

Simple Twine Dialogue Importer for Unity

Transform your interactive stories into dynamic Unity projects with the Simple Twine Dialogue Importer for Unity! This lightweight and efficient tool allows you to seamlessly import Twee files into Unity, enabling rapid creation of branching dialogues and narrative-driven experiences. Perfect for game developers, writers, and designers looking to integrate text-based storytelling into their game.



Key Features:

- **Easy Integration:** Easily import Twee files directly into Unity, converting them into interactive dialogues with just a few clicks.

- **A Simple Project:** Designed with simplicity in mind, this parser requires minimal setup, allowing you to focus more on crafting compelling narratives and less on technical details. Don't get bogged down with documentation, tutorials, and scripts.
- **Example Scene:** Try out a simple example scene to get you started.
- **Example Twee:** An example file exported from Twine is included in the project to get you started.
- **Platform Agnostic:** Twee and image assets can be loaded onto mobile and web by loading from an external server.

Getting Started

Prerequisites

- **Unity:** Make sure you have Unity installed and import the package.
- **Twee Files:** Prepare your [Twee](#) files with passages, choices, and image references. You can export these from [Twine](#).
- **Images** (optional): Prepare any images you want to include in each passage.

Setup

1. Import the Project into Unity

Open Unity and import the **TweeParser** folder into your project. You can do this by dragging the folder into the Unity Editor or using the **Assets > Import Package > Custom Package** option.

2. Configure the Scene

- **Add Components:** Drag the **TextAdventure** script onto a GameObject in your scene.
- **Assign Prefabs:** Make sure to assign the **imagePrefab** and **choiceButtonPrefab** in the Unity Inspector.
- **Set URLs:** Enter the URLs or paths for your Twee file and image base path. These external files can be hosted on a webserver like a personal site, or AWS.

Scripts

There are two main scripts to this package including:

1. **TweeParser.cs:** This script parses the Twee file using regex.
2. **TextAdventure.cs:** This script loads the resources and displays the passage on the UI.

Using the Parser

Prepare Your Twee File

Make sure your Twee file follows the expected format. Here's a sample passage:

```
:: Listen to Riddle
```

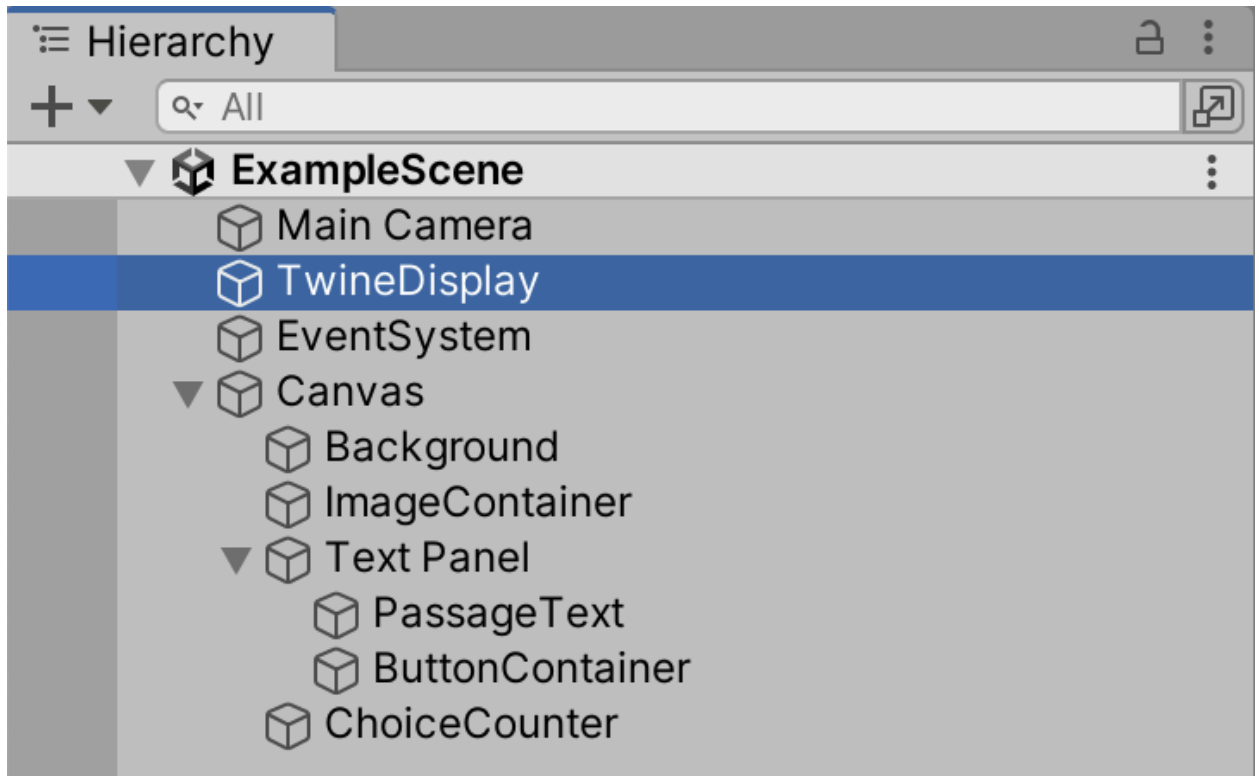
The cultist's eyes twinkle with mischief as they begin to recite in a sing-song voice:

"I speak without a mouth and hear without ears. I have no body, but I come alive with the wind. What am I?"

```
[[Image:cultists2.png]]  
[[Attempt to solve the riddle.|Solve Riddle]]  
[[Refuse to answer and walk away.|Walk Away]]
```

Import Your Own Twee Dialogue

1. In your Unity project, load the ExampleScene.
2. Make sure to load your Twee file and images onto a web server so the project can request them.
3. In the project's hierarchy, find the TwineDisplay gameobject.



4. In the TwineDisplay gameobject, you can choose to 'Load From Web' or use the Streaming Assets folder.

- Update the associated links or files for the Twine File URL and Image Base URL.

The screenshot shows the Unity Inspector window with the following components and settings:

- Inspector** (Header)
- TwineDisplay** (Component)
 - ☒ **TwineDisplay** ☐ **Static**
 - Tag: **Untagged** Layer: **Default**
- Transform** (Component)
 - Position: X **728.1655** Y **1066.187** Z **-3.746369**
 - Rotation: X **0** Y **0** Z **0**
 - Scale: X **1** Y **1** Z **1**
- Text Adventure (Script)** (Component)
 - Script: **TextAdventure**
 - UI Components**
 - Passage Text: **PassageText (Text Mesh Pro UGUI)**
 - Choice Button Prefab: **Button (Button)**
 - Choice Button Container: **ButtonContainer (Rect Transform)**
 - Image Container: **ImageContainer (Rect Transform)**
 - Image Prefab: **Image (Image)**
 - My Choice Counter UI: **ChoiceCounter (Text Mesh Pro UGUI)**
 - File Loading**
 - Load From Web: ☐
 - Load from Web**
 - Web File URL: **https://alexiabucket.s3.us-east-2.amazonaws.com/twee/ExampleTwee.twee**
 - Image Base URL: **https://alexiabucket.s3.us-east-2.amazonaws.com/twee/images**
 - Load from Local**
 - Local File Name: **ExampleTwee.twee**

- If you are using 'Load From Web' the Web File URL will be the exact path of your Twine file. The Image Base URL will be the directory your images live in.
- Press play to load your dialogue. This will load the Twine file, parse it, and display the "Start" passage.