Personal information

- Time Management
- Kabir Bissessar
- 1007073
- CS-A1123
- 23/02/22

General description and difficulty level

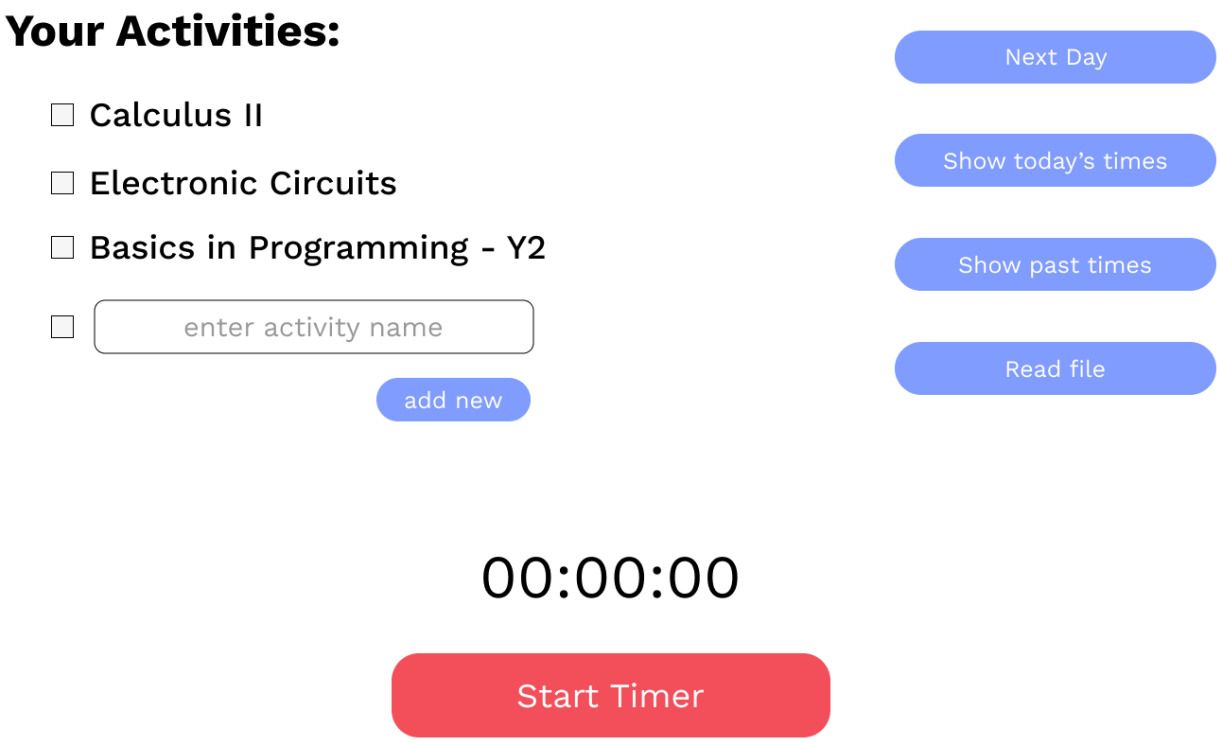
I will be building a program that allows users to time their activities, for example when studying for different courses.

I will be attempting the Medium difficulty for this project. This will include a graphical interface, having distinct activities (i.e. time can only be measured for one activity at a time) and unittesting for part of the program.

Use case description and draft of the user interface

The user will interact with the program through a Graphical User Interface. The user will be able to create activities using a text field, select an activity from a checklist and start/stop the timer using a button. The program will print its messages in the gui.

Graphic interface



Use case

A user studying for their courses in a period can use this program to track how much time they spend on different topics. The user can open the program, enter the name of the topic they are currently study and add it using the "add new activity" button. They can then start the timer and begin their work. When they are finished, they will end the timer using the same button (which will display stop timer). The time data will then be saved to a .txt file, which can be accessed in the future to read and write time data.

Program's structure plan

- Day Object
- Activity Object
- Time Object
- Data Object

Data structures

This program will use custom classes along with some builtin classes and data structures in Python.

A day's information will be stored in a dictionary in the form {'activity':'total time', }.

Activity times will be stored in lists.

Files and file formats

Data about days and activities will be stored in .json files, as these can be easily converted to python data types.

The list of .json files for different days can be stored in a .txt file.

Algorithms

Testing plan

The program will be tested by:

- Timing activities and ensuring accurate time is measured
- Reading activity and time data in files
- Writing activity and time data to files
- Adding activity and ensuring object is created
- Ensuring all elements graphical interface perform desired method

Libraries and other tools

PyQT5

PyQT5 will be used to create the graphical interface

Python Modules

- Time module
- json module

Schedule

- 12.03.2022 Classes should be configured; start programming methods
- 25.03.2022 Checkpoint 1 - classes should be working, activities can be created and time can be measured for an activity.
- 9.04.2022 Design of graphical interface should be in progress; some elements implemented
- 15.04.2022 Checkpoint 2 - Graphical interface should be completed, unittesting started
- 23.04.2022 Documentation should be completed, more testing in progress, graphical interface being refined
- 3.05.2022 Program should be fully completed and uploaded to submitted to A+

Literature references and links

- Python Documentation
- PyQT5 documentation