Quiz Game Description

May 14, 2025

1 Introduction

1.1 How to compile project

Ensure you have the following installed:

- g++

- make

To compile the project, copy the content of one from the folders below into a `Makefile`

- `compiled/makefiles/makefileWindows.makefile` - for windows

- `compiled/makefiles/makefileLinux.makefile` - for Linux

In root of the project in terminal run: *make*.
Or another way to compile the project is to move script `compiled/makefiles/windowsCompile.bat`.

2 Game Description

base idea: The quiz game will be about testing the user's knowledge in various categories. The game will present single-choice questions. Questions will be presented one at a time.

3 Implementation

There will bew used MVC pattern. Header files will be used to separate classes and will be placed in the `include` folder with the same name as the class and in the folder with same name. Different categories of questions will be in different files.

3.1 Model

There will be all logic of the game. This will include:

- Question class - this will be a base class for all questions

- Game class - this will be the main class of the game

- Player class - this will be a class for player

The question class will be a base class for all questions. In "Ideas for Future Development" there are some ideas connected with different types of questions - MVC will make it easier to add after implementing the base game.

3.2 View

Currently the view can be implemented in console. In future it can be in GUI. Splitting to different modules makes it easy to rebuild view without changing other components. This part will include classes:

- Main menu class - class to display main menu

- Answering question view class - to display question and give posiibility to answer

- ViewManager - to manage what should be displayed

3.3 Controller

The controller will be responsible for communication between the model and the view. It will also handle the user input and update the model accordingly.
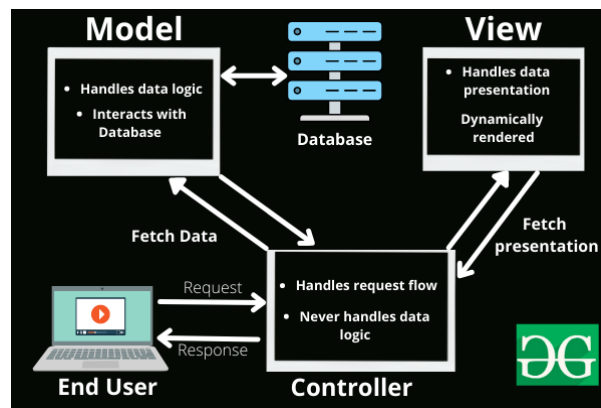

Figure 1: The image represents the way like MVC should work.

4 Technologies Used

- C++ - with STL libraries

- make - for building the project

- g++ - for compiling the project

- git - for version control

5 Ideas for Future Development

- Add more question categories. - obviously, currently there are none

- Implement a scoring system. - maybe

- Create a user-friendly interface. - now is only needed simple or console

- Add a timer for each question. - maybe

- few categories of questions - knowledge or knowledge about another person

- Add multiplayer functionality. - category like in cozy app and category where like opponents

- Adding own questions - I think needed in the future