Binary

Description

This binary library will help convert any c# object into binary format and save them in a file as long as the object is serializable. The functions will need you to provide a path, filename, and serializable objects to read and write to a file in binary.

Properties

WriteToFile	Write onto a file in binary
ReadFromFile	Read from a file that contains information in binary

Binary.WriteToFile

public static void WriteToFile(object obj, string path, string fileName);

Parameters

obj	Serializable C# object
path	The path where the file should be saved Note: The path should end with a '/' (for example: 'data/enemies/stats/)
fileName	Name of the file to which data will be saved to Note: The file extension must be provided (for example: 'theFile.data')

Description

Writes to provided data in binary to the specified file at a specified path.

This is a simple example of how to use this function, in this example where we write to the file in the start function (coded in Unity and using Unity's Library):

```
using UnityEngine;

public class ExampleClass: MonoBehavior

{

        private string playStatsDiractory = "/ExportToFbx/data/player/";

        private string playerStatsFileName = "playerStats.data";

        private Vector2 playerStats;

        private void Start() {

                playerStats = new Vector2(100, 100);

                savePath = Application.persistentDataPath + pStatsDiractory;

                Binary.WriteToFile(playerStats, savePath, pStatesFileName);

        }

}
```

Binary.ReadFromFile

public static object ReadFromFile(string path, string fileName);

Parameters

path	The path where the file should be saved Note: The path should end with a '/' (for example: 'data/enemies/stats/)
fileName	Name of the file to which data will be saved to Note: The file extension must be provided (for example: 'theFile.data')

Description

Reads the file from the specified path which contains data in binary and converts that data into an object and returns it.

This is a simple example on how to use this function, in this example where we read from the file and store it in a variable in the start function (coded in Unity and using Unity's Library):

```
using UnityEngine;

public class ExampleClass: MonoBehavior

{

        private string playStatsDiractory = "/ExportToFbx/data/player/";

        private string playerStatsFileName = "playerStats.data";

        private Vector2 playerStats;

        private void Start() {

                loadPath = Application.persistentDataPath + pStatsDiractory;

                playerStats = (Vector2) Binary.WriteToFile(playerStats, loadPath,
                pStatesFileName);

        }

}
```