Final Assignment

Submission Deadline: 09/06/2021 at 23:59 CEST
This is an individual assignment.
Deliverables: `assignment7.py`, `assignment7_software_documentation.pdf`, `assignment7_user_documentation.tex`, `assignment7_user_documentation.pdf`, (other optional files)

Learning goals:
After successful completion of this assignment, the student can

- Break down a large problem setting into smaller subproblems (divide and conquer).
- Translate these subproblems in isolated and dedicated abstractions.
- Combine basic python data types, control flow, functions, classes, modules and packages, to create a coherent application.
- Write professional documents (e.g., documentation) using LaTeX.

In this assignment you will implement your proposed and approved final assignment, and write documentation.

Contents

1 Prerequisites

1.1 Presumed foreknowledge

- Anything covered in the course so far. Please refer to the weekly assignments if you need to brush up certain parts.

1.2 Getting started

This is an individual assignment. Please make an `assignment7` folder in your git repository, and make separate sub-folders for each of your group members. Add all the files related to your final assignment to your own sub-folder. Make sure that all your files contain your name and your group number (either as a commented header e.g. in a `.py` file, or as a title page e.g. in a `.pdf`).

When working on your final assignment, it is recommended to work in an Agile way. This means you implement your application in a dynamic and sequential way using short sprints, working your way to a working prototype early in the process. Only then you start expanding by adding additional features, and you only add them once they are thoroughly tested. For more information on Agile, see The Atlassian Coach (click here).

Additionally, it is recommended that you use your git repository accordingly. Once you have a working prototype, commit and push it to your repository to track the stable version of your application. Subsequently, you can work on a development branch (and potential feature branches) which you only merge into the stable master branch when you completed unit testing as well as integration testing.

2 The final assignment

In this assignment, you will implement the application that you proposed in assignment 5. Make sure to stick to the proposal and incorporate the feedback that you received from the teachers.

2.1 The requirements

Before you start working on the final assignment, make sure you are familiar with the requirements. The following requirements apply for your final assignment. Note, these are the same as those listed in assignment 5b:

- Your project should be clear, should be feasible within the scope of what you have learned in this course, and it should be feasible to carry out in a two week period.
- Your project should include some degree of user interaction like in the games that were implemented throughout the course. It is not required to implement a GUI; shell-based applications (like in the assignments) are sufficient.
- Your code should be modular, so you should make extensive use of functions and classes. Specifically, you should apply object oriented programming (OOP). As a rule, you should make at least 4 class definitions, and use inheritance at least once. This means that your project contains multiple components and that you work with several objects.
- Your code should handle errors (e.g., invalid user input), and should be tested for common bugs (e.g., unit and system testing). As a rule, your code should contain at least one try-except clause.
- Importantly, your code should be yours and original. Do not simply copy solutions that can be found on the web. We will run a plagiarism check and in the unfortunate case will inform to the exam committee.

- The project must involve extensive Python programming, and follow the naming conventions and docstring conventions as mentioned throughout the course.
- There is no requirement on the use of SQL. It is fine to use SQL, it is also okay if you do not.
- We expect you to generate the full software-documentation (e.g., using Javadoc or Sphinx). Additionally, you have to write some user-documentation using LaTeX that contains high-level documentation as well as example input/output (e.g., like in the assignments).
- Your project should be tracked in your git repository using meaningful commit messages and should be clear of redundant files.

2.2 Deliverables

2.2.1 Software

Implement the main "script" that runs your application in the `assignment7.py` file. Specifically, it should be possible to run your application from a shell by calling:

```
python assignment7.py
```

Any other required files that you produced yourself to run your application (e.g., custom modules and packages) should be submitted to Brightspace as well, and should be included in the git repository. Make sure your code follows the Python naming conventions as well as Python docstring conventions, and add comments at appropriate places in your code. Finally, make sure to remove redundant code.

2.2.2 Software documentation

Apart from your software, you should compile software documentation as one single file named `assignment7_software_documentation.pdf`. If you used a proper docstring convention (e.g., ReST or Google), it should be possible to directly generate the software documentation from your code by using Javadoc or Sphinx. In general, compile a document that describes the working of your application, providing insight in what classes, methods, and functions you developed. The target audience for this piece of documentation is a fellow developer that likes to continue your work.

2.2.3 User documentation

Finally, you should write user documentation using LaTeX. This should be one file that is called `assignment7_user_documentation.pdf` that can be compiled from its source LaTeX file that is called `assignment7_user_documentation.tex`. The user documentation should describe your application to a potential user. In other words, it is the actual user manual. For instance, if you implemented a game, you should describe its goal, rules, and how the user is supposed to interact with the game.

Good luck!