“Wax on, wax off” - mr. Miyagi.

The development of saving and loading systems shares much in common with that of networked systems.

The development of both systems will require you to be capable of transforming the data that you have stored away in class instances with that of a data stream, a sequence. This process is commonly referred to as serialization, with deserialization representing the transformation of data from a sequence stream into that of variables or class instances.

So, what does this mean for you, a scholar of net code development?

It means that we can strengthen your muscles, your underlying understandings, your comfort level and confidence, by working through the development of a less framework tool complex system, a saving and loading system.

In this lab, you are being challenged to develop the saving and loading system for a hypothetical RPG.

1. Start by cloning or downloading a copy of the lab project, opening it, and ensuring it works (there should be no errors in your output when you run it). This may require you to update the project to your more newer version of Unity, this process should be fine. It is likely that Unity will not open the assignment scene for you, if you see nothing on screen when you hit play, you need to open the scene in the scene folder.

2. Familiarize yourself with the UI. You can roll a new party, view party stats and hit a save and load button, both of which do nothing.

3. Open the Assignment.cs file and you will find two functions, SavePartyButtonPressed and LoadPartyButtonPressed. Both functions are called by the internal lab system when the respective button is hit.

4. Inside of each function you will find some sample code. You are being provided with:
    a. The ability to loop through the list of the party characters being displayed on screen, GameContent.partyCharacters.
    b. You may access the stats of each character via the iterator named pc.
    c. The sample code also provides you with the ability to clear the list of party members and repopulate it with new instances of the PartyCharacter class.
    d. The sample code provides you with the ability to refresh the UI so that it will display contents of GameContent.partyCharacters.

5. There is one last thing. You will need to make use of the StreamReader and StreamWriter classes. These classes can be accessed via the inclusion of System.IO in your using statements. You can find sample code on using the reader/writer here:

a. https://docs.microsoft.com/en-us/dotnet/api/system.io.streamwriter
b. https://docs.microsoft.com/en-us/dotnet/api/system.io.streamreader

Journey well.