

Complete First-Person Controller

Introduction

Thank you for purchasing the Complete First-Person Controller. This package was created in order to provide a functional First-Person character that could interact believably with the world around it. This controller can be fully customized to fit your game and will provide a realistic and responsive feel.

The code is commented very extensively with every line explained. This was done so that you can rewrite or change anything that you need to in order to fit your specific project. The script FPC_Actions.cs is intended to be built on top of so that any additional controls (attack, aim, interact) can be added in easily without compromising the rest of the movement scripts.

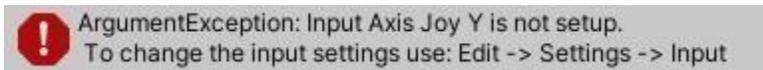
Included

This asset includes 4 Prefab Character Controller Variants as well as 14 scenery objects and an example scene. The Character prefabs mimic various popular FPS games and use the same scripts, but show the versatility of the variables that you can change. This also allows for a quicker set up if you just want something to work.

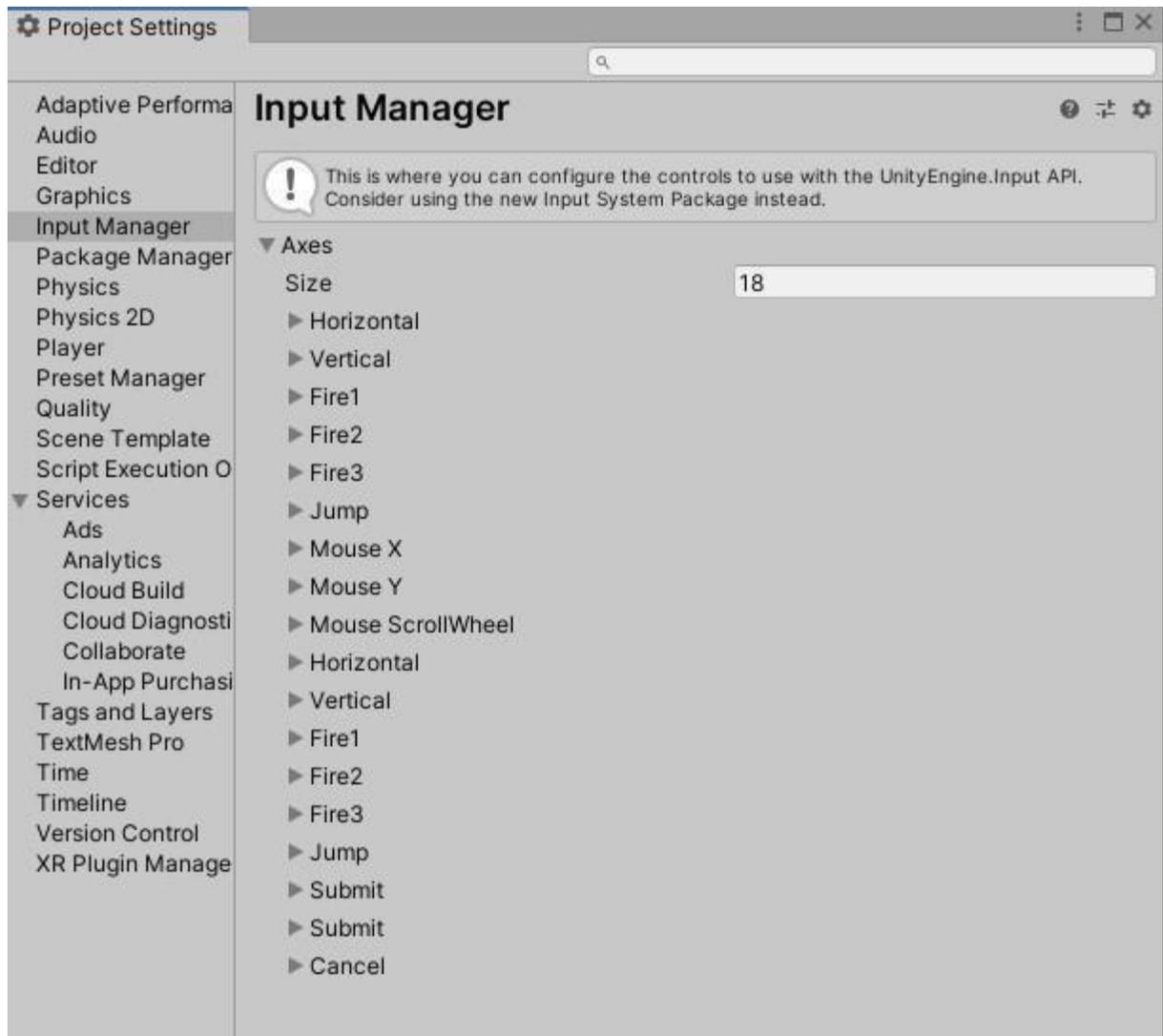
Included with the scenery items are prefabs for the water and ladder objects. These prefabs have colliders already set up so that the player can swim and climb. You don't necessarily need these prefabs in order for swimming and climbing to work, as long as the asset you create is named and tagged correctly. (See the *"Swimming and Climbing"* section for more details)

Setup

While this asset is mostly plug and play, there is one error that you will most likely encounter when you first import this asset to your game.



If you are seeing this error, it is because the joystick has not been set up in the Input Manager section of Project Settings. To fix this, new inputs will need to be added to the Input Manager. This menu can be found by selecting Edit -> Project Settings -> Input Manager, and looks like the screen below.



In order to get the Joystick working, you will need to create new inputs and name them accordingly. This can be done by changing the 'Size' variable to create additional inputs. There will need to be a total of 4 additional buttons, one for crouch, one for sprint, and 2 for joystick camera movement. Change the 'Size' variable to 22 (or its current number plus 4) and 4 additional buttons will appear at the bottom of the button list. The four new buttons will be duplications of the last button in the list.

See the reference images below for how to set up the various buttons.

*Note: The names in the 'Name' field are case-sensitive and must match the below images EXACTLY in order for the code to read them correctly.

<p>▼ Joy X</p> <p>Name: Joy X</p> <p>Descriptive Name: </p> <p>Descriptive Negative Name: </p> <p>Negative Button: </p> <p>Positive Button: </p> <p>Alt Negative Button: </p> <p>Alt Positive Button: </p> <p>Gravity: 0</p> <p>Dead: 0.35</p> <p>Sensitivity: 1</p> <p>Snap: <input type="checkbox"/></p> <p>Invert: <input type="checkbox"/></p> <p>Type: Joystick Axis</p> <p>Axis: 4th axis (Joysticks)</p> <p>Joy Num: Get Motion from all Joysticks</p>	<p>▼ Sprint</p> <p>Name: Sprint</p> <p>Descriptive Name: </p> <p>Descriptive Negative Name: </p> <p>Negative Button: </p> <p>Positive Button: left shift</p> <p>Alt Negative Button: </p> <p>Alt Positive Button: joystick button 8</p> <p>Gravity: 3</p> <p>Dead: 0.001</p> <p>Sensitivity: 3</p> <p>Snap: <input type="checkbox"/></p> <p>Invert: <input type="checkbox"/></p> <p>Type: Key or Mouse Button</p> <p>Axis: 6th axis (Joysticks)</p> <p>Joy Num: Get Motion from all Joysticks</p>
<p>▼ Joy Y</p> <p>Name: Joy Y</p> <p>Descriptive Name: </p> <p>Descriptive Negative Name: </p> <p>Negative Button: </p> <p>Positive Button: </p> <p>Alt Negative Button: </p> <p>Alt Positive Button: </p> <p>Gravity: 0</p> <p>Dead: 0.35</p> <p>Sensitivity: 1</p> <p>Snap: <input type="checkbox"/></p> <p>Invert: <input checked="" type="checkbox"/></p> <p>Type: Joystick Axis</p> <p>Axis: 5th axis (Joysticks)</p> <p>Joy Num: Get Motion from all Joysticks</p>	<p>▼ Crouch</p> <p>Name: Crouch</p> <p>Descriptive Name: </p> <p>Descriptive Negative Name: </p> <p>Negative Button: </p> <p>Positive Button: left ctrl</p> <p>Alt Negative Button: </p> <p>Alt Positive Button: joystick button 9</p> <p>Gravity: 3</p> <p>Dead: 0.001</p> <p>Sensitivity: 3</p> <p>Snap: <input type="checkbox"/></p> <p>Invert: <input type="checkbox"/></p> <p>Type: Key or Mouse Button</p> <p>Axis: 6th axis (Joysticks)</p> <p>Joy Num: Get Motion from all Joysticks</p>

Another thing to note when using a controller for movement is that Unity views controller input a little differently depending on what type of system you are using with Unity, as well as the type of controller you are using. This will affect the “Type”, “Axis”, and “Alt Positive Button” fields of each of the buttons. These options can also be changed depending on which button you want to do what.

Note: “Positive Button” is used for PC controls and “Alt Positive Button” is used for Gamepad controls. The “Type” field also changes depending if the control is a button or an axis.

This package was developed using a Windows 10 PC and an Xbox 360 controller. If your set up is not the same, then the Joystick button references will most likely be different for you. Below is an image that was taken from <https://wiki.unity3d.com/index.php/Xbox360Controller> in order to show you what Unity sees from an Xbox360 controller.

WINDOWS



MAC OS X



Swimming and Climbing

Swimming and climbing both work very similarly. Both require the player to be standing inside of a hit box in order to activate the correct movement behavior. Both of these hit boxes are set up on prefab items for you to drag into your scene view, however both require a new tag to be created. The names of the tags are 'BodyOfWater' and 'Ladder'. Both of these names are case sensitive and can be added by selecting any game object, clicking on the 'Tag' dropdown arrow, and selecting 'Add tag...'. Once those are added, make sure that the object with a trigger box collider has the corresponding tag. Place the prefab in your scene and your player can now swim and climb in the designated areas.

It is also possible to create your own water and ladder objects by following the same set up and using a custom 3D model. Keep in mind however that if you create a new water object, it requires an empty child object named 'WaterSurface' as well. Place the child object at the surface level of your water and change all objects to the 'Ignore Raycast' layer, as swimming relies on this to work correctly.

Additional Notes

- Visual Capsule and Glasses can be safely removed, simply Unpack the prefab and delete the objects.
- When using water, make sure that the player can only enter or exit by surfacing. If the player leaves the water area without surfacing, then the player will continue to swim while outside of water.
- Do not change the size of the capsule as the code's calculations are based on the measurements of the default player.

Conclusion

I hope that you find this character controller to be a great addition to your already awesome game. If you have any issues with this package, find a bug, or need help with set up then please feel free to reach out to me via email at FrodoUndead@gmail.com I will do my very best to help you with any problems you may encounter. Thank you again for your purchase. Happy development! 📧