

Dialogue Lang

v0.1

Generated by Doxygen 1.9.8

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 Dialogue Class Reference	7
4.1.1 Member Function Documentation	9
4.1.1.1 DisplayConversation()	9
4.1.1.2 DisplayPlayerChoices()	9
4.1.1.3 EvaluateCondition()	10
4.1.1.4 ExitDialogue()	10
4.1.1.5 FollowRedirects()	11
4.1.1.6 GetDialogueVariable()	11
4.1.1.7 GetDialogueVariableAsInt()	11
4.1.1.8 Interact()	13
4.1.1.9 LoadTextAsset()	13
4.1.1.10 ParseDialogueFile()	14
4.1.1.11 ParseVariablesAndInitializeSections()	14
4.1.1.12 ProcessDialogueSectionsAndChoices()	15
4.1.1.13 SetDialogueVariable()	15
4.1.1.14 SplitDialogueLines()	15
4.1.1.15 Start()	16
4.1.1.16 Update()	16
4.1.2 Member Data Documentation	16
4.1.2.1 buttonList	16
4.1.2.2 choiceButtonPrefab	16
4.1.2.3 choiceButtons	16
4.1.2.4 cursor_previousLockMode	16
4.1.2.5 cursor_previousVisible	16
4.1.2.6 dialogue	17
4.1.2.7 dialogue_dict	17
4.1.2.8 dialogue_file_name	17
4.1.2.9 dialogue_npcName	17
4.1.2.10 dialogue_npcText	17
4.1.2.11 dm	17
4.1.2.12 isInDialogue	17
4.1.2.13 letterDelay	17
4.1.2.14 letterSpeedMultiplier	17

4.1.2.15 npc_display_name	17
4.1.2.16 onInvoke	18
4.1.2.17 sentenceDelay	18
4.1.2.18 uniformsChanged	18
4.1.2.19 uniformVariables	18
4.1.2.20 uniformVariablesAlreadyDefined	18
4.2 DialogueChoice Class Reference	18
4.2.1 Detailed Description	18
4.2.2 Property Documentation	19
4.2.2.1 NextHeader	19
4.2.2.2 Text	19
4.3 DialogueManager Class Reference	19
4.3.1 Member Function Documentation	20
4.3.1.1 Awake()	20
4.3.1.2 GetInstance()	20
4.3.2 Member Data Documentation	20
4.3.2.1 choiceButtonPrefab	20
4.3.2.2 choiceButtonsObject	20
4.3.2.3 dialogueObject	20
4.3.2.4 dialogueText	20
4.3.2.5 instance	20
4.3.2.6 letterDelay	20
4.3.2.7 letterSpeedMultiplier	20
4.3.2.8 npcNameText	21
4.3.2.9 sentenceDelay	21
4.4 DialogueSection Class Reference	21
4.4.1 Detailed Description	21
4.4.2 Property Documentation	21
4.4.2.1 Choices	21
4.4.2.2 Header	21
4.4.2.3 Lines	21
4.4.2.4 Redirect	22
4.5 UniformVariables Class Reference	22
4.5.1 Detailed Description	22
4.5.2 Constructor & Destructor Documentation	22
4.5.2.1 UniformVariables()	22
4.5.3 Member Function Documentation	22
4.5.3.1 GetVariable()	22
4.5.3.2 SetVariable()	22
4.5.4 Property Documentation	22
4.5.4.1 Variables	22

5 File Documentation	23
5.1 Dialogue Lang/Assets/Scripts/Dialogue.cs File Reference	23
5.2 Dialogue Lang/Assets/Scripts/DialogueManager.cs File Reference	23
Index	25

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

DialogueChoice	18
DialogueSection	21
MonoBehaviour	
Dialogue	7
DialogueManager	19
UniformVariables	22

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Dialogue	7
DialogueChoice	
Represents a choice in a dialogue system. This class is used to store information about a specific choice within a dialogue, including the text displayed to the user and the identifier of the next dialogue header to navigate to when this choice is selected	18
DialogueManager	19
DialogueSection	
Represents a section of a dialogue in a narrative or dialogue system. This class contains the content of a dialogue section, including the header (identifier), any redirection to other dialogue sections, the lines of dialogue, and the choices available to the user at the end of the section	21
UniformVariables	
Represents a collection of uniform variables. This class is used to store and manage a set of variables where each variable is identified by a string key. It provides methods to set and retrieve variable values	22

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

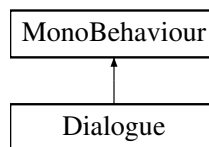
Dialogue Lang/Assets/Scripts/ Dialogue.cs	23
Dialogue Lang/Assets/Scripts/ DialogueManager.cs	23

Chapter 4

Class Documentation

4.1 Dialogue Class Reference

Inheritance diagram for Dialogue:



Public Member Functions

- void [SetDialogueVariable](#) (string key, object value)
Sets a variable in the dialogue system. This method assigns a value to a specified key in the uniform variables collection. It also marks that the uniforms have changed.
- object [GetDialogueVariable](#) (string key)
Retrieves a variable from the dialogue system. This method returns the value of a specified key from the uniform variables collection.
- int [GetDialogueVariableAsInt](#) (string key)
Retrieves a variable from the dialogue system and converts it to an integer. This method attempts to convert the value of a specified key from the uniform variables collection to an integer. If the conversion is not possible, an exception is thrown.
- void [Interact](#) ()
Handles the interaction with an NPC (non-player character) in a dialogue system. This method initiates a dialogue if the player is not already in one. It sets up the dialogue environment, including cursor state and dialogue UI elements. It also parses and displays the conversation from the dialogue file.

Public Attributes

- string [npc_display_name](#)
- string [dialogue_file_name](#)
- UnityEvent [onInvoke](#)

Static Public Attributes

- static bool `isInDialogue` = false

Private Member Functions

- void `Start` ()
- void `Update` ()
- `DialogueSection FollowRedirects` (`DialogueSection` section)

Resolves any redirects within a dialogue section to find the final section to be displayed. This method recursively follows the redirect chain in the dialogue until it reaches a section without a redirect.
- IEnumerator `DisplayConversation` (`DialogueSection` ds)

Coroutine for displaying a conversation in the dialogue system. This method sequentially displays each sentence in the dialogue section, simulating the typing effect with a delay between letters and sentences. After displaying all sentences, it presents the player choices.
- void `DisplayPlayerChoices` (List< `DialogueChoice` > choices)

Coroutine for displaying a conversation in a dialogue system. It sequentially shows each sentence in the dialogue section, giving a typewriter-like effect. After displaying all sentences, it shows player choices.
- void `ExitDialogue` ()

Displays the player choices at the end of a dialogue section. This method generates interactive buttons for each choice, allowing the player to select their response or action.
- Dictionary< string, `DialogueSection` > `ParseDialogueFile` (string resourceName)

Parses a dialogue file and constructs a dictionary of dialogue sections. This method reads and interprets the contents of a dialogue file, organizing it into a structured format that can be used in the dialogue system.
- void `ParseVariablesAndInitializeSections` (string[] lines, Dictionary< string, `DialogueSection` > dialogue↵Sections, `DialogueSection` currentSection)

Parses the variables and initializes dialogue sections from the provided lines of dialogue. This method is responsible for the first pass of dialogue parsing, setting up headers and variables.
- void `ProcessDialogueSectionsAndChoices` (string[] lines, Dictionary< string, `DialogueSection` > dialogue↵Sections, ref bool lastIfConditionMet, `DialogueSection` currentSection)

Processes the dialogue sections and choices based on the lines of dialogue. This method is responsible for the second pass of dialogue parsing, handling dialogue text, sections, and choices.
- string[] `SplitDialogueLines` (TextAsset textAsset)

Splits the content of a TextAsset into individual lines for processing.
- TextAsset `LoadTextAsset` (string resourceName)

Loads a TextAsset resource by its name.
- bool `EvaluateCondition` (string condition)

Evaluates a condition expressed as a string, comparing a variable from the uniform variables against a value.

Private Attributes

- Dictionary< string, `DialogueSection` > `dialogue_dict`
- `DialogueManager` dm
- GameObject `choiceButtonPrefab`
- Transform `choiceButtons`
- Transform `dialogue`
- TMP_Text `dialogue_npcText`
- TMP_Text `dialogue_npcName`
- float `letterDelay`
- float `sentenceDelay`
- float `letterSpeedMultiplier`
- List< GameObject > `buttonList` = new List<GameObject>()
- `UniformVariables` uniformVariables = new `UniformVariables`()
- bool `uniformsChanged` = false
- bool `uniformVariablesAlreadyDefined` = false
- CursorLockMode `cursor_previousLockMode`
- bool `cursor_previousVisible`

4.1.1 Member Function Documentation

4.1.1.1 DisplayConversation()

```
IEnumerator Dialogue.DisplayConversation (
    DialogueSection ds ) [private]
```

Coroutine for displaying a conversation in the dialogue system. This method sequentially displays each sentence in the dialogue section, simulating the typing effect with a delay between letters and sentences. After displaying all sentences, it presents the player choices.

Parameters

<i>ds</i>	The dialogue section to display.
-----------	----------------------------------

Returns

An enumerator needed for the coroutine execution.

The method performs the following steps:

- Clears and destroys any existing choice buttons.
- Sets the dialogue UI elements to active.
- Follows any redirects to find the final dialogue section to display.
- Iterates through each sentence in the dialogue section, displaying them one character at a time.
- After displaying each sentence, there's a delay before moving to the next one.
- Once all sentences are displayed, it displays the player choices for the current section.

4.1.1.2 DisplayPlayerChoices()

```
void Dialogue.DisplayPlayerChoices (
    List< DialogueChoice > choices ) [private]
```

Coroutine for displaying a conversation in a dialogue system. It sequentially shows each sentence in the dialogue section, giving a typewriter-like effect. After displaying all sentences, it shows player choices.

Parameters

<i>ds</i>	The dialogue section containing the sentences and choices to be displayed.
-----------	--

Returns

An IEnumerator required for coroutine execution in Unity.

The coroutine performs the following steps:

- Clears existing choice buttons from the UI.
- Activates necessary UI elements for displaying dialogue and choices.
- Checks and follows any redirects in the dialogue section to ensure the correct section is displayed.
- Iterates through each sentence in the dialogue section, adding characters one by one to simulate typing, with a delay defined by 'letterDelay'.
- After each sentence, waits for a duration defined by 'sentenceDelay' before continuing to the next sentence.
- Once all sentences are displayed, it invokes 'DisplayPlayerChoices' to show choices to the player based on the dialogue section.

4.1.1.3 EvaluateCondition()

```
bool Dialogue.EvaluateCondition (
    string condition ) [private]
```

Evaluates a condition expressed as a string, comparing a variable from the uniform variables against a value.

Parameters

<i>condition</i>	The condition to evaluate, expressed in a format like "variable operator value".
------------------	--

Returns

True if the condition is met, false otherwise.

This method supports various operators (==, !=, <, <=, >, >=) and handles both equality/inequality and numerical comparisons. It logs an error if non-integer values are used for numerical comparisons.

4.1.1.4 ExitDialogue()

```
void Dialogue.ExitDialogue ( ) [private]
```

Displays the player choices at the end of a dialogue section. This method generates interactive buttons for each choice, allowing the player to select their response or action.

Parameters

<i>choices</i>	A list of DialogueChoice objects representing the choices available to the player.
----------------	--

The method performs the following steps:

- Activates the UI element for displaying choices and deactivates the NPC dialogue text element.
- Iterates through each choice in the provided list, creating a button for each one.
- Sets the text of each button to match the text of the dialogue choice.
- Adds an event listener to each button. The listener behavior depends on the 'NextHeader' property of the choice:

- If 'NextHeader' is "EXIT", the button will end the dialogue.
 - If 'NextHeader' is "INVOKE", the button will end the dialogue and invoke a specific event (defined by 'onInvoke').
 - Otherwise, the button will trigger the display of the next dialogue section indicated by 'NextHeader'.
- Adds the newly created buttons to the UI and stores them in 'buttonList' for management.

4.1.1.5 FollowRedirects()

```
DialogueSection Dialogue.FollowRedirects (
    DialogueSection section ) [private]
```

Resolves any redirects within a dialogue section to find the final section to be displayed. This method recursively follows the redirect chain in the dialogue until it reaches a section without a redirect.

Parameters

<i>section</i>	The initial dialogue section from which to start following redirects.
----------------	---

Returns

The final dialogue section after resolving all redirects.

This method checks if the provided section has a redirect. If it does, and the redirect points to a valid section key in the dialogue dictionary, the method calls itself recursively with the new section. This process repeats until a section without a redirect is reached, which is then returned as the final section to display.

4.1.1.6 GetDialogueVariable()

```
object Dialogue.GetDialogueVariable (
    string key )
```

Retrieves a variable from the dialogue system. This method returns the value of a specified key from the uniform variables collection.

Parameters

<i>key</i>	The key of the variable to retrieve.
------------	--------------------------------------

Returns

The value of the variable if it exists; otherwise, null.

4.1.1.7 GetDialogueVariableAsInt()

```
int Dialogue.GetDialogueVariableAsInt (
    string key )
```

Retrieves a variable from the dialogue system and converts it to an integer. This method attempts to convert the value of a specified key from the uniform variables collection to an integer. If the conversion is not possible, an exception is thrown.

Parameters

<i>key</i>	The key of the variable to retrieve and convert.
------------	--

Returns

The integer value of the variable.

Exceptions

<i>ArgumentException</i>	Thrown when the value cannot be converted to an integer.
--------------------------	--

4.1.1.8 Interact()

```
void Dialogue.Interact ( )
```

Handles the interaction with an NPC (non-player character) in a dialogue system. This method initiates a dialogue if the player is not already in one. It sets up the dialogue environment, including cursor state and dialogue UI elements. It also parses and displays the conversation from the dialogue file.

The method performs several steps:

- Checks if the player is already in a dialogue; if so, it returns immediately.
- Sets the 'isInDialogue' flag to true to indicate the start of a dialogue.
- Sets up the dialogue UI with the NPC's name and clears any existing text.
- Saves the current cursor state and updates the cursor for dialogue interaction.
- Parses the dialogue file if there were changes to the uniform variables since the last interaction.
- Retrieves and displays the initial section of the dialogue.

4.1.1.9 LoadTextAsset()

```
TextAsset Dialogue.LoadTextAsset (
    string resourceName ) [private]
```

Loads a TextAsset resource by its name.

Parameters

<i>resourceName</i>	The name of the resource to load.
---------------------	-----------------------------------

Returns

The loaded TextAsset, or null if the resource cannot be found.

This method attempts to load a TextAsset using Unity's Resources.Load method. It logs an error if the resource cannot be loaded.

4.1.1.10 ParseDialogueFile()

```
Dictionary< string, DialogueSection > Dialogue.ParseDialogueFile (
    string resourceName ) [private]
```

Parses a dialogue file and constructs a dictionary of dialogue sections. This method reads and interprets the contents of a dialogue file, organizing it into a structured format that can be used in the dialogue system.

Parameters

<i>resourceName</i>	The name of the resource file containing the dialogue.
---------------------	--

Returns

A dictionary where keys are section headers and values are [DialogueSection](#) objects, or null if the text asset cannot be loaded.

The method performs the following steps:

- Loads the text asset using the provided resource name.
- Splits the text asset into individual lines of dialogue.
- Initializes a new dictionary to hold dialogue sections.
- Processes each line to build up dialogue sections, including handling conditions and choices.
- Adds the final dialogue section to the dictionary if it exists.
- Returns the constructed dictionary of dialogue sections, or null if the resource cannot be found or loaded.

4.1.1.11 ParseVariablesAndInitializeSections()

```
void Dialogue.ParseVariablesAndInitializeSections (
    string[] lines,
    Dictionary< string, DialogueSection > dialogueSections,
    DialogueSection currentSection ) [private]
```

Parses the variables and initializes dialogue sections from the provided lines of dialogue. This method is responsible for the first pass of dialogue parsing, setting up headers and variables.

Parameters

<i>lines</i>	The lines of dialogue to be parsed.
<i>dialogueSections</i>	A dictionary to store the initialized dialogue sections.
<i>currentSection</i>	The current dialogue section being processed.

The method performs the following actions:

- Iterates through each line, trimming leading and trailing whitespace.
- Identifies and initializes dialogue section headers.

- Parses and sets variables that haven't been defined before.

4.1.1.12 ProcessDialogueSectionsAndChoices()

```
void Dialogue.ProcessDialogueSectionsAndChoices (
    string[] lines,
    Dictionary< string, DialogueSection > dialogueSections,
    ref bool lastIfConditionMet,
    DialogueSection currentSection ) [private]
```

Processes the dialogue sections and choices based on the lines of dialogue. This method is responsible for the second pass of dialogue parsing, handling dialogue text, sections, and choices.

Parameters

<i>lines</i>	The lines of dialogue to be processed.
<i>dialogueSections</i>	A dictionary storing dialogue sections.
<i>lastIfConditionMet</i>	A reference to a boolean flag indicating if the last 'if' condition was met.
<i>currentSection</i>	The current dialogue section being processed.

The method performs the following actions:

- Skips variable definitions processed in the first pass.
- Processes headers, including saving the last section and setting the current section.
- Handles conditional headers and redirects.
- Parses choices, including conditional choices, and adds them to the current section.
- Adds normal dialogue lines to the current section.

4.1.1.13 SetDialogueVariable()

```
void Dialogue.SetDialogueVariable (
    string key,
    object value )
```

Sets a variable in the dialogue system. This method assigns a value to a specified key in the uniform variables collection. It also marks that the uniforms have changed.

Parameters

<i>key</i>	The key of the variable to set.
<i>value</i>	The value to assign to the variable.

4.1.1.14 SplitDialogueLines()

```
string[] Dialogue.SplitDialogueLines (
```

```
TextAsset textAsset ) [private]
```

Splits the content of a TextAsset into individual lines for processing.

Parameters

<i>textAsset</i>	The TextAsset containing the dialogue text.
------------------	---

Returns

An array of strings, each representing a line in the TextAsset.

This method splits the text based on carriage returns and newlines, removing empty entries. It's used to prepare dialogue text for further parsing.

4.1.1.15 Start()

```
void Dialogue.Start ( ) [private]
```

4.1.1.16 Update()

```
void Dialogue.Update ( ) [private]
```

4.1.2 Member Data Documentation

4.1.2.1 buttonList

```
List<GameObject> Dialogue.buttonList = new List<GameObject>() [private]
```

4.1.2.2 choiceButtonPrefab

```
GameObject Dialogue.choiceButtonPrefab [private]
```

4.1.2.3 choiceButtons

```
Transform Dialogue.choiceButtons [private]
```

4.1.2.4 cursor_previousLockMode

```
CursorLockMode Dialogue.cursor_previousLockMode [private]
```

4.1.2.5 cursor_previousVisible

```
bool Dialogue.cursor_previousVisible [private]
```

4.1.2.6 dialogue

```
Transform Dialogue.dialogue [private]
```

4.1.2.7 dialogue_dict

```
Dictionary<string, DialogueSection> Dialogue.dialogue_dict [private]
```

4.1.2.8 dialogue_file_name

```
string Dialogue.dialogue_file_name
```

4.1.2.9 dialogue_npcName

```
TMP_Text Dialogue.dialogue_npcName [private]
```

4.1.2.10 dialogue_npcText

```
TMP_Text Dialogue.dialogue_npcText [private]
```

4.1.2.11 dm

```
DialogueManager Dialogue.dm [private]
```

4.1.2.12 isInDialogue

```
bool Dialogue.isInDialogue = false [static]
```

4.1.2.13 letterDelay

```
float Dialogue.letterDelay [private]
```

4.1.2.14 letterSpeedMultiplier

```
float Dialogue.letterSpeedMultiplier [private]
```

4.1.2.15 npc_display_name

```
string Dialogue.npc_display_name
```

4.1.2.16 onInvoke

```
UnityEvent Dialogue.onInvoke
```

4.1.2.17 sentenceDelay

```
float Dialogue.sentenceDelay [private]
```

4.1.2.18 uniformsChanged

```
bool Dialogue.uniformsChanged = false [private]
```

4.1.2.19 uniformVariables

```
UniformVariables Dialogue.uniformVariables = new UniformVariables() [private]
```

4.1.2.20 uniformVariablesAlreadyDefined

```
bool Dialogue.uniformVariablesAlreadyDefined = false [private]
```

The documentation for this class was generated from the following file:

- Dialogue Lang/Assets/Scripts/[Dialogue.cs](#)

4.2 DialogueChoice Class Reference

Represents a choice in a dialogue system. This class is used to store information about a specific choice within a dialogue, including the text displayed to the user and the identifier of the next dialogue header to navigate to when this choice is selected.

Properties

- string [Text](#) [get, set]
- string [NextHeader](#) [get, set]

4.2.1 Detailed Description

Represents a choice in a dialogue system. This class is used to store information about a specific choice within a dialogue, including the text displayed to the user and the identifier of the next dialogue header to navigate to when this choice is selected.

4.2.2 Property Documentation

4.2.2.1 NextHeader

```
string DialogueChoice.NextHeader [get], [set]
```

4.2.2.2 Text

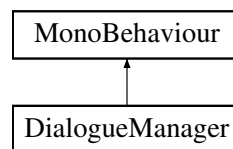
```
string DialogueChoice.Text [get], [set]
```

The documentation for this class was generated from the following file:

- Dialogue Lang/Assets/Scripts/[Dialogue.cs](#)

4.3 DialogueManager Class Reference

Inheritance diagram for DialogueManager:



Static Public Member Functions

- static [DialogueManager](#) [GetInstance](#) ()

Public Attributes

- GameObject [choiceButtonPrefab](#)
- float [letterDelay](#) = 0.05f
- float [sentenceDelay](#) = 1.0f
- float [letterSpeedMultiplier](#) = 0.5f
- Transform [dialogueObject](#)
- Transform [choiceButtonsObject](#)
- TMP_Text [dialogueText](#)
- TMP_Text [npcNameText](#)

Private Member Functions

- void [Awake](#) ()

Static Private Attributes

- static [DialogueManager](#) [instance](#)

4.3.1 Member Function Documentation

4.3.1.1 Awake()

```
void DialogueManager.Awake ( ) [private]
```

4.3.1.2 GetInstance()

```
static DialogueManager DialogueManager.GetInstance ( ) [static]
```

4.3.2 Member Data Documentation

4.3.2.1 choiceButtonPrefab

```
GameObject DialogueManager.choiceButtonPrefab
```

4.3.2.2 choiceButtonsObject

```
Transform DialogueManager.choiceButtonsObject
```

4.3.2.3 dialogueObject

```
Transform DialogueManager.dialogueObject
```

4.3.2.4 dialogueText

```
TMP_Text DialogueManager.dialogueText
```

4.3.2.5 instance

```
DialogueManager DialogueManager.instance [static], [private]
```

4.3.2.6 letterDelay

```
float DialogueManager.letterDelay = 0.05f
```

4.3.2.7 letterSpeedMultiplier

```
float DialogueManager.letterSpeedMultiplier = 0.5f
```

4.3.2.8 npcNameText

```
TMP_Text DialogueManager.npcNameText
```

4.3.2.9 sentenceDelay

```
float DialogueManager.sentenceDelay = 1.0f
```

The documentation for this class was generated from the following file:

- Dialogue Lang/Assets/Scripts/[DialogueManager.cs](#)

4.4 DialogueSection Class Reference

Represents a section of a dialogue in a narrative or dialogue system. This class contains the content of a dialogue section, including the header (identifier), any redirection to other dialogue sections, the lines of dialogue, and the choices available to the user at the end of the section.

Properties

- string [Header](#) [get, set]
- string [Redirect](#) [get, set]
- List< string > [Lines](#) = new List<string>() [get, set]
- List< [DialogueChoice](#) > [Choices](#) = new List<[DialogueChoice](#)>() [get, set]

4.4.1 Detailed Description

Represents a section of a dialogue in a narrative or dialogue system. This class contains the content of a dialogue section, including the header (identifier), any redirection to other dialogue sections, the lines of dialogue, and the choices available to the user at the end of the section.

4.4.2 Property Documentation

4.4.2.1 Choices

```
List<DialogueChoice> DialogueSection.Choices = new List<DialogueChoice>() [get], [set]
```

4.4.2.2 Header

```
string DialogueSection.Header [get], [set]
```

4.4.2.3 Lines

```
List<string> DialogueSection.Lines = new List<string>() [get], [set]
```

4.4.2.4 Redirect

```
string DialogueSection.Redirect [get], [set]
```

The documentation for this class was generated from the following file:

- Dialogue Lang/Assets/Scripts/[Dialogue.cs](#)

4.5 UniformVariables Class Reference

Represents a collection of uniform variables. This class is used to store and manage a set of variables where each variable is identified by a string key. It provides methods to set and retrieve variable values.

Public Member Functions

- [UniformVariables](#) ()
- void [SetVariable](#) (string key, object value)
- object [GetVariable](#) (string key)

Properties

- Dictionary< string, object > [Variables](#) [get, private set]

4.5.1 Detailed Description

Represents a collection of uniform variables. This class is used to store and manage a set of variables where each variable is identified by a string key. It provides methods to set and retrieve variable values.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 UniformVariables()

```
UniformVariables.UniformVariables ( )
```

4.5.3 Member Function Documentation

4.5.3.1 GetVariable()

```
object UniformVariables.GetVariable (
    string key )
```

4.5.3.2 SetVariable()

```
void UniformVariables.SetVariable (
    string key,
    object value )
```

4.5.4 Property Documentation

4.5.4.1 Variables

```
Dictionary<string, object> UniformVariables.Variables [get], [private set]
```

The documentation for this class was generated from the following file:

- Dialogue Lang/Assets/Scripts/[Dialogue.cs](#)

Chapter 5

File Documentation

5.1 Dialogue Lang/Assets/Scripts/Dialogue.cs File Reference

Classes

- class [UniformVariables](#)
Represents a collection of uniform variables. This class is used to store and manage a set of variables where each variable is identified by a string key. It provides methods to set and retrieve variable values.
- class [DialogueChoice](#)
Represents a choice in a dialogue system. This class is used to store information about a specific choice within a dialogue, including the text displayed to the user and the identifier of the next dialogue header to navigate to when this choice is selected.
- class [DialogueSection](#)
Represents a section of a dialogue in a narrative or dialogue system. This class contains the content of a dialogue section, including the header (identifier), any redirection to other dialogue sections, the lines of dialogue, and the choices available to the user at the end of the section.
- class [Dialogue](#)

5.2 Dialogue Lang/Assets/Scripts/DialogueManager.cs File Reference

Classes

- class [DialogueManager](#)

Index

- Awake
 - DialogueManager, [20](#)
- buttonList
 - Dialogue, [16](#)
- choiceButtonPrefab
 - Dialogue, [16](#)
 - DialogueManager, [20](#)
- choiceButtons
 - Dialogue, [16](#)
- choiceButtonsObject
 - DialogueManager, [20](#)
- Choices
 - DialogueSection, [21](#)
- cursor_previousLockMode
 - Dialogue, [16](#)
- cursor_previousVisible
 - Dialogue, [16](#)
- Dialogue, [7](#)
 - buttonList, [16](#)
 - choiceButtonPrefab, [16](#)
 - choiceButtons, [16](#)
 - cursor_previousLockMode, [16](#)
 - cursor_previousVisible, [16](#)
 - dialogue, [16](#)
 - dialogue_dict, [17](#)
 - dialogue_file_name, [17](#)
 - dialogue_npcName, [17](#)
 - dialogue_npcText, [17](#)
 - DisplayConversation, [9](#)
 - DisplayPlayerChoices, [9](#)
 - dm, [17](#)
 - EvaluateCondition, [10](#)
 - ExitDialogue, [10](#)
 - FollowRedirects, [11](#)
 - GetDialogueVariable, [11](#)
 - GetDialogueVariableAsInt, [11](#)
 - Interact, [13](#)
 - isInDialogue, [17](#)
 - letterDelay, [17](#)
 - letterSpeedMultiplier, [17](#)
 - LoadTextAsset, [13](#)
 - npc_display_name, [17](#)
 - onInvoke, [17](#)
 - ParseDialogueFile, [13](#)
 - ParseVariablesAndInitializeSections, [14](#)
 - ProcessDialogueSectionsAndChoices, [15](#)
 - sentenceDelay, [18](#)
 - SetDialogueVariable, [15](#)
 - SplitDialogueLines, [15](#)
 - Start, [16](#)
 - uniformsChanged, [18](#)
 - uniformVariables, [18](#)
 - uniformVariablesAlreadyDefined, [18](#)
 - Update, [16](#)
- dialogue
 - Dialogue, [16](#)
- Dialogue Lang/Assets/Scripts/Dialogue.cs, [23](#)
- Dialogue Lang/Assets/Scripts/DialogueManager.cs, [23](#)
- dialogue_dict
 - Dialogue, [17](#)
- dialogue_file_name
 - Dialogue, [17](#)
- dialogue_npcName
 - Dialogue, [17](#)
- dialogue_npcText
 - Dialogue, [17](#)
- DialogueChoice, [18](#)
 - NextHeader, [19](#)
 - Text, [19](#)
- DialogueManager, [19](#)
 - Awake, [20](#)
 - choiceButtonPrefab, [20](#)
 - choiceButtonsObject, [20](#)
 - dialogueObject, [20](#)
 - dialogueText, [20](#)
 - GetInstance, [20](#)
 - instance, [20](#)
 - letterDelay, [20](#)
 - letterSpeedMultiplier, [20](#)
 - npcNameText, [20](#)
 - sentenceDelay, [21](#)
- dialogueObject
 - DialogueManager, [20](#)
- DialogueSection, [21](#)
 - Choices, [21](#)
 - Header, [21](#)
 - Lines, [21](#)
 - Redirect, [21](#)
- dialogueText
 - DialogueManager, [20](#)
- DisplayConversation
 - Dialogue, [9](#)
- DisplayPlayerChoices
 - Dialogue, [9](#)
- dm
 - Dialogue, [17](#)

- EvaluateCondition
 - Dialogue, [10](#)
- ExitDialogue
 - Dialogue, [10](#)
- FollowRedirects
 - Dialogue, [11](#)
- GetDialogueVariable
 - Dialogue, [11](#)
- GetDialogueVariableAsInt
 - Dialogue, [11](#)
- GetInstance
 - DialogueManager, [20](#)
- GetVariable
 - UniformVariables, [22](#)
- Header
 - DialogueSection, [21](#)
- instance
 - DialogueManager, [20](#)
- Interact
 - Dialogue, [13](#)
- isInDialogue
 - Dialogue, [17](#)
- letterDelay
 - Dialogue, [17](#)
 - DialogueManager, [20](#)
- letterSpeedMultiplier
 - Dialogue, [17](#)
 - DialogueManager, [20](#)
- Lines
 - DialogueSection, [21](#)
- LoadTextAsset
 - Dialogue, [13](#)
- NextHeader
 - DialogueChoice, [19](#)
- npc_display_name
 - Dialogue, [17](#)
- npcNameText
 - DialogueManager, [20](#)
- onInvoke
 - Dialogue, [17](#)
- ParseDialogueFile
 - Dialogue, [13](#)
- ParseVariablesAndInitializeSections
 - Dialogue, [14](#)
- ProcessDialogueSectionsAndChoices
 - Dialogue, [15](#)
- Redirect
 - DialogueSection, [21](#)
- sentenceDelay
 - Dialogue, [18](#)
- DialogueManager, [21](#)
- SetDialogueVariable
 - Dialogue, [15](#)
- SetVariable
 - UniformVariables, [22](#)
- SplitDialogueLines
 - Dialogue, [15](#)
- Start
 - Dialogue, [16](#)
- Text
 - DialogueChoice, [19](#)
- uniformsChanged
 - Dialogue, [18](#)
- UniformVariables, [22](#)
 - GetVariable, [22](#)
 - SetVariable, [22](#)
 - UniformVariables, [22](#)
 - Variables, [22](#)
- uniformVariables
 - Dialogue, [18](#)
- uniformVariablesAlreadyDefined
 - Dialogue, [18](#)
- Update
 - Dialogue, [16](#)
- Variables
 - UniformVariables, [22](#)