Dialogue System for Unity

How To Create A Tutorial Conversation

To use a conversation as a tutorial, leave the player's control enabled. To do this, you can disable the player's SetComponentEnabledOnDialogueEvent component or remove the player control components from its OnStart and OnEnd sections. This way the player can continue to move and do things during the conversation.

Since the player will probably have control of the camera, you should not use Camera() sequencer commands. Make sure that the Dialogue Manager's Default Sequence also doesn't use Camera() commands. You can temporarily override the Default Sequence and other display settings by adding an OverrideDisplaySettings component to the GameObject that triggers the tutorial (i.e., the “conversant” in this conversation).

The included example scene demonstrates how to wait for the player to pick up blocks and trigger QTEs.

The dialogue entry "Pick up a red block" stays on the screen until the player picks up a red block. The dialogue entry's Sequence uses the @Message() syntax. This makes the sequence wait until it receives a message from Unity. In the case of the blocks, a short script named Pickup.cs sends the message when the player walks over the block. The @Message() syntax is described in more detail here:

http://www.pixelcrushers.com/dialogue_system/manual/html/sequencer_commands.html#sequenceCommandExampleMessage

Briefly, this is how it's set up:

Dialogue Entry
Dialogue Text: "Pick up a red block"
Sequence: None()@Message(GotRedBlock)
Links To: [Dialogue Entry "Good job. Now get a blue block."]

Pickup Script

```
void PlayerTakesRock() {
    Sequencer.Message(“GotRedBlock”);
}
```

When the conversation gets to "Pick up a red block", it displays the text and runs the sequence. The sequence waits until it receives a message “GotRedBlock”.