InGAME
Innovation for Games
and Media Enterprise

DYONISIAN
DIVIJ SOOD

CSV Reader Tool

One-click code generation to read your CSVs

One-click CSV Reader Tool

Developed by Divij Sood, R&D Fellow at InGAME

InGAME is the UK's dedicated centre for innovation in games and media technology. It is funded by the AHRC as part of the UK Industrial Strategy.

Since inception in 2018, InGAME has:

- Funded 173 Collaborative R&D projects
- Leveraged £12m in co-investment
- Upskilled 1,680 professionals from industry and academia.

https://www.innovationforgames.com/

Contents

Introduction

The One-click CSV Reader tool gives you an easy way to work with CSVs in your project. CSVs allow you to separate data from code and allow easy content creation and editing workflows for designers and writers. CSVs and this tool are particularly useful for adding dialog and other authored content to your game.

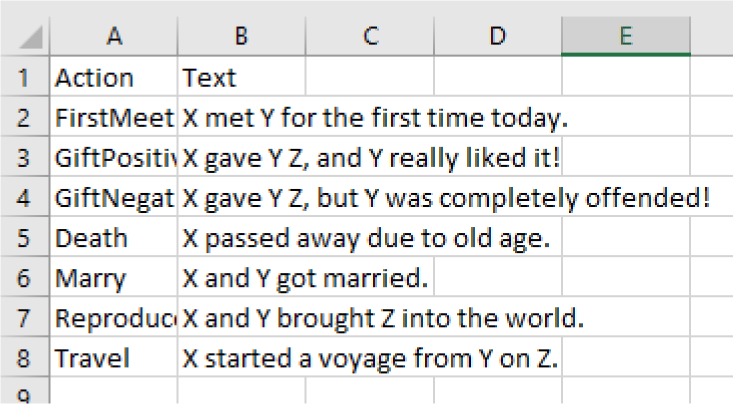
This tool generates code that will read your CSVs, store them in appropriately-named variables, and give you enums and helper functions to more easily access that data in any part of your code.

This tool was developed in collaboration with a writer for a procedural narrative generation project.

Uses Unity's UI Toolkit
Requires Unity 2021+
Works on PC and Mac
Generated code has been tested on PC and Android builds, but should work for other common platforms as well, as long as those platforms support the Resources.Load function.

CSV Files

A sample file called "DoubleFieldTest.csv" is included with this asset in the Resources/CSV folder to help demonstrate the ideal use case for this tool.



	A	B	C	D	E
1	Action	Text			
2	FirstMeet	X met Y for the first time today.			
3	GiftPositi	X gave Y Z, and Y really liked it!			
4	GiftNegat	X gave Y Z, but Y was completely offended!			
5	Death	X passed away due to old age.			
6	Marry	X and Y got married.			
7	Reproduc	X and Y brought Z into the world.			
8	Travel	X started a voyage from Y on Z.			
9					

Tutorial

How to use:

TO TEST: Move DoubleFieldTest.CSV from the SampleCSVFiles folder to Assets/Resources/CSV first

1. Open the editor window by clicking on Tools > CSVReader in the Unity Menu bar

2. Preview CSV section - Use this by specifying a path to a CSV file to preview a CSV in the editor window

3. Read All CSVs section - This is the main tool that reads all CSVs in a folder and generates code that will, at runtime, read from those CSVs and store data in varaibles for you to use in your code.

4. Input folder path: Enter the name of the folder that holds your CSV files (Ideally, this folder should be in the Resources folder.

For example, store all your CSVs in a folder named "CSV" in Assets/Resources. Then, the input for this field should just be "CSV" without quotes)

5. Streaming Assets? Checkbox: Check to load CSVs from Unity's Streaming Assets folder. Leave unchecked to instead use Resources.Load.

Leaving this unchecked and storing your CSVs in a CSV folder under Assets/Resources is the recommended way to use this tool as this method has higher compatibility with most platforms like PC, Android, and iOS.

6. Generate enums and helpers?: Check to generate enums and helper functions that you can easily access anywhere in your code to easily access the data that will be read from the CSVs.

7. Ignore first row?: Check to ignore the first rows of all CSVs, useful if the first row in each CSV is a row header

8. Output script name: Enter the name you'd like the generated code file that reads and stores your CSVs to have. Eg. TextManager

9. Read all CSVs and generate script with data: Click here to find CSVs and generate code that will read them at runtime. You can find the generated script in the "Scripts" folder. You can immediately open this to double-check the code and see how you can access the data from CSV files in your own code.

10. In case of an error, it's best to try to understand the error and then delete the generated script and start over again.

Follow the instructions above to generate code that can read and store data from this file at run-time.

Sample code to test generated code:

```
Debug.Log(TextManager.GetSampleCSVFile(TextManager.DoubleFieldTest.Attack, 1));
```

Troubleshooting

First, try to replicate the sample use case as closely as possible. Follow the instructions above, use the Resources option and make sure your file names, folder paths, and options are correct.

This tool has been tested for compatibility with CSV files created on both Windows and Mac. The generated code has been tested and works on both PC and Android platforms, but should work for other platforms as well.

Contact

If you need help, find me on Discord: Dyonisian_Divij#4947

Or write to me at dyonisiandesign@gmail.com