Dialogue System Package

Contents

[Creating and Editing Conversations 3](#_Toc513475212)

[Dialogue Entries 3](#_Toc513475213)

[Responses 3](#_Toc513475214)

[Conditions 3](#_Toc513475215)

[Dialogue Events 3](#_Toc513475216)

[Cutscene Events 3](#_Toc513475217)

[Using Conversations 3](#_Toc513475218)

[Dialogue Manager 3](#_Toc513475219)

[Dialogue UI 3](#_Toc513475220)

[Localization Manager 3](#_Toc513475221)

[Actors 3](#_Toc513475222)

[Cutscene Manager 3](#_Toc513475223)

[Class Reference 4](#_Toc513475224)

[Conversation 4](#_Toc513475225)

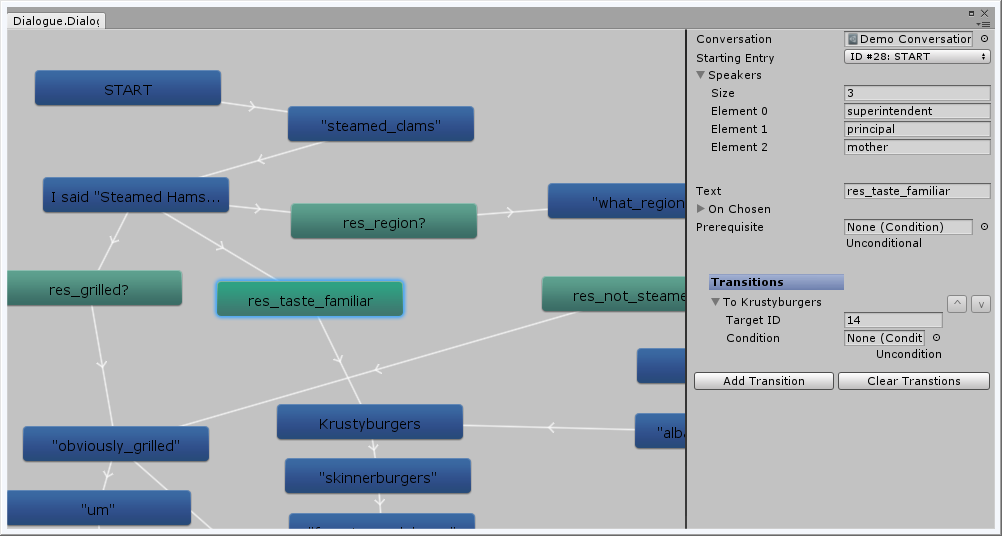
# Introduction

# Creating and Editing Conversations

A Conversation can be created through the Assets/Create/Dialogue submenu. Conversation assets can be edited using the inspector or with the Dialogue Window.

## Dialogue Window

The Dialogue Window allows you to view and edit conversations.



On the left is the Node Panel, showing dialogue entries (blue) and responses (green) as nodes. Dialogue entries and responses can be created, connected, and deleted in the Node Panel. Clicking a dialogue entry or response selects it in the Edit Panel. Nodes can be dragged around the Node Panel, and dragging with the middle mouse buttons moves around the Node Panel.

On the right is the Edit Panel. It allows the current conversation, and the selected dialogue entry or response, to be edited like in the inspector.

Conversations have a list of Speakers which can be assigned to dialogue entries. These are keys used by the Dialogue Manager to identify the Actor object involved.

## Dialogue Entries

Dialogue Entries are the individual parts of a conversation, containing lines of dialogue. They contain text to display, events to perform and animations to play.

**Creating new Dialogue Entries**

In the Inspector: Click the Add Entry button

In the Dialogue Window: Right click in the Node Panel and click the Add Dialogue Entry option

**Deleting Dialogue Entries**

In the Inspector: Select the Dialogue Entry with the dropdown list, and click the Remove Entry button

In the Dialogue Window: Right click on the Dialogue Entry and select the Delete option

## Transitions

Dialogue Entries and Responses have Transitions. These allow the Dialogue Manager to determine the next Dialogue Entry in the conversation. Transitions may have Conditions for being followed, and when there are multiple Transitions on a dialogue entry the first one meeting its conditions is selected.

**Adding Transitions**

In the Dialogue Window: Right click a Dialogue Entry or Node and select the Add Transition option. Click the Dialogue Entry to transition to.

**Removing Transitions**

In the Inspector or Dialogue Window Edit Panel, open the Transitions list on the selected dialogue entry or response, right click the transition and select “Delete Array Element”. The Clear Transitions button can be used to remove all transitions.

**Changing order**

To change the order of transitions in a list, open the Transitions List in the selected dialogue entry or response, and click the up and down buttons to the right of the transition

## Responses

As well as transitions, dialogue entries can have Responses, which are options the player can select from. Responses have text to display for that option, events to execute on choosing it, and the transitions to Dialogue Entries from that response.

**Adding a response**

In the Inspector: Open the Responses list on the selected Dialogue Entry and click Add Response

In the Dialogue Window: Right click a Dialogue Entry and select the Add Response option

**Removing a response**

In the Inspector: Open the Responses list and click the “X” button to the right of the response. Or click Clear Responses to remove all responses.

In the Dialogue Window: Right click the Response and select the Delete option

## Conditions

Conditions can be added to responses and transitions to control when they can be selected. To create a condition, open the Assets/Create/Dialogue/Conditions submenu and select the type of condition to create.

## Dialogue Events

Dialogue Events are actions to perform when a dialogue entry is reached or an response is chosen. They can be created with the Assets/Create/Dialogue/Event submenu. When added to a dialogue entry or response, the Target and Parameters for the event can be set. Target can be used to select a particular Actor, or could be left blank. Parameters is used to control the event. For example, the Set Flag Event’s Parameter field is used to specify the flag to set.

## Cutscene Events

Cutscene Events are used to play animations for a dialogue entry. The Target field is a key used to identify an Animator for the Cutscene Manager, and the Animation field specifies the animation to play.

# Using Conversations

## Dialogue Manager

The Dialogue Manager component is used to play conversations. It needs to have references to DialogueUI and CutsceneManager components so it can display dialogue and play animations.

To play a conversation, assign a conversation asset to the Conversation field, and call the StartConversation() function from a script.

The Dialogue Manager has a dictionary of strings to DialogueActor objects. This is used by conversations to identify the DialogueActor matching the Speaker set for a dialogue entry.

Dialogue Manager also has the unity events On Conversation Start and On Conversation End, which are invoked when a conversation starts or ends.

## Dialogue UI

The DialogueUI displays dialogue and responses for the Dialogue Manager. To do this it needs a UIDisplayStrategy component, where you implement the specific methods for displaying dialogue and accepting player input for your game.

The DialogueUI needs references to the UIDisplayStrategy used, and the LocalizationManager component if localized lines are used by the Display Strategy.

The DefaultTextStrategy and ResponseButtonStrategy strategies are included with this package.

## Localization Manager

The LocalizationManager component is used to get localized dialogue from a file. It uses a LocalizationLoader asset, which contains and parses the localization file.

Calling SetLocale on the Localization Manager makes it request lines for that locale from the Localization Loader. If found the lines are loaded and used for the GetLine function.

This package comes with the CSVLoader localization loader, which loads dialogue from a Text Asset formatted as a comma separated value spreadsheet with the first row giving locale codes.

## Dialogue Actors

\*\*\*\*\*\*TODO\*\*\*\*\*\*

## Cutscene Manager

The Cutscene Manager component is used to play animations specified by a Dialogue Entry.

To use it, the Animators dictionary needs to be filled out with the keys used in any CutsceneEvent and the Animator component they should refer to.

Calling the DoCutsceneEvents method will play the animations specified for an animator.

# Class Reference

## Conversation

A ScriptableObject containing a list of DialogueEntries which make a conversation.

## DialogueManager

|  |  |
| --- | --- |
| **Method** | **Description** |
| StartConversation() | Begins the conversation |
| NextEntry() | If the current dialogue entry has valid transitions, goes to the next dialogue entry. If none can be found, ends the conversation. |
| ResponseSelected(int id) |  |
|  |  |
|  |  |
|  |  |
|  |  |
| EndConversation() | Ends the conversation |

|  |  |
| --- | --- |
| **Unity Event** | **Description** |
| OnConversationStart | Invoked after a conversation has started |
| OnConversationEnd | Invoked after a conversation is ended |