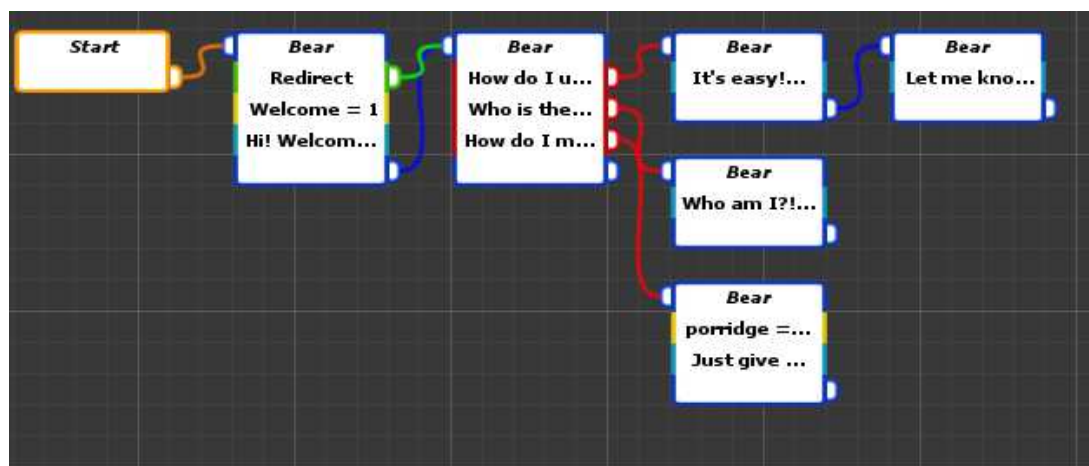
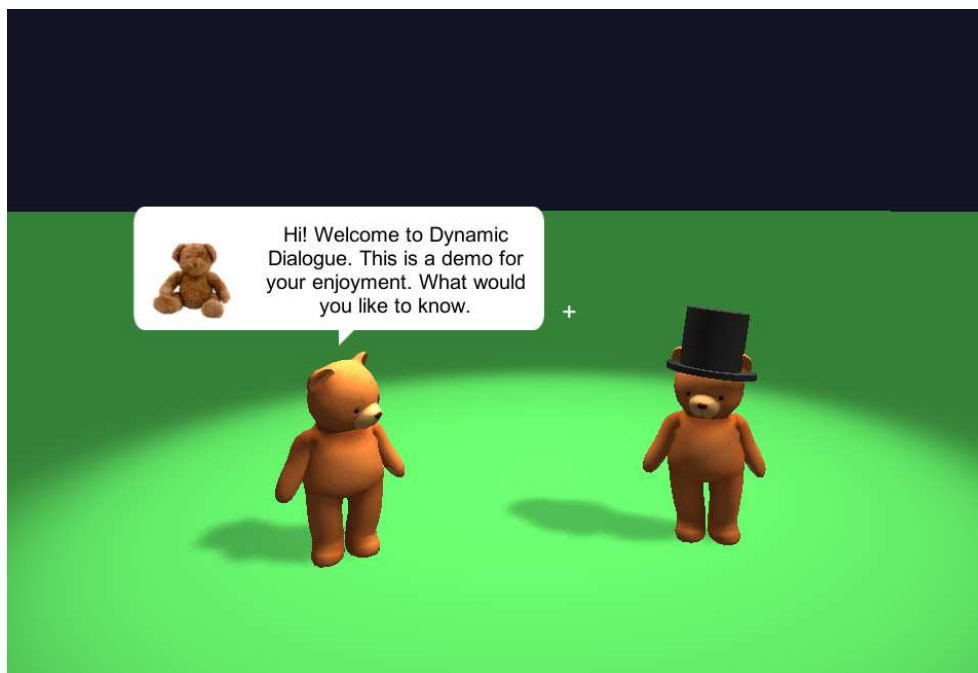


Dynamic Dialogue

What is Dynamic Dialogue?

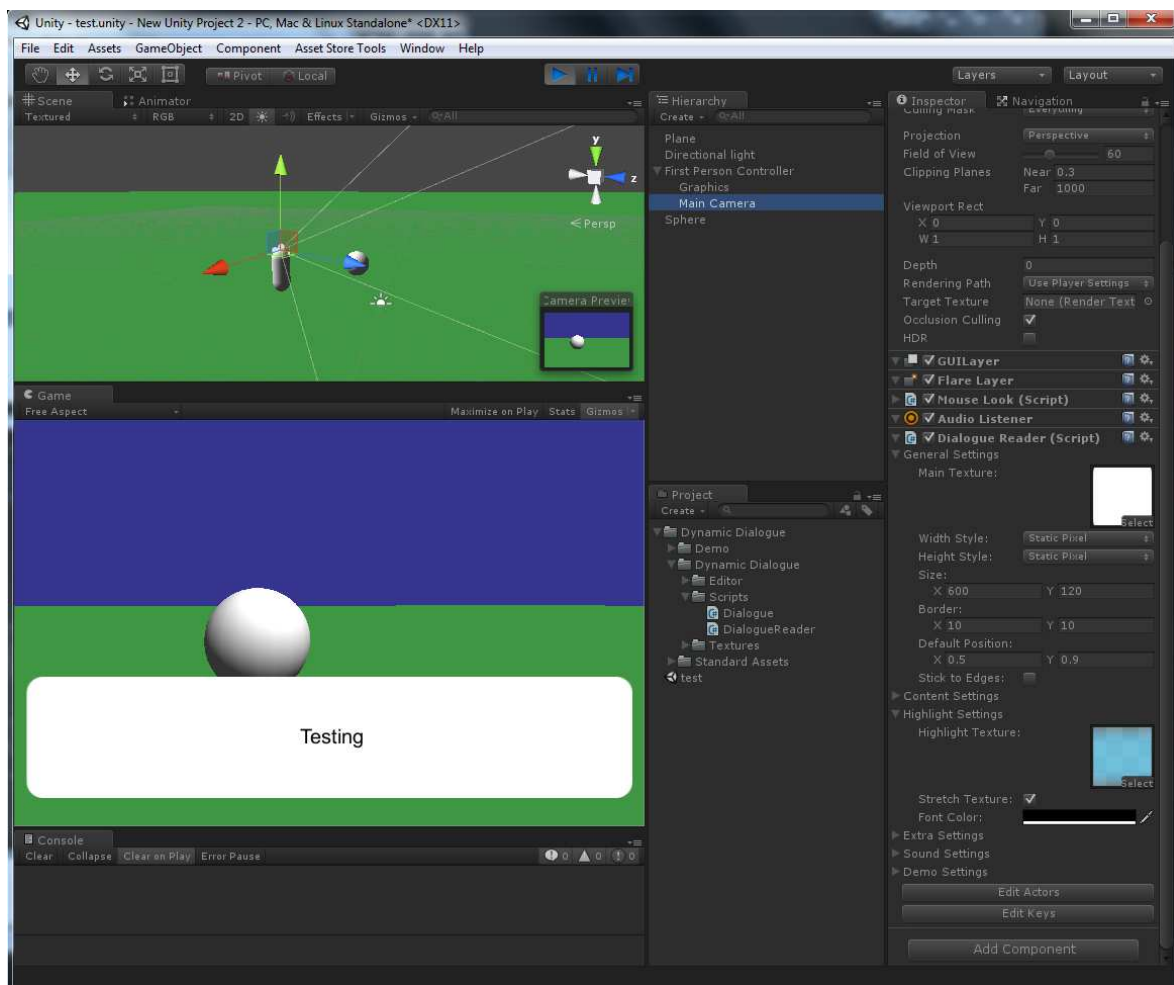
Dynamic Dialogue is a node based dialogue editor and display module used to create dialogue which is both easy to read and dynamic. For instance, the options available during dialogue can change depending on the player's answers to previous questions. This allows for a more interactive experience from the user's perspective when compared to traditional static dialogue systems.

Additionally, Dynamic Dialogue allows the developer to easily customize the appearance of dialogue in game for a unique look and feel. This includes anything from a traditional RPG layout to floating speech bubbles for a comic book feel.



Setting up a basic scene:

1. Import the latest version of Dynamic Dialogue to your project workspace in Unity. If you are starting a new project, create a basic scene with a light, a plane, and a character controller.
2. Add a DialogueReader script to an object in the scene, such as a camera or character controller. For first person games, it is suggested that DialogueReader be attached to the camera.
3. In the inspector, on the newly added script under general settings, add a main texture (try Textures/bubble). For this example, set the width and height style to Static Pixel and set the size to x 600 y 120. Set the border to x 10 y 10 and the default position to x 0.5 y 0.9.
4. Under highlight settings add a highlight texture (try Textures/highlight) and tick stretch texture.
5. Create at least one NPC to interact with. This can be any game object with a collider attached. Attach a 'Dialogue' script to this object. In the inspector click Edit Dialogue on the newly attached dialogue script. This will open up the Dialogue Editor. For now, right click anywhere on the grid and select Load Dialogue. Open Dynamic Dialogue/Demo/Dialogue/test.asset. Close the dialogue editor window.
6. Press play and test if the new scene works. By default, press 'E' to talk to the NPC when standing right on front of them.



Creating Dialogue

Now that the scene has been set up and everything is working correctly, it is time to focus on creating dialogue. Dialogue in Dynamic Dialogue is comprised of nodes. Each node has an actor and may contain a number of elements such as text, options, redirections and key actions. Each node is connected to other nodes by links. If no link is present, the dialogue will end. What follows is a description of how to create and use nodes, elements, and links to create dialogue.

Actors

Before creating nodes, actor/s must be created for the nodes to refer to. To do this select the attached DialogueReader script and in the inspector and then click the 'Edit Actors' button. This will open up the Actors editor. Click 'Create Actor' to create a new actor and change the name and other settings as required.

Actor Properties

Name – the name that will identify the actor and be displayed if display name is selected.

Position Type – static displays the dialogue in the same position every time. Dynamic displays the dialogue in relation to a transform in 3D space (used for floating speech bubbles).

Position – the basic position of static dialogue.

Offset – the amount in viewport space to offset a static position by. For instance, x 0.1 will offset the dialogue by 10% to the right.

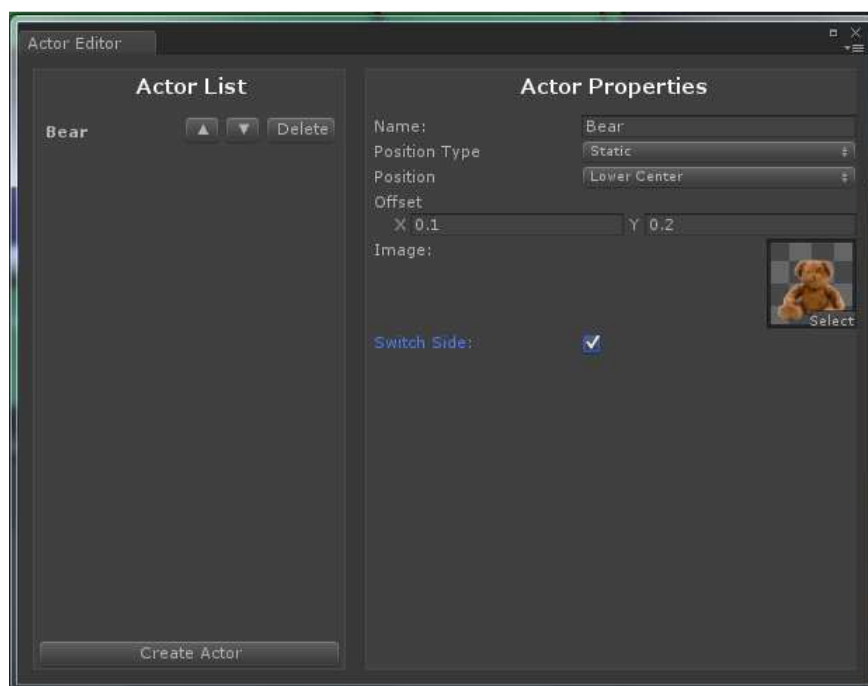
Image – an image that will be displayed if display image is selected.

Switch Side – changes the image to be displayed on the right side of the dialogue.

Transform – the transform to display the dialogue in relation to if position type is dynamic.

Auto Position – positions the dialogue at the exact top of the transform.

Offset – the amount in world space to offset the position by. For instance, y 1 will offset the dialogue by 1m in world space.

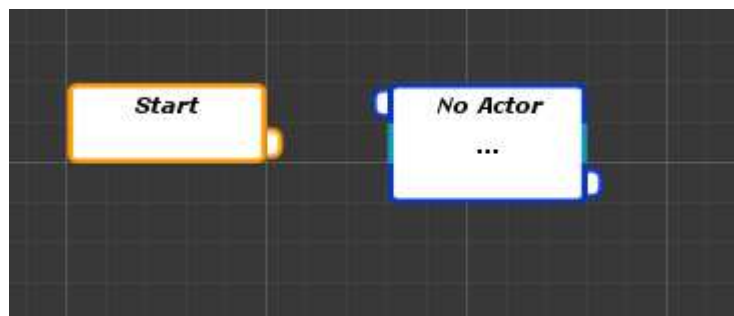


Nodes

Now that an actor has been defined, it is time to start creating the actual dialogue. To do this, go to any object with a 'Dialogue' script attached and click 'Edit Dialogue' to bring up the dialogue editor. To create a node, right click on the grid and select 'Create Node'. To delete a node, right click on the node and select 'Delete Node'.

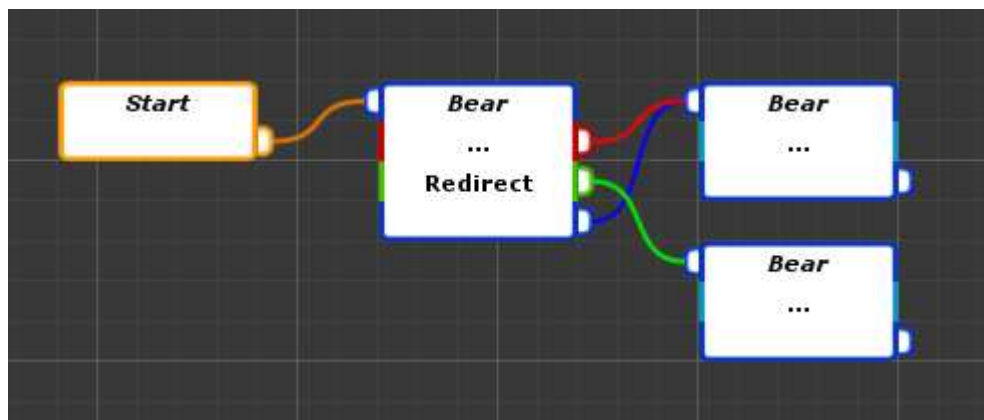
Once a node has been created it, if the main area of the node (within the dark blue lines) is left clicked it will become selected. When a node is selected the actor of that node can be changed in the properties panel to the right. All actors created in the Actor Editor will be displayed in the drop down actor list. To move a node, left click and drag the main area of a node to the desired location.

A special type of node, known as a Start Node, can be created only once per dialogue tree. A Start Node indicates where the dialogue reader starts to read the dialogue from.



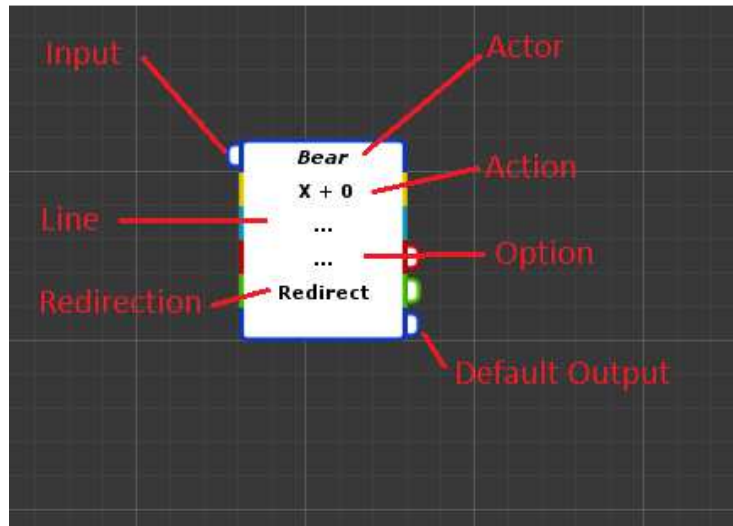
Node Links

Nodes can be linked together using node links. A node link connects an output from a node to the input of another node. This can be done simply by left clicking a node output and then left clicking again on the desired node input. Each node can have many outputs as option and redirection elements each have an output. A node only has one input.



Node Elements

To add elements to a node first right click the node, then select 'Add' to add a line, option, redirection or action element. Once an element has been created it can be selected by left clicking it. A selected element's properties will be displayed in the properties panel. To delete or duplicate an element, right click on that element and select the appropriate option.



Element Types

Line (light blue) –Displays text

Option (red) – Displays an option

Redirection (green) – Redirects dialogue to another node either automatically or manually.

Action (yellow) – Adds, subtracts or sets a key to a given value

Element Processing Order

Elements are processed during runtime in the same order they are displayed in the dialogue editor. The order is as follows:

1. Automatic Redirection
2. Action
3. Line
4. Option
5. Normal Redirection

This is important to remember as automatic redirections are processed before keys are set so an action will not affect an automatic redirection if they are on the same node. This is not true for an action and a line/option/normal redirection.

Keys

A key is a variable with a name and a single numeric value. Keys can be used to make dialogue dynamic. The value of a key can be set, manipulated or tested against. An Action element allows a given key to be added to, subtracted from, or set with a specific number.

The presence or quantity of a key can then be used to test whether an element should become active or not. To do this, each element has optional 'conditions' which can be added from the properties panel when that element is selected. If the condition is met during runtime, the element will become active, if it is not met, the element will be ignored.

For example, a key named 'welcome' could be set to 1 using an action element with a key of 'welcome' an action type of 'set' and the number '1'. A condition could then be set on a redirection element that automatically redirects the dialogue if welcome equals 1. If welcome does not equal 1, this redirection is ignored. The same principle can be applied to line, option and even action elements.



Note: In this example, the first time these nodes are displayed 'welcome' will equal 0 as it has not yet been set. Therefore the condition set on the redirection element will return false and the redirection will be ignored. However, as 'welcome' is set to 1 during the first run-through, the second time the dialogue is displayed the redirection condition will return true and the dialogue will be automatically redirected to the second node.

The initial value of keys can be set using the Key Editor. This can be found clicking 'Edit Keys' from the inspector when an object with the DialogueReader script is selected. Additionally, the Key Editor can be used to monitor keys at runtime within unity, a useful tool for debugging.

Demonstration

To demonstrate principles discussed so far, have a look at Dynamic Dialogue/Demo/Bubbles. Within the scene, click on Bear1 or Bear2 in the hierarchy and click 'Edit Dialogue'.

Customising Appearance

To customise the appearance of dialogue when using Dynamic Dialogue, click on the object that Dialogue Reader is attached to in the hierarchy. In the inspector a number of values can be manipulated in General Settings, Content Settings, Highlight Settings, Extra Settings and Sound Settings to stylize your dialogue. To get an idea of how these options can be used, take a look at the Bubbles and RPG demo scenes.

General Settings

Main Texture – the main background texture displayed behind the dialogue text

Width and Height Style – the width and height style

- Dynamic Pixel - recalculates based on content size, max size set in pixels
- Dynamic Percent - recalculates based on content size, max size set as a percent
- Static Pixel - a set pixel size not based on content
- Static Percent - a set screen percentage not based on content

Size - the actual pixel or percent size/max size depending on Width and Height Style

Border - the border surrounding the dialogue content

Default Position – the position of dialogue in viewport space (0 - 1) if no actor is given

Content Settings

Display Name – whether to display actor name

Display Image – whether to display actor image

Image Size – the size of the actor image in pixels

Text Indent – how much to indent line text by in pixels

Option Indent – how much to indent option text by in pixels

Alignment – how to align both line and option text

Font – the font used to display line and option text

Font Color – the color of the font

Font Size – the size of the font

Highlight Settings

Highlight Texture – the texture used to highlight selected options

Stretch Texture – whether to stretch texture over selected option text

Font Color – font color of selected option text

Extra Settings

Extra Texture – an optional texture used to give dialogue more pizzazz (e.g speech bubble point)

Extra Size – the size of the extra texture in pixels

Extra Offset – the position of the extra texture in relation to the main texture (0 - 1)

Sound Settings

Open Noise – the sound when the dialogue starts

Close Noise – the sound when the dialogue ends

Next Noise – the sound when the next line of dialogue is displayed

Option Noise – the sound when an option is selected

Interacting with Dynamic Dialogue

Demo Settings

By default, demo settings used to interact with dynamic dialogue are turned on. These settings can be found under DialogueReader/Demo Settings. It should be noted that this is only an example of how to interact with Dynamic Dialogue. For a more customized experience, please use the programming interface bellow. To see how these settings are used, have a look at the Bubbles or RPG demo scenes.

Demo – whether to use the demo code to interact with Dynamic Dialogue

Demo Talk Distance – the distance the player is able to start dialogue from

Demo Talk Width – the width from the centre of the screen that dialogue can be started from

Demo GUIText – GUIText used to indicate when dialogue is within the specified distance/width

Demo GUIColor – the color the GUIText turns when in range of dialogue

Custom Keys – the input keys used for interacting with Dynamic Dialogue

Programming Interface

DialogueReader Class

void OpenConversation(Dialogue dialogue)

- Opens conversation with a given dialogue class (usually attached to an actor/object)

void CloseConversation()

- Closes current conversation

void Selection(string direction)

- Used to change selected option
- Example: Selection("up");
 Selection("down");

void NextNode()

- Moves to next node

Note: for an example of how to use these methods, see Update() within DialogueReader.cs