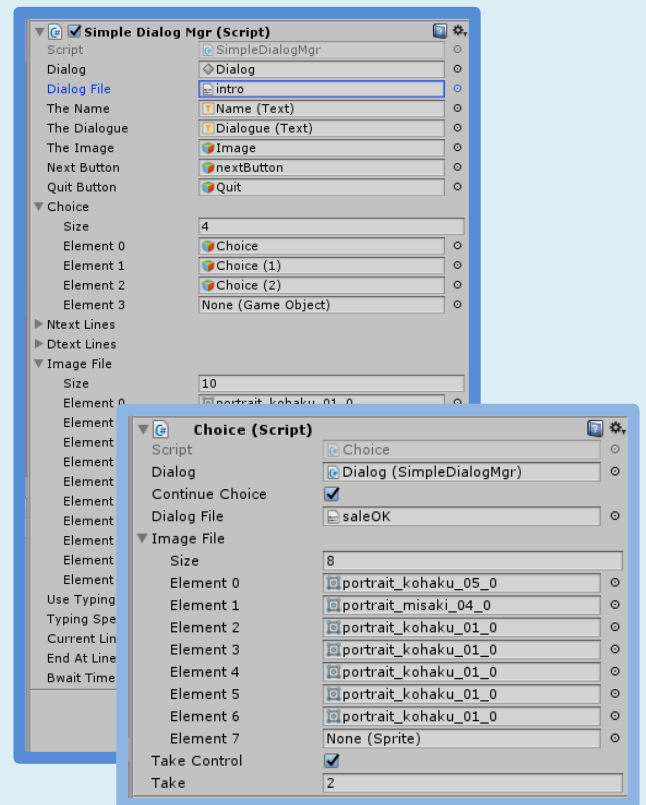


# Easy&Simple Dialog System

## Contents

1. Components - - - - - (2)
2. Let's create a Dialog - - - - - (3~13)
3. Let's control the settings window - - (14~16)
4. Let's customize the script - - - - - (17)



Thank you for purchasing this asset

Easy&Simple Dialog System can make Dialog with one or two scripts

That's why that it's very simple asset.

Don't be afraid. It's okay if you only read the Guide Book. Cheer up!!!^^

- I tried my best, but there may be mistranslation, when I translate it into a Google Translator. Sorry.

# 1. Components

## Easy&Simple Dialog System

- > SimpleDialogMgr.cs *// The main of this asset*
- > Choice.cs *// Feature extension script for ChoiceSentence*
- > Guide Book *// User's Guide folder*
- > License *//License document folder, Please READ ME!!!*
- > Example *// Example file*

User Guide.unity *// Example scene file*

Anim *// Animation folder*

Script *// Script folder (Just think that you can apply it in this way.)*

Sprite*// Sprite file*

Text *// Dialog file folder*

## 2 .Let's make Dialog

The dialogs that you can make with this script is made of NameTextObject, DialogueTextObject, Image Object, nextButton, ChoiceButton, QuitButton.

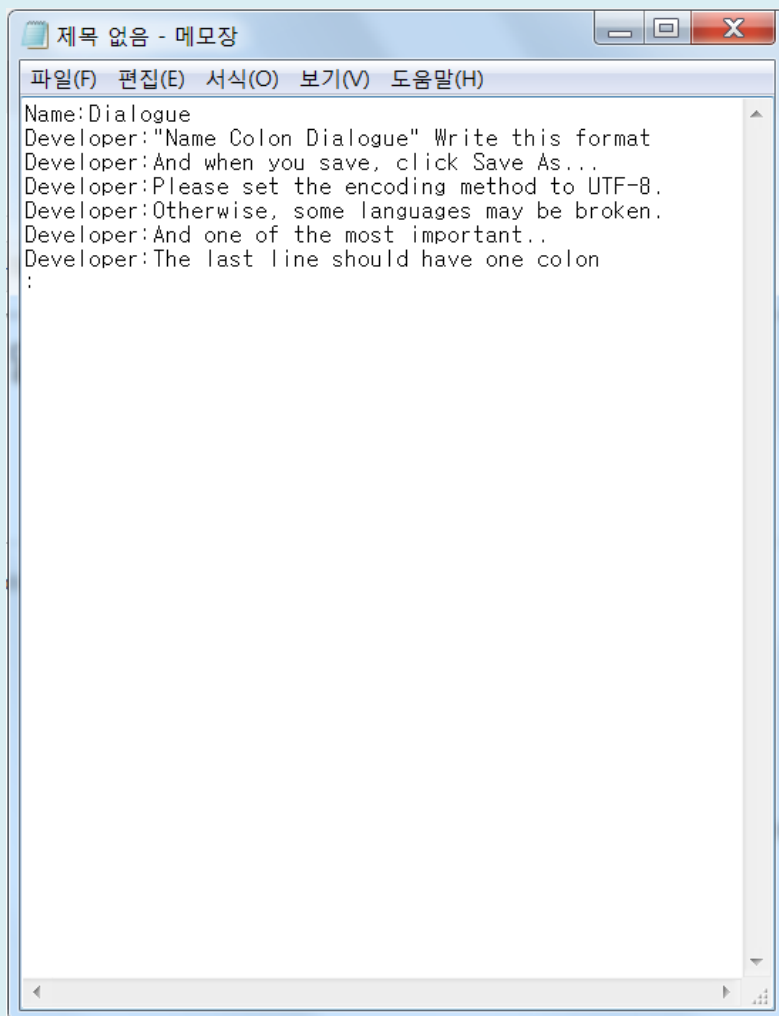
But, you can only use it if you have NameObject and DialogueObject. It does not matter whether you put the rest or not. Or you can replace it. You can add what does not exist here.

I'll teach you the basics. So the rest you can decorate freely!

### 0. Ready the preparation

You need some preparation. If you want to create your own dialog, you need enough material for decorate the dialog. The **illustration of the chat character** is good and the **some button images and the background image which can decorate the dialog** are also good

If you have prepared several sprites, You have to prepare the most important Diaog file that will be made in the dialog. That's nothing. Open the notepad. And you have to write the dialogue you want to show in the dialog.



**Name:Dialogue** Please enter in order.

If one line is over **Press Enter**.

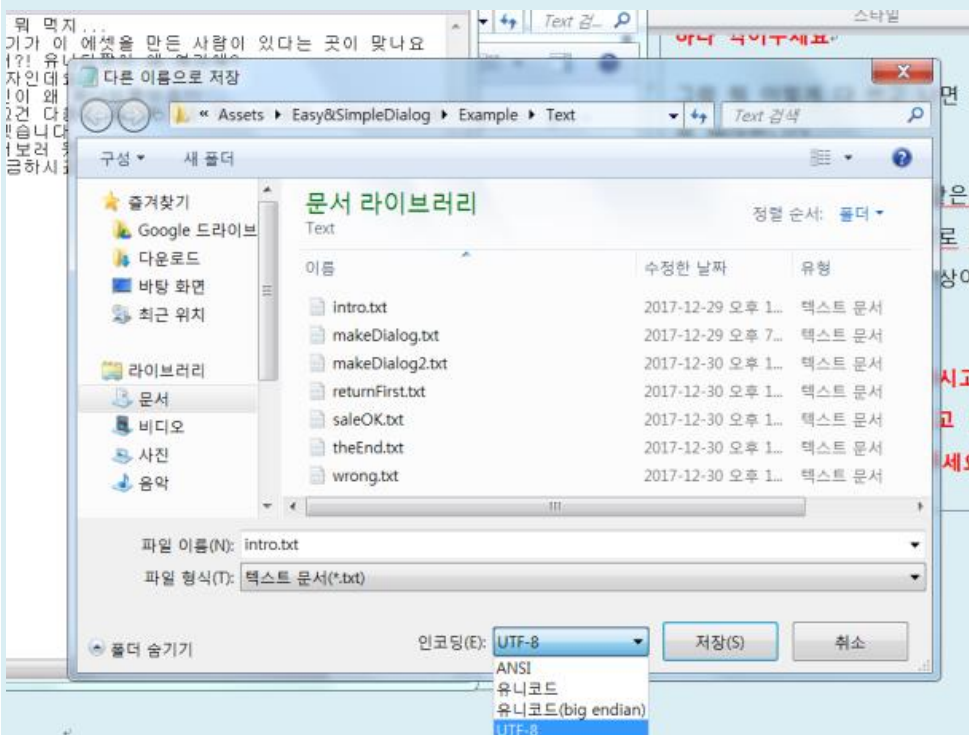
And **please enter one colon(:) on the last line**.

You must save it when you write finished.

You have to be careful here!!!

Some languages crash due to encoding errors when they are loaded into Unity.

So **Click Save As, set the encoding method to UTF-8, and save it**.



Save this way.

Then the preparations are finished.

## 1. Put Script

If the object is not deleted or disabled,

It does not matter where you put the SimpleDialogMgr script. I put it in the top-level parent object of the dialog object for easy management.

In the Hierarchy window, select the object you want to place the script in

Drag the SimpleDialogMgr script in the Asset to the Inspector window finished.

You'll see a lot of settings, but do not touch it yet.

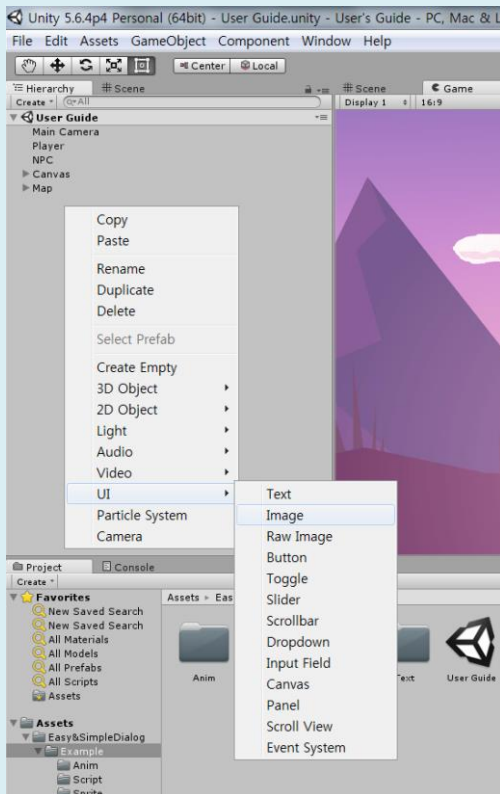
## 2. Let's make a dialog background.

There is something you need to know before you do it in earnest.

You must have a parent object that contains everything that makes up the dialog

I made the dialog window background the top-level parent object of the objects that make up the dialog. But, The background doesn't have to be a parent object.

You only need to know the basic structure and make it freely.



It's a background, so you have to have an image.

Click the right mouse button to create an Image object in the UI. Next, resize and decorate as you wish

Then drag the decorations to the dialog background through the Hierarchy window.

### 3. Let's make NameTextObject and DialogueTextObject

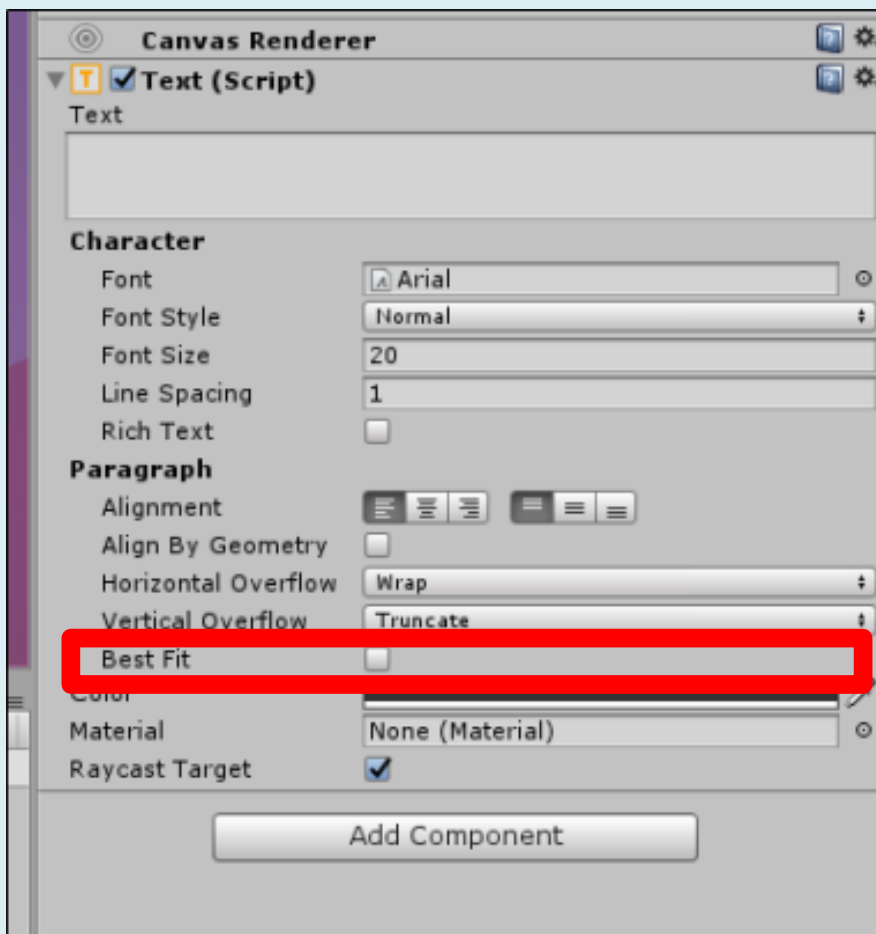
This time, press the right mouse button and press **UI > Text** to create

**two Text objects** that are children of the dialog background object.

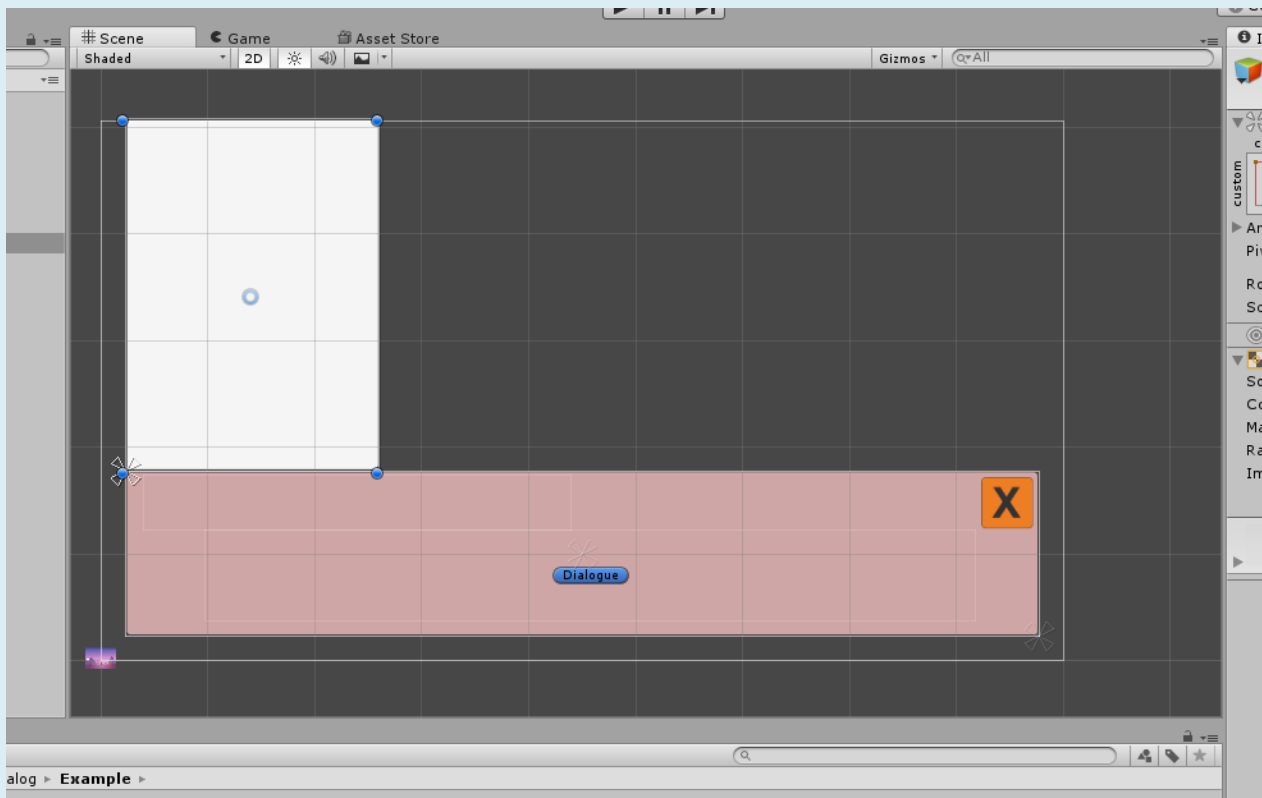
Make sure **each name and dialogue** is **large enough** to display. If desired, select the font and font color. You do not need to write anything in the text.

There is one caution.

**If you look at the dialogue object, there is a Text component. Turn off BestFit. Turning on this will not work properly.**



#### 4. Let's make image object (Optional)



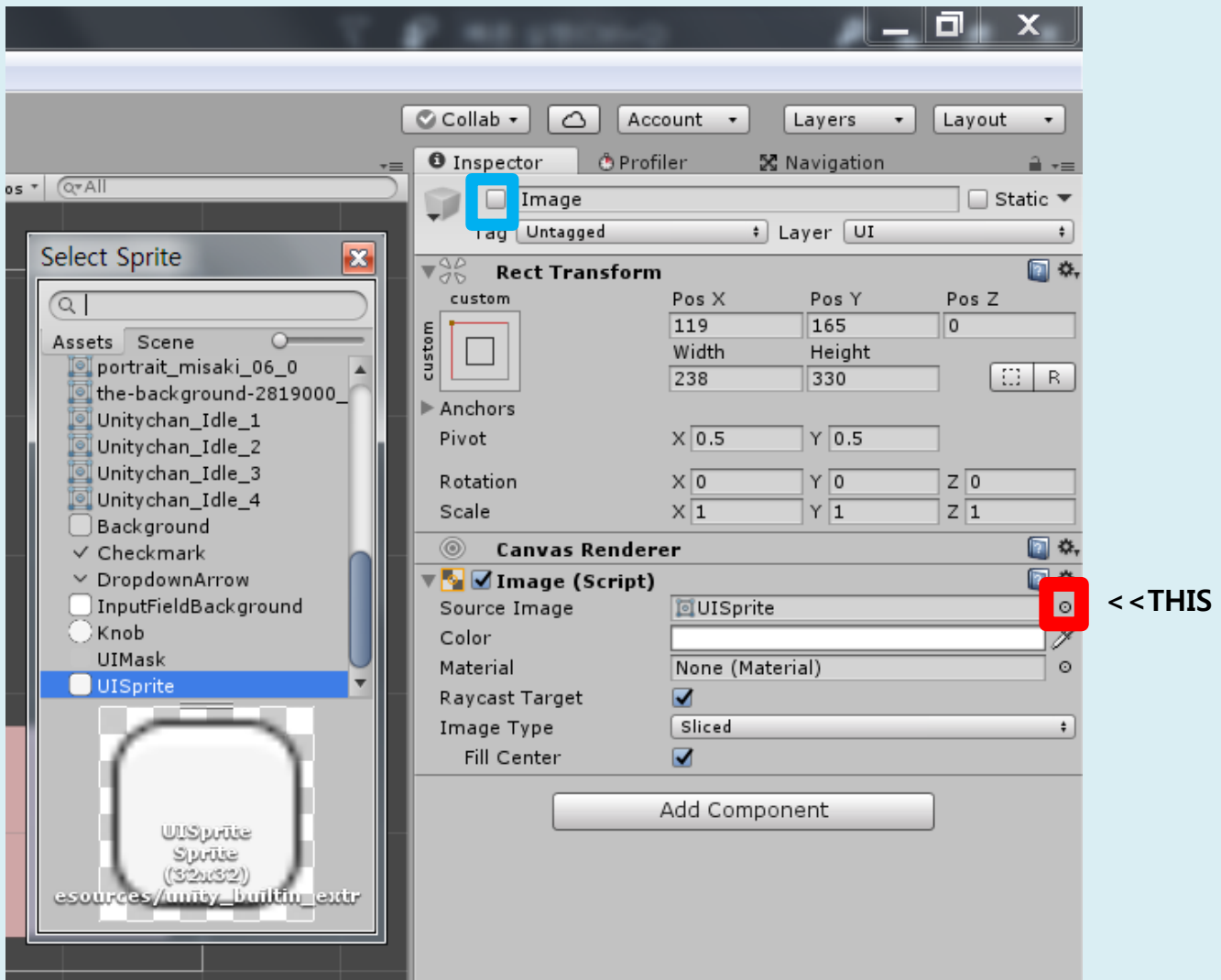
I made it roughly about this size

Press the right mouse key and make it **UI > Image**. Put this as a child of the dialog background object. This is the place for the illustration. **Please adjust the size of the proportion of illustrations to fit roughly.** Otherwise the illustration will be distorted.

As you can see, there must be an image that is white.

Set the color of **the Image component to white**





If you press the button marked with a red square box, a window called Select Sprite pops up.

Find **UISprite** over there and put it in.

And if there is check in the blue square box, click to **uncheck** it.

You should **Deactivate Image Objects**

## 5. Let's make nextButton (Replaceable with keyboard)

It is a **button to continue the Dialogue**.

Because there are many buttons, it is better to put them in the parent object

If you hate it, you do not have to. UI > Button makes a button by pressing

Put this as a child object in the dialog background. Then write the word "continue" or make a cool button as you want. But **decorations should be child objects of nextButton**.

And one important thing, you need to connect to the script to get the dialog going.

1. Select the nextButton object and look at the Inspector window.
2. At the bottom you see a button Add Component. Press it.
3. Then click Event and click EventTrigger to add it
4. Click Add New Event Type.
5. Click Point Up.
6. Press + to add a list
7. Drag and drop the object that contains the script to the place where None is written
8. Click No Function
9. Click SimpleDialogMgr and click skipButtonControl ()

It's over. Remember that because you have a lot of similar things in the future.

+ If you want to replace it with a keyboard..

See the SimpleDialogMgr script.

**Void Update ()** function is commented out.

If you uncomment it, there will be **KeyCode.Space**

**KeyCode.** You can put the key you want.

## 6. Let's make Quit button(Optional)

**The button to close the dialog during a dialogue.** It's okay not to have.

Make a UI-Button in the appropriate place. After appropriate sized and decorated, add an event trigger as you did before. Then set it to Point Up, and Drag the object containing the script to None.

Click SimpleDialogMgr and click DisableDialog ()

The end.

## 7. Let's make ChoiceSentence(Optional)

Now it remained the hardest to make ChoiceSentence

Make one empty object.

Put choice button objects in an empty object

Make your choice buttons cool and position them appropriately.

Then add an event trigger to the button and connect it to the function that performs the desired action.

But I hope it's done,

However, you can press the choice button next dialogue may follow

1. If that's the case, add the Choice script to the **button that "Dialogue will follow"**
2. Drag the object that we just put the Choice script into and put it into the event trigger of that object. (Putting yourself in the event trigger)
3. Make a Dialog file that will be followed and Put it in DialogFile of Choice script.
4. If you use an image, put the illustrations for the next conversation in the image array in order from the top.
5. It also deactivates the empty object that initially puts the Choice buttons

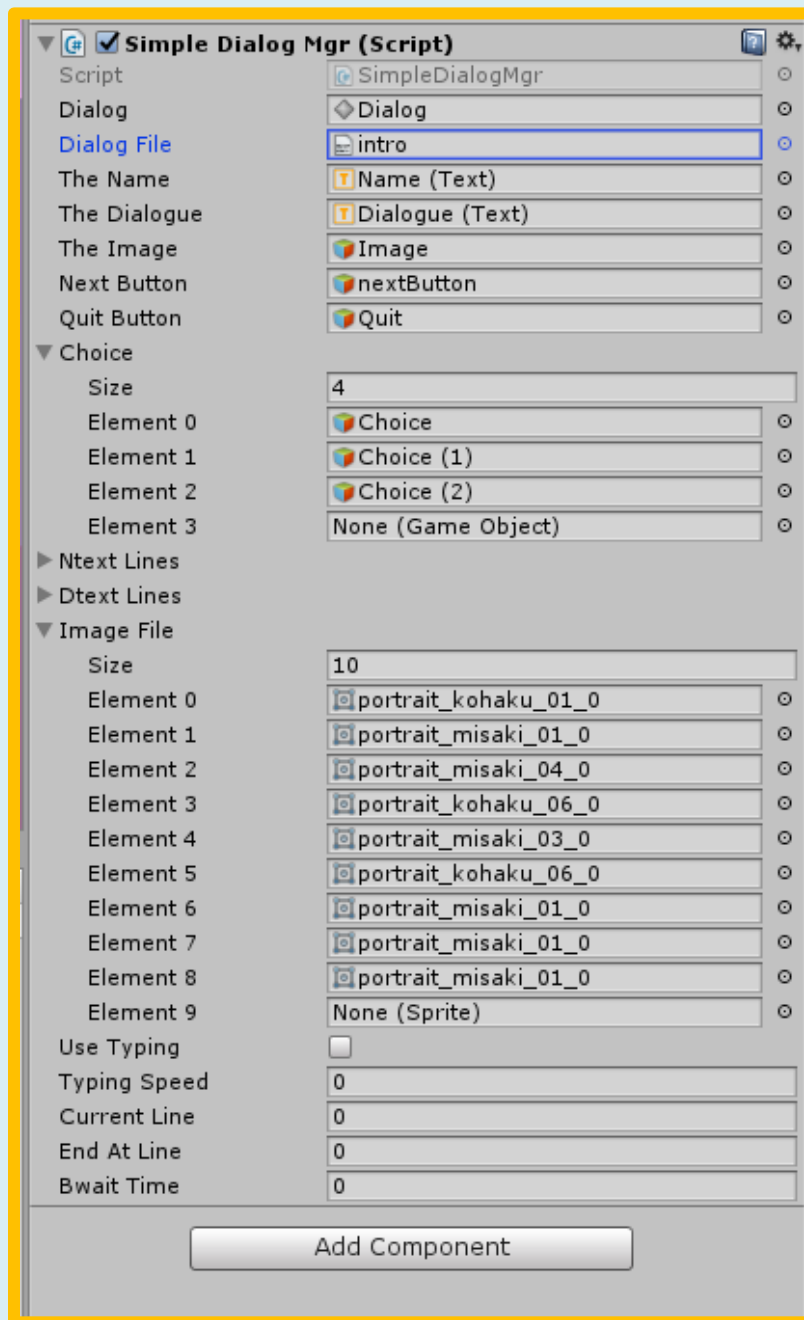
The rest of the settings are explained in more detail in "3. Let's control the settings window ".

## 8. The final step of making a Dialog

1. Click on the object containing the SimpleDialogMgr script and **look in the Inspector window**.
2. Drag the object containing the SimpleDialogMgr script into the **"Dialog"**.
3. Put the first dialog file into **"DialogFile"**.
4. Put the NameTextObject into **"The Name"**.
5. Put a DialogueTextObject into **"The Dialogue"**.
6. If you use illustrations, put the image objects you created before into **"The Image"**.  
(+If you leave it blank, it is automatically deactivated.)
7. If you use nextButton, put nextButton object in **"nextButton"**.  
(+Put object even if keyboard is used.)
8. If you use the Exit button, put the Exit button in the **"QuitButton"**.  
(+If you leave it blank, it is automatically deactivated.)
9. If you use the Choice button, put it in the **"Choice"** in order from top to bottom.  
(+Add one more element to use, next leaving the last element blank)
10. If you use illustrations, put the illustrations in the order from the top.  
(+Add one more element to use, next leaving the last element blank)
11. Deactivates **the parents object at the top of the dialog**
12. If you reactivate the dialog in any way, it will start to work.

# 3. Let's control the settings window

## 1. SimpleDialogMgr.cs



**DialogFile:** dialog file

**TheName:** name Text object

**TheDialogue:** dialogue Text object

**TheImage:** image object

**Choice:** ChoiceSentence object

**NtextLines:** name array

**DtextLines:** dialogue array

**ImageFile:** illustration files

**UseTyping:** Do you use a typing effect?

**TypingSpeed:** typing effect speed

**CurrentLine:** dialogue progressive variable

**EndAtLine:** last line

**BwaitTime:** nextButton waiting time

**Choice:** The parent objects of the select button enter the Choice.

Basically it outputs from top to bottom.

**ImageFile:** Basically it outputs from top to bottom. It is also possible to prevent anything from appearing unless you put the image in the middle.

*Makes the size of the array larger by one than the line in the dialog box file.*

**UseTyping:** You can turn AutoTyping effect off and on.

**TypingSpeed:** The closer to 0, the faster, and *It should not be negative number.*

**CurrentLine:** Increase from 1 to 1 by (array size - 1).

*// Don't carelessly touch it because it's a progressive variable.*

**EndAtLine:** (array size - 1) *// You should never touch this too.*

**DisableDialog()** : This is a function that disables the dialog.

If you want to turn off the dialog,

Please deactivate this by executing this function.

And, **Auto-typing is working**

**and If the Dialog has not yet been printed**

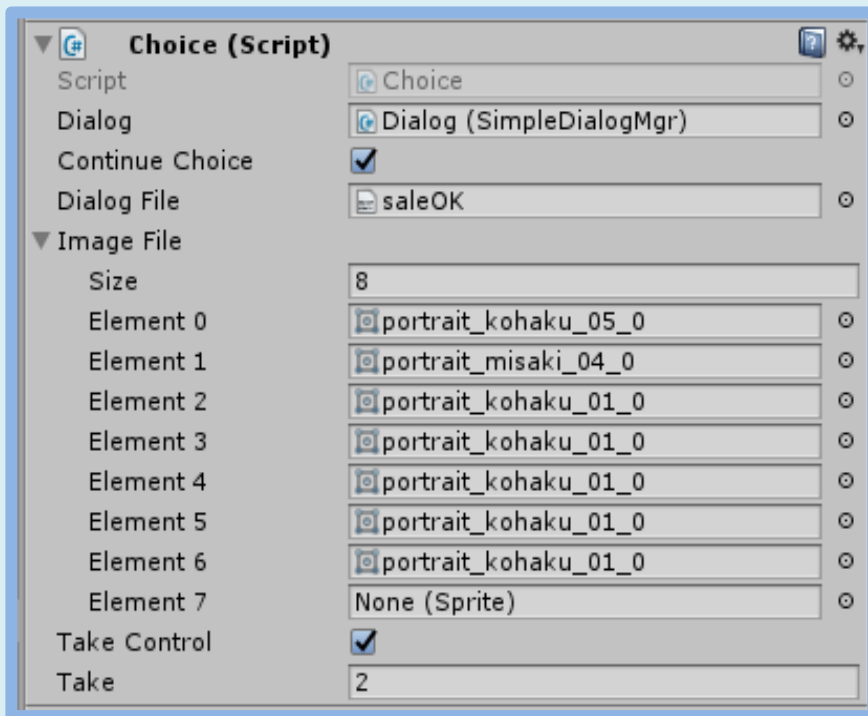
**Never deactivate it.**

**ReadyDialog(TextAsset)** : Dialog preparation function. By putting a dialog file in the

Parameter Replace dialog file. *It should be control indirectly through the Choice script and not recommended for direct touching.*

**ChoiceOnOff()** : The Function who decides whether to continue using the ChoiceSentence in the next Take. *It should be control indirectly through the Choice script and not recommended for direct touching.*

## 2. Choice.cs



**Dialog:** dialog script

**DialogFile:** continue Dialog File

**ContinueChoice:** Whether there is a choice sentence in the continue dialogue

**ImageFile:** image File for next dialogue

**TakeControl:** Do you take variable?

**Take:** ChoiceSentence progress variable

**ContinueChoice:** This variable determines whether there is a choice sentence in the dialogue to follow. If checked, a selection button for the Choice array corresponding to the next take is displayed. If unchecked, the take variable does not grow and the select button does not appear.

**Take :** Each time the Select button is pressed, it is incremented by 1 from 0 to (Choice array size – 1). I can say "take variable=The index number of the Choice array".

Basically, Choice array are output from Choice[0] in order, However If you control the take variable, you can pre-use the selection buttons you have used or the selection buttons you have not yet used. Use the size of the Choice array carefully. This is because there is a high probability that errors will occur by mistake.

**TakeControl:** You need to check if you want to control Take variables.

Otherwise, changing the value of the take variable has no effect.



### 3. Let's customize the script

You may think that you want to make an effect or add a new feature through the Dialog.

I'm going to give you tips for that extension.

**I think most of those keys are in the `currentLine` variable, `Take` variable, and the Event Triggers of Choice Button.**

As an example, an object named `Dfileguide` was made in the example file as an Event Trigger for the Choice button. It is a simple principle to activate the object with the event trigger of the select button

Using the `currentLine` variable and the `Take` variable, you can see what dialogue is going on. **This means that you can use the conditional statements to make an event when you become a certain dialogue.**

Event Triggers on choice buttons can be more useful. Just by being able to connect event triggers to functions in a particular script, you can do various things.

For example, if you click on the Choice Button, the parameter 1500, which is entered in the Event Trigger, is passed to the function in the specific script, which makes it possible to reduce the variable `money` by 1500.

*It's not a great tip, but I hope you'll make a great game*

*Thank you so much for reading so far.*

*All of you cheer up! ^^*