EASY INSTRUCTION DIALOGUE

The simplest way of adding Instruction or Dialogue to your project

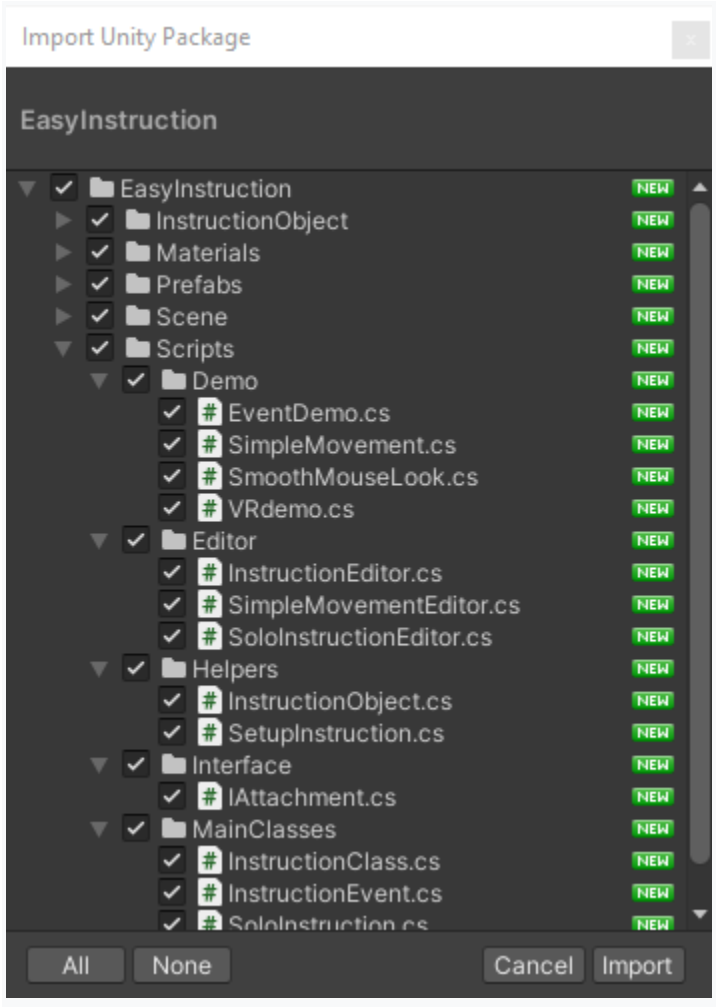
USER GUIDE

Release 1.0.0

May 2022

Quick setup procedure to get started with usage

Begin by Importing the EasyInstructionDialogue.package.



For Automatic Setup, you may refer to: (Automatic Setup Video) for better understanding

The steps include:

1. Drag any of the prefabs in (Asset > EasyInstructionDialogue > Prefabs) into the scene. This should create a InstructionManager GameObject with InstructionClass attached and all the references from the canvas prefab are properly set. The GameObject would also have InstructionEvent attached.
2. If you would be using InstructionObject Type, Create the InstructionObject anywhere in Asset folder or subfolders and reference it to the InstructionClass. If you would be using CustomInstruction, refer to Page 7-8

3. Call ShowInstruction() method on the InstructionClass from anywhere (refer to page 7) to see how the method is called properly.
4. Play and test instruction.
5. You may Setup Open-Close animation with the Open & Close Delay along with the InstructionEvent for extra polish. (refer to page 6 - Animated Panel Definition, and also page 9)

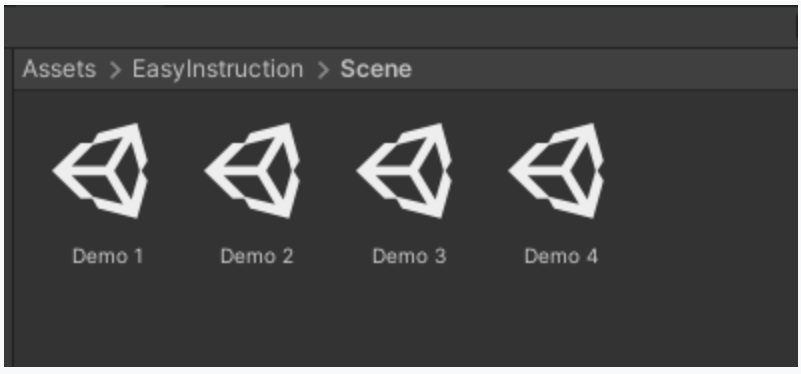
For Maual Setup, you may refer to: (Manual Setup Video) for better understanding

The steps include:

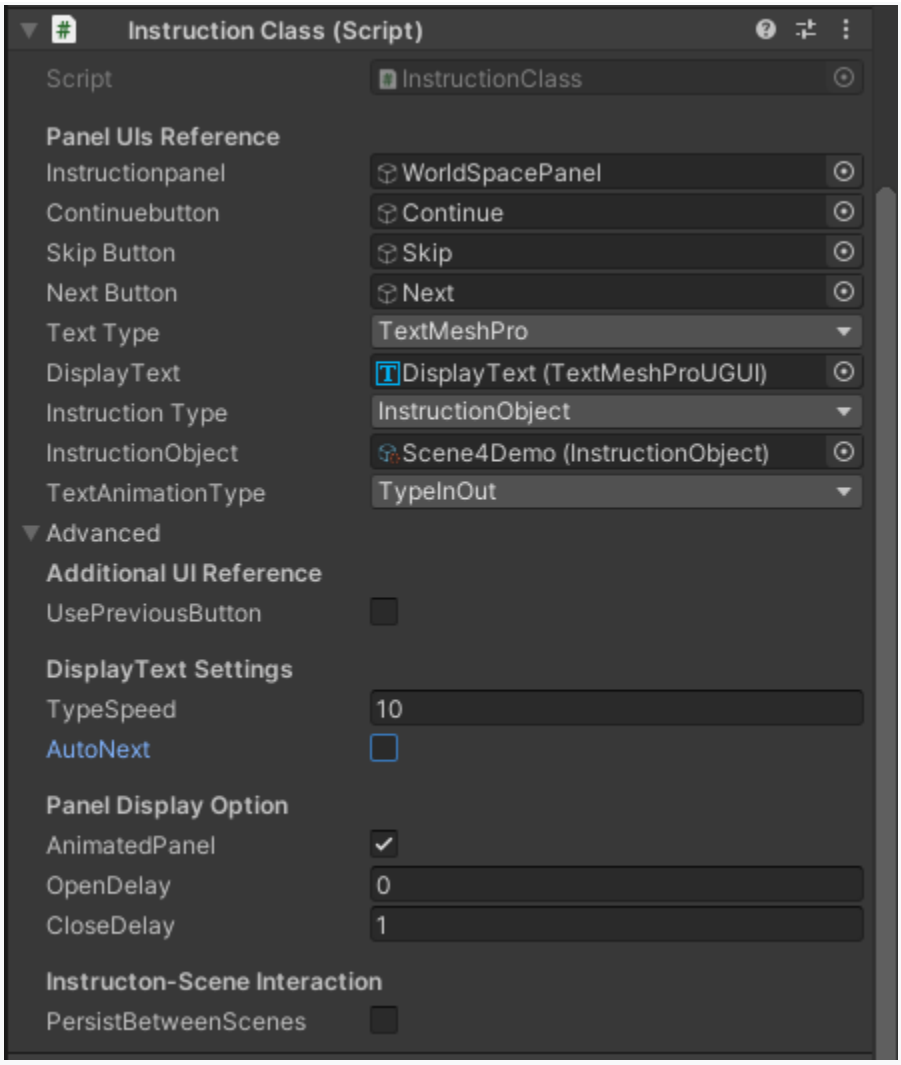
1. Design a Dialogue/Instruction Window which should have buttons for Skip, Next, Close/Continue and optionally previous. It is advised to put this window on a separate Canvas. Then create a new GameObject with the name "InstrutionManager" (you can name it anything you want) and attach the InstructionClass to it. Then references all the Panel UI from the Canvas. You may also add InstructionEvent to the gameObject.

Follow the same steps 2-5 of the automatic Setup above.

Open any of the Demo scenes in (Asset > EasyInstructionDialogue > Scenes) and check out how it works.

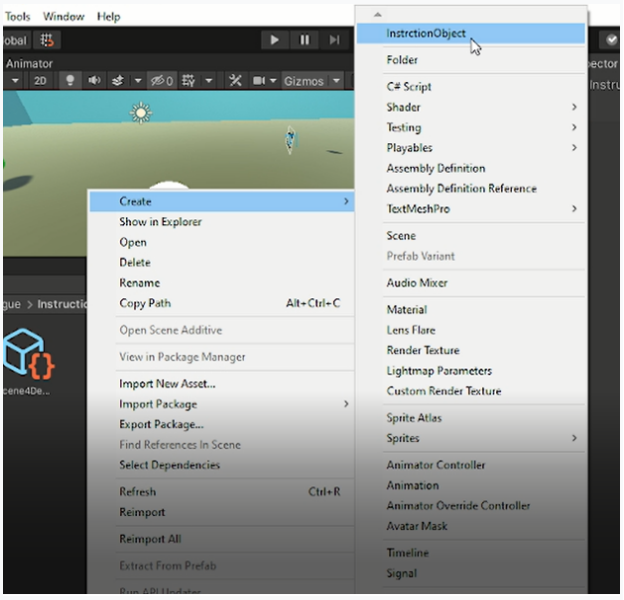


Understanding the variables of the InstructionClass



- Instruction Panel: This holds the reference to the Canvas that contains the display window, text, and others
- Continue Button: This holds the reference to a UI GameObject **(with a button attached)** that closes the panel (disable the Canvas) after all the instruction is displayed at that moment. It could be named Close, Continue or even be an Icon.

- Skip Button: This holds the reference to a UI GameObject **(with a button attached)** that closes the panel (disable the Canvas) after any instruction is displayed. However, it is different from Continue in the sense that, for multiple instructions, it shows from the beginning as opposed to the end.
- Next Button: This holds the reference to a UI GameObject **(with a button attached)** that shows the following dialogue (when there are multiple dialogues).
- TextType: An enum to switch between using Unity's legacy UI text or TextmeshPro
- DisplayText: This is a reference to the text to be displayed which may be a legacy UI Text or TextmeshProUIGUI based on the value of the selected enum above.
- InstructionType: This defines how you want your instructions to be stored either in a single list of sentences or Individually from different objects
- Instruction Object: This field only appears if the InstructionObject Type is chosen above. And it is a reference to the Instruction Object which can be created in the project windows by **Left-Click, Create > InstructionObject**



Then fill your Instruction/Dialogue sentences in it.

- TextAnimationType: This is a list that contains varieties of options to animate the display text
- Advanced Drop-drown Menu: This shows some more advanced settings for the InstructionClass
- UsePreviousButton: When enabled, Shows a field that references a UI GameObject **(with a button attached)** which allows the player to go to a previous instruction or dialogue in a series of Instructions. This GameObject is Optional.
- Type Speed and Fade Rate: This field shows based on the Text animation type selected above, and allows you to control the speed of the text animations.
- Auto-Next: This is an option that enables the Instruction or dialogue to move to the next one in the list automatically after a few seconds. When enabled, Shows a field that allows you to specify the time delay before moving to the next dialogue.
- AnimatedPanel: When enabled, shows you two fields that allow you to specify the
  - Open delay i.e how long it should wait from when ShowInstruction() (which would enable the InstructionCanavas) is called till it starts displaying the text.
  - CloseDelay i.e how long it should wait from when skip or continue is pressed (which should disable the canvas) till the Canvas actually disables.

  This is especially useful for animating the panel in and out when combined with the InstructionEvent.
- PersistBetweenScenes: When enabled the InstructionManager object and the InstructionCanvas would not be destroyed when switching scenes.

Calling Instructions

Instruction is easily displayed by referencing the InstrutionManager GameObject in the editor to say a button for example and calling **ShowInstructions(int i) where i is the InstructionIndex for** example **ShowInstruction(3)** or call **ShowInstruction(string s) where s is a string that is in the format Instruction StartingIndex-StopingIndex** for example "2-4" in use **ShowInstruction(2-4).**

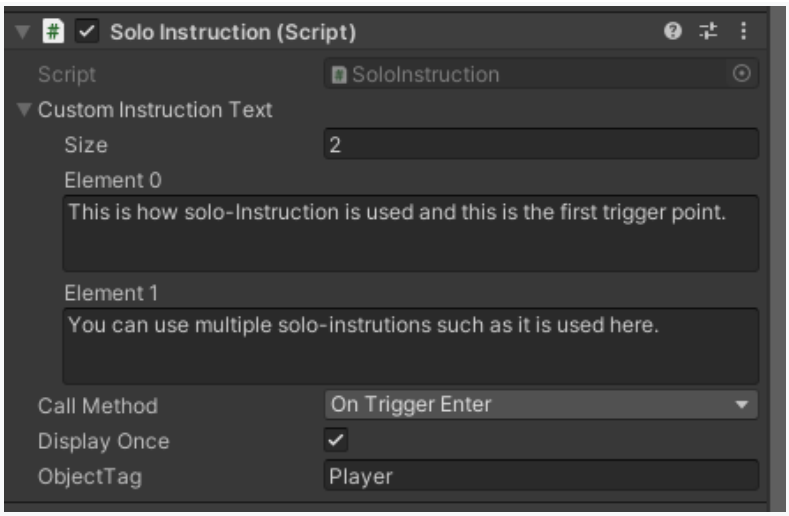
The ShowInstruction() method can also be called from another Class with its static instance such as **Instructions.options.ShowInstruction(int i)** or **Instructions.options.ShowInstruction(string s)** or **Instructions.options.ShowInstruction(int start,int stop)** where "start" means starting Index and "stop" for stopping Index.

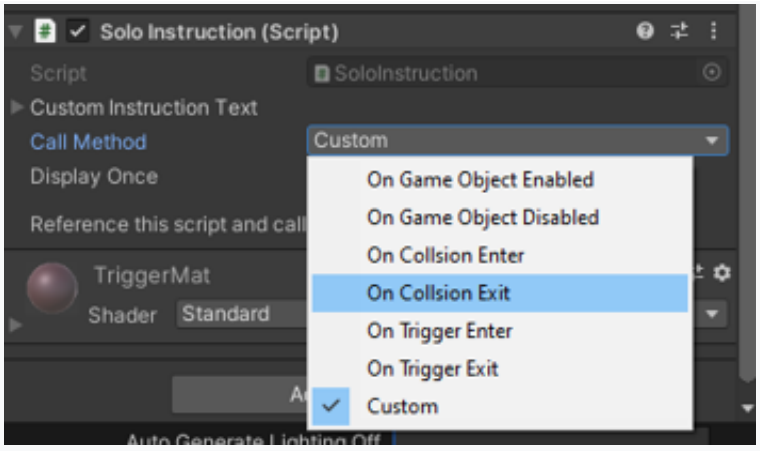
Custom or SoloInstruction

If the InstrutionType selected in the InstructionClass is CustomInstruction, Then we intend to store the instruction on different objects and call them when the player interacts with them. This can be easily achieved by the SoloInstuction Class.

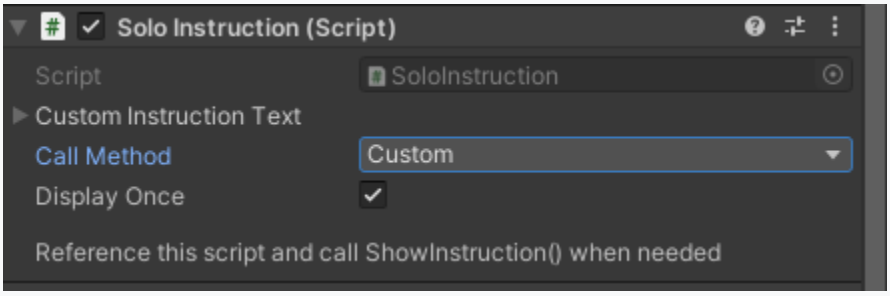
How to use

1. Attach the SoloInstruction to any Object
2. Input the Instruction or Dialogue text (maybe single or multiple)
3. Select call method. Note that for Physics-based activations i.e Collision/Trigger Enter and Exit, A field is shown which enabled you to specify the tag of the player it should check for Collisions/Trigger against.





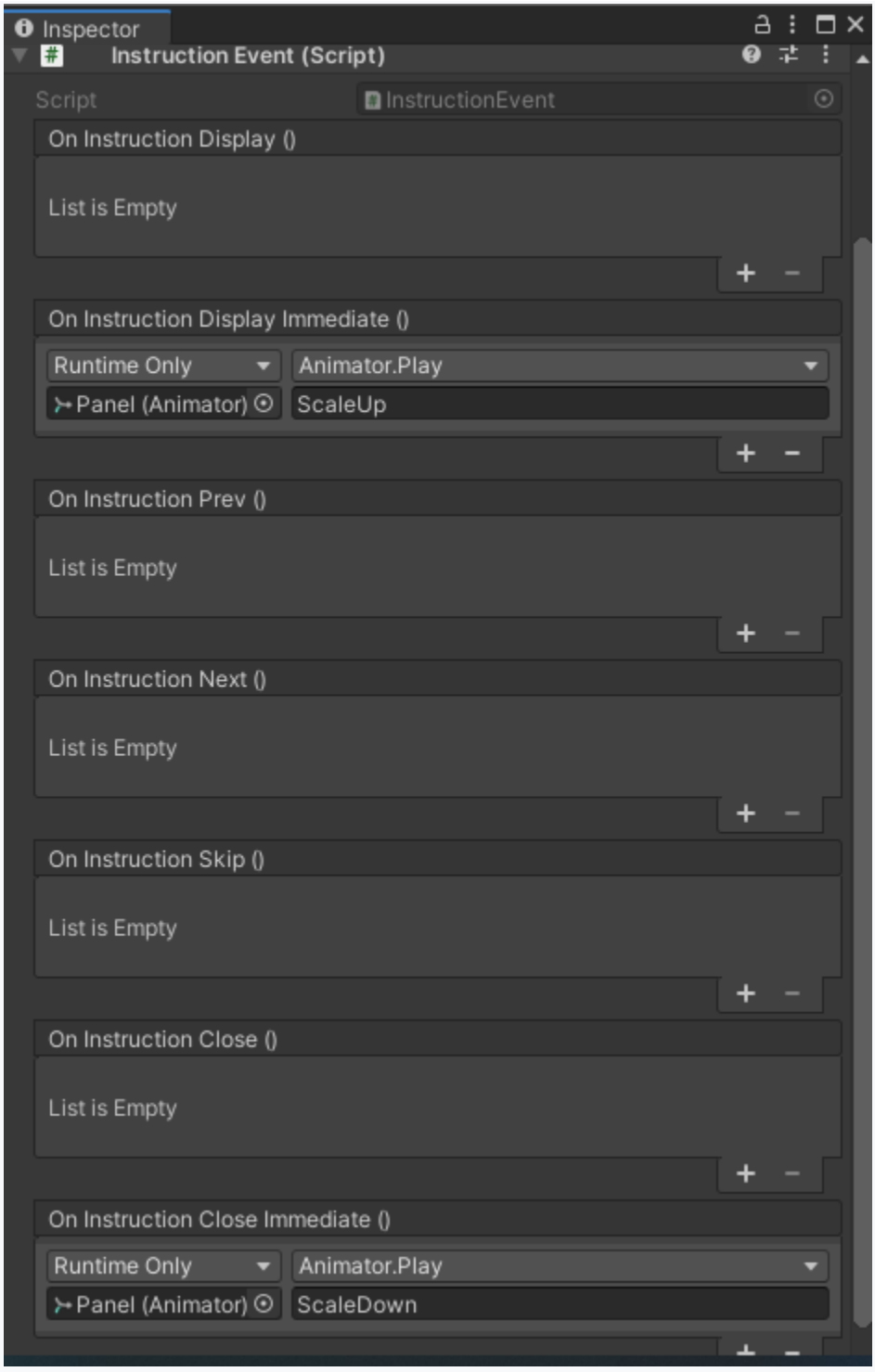
4. If any activation method you desire is not included in the list, you can select custom and use reference to call **ShowInstruction(int i)** or **ShowInstruction(string s)** on the SoloInstruction.



Refer to Demo Scene 2 to checkout how its is used or check out Custom Instruction Demo

InstructionEvent

This is a series of event that is fired off with certain conditions of an InstructionClass.



OnDisplayImmediate is called immediately ShowInstruction() is called (before text displays) This is useful to add an Open animation to the InstructionPanel given an slight open delay.

OnCloseImmediate is called immediately the skip or continue button pressed before the Canvas is disabled if there is a slight close delay. This is useful to add an Close animation to the InstructionPanel.

Refer to Demo Scene 3 to see how instruction event is used extensively or check out Instruction Event Demo