Creates a User Interface (UI) to assist the user in setting up the clock
- The user can navigate across the menu using the keys J,K and TAB – on the keyboard and the user can also make use of the left mouse-click and SPACE key to make a selection.

  (author : Rajan Sawhney [on modifications of sample_menu.py authored by Prof. A Hornof])

1.2 sound.py

- Program to play the sound objects loaded in set_clock.py, through the python module pyaudio. Contains two main functions:
  1. play()
     Plays the audio file. User can press the keyboard buttons and chose to stop/skip the audio
  2. play_to_end()
     Plays the audio file till the end of file. If sound objects are played through this function, they have to be heard completely and cannot be skipped

  (author: Prof. A Hornof)

1.3 pyaudio.py

- Uses PortAudio bindings with the system to allow Python to play and record audio

(author: Hubert Pham)

1.4 tkinter
- The tkinter package ("Tk interface") is the standard Python interface to the Tk GUI toolkit.
- Allows creating UIs using the built-in functions

2. Important procedures

2.1 initUI(object) - Initializes the UI. Sets up the bindings of keys and events. Creates the grid for widget alignment

2.2 createWidgets(object) - Creates and aligns widgets in the grid in the passed object

2.3 keyPress(object, event) - Call report_event(object) when a keyboard key is pressed

2.4 report_event(event) - Informs of the occurred event, e.g. key press, mouse press, and call key_action(event)

2.5 key_action(event) - Depending on the event, carries out the required actions, e.g. traversal through menu and selection.

2.6 focusInHandler(event) - Called when a widget gets focus. Plays the appropriate sound file corresponding to the widget in focus.

3. Important data structures

Following data structures have been used in the program:

3.1	hour_sound_objects	- List containing the audio files for the hour menu
3.2	minute_sound_objects	- List containing the audio files for the minutes menu
3.4	days_sound_objects	- List containing the audio files for days of the week
3.5	visual_clock	- Class that contains most of the functions described above.

Audio files have been used to assist the user in understanding and navigating through the system. The .wav files have been obtained from the sample provided by Prof. A Hornof and some have been created using the following Text-to-speech websites:

1. http://www.text2speech.org/
2. http://www.fromtexttospeech.com/

4. Setup requirements

The pyaudio package has a dependency on the python version on the following platforms:

1. Windows – kindly install Python version 3.3.4 -32 bit(x86)
   Download link: https://www.python.org/download/releases/3.3.4/

To install pyaudio:

Run command in cmd or terminal

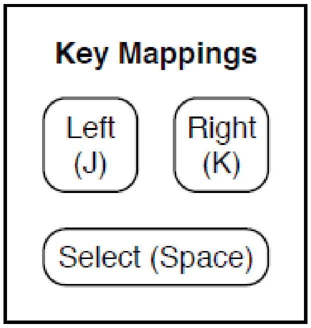
```
>> pip install pyaudio
```

Or download manually from: https://people.csail.mit.edu/hubert/pyaudio/
and run the .exe to install

To run the program:

1. Extract set_clock.zip to any directory.
2. Open the terminal / command prompt to the above directory
3. Run : >> python3 set_clock.py
   or
   python set_clock.py

To use the program:



The program will start by auditorily informing the user that they are in the "Main Menu"

-Press key 'J' or 'Shift' + 'TAB' – to navigate left
-Press key 'K' or 'TAB' - to navigate right
-Press key 'Space' or left mouse-button - to select (current functionality only highlights the button to cyan)

As the user navigates through the options, the program will auditorily and visually inform them of their position in the program and of their selection.