Assessment in Computer Programming with C++

Movie Browser

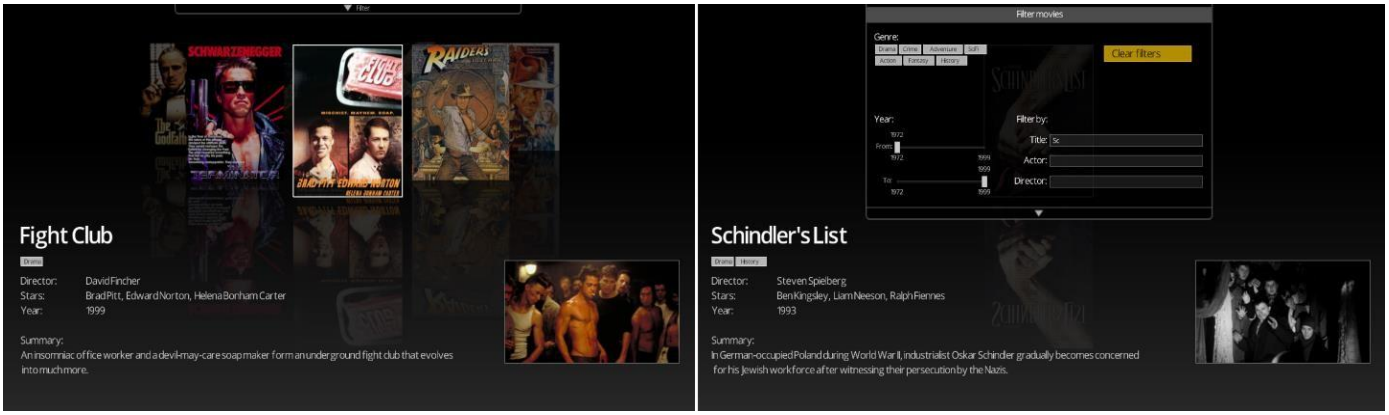

Image 1. *Example of implementing the movie browser with the SGG library.*

Description

The goal of the assignment is to create the graphical interface for an online movie browser, based on the Simple Graphics Library (SGG) built for the course. Each movie in your movie library should have at least the following characteristics, based on which you should be able to filter the movie search results through the interface:

- Movie name
- Release date
- Director
- Lead Actors
- Film genre

Example: “Schindler's List", 1993, "Steven Spielberg", "Liam Neeson, Ralph Fiennes, Ben Kingsley", "drama, history".

You can obviously incorporate other elements for each film included in the film library, such as, for example, a short description of the script. You can retrieve all of the above elements for movies from the movie database IMDB (https://www.imdb.com/).

You are free to design and create the graphical interface in whatever way you think is most convenient, possibly inspired by existing services (e.g. Disney+, Netflix, Ertflix).

Implementation

When implementing your application, you are required to combine knowledge gained from the lectures and to think well about your code architecture in order to achieve a) good code reuse, b) uniform and polymorphic way of calling methods, d) efficient use of ready-made STL structures, c) speed.

For your implementation, you need to create a hierarchy of building block classes for your GUI and use polymorphic method calling to operate them. The minimum constraints for your implementation are as follows:

- Mandatory and exclusive use of the SGG library as the only external library. It is not allowed to use libraries that already implement a graphical environment, such as Qt etc.
- Implement the graphical elements (buttons, sliders, radio buttons, checkboxes, etc.) with at least two levels of inheritance, where the ancestor class should be the abstract Widget class. Widget will represent a generic element of the GUI that specifies the common methods (with polymorphic call) and data that all graphical elements are to have, regardless of their specific function (e.g. position, dimensions, draw methods, update).
- Search for movies via the graphical interface at least according to the following items (see grading of work below):
    - Dateof Production
    - Film genre
- The source code you provide for your implementations should be properly commented (English is recommended). For functions and methods you should give a brief description of what they do, what the output is and what the required input is. Obviously, it is a good idea for the readability of your code to have comments within the implementation of the methods and functions themselves, at points where you are performing some complex procedure. Brief comments are also given for the fields of your classes.

Desirable features. The following will be positively assessed:

- Free string-based search for features that have free text (title, director, actors, etc.). See text fields in Image 1 - right.
- Proper code design and structuring, thorough method declaration (e.g., proper use of references and const definitions or methods).