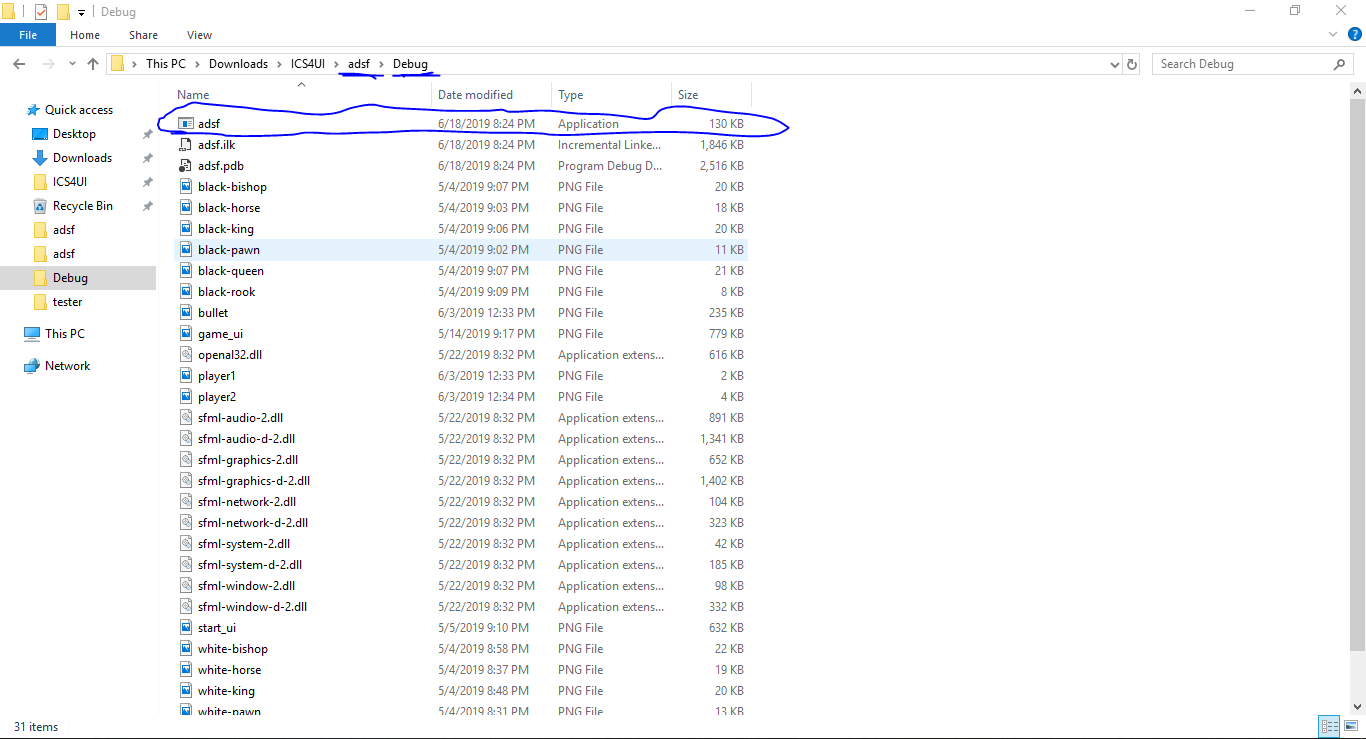
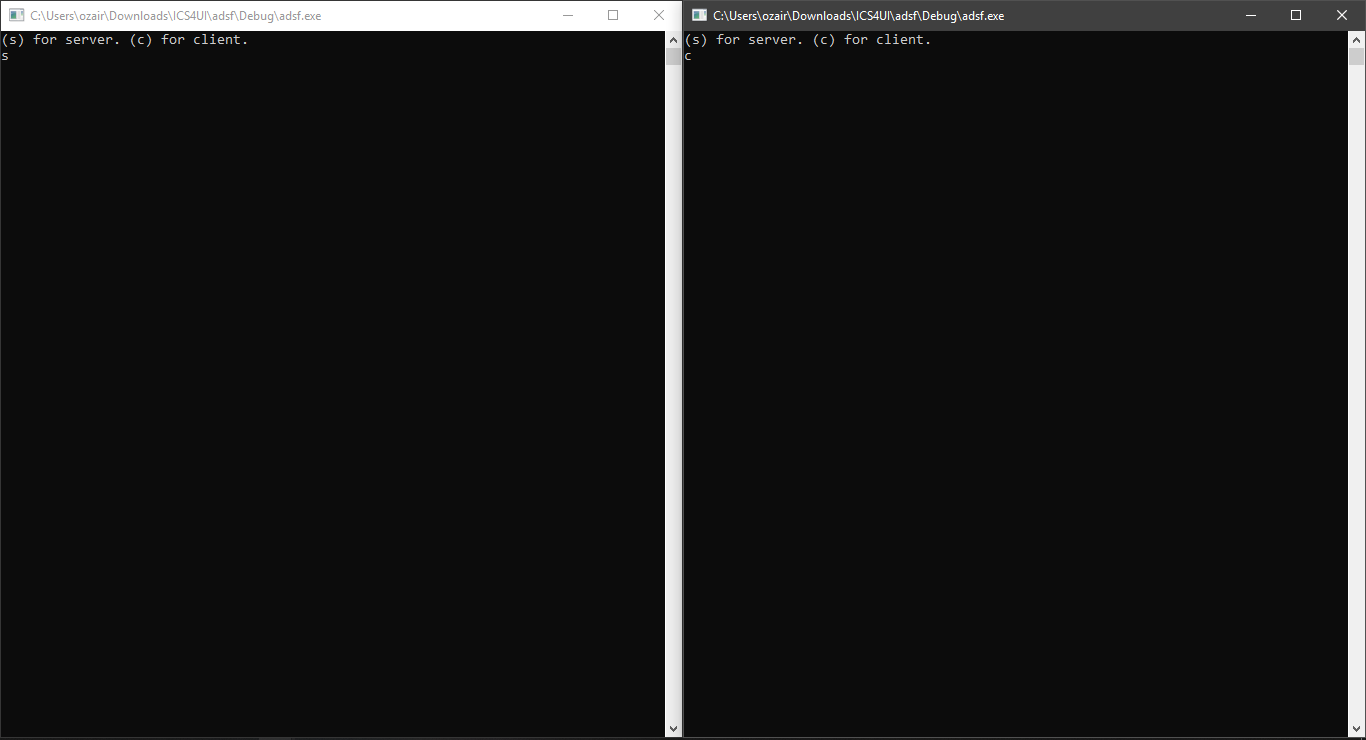
**Two Player Chess Game:**

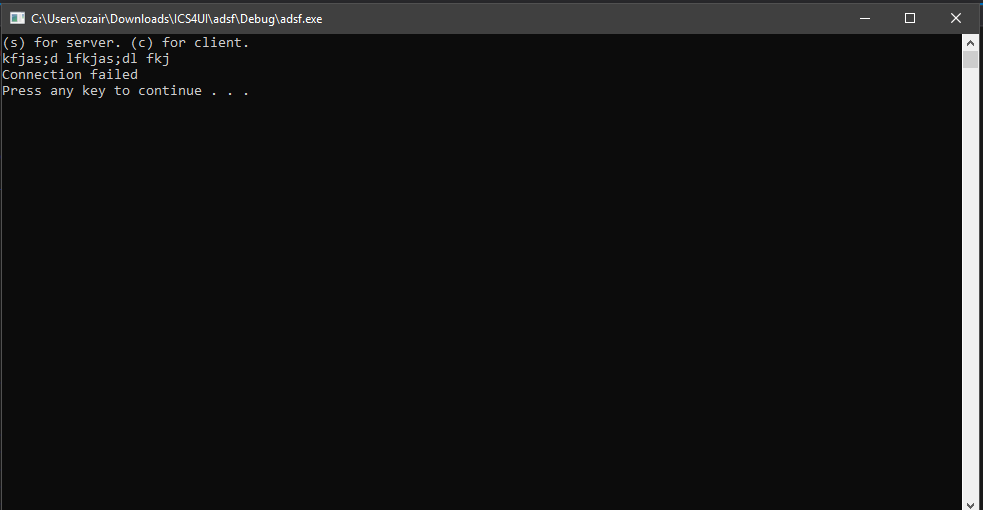
This chess game begins when the user launches the gui window from the **project\_folder/Debug/project\_name**. The game can also be launched from the IDE, however it will only open one gui window, which is not very helpful.



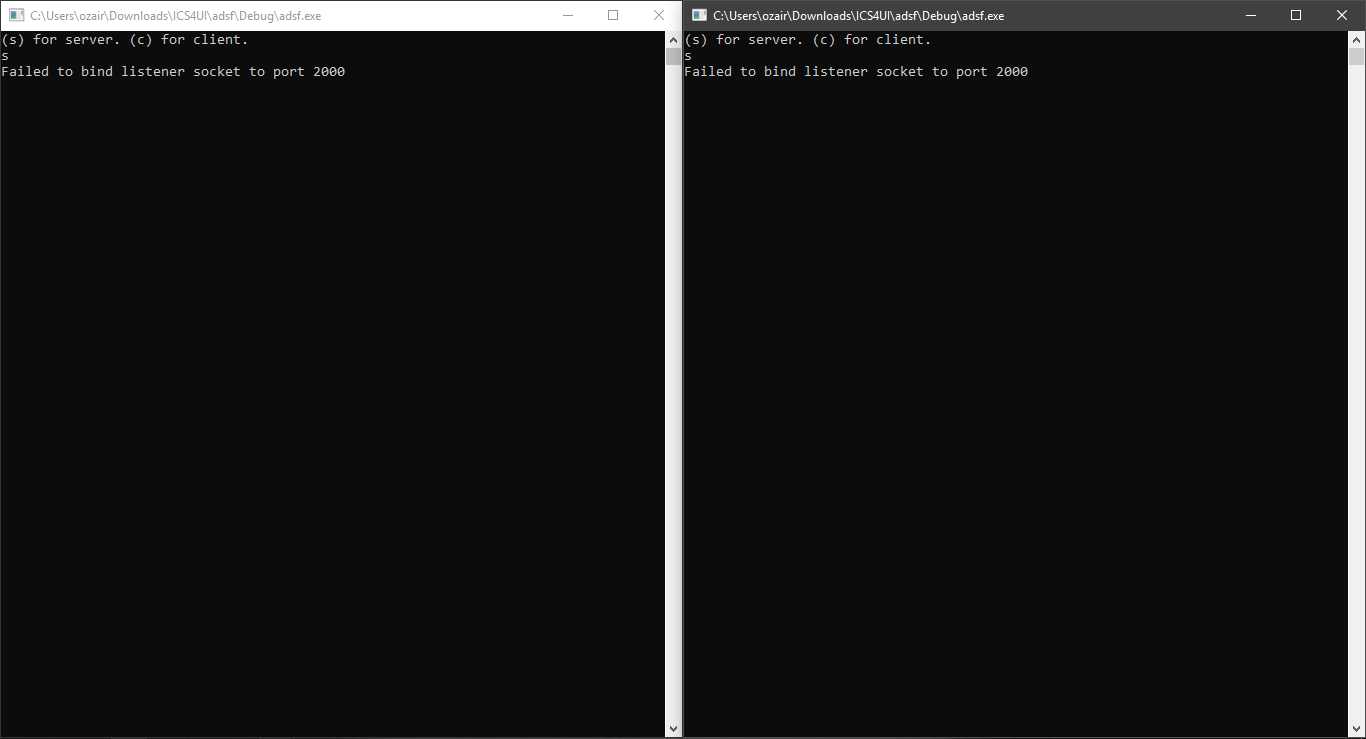
After clicking on the gui application, a console window will appear prompting ***“(s) for server. (c) for client”***. The gui window will appear as many times as it is opened however only the first two gui windows will bind to each other, and any additional windows will remain unconnected and may create additional errors, which depends on some factors. If one of the additional console windows is specified as a server, it will result in the total program crashing including the first two original windows. If the additional console windows are specified as clients, they will simply remain unconnected and frozen, and not affect the game that is running. After opening both console windows, the user must enter a **“s”** in one window, and a **“c”** in the other. It is important to note that **the connection must be made in order, with the server being initialized first and the client being initialized second**. Failure to do so will result in both console windows freezing and not proceeding with the rest of the program. Here is an example of the order, the connection should be made. (Although I have not pressed enter on either console window, it would be done first with the left side being the server, and then the right side being the client).



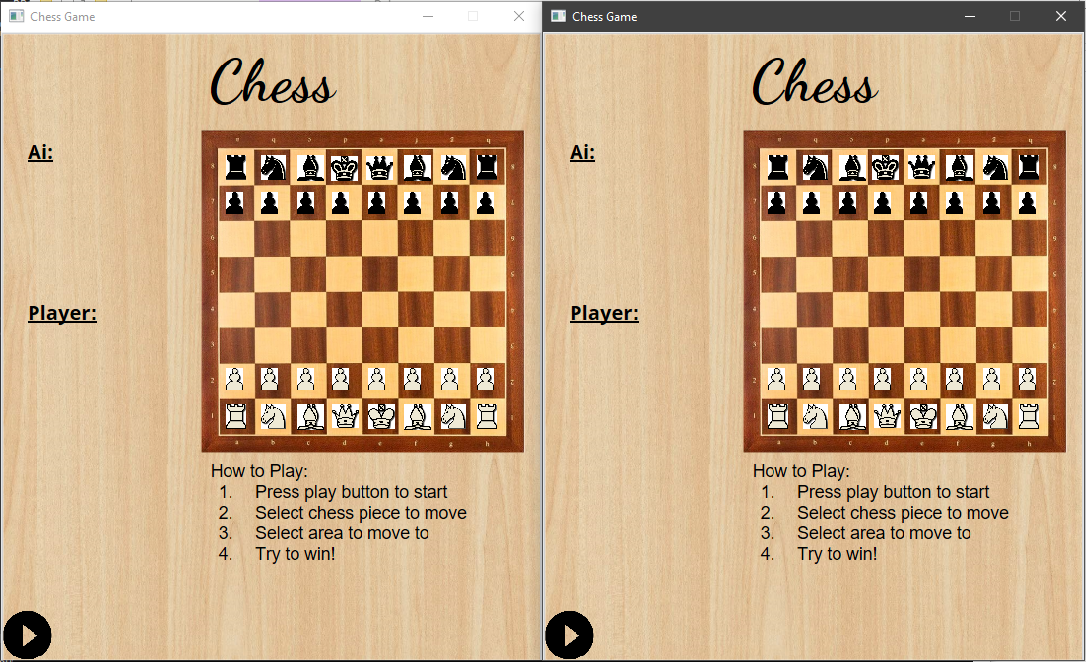
If the user(s) enters the wrong char into the console, when opening either a server or client, the error message **“Connection failed”** will be displayed and the console window will be closed. It should be noted that if game gui has failed connecting to the server, another gui window may be opened and the process may be repeated to attempt another server-client connection.



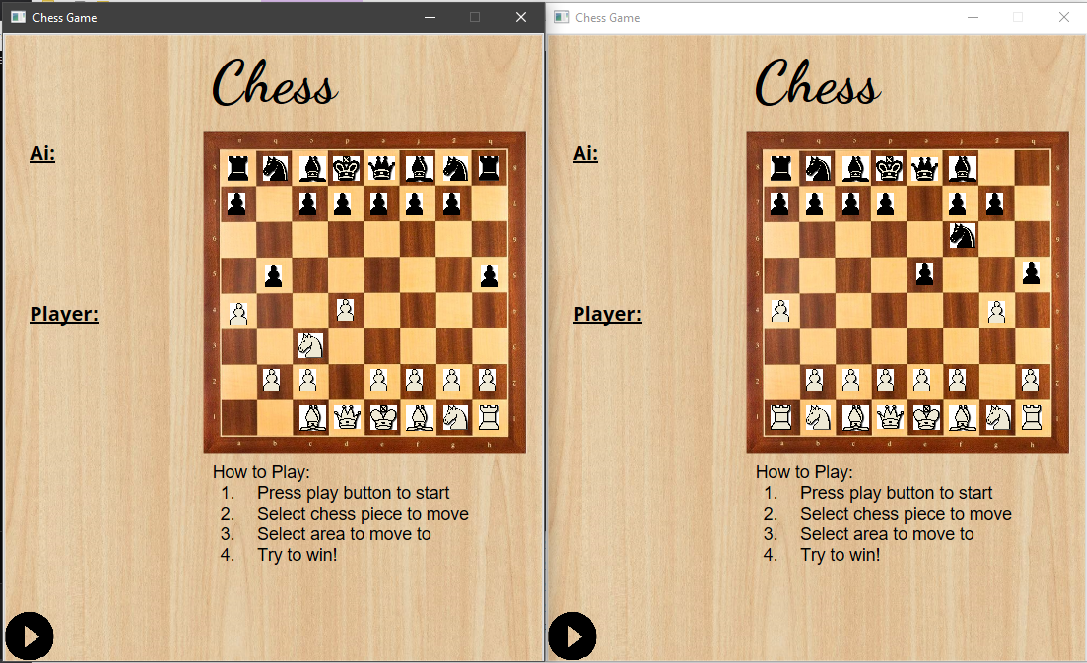
If for some reason the user(s) attempts to create more than one server, it will result in a **“Failed to bind listener socket to port 2000”**. This is because, the additional creations of server create additional listener sockets. Since every listener attempts to bind to the same communication port, it results in none of them connecting successfully. After this error all the console windows will remain frozen until closed by the user.



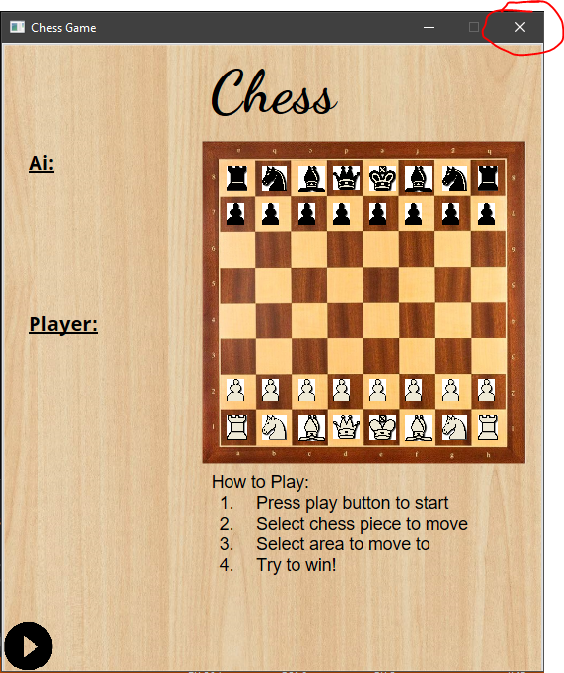
After both windows have successfully appeared, one or both users may begin to use the programs. In order to consecutively go back and forth and play the chess game, the user(s) must switch from one game window to another, after each turn. Here are both windows side by side. It is also important to note, that any time the program is running, the user(s), **must not close any of the console windows**. If done so the game will close out and any game progress will be lost. It is however fine to exit the IDE so as long the console window does not exit with it.



To play the game, the user must hold onto a piece they wish to move. The user must **continuously click down on the chess piece to change its position**. To place a piece in a chosen location, the user must simply let go of the left click where they wish to drop the piece. The rules of chess are incorporated here as well. If one of the users attempts to proceed with an illegal move, the moved piece will be sent back to its original location or spot before it was moved illegally. If an opposing piece is captured, it will be sent off the board and the opposing user will lose control of that chess piece. If either king is captured, the gui windows will close out to signify the end of the game. It should be noted that the **right mouse button posses no player control**, so all movement must be done with the left mouse button. While playing, it might be noticed that both gui windows are mirrored images. Whatever happens on one board is displayed on the other board as an opponent. Because of this, the user(s) should make sure to keep track of which gui they are playing with, as there are no methods to determine which gui is who’s.



If the user(s) wish to exit the game. They may at any time, press the **x** on the gui window which will allow them to exit. This will however only close one of gui windows, so therefore this must be done twice.



However, if one of the windows was initially started from the IDE, pressing the red square (aka stop debugging) will terminate any program windows that are open (does not matter from where they were openned from), without the need of accessing the **x** button.

