

WordBox

This is a simple asset that is mainly about the dialog box sprites. However, it also includes our small WordBox tool, which is used to quickly and easily pop up dialog boxes per script or Timeline, in which the letters are printed piece by piece. The function call will be passed a callback function which will be called as soon as all letters have been printed.

Color Legend:

- **purple = important information**
- **blue = unity directory or file path**
- **blue and underlined = link for website**
- **green = Scene path**

Folder structure:

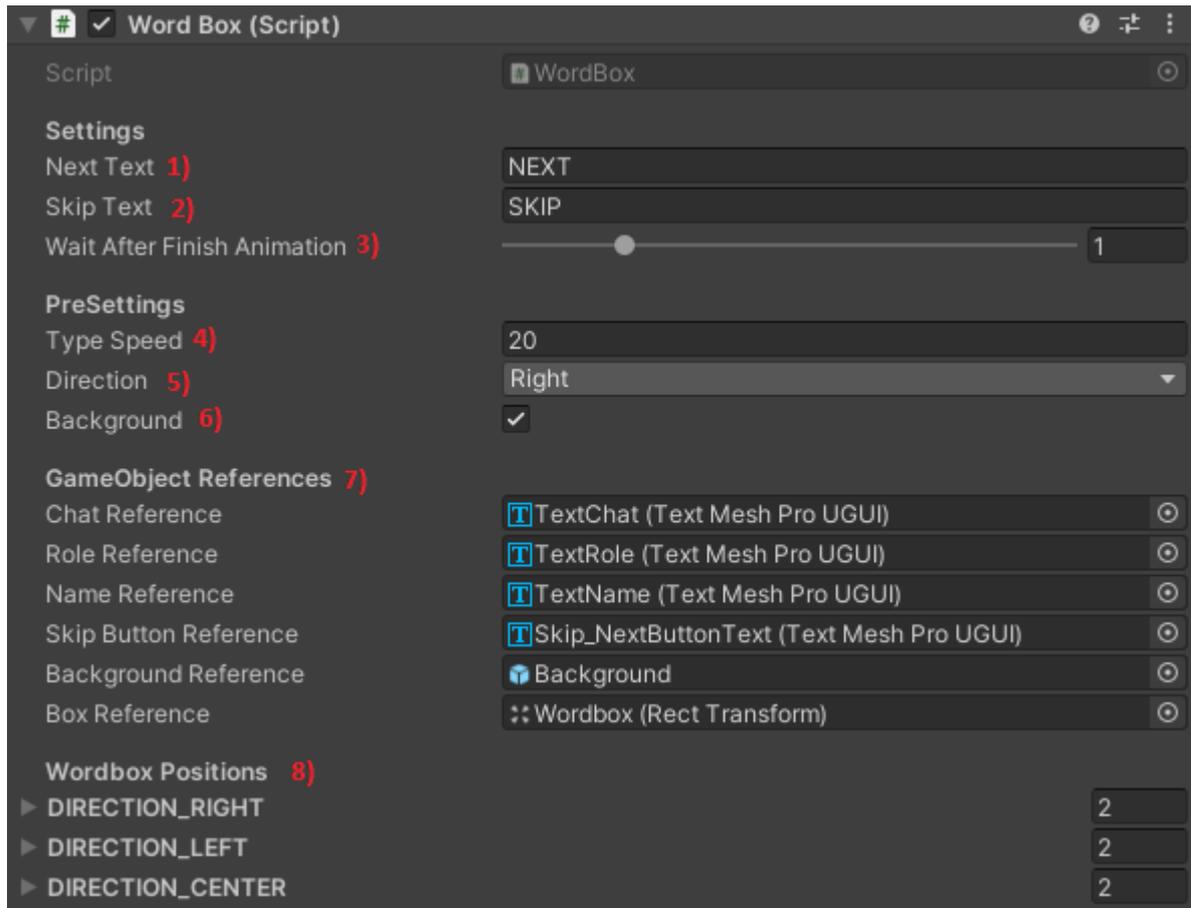
- Art
 - Dialog Boxes Sprites: [/Assets/49Wares/Art/GUI/DialogBoxes](#)
 - Background Sprites: [/Assets/49Wares/Art/GUI/Backgrounds](#)
 - Icons Sprites: [/Assets/49Wares/Art/GUI/Icons](#)
- Prefabs: [/Assets/49Wares/Systems/WordBox/Prefabs](#)
- Examples
 - Example Scene: [/Assets/49Wares/Systems/WordBox/ExampleScene](#)
 - Example Script: [/Assets/49Wares/Systems/WordBox/Scripts/ExampleScene](#)
- WordBox Script: [/Assets/49Wares/Systems/WordBox/Scripts/Core](#)
- Timeline Scripts: [/Assets/49Wares/Systems/WordBox/Scripts/Timeline](#)

Quick Start:

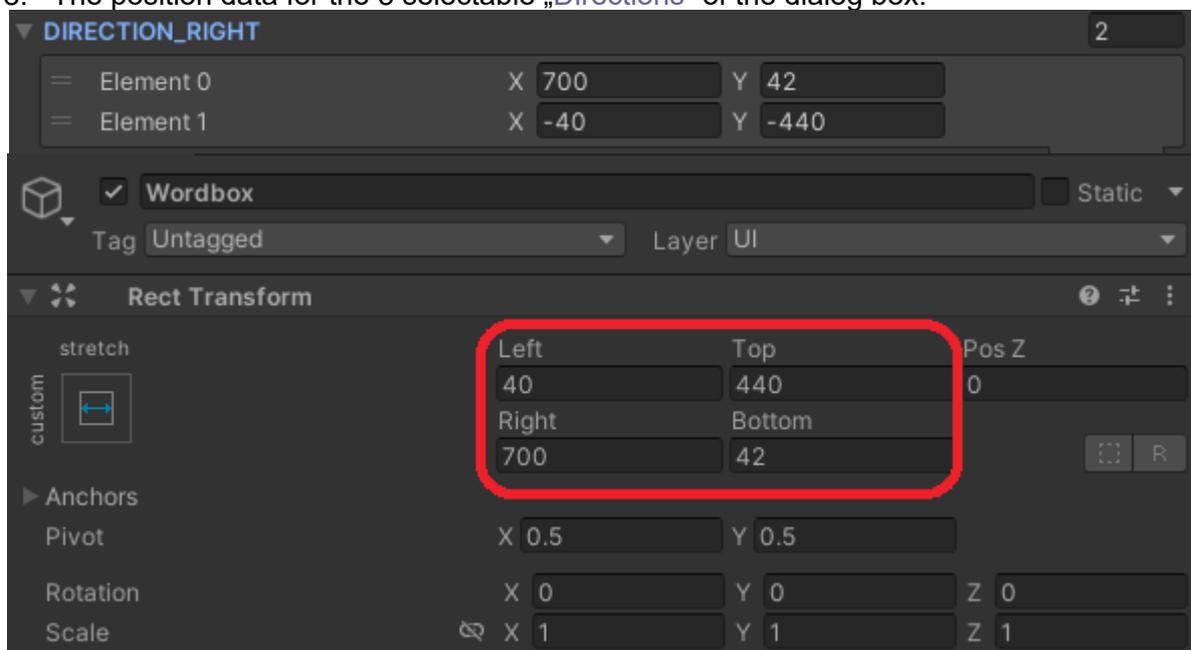
Take the wordbox prefab and drag it into your scene. Adjust everything as you need it and start a wordbox by getting the „[WordBox](#)“ component and calling the function „[CreateWordBox\(\)](#)“ on it.

You can check out the „[ExampleScene](#)“ and the „[WordBoxExample.cs](#)“ script to see how a possible way to call the wordbox by code can work.

Word Box Settings:



1. Text shown in the „Next-Button“ (The Button that the player can use to see the next WordBox through triggering the callback function).
2. Text shown in the „Skip-Button“ (The button with which the player can skip the writing animation from the current wordbox).
3. Seconds to wait before showing the „Next-Button“ or executing the callback function.
4. The speed in letters/second at which the letters shall be written. You can choose a very high value to print all Letters at once.
5. The location where the dialog box should be displayed.
6. Defines whether the background should be active or not.
7. The references for all GameObjects needed by the „WordBox.cs“ script. So if you want to create your own WordBox gameobjects remember to add these references.
8. The position data for the 3 selectable „Directions“ of the dialog box.



The position data you enter there will be transferred as follows:

Element 0, X => Rect Transforms Right value

Element 0, Y => Rect Transforms Bottom value

-1 * (Element 1, X) => Rect Transforms Left value

-1 * (Element 1, Y) => Rect Transforms Top value

Calling Word Box from Script:

- Drag the WordBox prefab into your scene
- Create a script that get the „WordBox Component“ of the „WordBox prefab“ in your Scene
- Write a callback method that you want to be called after the WordBoxed finished printing letters
- Call the „CreateWordBox()“ method from your script

„CreateWordBox()“ has the following parameters:

- „tchat“ = the text which shall be printed out.
- „callback“ = the callback function which should be executed after finishing the dialog box.
- „backgroundOn“ = defines whether the background should be on or not.
- „tname“ = the name of the person speaking.
- „trole“ = the role of the person speaking.
- „customTypeSpeed“ = the speed in letters/second at which the letters shall be written.
- „customDirection“ = the location where the dialog box should be displayed.
- „nextbutton“ = if enabled the callback will only be executed after the user presses the next button.
- „timeToHold“ = if used it will replace the „waitAfterFinishAnimation“ variable and wait n seconds after all letters are printed before executing the callback function

The parameters „tchat“ and „callback“ are non-optional parameters. All other parameters are optional.

Calling Word Box from Timeline:

- Drag the WordBox prefab into your scene
- Select the WordBox GameObject and set [WordBox/MainPanel/Wordbox/Skip](#) inactive
- Create an empty GameObject
- Assign a PlayableDirector component to the Gameobject.
- Create a Timeline asset
- Do a right click on the track side and choose „Timeline Word Box Track“
- Do a right click on the clip side and choose „add Timeline Word Box Asset“
- Click on the clip and put your text into the field „Text“
- Right click the clip and choose „Editing→Reset Duration“
- Set all other clip settings as you wish
- Make sure the clips duration is always exactly as recommended (no „Hold“ and not too short).
- Make sure the clips are not overlapping each other

More Information:

- You can simply add your own Background sprite by replacing the sprite on [WordBox/MainPanel/Background](#).
- You can add your own dialog box sprite by replacing the sprite on [WordBox/MainPanel/Wordbox](#) and changing the „Wordbox Positions“ Settings „DIRECTION_RIGHT“, „DIRECTION_LEFT“ and „DIRECTION_CENTER“ as explained above (WordBox Settings nr. 8).
- You can change the Background sprites color on the [WordBox/MainPanel/Background](#) „Image“ components Color picker.

- You can change the Icon next to the Skip/Next Button by replacing the sprite on [WordBox/MainPanel/Wordbox/Skip](#).
- You can change the position and size of „TextChat“, „TextName“ and „TextRole“ freely as you wish without the need to change any values on the WordBox Component.

Player to NPC Dialog:

Even though this asset is not meant for that, here is a small explanation in case you still want to use it that way. You can theoretically just add more buttons in the wordbox and then use their onClick events to call public functions from specific scripts. Either you create an empty gameobject and add a script to it that contains the public functions that should do something specific in case of certain answers. Or you do this directly in the script in which you also call the dialog box and have your callback functions. You can also deactivate the „Next“ button without any problems.

Troubleshooting:

- Please check if you are on the newest asset version, by opening the package manager window and choosing „[packages:My Assets](#)“. From this list choose our asset and check if there is an update button on the bottom right side of the window.
- Check if you have a valid „MainCamera“ in the scene and if there are any problems. The asset always takes the „MainCamera“ of the scene and tries to display on it. If you prefer to change how the script selects the camera to render, create a public „Camera“ variable in the script „[WordBox.cs](#)“ and then change line 58 to „GetComponent<Canvas>().worldCamera = {your_camera_variable_name};“
- If errors occur with the timeline clips, check that the clips do not overlap each other and select „[right-click](#)→[Editing](#)→[Reset Duration](#)“ on each clip.
- If after the previous tip you still have errors when using timeline and clips please try changing the files [„/Assets/49Wares/Systems/WordBox/Scripts/Timeline/TimelineWordBoxAsset“](#) line 37 from „else return ((double)text.Length / typeSpeed) + waitSecondsAfterFinishTyping + 0.05f;“ to „else return ((double)text.Length / typeSpeed) + waitSecondsAfterFinishTyping + 0.5f;“

If none of these tips helped, join our discord so we can help you: <https://discord.gg/qzz2DrU9nV>