DIALOGUE SYSTEM for unity Procedural Dialogue Addon

Version 1.x

Copyright © Pixel Crushers

Table of Contents

Introduction	3
We're Here to Help	3
Quick Start	3
Procedural Dialogue Addon Component	4
Procedural Dialogue Actor Component	5
Dialects	6
Moods	6
Lua Functions	7
Procedural Grammars	8
Regular Expressions	9
Procedural Grammar Set	9
Sample Grammars	10
How to Write Procedural Text	11
Procedural Text Instructions	11
Available Instructions	12
Overview of Available Instructions	12
Insert a Dialect Word	13
Choose a Random Element	13
Variable Values	14
Lua Code	15
Moods	15
A Note About Whitespace	16
English Text Parser	16
Custom Text Parsers for Other Languages	17
Questions? Feature Suggestions?	17

Introduction

Thank you for supporting the Dialogue System for Unity and the Procedural Dialogue Addon! This addon requires the <u>Dialogue System for Unity</u>. The future of dialogue is procedural. Using the Procedural Dialogue Addon, you can add variety and reactivity to the dialogue that your players experience.

This Addon does not use generative AI. It uses custom-built systems to transform your dialogue so you retain creative control of your writing. It performs three main functions:

- Allows you to write special <u>procedural text instructions</u> in your dialogue text.
- Performs text substitution to give actors unique vocabularies that you define called <u>dialects</u>.
- Transforms text using <u>grammars</u> that you define to change the way actors speak, such as making them talk like a pirate.

We want your project to be a success! If you have any questions or feature requests, please contact us any time at support@pixelcrushers.com.

We're Here to Help

- Email: support@pixelcrushers.com
- Forum: <u>https://pixelcrushers.com/phpbb</u>
- Discord: <u>https://discord.gg/FwUaCNt</u>
- Web: Tools \rightarrow Pixel Crushers \rightarrow Dialogue System \rightarrow Help \rightarrow Report a Bug

Quick Start

- Read the documentation and/or watch the tutorial videos.
- Import the Dialogue System for Unity into your project.
- Import the Procedural Dialogue Addon into your project.
- Play the demo scene in the Demo folder.
- Set up a scene with a Dialogue Manager GameObject. (See Dialogue System Quick Start).
- Add a Procedural Dialogue Addon component to the Dialogue Manager.
- Add a Procedural Dialogue Actor component to any dialogue actor in your scene.
 - Assign a grammar set such as PirateGrammarSet to make the actor talk like a pirate.
 - Assign a <u>dialect table</u> to give the actor a specific vocabulary.
- Include special <u>procedural text instructions</u> in your dialogue text.

Procedural Dialogue Addon Component

Inspector				а	:
🍸 🗸 Dialogue Manager			Sta	atic	•
Tag Untagged	▼ Layer	Default			T
Prefab Open	Select	Overrides			T
🕨 🙏 🛛 Transform			0	ᅷ	
🕨 # 🔽 Dialogue System Controller 🛛 🚱			0	ᅷ	
🕨 🗯 🗹 Instantiate Prefabs 🛛 🛛 😨 👎			ᅷ		
▶ # ✔ Input Device Manager 🛛 🛛 구			ᅷ	:	
🔻 🚓 🛛 Procedural Dialogue Addon (Script) 🛛 🛛 😧 ই			:		
Script	ProceduralDialo	gueAddon			۲
Default Dialect Table	🖩 GenericDialect1	able (Text Table)			\odot
Default Grammar Set None (Procedural Grammar Set)				\odot	
Require All Syntax In Braces					

Add the **Procedural Dialogue Addon** component to your Dialogue Manager. This component handles processing of procedural text generation when the Dialogue System shows barks and lines of dialogue and response menu options in conversations. It also adds <u>Lua functions</u> to set Procedural Dialogue Actors' moods.

You can optionally assign a **Default Dialect Table** and/or **Default Grammar Set**. If an actor doesn't specify its own dialect table or grammar set, it will use these defaults.

The **Require All Syntax In Braces** checkbox is explained in <u>How To Write Procedural Text</u>.

Procedural Dialogue Actor Component

Inspector	a:
Ravenbeard	Static 🔻
Tag Untagged Layer Default	•
🕨 🙏 🛛 Transform	0 ‡ ∶
🕨 # 🗹 Dialogue Actor (Script)	0 ‡ i
# Procedural Dialogue Actor (Script)	0 ‡ !
Script ProceduralDialogueActor	⊙
Dialect Table 🛛 🖬 RavenbeardDialectTable (Text Tab	ole) 💿
▼ Grammar Sets	1
= Element 0 💀 PirateGrammarSet (Procedural Gra	ammar Set) 💿
	+ -
Mood	

Add **Procedural Dialogue Actor** components to your dialogue actors. You can assign a **Dialect Table** and one or more **Grammar Sets**. If they're not assigned, the actor will use the Procedural Dialogue Addon's default dialect table and grammar set. The actor will use the dialect table and/or grammar sets to modify the dialogue text before it's displayed.

You can also optionally assign a **Mood**, which is covered in the <u>Dialects</u> section. Moods are usually set at runtime by <u>Lua functions</u> in response to player actions or other changes in the game state.

Dialects

🍄 Text Table	- 0	>	<
Text Table			:
🖩 Pirate Text Table (Text Table)		\odot	¢
Languages	Fields		
Field	Default		▼
= hello	ahoy		
= friend	matey / bucco		
= friend angry	ye scurvy mate / scurvy dog		
yikes	shiver me timbers		
	+	-	

A dialect is a table containing key words or phrases and a list of synonyms. When an actor's text contains the key word or phrase, the text will instead use of one of the synonyms. In the table entry's Fields section, use a forward slash (/) character to separate synonyms. For example, the "friend" entry above has two synonyms: "matey" and "bucco". Since dialect tables are Text Table assets, you can localize them to multiple languages. To create a new dialect table, select menu item **Asset > Create > Pixel Crushers > Common > Text > Text Table**.

Moods

🍄 Text Table	— D X
Text Table	:
🖩 Pirate Text Table (Text Table)	• •
Languages	Fields
Field	Default 🔹
= hello	ahoy
friend	matey / bucco
friend angry	ye scurvy mate / scurvy dog
= yikes	shiver me timbers
	+ -

Procedural dialogue actors have a Mood field that can be set to a value. (By default, the Mood field is blank.) If the Mood field is set, then the dialect table will look for an entry that has the mood appended to the end. In the example above, if the actor's Mood is set to "angry", the dialect will use the "friend angry" field. If no mood-specific entry exists, it will use the default field (e.g., "angry"). You can set an actor's mood in the Procedural Dialogue Actor component's Mood field, by script in C#, or using the <u>SetMood() Lua function</u>.

Lua Functions

The Procedural Dialogue Addon component adds these Lua functions, which you can enter directly into Conditions and Script fields or access via the "…" dropdown menus under the category **Custom** > **Procedural**.

SetMood(actorName, mood)

Sets an actor's mood. This affects the words that the actor uses from its dialect table.

```
Example: SetMood("Ravenbeard", "happy")
```

GetMood(actorName)

Returns an actor's current mood.

Example: GetMood("Ravenbeard") == "happy"

Procedural Grammars

Inspector	а:
Pirate Grammar (Pirate Grammar)	07‡ :
40	Open
Script 🔹 PirateGrammar	
Description	
This grammar translates common words into movie pirate	e speak.
Translations	
<pre>@alcohol ale beer liquor rum: grog @am are is: be boy: lad / laddy castle: fort / garrison chest: coffer coin: doubloon coins: doubloons does: do friend: matey / bucco friends: mateys / buccoes girl: lassie good: brave/hearty hello: ahoy hey: yar/arr/avast it's: it be judge: magistrate lame: land-lubbing / scallywagging map: chart my: me pirate: buccaneer pull: heave @reward treasure: booty sailor: seaman / seadog shackles: irons</pre>	
Alphabetize	
Question Matey Chance	- 0.5
Exclamation Matey Chan	0.5
Exclamation Yar Chance	- 0.5
▼ Test	
Test Text	
Hello, my friends! Let's spend our reward on beer and wo	omen!
Test	
Ahoy, me mateys! Arr! Let's spend our booty on grog and	wenches!

A grammar is an asset that transforms text. Basic grammars do simple word substitution, which is easy to set up, efficient, and sufficient for most purposes. The **Alphabetize** button sorts the Translations section alphabetically. The **Test** section provides a quick way to test your grammar.

To create a new grammar, select **Asset > Create > Pixel Crushers > Dialogue System > Procedural > Grammars** and any of the choices such as **Basic Grammar**. You can also create your own subclass of **ProceduralGrammar** and override the **ProcessText()** method to perform additional or different text transformation, such as using tensor flow to process text through a machine learning model.

The Basic Grammar is usually adequate for most purposes and doesn't require any additional scripting. Add your keywords to the Translations section, separated by a colon (:) and then a list of words that the actor should use instead of the keyword. Separate this list of words with forward slashes (/).

For example, in the pirate grammar above castle: fort/garrison will translate instances of the word castle into fort or garrison.

Regular Expressions

You can also include <u>.NET regular expressions</u> by starting the line with @.

For example, in the pirate grammar above @reward|treasure: booty will translate instances of the regular expression "reward|treasure" into booty. The only exception is that your regular expressions cannot contain colons (:) because this character specifies the split between the regular expression and its translated value.

Procedural Grammar Set



A Procedural Grammar Set is an asset that manages a list of Procedural Grammar assets. Since every language has its own unique structures and rules, you will need to create a unique procedural grammar for each language. Assign the default language's grammar set to the **Default Grammar** field. Assign additional language-specific grammar sets to the **Language-Specific Grammars** list. In the example above, the Language-Specific Grammars list contains a Spanish version of the pirate grammar. If the Dialogue System's language is set to Spanish, the addon will use PirateGrammar_es instead of PirateGrammar to substitute words and phrases.

To create a new procedural grammar set, select **Asset > Create > Pixel Crushers > Dialogue System > Procedural > Grammars > Procedural Grammar Set**.

Sample Grammars

Grammar	Description
HighMedieval	Medieval upper class or Elizabethan speech
Leetspeak	1990s/2000s-era computer leetspeak
ModernSlang	Example of modern slang
Pirate	Movie pirate speech

The addon comes with these sample grammar sets, located in the **Data** > **Grammars** subfolder:

How to Write Procedural Text

Dialects and grammars are a low-effort, high-reward way to give your characters personality. However, the real power of the Procedural Dialogue Addon comes when you combine them with the addon's text parsing feature.

The Procedural Dialogue Addon lets you include instructions in your dialogue text. Rather than presenting the player with static text that's the same every time, the text will have variety that's aware of runtime states such as the actor's dialect and current mood. These runtime choices can also be remembered so they can be accurately referenced later.

Procedural Text Instructions

When writing your dialogue text and bark text, you can place procedural text instructions inside double braces, such as the following instruction to choose between "happy" or "sad":

• Dialogue Text: I feel {{happy/sad}}.

On the <u>Procedural Dialogue Addon</u> component (on the Dialogue Manager), if you've ticked the **Require All Syntax In Braces** checkbox, then all procedural text instructions must be inside double braces. If you <u>un</u>tick this checkbox:



then procedural text instructions may also be in angle brackets for less typing, such as:

• Dialogue Text: I feel <happy/sad>.

You can also nest angle bracket instructions inside double braces or inside angle brackets, such as:

• **Dialogue Text**: I feel <<**elated/joyful**> / **sad**>.

The instruction above will choose randomly between "<elated/joyful>" and "sad". If it chooses "<elated/joyful>", then it will randomly choose between "elated" or "joyful".

Available Instructions

You can use these procedural text instructions in your text:

Instruction	Description	
<word></word>	Insert a dialect word	
	Randomly choose from a list of values	
< var variable>	Show a Dialogue System variable value	
< var variable = value>	Set a variable and show its value	
< var variable ? value>	Show a value only if a variable is true	
<lua code=""></lua>	Show the result of a Lua expression	
<lua ?="" code="" value=""></lua>	Show a value only if the Lua expression is true	
<mood =="" mood=""></mood>	Sets the speaker's current mood	
<mood ?="" mood="" value=""></mood>	Show a value if the speaker's current mood is the specified mood	
&sp	Force a space character that won't be trimmed by the addon	

Overview of Available Instructions

Reminder:

- If the Procedural Dialogue Addon component's **Require All Syntax In brackets** is ticked, then instructions must be included in double braces such as **{{hello}}**.
- You can nest additional instructions using angle brackets such as {{hello / <hi/howdy>}}.
- If **Require All Syntax In Braces** is <u>un</u>ticked, you can use angle brackets instead of double braces for the outermost instruction such as **<hello** / **<hi/howdy>>**.

Each of these instructions is detailed on the next page.

Insert a Dialect Word

A word by itself in double such as **{{friend}}** or angle brackets such as **<friend>** will be replaced by the speaker's dialect. For example, say you've assigned this table to the speaker's <u>Procedural Dialogue</u> <u>Actor</u> component:

🚱 Text Table	— D X
Text Table	:
🖩 Pirate Text Table (Text Table)	⊙ ≎
Languages	Fields
Field	Default 👻
= hello	ahoy
= friend	matey / bucco
= friend angry	ye scurvy mate / scurvy dog
yikes	shiver me timbers
	+ -

In this text:

• **Dialogue Text**: Have a drink, **{{friend}}**.

The instruction **{{friend}}** will be replaced by either "matey" or "bucco", as in "Have a drink, bucco."

If the dialect text table doesn't have an entry for the word you've specified, it will use the word itself. For example, since "airplane" is not in the text table above, **{{airplane}}** will simply be replaced with "airplane".

Choose a Random Element

Separate random choices with forward slash (/) characters. The addon will replace the instruction with one of the choices. For example:

• Dialogue Text: Time to <walk the plank / feed the sharks / die>!

The instruction above will be replaced by "walk the plank", "feed the sharks", or "die". If any of those texts are in the dialect table, it will choose a dialect word. For example, if "die" is in the dialect table:

```
e die perish / croak
```

then the text might end up being "Time to perish!".

Note: To include a literal slash characte (/) in your text, precede it with a backslash (\), such as:

• **Dialogue Text**: Hi! My name's Jess. My pronouns are <she\/her / he\/him>.

Variable Values

Show a Variable Value: {{var variable}}

Use **{{var variable}}** or **<var variable>** to show the value of a Dialogue System variable. Example:

• Dialogue Text: Hello, <var Actor>.

Note that this is similar to the Dialogue System [var=*variable*] markup tag, except that it's more flexible as you'll see below.

Set a Variable Value: {{var variable = value}}

Use **{{var variable = value}}** or **<var variable = value**> to choose a value, show it in the text, and record the value in a Dialogue System variable. Example:

- Dialogue Text: Time to {{var Fate = <walk the plank / feed the sharks>}}!
- **Dialogue Text**: Har har! Remember when we made that bloke **<var Fate**>?

In the text above, the first dialect text might be "Time to walk the plank!", which would set the Dialogue System variable "Fate" to "walk the plank". The second dialogue text would include the recorded value of variable: "Har har! Remember when we made that bloke walk the plank?"

Conditionally Show Text Based on a Variable Value: {{var variable ? value}}

Use **{{var variable ? value}}** to conditionally show text based on the value of a Boolean (true/false) Dialogue System variable. Example:

• Dialogue Text: Swab the decks! {{var IsPirateAngry ? Or you'll <var Fate>, too!}}

In the text above, if the Dialogue System variable "IsPirateAngry" is true, then the text might be "Swab the decks! Or you'll walk the plank, too!". But if the variable is false, it will omit the instruction's text and only be "Swab the decks!".

Lua Code

Show Lua Code Result: {{lua code}}

Use **{{lua** *code*}**}** to show the result of running some Lua code. Example:

• **Dialogue Text**: I'll give you **{{lua math.random(2,5)}}** chances to answer my riddle.

In the text above, math.random(2,5) returns a random number between 2 and 5, so the text could be "I'll give you 3 chances to answer my riddle."

You can also save this in a variable:

• **Dialogue Text**: I'll give you **{{var Chances = <lua math.random(2,5)**>}} chances.

Conditionally Show Text Based on Lua Code: {{lua code ? value}}

Use **{{lua** *code* ? *value*}} to conditionally show text based on the true/false return value of Lua code. Example:

Dialogue Text: Swab the decks! {{lua math.random(100) < 20 ? Or you'll <var Fate>, too!}}

In the text above, instruction runs the Lua code math.random(100) < 20, which returns true 20% of the time. This means 20% of the time the text will be "Swab the decks! Or you'll walk the plank, too!" but 80% of the time it will only be "Swab the decks!".

Moods

Set a Mood: {{mood = value}}

Use **{{mood = value}}** to set the speaker's current mood. This will take effect in the next dialogue entry spoken by the actor. Example:

• **Dialogue Text**: That's wonderful! {{mood = happy}}

Conditionally Show Text Based on Mood: {{mood mood ? value}}

Use **{{mood mood ? value}}** to conditionally show text only if the speaker's current mood is *mood*. Example:

• Dialogue Text: Hello. {{mood happy ? It's a <beautiful> day, isn't it?}}

A Note About Whitespace

The procedural text parser trims extra whitespace. To include a space character without it being trimmed, use the character combination **&sp**. Example:

• **Dialogue Text**: My mother<-in-law / &sp of pearl necklace> is nice.

If the text parser chooses "-in-law", then the text will be "My mother-in-law is nice."

If the text parser chooses "&sp of pearl necklace", the text will be "My mother of pearl necklace is nice." Since the text parser trims whitespace, without the forced space indicated by &sp the text would have incorrectly been "My mother pearl necklace is nice."

English Text Parser

When the Dialogue System's language is set to default (i.e., no language code) or the language code contains "**en**", the addon will use a special text parser that contains additional handling for English. This adds the following features:

Feature	Description			
Indefinite Articles	Ensures the indefinite articles a and an match the nouns that they precede. For example, if the text is "a apple", it will correct this to "an apple".			
Noun Pluralization	If syntax such as <knife> is followed by an "s" as in "<knife>s", it properly pluralizes the word to "knives".</knife></knife>			
Linking Verbs	Ensures the verbs is and are match their subjects. For example, it corrects "games is fun" to "games are fun" and "they is playing" to "they are playing".			
Verb Tenses	If syntax such as <walk> is followed by any of the following modifiers, it changes the verb tense:</walk>			
	Modifier	Source	Becomes	
	-S	<walk>-s</walk>	walks	
	-ing	<walk>-ing</walk>	walking	
	-is	<walk>-is</walk>	is walking	
	-are	<walk>-are</walk>	are walking	
	-ed	<walk>-ed</walk>	walked	
	-en	<walk>-en</walk>	walked	
	-have	<walk>-have</walk>	have walked	
	-had	<walk>-had</walk>	had walked	

While these features aren't usually necessary when you're writing lists such as "<a/b>" directly into the text, they're critical to ensuring correct grammar when replacing words from dialect and grammar tables since you won't necessarily know ahead of time what the possible values are.

Exceptions to these rules are in text files located in Assets / Plugins / Pixel Crushers / Dialogue System Addon for Procedural Dialogue / Data / Resources / English.

Custom Text Parsers for Other Languages

If you want to write a special text parser for another language, make a subclass that extends **ProceduralTextParser**. Override the **ProcessLanguageSpecificSyntax()** method.

Then make a subclass of **ProceduralDialogueAddon** and override the TextParser property to return your subclass when the Dialogue System is set to the language.

Questions? Feature Suggestions?

If you're stuck or have feature suggestions, please <u>contact us</u>. We're here to help!