Saving and loading data

A step-by-step guide by @Sambamuel

Whats happening?

There is a function in unity which converts a c# class instance to a JSON file and there is another one which converts a JSON file back to a class instance.

So essentially we have a class called DataScript. Our instance of this class is called ‘data’, where we store all our stored-variables.

Then when we “save” we copy our game variables to the stored variables in the ‘data’ instance and convert the instance to a JSON file.

Then, when we “load” we copy the stored variables to our ‘data’ instance and copy the variables from the ‘data’ instance into our game variables.

```
DataScript data;
```

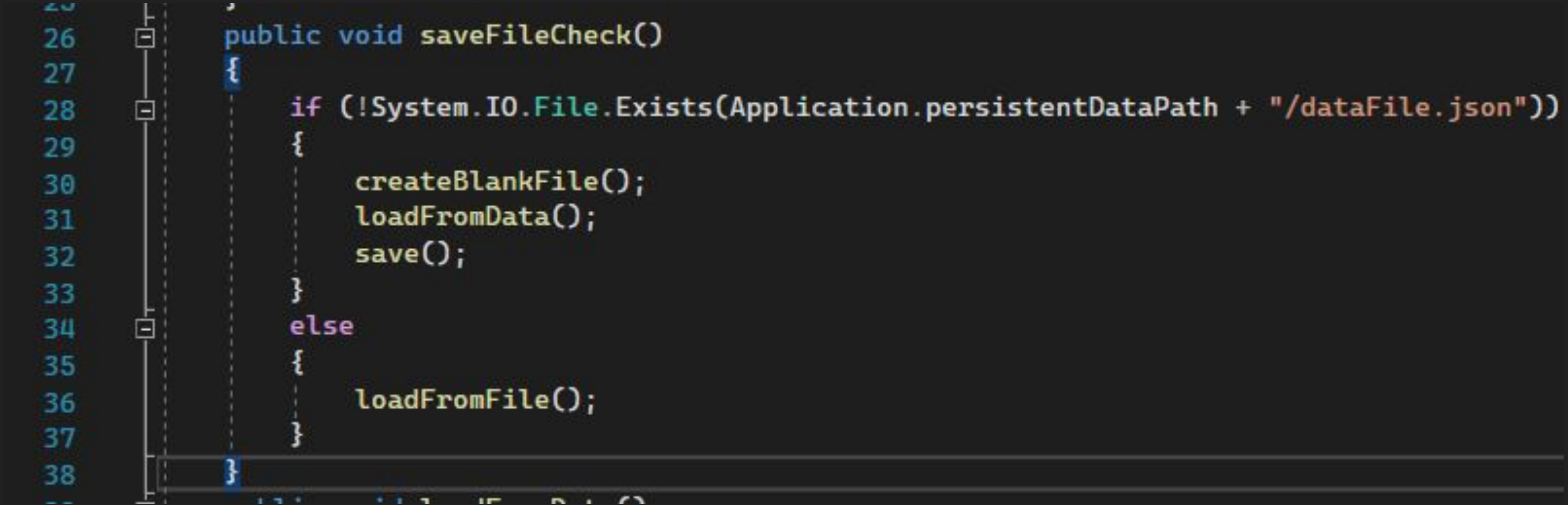
Awake()

When starting the game. You will need to check if the save file exists or not.

If the save file exists, you will need to Load the data from the file to the game variables.

If the save file does not exist, this will be the first time the game is run, you will need to instantiate a data object, then set the game variables to equal the data object’s default variables, then save the data object to a JSON file.

Awake() - the code



```
}
public void saveFileCheck()
{
    if (!System.IO.File.Exists(Application.persistentDataPath + "/dataFile.json"))
    {
        createBlankFile();
        loadFromData();
        save();
    }
    else
    {
        loadFromFile();
    }
}
public void loadFromData()
```

Create Blank file

```
data = new DataScript();

save();
```

This creates a new data object which will have the default data. Then save() will create a new JSON file if one doesnt exist and assign the default data to the file

save()

Current data object = game variables

Save file variables = current data object

save() code

For the first part it will look something like

```
data.yourFloat = GameManager.yourFloat;
```

Then, this is responsible for casting the data object to a JSON object

```
    string jsonObject = JsonUtility.ToJson(data, true);

    System.IO.File.WriteAllText(Application.persistentDataPath + "/dataFile.json", jsonObject);
```

Where i’ve written “dataFile.json” that can be any name. If the file exists, it will rewrite it. If the file doesn’t exist it will create one.

Load From Data

Game variables = current data object variables

```
GameManager.yourFloat = data.yourFloat;
```

Load From File

Current data object variables = save file data variables

```
string jsonObject = System.IO.File.ReadAllText(Application.persistentDataPath + "/dataFile.json");

data = JsonUtility.FromJson<DataScript>(jsonObject);

LoadFromData();
```

Example code

https://www.slimeyhatch.com/useful-code/saving-and-loading