

## THE PROCESS OF DEVELOPING A SOFTWARE FOR DEAD RECKONING PROBLEMS USING PYTHON

Paul BURLACU, Elena-Grațîela ROBE-VOINEA, Georgiana-Cătălina CRISTEA

”Mircea cel Bătrân” Naval Academy, Constanța, Romania (paul.burlacu@anmb.ro, elena.robe@anmb.ro, grgn\_crst@yahoo.com)

DOI: 10.19062/2247-3173.2022.23.8

***Abstract:** This paper aims to give an insight into the process of building a software from scratch, using Python. In this respect, every stage of building a software will be presented thoroughly, namely: requirements analysis, the software design projection, the computational implementation, the assembling and conversion to a standalone application. Furthermore, there will be made a short presentation of the final product, NavyCalc software, along with its functions, in order to complete both the paper and the process of development.*

***Keywords:** software, development, Python, NavyCalc*

### 1. INTRODUCTION

Nowadays, technology is growing exponentially and therefore we are forced to align ourselves with this whether we like it or not. In this respect, we are obliged to make easy means of calculation, thus allowing a transition from classic to virtual in the most friendly way possible. This paper also seeks to demonstrate that technology is the way to an easier life and a powerful tool that has changed how teachers teach and how students learn.

### 2. REQUIREMENTS ANALYSIS

Requirements analysis is the first stage of the product development cycle, in which the requirements of the application are established, starting from the requirements of the end user, the functionalities of the future software product, as well as the data involved.

This stage practically answers the question “What will be achieved by developing this software product?”.

In this case, it all began with the idea of creating a program that helps both professors and students during the seminar at the Basics of Navigation classes, a subject that is studied in the first year of college. More specifically, the aim was to make the solution of navigation problems more efficient, by translating the calculations on the sheet into a special designed software, thus reducing time and the calculation errors as much as possible.

As this matter proposes a wide range of problems, the focus has been on the method of solving dead reckoning problems by direct estimation. The following versions of this software will cover all types of dead reckoning problems,

### 3. THE SOFTWARE DESIGN PROJECTION

The designing is that stage of product's development cycle that determines how the requirements identified in the analysis stage are met. In other words, it must answer the question "How will these requirements be met globally and in detail?"

This stage therefore starts from the requirements and specifications defined above, and continues with their detailing and transformation, until the structure of a solution is achieved, which could be represented by a graphic, textual, or mixed language. The program thus obtained must be able to be used further in the development of the software.

The design of the program was done entirely with the help of Qt Designer (FIG. 1). It is a free software that allows the construction of graphical interfaces (GUI), which can be later programmed using Python. This software is based on the "what-you-see-is-what-you-get" and "drag and drop" methods. Specifically, Qt offers a variety of widgets, that can be individually customized to meet the user's requirements.

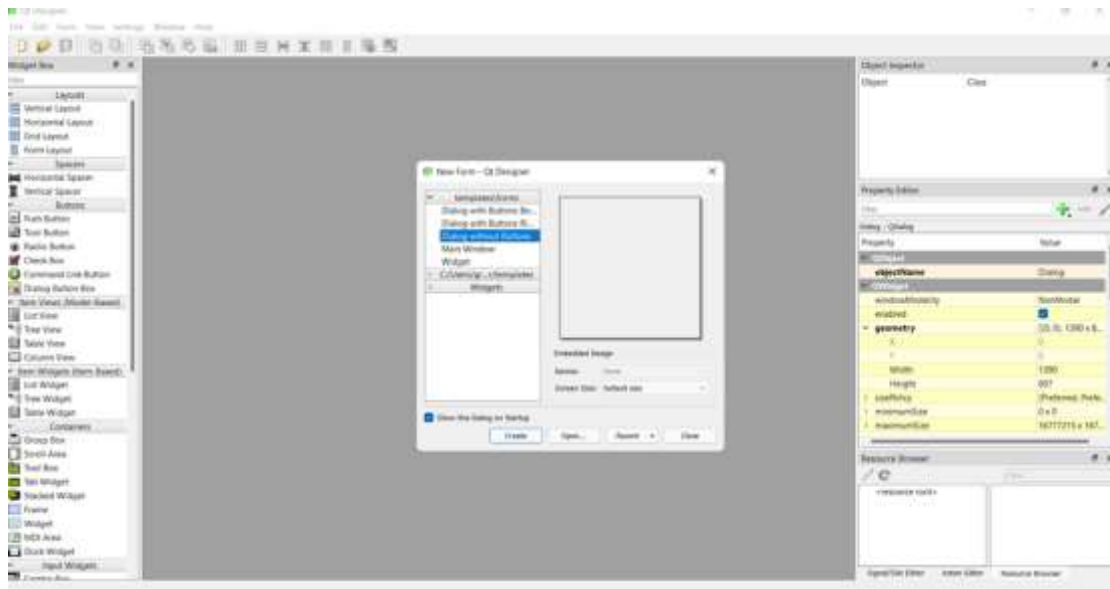


FIG. 1 Qt Designer

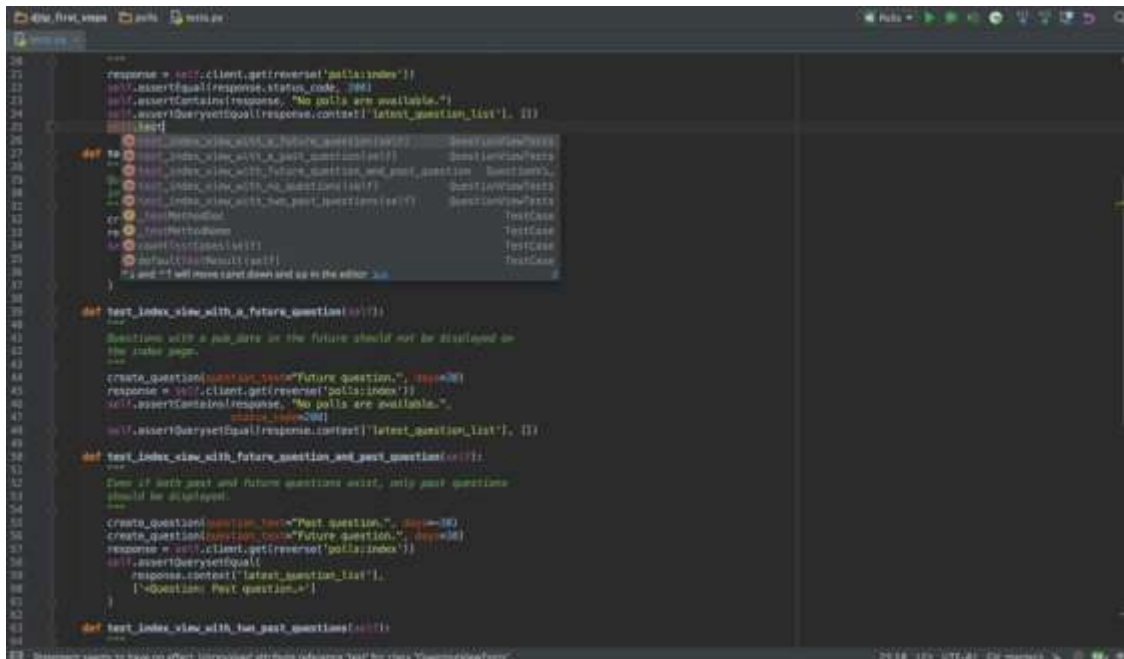
The NavyCalc program contains a variety of widgets, such as: labels (QLabel), used for written texts, buttons (QPushButton), with various functionalities, tables (QTableWidget), used for the display of both input and output data, etc. All of these, along with the additional settings, contributed to the final graphical form of the program.

Each window has been individually customized, and, finally, saved as a file with the .ui extension (i.e. main.ui).

### 4. THE COMPUTATIONAL IMPLEMENTATION

Generally speaking, the code implementation is the written text using a the format and syntax of a programming language. In our case, the chosen programming language was Python. Python is a dynamic, multi-paradigm, object-oriented programming language, that emphasizes code cleanliness and simplicity. Its syntax allows developers to express some programming ideas in a clearer and more concise manner than in other programming languages, such as C++.

For the development of NavyCalc, there was used PyCharm Community, which is a free development environment for Python users, offering a wide range of essential tools that serve to achieve any proposed goal.



**FIG. 2** PyCharm Community

One of the biggest advantages of this programming language is that it has a large number of standard libraries. The libraries underlying NavyCalc are PyQt5, datetime, sys, math, scipy, LatLon23, and pyqtgraph. The first library, PyQt5, is the mainstay of this application, as it is the bridge between Qt Designer and Python, which allowed the programming of the basic functionalities of our software. The other libraries were used to perform various calculations in the dead reckoning problems.

After programming all the features, the source code was saved in a file with the extension `.py`, more precisely, `main.py`.

## **5. THE ASSEMBLING AND CONVERSION TO A STANDALONE APPLICATION**

Following the design steps of the software architecture and the implementation of the code, a series of `.ui` files resulted, one for each window of the software, which are practically the result of working with Qt Designer development environment, and a `.py` file, which is the rough implementation of solving algorithms. One very important thing to mention is that this assembly stage is an optional one, because at the end of the implementation phase, we obtained a fully functional program. However, our initial goal was to obtain a software program that is as easy to use as possible, aesthetic and interactive. Thus, this assembly stage, in our case, became mandatory, because it allowed us to complete the program obtained in the executable program.

The particularity of the fact that the final form of the program is the nature of an executable is that it allows the user to install and transfer the program on any computer with Windows operating system, without the need to install a version of the programs used to develop the software on the computer.

In order to achieve this step, we used PyInstaller. More precisely, a Command Window was opened in the folder where the .py file is located, and the command “py -m PyInstaller main.py” was typed, thus creating the final file, NavyCalc.exe.

## 6. NAVYCALC PRESENTATION

NavyCalc software is comprised of 4 windows, “Home”, “Example”, “Help”, and “About”, each with a different structure and functionalities, adapted to the requirements, which together aim to provide a good user experience.

**7.1 Home window.** The home window, suggestively named, is the window that will appear everytime the application is opened. It is comprised of a variety of buttons, tables, labels, and a 2D chart. In the upper left corner, the menu can be found, which consists of 4 buttons: “Home”, “Example”, “Help”, and “About”, with which you can navigate between windows. Also, the title is located on the top of the page, which was chosen in such a way as to concisely explain the functionality of NavyCalc. The two buttons, “+” and “-“, help the user choose the number of sequences that the problem has. Following, the user has to enter the data that is provided in the first table. It should be noted that a certain data entry is required. For the sections corresponding to latitude and longitude, we will have the following form: “degrees-minutes-seconds.decimals”, immediately followed by the initial corresponding cardinal point (N for North, S for South, E for East, V for West), i.e. “43°12’12.13N” will be written as “43-12-12.13N” in the corresponding cell. In the column for time, the value will be written in the format “hh:mm:ss”, i.e. “12:30:00”. The rest of the columns, do not require any special format. The boxes with no provided values, should be filled with “0”.

Once this is done, by clicking the button “Calculate”, all the corresponding answers will be displayed in the second table, along with the value of the magnetic declination and a simulation of the map drawing in the two-dimensional graph.

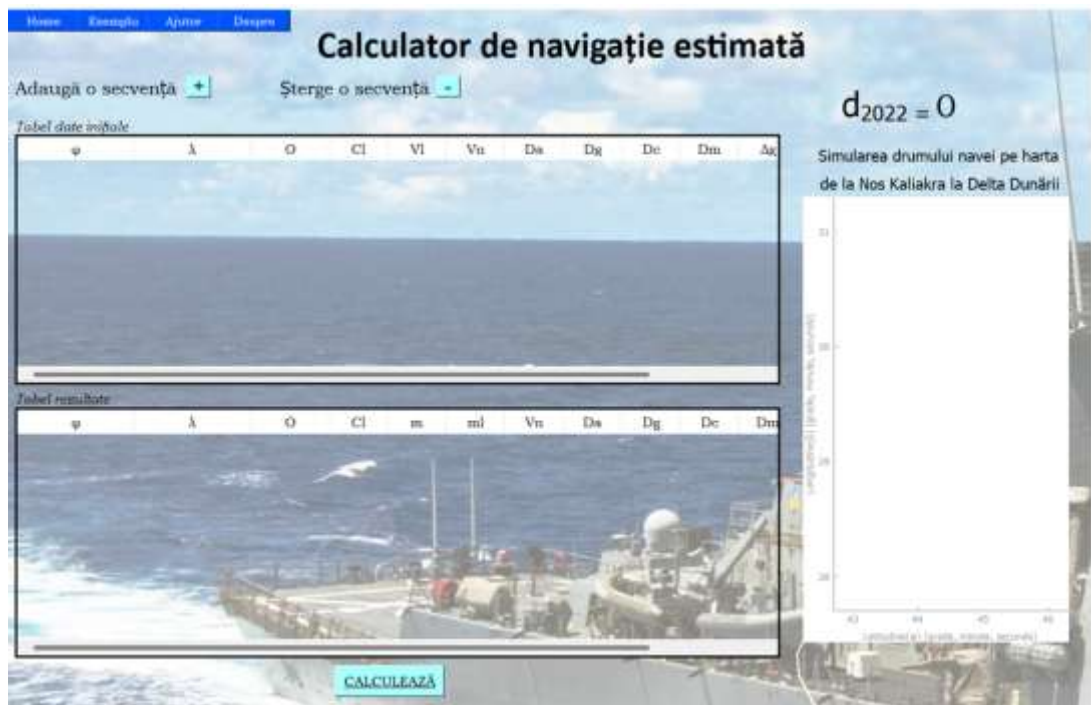


FIG. 3 Home Window

**7.2 Example window.** This window contains, as the title says, an example of a solved dead reckoning problem, more precisely, the final shape of the window, after pressing the “Calculate” button.

Regarding its composition, it has the same elements as the “Home” window.

In order to navigate to this window, one must press the “Example” button, located in the upper left corner.

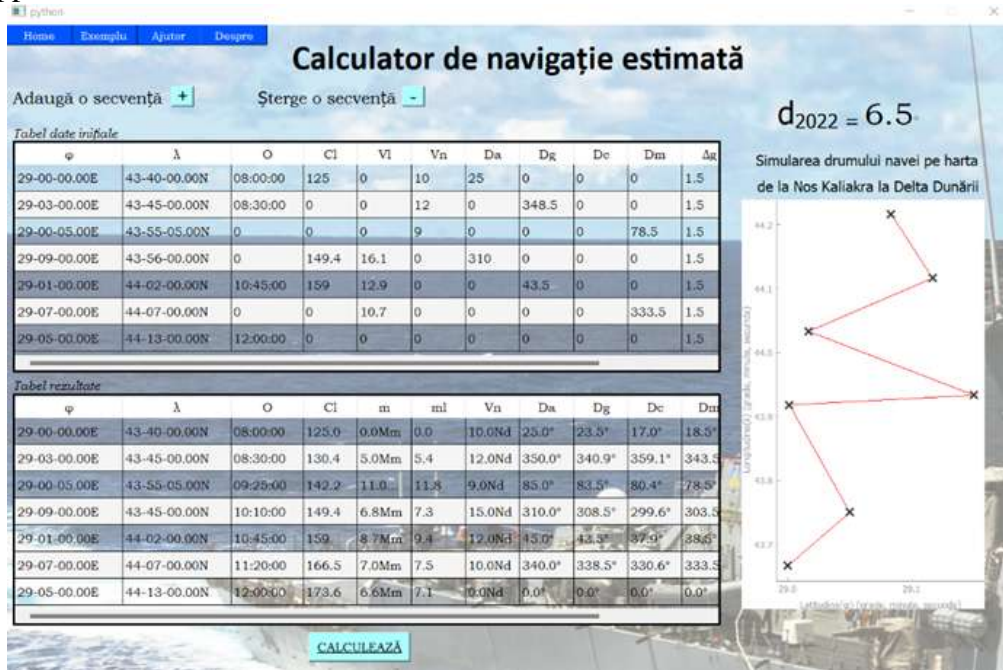


FIG. 4 Example Window

**7.3 Help window.** This window contains all the information you need to know in order to use NavyCalc.

It consists of a series of tags (QLabel), used to write the text.

In order to navigate to this window, one must press the “Help” button, located in the upper left corner.



FIG. 5. Help Window

**7.4 About window.** This window contains information regarding the purpose, development and the author of NavyCalc.

Its components are similar to the previous window presented.

In order to navigate to this window, one must press the “About” button, located in the upper left corner.



FIG. 6. About Window

## CONCLUSIONS

In conclusion, the purpose of this paper was to develop a modern way to solve dead reckoning problems through a completely functional and responsive software in order to make the students' and professors' lives easier.

## REFERENCES

- [1] Siahaan, Vivian, și Rismon Sianipar, *Learning PyQt5: A Step by Step Tutorial to Develop MySQL-Based Applications*, Indiana, Sparta Publishing, 2019;
- [2] William, Joshua M., *Creating GUIs with Qt Designer. Beginning PyQt*, Berkeley, Apress Publishing, 2020;
- [3] Meier, Burkhard, *Python GUI Programming Cookbook: Develop Functional and Responsive User Interfaces with Tkinter and PyQt5*, 3<sup>rd</sup> edition, Birmingham, Packt Publishing, 2019;
- [4] Moore, Alan D., *Mastering GUI Programming with Python: Develop Impressive Cross-Platform GUI Applications with PyQt*, Birmingham, Packt Publishing, 2019;
- [5] Moore, Alan D., Harwani, *Python GUI Programming – a Complete Reference Guide: Develop Responsive and Powerful GUI Applications with PyQt and Tkinter*, Birmingham, Packt Publishing, 2019;
- [6] Pine, David J Milton, *Introduction to Python for Science and Engineering*, Taylor & Francis Group, 2018;
- [7] A. Pintilie, P. Burlacu, E. G. Robe-Voinea, G. C. Cristea, *Standalone Application For Hull Volume Calculation Based On Simpson's Rule Using Pyqt5*, Scientific Bulletin of Naval Academy, Vol. XXIV 2021, pg. 128-134.