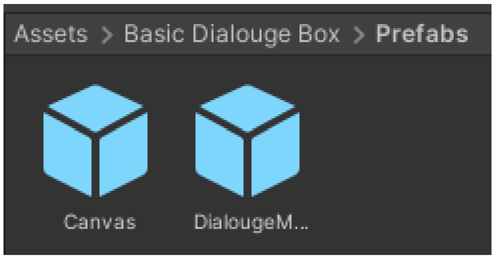
Latest Verified Unity Version: 2021.3.3f1

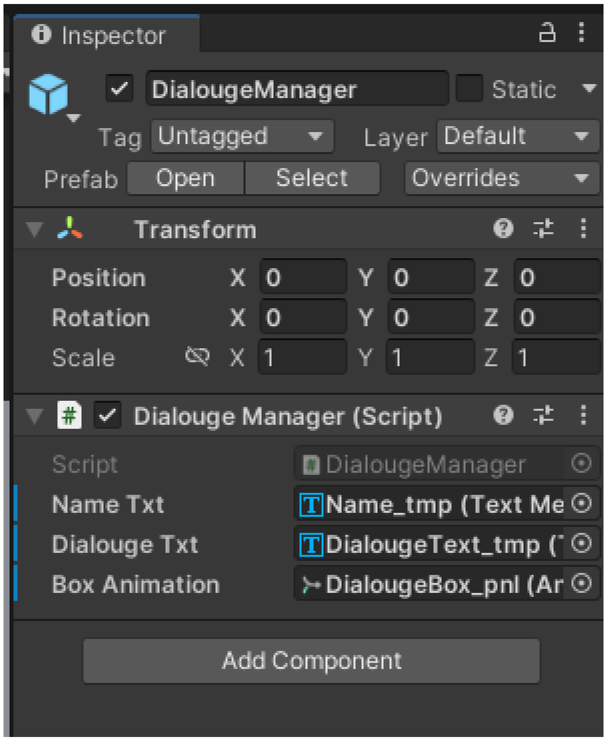
Installation

1) Import the Unity package into your Unity Project
2) Drag both the ***Canvas*** and ***DialogueManager*** prefabs into the scene.



Assets > Basic Dialouge Box > Prefabs

3) Inside your scene's ***DialogueManager***, drag and drop ***DialougeText_tmp***, ***Name_tmp***, and ***DialougeBox_pnl*** into the inspector fields shown below.



☑ DialougeManager

Tag Untagged

Layer Default

Transform

Position X 0 Y 0 Z 0

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

☑ Dialouge Manager (Script)

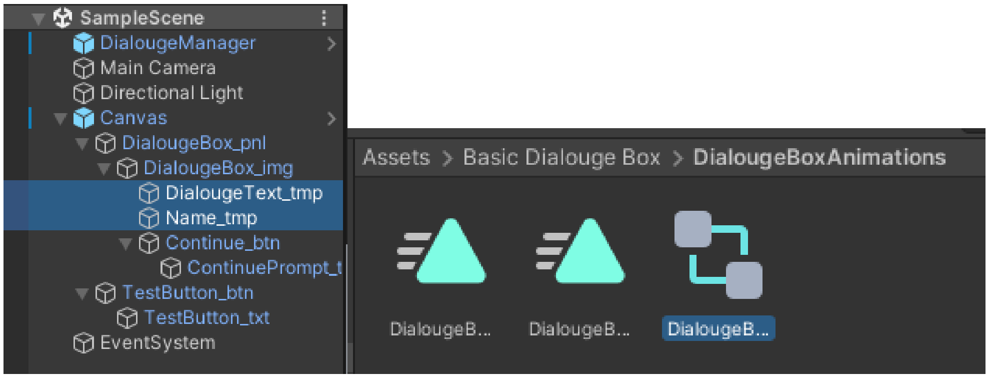
Script DialougeManager

Name Txt Name_tmp (Text Me

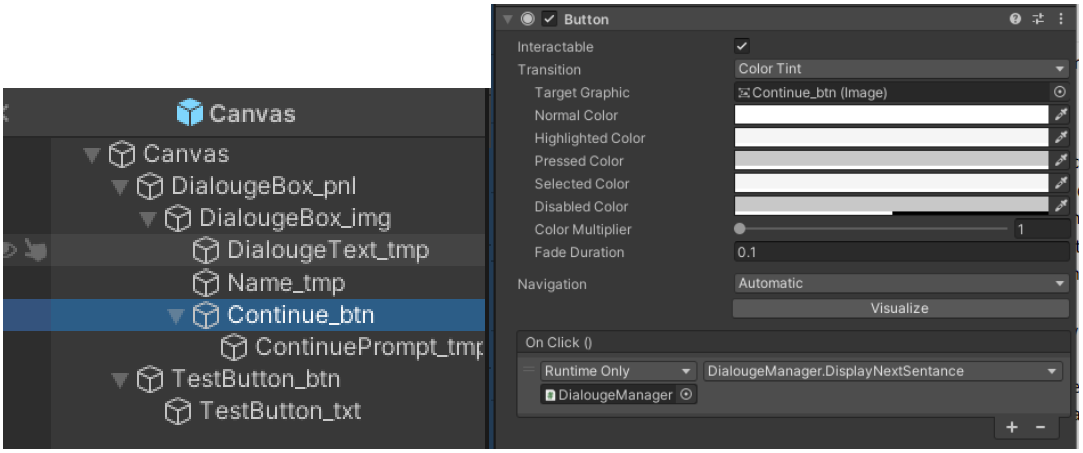
Dialouge Txt DialougeText_tmp ('

Box Animation DialougeBox_pnl (Ar

Highlighted below are the locations of these components:



4) On ***Continue_btn***, make sure that the ***OnClick()*** event calls ***DialogueManager.DisplayNextSentance***. If not, then make sure to set it up as shown below:



Interactable ☑
Transition Color Tint
Target Graphic Continue_btn (Image)
Normal Color
Highlighted Color
Pressed Color
Selected Color
Disabled Color
Color Multiplier 1
Fade Duration 0.1
Navigation Automatic
On Click ()
Runtime Only DialougeManager.DisplayNextSentance
DialougeManager

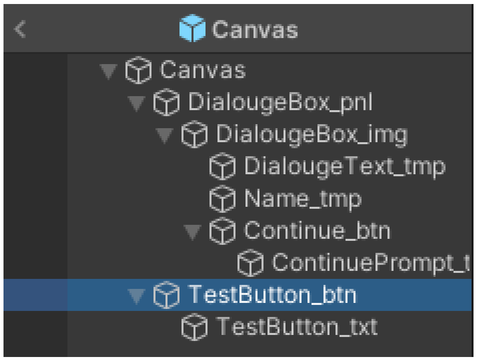
And that’s it!

How to Use

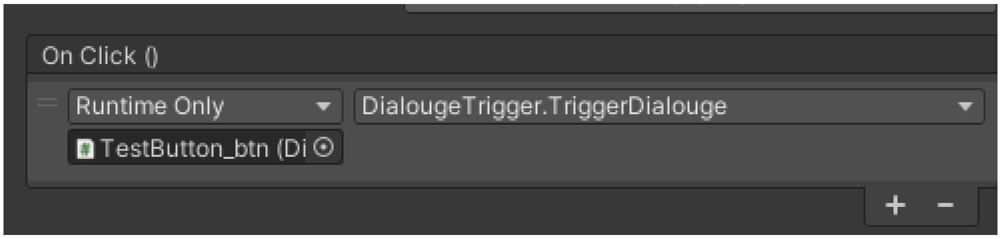
For further details, there are descriptions written inside each script going into detail on how the code works.

- Implementation

To activate the dialogue box, all you need is to call the ***TriggerDialogue()*** function within the ***Dialogue Trigger*** script. Included with the prefab should be an example of this in practice, ***TestButton_btn*** (shown below)



All this button does is call the ***DialogueTrigger.TriggerDialogue*** function when clicked. Calling this function is all you need to trigger the dialogue box.


On Click ()

Runtime Only | DialougeTrigger.TriggerDialouge

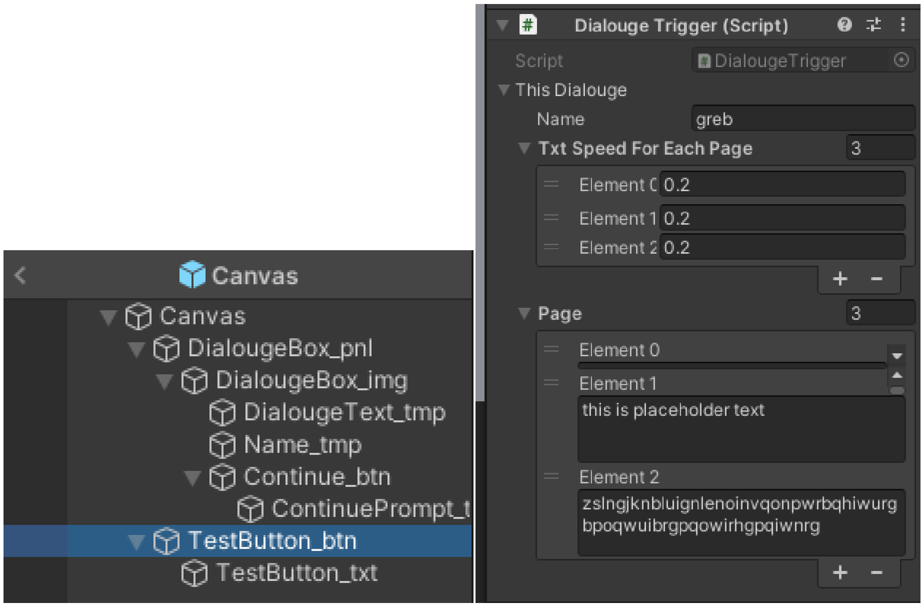
TestButton_btn (Di

To change what objects can trigger dialogue, all you need to do is first attach the ***Dialogue Trigger*** script to whatever it is you want to house the dialogue and then call ***DialogueTrigger.TriggerDialogue*** through whatever means you wish. The ***Dialogue Trigger*** script can even be put on multiple different objects, but you will need to insert the Text, Text Speed, Name, and # of Pages on each object.

- Adjusting Text, Text Speed, Name, and Pages

Changing the Text, Text Speed, Name, and Pages can all be done in the inspector of whatever object the ***Dialogue Trigger*** script is attached to. In the example provided in the prefab, the ***Dialogue Trigger*** script is attached to ***TestButton_btn***, so changing these parameters can be done there.

NOTE: Make sure that the # of *Txt Speed For Each Page* elements is the same as the # of *Page* elements. For example, if you have 2 pages of dialogue then you need to make sure that you have a text speed set for each page.



Canvas
- Canvas
  - DialougeBox_pnl
    - DialougeBox_img
      - DialougeText_tmp
      - Name_tmp
      - Continue_btn
        - ContinuePrompt_t
  - TestButton_btn
    - TestButton_txt

Dialouge Trigger (Script)

Script: DialougeTrigger

This Dialouge

Name: greb

Txt Speed For Each Page: 3

Element 0: 0.2

Element 1: 0.2

Element 2: 0.2

Page: 3

Element 0

Element 1: this is placeholder text

Element 2: zslngjknbluignlenoinvqonpwrbqhiwurg bpoqwuibrgpqowirhgpqiwnrg

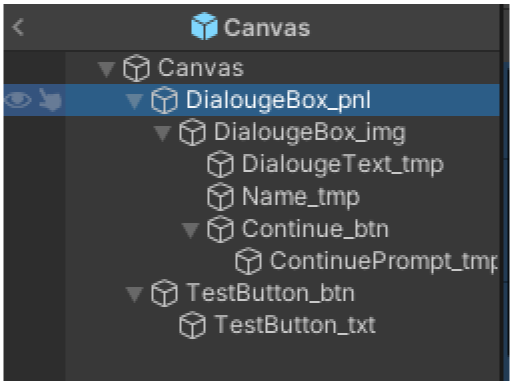
- Changing Visuals

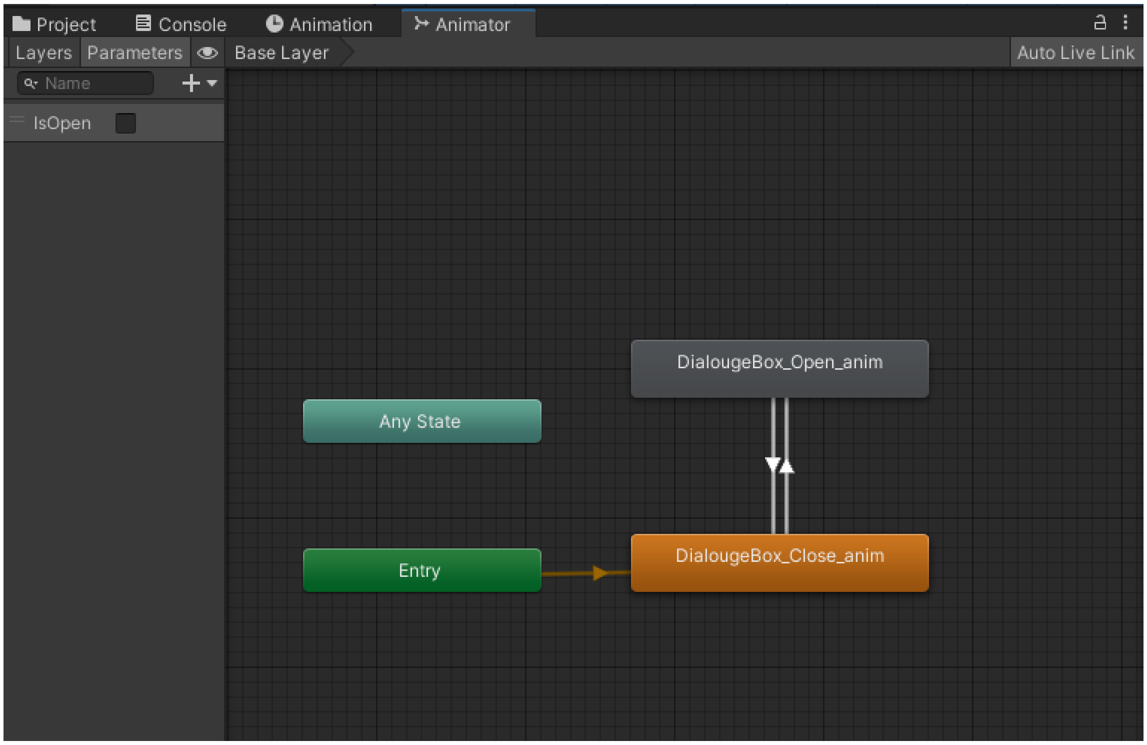
Visuals can be easily subbed out and changed to whatever you want. However, if you completely replace either ***DialougeText_tmp*** or ***Name_tmp***, make sure you put their replacements into the appropriate ***DialogueManager*** fields.

The continue button can also be altered or even replaced entirely. All that is needed to the next page of dialogue is for whatever new method you use to call the ***DialogueManager.DisplayNextSentance*** function.

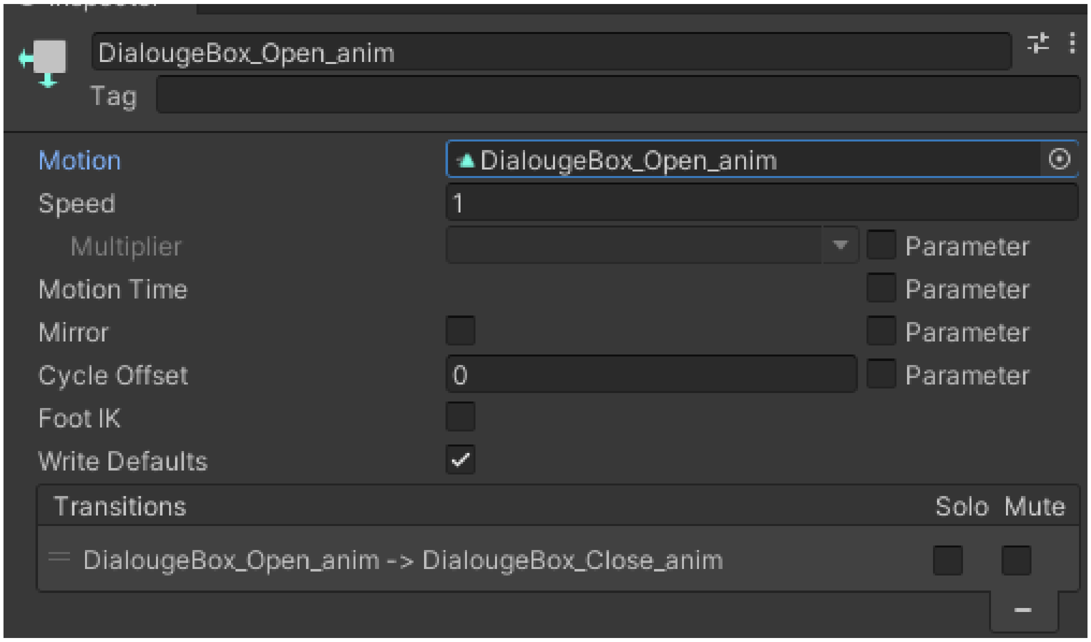
- Changing Animations

The animations of opening and closing the dialogue box can be substituted freely as well. To do this, open the Animator tab *(Window -> Animation -> Animator)* while ***DialogueBox_pnl*** is selected in the hierarchy.





From here, you can select either ***DialougeBox_Open_anim or DialougeBox_Close_anim***, In either one, simply drag and drop whatever new animation you want to use in their ***Motion*** field.



DialougeBox_Open_anim

Tag

Motion: DialougeBox_Open_anim

Speed: 1

Multiplier □ Parameter

Motion Time □ Parameter

Mirror □ □ Parameter

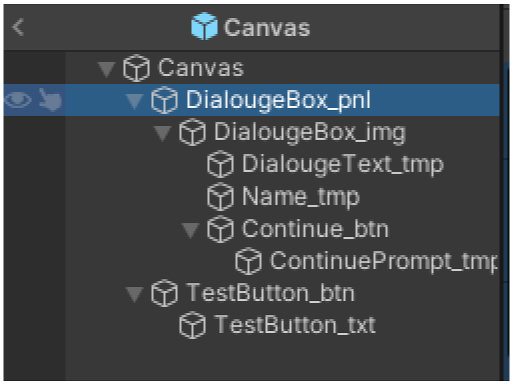
Cycle Offset: 0 □ Parameter

Foot IK □

Write Defaults ☑

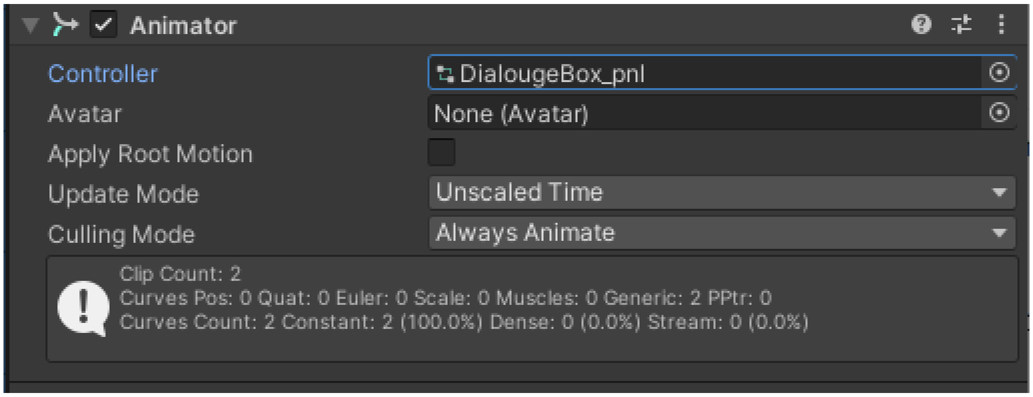
Transitions	Solo	Mute
DialougeBox_Open_anim -> DialougeBox_Close_anim	□	□

Alternatively, if you want to use a custom Animator you can put it in the ***Controller*** field on the Animator for ***DialougeBox_pnl***.



Canvas

- Canvas
  - DialougeBox_pnl
    - DialougeBox_img
      - DialougeText_tmp
      - Name_tmp
      - Continue_btn
        - ContinuePrompt_tmp
  - TestButton_btn
    - TestButton_txt



Animator

Controller: DialougeBox_pnl

Avatar: None (Avatar)

Apply Root Motion □

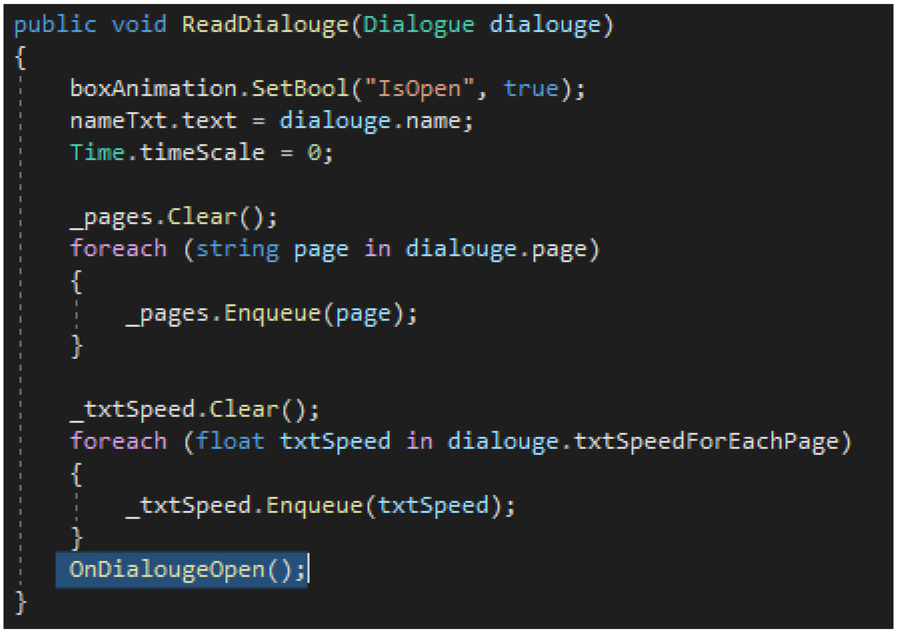
Update Mode: Unscaled Time

Culling Mode: Always Animate

```
Clip Count: 2
Curves Pos: 0 Quat: 0 Euler: 0 Scale: 0 Muscles: 0 Generic: 2 PPtr: 0
Curves Count: 2 Constant: 2 (100.0%) Dense: 0 (0.0%) Stream: 0 (0.0%)
```

I have also included a function within the ***Trigger Dialogue*** called ***OnDialougeOpen()***. This function is what starts the text-typing animation and can be manually called whenever you want this animation to start playing – say, for example, in a custom animator for the dialogue box. All you need to do is call ***OnDialougeOpen()*** whenever you want the typing animation to start.

Currently, the code is set up so that ***OnDialougeOpen()*** is called at the same time the dialogue box opens. If you want to call this function elsewhere instead manually, make sure you go into the ***Trigger Dialogue*** script and delete the function call highlighted below.



```
public void ReadDialouge(Dialogue dialouge)
{
    boxAnimation.SetBool("IsOpen", true);
    nameTxt.text = dialouge.name;
    Time.timeScale = 0;

    _pages.Clear();
    foreach (string page in dialouge.page)
    {
        _pages.Enqueue(page);
    }

    _txtSpeed.Clear();
    foreach (float txtSpeed in dialouge.txtSpeedForEachPage)
    {
        _txtSpeed.Enqueue(txtSpeed);
    }
    OnDialougeOpen();
}
```