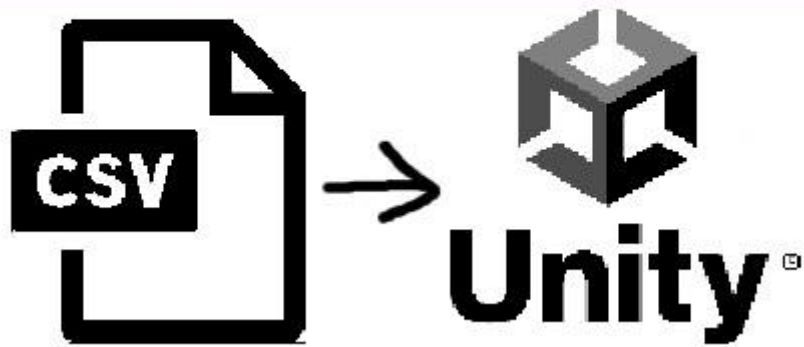


CSVUtility

CSVUtility



このアセットに含まれるもの	3
CSVUtility で出来ること	4
扱うことが出来るデータの型	4
CSVファイルの読み込み	5
データの扱いを行う2つのクラスを作成	5
CSVファイルの準備	6
データを読み込むサンプルプログラム	6
デモーションについて	7

このアセットに含まれるもの

このアセットにはCSVファイルを読み込んで、ゲーム内のデータとして利用することが出来るためのサポートプログラムが含まれています。また、それらの動作を理解するためのサンプルシーンを用意しています。

CSVUtilityで出来ること

CSVUtilityを使うとcsvファイルをゲーム内のデータとして利用することが出来るようになります。
主な機能としては以下

- CSVファイルの読み込み
- CSVファイルの書き出し保存

実際の利用方法に関しては次の項目を参照してください。

扱うことが出来るデータの型

データの型は以下

- int
- float
- bool
- string

CSVファイルの読み込み

CSVファイルの読み込みを行うためには次の手順が必要になります。

1. データの型になるパラメータクラスの作成
2. パラメータクラスを利用したモデルクラスの作成
3. 利用するデータが記載されたCSVファイル

データの扱いを行う2つのクラスを作成

例えば次のデータを持ったクラスを扱う場合

型	変数名
int	test_int
float	test_float

準備に必要なクラスファイルは以下のようになります。

```
using anogamelib;

public class SampleModelParam : CsvModelParam
{
    ... public int test_int;
    ... public float test_float;

    ... public override string ToString()
    ... {
    ...     ... return $"test_int={test_int} test_float={test_float}";
    ... }
}

public class SampleModel : CsvModel<SampleModelParam>
{
}
```

ToStringのオーバーライドはデバッグログ表示に使うだけなので、実装は不要です。

using anogamelib;の追加と各クラスの継承しているクラスに注意してください。

上記例だと、SampleModelParamクラスが型に相当します。CSVファイルの読み込みなどは、SampleModelクラスを経由して取り扱いを行います。

CSVファイルの準備

上記例のクラスデータに対応したcsvファイルを参考に説明します。
お手持ちの表計算ツールや、テキストエディタにて、次のようなデータを準備してください。

	A	B	C
1	test_int	test_float	
2	123	12.3	
3	456	45.6	
4			
-			

1行目はヘッダー情報として、各パラメータ名と同じテキストを入力してください。2行目以降は各レコードに対応したデータを追加していきます。今回はデータは2つ分ですが、思う存分追加してください。

また、拡張子をcsvに変更してください。

データを読み込むサンプルプログラム

上項目で作成したcsvファイルをゲーム内で読み込んでみたいと思います。保存したデータをUnity内にimportして、次のスクリプトを用意します。

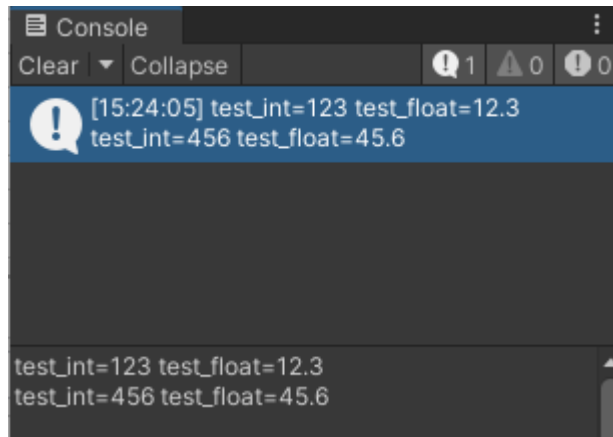
```
using UnityEngine;

public class DemoScript : MonoBehaviour
{
    ... [SerializeField] private TextAsset localCsvFile;

    ... private void Start()
    ... {
    ...     ... SampleModel sampleModel = new SampleModel();
    ...     ... sampleModel.Load(localCsvFile);
    ...     ... sampleModel.CheckDebugLog();
    ... }
}
```

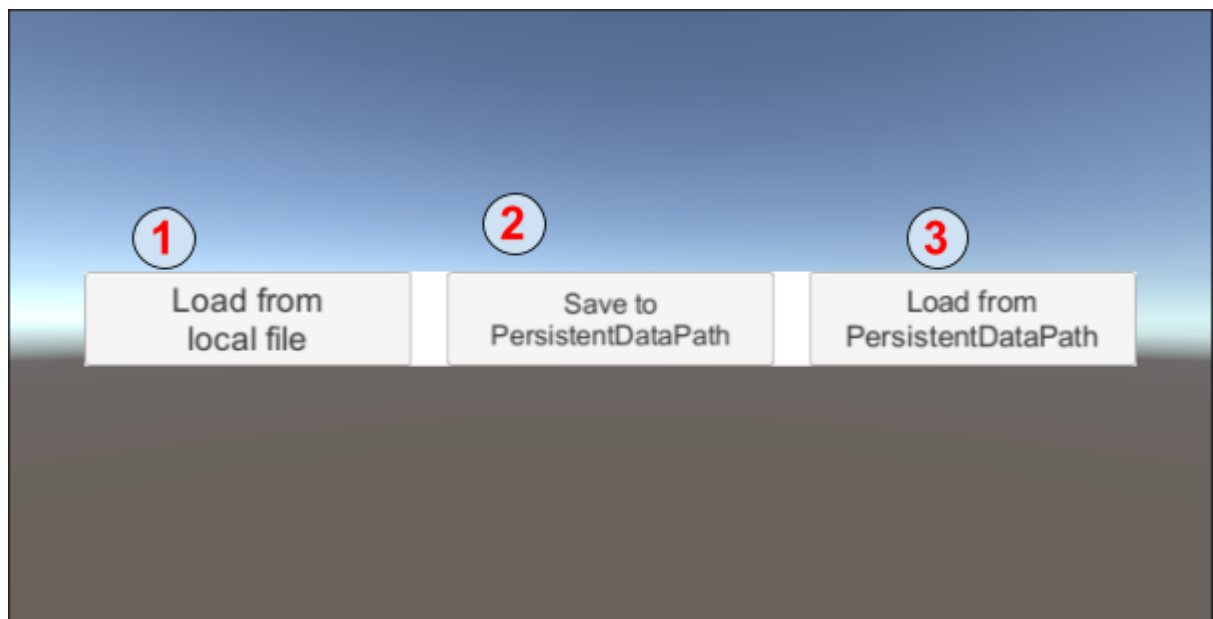
スクリプトが準備出来たらシーン内に空のゲームオブジェクトをアタッチし、インスペクターのLocalCsvFileに保存しているcsvファイルをセットしてください。

うまく出来ていると、次のようなデバッグログが表示されます。



デモシーンについて

でもシーンでは、次の操作について確認することができます。



1. ローカルのcsvファイルをロードするデモ。
 2. ロード済みのモデルをPersistentDataPathに保存します。
 3. PersistentDataPathに保存されたcsvファイルをロードします。
- 2,3をうまく使うことで、ゲームデータのセーブ・ロードを行うことが可能です。
保存先などはデバッグログに表示されていますので、ご確認ください。