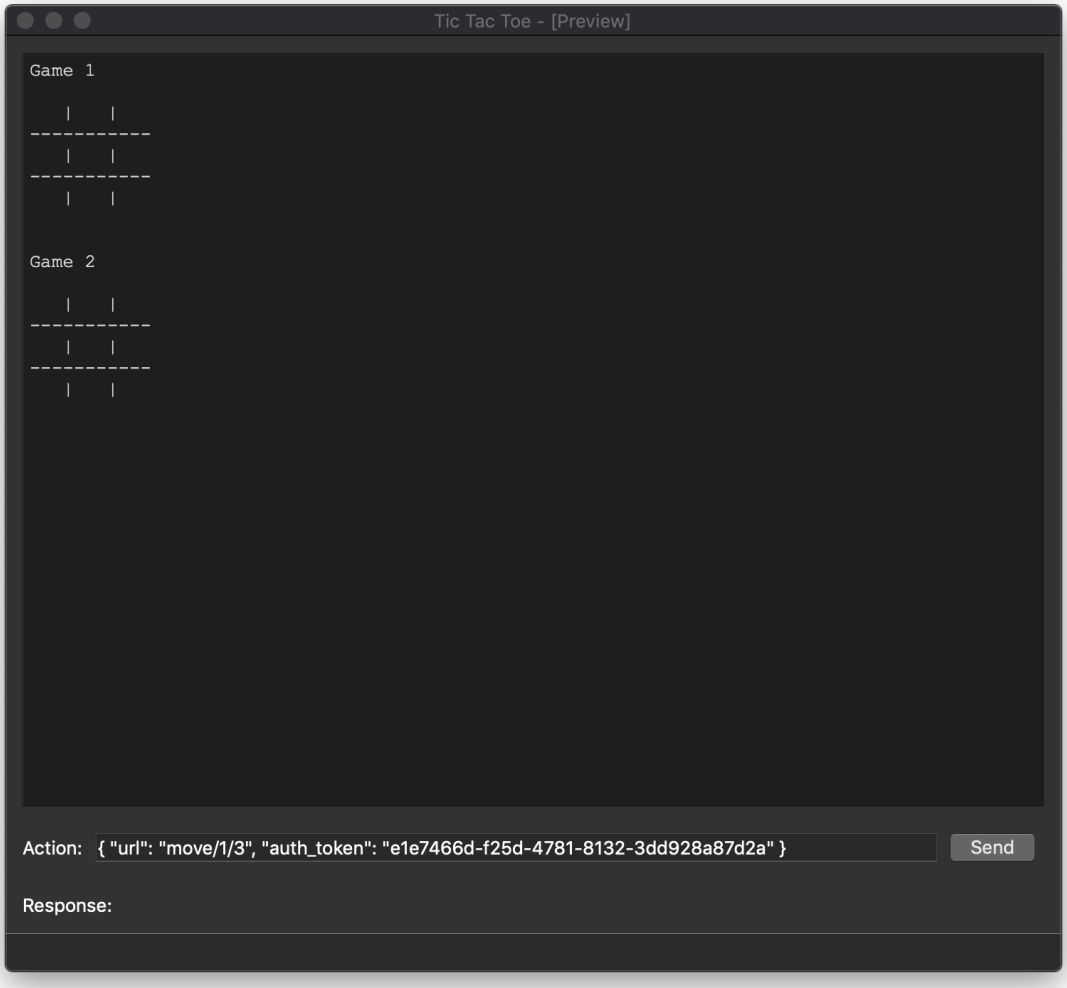
Tic Tac Toe task

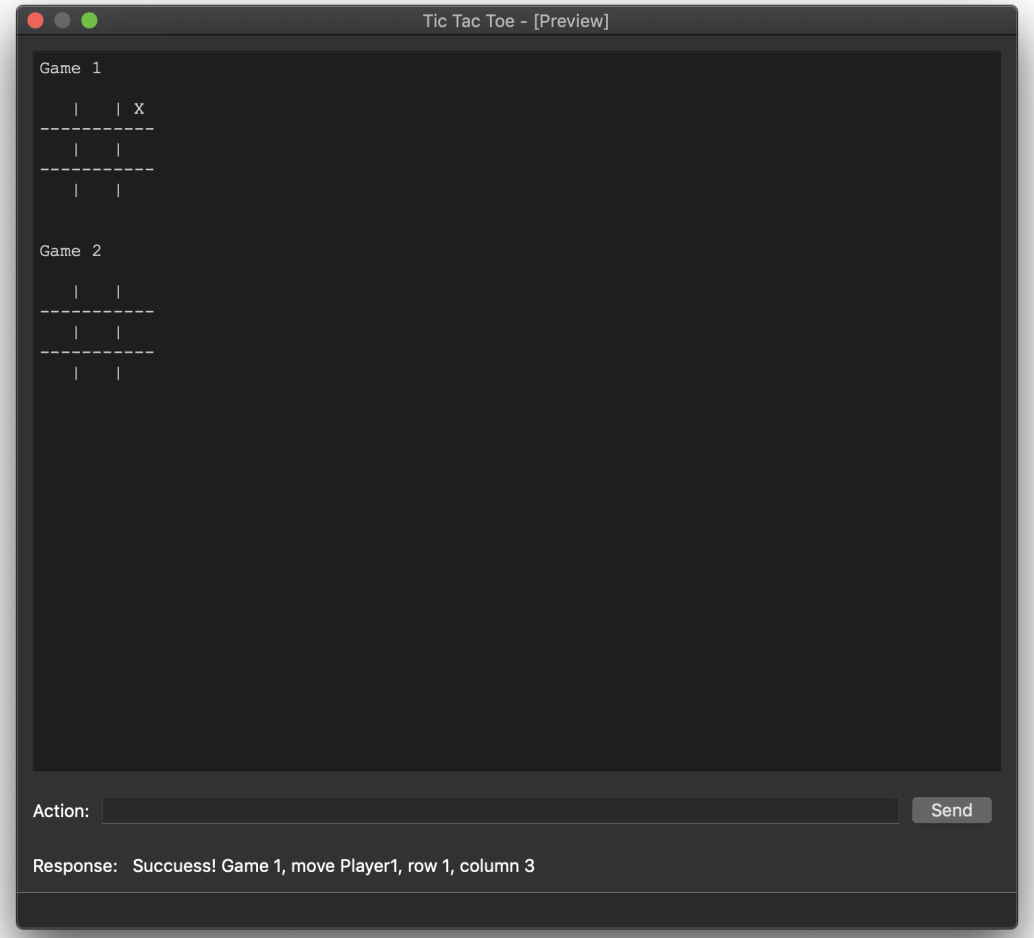
Your task is to implement the backend and a representation for a tic tac toe game in C++. It should be possible to use the backend with different frontends. Thus there should be an access layer that parses input and generates output in JSON format. There should also be a simple UI to interact with the backend/access layer written in Qt.

The rationale for asking you to do this task is to have a base to facilitate a technical discussion between engineers. When asked about particular aspects or details you should be able to argue why you chose to do things a certain way, what alternatives/options you considered and explain how things work in practice. See this as an opportunity to prepare a stage for presenting your skills the way you would like them to be seen. Once you have submitted your solution, we will review and schedule the technical discussion.

For example, the UI could look like this:



After hitting Send it would change to this:



So one player with a specific authentication token makes a move to row 1 and column 3. You can also choose other authentication methods and you can also design the call syntax in another way. It is only necessary to fulfill the requirements mentioned below.

Product requirements:

- It should be possible for multiple pairs of players to play games at the same time
- The application should check whether a move is actually possible and report a failure if it is not
- The won games should be stored per user and there should be a way to display a high score list
- You should be prepared to present and explain your code for other developers
- You should be able to give a live demo in some way, please prepare JSON representations beforehand

Technical requirements:

- Please implement this project in C++ with Qt
- In the end, there should be a working application where it is possible to play multiple games at a time
- Some basic authentication mechanism should work and actions should be checked against it (i.e. if it's Player1's turn, Player2 should not be able to make the move)
- Your code and architecture should be clean and follow guidelines that you would also like to see in a professional project (also taking in consideration how the project could be extended later but there is no need to over-engineer or bloat the code with design patterns if they have no use; find a reasonable trade-off yourself)

Please upload your project to a public Github/Gitlab/… repository where it's possible to view the code online.

If you feel that information is missing in this task, please make reasonable decisions yourself. If you cannot implement all features of this project as you would like to (e.g. because there is just not enough time), be prepared to explain extension points or to-dos clearly.