**Saving and Loading Data**

**Game Saving:**

In order to save the player’s game, we will build a blueprint with a variable for each piece of data we wish to save. This will be done by creating a ‘savegame’ class, setting its instance then setting all the variables we wish to save and finally calling the ‘asynch’ save game to slot function. We will have an input allowing the player to select a save game slot and will give each slot a unique ID.

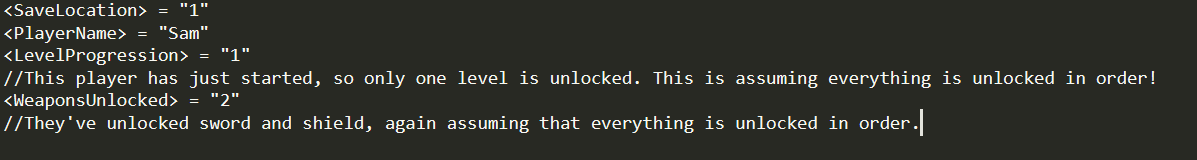


Figure 1 - Example save data for a fresh profile named "Sam".

Similar save processes will store other data, such as level statistics:

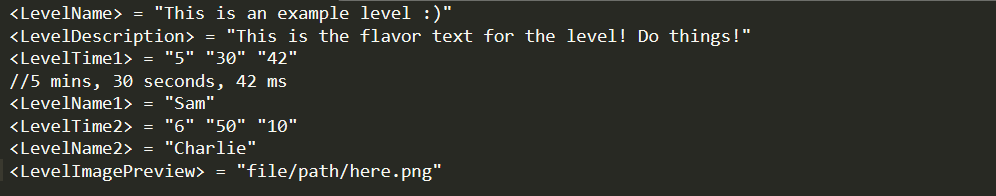


Figure 2 - An example of a saved level high score table.

And will also be used to store player settings:

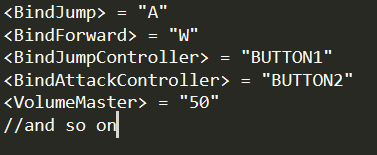


Figure 3 - An example of saved player settings

Note that level scores and player profiles are stored separately- this allows for the creation of development tools to unlock all levels, as well as storing a player's high scores even if their profile is deleted.

**Loading Data:**

In Unreal, the blueprint for saving data can also be used to load it. This allows us to simply call the built-in load function, with the values within the file mapped to the correct values in the modified class and obtain the information within a file. This can then be loaded to show high scores, or to obtain a player's progress throughout the game.