Manu-Scriptwriter Testing Manual

Thank you for volunteering to take part. This document should cover any setup necessary for testing, as well as the available functionality.

Setup

If you have any issues during setup, or with using the Unity Engine itself, feel free to request help.

What do you need?

- A copy of Unity 6, version 6000.0.19f1 or later.
- A text editor, such as Notepad, TextEdit, Notepad++, or otherwise. Most computers come with a text editor pre-installed.
- A copy of the project itself, which should have been provided alongside this manual.

Installing Unity 6

If you already have a copy of Unity 6 installed on your device, you can skip this heading.

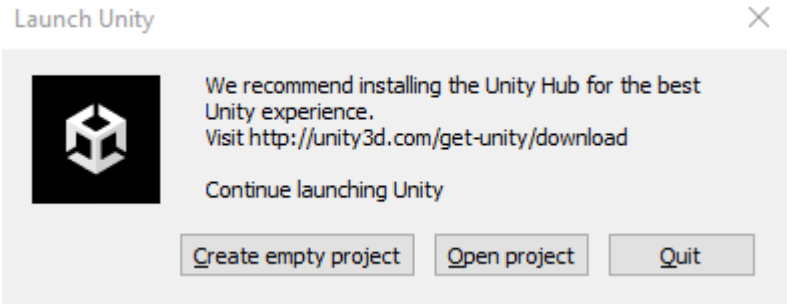
An installation for version 6000.0.19f1 for Windows, MacOS, or Linux, can be found on the Unity website. https://unity.com/releases/editor/whats-new/6000.0.19

To install, download the installer for your machine, run the executable file, and follow the setup wizard. Wait for the install to finish before continuing.

Opening the Project

Download and unzip the provided copy of the Manu-Scriptwriter project.

On opening Unity 6, select Open Project.

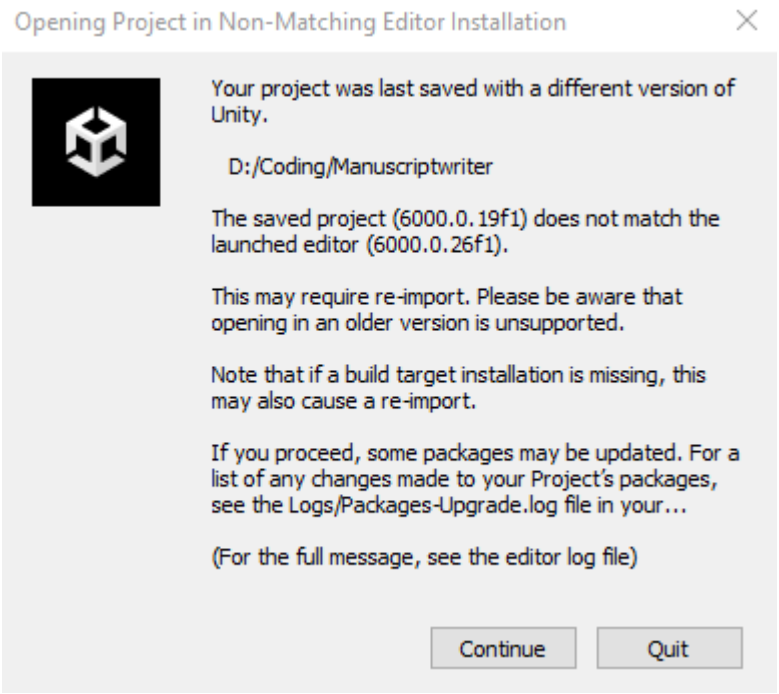


Then, navigate to the unzipped Manu-Scriptwriter folder. Click on the folder or open it, then click select folder. Wait for Unity to load the project.

Optional: Updating the Project to a Later Version

If, on opening the project, you are met with the following window, it is safe to click continue, then wait for it to load once more.

The project does not depend on version 6000.0.19f1, and the update process should not require any other input. If you experience any issues, feel free to request help, or delete and re-download the project and the correct version of the editor.



Testing Instructions and Hints

From here on, what you do with the project is up to you. You should not need to touch any of the code itself, outside of anything within a text file.

Play around with the provided functionality using the Manu-Scriptwriter language or add new things to scenes based on existing objects until you are satisfied that you've made a good scene.

Controls

To start a scene, enter Play mode by clicking the Play button in the top middle of the editor.

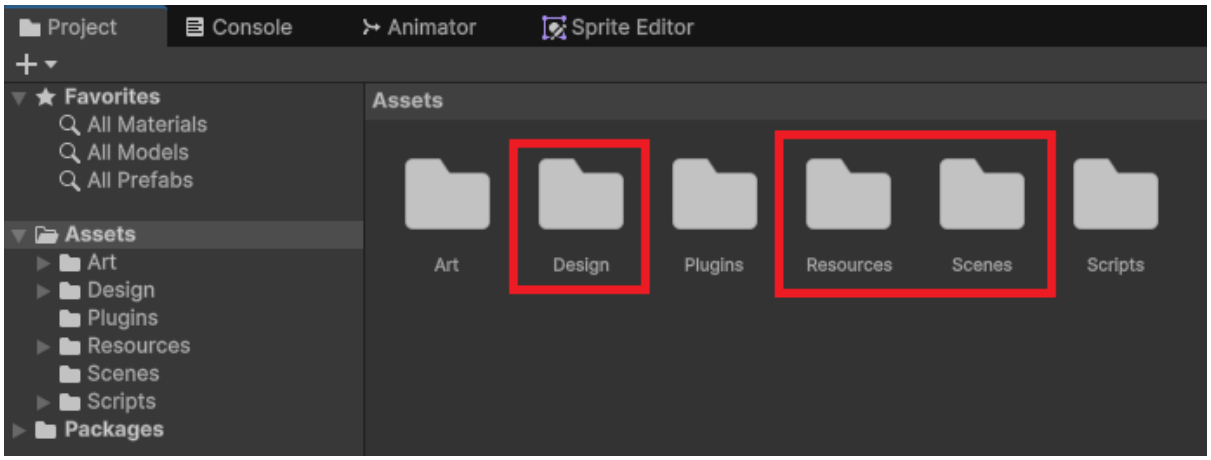
While in Play mode:

- Use WASD or the arrow keys to move the player character.
- Press Enter to interact with something.
- Press Enter to advance text during an interaction.

To stop the scene at any time, leave Play mode by clicking the stop button. You should do this whenever you want to edit a script, or if you encounter an issue, such as falling off the world, or encountering bugs.

Organisation

There are a few folders which you'll either work in or take assets from, which can be viewed in the project window.

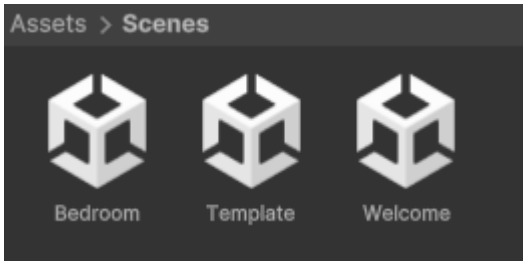


Scenes Folder

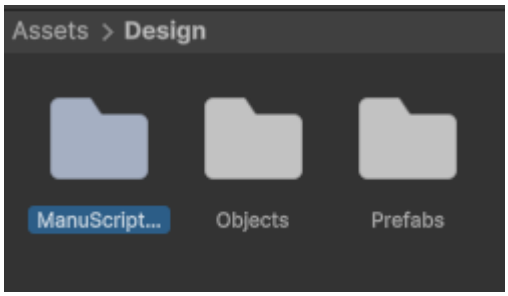
Within this folder are two sample scenes with simple interactions, and one template with everything but the interaction set up for use.

To open a scene, double click the icon for the desired scene. For a simple example, I recommend the ‘Welcome’ scene!

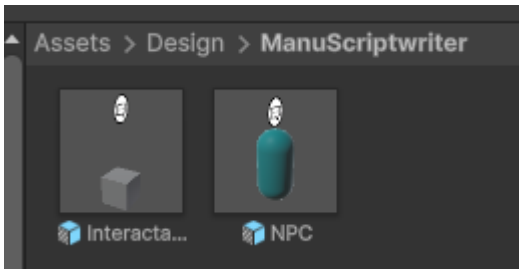
Feel free to edit an existing scene and scripts to your own liking, or create something entirely new within ‘Template’. Whichever is easiest for you.



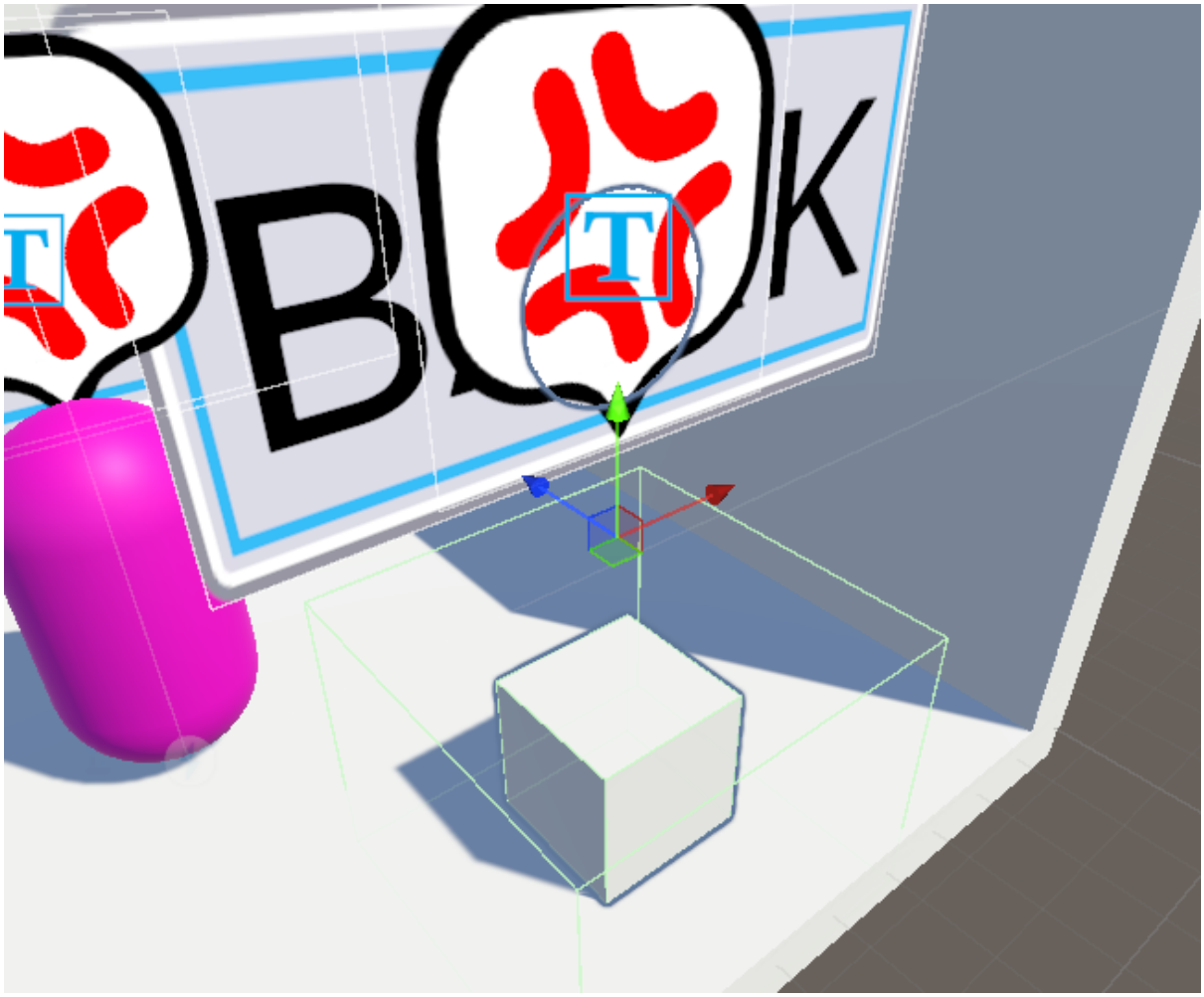
Design/ManuScriptwriter Folder



Within this folder are common, reusable objects for a scene/interaction, known as prefabs. Importantly, this includes an NPC and an “Interactable Object” prefab, which has been setup with everything necessary to work with the text scripts in advance.



To use a prefab, simply click and drag from the project window and into the scene or the hierarchy windows. You can move that object within the world either using the transform tool or the Transform section of the inspector window.



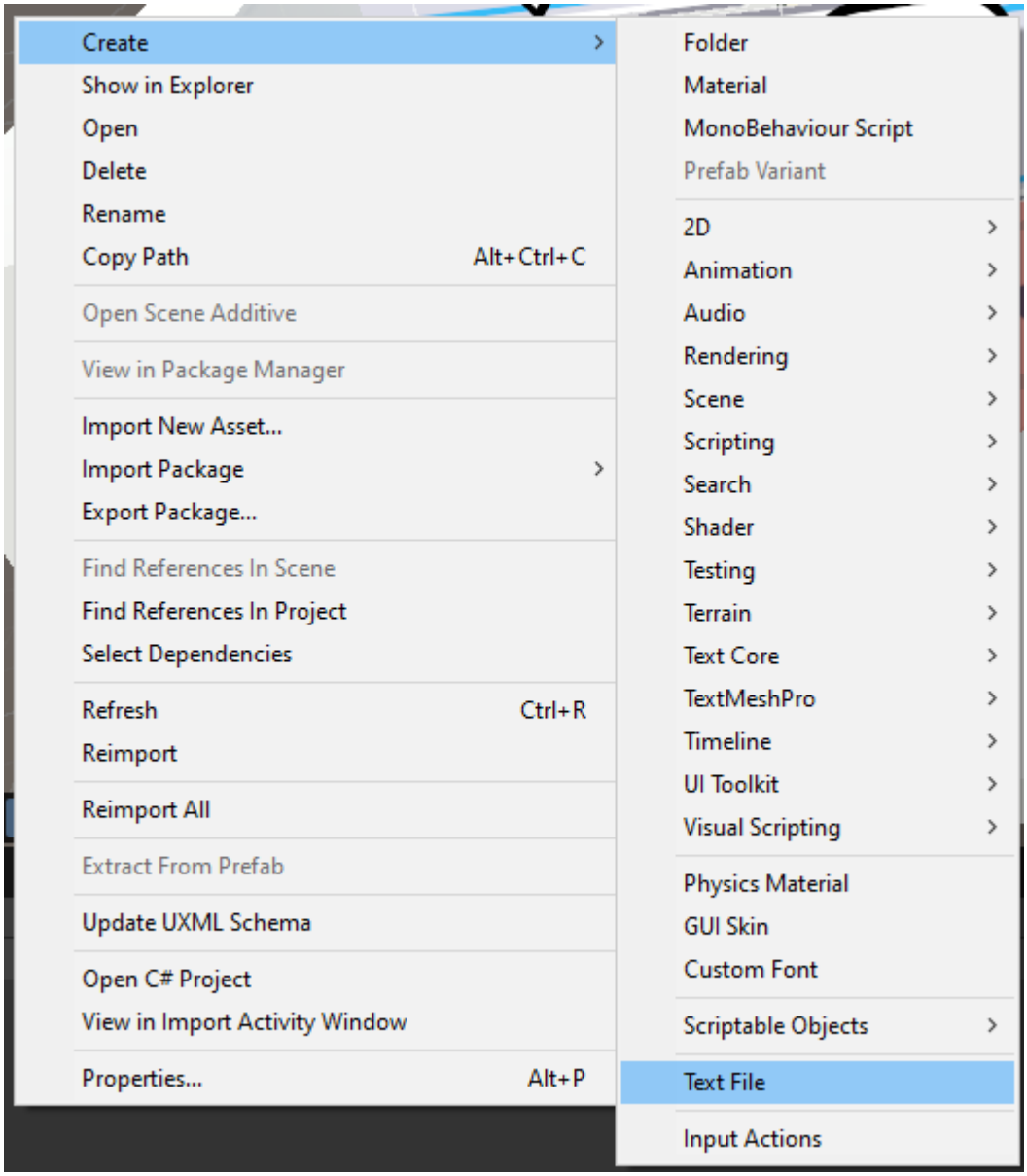
If you place an object in the scene that doesn't use one of these prefabs, and then decide to make it interactable later, you can right click that object in the hierarchy, then go to ManuScriptwriter -> Make Selection Interactable. This should add everything necessary.

Resources Folder

Within this folder are any text files used within scenes in the project. All scripts you create should be placed somewhere within this folder, regardless of how you organise them.

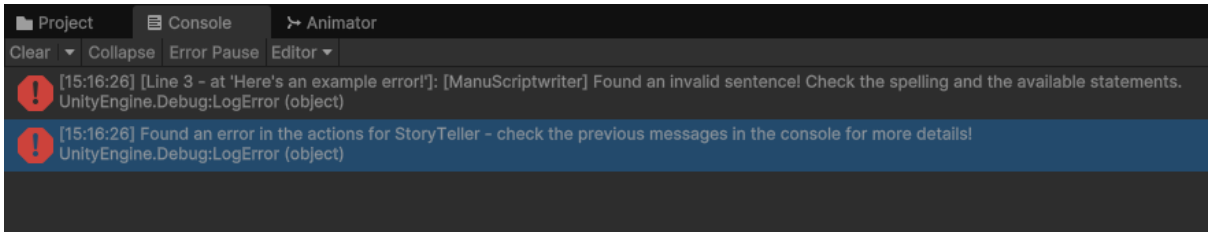
You should be able to open a text file in this folder by double clicking the file.

To create a new text file, right click within the project window, go to Create -> Text File.



Errors, Problems, and Debugging

Chances are, one of your custom scripts will break if they aren't understood by the game. In the case that an error occurs, check the Console window.



If the error message mentions "[ManuScriptwriter]", you'll want to follow the instructions to correct something in one of your own text files. See if you can figure out how to fix the issue yourself!

If not, then you may have encountered a bug with my code that is not your fault. While this project has been provided to you with the expectation that it will not break for the purposes of this research, there is always the chance that something like this will occur.

In general, make note of the problem, but see if you can work around it.

Additional Guidance

If you have trouble with some of the core functionality within the Unity editor, unrelated to the provided language, you may want to look towards the official Unity documentation or tutorials.

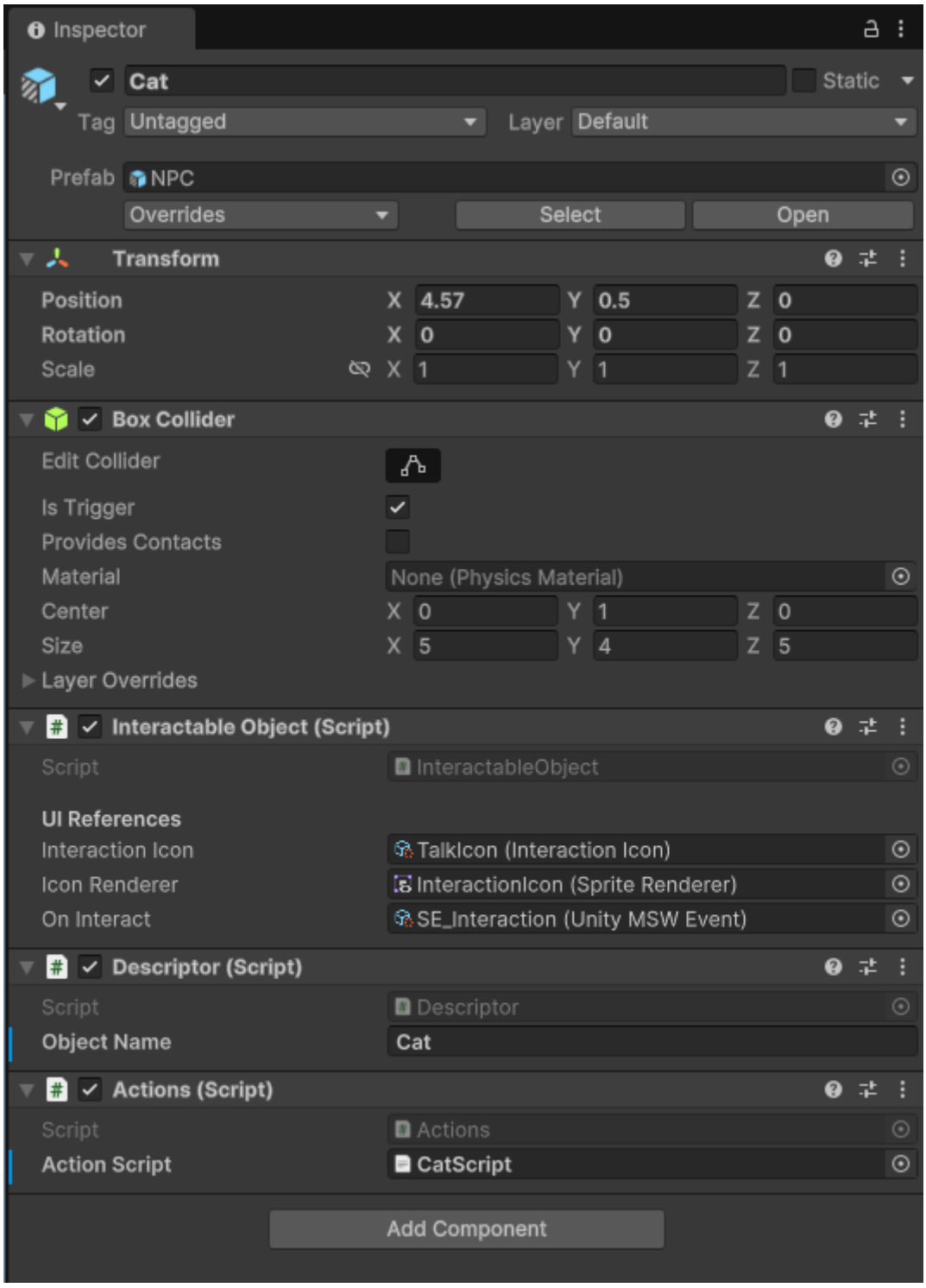
This tutorial serves as a good introduction to the Unity Editor, though not all the content covered may be relevant or necessary.

https://learn.unity.com/tutorial/explore-the-unity-editor-1

Using Manu-Scriptwriter

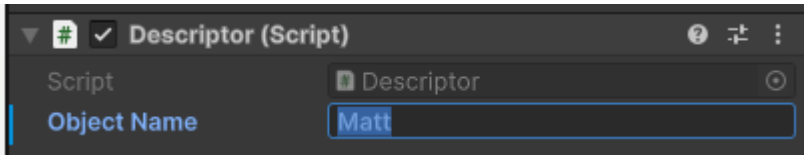
Within Unity Editor

Some use of the Unity Editor is required for a scene to work, as a script will not run otherwise. This logic is displayed in the inspector panel, usually to the right of the screen.



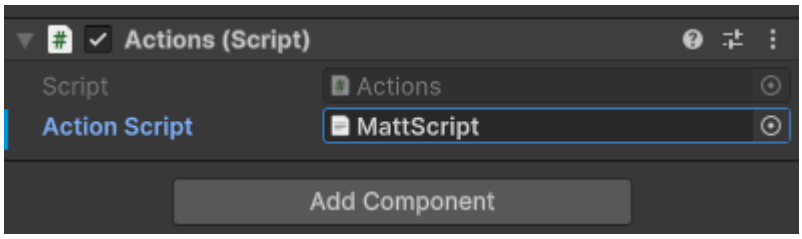
Objects and NPCs can be given a name using the “Descriptor” component in the inspector. This can be changed by editing the “Object Name” property.

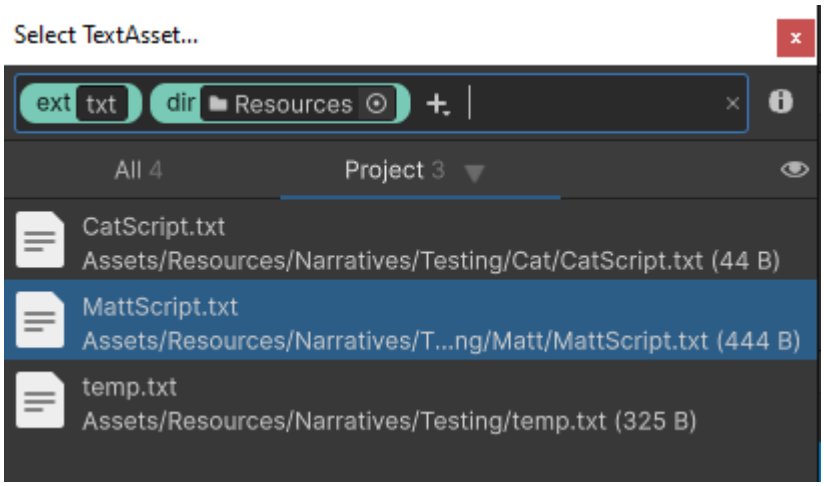
Multiple objects in one scene should not share the same name.



Each object and NPC holds its own script within the “Actions” component in the inspector. This can be changed by editing the “Action Script” property.

Either click and drag the script from the resources folder in the project window to this property or click the target button and search for the file in the search window.





Within Manu-Scriptwriter Scripts

Parts of any sentences that can be changed are highlighted in blue.

Events

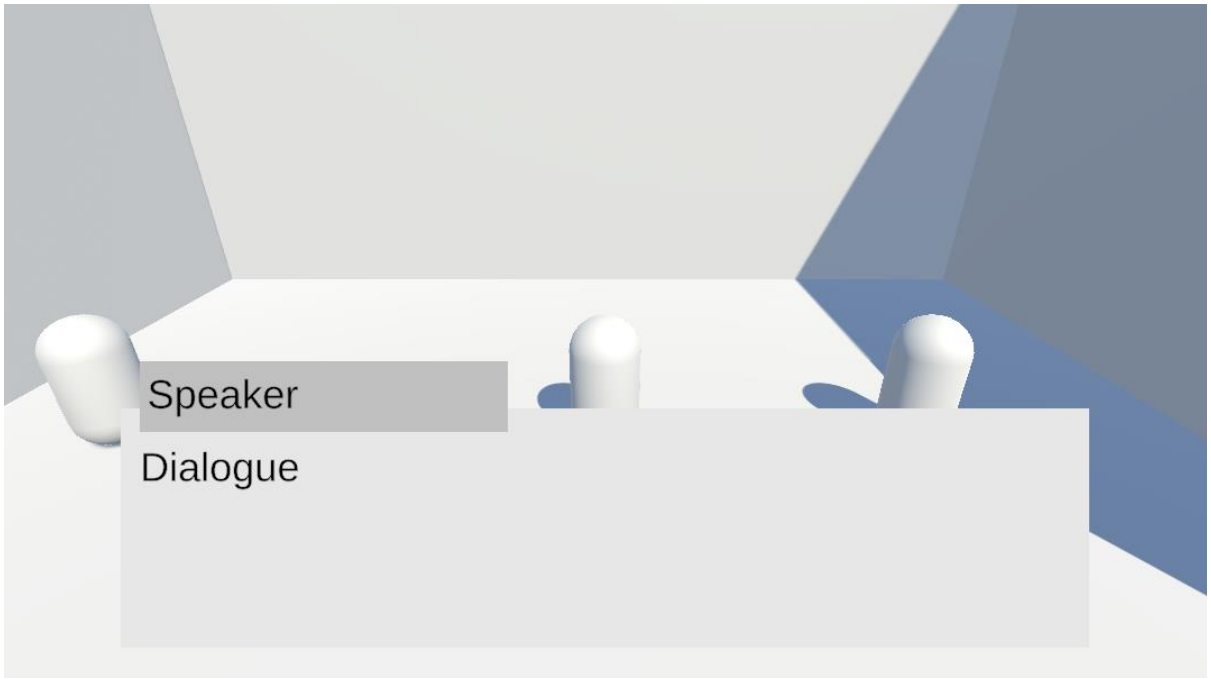
When the Player speaks with Target
When the Player talks to Target
When the Player talks to the Target
When the Player interacts with the Target

The only currently available event within Manu-Scriptwriter. This indicates that dialogue should occur after interaction with the specified target. “Target” must share a name with an object or NPC.

Dialogue

Speaker: Dialogue

The standard method of writing a line of dialogue. The Speaker does not need to share a name with an object or an NPC.



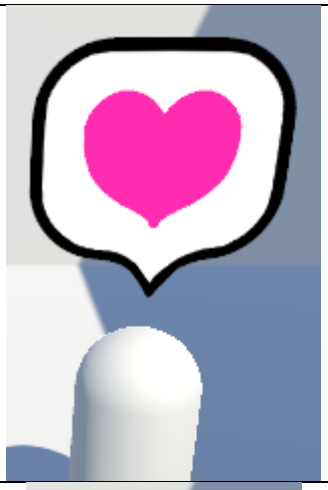
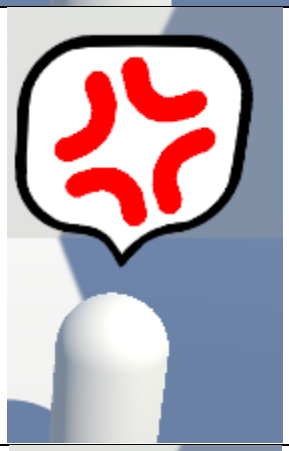
Speaker barks: Dialogue
The Speaker barks: Dialogue

An alternative method of communicating dialogue. This is intended for short sentences. “Speaker” must share a name with an object or an NPC. After a short period of time, the bark will disappear automatically.

Target feels emotion.
The Target feels emotion.

A simple method of conveying little emotions using iconography. “Target” must share a name with an object or an NPC. The available emotions are as follows:

“happy” or “sunny”	

“love”	
“mad”, “angry”, or “annoyed”	
“nervous” or “awkward”	
“sad” or “depressed”	

“tired”	

Camera

The camera freezes.

The camera will stop moving entirely. This continues after the end of an interaction.

The camera focuses on Target.
The camera focuses on the Target.

The camera will follow the specified target. “Target” must share a name with an object or an NPC. This continues after the end of an interaction. Return focus to the player by specifying “**The camera focuses on the player.**”

The camera shakes from side to side.

This causes the camera to shake erratically from left to right. This effect is removed at the end of an interaction.

The camera stops shaking.

This stops the camera from shaking if it was previously shaking.