ADVANCED MOBILE FIRST-PERSON CONTROLLER

2.0

By DragonBox

dragonboxstudio@gmail.com

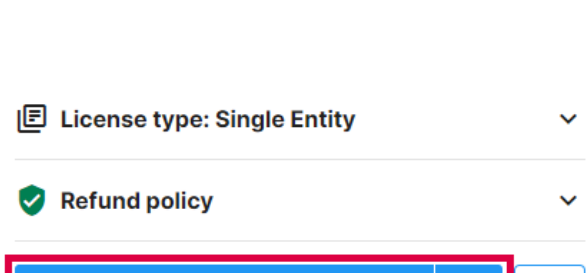
Contents

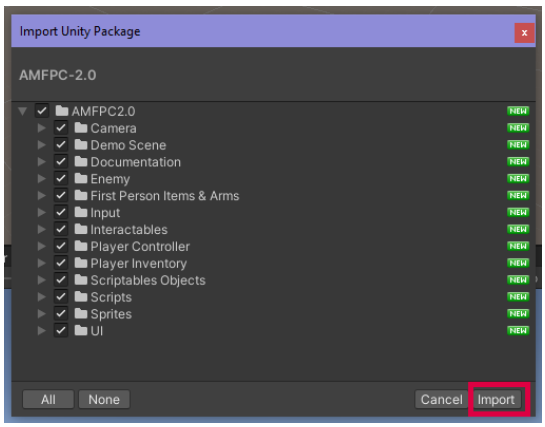
1. Setup

1.1. Importing the package

Open your Unity project or create a new one.

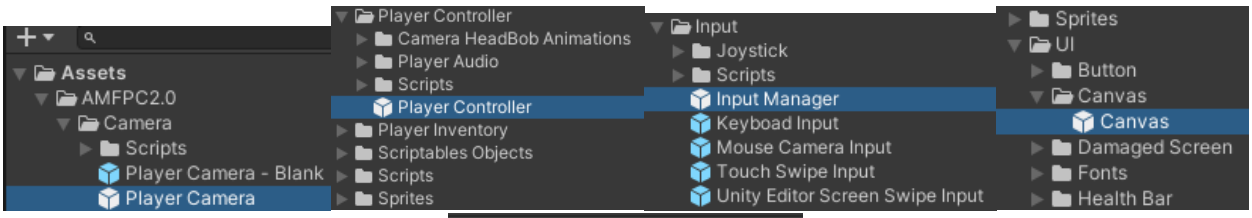
On the package page click “Open in Unity”, then download and import the package to your project.

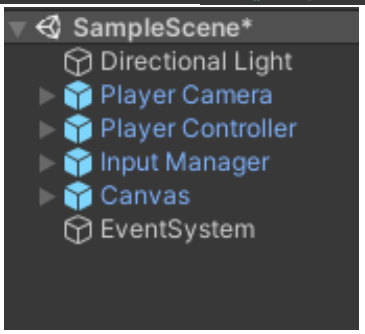




1.2. Setting up the controller

1- Open your project window and expand the AMFPC.20 folder.
2- Expand the “Camera” folder and drag the “Player Camera” prefab to the hierarchy.
3- Expand the “Player Controller” folder and drag the “Player Controller” prefab to the hierarchy.
4- Expand the “Input” Folder and drag the “Input Manager” prefab to the hierarchy.
5- Expand the “UI” folder > open the “Canvas” folder and drag the “Canvas” prefab to the hierarchy.
6- Add Event System: Go to GameObject > UI > Event System.

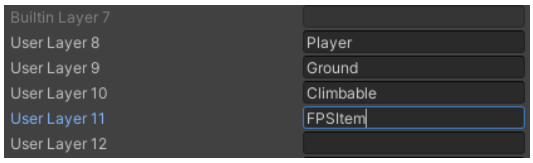




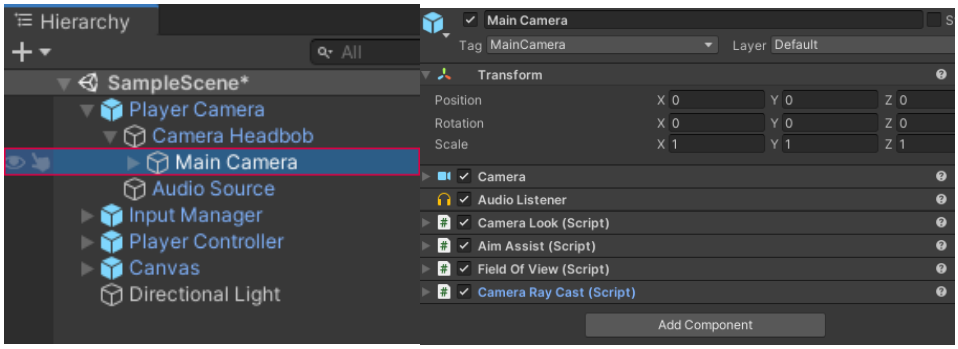
1.3. Setting up layers

1- Go to “Edit” > “Project Settings” > “Tags and Layers’ and add the following layers:

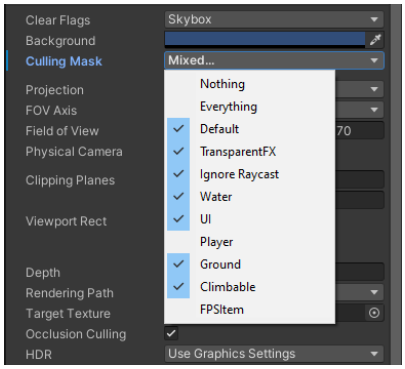
“Ground”, “Climbable”, “FPSItem”, “Player”



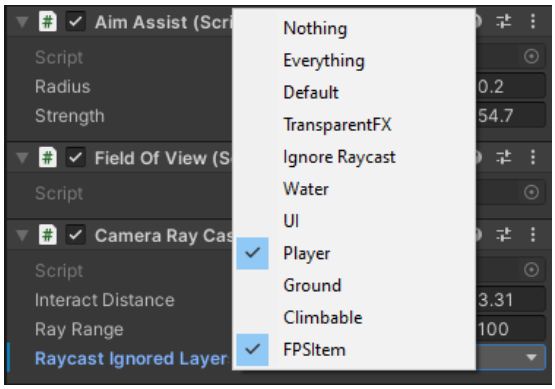
2- Expand the “Player Camera” and select “Main Camera.



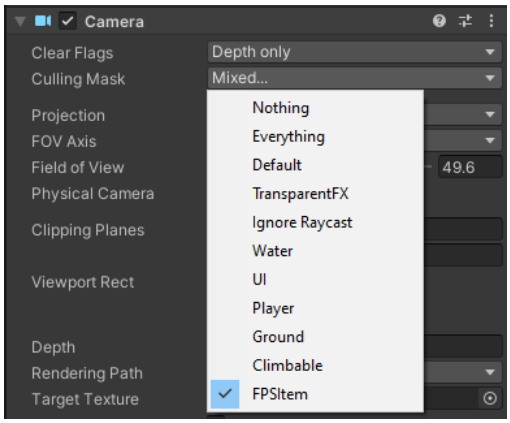
In the “Camera” Component set the “Culling Mask” as shown here:



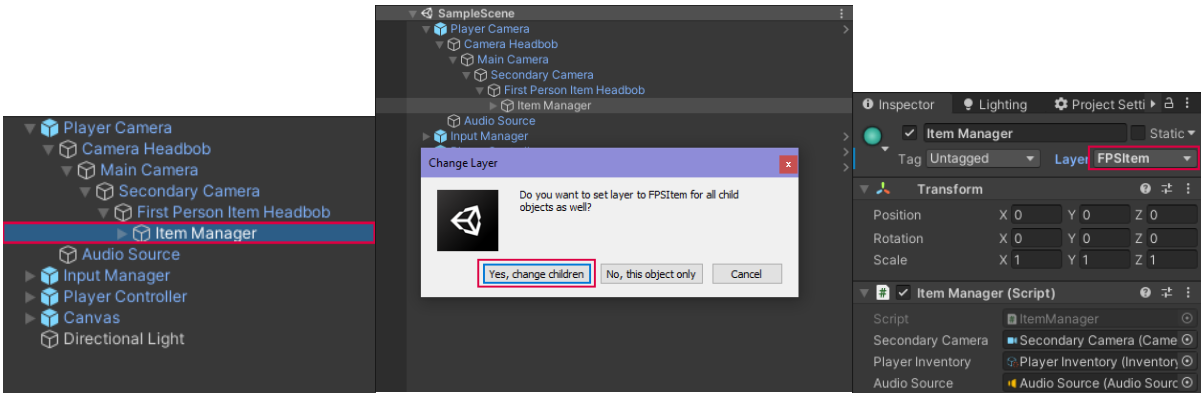
In the “Camera Ray Cast” script component set the “Raycast Ignored Layers” as shown here:



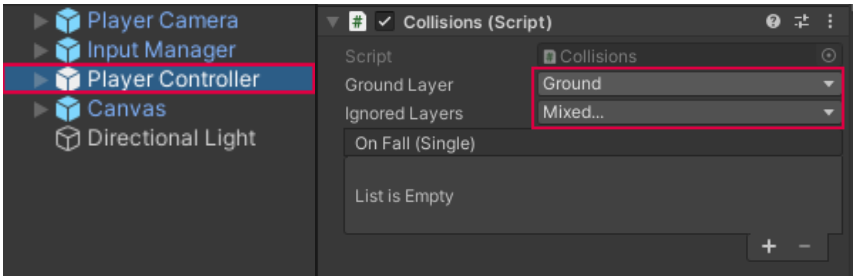
2- Select the “Secondary Camera” Under “ Main Camera”, in the “Camera” component set the “Culling Mask” as shown here:



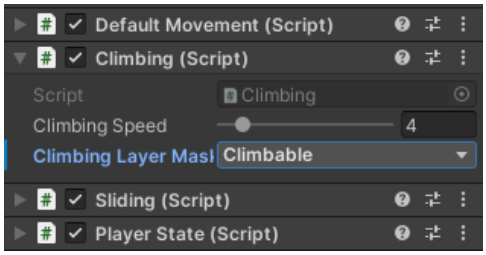
3- Under the “Main Camera” Expand to “Item Manager” and set its layer to “FPSItem”(*make sure to change it for all the child GameObjects under “Item Manager”*).



4- Select the “Player Controller” > Expand the “Collisions” script component and set the “Ground” Layer Mask to “Ground”,set “Ignored Layers” to “FPSItem” and “Player”.



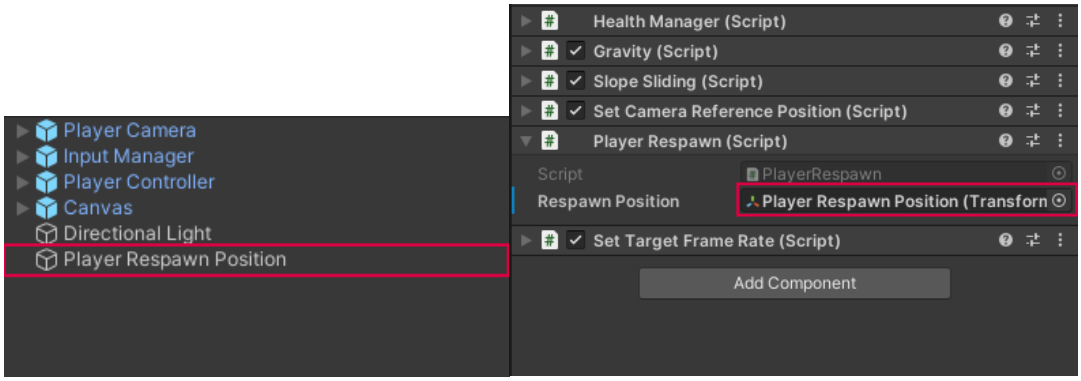
Expand the “Climbing” script component and set the “Climbing Layer” to “Climbable”.



1.4. Setting the player respawn position.

Create a new GameObject in the scene and name it “Player Respawn Position”, make sure to place it where you want the player to respawn to when it respawns.

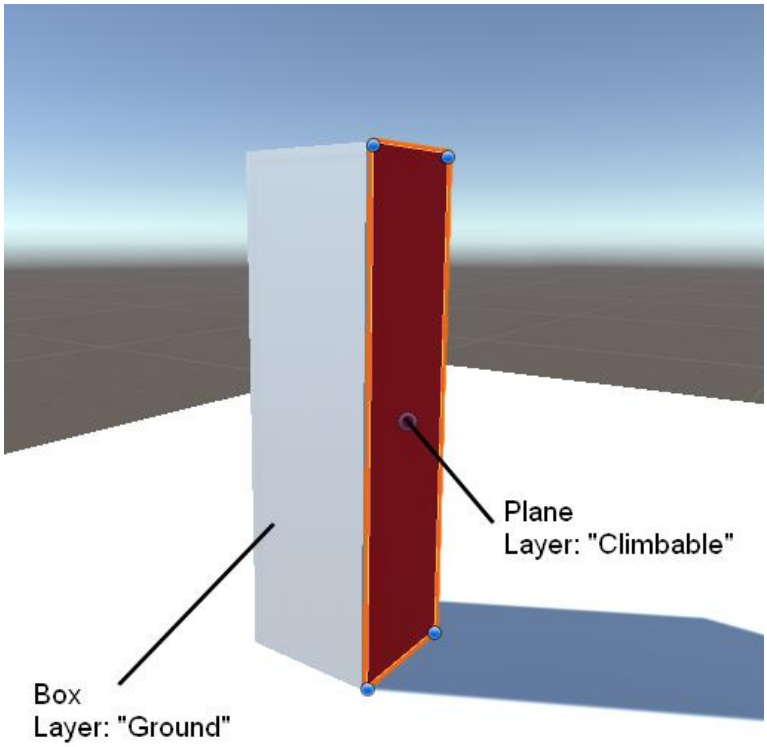
Expand the “Player Respawn” script component and drag and drop your created GameObject to “Respawn Position” exposed variable in the inspector.



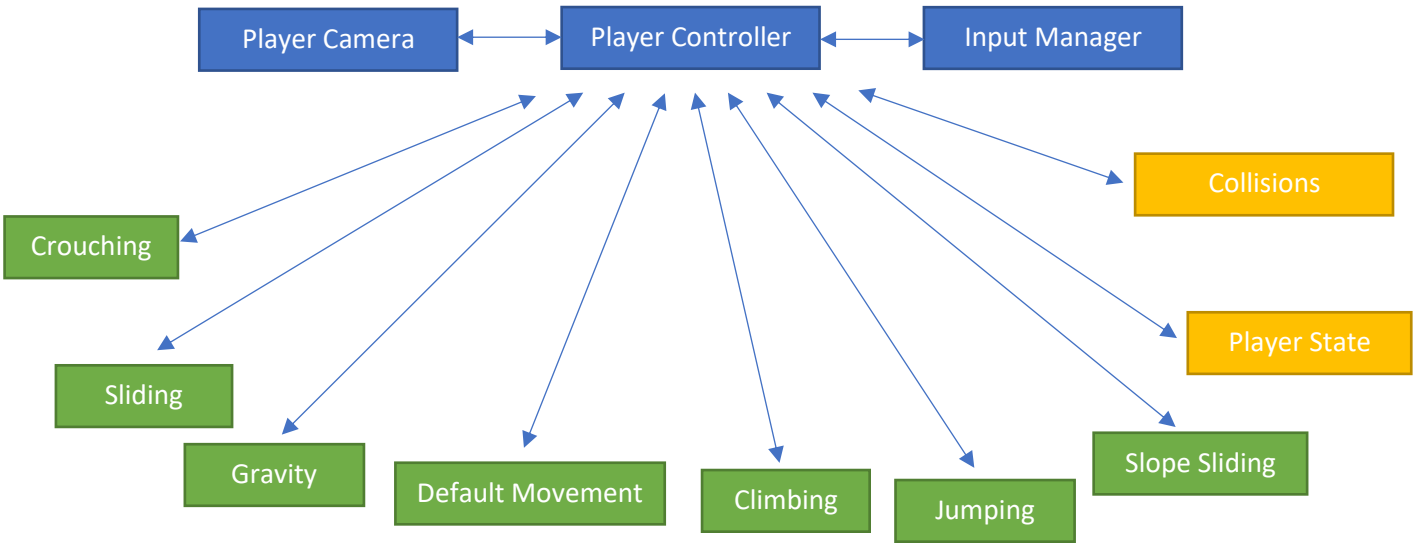
1.5. Setting up Ground and Climbable Surfaces.

1- Select each GameObject with a Collider that you want the controller to be able to walk on/Ground and set its layer to “Ground”.
2- Select each Collider you want the controller to be able to climb and set its layer to “Climbable” Layer

Preferably you can create Climbable surfaces using planes to limit the area you want the controller to be able to detect as a climbing surface.



2. How the controller works



2.1. Controller movement and mechanics

The controller moves using a vector “velocity” which is a sum of an array of vectors called “Velocity Vectors”, “characterController.Move(velocity)”.

In the Start function the “Default Movement” for example checks the count of “Velocities” array and stores an index (i) of an empty slot and assigns value to it “velocites[i] = vector3”.

You can refer to “Sliding” “Gravity” “Jumping” “Climbing” as mechanics, each of them has a Boolean public variable “mechanicEnabled” which is used to turn on/off a mechanic.

2.2. Player Camera

The “Player Camera” position follows a GameObject transform reference “Camera Position Reference” (child of the “Player Controller”)”, controlled by the “Crouching” script/mechanic.

The “Main Camera” is parented to “Camera Headbob” GameObject which is responsible for animating camera head bobs.

A secondary camera is parented to the “Main Camera”, which is responsible for rendering only first-person items (weapons, tools, arms…) to avoid clipping.

The “Item Manager” is parented to “First Person Item Headbob”, which is responsible for an additional head bob on the first-person items for mor detail, also to limit the animations required for each first-person item.

2.3. Input Manager

The “Input Manager” holds different events which can be subscribed to and other variables like “moveInput” and “cameraInput”.

2.3.1. Mouse and keyboard input

To switch to mouse and keyboard controls, expand the “Input Manager” and enable “Mouse Camera Input” GameObject and “Keyboard Input” GameObject, also disable the Joystick GameObject in the canvas and the all the buttons.

3. How to create first person items

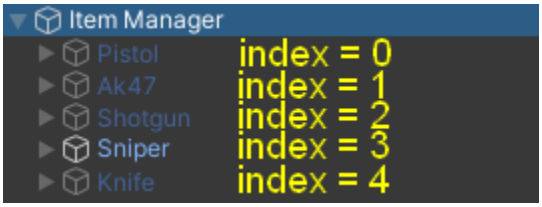
3.1. Item Manager

The “Item Manager” is mainly responsible of switching between first person items which are parented to it, also some other variables which first person items might use.

To switch to an item, call the method “SwitchToItem(index)” in the “Item Manager”.

Example:

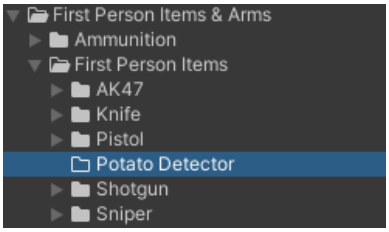
To switch to the “Knife” we call “SwitchToItem(4)”



3.2. Creating a first-person item

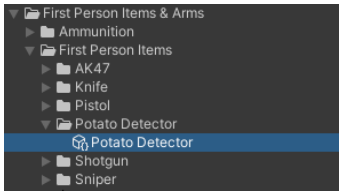
To create a new first-person item, first of all, create a new folder inside of the “First Person Items” folder.

Example: “Potato Detector” folder will hold all the necessary files we need for “Potato Detector”, which is for example a device that detects potatoes around the player.



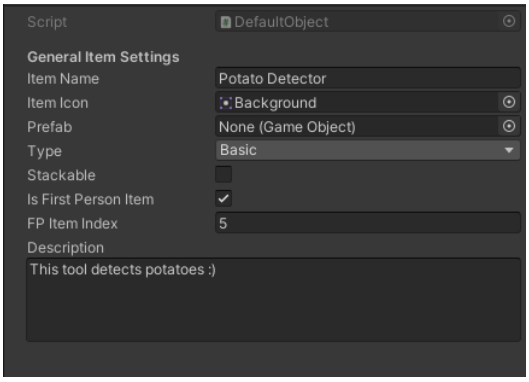
In the newly created folder right click > Create > Inventory System > Items > Default

Once you create your scriptable object make sure to name it properly.



Next, create a new GameObject under “Item Manager” name it properly and add an “Item Settings” script to it.

Select the scriptable object and change its settings in the inspector window.

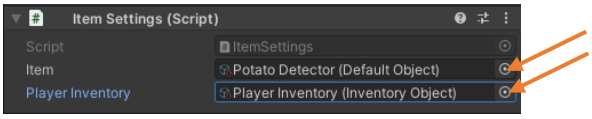


- Tick the “Is First Person Item” bool variable
- The “FP Item Index” is the position of the first-person item under the “Item Manager”, the “Potato Detector” index is 5 because it is the last one of the 6 first person items, counting from 0.

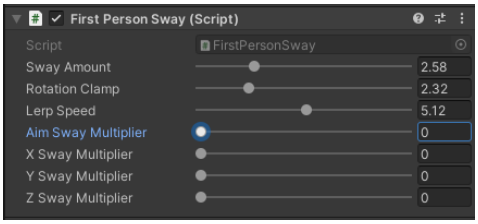
Drag and drop the created scriptable object to the “item” slot.

Drag and drop the Player Inventory (located in the “Player Inventory” folder) in the “Player Inventory” slot.

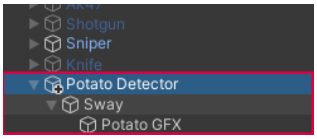
You can directly assign the two variables by clicking on the dot in the right as shown below.



Create another GameObject Named “Sway” under and add “First Person Sway” Script to it, you can play with the settings as you want.

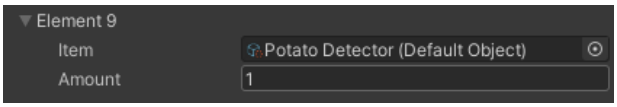


The first-person item graphics will be parented to the “Sway” GameObject.



Select the “Player Inventory” (located in the “Player Inventory” folder) scriptable object and add your first-person item scriptable object in the list, in this case “Potato Detector” we created.

- Increase the size of the container by 1 > Expand the last Element created and assign the first person “Scriptable Object” to it.



4. Creating interactable objects

1.1. Creating an interactable GameObject

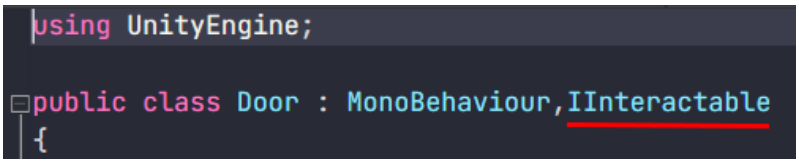
The “Camera Ray Cast” is responsible of detecting interactable GameObject and displaying the “Interact” button UI whenever an interactable GameObject.

To make a GameObject interactable it simply needs:

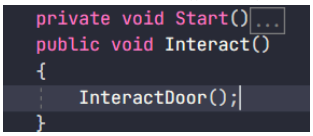
- A collider.
- A script which implements the “IInteractable” interface.
- “Interactable Settings” script component added to it.

How to implement the “IInteractable” interface:

- In your script simply add “,IInteractable” as shown here



- Add a method “Interact()” to your script, and add to it what you want to be executed.



1.2. Creating an item / GameObject to pickup

Similar to creating an interactable GameObject add the following components to your GameObject.

- A collider
- “Pickup Item” Script component. (click “Add Component” > Search “Pickup”)
- “Interactable Settings” script component.

Thank you for your purchase 😊

dragonboxstudio@gmail.com