C++ Challenge

Today you will implement together with your teammates a client program which can play Connect-Four against a server provided by us. Whenever you have got any questions, just ask!

Exercise 1: First Steps (15 minutes)

Create a new project named “CppChallenge” in CLion. Copy the code template provided with this exercise sheet into your “CppChallenge.cpp”. Run the project. If no compiler error shows up, you are ready to start. In this step you can ignore any runtime errors!

Exercise 2: Network Code (1 hour)

The provided code template is not complete. Read through the code and understand what is happening. You can ignore the first 60 lines.

a) Implement the Constructor and Destructor in the *Connection* class.
b) Implement the *send(<int>), sendString(<string>)* function in the *Connection* class.
c) Write a function that can print a *GameInfo* object to the console window
d) Choose a unique teamname, not longer than 15 characters
e) Test your code by connecting to the server and sending your teamname. The server responds with sending a *GameInfo* object. You can receive it by the *Connection::receive* function. Print it to the console and let your program exit.

Exercise 3: Playing Connect-Four (20 minutes)

In this step you should implement parts of the protocol specified below. The goal is to play a single game against the server. After each game you played your program should exit.

a) After receiving the first *GameInfo* object from the server and printing it to the console, read an integer from the console Window.
b) Send the integer to the server. This is the column (from 0 to 6) where you want to insert your next stone.
c) Again, the server responds by sending a *GameInfo* object. Continue at a) until the *Result* variable in the *GameInfo* object does not hold the value *GAME_RESULT::CONTINUE* anymore.
d) Print the winner, and check for the reason
e) After the game is over let your program exit, ignore the part of sending a ‘+’ to restart in the protocol.

Remember: You only got a few (~10) seconds to answer to the server, otherwise the connection will timeout!

Exercise 4: AI (30 minutes)

Write an AI to play Connect-Four for you. Instead of reading an Integer from the console window, you should now write code playing Connect-Four by itself.

Hint: Start with a simple proof-of-work AI. Improve your AI after finishing exercise 5!

Exercise 5: Play multiple games (30 minutes)

Implement the full protocol as specified below. After each game you should restart immediately a new game. The server will send you a new *GameInfo* struct to let you play again, or otherwise close the connection!

Challenge!

When everybody finished coding, we will let you connect to the competitive server. This server will schedule matches between all the teams, such that every team will play against all the other teams. If your client wins, you gain 3 points, for a draw you gain 1 point and if you loose, you get 0 points. The winner team, which scored the most points. Have fun and improve your AI!