Application for Filtering Football Players
Made by Nedbay Pavel
Group BPAD 213-2
Date: 13.06.2022
Workshop trainer: Georgii Zhulikov

Problem Statement

Main problem: making a GUI application using Model-View and OOP approaches with the help of QT framework. Working with user interface sketches and advanced blueprints.

Individual problem specification

You are given a Top_650_FIFA.cvs file, which includes the data of top 650 best players in FIFA 22. You will have to develop an application that will help users to assemble his own team from the presented list of football players using several ways of filtering of data.

Implementation details

URL: https://github.com/panedbay/dsba-itop2022-hw

The application was created using QT version 5 framework with the help of QT Creator (which includes a version of QT Designer). The program consists of one main window, from which other ones can be invoked. The application is cross platform, although in some tiny details, due to implementation of OSes, there are insignificant changes. The main window (application.h/cpp/ui) provides access to all functions, such as loading data from file, saving data to a file, manipulating the data (sorting, filtering) and gives access to helper windows (such as help window, add custom player window). All the information is encapsulated in private fields with public functions to access them. There are exception handlers, that prevent crashing, for example, handler that prevents loading csv with incorrect number of columns. The application is dynamically resizable, due to all widgets sharing a layout.

F1 or Help Menu opens a window with a logo and student number
Load File loads file from system
Save to File saves to file

Add Player opens dialog window for creating new player
Delete player deletes player that was selected previously
Add to My Team adds player to “your team”
My Team Only is a filter that shows only your team’s players

Search bar filters players by string appearing in their long name

Results and discussion

The results that were obtained the work are more than satisfactory: code works as intended, interface is good-looking, and the code works on different OS well.

Conclusion

There could be some improvements for the work, for example it was possible to configure load path based on the OS or implement native language translation.