Tuto: VR Inputs Setup (scripting needed)

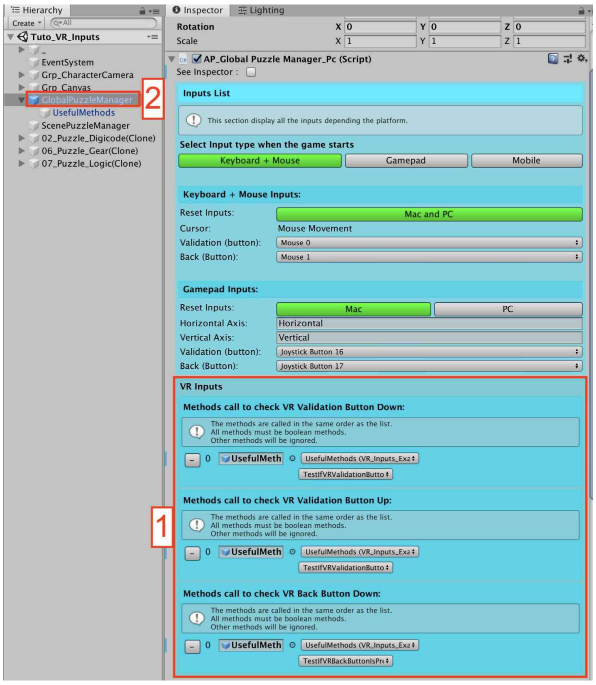
Overview:

Each VR headset manage its Inputs differently.

***Info**: To use VR Gamepad it is needed to add custom code to detect your VR gamepad Inputs.*

In the Input Puzzle manager (spot 1):
You must add your own script to use VR gamepad.

***Info**: To access this section:*
-Select GlobalPuzzleManager in the Hierarchy (spot 2)
-In the Inspector go to section VR Input (spot 1).



Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

☑ AP_Global Puzzle Manager_Pc (Script)

See Inspector : □

Inputs List

This section display all the inputs depending the platform.

Select Input type when the game starts

Keyboard + Mouse | Gamepad | Mobile

Keyboard + Mouse Inputs:

Reset Inputs: Mac and PC
Cursor: Mouse Movement
Validation (button): Mouse 0
Back (Button): Mouse 1

Gamepad Inputs:

Reset Inputs: Mac | PC
Horizontal Axis: Horizontal
Vertical Axis: Vertical
Validation (button): Joystick Button 16
Back (Button): Joystick Button 17

VR Inputs

Methods call to check VR Validation Button Down:

The methods are called in the same order as the list.
All methods must be boolean methods.
Other methods will be ignored.

– 0 UsefulMeth ⊙ UsefulMethods (VR_Inputs_Exa
TestIfVRValidationButto

Methods call to check VR Validation Button Up:

The methods are called in the same order as the list.
All methods must be boolean methods.
Other methods will be ignored.

– 0 UsefulMeth ⊙ UsefulMethods (VR_Inputs_Exa
TestIfVRValidationButto

Methods call to check VR Back Button Down:

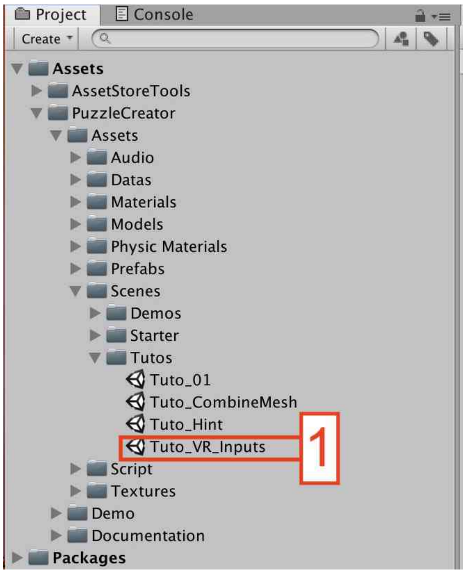
The methods are called in the same order as the list.
All methods must be boolean methods.
Other methods will be ignored.

– 0 UsefulMeth ⊙ UsefulMethods (VR_Inputs_Exa
TestIfVRBackButtonIsPr

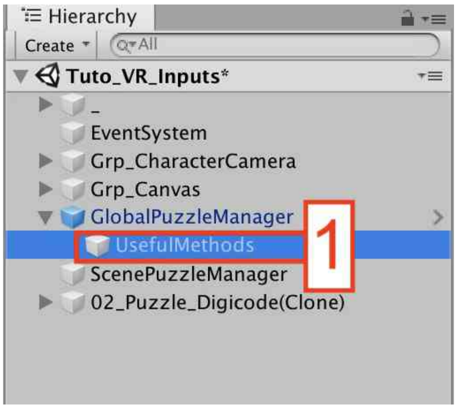
Example:

***Info**: In the following example we will show:*
-How to add a method to detect that the player presses Down the validation button on his VR controller
-How to add a method to detect that the player presses Up the validation button on his VR controller
-How to add a method to detect that the player presses Down the back button on his VR controller.

-Open the scene Tuto_VR_Inputs (spot 1).
(Project Tab: PuzzleCreator – Assets → Scenes → Tutos → Tuto_VR_Inputs)



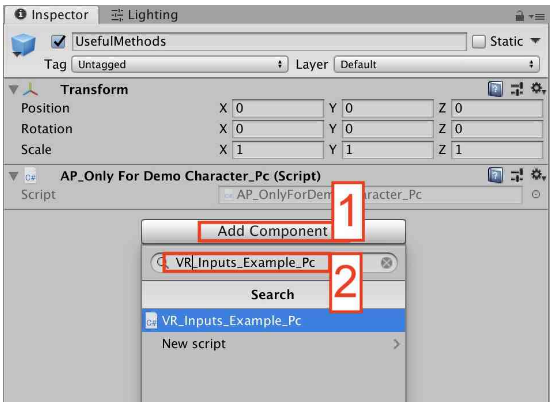
-Select the object UsefulMethods in the Hierarchy (spot 1).
(Hierarchy → GlobalPuzzleManager → UsefulMethods)



-In the Inspector press button Add Component (spot 1).

-Write VR_Inputs_Example_Pc in the search field (spot 2)

-Press Enter.



☑ UsefulMethods

Tag Untagged Layer Default

Position X 0 Y 0 Z 0

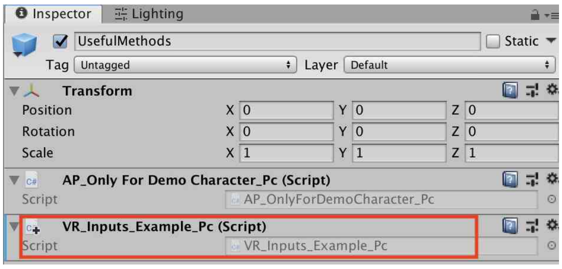
Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

Script AP_OnlyForDem[o]...racter_Pc

VR_Inputs_Example_Pc

***Info**: Script VR_Inputs_Example_Pc is now attached to UsefulMethods object. This script contains ready to use methods to help you detect the VR gamepad Inputs.*


Inspector
Lighting
Static
Transform
AP_Only For Demo Character_Pc (Script)
VR_Inputs_Example_Pc (Script)

☑ UsefulMethods

Tag Untagged Layer Default

Position X 0 Y 0 Z 0

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

Script AP_OnlyForDemoCharacter_Pc

Script VR_Inputs_Example_Pc

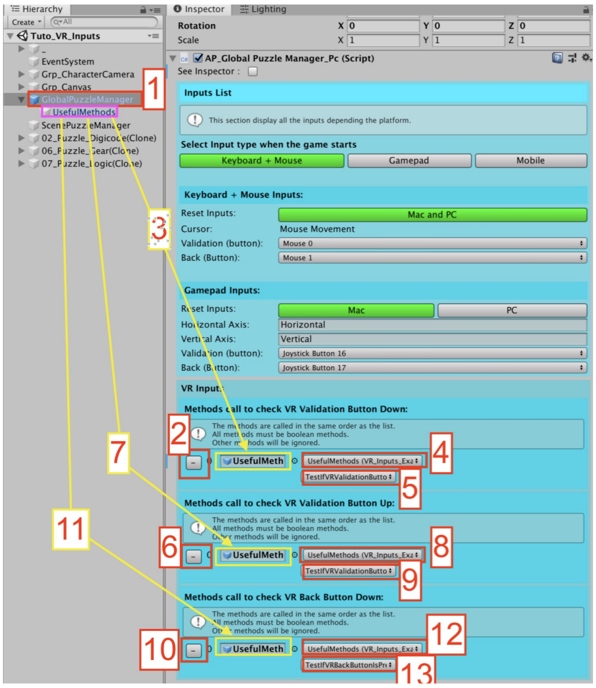
-In Hierarchy select GlobalPuzzleManager (spot 1).

-In the inspector section VR Inputs press the button + in the sub section Methods call to check VR Validation button Down(spot 2).
*(**Info**: This section is used to know if the player is pressing down the validation button on his VR gamepad.)*

-Drag and drop the object named UsefulMethods in the empty slot (spot 3).

-In the dropdown menu (spot 4) select the script VR_Inputs_Example_Pc.

-In the dropdown menu (spot 5) select the method TestIfVRValidationButtonIsPressed_Down.
*(**Info** The boolean method TestIfVRValidationButtonIsPressed_Down check in this example if the button S is pressed down)*



Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

☑ AP_Global Puzzle Manager_Pc (Script)
See Inspector : □

Inputs List

This section display all the inputs depending the platform.

Select Input type when the game starts
Keyboard + Mouse | Gamepad | Mobile

Keyboard + Mouse Inputs:
Reset Inputs: Mac and PC
Cursor: Mouse Movement
Validation (button): Mouse 0
Back (Button): Mouse 1

Gamepad Inputs:
Reset Inputs: Mac | PC
Horizontal Axis: Horizontal
Vertical Axis: Vertical
Validation (button): Joystick Button 16
Back (Button): Joystick Button 17

VR Inputs:

Methods call to check VR Validation Button Down:
The methods are called in the same order as the list.
All methods must be boolean methods.
Other methods will be ignored.
UsefulMeth | UsefulMethods (VR_Inputs_Exa | TestIfVRValidationButto

Methods call to check VR Validation Button Up:
The methods are called in the same order as the list.
All methods must be boolean methods.
Other methods will be ignored.
UsefulMeth | UsefulMethods (VR_Inputs_Exa | TestIfVRValidationButto

Methods call to check VR Back Button Down:
The methods are called in the same order as the list.
All methods must be boolean methods.
Other methods will be ignored.
UsefulMeth | UsefulMethods (VR_Inputs_Exa | TestIfVRBackButtonIsPr

___

-In the inspector section VR Inputs press the button + in the sub section Methods call to check VR Back button (spot 6).
*(**Info**: This section is used to know if the player is pressing Up the back button on his VR gamepad.)*

-Drag and drop the object named UsefulMethods in the empty slot (spot 7).

-In the dropdown menu (spot 8) select the script VR_Inputs_Example_Pc.

-In the dropdown menu (spot 9) select the method TestIfVRBackButtonIsPressed_Up.
*(**Info:** The boolean method TestIfVRValidationButtonIsPressed_Up check in this example if the button S is pressed Up)*

___

-In the inspector section VR Inputs press the button + in the sub section Methods call to check VR Back button Down (spot 10).
*(**Info**: This section is used to know if the player is pressing down the back button on his VR gamepad.)*

-Drag and drop the object named UsefulMethods in the empty slot (spot 11).

-In the dropdown menu (spot 12) select the script VR_Inputs_Example_Pc.

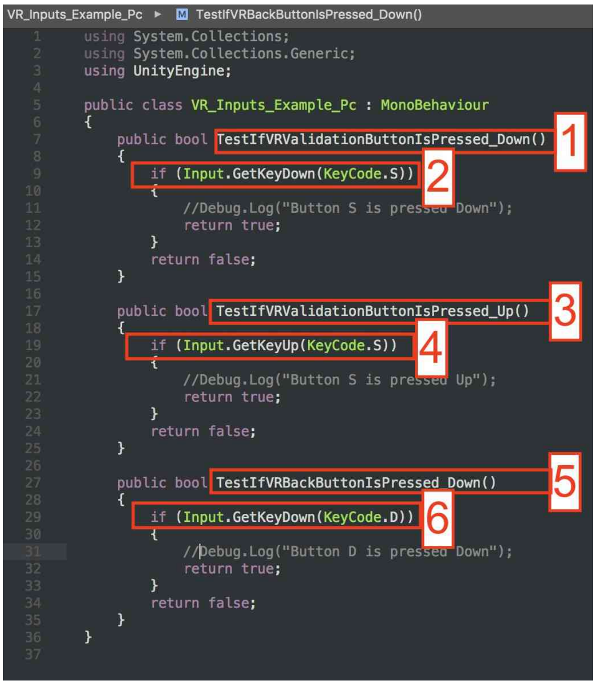
-In the dropdown menu (spot 13) select the method TestIfVRBackButtonIsPressed_Down.
*(**Info:** The boolean method TestIfVRBackButtonIsPressed_Down check in this example if the button D is pressed down)*

-Open the script VR_Inputs_Example_Pc in your script editor.
(Project Tab→ PuzzleCreator → Assets → Script → Demo → VR_Inputs_Example_Pc)

<u>The boolean method:</u> TestIfVRValidationButtonIsPressed_Down (spot 1) check in this example if the button S is pressed down (spot 2).

***Info**: If the player presses button S down the boolean method return true. If the player do not presses button S down the boolean method return false.*

<u>In your case</u>:
-Replace the line 9 with your code to detect the button on the VR gamepad *(This line of code depends of the Headset you are using)*.



```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class VR_Inputs_Example_Pc : MonoBehaviour
6   {
7       public bool TestIfVRValidationButtonIsPressed_Down()
8       {
9           if (Input.GetKeyDown(KeyCode.S))
10          {
11              //Debug.Log("Button S is pressed Down");
12              return true;
13          }
14          return false;
15      }
16
17      public bool TestIfVRValidationButtonIsPressed_Up()
18      {
19          if (Input.GetKeyUp(KeyCode.S))
20          {
21              //Debug.Log("Button S is pressed Up");
22              return true;
23          }
24          return false;
25      }
26
27      public bool TestIfVRBackButtonIsPressed_Down()
28      {
29          if (Input.GetKeyDown(KeyCode.D))
30          {
31              //Debug.Log("Button D is pressed Down");
32              return true;
33          }
34          return false;
35      }
36  }
37
```

<u>The boolean method:</u> TestIfVRValidationButtonIsPressed_Up (spot 3) check in this example if the button S is pressed up (spot 4).

***Info**: If the player presses button S up the boolean method return true. If the player do not presses button S up the boolean method return false.*

<u>In your case</u>:
-Replace the line 19 with your code to detect the button on the VR gamepad *(This line of code depends of the Headset you are using)*.

<u>The boolean method</u>: TestIfVRBackButtonIsPressed_Down (spot 5) check in this example if the button D is pressed (spot 6).

***Info**: If the player presses button D the boolean method return true. If the player do not presses button D the boolean method return false.*

<u>In your case:</u>
-Replace the line 29 with your code to detect the button on the VR gamepad *(This line of code depends of the Headset you are using)*.