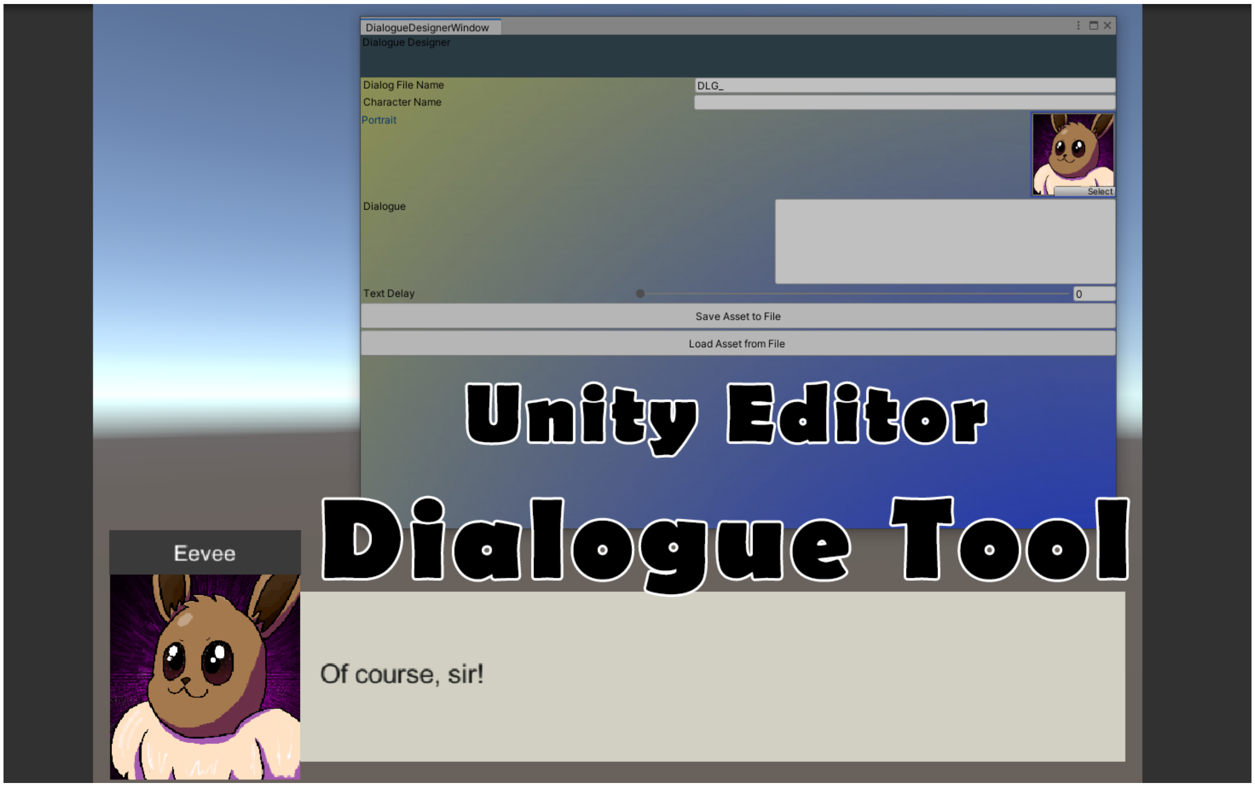
Instructional Doc for Dialogue Editor

by Cloyce Stetson



This package has been approved for Unity Version 2019.4.2f1

Installation:

Step 1: Download the .unitypackage file.

Step 2: In the Unity Editor, go to Assets > Import Package > Custom Package, and use the window to navigate to and select the unity package file.

Step 3: Select which items you want to import or click import to install them all. The package includes a premade UI.

Step 4: The package is now installed! The window editor script is in the Editor Folder and the Scripts folder has the runtime scripts that interact with the gui.

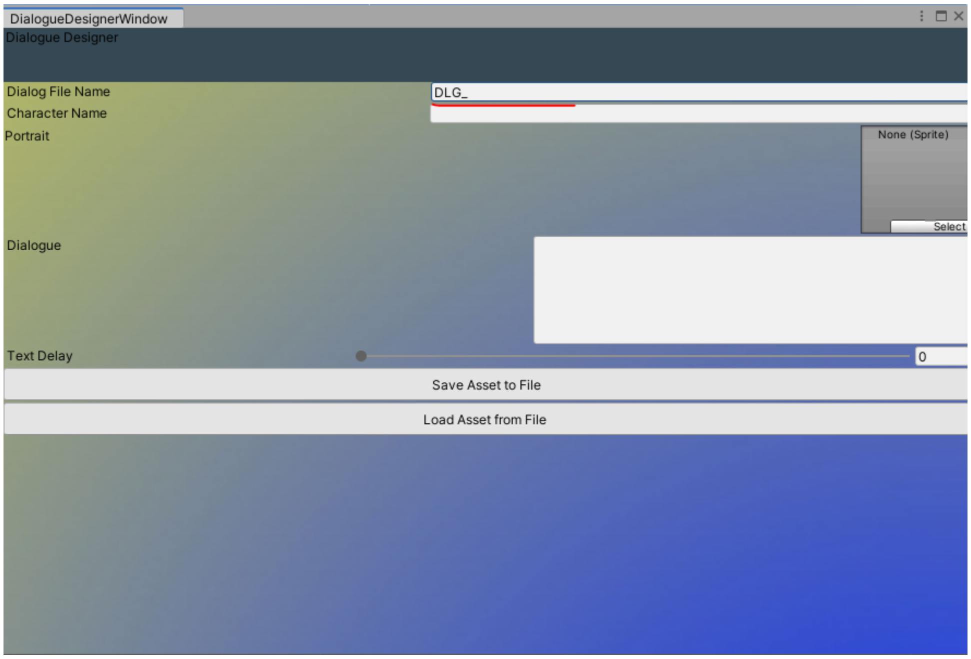
Tutorial:

<u>Creating New Dialogue Assets</u>

This guides the user on how to use the editor window to load and save dialogue assets.

Step 1: Go to Window > Dialogue Designer. It should be between Layouts and the Asset Store

Step 2: Now at the Dialogue Designer Window, the first step is to choose the file name for the object. Its default starts out with “DLG_”, but it can be replaced by selecting and typing in the field underlined in red below.



Dialog File Name: DLG_

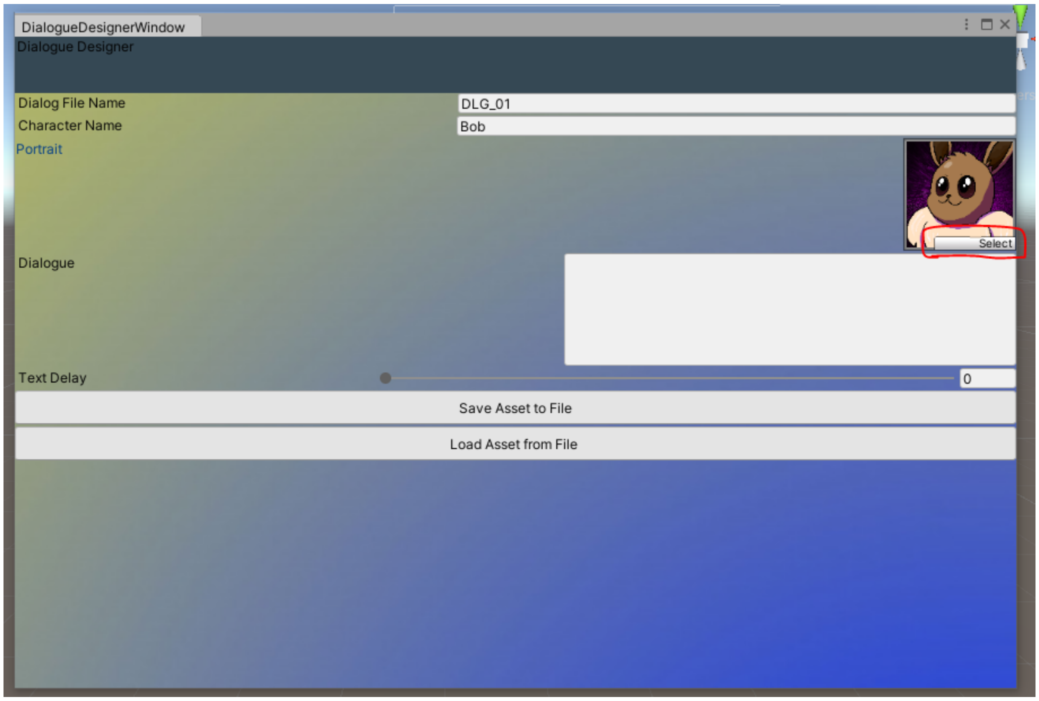
Character Name:

Text Delay: 0

This is how the empty window should look.

Step 3: Choose the character’s name. It will take any string value.

Step 4: Choose the portrait Sprite. The package comes with three starting sprites under Resources>Icons, and any custom sprites can be added.



Dialog File Name: DLG_01

Character Name: Bob

Text Delay: 0

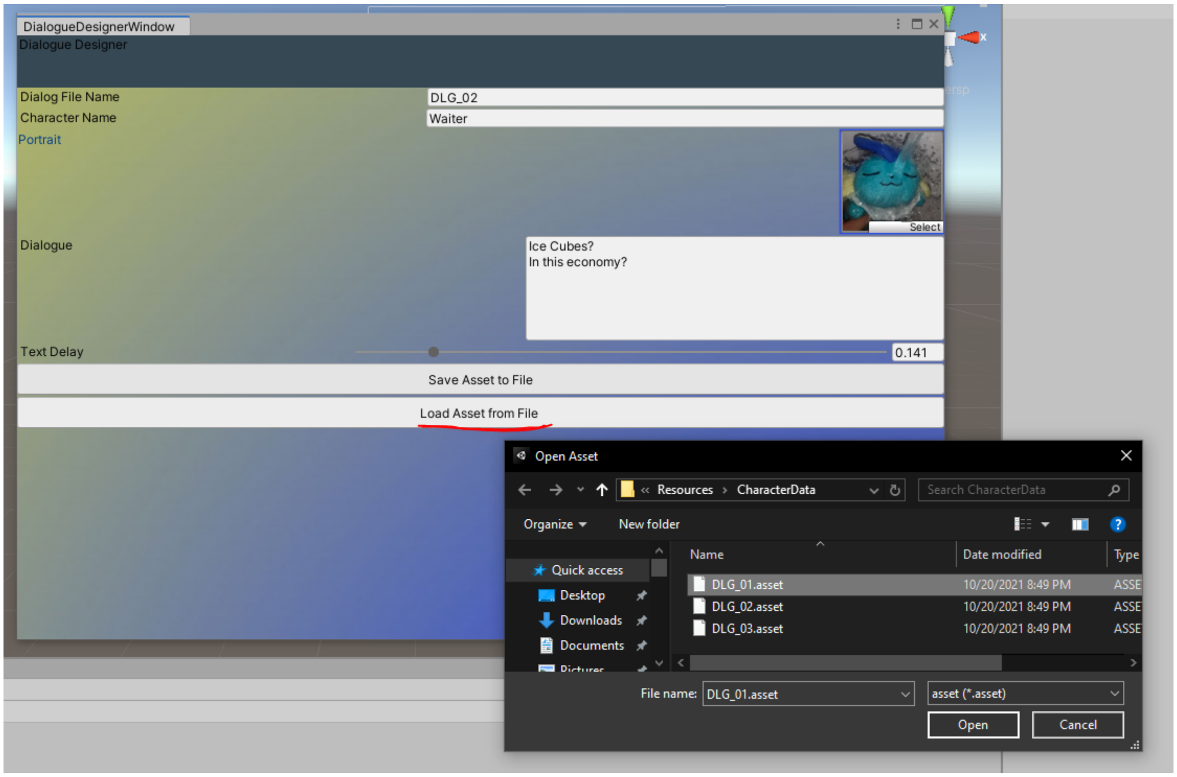
Adding a sprite. Included sprite shown, button to change sprite circled in red.

Step 5: Add Dialogue to appear in the text field on the UI

Step 6: (Optional) Change Text delay. The slider allows values between 0 and 1, representing the delay for each character printed in the text field on the UI. 0 means text is printed instantly, and 1 is a 1 second delay per character.

Step 7: Save the Asset to File. Click on the button. It should close the window and save the new file under Resources>CharacterData. If no folder exists, it will create a new folder with that name.

Step 8: Load an Asset to the Window. Click on the “Load Asset from File Button” and select from the file explorer what dialogue data you want to load.



Dialog File Name: DLG_02

Character Name: Waiter

Dialogue:
Ice Cubes?
In this economy?

Text Delay: 0.141

Resources › CharacterData

Search CharacterData

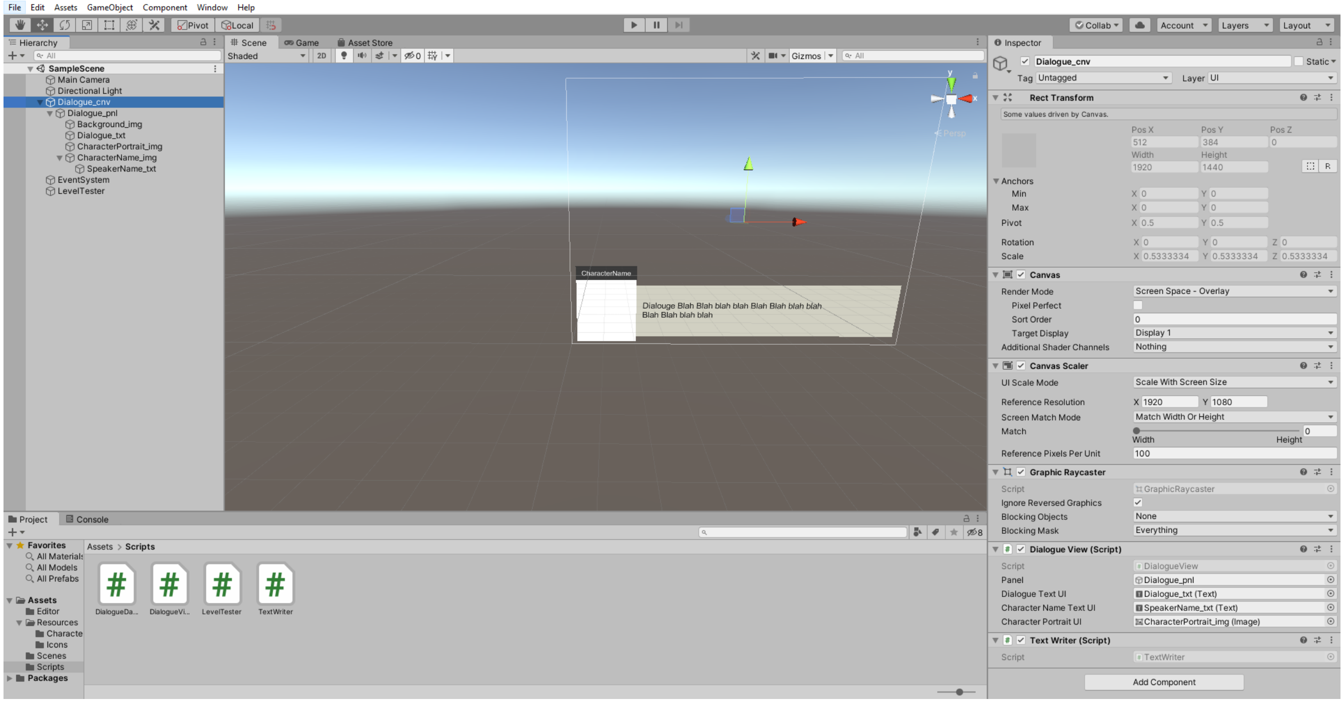
Name	Date modified	Type
DLG_01.asset	10/20/2021 8:49 PM	ASSE
DLG_02.asset	10/20/2021 8:49 PM	ASSE
DLG_03.asset	10/20/2021 8:49 PM	ASSE

File name: DLG_01.asset

asset (*.asset)

It will look for a file with .asset.

Using the Included GUI



Dialogue_cnv ☐ Static
Tag Untagged Layer UI

Rect Transform
Some values driven by Canvas.

	Pos X	Pos Y	Pos Z
	512	384	0
	Width	Height	
	1920	1440	
Anchors			
Min	X 0	Y 0	
Max	X 0	Y 0	
Pivot	X 0.5	Y 0.5	
Rotation	X 0	Y 0	Z 0
Scale	X 0.5333334	Y 0.5333334	Z 0.5333334

Canvas

Render Mode	Screen Space - Overlay
Pixel Perfect	☐
Sort Order	0
Target Display	Display 1
Additional Shader Channels	Nothing

Canvas Scaler

UI Scale Mode	Scale With Screen Size
Reference Resolution	X 1920 Y 1080
Screen Match Mode	Match Width Or Height
Match	0 (Width – Height)
Reference Pixels Per Unit	100

Graphic Raycaster

Script	GraphicRaycaster
Ignore Reversed Graphics	☑
Blocking Objects	None
Blocking Mask	Everything

Dialogue View (Script)

Script	DialogueView
Panel	Dialogue_pnl
Dialogue Text UI	Dialogue_txt (Text)
Character Name Text UI	SpeakerName_txt (Text)
Character Portrait UI	CharacterPortrait_img (Image)

Text Writer (Script)

Script	TextWriter

Add Component

Dialogue_cnv: Contains the Dialogue View and the Text Writer Script. The Dialogue View script needs a scriptable field reference to the dialogue panel, text, and portrait.
Dialogue_pnl: Holds the UI for all of the elements. The UI can be adjusted as long as it is contained in this panel and holds the objects the canvas is referencing.

Using Dialogue Objects in Code

The Dialogue asset is used by these 4 scripts:

- DialogueDesignerWindow: Handles the Editor Window to save/load dialogues
- DialogueData: Scriptable Object that stores the dialogue values.
- DialogueViewer: Loads the data from Dialogue Data and displays them on the UI.
- TextWriter: Extends Dialogue Viewer to handle text speed
- LevelTester: Example Test Code to show functionality of the other scripts. Calls DialogueViewer.

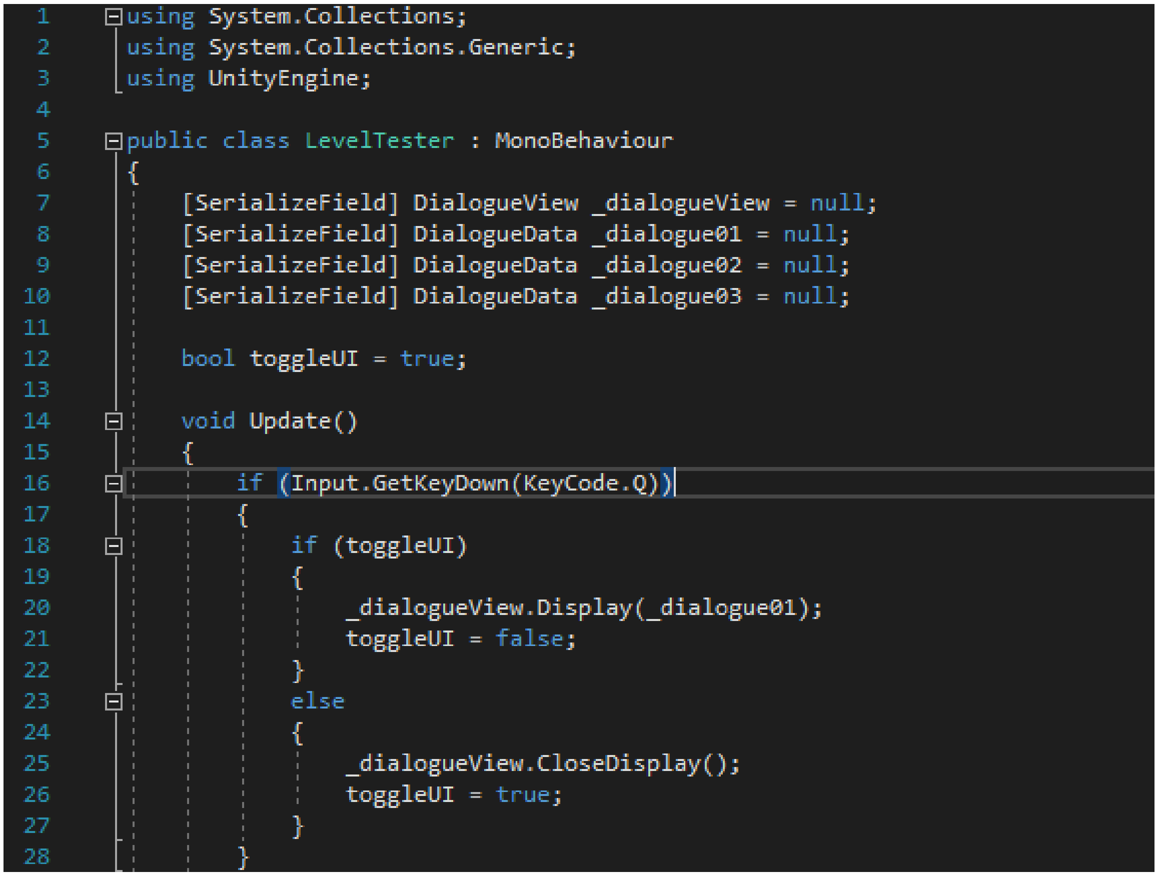
Once DialogueView has references to the UI, you can use the following functions to interface with it.

- dialogueView.Display(dialog_asset);

Opens the UI and loads in the data from the saved dialog asset.

- dialogueView.CloseDisplay();

Hides the UI away from view.



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class LevelTester : MonoBehaviour
{
    [SerializeField] DialogueView _dialogueView = null;
    [SerializeField] DialogueData _dialogue01 = null;
    [SerializeField] DialogueData _dialogue02 = null;
    [SerializeField] DialogueData _dialogue03 = null;

    bool toggleUI = true;

    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Q))
        {
            if (toggleUI)
            {
                _dialogueView.Display(_dialogue01);
                toggleUI = false;
            }
            else
            {
                _dialogueView.CloseDisplay();
                toggleUI = true;
            }
        }
```

Code from LevelTester. It calls DialogueViewer with a reference to the asset file.

Thanks for Reading!