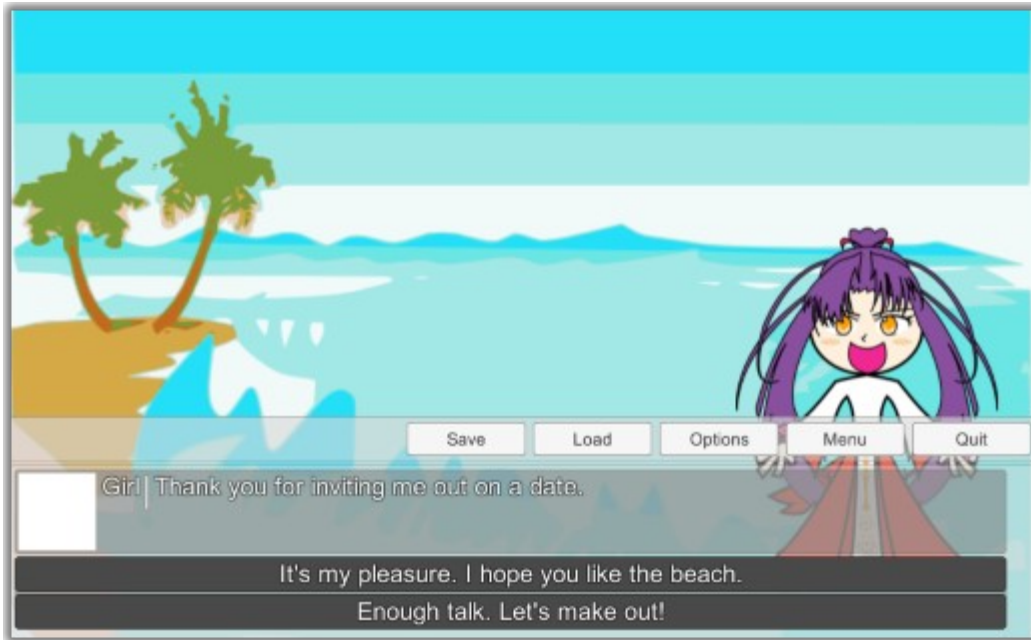


Dialogue System for Unity Visual Novel Framework

Version 1.1.1

Copyright © Pixel Crushers



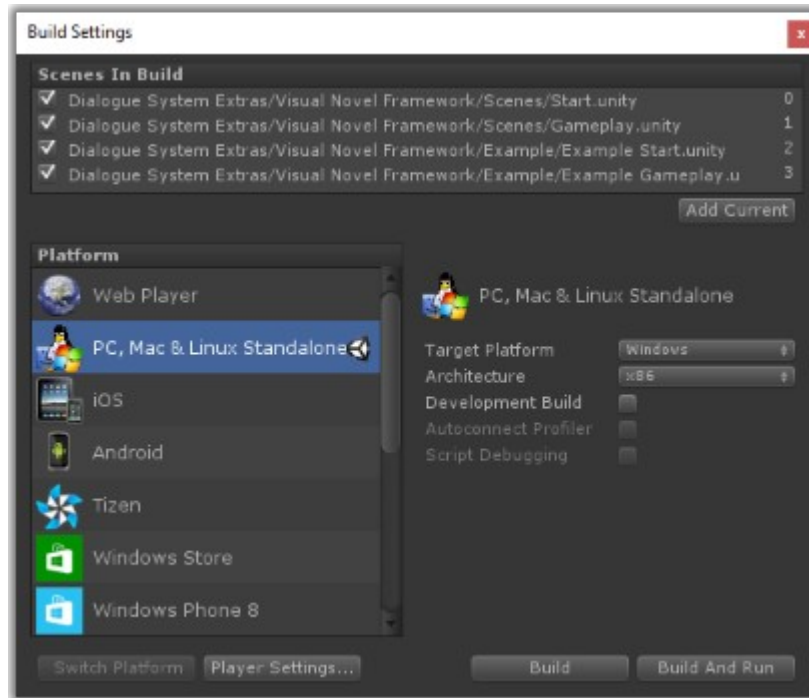
This package contains a visual novel framework for the Dialogue System for Unity. You must also import the Dialogue System for Unity (<https://www.assetstore.unity3d.com/en/#!/content/11672>).

This framework assumes you have a basic familiarity with the Dialogue System. To familiarize yourself with the Dialogue System, [click here](#).

Example Game

To play the example game, you must add the two scenes in the Example folder to your project's Build Settings:

1. Select the menu item **File > Build Settings...**
2. Drag both scenes (*Example Start* and *Example Gameplay*) into the **Scenes to Build** section.



Then play *Example Start*.

The example is very simple, but it will give you an idea of how the framework plays.

How to Make Your Own Visual Novel

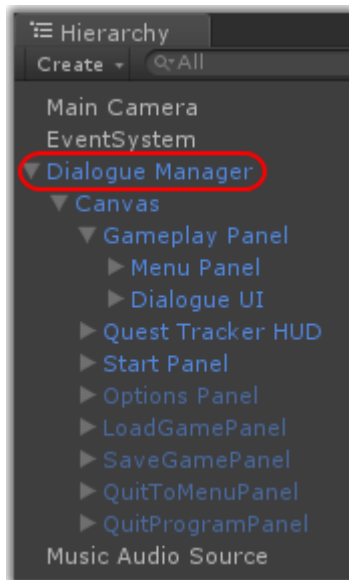
Follow the steps below to make your own visual novel:

Initial Setup

1. Select the menu item **File > Build Settings...**
2. Drag all three scenes in the Scenes folder (*Start*, *Credits*, and *Gameplay*) into the **Scenes to Build** section. If this section still has the example scenes, delete them or move them to the bottom of the list.

Start Scene

3. Open the *Start* scene.
4. The *Start* scene has a *Dialogue Manager* game object.



This is the engine that drives the Dialogue System. The *Dialogue Manager* game object will not be destroyed when changing scenes. It contains all of the user interfaces (UIs) and settings used by the Dialogue System and this visual novel framework.

In this framework, the content of a visual novel is contained in one or more “conversations,” which are lines of text linked together in a certain order.

- a) The conversations are contained in a *dialogue database*. The *Dialogue Manager* already points to a starter dialogue database located in the Data folder. To edit it, click on the *Dialogue Manager*'s blue and white logo image.
To read more about dialogue databases, [click here](#).
To read more about editing dialogue databases, [click here](#).
To use third party editors such as Chat Mapper or articy:draft, [click here](#).
 - b) The *Dialogue Manager*'s properties specify how lines are shown, such as how long to show the text, whether to wait for a continue button click after each line, etc.
To read more about setting properties, [click here](#).
5. In the Hierarchy view, expand *Dialogue Manager* > Canvas > *Gameplay Panel*. *Canvas* is a Unity UI canvas. It contains four main UIs:
- a) *Start Panel*: The main menu. As with all of the UIs, the menu graphics are placeholders. At some point you'll want replace the graphics and maybe even change the layout. The UI elements are also configured for language localization using the *Localized Text* table in the Data folder.
To read more about localizing UIs, [click here](#).

You probably want to get to the meat of your game, so you can save this step for later.

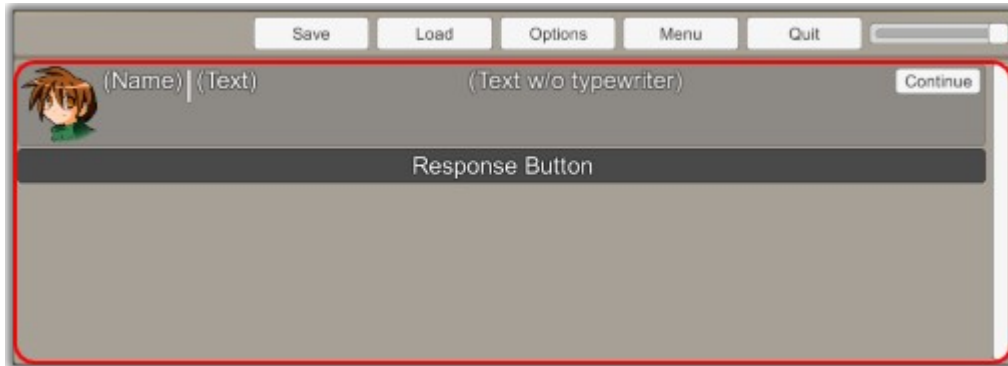


Start Panel overlaps the other UIs. Temporarily deactivate *Start Panel* to see them.

- b) *Menu Panel*: Contains gameplay control buttons such as Save, Load, Options, Menu, and Quit.



- c) *Dialogue UI*: Shows dialogue and player response menus.



To read more about dialogue UIs, [click here](#).

- d) *Quest Tracker HUD*: Shows tracked quests, if you create quests in your novel.

To read more about quests, [click here](#).

6. The *Start* scene also has a *Music Audio Source* game object. If you want background music in your main menu, set the *Music Audio Source*'s **AudioClip**.

Gameplay Scene

7. Open the *Gameplay* scene.
8. This scene contains a modified copy of the *Dialogue Manager* game object that lets you play your game directly from this scene instead of going through the *Start* scene first. This is just a convenience to be able to test your game more easily.

When you modify the master copy of the *Dialogue Manager* game object in the *Start* scene, click the **Apply** button to apply the changes to the Unity prefab. The copy of the *Dialogue Manager* in the *Gameplay* scene will inherit those changes.

9. In the *Gameplay* scene, you can set a different background image, configure a different *Music Audio Source*, etc.
10. You can use multiple gameplay scenes if you want. To change scenes during a conversation, use the included `VNLoadLevel()` sequencer command, which has the same syntax as [LoadLevel\(\)](#).

Writing Your Novel

To begin writing your novel, click on the *Dialogue Manager's* blue and white logo. This will open the [Dialogue Editor](#). Then follow these basic steps:

11. Click on the *Actors* tab.
 - a) From the dropdown menu in the upper right, select **New Actor** to add each character.
 - b) You can set one or more portrait images for each character. However, typically in visual novels a portrait image is not shown next to the text. Instead, a larger character image is shown onscreen. We'll cover that in the next step.
12. Click on the *Conversations* tab.
 - a) The dropdown in the upper left starts with one conversation titled *Start Conversation*. The framework will start this conversation when starting a new game. You can write your entire novel in this conversation, or you can define additional conversations and link between them.
 - b) Each dialogue entry node has **Dialogue Text**, which is the text shown onscreen; a **Sequence** field, which can play sequencer commands to make things happen in the game; and other fields. Sequences are incredibly useful to show and hide game objects such as character images, play sounds, load new levels, and more. We recommend you create all of your backgrounds and character images as game objects (such as UI images under the Canvas), and then use the [SetActive\(\)](#) sequencer command to activate and deactivate them as needed.

To read about the fields in a dialogue entry node, [click here](#).

To read about sequences, [click here](#).

To read about language localization, [click here](#).

13. On the *Variables* tab, you can define variables to remember choices that the player has made.

The dialogue database already has a definition for a variable named *CurrentStage*. You can set this variable at any point in the conversation. The contents of *CurrentStage* will be included in the summary shown next to each saved game in the load game menu.

To read more about variables, [click here](#).

To read more about setting and checking variable values, [click here](#).

14. On the *Quests/Items* tab, you can define quests. You can also use quests to show a “to do” list for the player.

15. Normally the game ends when the player reaches the end of the conversation. If your game incorporates other kinds of gameplay between conversations, inspect the Dialogue Manager's *Menus* component and untick **Return To Menu On Conversation End**. To return to the main menu in this case, you will have to manually use the `VNMainMenu()` sequencer command.

Final Steps

15. Open the *Credits* scene. The game switches to this scene when the player clicks the *Credits* button in the main menu. Customize this scene to contain your own credits. The *Credits Panel* in the *Start* scene's canvas is shown on top of the scene. It contains a *Back* button that returns to the main menu.
16. When you're ready to distribute your game, open the *Build Settings* window.
17. Make sure you don't have any extra scenes in the **Scenes to Build** list such as the example game scenes. Then set your platform and click the **Build** button.

Need Help?

We're here to help! If you have any questions or feature requests, please contact us at:

Email: support@pixelcrushers.com

Forum: <http://www.pixelcrushers.com/phpbb/index.php>

Games Showcase

Write a game with the Dialogue System's Visual Novel Framework? We'll be happy to feature it on the Dialogue System's Games Showcase. Just send screenshots and/or a video and a link to the game's website to support@pixelcrushers.com!