

Overview

The Dialogue system is a modular system used to display and control game dialogue UI elements. The dialogue system can be used to determine when to display dialogue, what text to display, what image to display, how quickly displayed text should be gone through, link various text lines to each other, and determine what happens once dialogue has finished.

This system is designed to be very versatile for use in many different types of games.

How to Use

In order for the dialogue system to work, the following objects must have the following scripts and objects. Scripts made for the Dialogue System are *italicized*.

-**Canvas:** The Dialogue Box is the GUI that is displayed when the Dialogue System is running.

-The Canvas must also have the following components attached to it.

-*Dialogue UI Script:* Controls and tracks the usage of the Dialogue system.

-**Dialogue Box:** The container that holds all Dialogue UI elements. Set to *Dialogue Box*.

-**Char Portrait Holder:** Determines where the character portrait sprite is located. Set to the *Character Portrait* child object of the Dialogue Box.

-**Text Label:** Determines where the dialogue text is located. Set to the *Text (TMP)* child object of the Dialogue Box.

-**Controls:** Allows the player to confirm or skip the dialogue. Set to the **GuiControls** Input Action Asset.

-*Typewriter Effect Script:* Prints dialogue text one letter at a time.

-Audio Source: Where the typewriting sound effect is played from. As of this writing, it is set to an Audio Source on the Main Camera to ensure the player will always hear it regardless of what they are interacting with.

-Player Input: Used to let the player to continue or skip the text.

-Use the **Default** map of the **GuiControls** Input Action Asset. Set the behavior to “Invoke C Sharp Events”

-*Response Handler Script:* Creates response buttons which allow players to respond to dialogue prompts.

-**Response Box:** The holder container of all response buttons. Set to the *ResponseBox* child object of the Dialogue Box.

-**Response Button Template:** A template that determines what all option buttons look like. Set to the *Option Template* child object of the Dialogue Box.

-**Response Container:** A horizontal layout group that holds the individual response buttons. Set to the *Responses* child object of the Dialogue Box .

-Player:

-Player Input: Use the **Player_Test** map of the **PlayerInput** Input Action Asset. Ensure the *Interact* callback event is set to the *DialogueController.GetInteract* function.

-*Dialogue Controller* Script: Allows the player to initiate a dialogue.

-**Dialogue UI**: The graphical elements of the Dialogue system. Set to Canvas.

-**Interactable Object**: Whatever the player is interacting with. This can be an NPC a sign, etc.

-Collider: As of this writing, any object that uses the dialogue system must have a collider with 'Is Trigger' marked as true. The object uses this to set what Dialogue Object will be used.

-*Dialogue Activator* Script: Sets a Dialogue Object to the player and allows the player to initiate a dialogue. Set to the desired Dialogue Object.

Dialogue Objects:

-Description: Dialogue objects determines what text is displayed in a dialogue box, how it is displayed, and how quickly it is written.

-How to make: Right click anywhere in a Unity Asset folder or sub-folder. Mouse over the *Create->* option and then mouse over the *Dialogue->* option. From there click on the *DialogueObject* option.

-Elements:

-Dialogue List: A list of different dialogues. Each dialogue has the following elements. You can put as many dialogue objects in the list as you like. You can use Dialogues to convey different speakers in a conversation, or even the same speaker talking in a different way. If you need to change anything other than the text, make a new Dialogue element

-Text: A list of strings that determine what dialogue is shown. You can put in as many text objects as you like. When the player continues from a text box, the dialogue box will clear and then fill in with text from the next element in the list. While the text changes, the other elements in the Dialogue Object do not.

-Character Template: A Character Template Object that fills in the rest of the elements with predetermined values. These values can be changed without effecting the Character Template.

-Character Portrait: A sprite that displays the speaker.

-Voice: An Audio Clip that plays when the text is typed out.

-Text Speed: A float that determines how quickly the text is typed out. The larger the number, the faster the write out.

-Pause Time: A float that determines how long the system waits at a punctuation mark before continuing to type out the text.

-Responses: Choices players can make in response to dialogue. Responses are linked to the final text element of the final dialogue object.

-Response: Takes a string and creates a button labeled with the provided string.

-Dialogue: A Dialogue Object that will start up if the player presses the accompanying response button. Ends the dialogue if no Dialogue Objects are provided.

Character Template Objects:

-Description: Character Templates are used to quickly and consistently fill in Dialogue Object fields.

-How to make: Right click anywhere in a Unity Asset folder or sub-folder. Mouse over the *Create->* option and then mouse over the *Character Template->* option. From there click on the *CharacterTemplateObject* option.

-Elements:

-Char Name: A string that tracks the speaker's name. As of this writing, this value currently has no function, but in the future will be used to display who is speaking in a given dialogue.

-Char Portrait: A sprite that displays the speaker.

-Voice: An Audio Clip that plays when the text is typed out.

-Text Speed: A float that determines how quickly the text is typed out. The larger the number, the faster the write out.

-Pause Time: A float that determines how long the system waits at a punctuation mark before continuing to type out the text.

Dialogue Response Events: Allows you to run any script when the player selects a response.

-Set the *Dialogue Response Events* script on an the intractable object with the *Dialogue Activator* you wish to respond to.

-Provide the Dialogue Object that has the responses you want to add an event to.

-Select from the provided list of events, and add your desired function to the event. You can add as many functions you'd like to any of the events.

Dialogue Selector: Changes which dialogue object will run when the player interacts with the object based on event flags.

-Set the *Dialogue Selector* script on an the interactable object with the *Dialogue Activator* you wish to respond to.

-Fill the **Dlog List** with the dialogue objects, as well as the name and value of the event flags you wish to set them with. *Order matters!* Which ever element is listed first has highest priority.

-Place the default dialogue object you desire in **Def Dlog**. This is the dialogue object that will be used if none of the elements in the list are valid.

- Optional: If a dialogue object used by the selector will change an event flag that controls an element in the **Dlog List**, provide a link to that flag in **Flag Event SO**. This will allow the dialogue to update without needing to place an extra event in each Dialogue Object.

Player use:

-Controls

-Interact (PlayerInput): As of this writing, set to 'Z'. Used to initiate the dialogue event.

-Confirm (GuiControls): As of this writing, set to 'Z'. Used to proceed to the next text box or to end the dialogue event.

-Skip (GuiControls): As of this writing, set to 'X'. Used to skip the text typing and to fill the dialogue box immediately.