All in One Game Kit - Easy Ledge Climb Character System

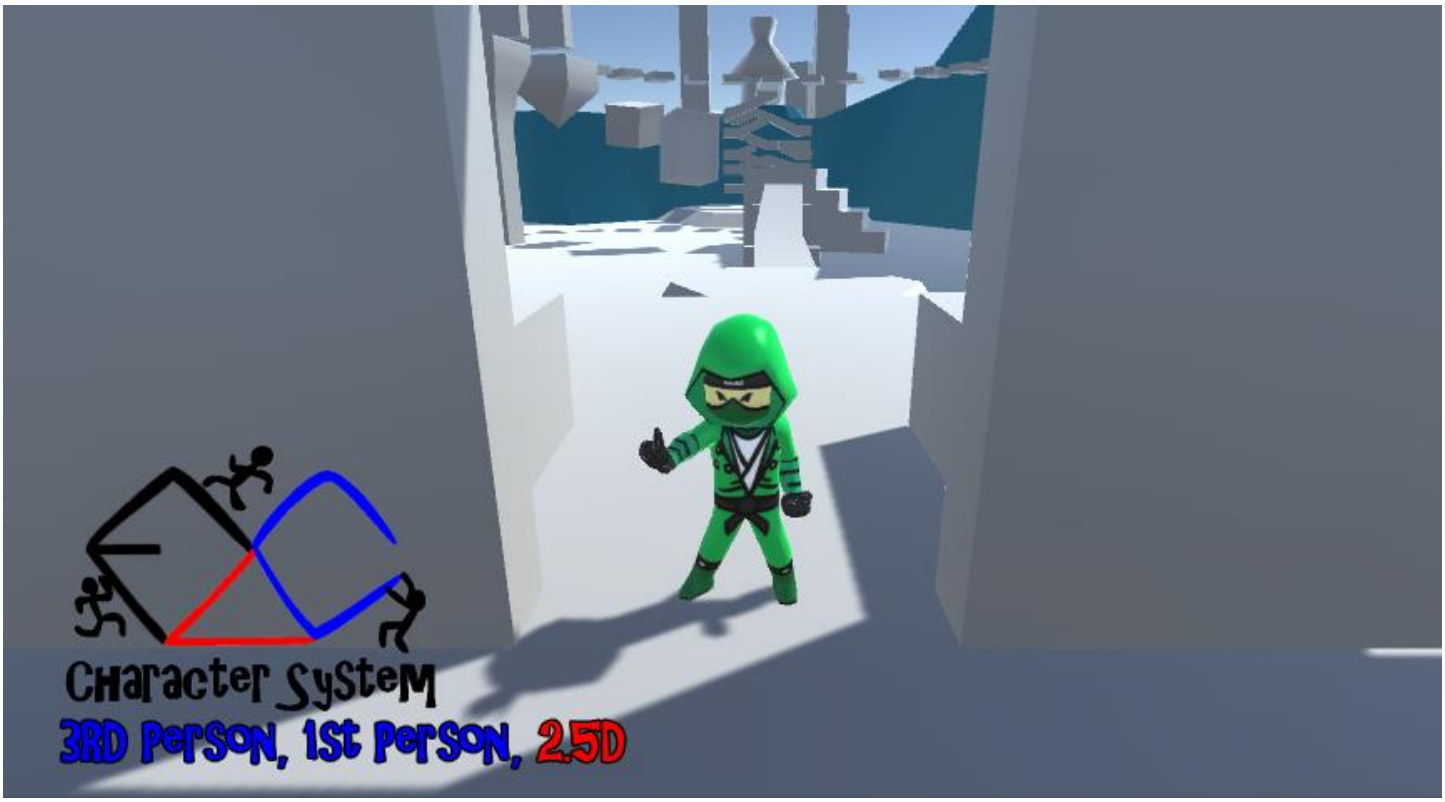


Table of Contents

Folders and Files

GUI:

- ArrowPointer

    - Materials

        Arrow.mat

            (The material of the arrow pointer)

    Arrow.png

(Texture file for the arrow pointer)

Arrow.prefab

(Prefab for the arrow pointer)

- Coins

ARCHRISTY.ttf

(Text font file for the GUI item count)

CoinGUI.png

(Texture file for the coin GUI)

- EnemyHealth

HealthMaterial.mat

(The material of the health bar's health percentage)

NoHealthMaterial.mat

(The material of the health bar's no health percentage)

OutlineMaterial.mat

(The material of the health bar's outline)

- LockOnBars

BlackBars.png

(Texture file for the black bars that appear on screen while locking on)

- PlayerHealth

- Hearts

0.psd

(Texture file of a 0/4 full heart for the player’s health)

1.psd

(Texture file of a 1/4 full heart for the player’s health)

2.psd

(Texture file of a 2/4 full heart for the player’s health)

3.psd

(Texture file of a 3/4 full heart for the player’s health)

4.psd

(Texture file of a 4/4 full heart for the player's health)

Models:

- Enemies

    - Logs

        - Materials

            Body.mat

                (The material of the log enemy)

        Body.png

            (Texture file for the log enemy)

        LogEnemy.FBX

            (.FBX model file of the log enemy)

        LogEnemy.prefab

            (Prefab for the log enemy)

    - Robots

        - Materials

            Body.mat

                (The material of the robot enemy's body)

            Mouth.mat

                (The material of the robot enemy's mouth)

            Screws.mat

                (The material of the robot enemy's screws)

            Shoes.mat

                (The material of the robot enemy's shoes)

            Unlit-CullOff.shader

                (Shader for the robot enemy's mouth material)

        - Textures

            Body.png

                (Texture file for the robot enemy's body)

            Mouth.png

                (Texture file for the robot enemy's mouth)

            ReflectionCubemap.png

(Texture file for the robot enemy's reflection)

Screws.png

(Texture file for the robot enemy's screws)

Shoes.png

(Texture file for the robot enemy's shoes)

EnemyController.controller

(Animator controller for the enemy)

RobotEnemy.FBX

(.FBX model file of the robot enemy)

RobotEnemy.prefab

(Prefab for the robot enemy)

- Items

- Coin

- Materials

Coin.mat

(The material of the coin item)

Coin.FBX

(.FBX model file of the coin)

Coin.png

(Texture file for the coin)

Coin.prefab

(Prefab for the coin)

- Heart

- Materials

Heart.mat

(The material of the heart item)

Heart.FBX

(.FBX model file of the heart)

Heart.png

(Texture file for the heart)

Heart.prefab

(Prefab for the heart)

- LedgeClimbNinja

- Materials

LedgeClimbNinja.mat

(The material of the Ledge Climb Ninja)

FirstPersonArms.FBX

(.FBX model file of the Ledge Climb Ninja with separate arm models used for First Person mode)

LedgeClimbNinja.FBX

(.FBX model file of the Ledge Climb Ninja (player))

Contains 43 Mecanim animations; all root motion compatible.)

LedgeClimbNinja.png

(Texture file for the Ledge Climb Ninja)

PlayerController.controller

(Animator controller for the Ledge Climb Ninja)

- Stages

- ExampleStage1

- Materials

Ladder.mat

(The material of the ladder)

logo.mat

(The material of the Easy Ledge Climb (ELC) Character System logo)

mountains.mat

(The material of the stage's mountains)

standard.mat

(The main material of the stage)

Vines.mat

(The material of the vines)

ExampleStage1.FBX

(.FBX model file of the first example stage)

Ladder.png

(Texture file for the ladders)

logo.png

(Texture file for the Easy Ledge Climb (ELC) Character System logo)

Vines.png

(Texture file for the vines)

- ExampleStage2

- Materials

logo.mat

(The material of the Easy Ledge Climb (ELC) Character System logo)

mountains.mat

(The material of the mountains)

standard.mat

(The main material of the stage)

ExampleStage2.FBX

(.FBX model file of the second example stage)

- SideScrollerStage

- Materials

lava.mat

(The material of the lava)

lava.png

(Texture file for the lava)

SideScrollerStage.prefab

(Prefab for the side-scroller example stage)

- Water

- Materials

underwater.mat

(The material used for underwater)

water.mat

(The material of the water)

waterShadows.mat

(The material used for the water's shadows)

ScrollTexture.cs

(Scrolls the texture of a material)

StandardCullOff.shader

(.FBX model file of the first example stage)

UnlitTransparentCullOff.shader

(Texture file for the ladders)

water.png

(Texture file for the water)

ParticleEffects:

- Dust

- Materials

DustCloud.mat

(The material of the dust cloud particle effect)

DustDoubleJump.mat

(The material of the dust cloud particle effect used for the double jump)

DustCloud.prefab

(Prefab for the dust cloud particle effect)

DustDoubleJump.prefab

(Prefab for the DustDoubleJump particle effect)

DustParticle.png

(Texture file for the dust cloud particle effect used for the double jump)

- Spark

PaticleAfterburner.mat

(The material of the spark particle effect)

Spark.prefab

(Prefab for the spark particle effect)

- Splash

- Materials

ParticleSplashes.mat

(The material of the water drops from the splash particle effect)

ParticleWaterSpray.mat

(The material of the water ripples from the splash particle effect)

PaticleCloudWhite.png

(Texture file for the water ripples in the splash particle effect)

Splash.prefab

(Prefab for the spark particle effect)

SplashesFineParticle.png

(Texture file for the water drops in the splash particle effect)

AutoDestroyParticleSystem.cs

(Automatically destroys the script holder once its ParticleSystem has completed)

Scenes:

- FirstPerson

- ExampleStage1

DemoSceneCharacterController.unity

(Demo scene that uses a 1st person camera view, "ExampleStage1" as the stage, "FirstPersonArms" as the player, and a Character Controller for the movement)

DemoSceneRigidbody.unity

(Demo scene that uses a 1st person camera view, "ExampleStage1" as the stage, "FirstPersonArms" as the player, and a Rigidbody for the movement)

- ExampleStage2

DemoSceneCharacterController.unity

(Demo scene that uses a 1st person camera view, "ExampleStage2" as the stage, "FirstPersonArms" as the player, and a Character Controller for the movement)

DemoSceneRigidbody.unity

(Demo scene that uses a 1st person camera view, "ExampleStage2" as the stage, "FirstPersonArms" as the player, and a Rigidbody for the movement)

- SideScroller

- SideScrollerStage

DemoSceneCharacterController.unity

(Demo scene that uses a side-scrolling camera view, "SideScrollerStage" as the stage, "LedgeClimbNinja" as the player, and a Character Controller for the movement)

DemoSceneRigidbody.unity

(Demo scene that uses a side-scrolling camera view, "SideScrollerStage" as the stage, "LedgeClimbNinja" as the player, and a Rigidbody for the movement)

- ThirdPerson

- ExampleStage1

DemoSceneCharacterController.unity

(Demo scene that uses a 3rd person camera view, "ExampleStage1" as the stage, "LedgeClimbNinja" as the player, and a Character Controller for the movement)

DemoSceneRigidbody.unity

(Demo scene that uses a 3rd person camera view, "ExampleStage1" as the stage, "LedgeClimbNinja" as the player, and a Rigidbody for the movement)

- ExampleStage2

DemoSceneCharacterController.unity

(Demo scene that uses a 3rd person camera view, "ExampleStage2" as the stage, "LedgeClimbNinja" as the player, and a Character Controller for the movement)

DemoSceneRigidbody.unity

(Demo scene that uses a 3rd person camera view, "ExampleStage2" as the stage, "LedgeClimbNinja" as the player, and a Rigidbody for the movement)

Scripts:

- Camera

    CameraController.cs - modified from John McElmurray's and Julian Adams' "ThirdPersonCamera.cs" script
    - Follow the player at a set speed, height, and distance.
    - Follow the player as a side-scroller.
    - Lock on behind the player.
    - Lock on to objects.
    - Mouse Orbit the player.
    - Enter First Person mode.
    - Mouse Orbit in First Person mode.

    *NOTE: *You should always set a layer for your player so that you can disable collisions with that layer (by unchecking it in the script's Collision Layers). If you do not, the camera will collide with the player himself!**

- Enemy

(Enemy must have a CharacterController or Rigidbody component in order for the enemy AI script to work!)

    EnemyAI.cs
    - Follow the player at a set speed and rotation.
    - Attack the player.
    - Receive damage.
    - Regain health.
    - Use a health bar.
    - Respawn.

    *NOTE: *You should always set a layer for your enemy so that you can disable collisions with that layer (by unchecking it in the script's Collision Layers). If you do not, the raycasts and linecasts will collide with the enemy itself and keep the script from working properly!**

- Items

    ItemRotation.cs
    - Rotate towards a set direction (the higher the number, the faster the item will rotate).

- MobileSupport

    - CrossPlatformInput

(The cross platform input from Unity's "Standard Assets" pack)

- MovingPlatforms

MovingPlatform.cs

- Move to and from a set position.
- Move at a set speed.
- Rotate towards a set direction.
- Wait before moving.

- Player

(Player must have a CharacterController or Rigidbody component in order for these to work!)

Health.cs

- Have a set health.
- Receive damage from enemies, items, and falls.
- Regain health over time, or from items.
- Respawn.
- Set GUI images for the health.
- Position the health GUI anywhere on the screen, and limit the number of GUI images (hearts) that can be on each row of health.

ItemManager.cs

- Have an inventory of different items.
- Count and display the number of items the player currently holds (either by simply counting the number, showing the number relative to the item's limit (with zeros before it), or by adding by a prefix or suffix to the number).
- Set limits for the number of items the player can have.
- Set GUI images and text for different items.
- Position the GUI images and text anywhere on the screen.

LedgeClimbController.cs

- Grab on to, and climb up and over ledges (Climbing Detectors).
- Move from side to side on ledges, either in a fully 3D setting, or in a 2.5D side-scrolling setting (Moving Detectors).
- Jump from one ledge to another / switch ledges (Ledge Switching Detectors).
- Grab back on to or jump off of a ledge that you just walked off of (Walking Off Ledge Detectors).

- Enable or disable scripts when grabbing on to a ledge, as well as when letting go of a ledge (Scripts To Enable On Grab/Scripts To Disable On Grab).

-> You should always disable any other scripts that affect the movement or rotation of the player. If you do not, this script will not work properly.

MovingPlatformController_Standalone.cs

(This script is the exact same as the "Moving Platforms" section of the PlayerController.cs script, only this is a standalone version of the section)

- Ride moving and rotating platforms (Moving Platforms).

PlayerController.cs

- Move and rotate (Movement).
- Slide down slopes (Movement).
- Perform any combo of jumps with different heights and animations (Jumping).
- Perform a double jump (Jumping).
- Wall jump (Wall Jumping).
- Perform any combo of attacks with different strengths and animations (Attacking).
- Climb ladders and walls (Climbing).
- Swim (Swimming).
- Ride moving and rotating platforms (Moving Platforms).

WallClimbController_Standalone.cs

(This script is the exact same as the "Climbing" section of the PlayerController.cs script, only this is a standalone version of the section)

- Climb ladders and walls (Climbing).
- Enable or disable scripts when grabbing on to a ladder or wall, as well as when letting go of one (Scripts To Enable On Grab/Scripts To Disable On Grab).

-> You should always disable any other scripts that affect the movement or rotation of the player. If you do not, this script will not work properly.

NOTE: *You should always set a layer for your player so that you can disable collisions with that layer (by unchecking it in the script's Collision Layers). If you do not, the raycasts and linecasts will collide with the player himself and keep the script from working properly!*

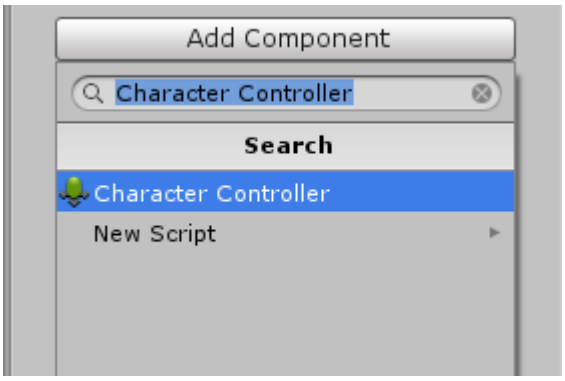
Setting Up the All in One Game Kit - ELC Character System with a New Character

Making the player scripts work with your character

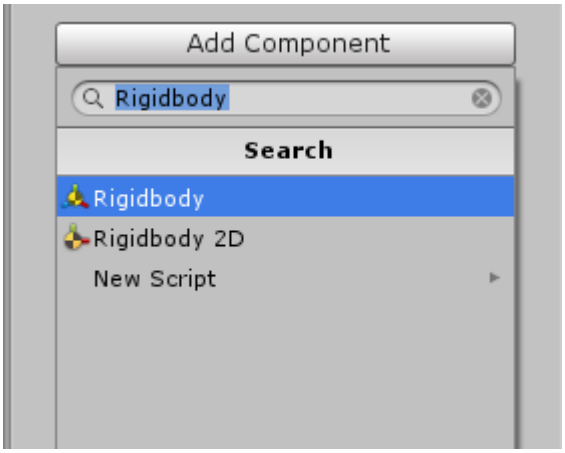
1. Select your character in the "Hierarchy" tab.

2. If your character does not already have one, add a CharacterController **OR** Rigidbody component (only add one) to your character from the “Inspector” tab:

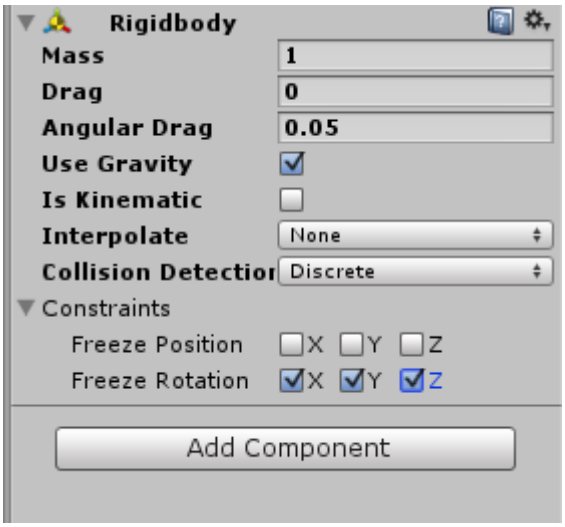
Character Controller:
I recommend using a CharacterController for the player.
To do this, simply click the “Add Component” button in the Inspector tab, search “Character Controller”, then select it.



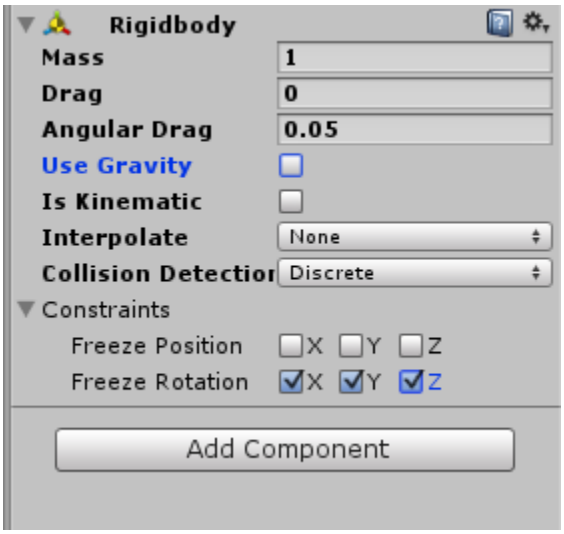
Rigidbody:
If you use a Rigidbody, you must freeze the Rigidbody’s rotation, disable its “Use Gravity” option, and add a collider component as well.
To do this, first click the “Add Component” button in the Inspector tab, search “Rigidbody”, then select it.



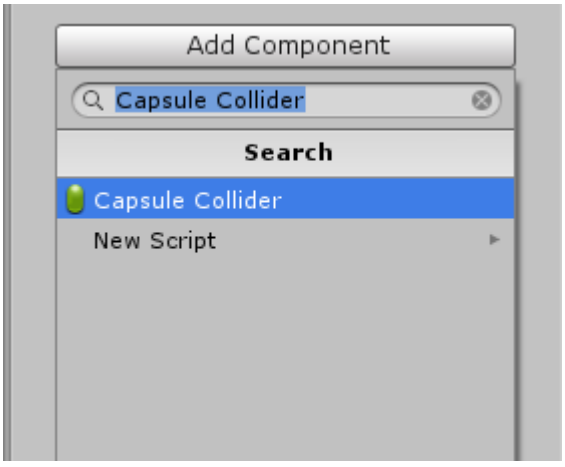
Next, freeze the Rigidbody’s rotation by enabling the X, Y, and Z axis options of “Freeze Rotation”.



Now disable the “Use Gravity” option.

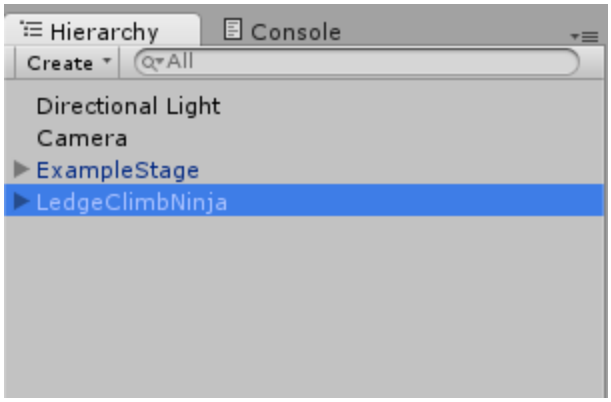


Finally, click the “Add Component” button in the Inspector tab, search a collider (I recommend using the “Capsule Collider” for the player), then select it.

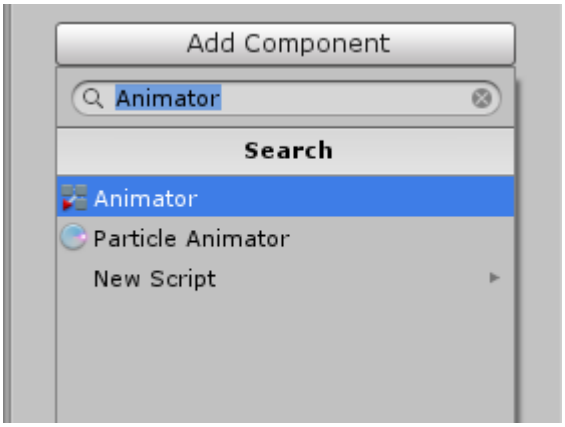


Making the animations work with your character

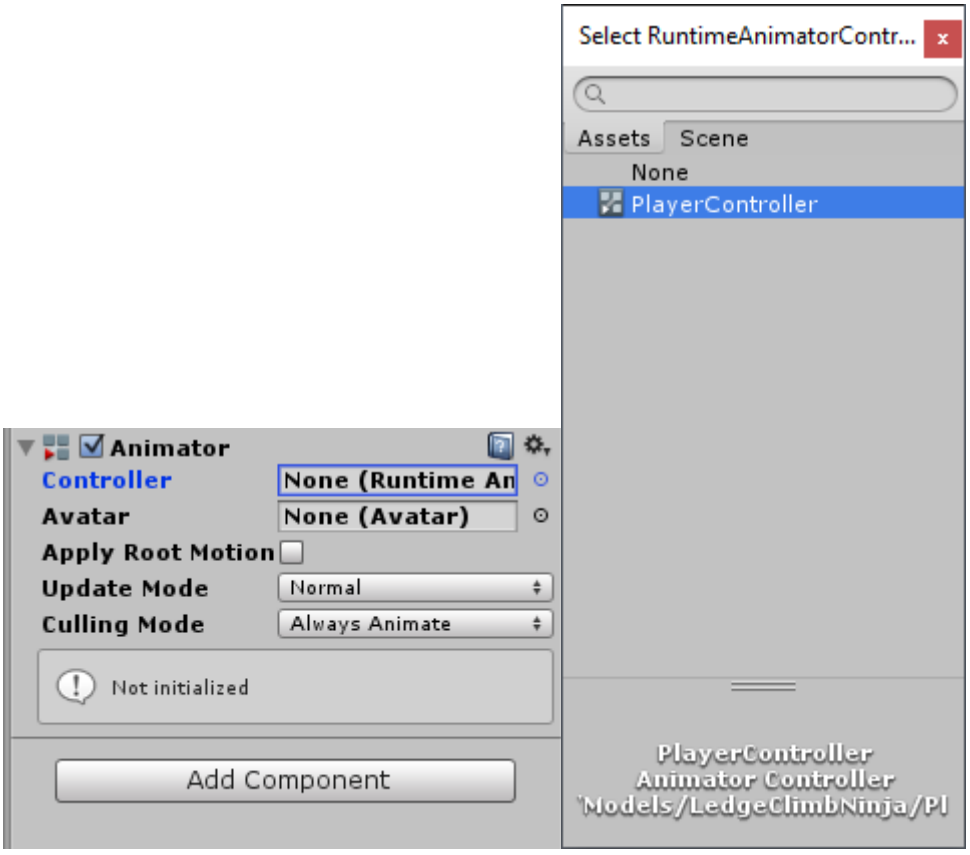
1. Select your character in the “Hierarchy” tab.



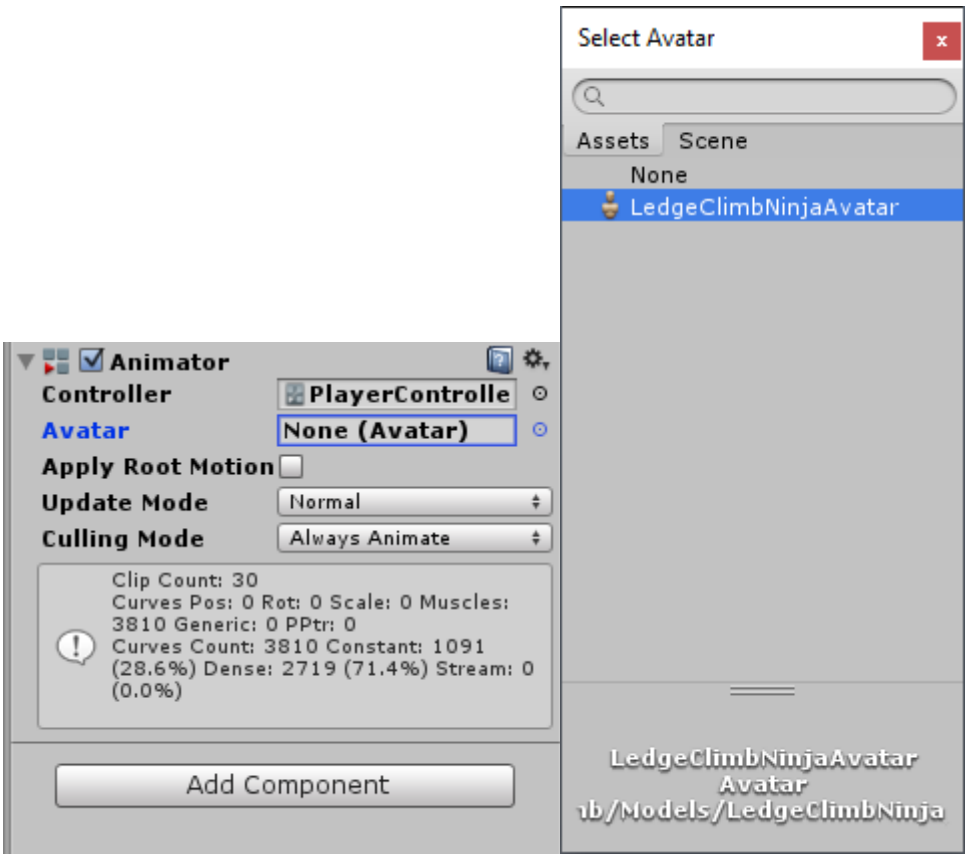
2. If your character does not already have one, add an Animator component to your character from the “Inspector” tab.



3. Set the Controller (Runtime Animator Controller) of the Animator Component to PlayerController.

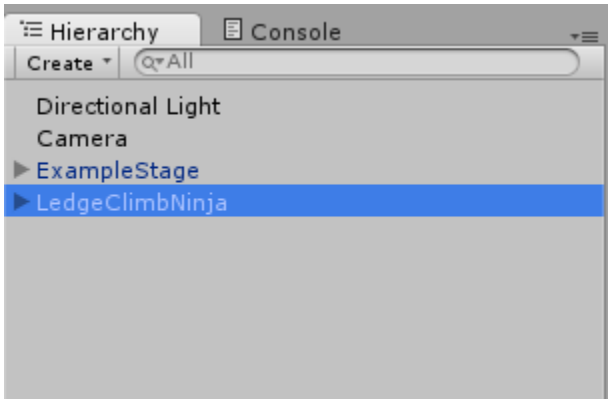


4. Set the Avatar of the Animator Component to LedgeClimbNinjaAvatar.



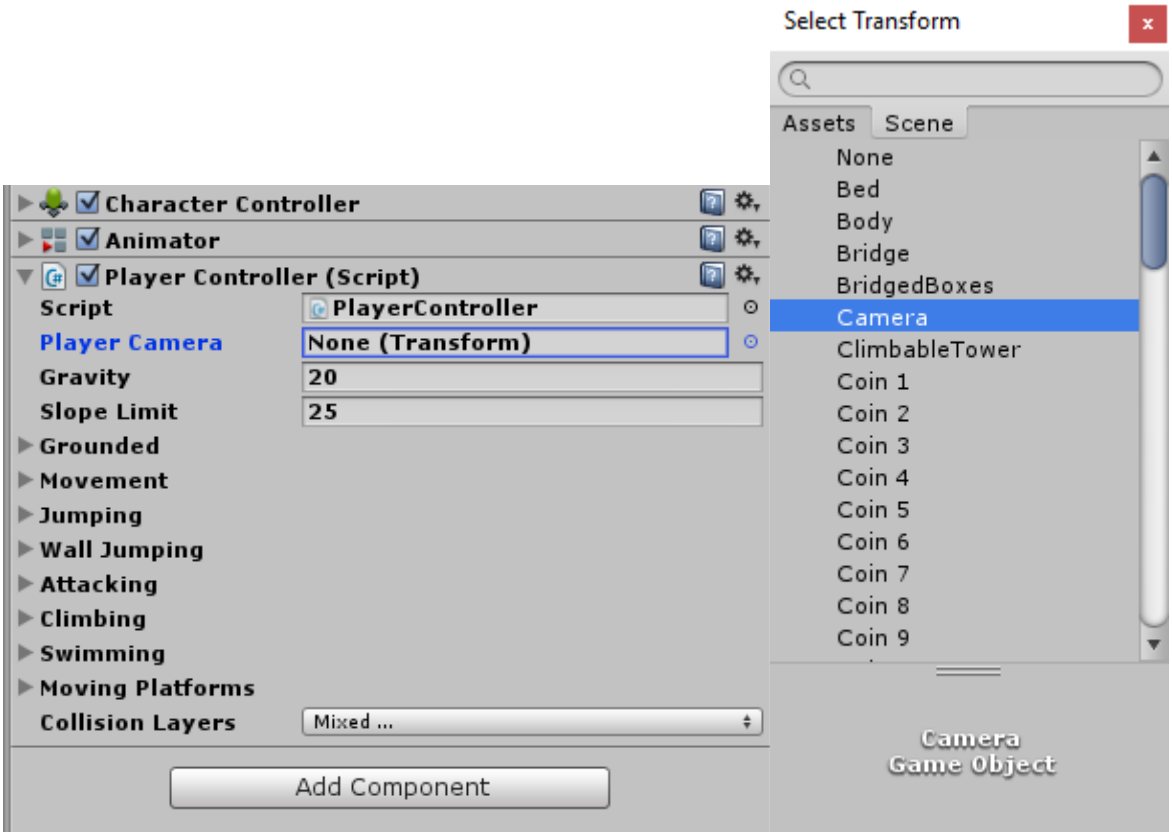
Setting up PlayerController.cs with your character

1. Select your character in the “Hierarchy” tab.

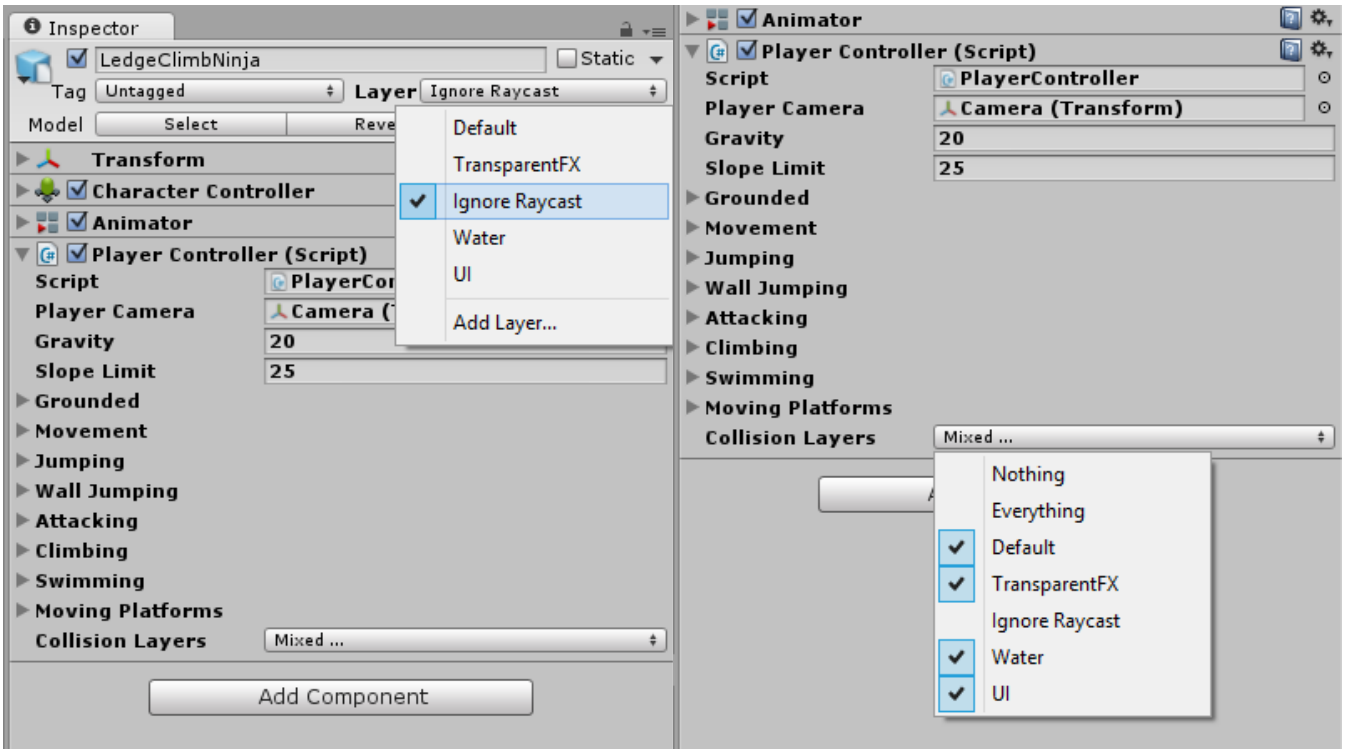


2. Add the PlayerController.cs script to your character (either by using “Add Component” in the “Inspector” tab or by dragging and dropping the script from the “Project” tab on to your player in the “Inspector” tab).

3. Set the Player Camera of the script to the camera you want to use for your player (in this case, the camera is simply named “Camera”).

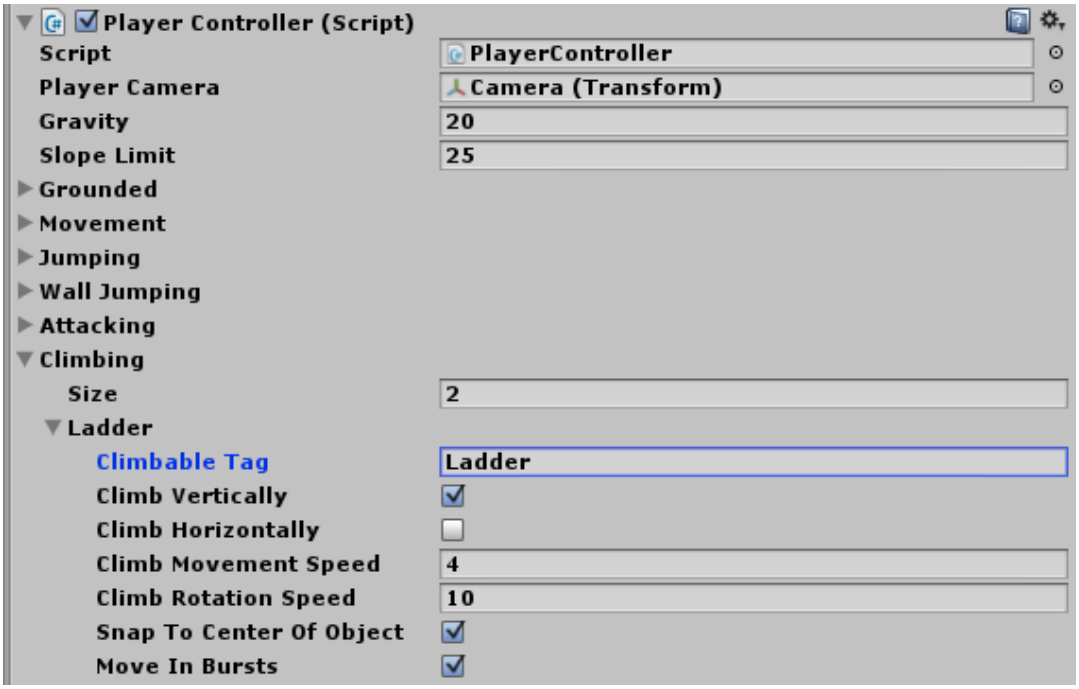


4. Set a layer for your player and uncheck it in the script’s Collision Layers (to avoid having the script’s linecasts and raycasts collide with the player himself).

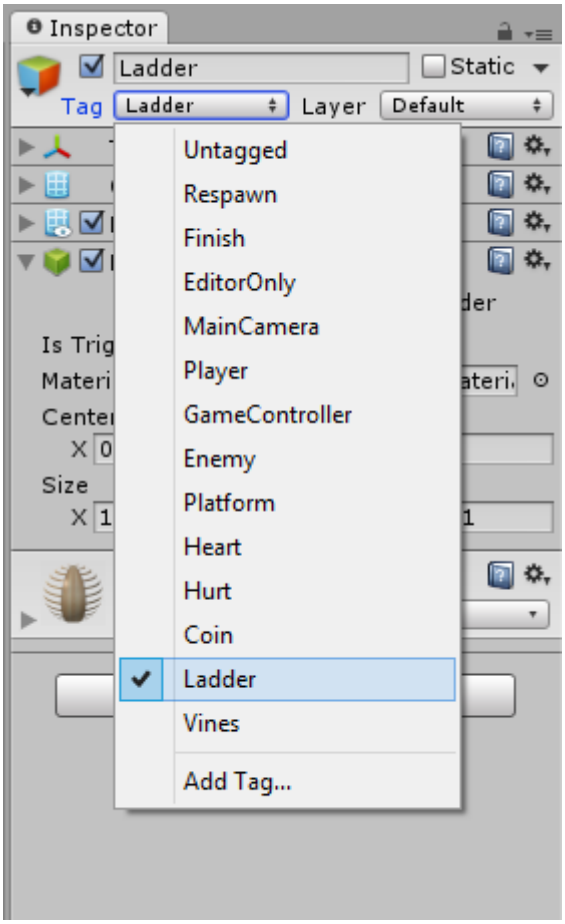


Defining which objects are climbable

1. In order for the player to know which objects are climbable and which are not, the user must define a tag in the "Climbable" section of the player controller.

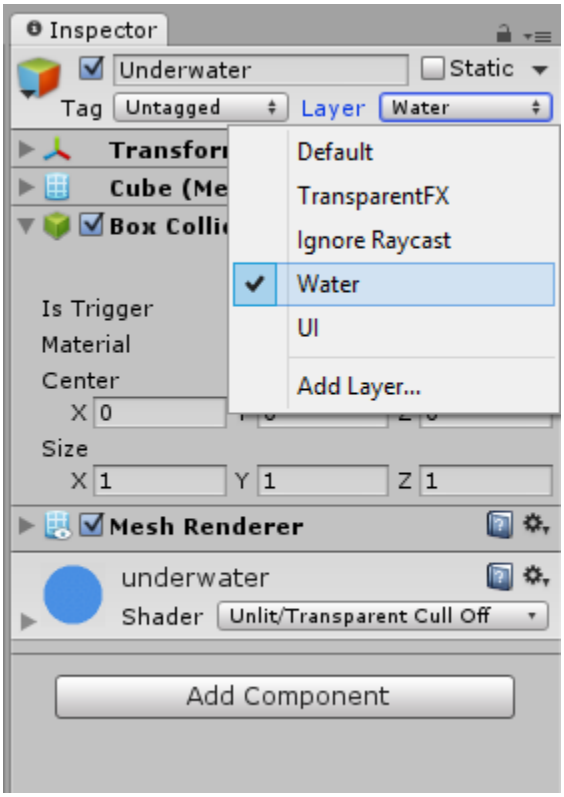


2. Once this tag is defined, you must assign this tag to whichever objects you wish to be climbable.

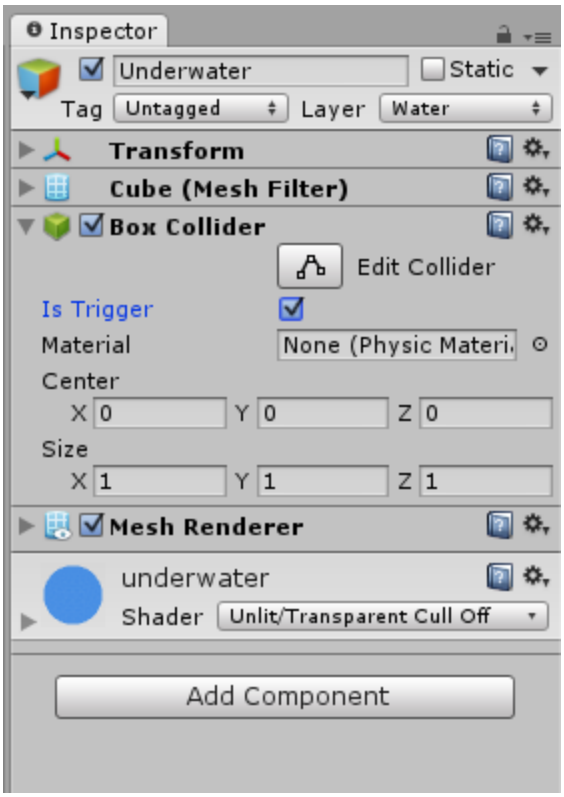


Defining which objects are water / can be swam in

1. In order for the player to know which objects are water or able to be swam in, the objects must have the layer “Water”, and a trigger collider. To do this, simply assign the layer of your object to “Water”.

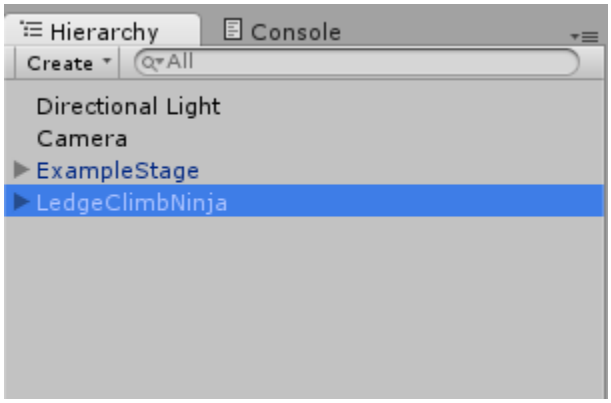


2. Then, in the collider component of your object, enable the option named “Is Trigger”.

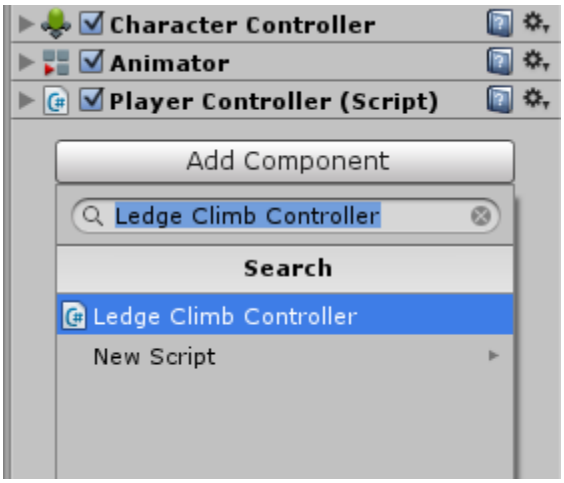


Setting up LedgeClimbController.cs with your character

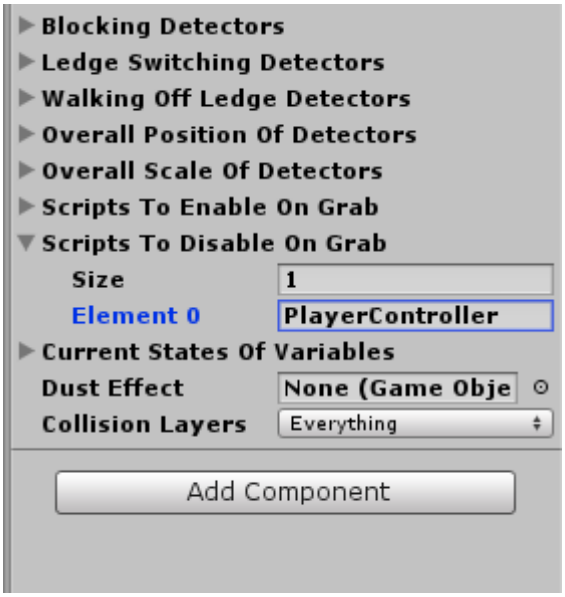
1. Select your character in the “Hierarchy” tab.



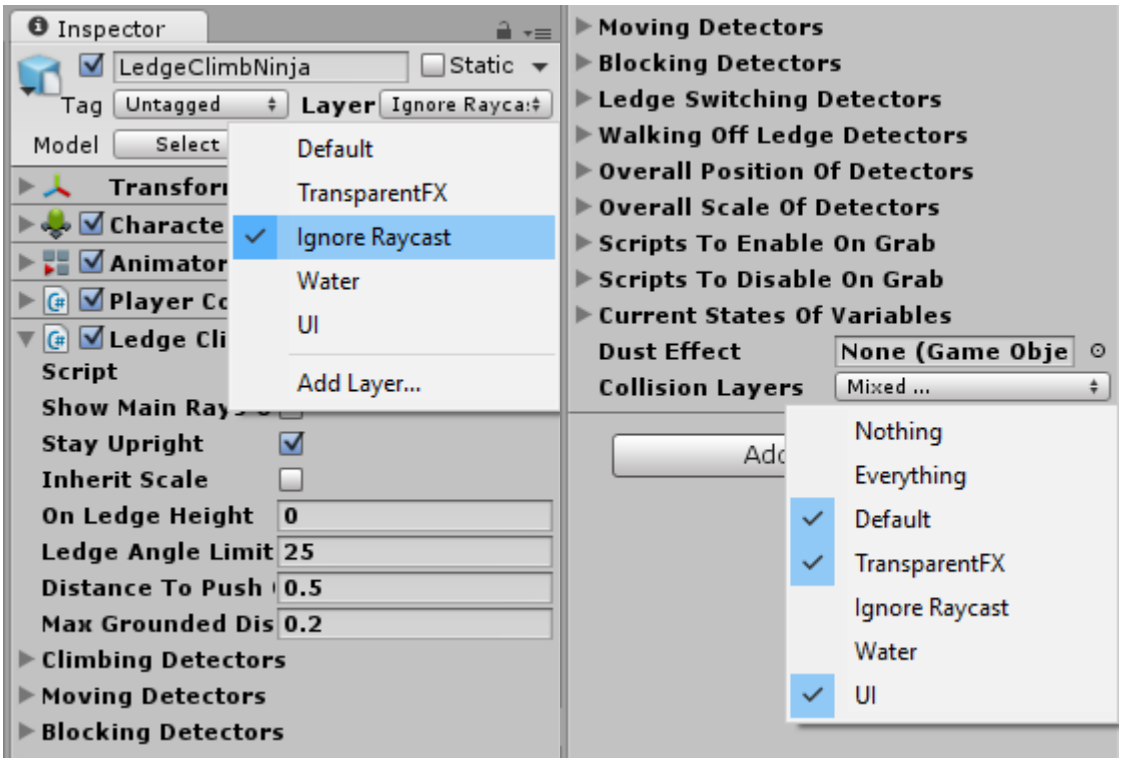
2. Add the LedgeClimbController.cs script to your character (either by using “Add Component” in the “Inspector” tab or by dragging and dropping the script from the “Project” tab on to your player in the “Inspector” tab).



3. If your character is using any other scripts that affect his movement or rotation, enter their names into the “Scripts To Disable On Grab” section so that they do not interfere with the movement and rotation of the player while grabbed on to a ledge.

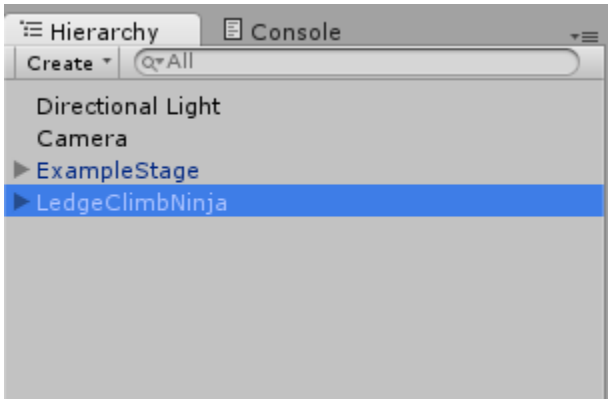


4. Set a layer for your player and uncheck it in the script’s Collision Layers (to avoid having the script’s linecasts and raycasts collide with the player himself).

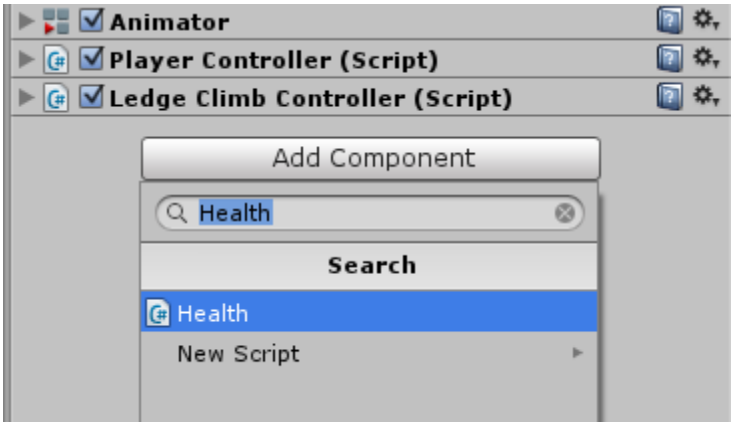


Setting up Health.cs with your character

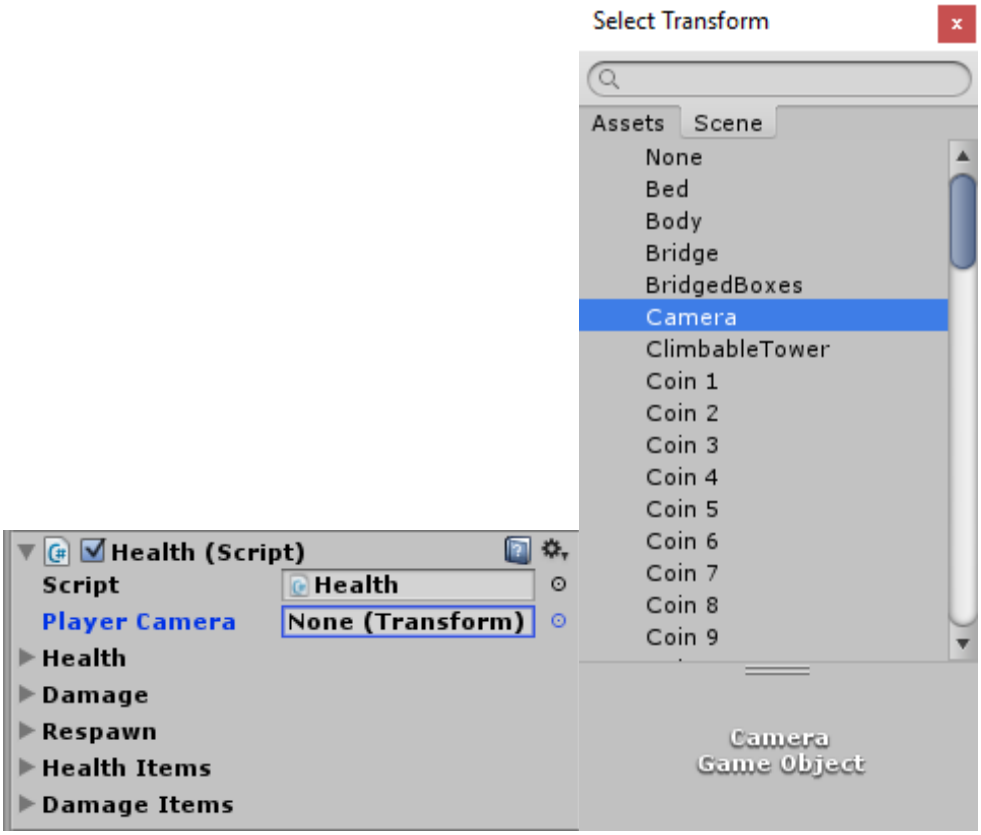
1. Select your character in the “Hierarchy” tab.



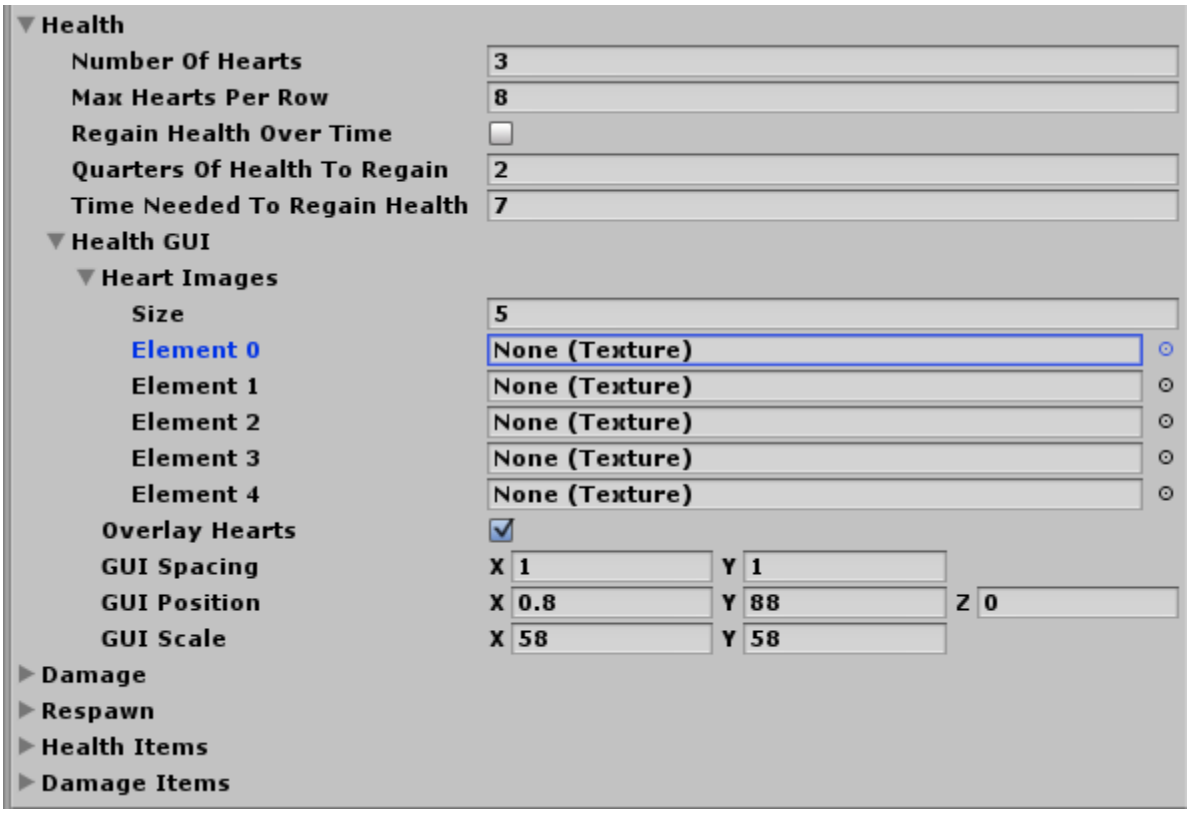
2. Add the Health.cs script to your character (either by using “Add Component” in the “Inspector” tab or by dragging and dropping the script from the “Project” tab on to your player in the “Inspector” tab).

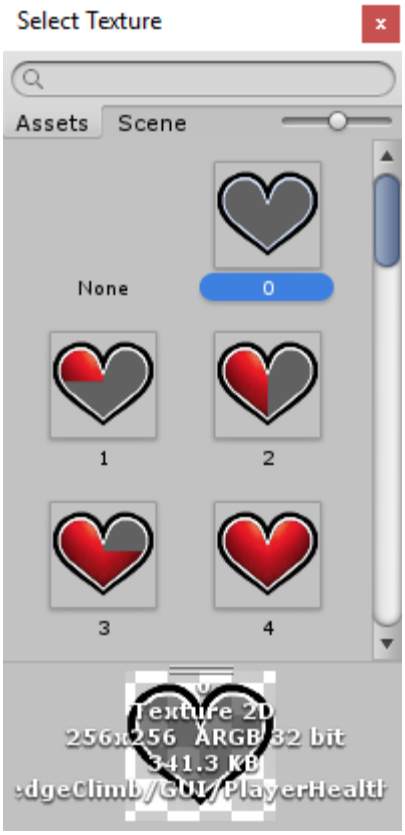


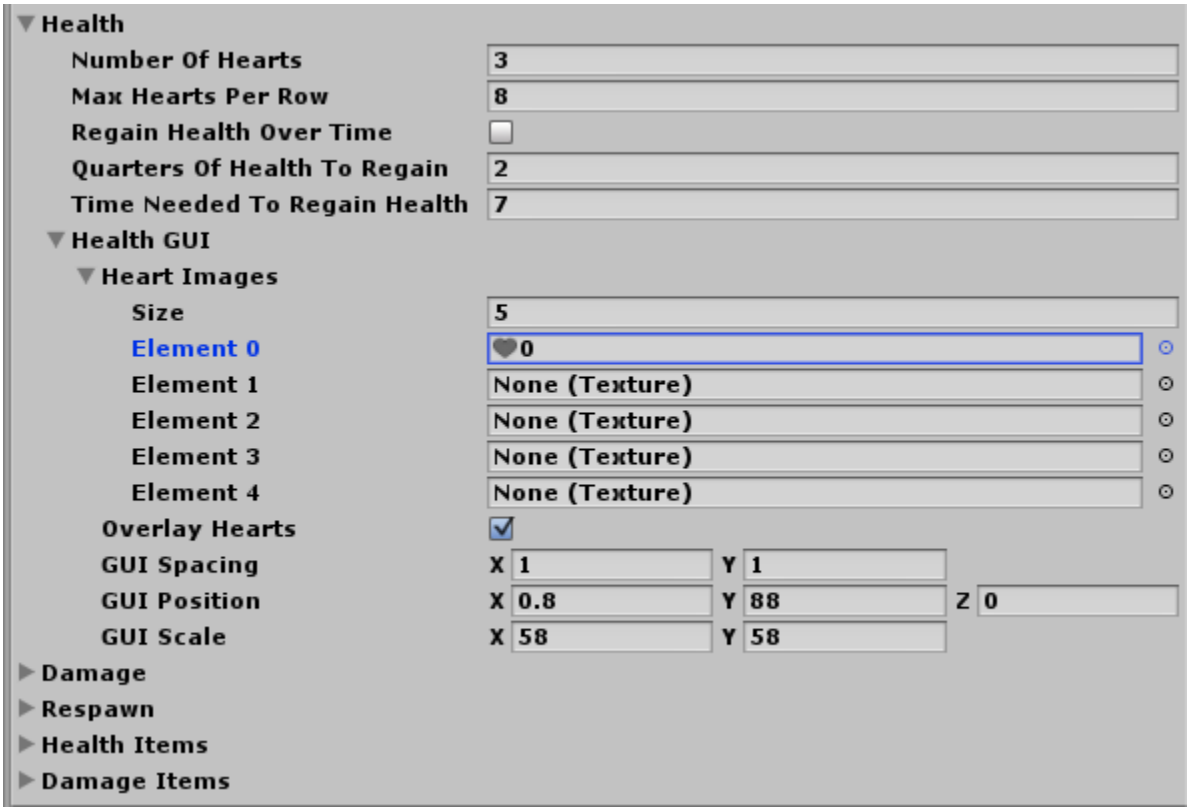
3. Set the Player Camera of the script to the camera you want to use for your player.



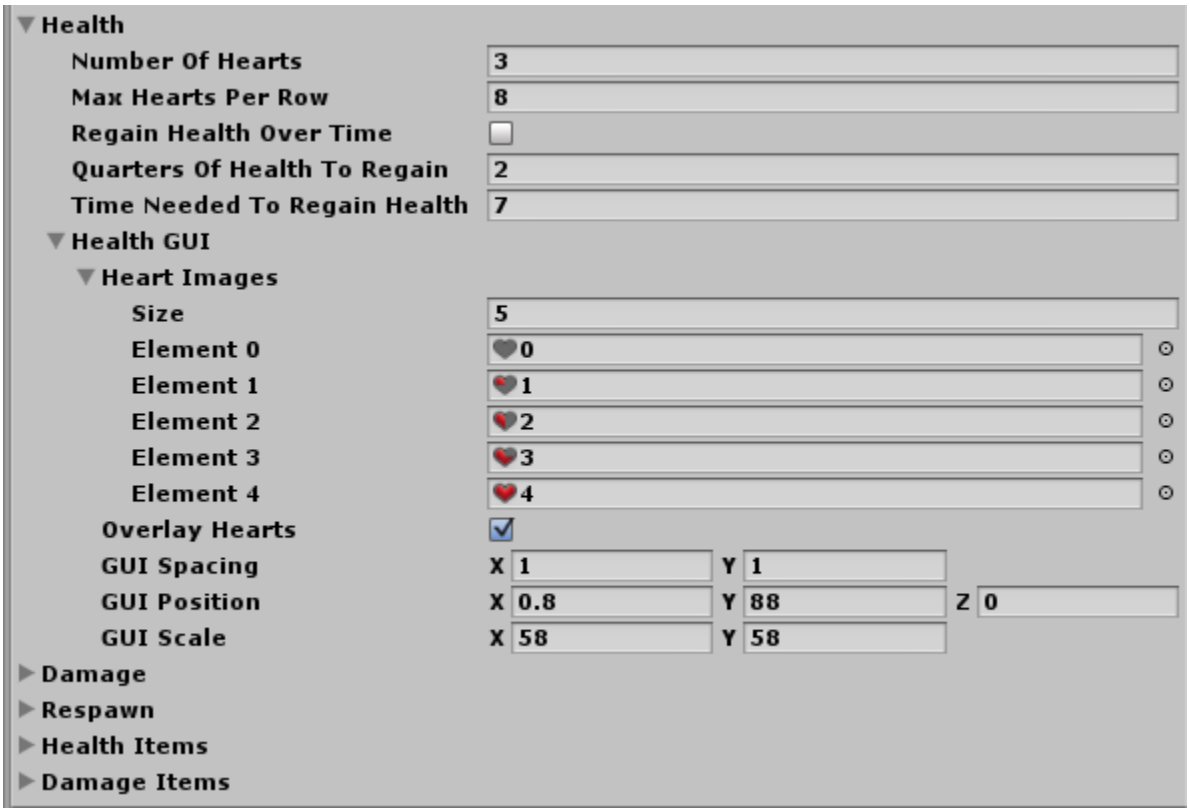
4. To set the GUI images for the health, expand the “Health” section of the Health.cs script, followed by the “Health GUI” section, then the “Heart Images” section. Now, set each heart texture to its corresponding number (0 being no health, and 4 being full health).





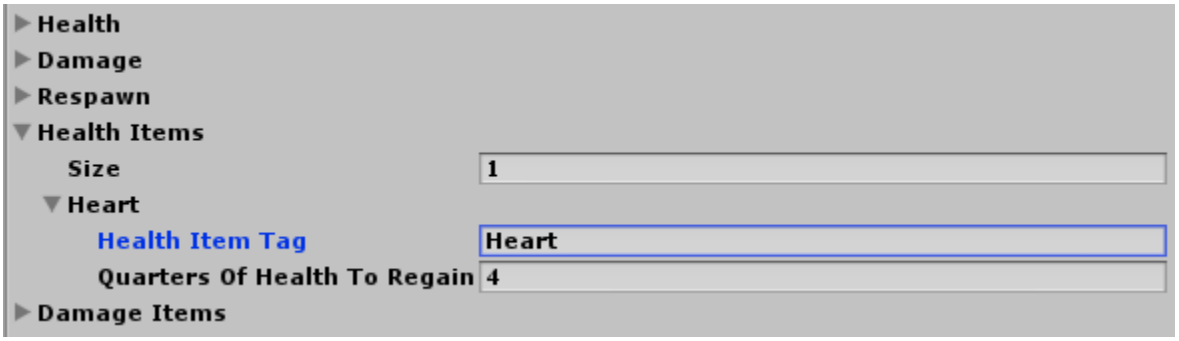


5. Repeat step 3 for every “Element,” until every heart image has been set.

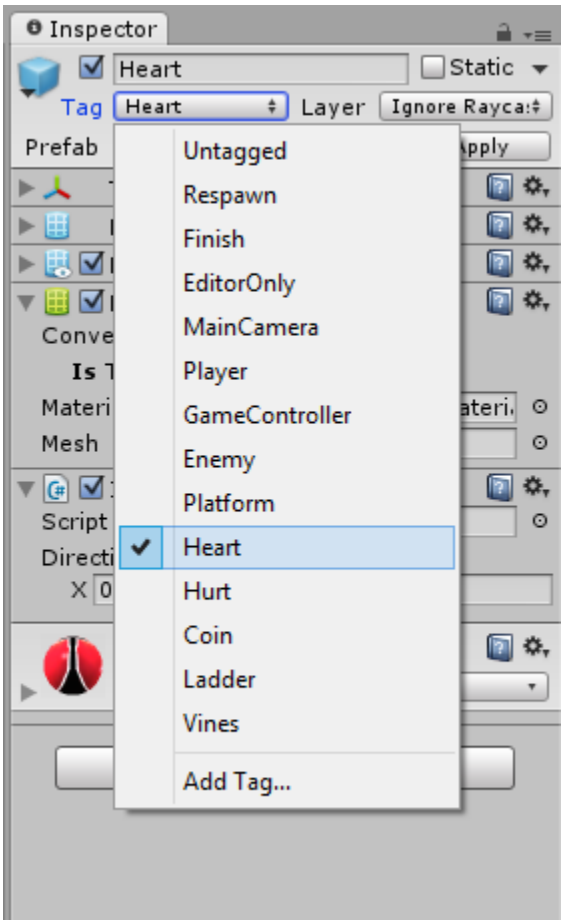


Defining which objects give health

1. In order for the player to know which objects give health, the user must define a tag in the "Health Items" section of the health script.

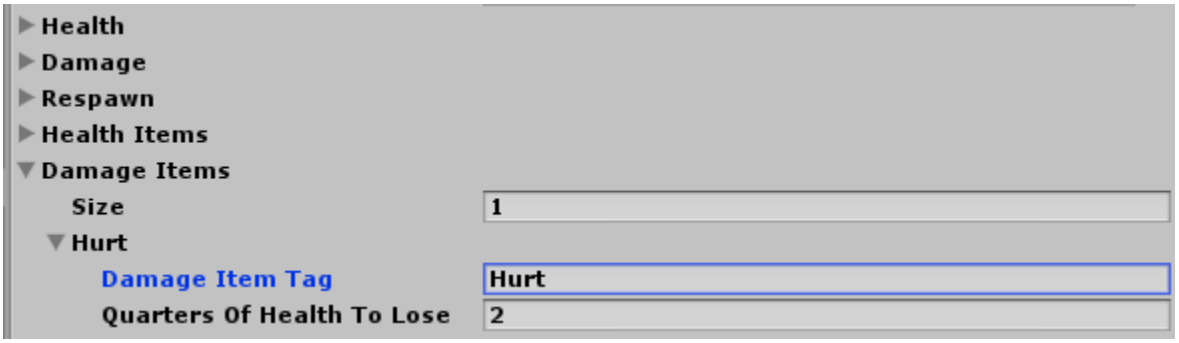


2. Once this tag is defined, you must assign this tag to whichever objects you wish to heal the player.

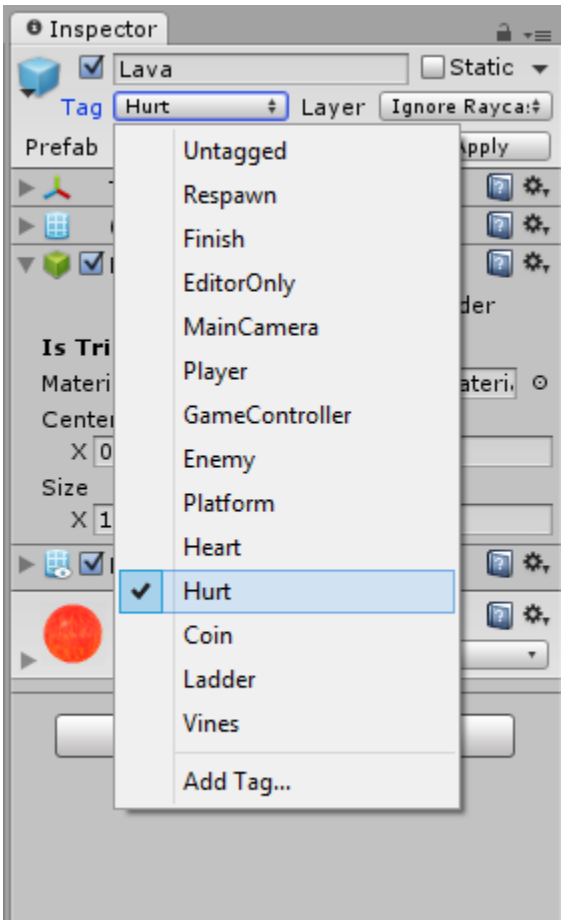


Defining which objects deal damage

1. In order for the player to know which objects deal damage, the user must define a tag in the “Damage Items” section of the health script.

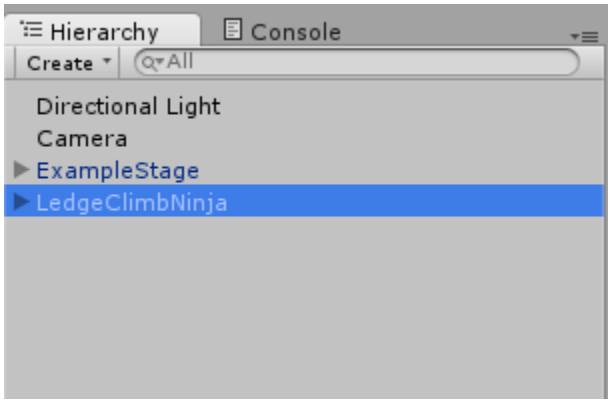


2. Once this tag is defined, you must assign this tag to whichever objects you wish to damage the player.

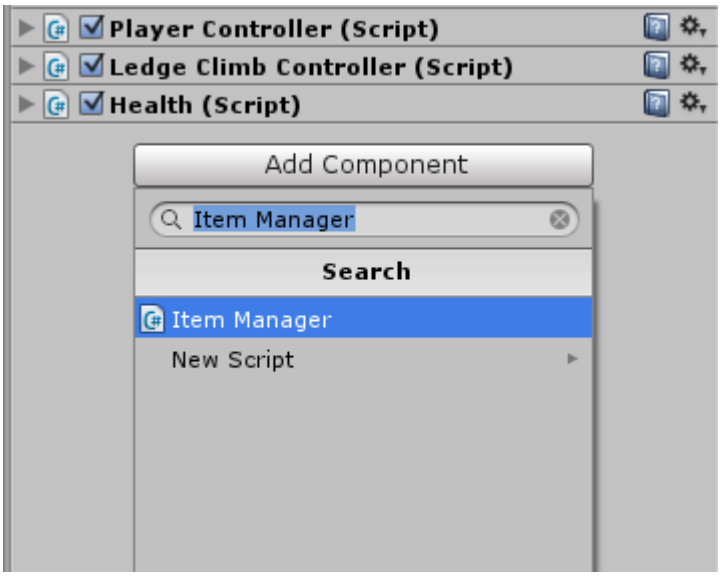


Setting up ItemManager.cs with your character

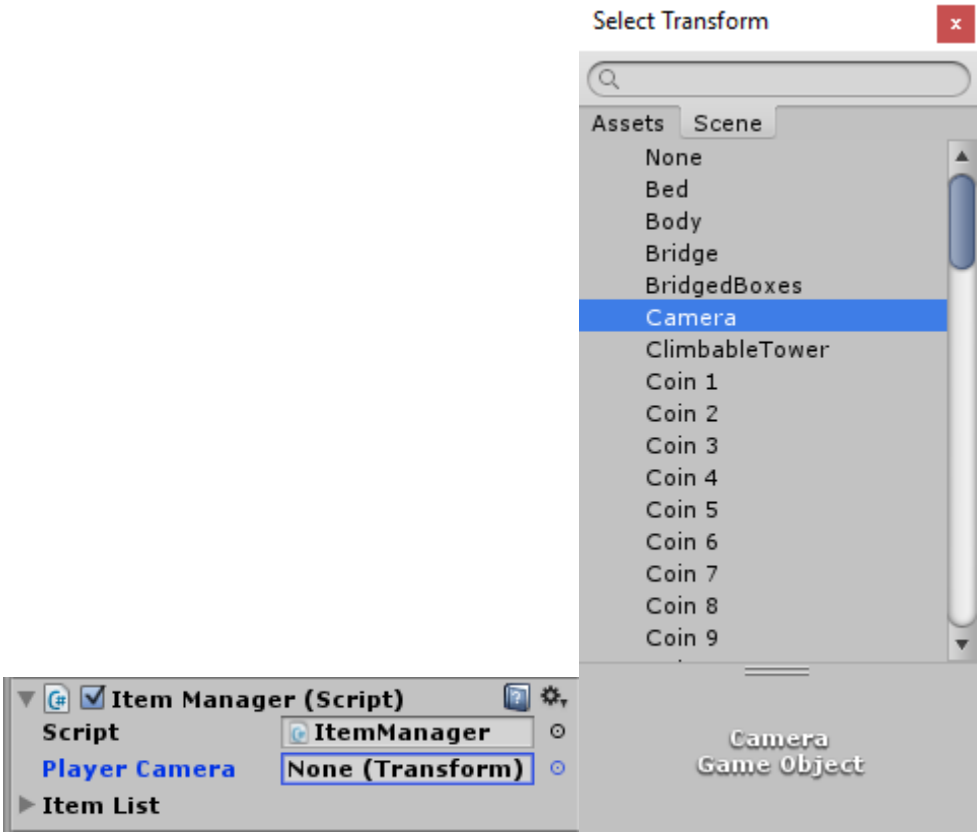
1. Select your character in the “Hierarchy” tab.



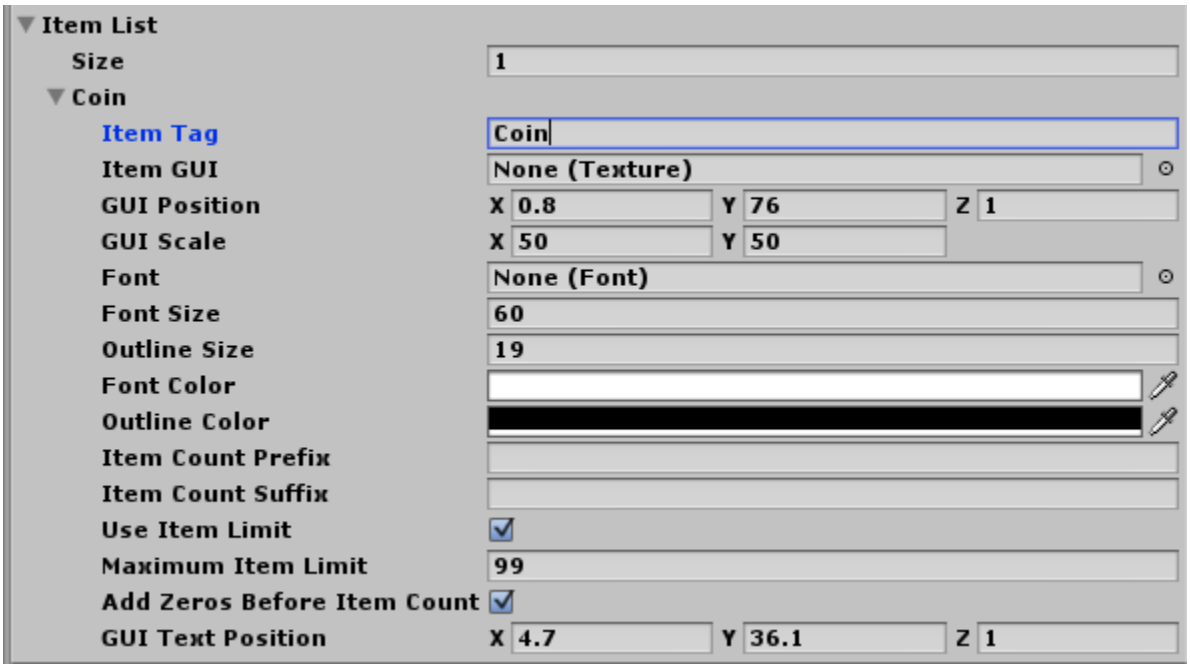
2. Add the ItemManager.cs script to your character (either by using “Add Component” in the “Inspector” tab or by dragging and dropping the script from the “Project” tab on to your player in the “Inspector” tab).

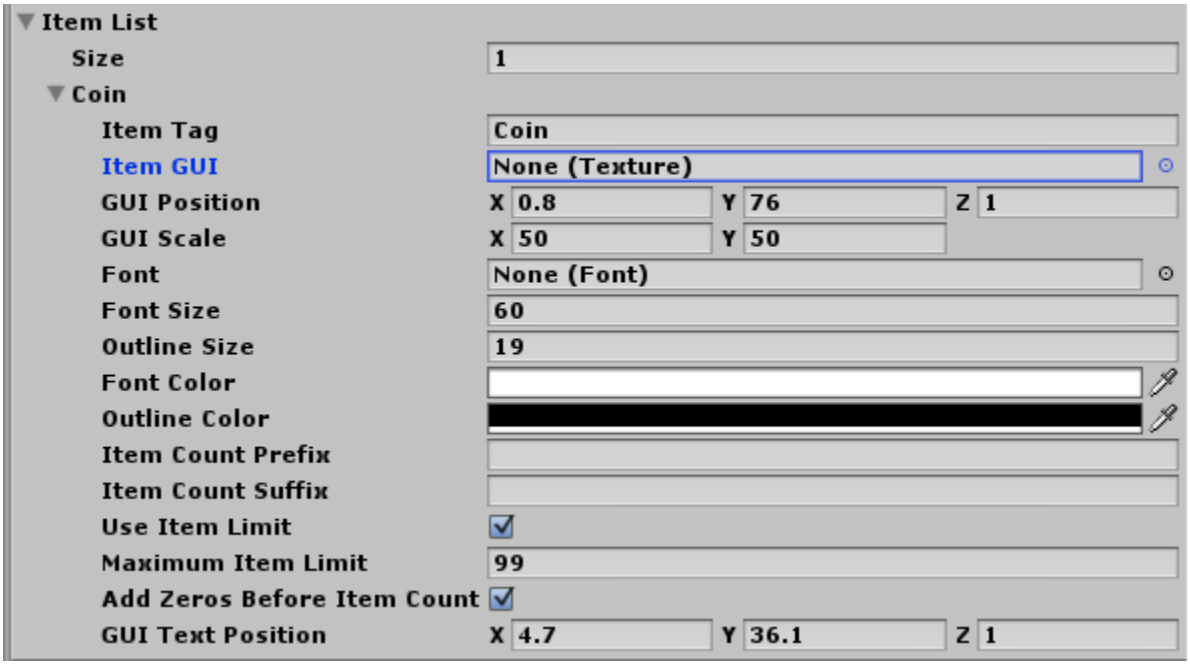


3. Set the Player Camera of the script to the camera you want to use for your player.

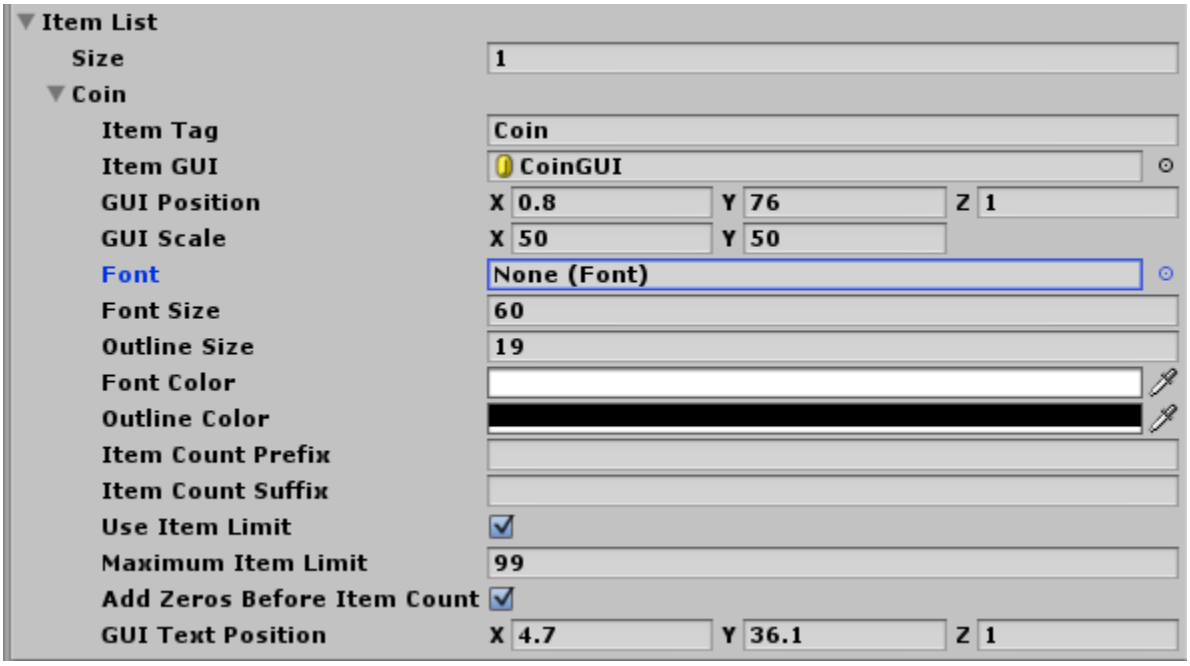


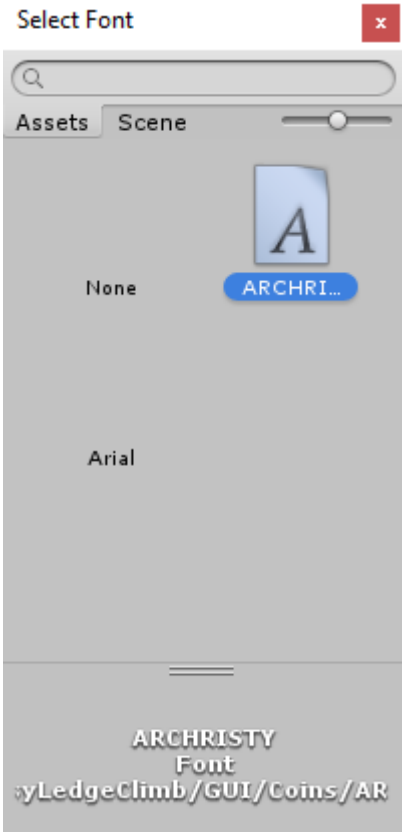
4. Set the tag, GUI image, and font for your items (in this case, we will be using coins as the item).


Item List
Size
1
Coin
Item Tag
Coin
Item GUI
None (Texture)
GUI Position
X 0.8
Y 76
Z 1
GUI Scale
X 50
Y 50
Font
None (Font)
Font Size
60
Outline Size
19
Font Color
Outline Color
Item Count Prefix
Item Count Suffix
Use Item Limit
Maximum Item Limit
99
Add Zeros Before Item Count
GUI Text Position
X 4.7
Y 36.1
Z 1


Item List
Size
1
Coin
Item Tag
Coin
Item GUI
None (Texture)
GUI Position
X 0.8
Y 76
Z 1
GUI Scale
X 50
Y 50
Font
None (Font)
Font Size
60
Outline Size
19
Font Color
Outline Color
Item Count Prefix
Item Count Suffix
Use Item Limit
Maximum Item Limit
99
Add Zeros Before Item Count
GUI Text Position
X 4.7
Y 36.1
Z 1

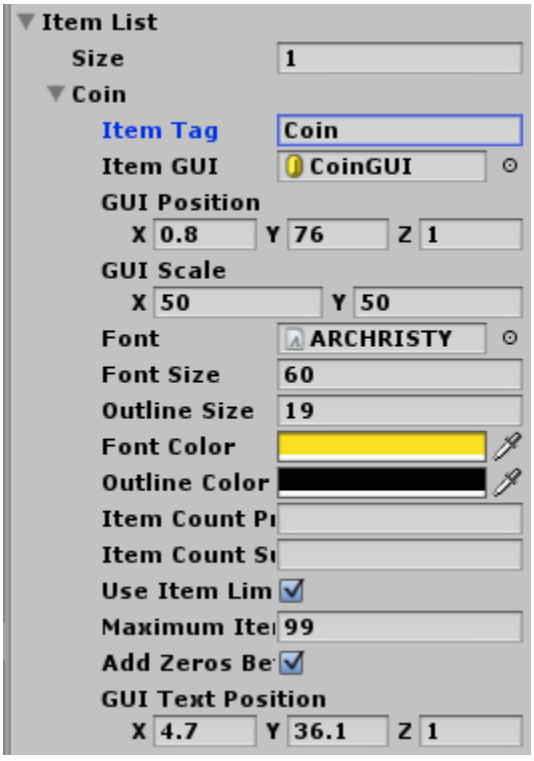

Select Texture
Assets
Scene
ButtonTi...
Coin
CoinGUI
DustParti...
Enemy
Heart
Texture 2D
256x256 ARGB 32 bit
341.3 KB


Item List
Size
1
Coin
Item Tag
Coin
Item GUI
CoinGUI
GUI Position
X 0.8
Y 76
Z 1
GUI Scale
X 50
Y 50
Font
None (Font)
Font Size
60
Outline Size
19
Font Color
Outline Color
Item Count Prefix
Item Count Suffix
Use Item Limit
Maximum Item Limit
99
Add Zeros Before Item Count
GUI Text Position
X 4.7
Y 36.1
Z 1

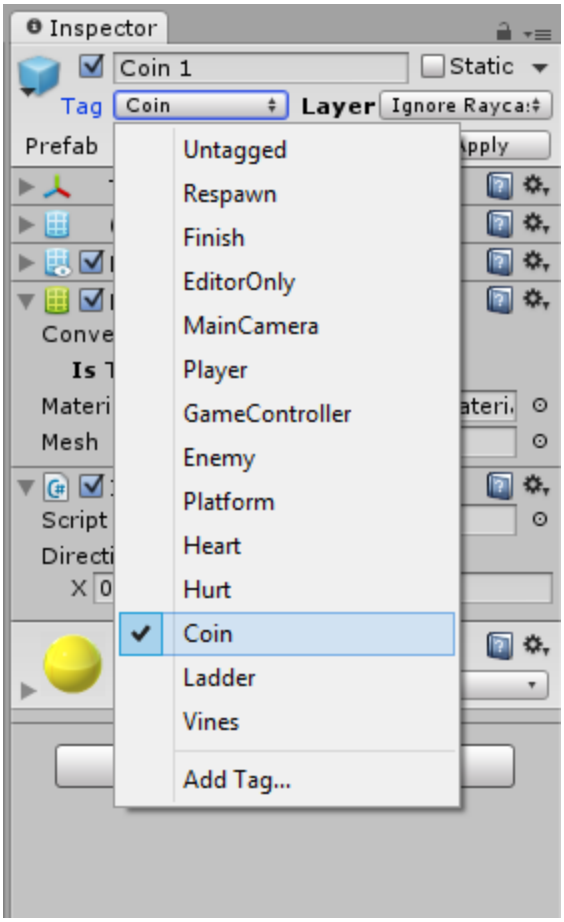


Defining which objects are items

1. In order for the player to know which objects are your items, the user must define a tag in the “Item List” section of the item manager.



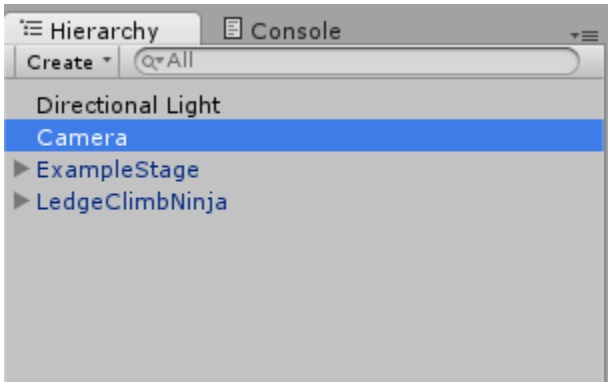
2. Once this tag is defined, you must assign this tag to whichever objects you wish to use as items.



Setting Up the All in One Game Kit - ELC Character System with a Camera

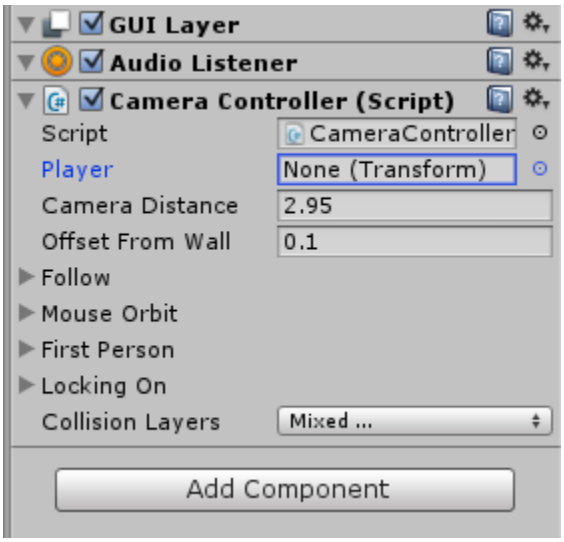
Adding CameraController.cs to your camera

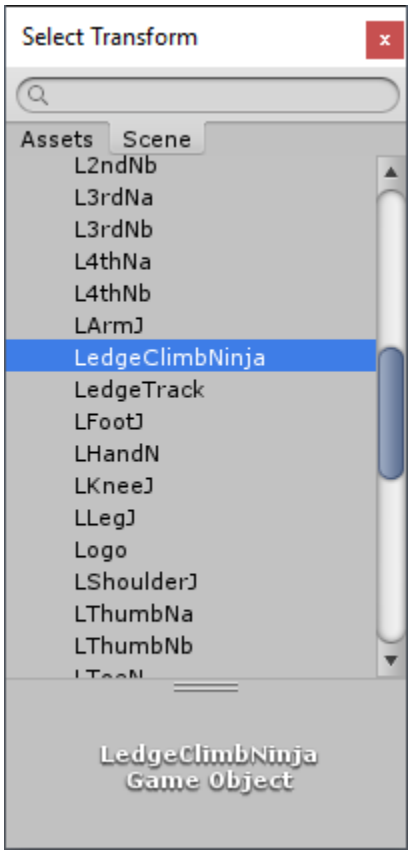
1. Select your camera in the “Hierarchy” tab.



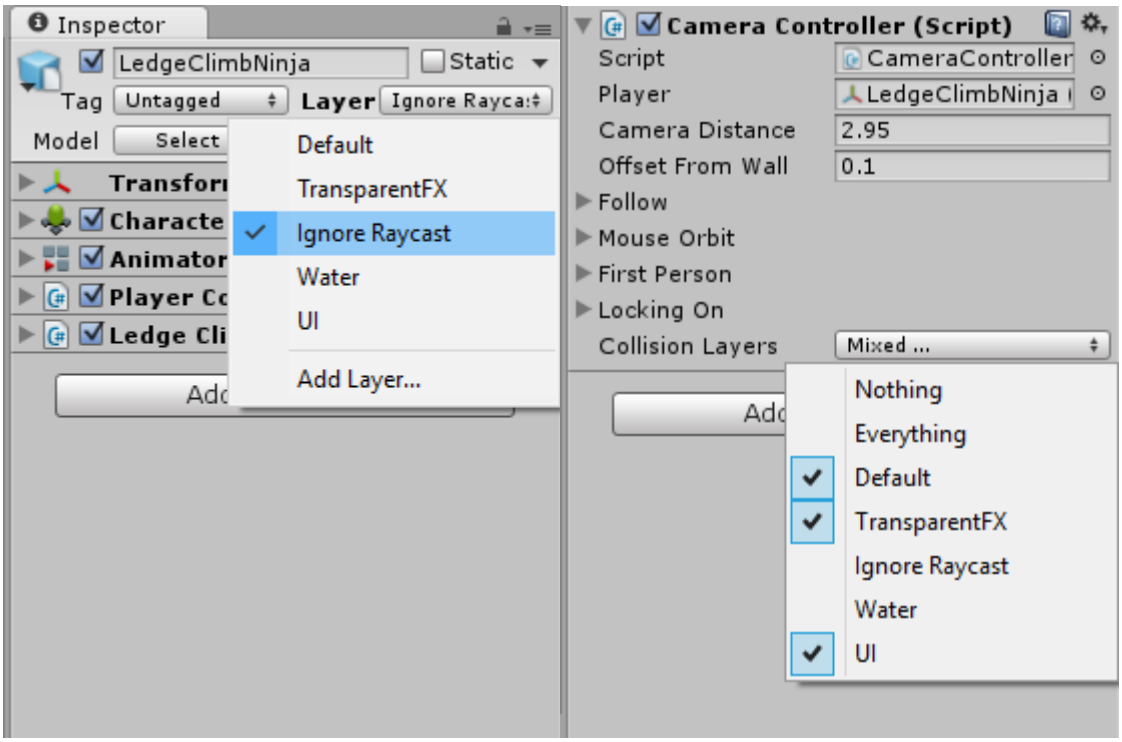
2. Add the CameraController.cs script to the camera (either by using “Add Component” in the “Inspector” tab or by dragging and dropping the script from the “Project” tab on to your camera in the “Inspector” tab).

3. Set the Player of the script to your player (in this case, the player is named “LedgeClimbNinja”).



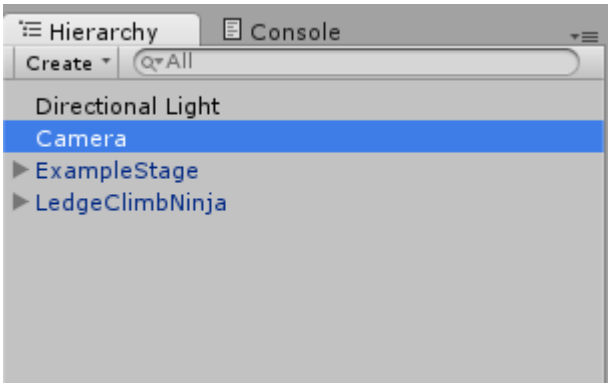


4. Set a layer for your player and uncheck it in the script’s Collision Layers (to avoid having the script’s linecasts and raycasts collide with the player himself).

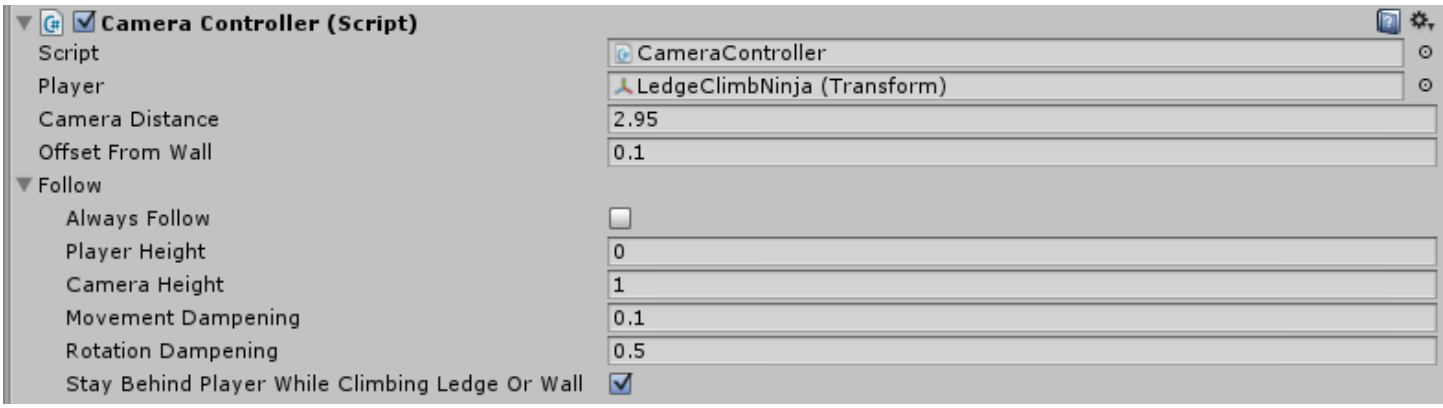


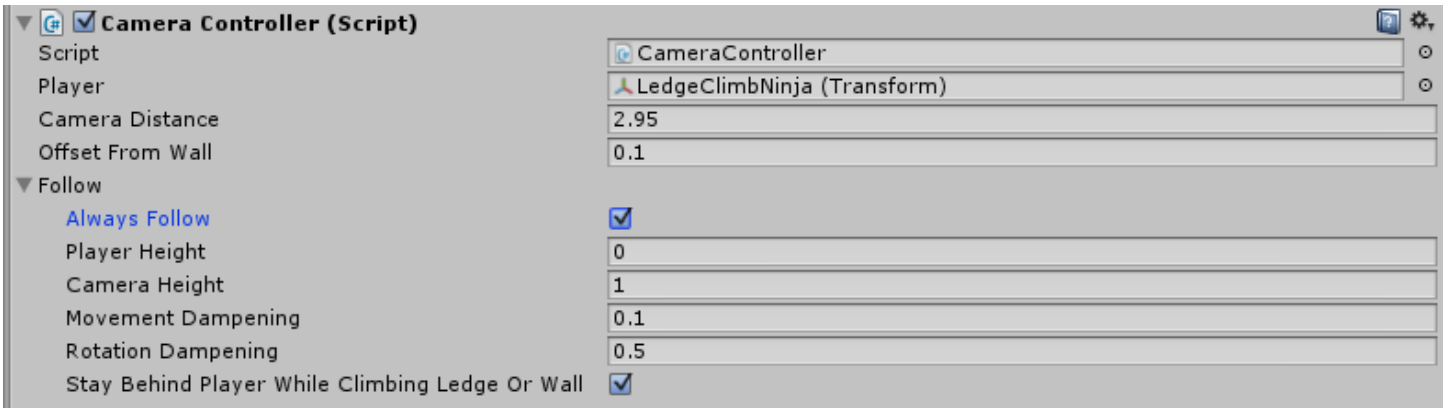
Setting up a following camera

1. Select your camera in the “Hierarchy” tab.



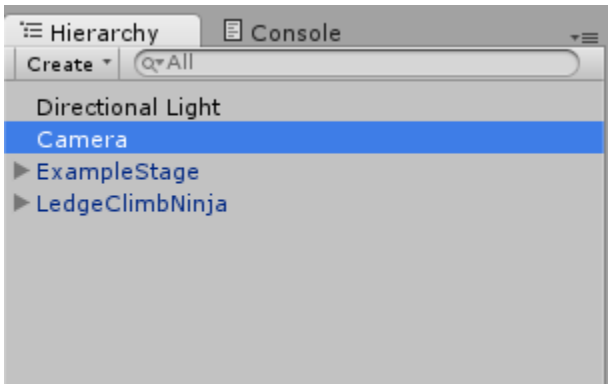
2. In the “Camera Controller” script we just added, check the box that says “Always Follow.”



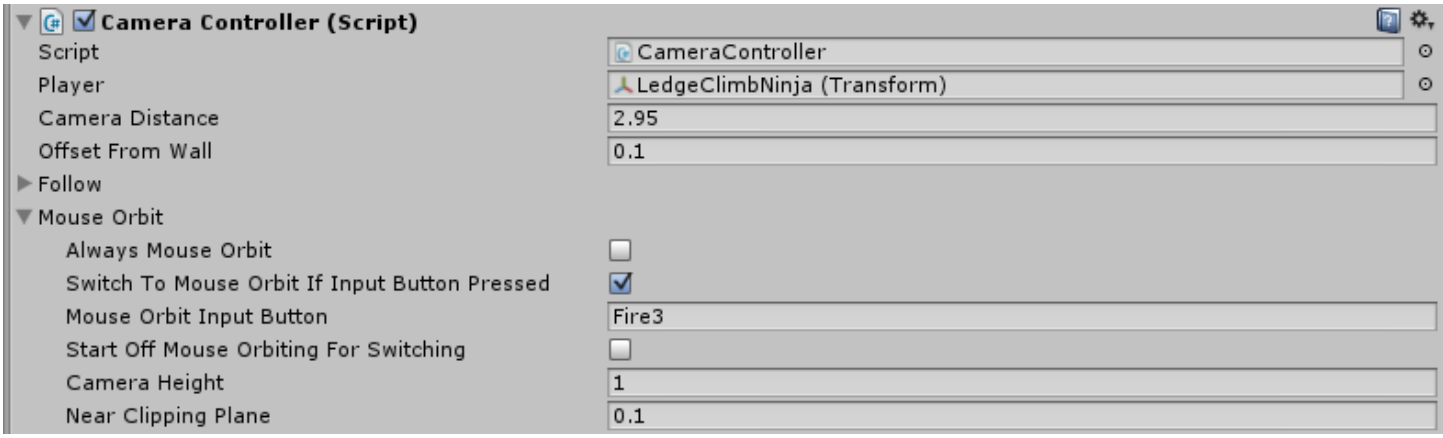


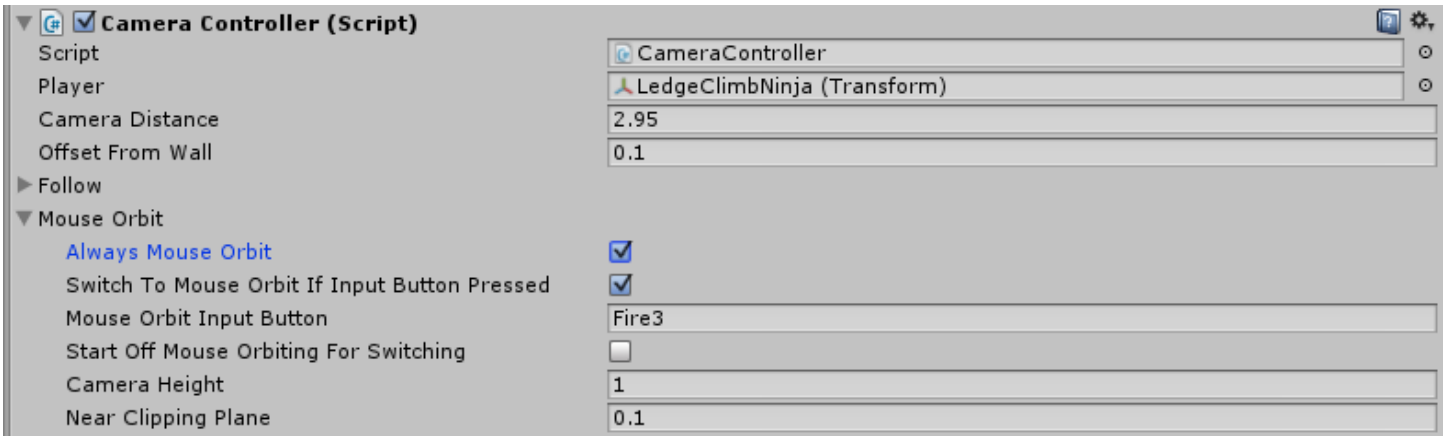
Setting up a mouse orbiting camera

1. Select your camera in the "Hierarchy" tab.



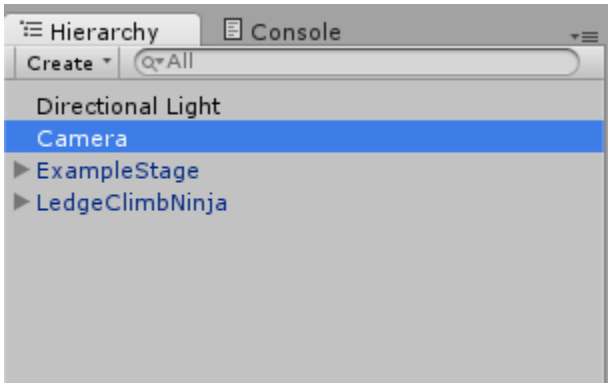
2. In the "Camera Controller" script we just added, check the box that says "Always Mouse Orbit" (or, if you would like to switch to mouse orbiting by pressing a button instead of always mouse orbiting, check "Switch To Mouse Orbit If Input Button Pressed," and leave "Always Mouse Orbit" unchecked).



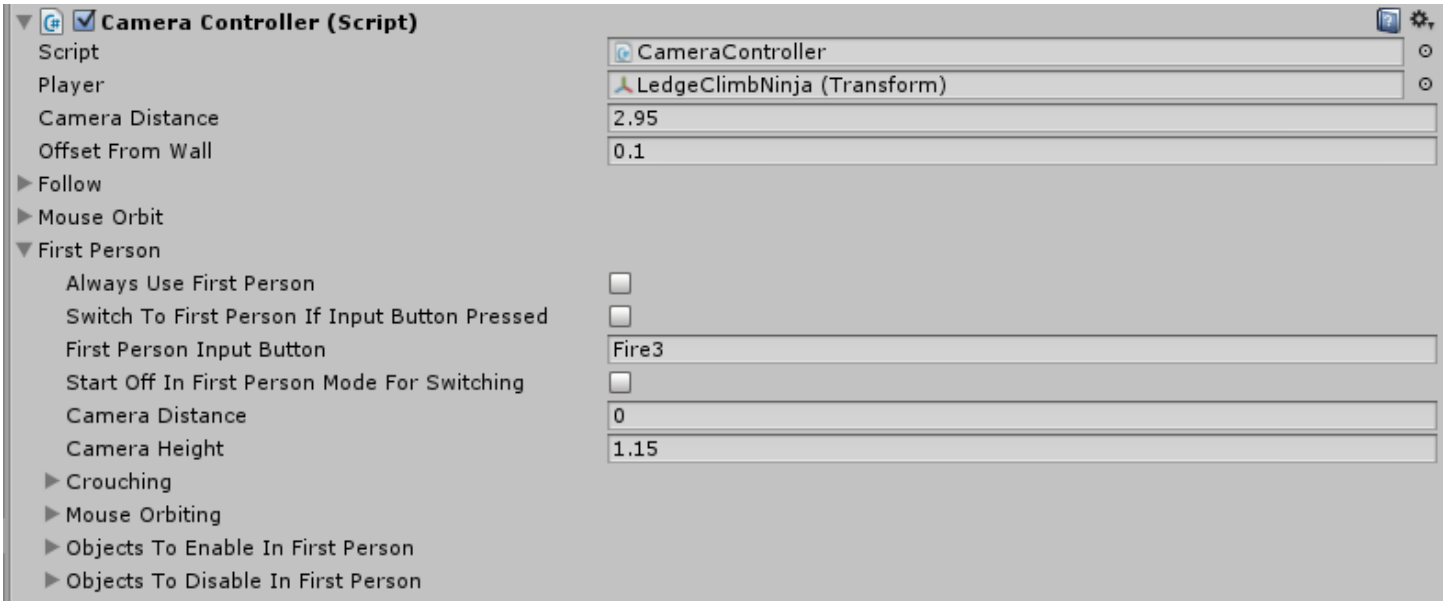


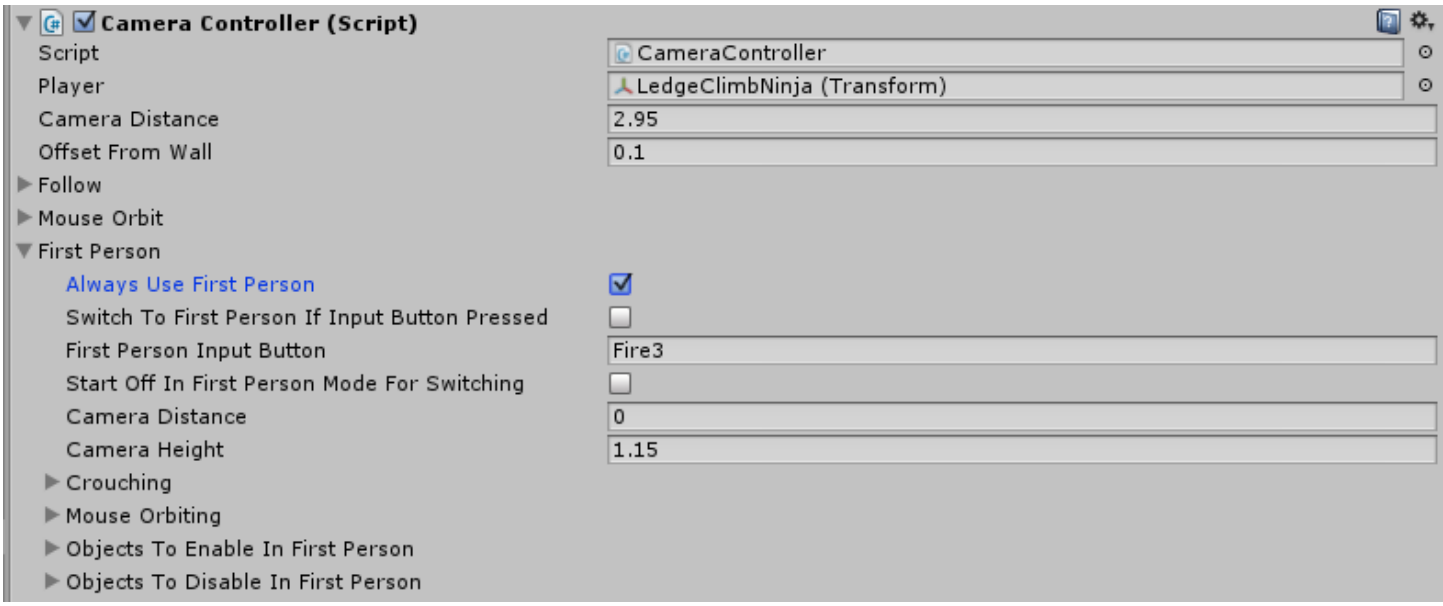
Setting up a first person camera

1. Select your camera in the "Hierarchy" tab.

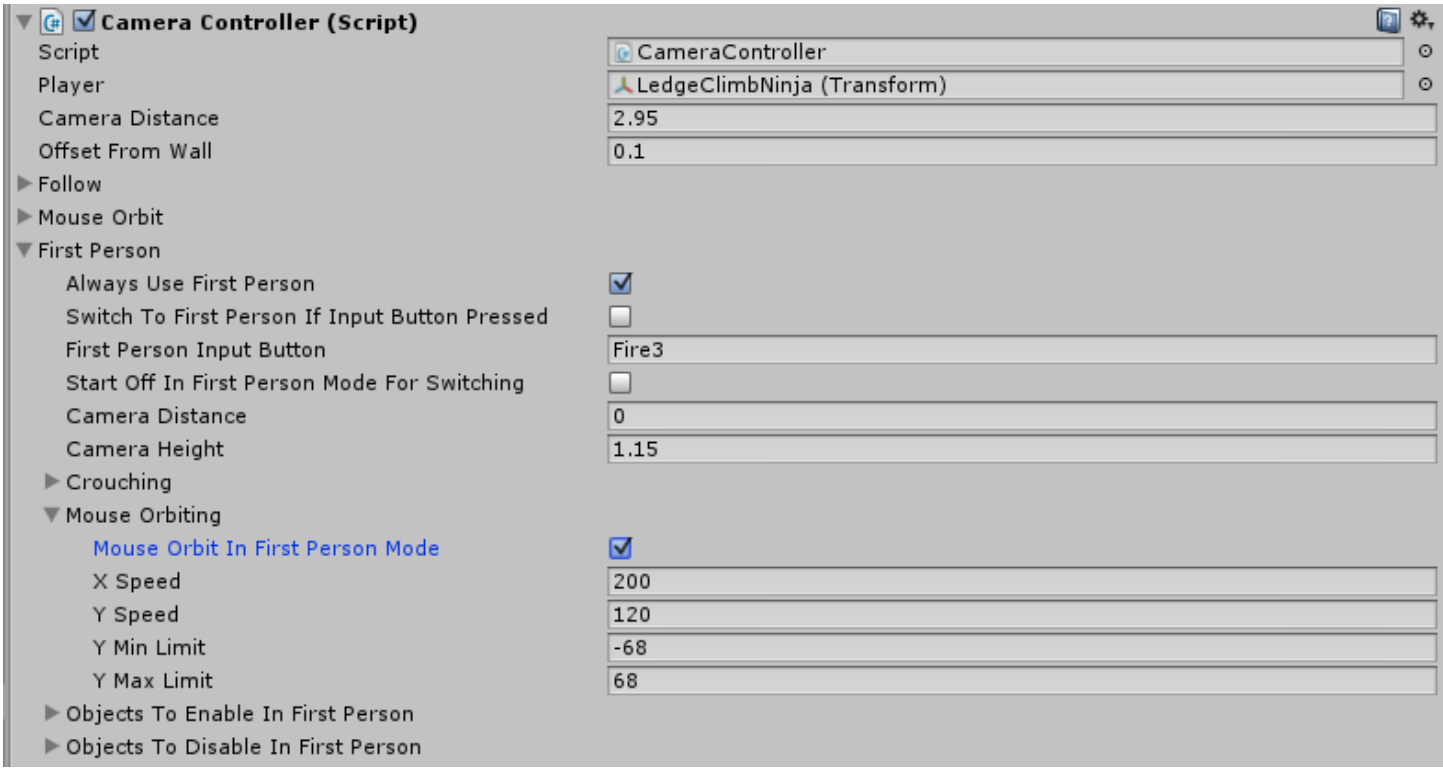


2. In the "Camera Controller" script we just added, check the box that says "Always Use First Person" (or, if you would like to switch to mouse orbiting by pressing a button instead of always mouse orbiting, check "Switch To First Person If Input Button Pressed," and leave "Always Mouse Orbit" unchecked).



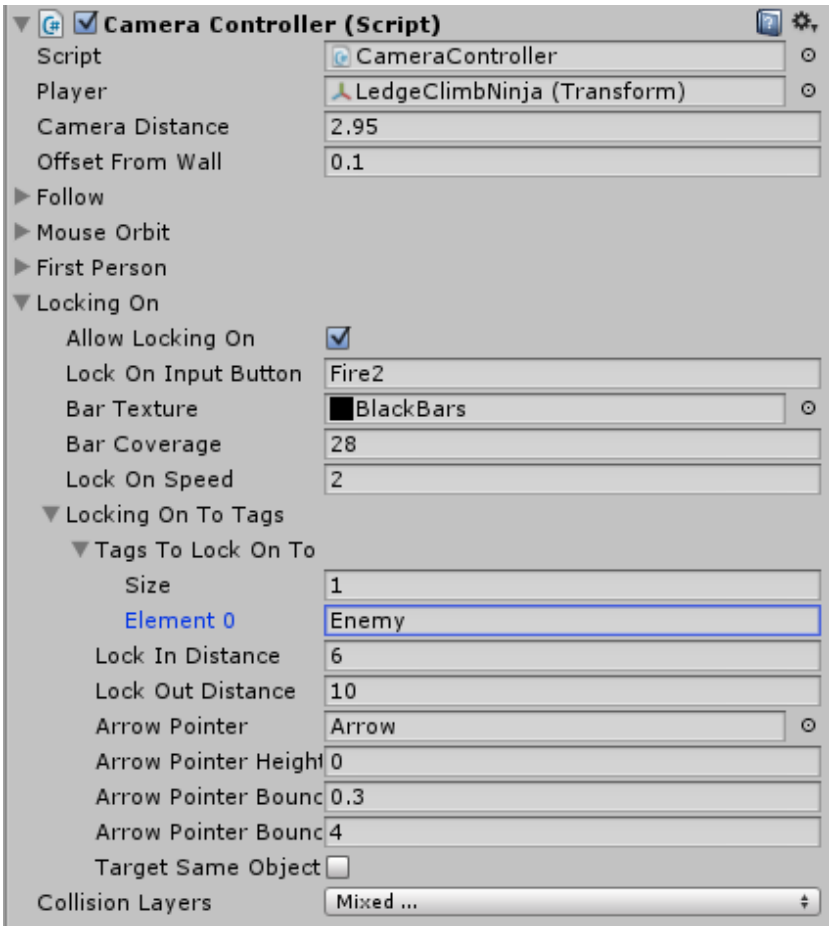


3. (OPTIONAL)
   If you would like camera to mouse orbit in first person mode, expand the “Mouse Orbiting” section of the “First Person” category, then make sure that “Mouse Orbit In First Person Mode” is checked.

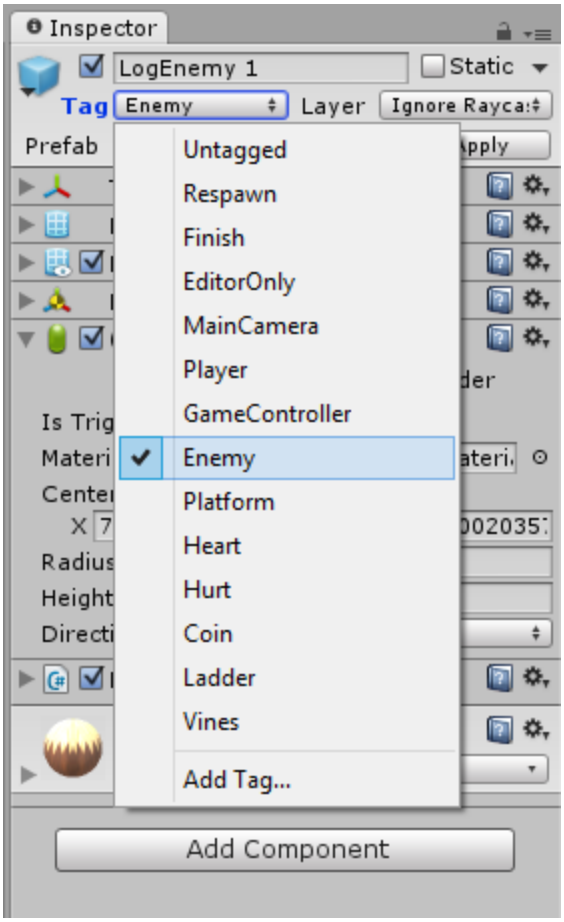


Defining which objects can be locked on to

1. In order for the camera to know which objects can be locked on to, the user must define a tag in the “Locking On” section of the camera controller.



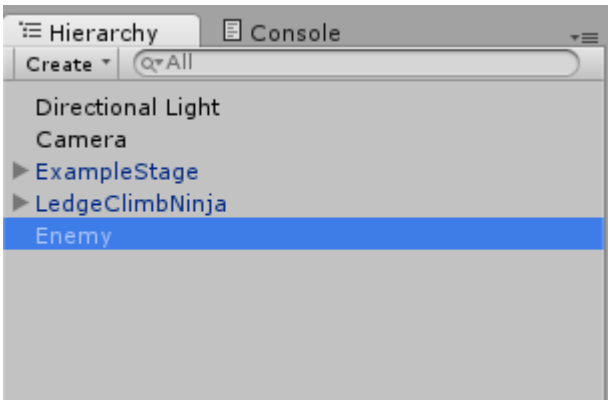
2. Once this tag is defined, you must assign this tag to whichever objects you wish to lock on to.



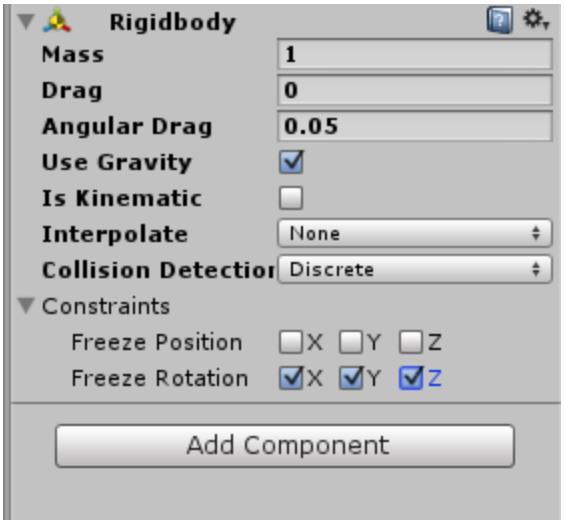
Setting Up the All in One Game Kit - ELC Character System with a New Enemy

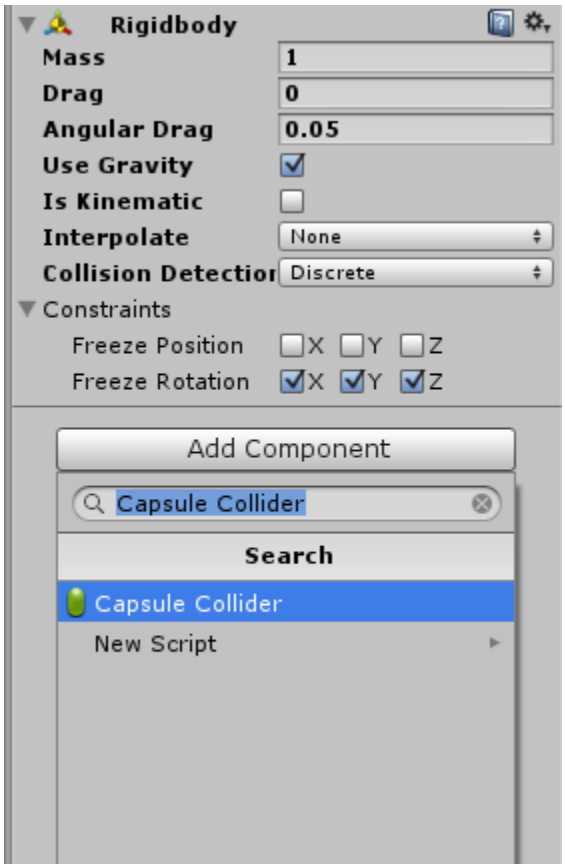
Making the Enemy AI script work with your enemy

1. Select your enemy in the “Hierarchy” tab.

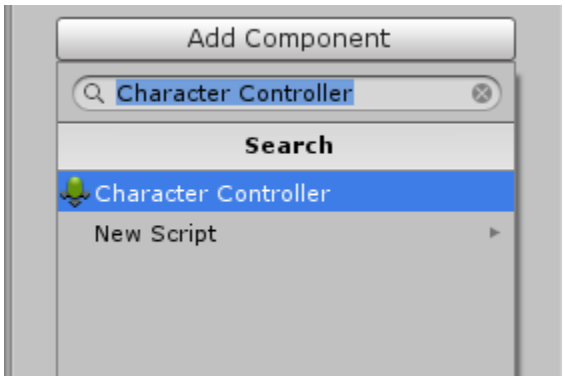


2. If your enemy does not already have one, add a Rigidbody **OR** CharacterController component to your enemy (I recommend using a Rigidbody for enemies) from the “Inspector” tab (if you use a Rigidbody, you must freeze your enemy’s rotation and add a collider component as well).
   Rigidbody:



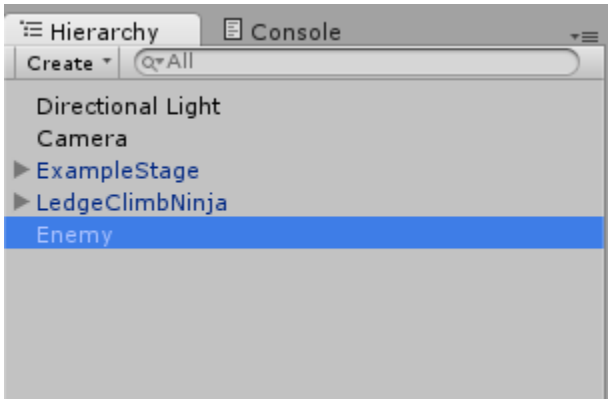


Character Controller:

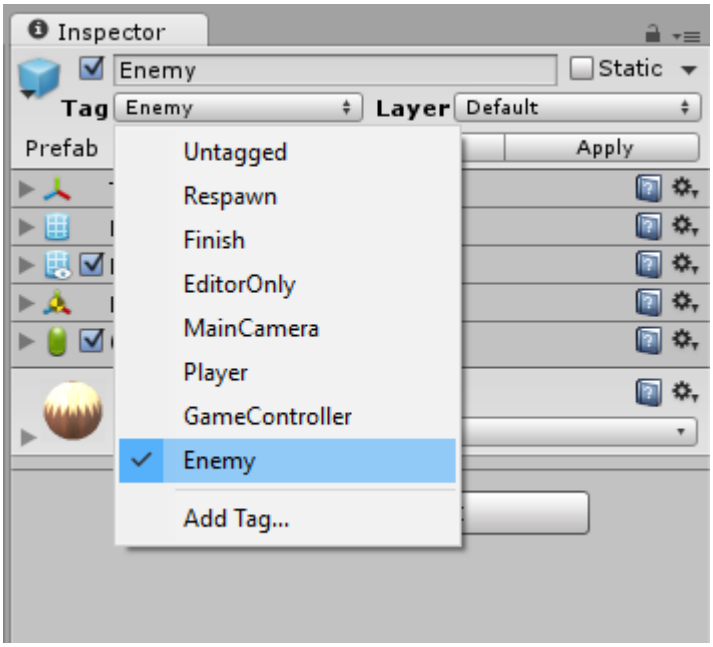


Setting the enemy's tag (so that it works with the player's Health.cs script)

1. Select your enemy in the "Hierarchy" tab.

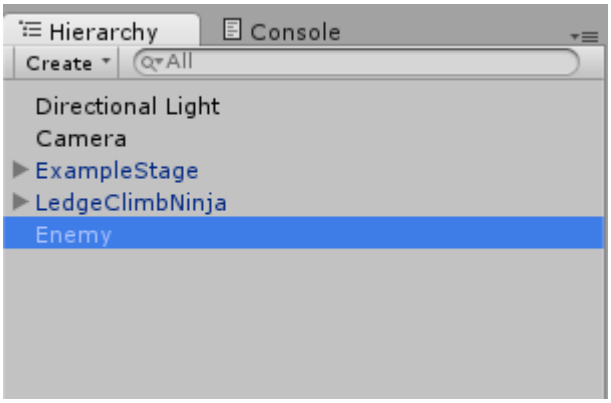


2. Set the tag of your enemy to "Enemy" (by default, this is the tag that the player's Health.cs script recognizes as an enemy. However, if you would like to change this tag, you can do so by changing the "Enemy Tag" (located in the Damage category of the Health.cs script) in the Health.cs script from "Enemy" to your new tag).

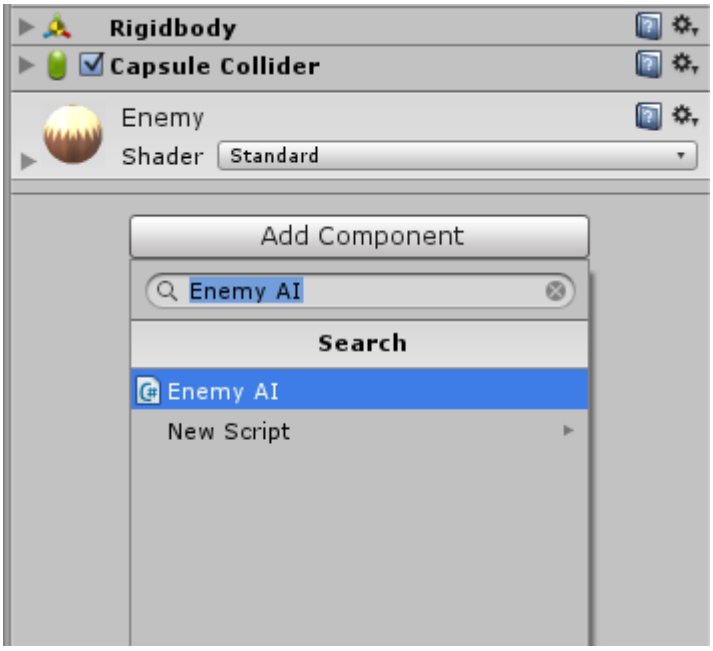


Setting up EnemyAI.cs with your enemy

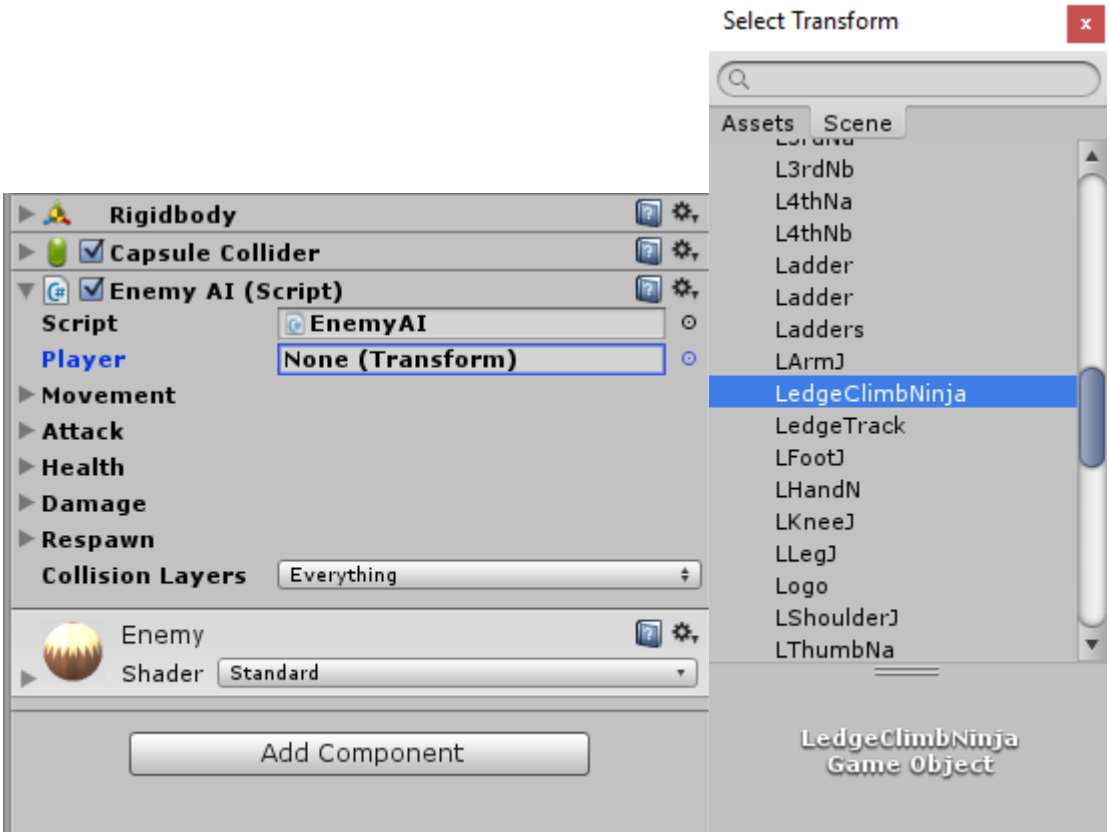
1. Select your enemy in the "Hierarchy" tab.



2. Add the EnemyAI.cs script to your enemy (either by using “Add Component” in the “Inspector” tab or by dragging and dropping the script from the “Project” tab on to your enemy in the “Inspector” tab).

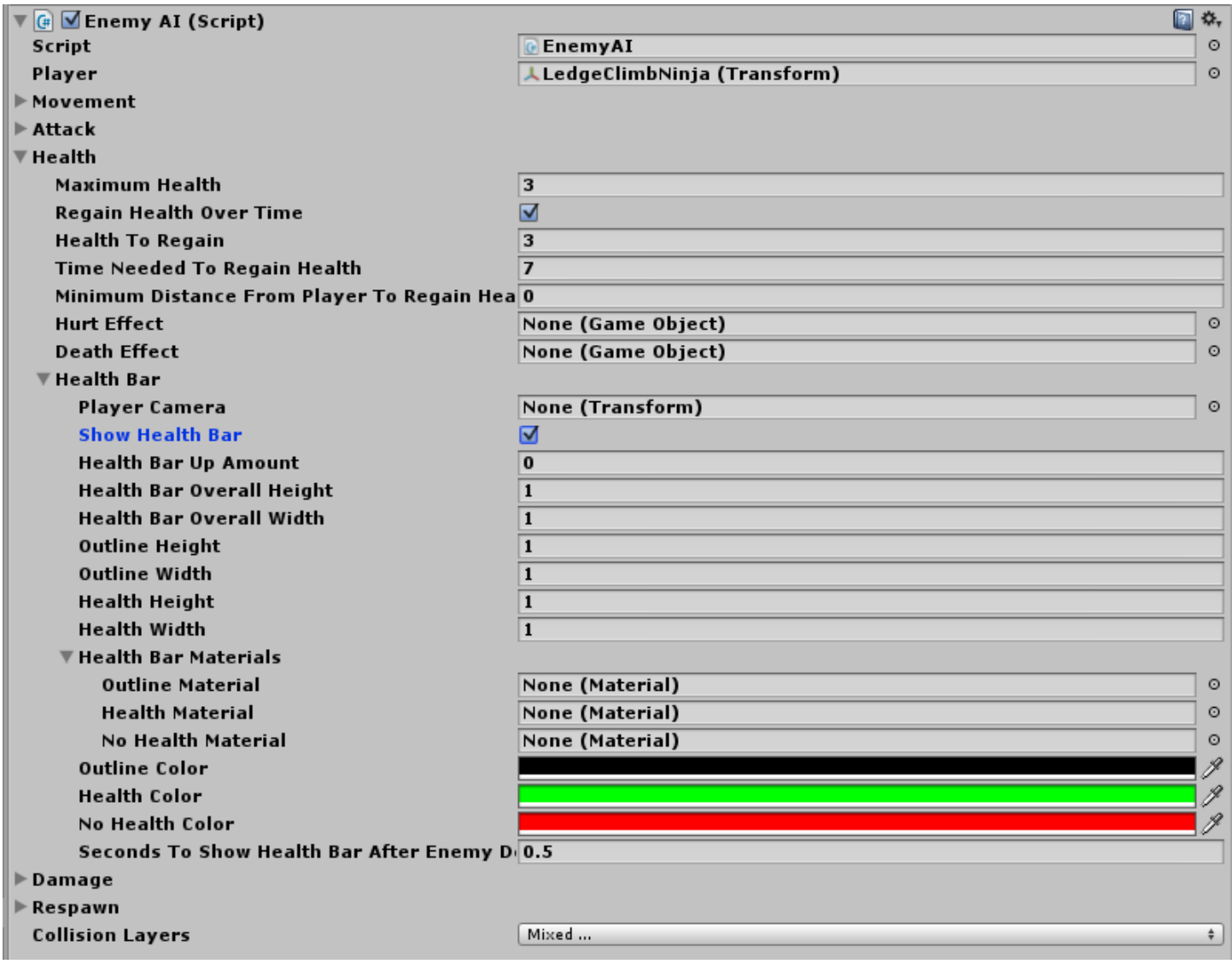


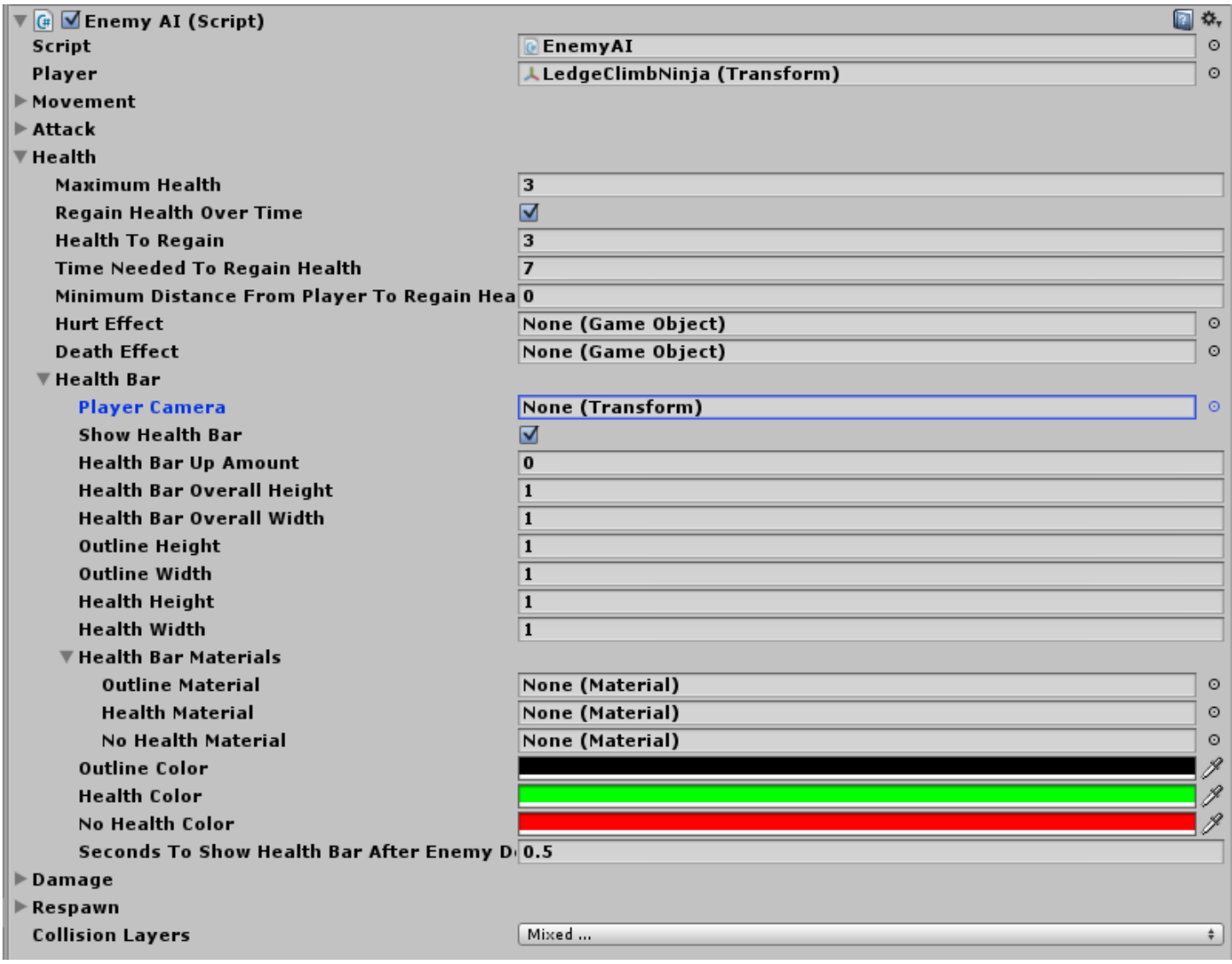
3. Set the Player of the script to your player (in this case, we will set it to LedgeClimbNinja).

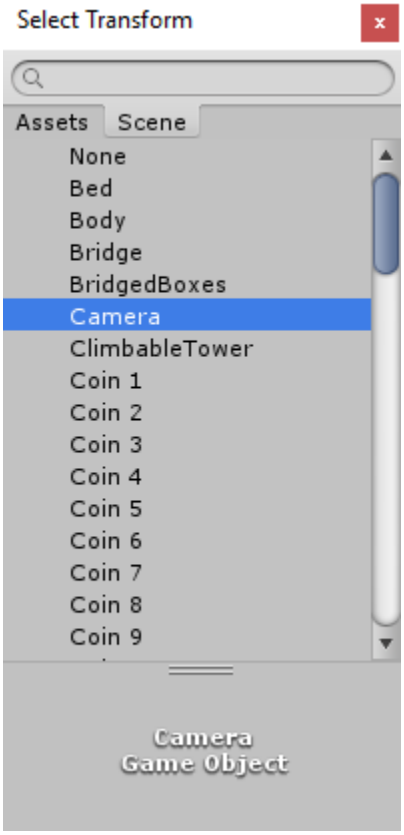


4. Set a layer for your enemy and uncheck it in the script’s Collision Layers (to avoid having the script’s linecasts and raycasts collide with the enemy itself).

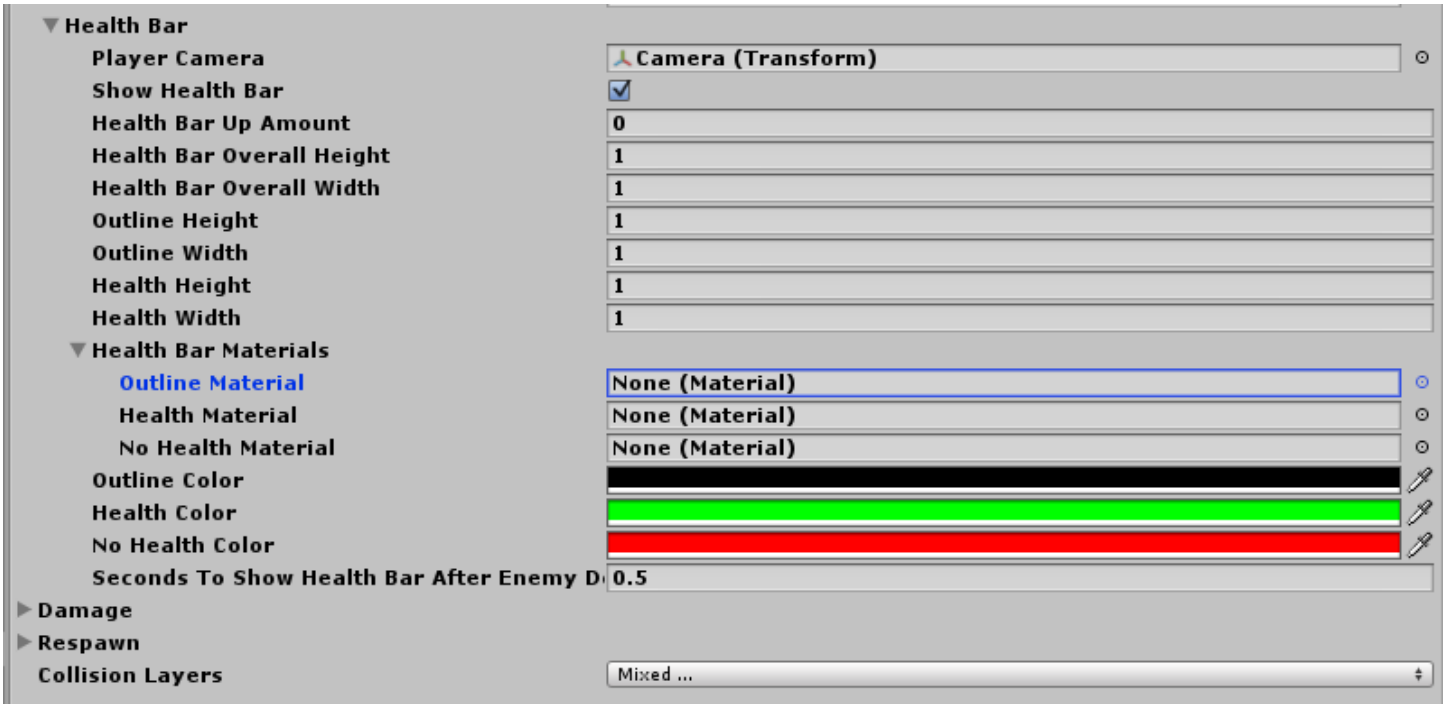
5. (OPTIONAL)
   If you would like the enemy to have a health bar float over his head, expand the “Health Bar” section of the “Health” category, then make sure that “Show Health Bar” is checked, and that the “Player Camera” is set.


Enemy AI (Script)
Script
EnemyAI
Player
LedgeClimbNinja (Transform)
Movement
Attack
Health
Maximum Health
3
Regain Health Over Time
Health To Regain
3
Time Needed To Regain Health
7
Minimum Distance From Player To Regain Hea
0
Hurt Effect
None (Game Object)
Death Effect
None (Game Object)
Health Bar
Player Camera
None (Transform)
Show Health Bar
Health Bar Up Amount
0
Health Bar Overall Height
1
Health Bar Overall Width
1
Outline Height
1
Outline Width
1
Health Height
1
Health Width
1
Health Bar Materials
Outline Material
None (Material)
Health Material
None (Material)
No Health Material
None (Material)
Outline Color
Health Color
No Health Color
Seconds To Show Health Bar After Enemy D
0.5
Damage
Respawn
Collision Layers
Mixed ...


Enemy AI (Script)
Script
EnemyAI
Player
LedgeClimbNinja (Transform)
Movement
Attack
Health
Maximum Health
3
Regain Health Over Time
Health To Regain
3
Time Needed To Regain Health
7
Minimum Distance From Player To Regain Hea
0
Hurt Effect
None (Game Object)
Death Effect
None (Game Object)
Health Bar
Player Camera
None (Transform)
Show Health Bar
Health Bar Up Amount
0
Health Bar Overall Height
1
Health Bar Overall Width
1
Outline Height
1
Outline Width
1
Health Height
1
Health Width
1
Health Bar Materials
Outline Material
None (Material)
Health Material
None (Material)
No Health Material
None (Material)
Outline Color
Health Color
No Health Color
Seconds To Show Health Bar After Enemy D
0.5
Damage
Respawn
Collision Layers
Mixed ...

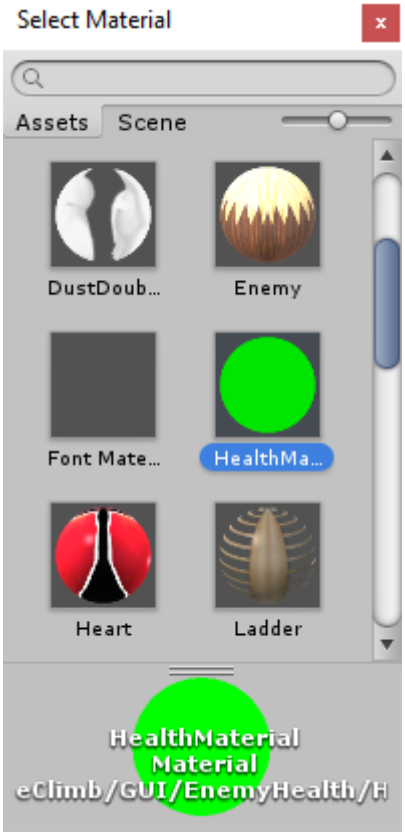


You must also make sure that the “Health Bar Materials” are set (so that the script knows which materials to use for the health bar).




Select Material
Assets
Scene
mountains
mountains
NoHealth...
OutlineMa...
standard
standard
OutlineMaterial
Material
eClimb/GUI/EnemyHealth/O


Health Bar
Player Camera
Camera (Transform)
Show Health Bar
Health Bar Up Amount
0
Health Bar Overall Height
1
Health Bar Overall Width
1
Outline Height
1
Outline Width
1
Health Height
1
Health Width
1
Health Bar Materials
Outline Material
OutlineMaterial
Health Material
None (Material)
No Health Material
None (Material)
Outline Color
Health Color
No Health Color
Seconds To Show Health Bar After Enemy D
0.5
Damage
Respawn
Collision Layers
Mixed ...


Select Material
Assets
Scene
DustDoub...
Enemy
Font Mate...
HealthMa...
Heart
Ladder
HealthMaterial
Material
eClimb/GUI/EnemyHealth/H

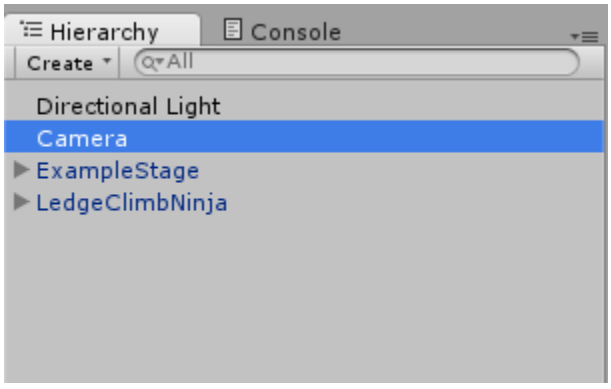

Health Bar
Player Camera
Camera (Transform)
Show Health Bar
Health Bar Up Amount
0
Health Bar Overall Height
1
Health Bar Overall Width
1
Outline Height
1
Outline Width
1
Health Height
1
Health Width
1
Health Bar Materials
Outline Material
OutlineMaterial
Health Material
HealthMaterial
No Health Material
None (Material)
Outline Color
Health Color
No Health Color
Seconds To Show Health Bar After Enemy D
0.5
Damage
Respawn
Collision Layers
Mixed ...

Setting Up the All in One Game Kit - ELC Character System with Different Perspectives

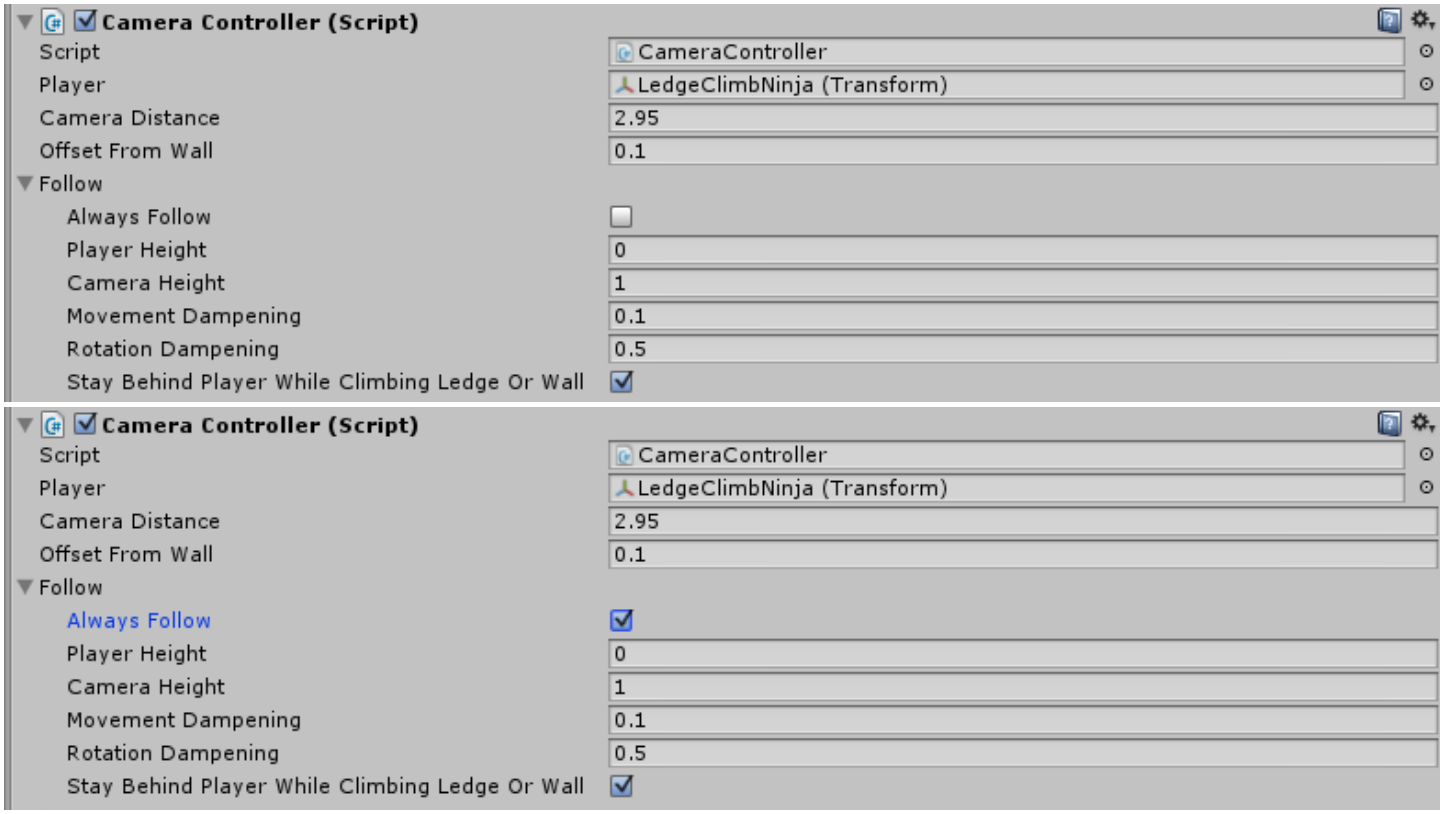
Third Person Mode

- **Third person mode with a following camera:**

1. Select your camera in the “Hierarchy” tab.

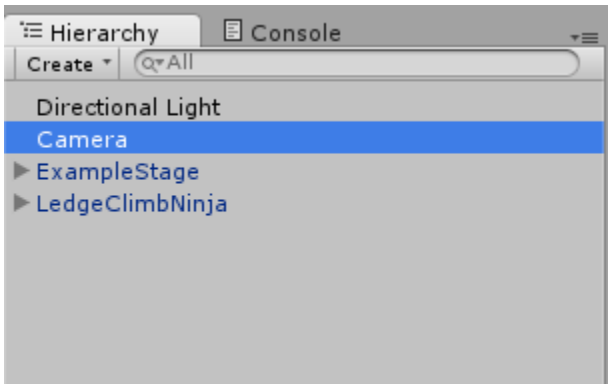


2. In the “Camera Controller” script we just added, check the box that says “Always Follow.”

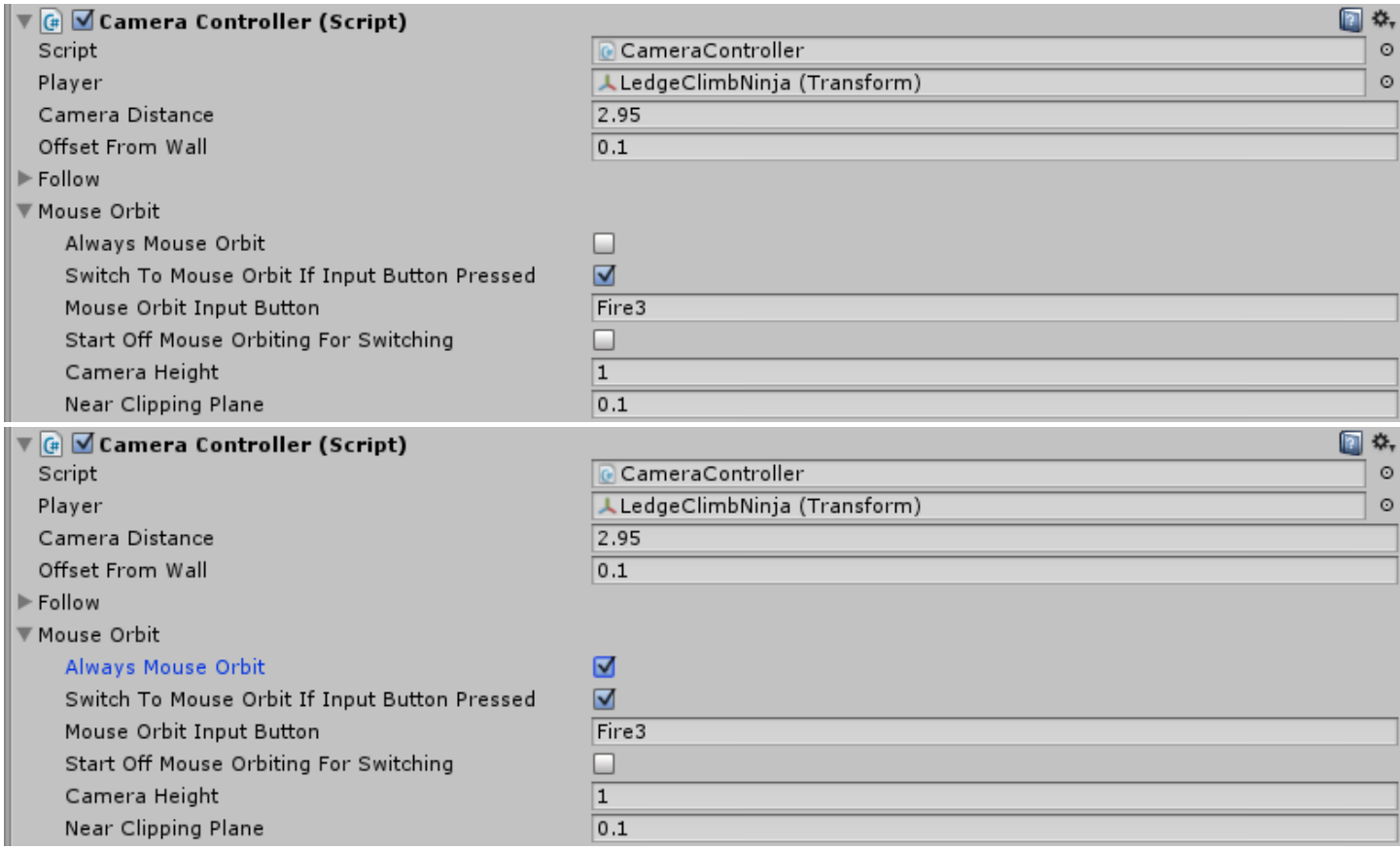


- **Third person mode with a mouse orbiting camera:**

1. Select your camera in the “Hierarchy” tab.

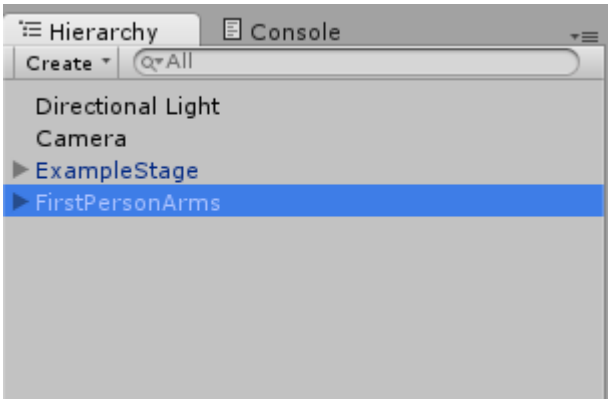


2. In the “Camera Controller” script, check the box that says “Always Mouse Orbit” (or, if you would like to switch to mouse orbiting by pressing a button instead of always mouse orbiting, check “Switch To Mouse Orbit If Input Button Pressed,” and leave “Always Mouse Orbit” unchecked).

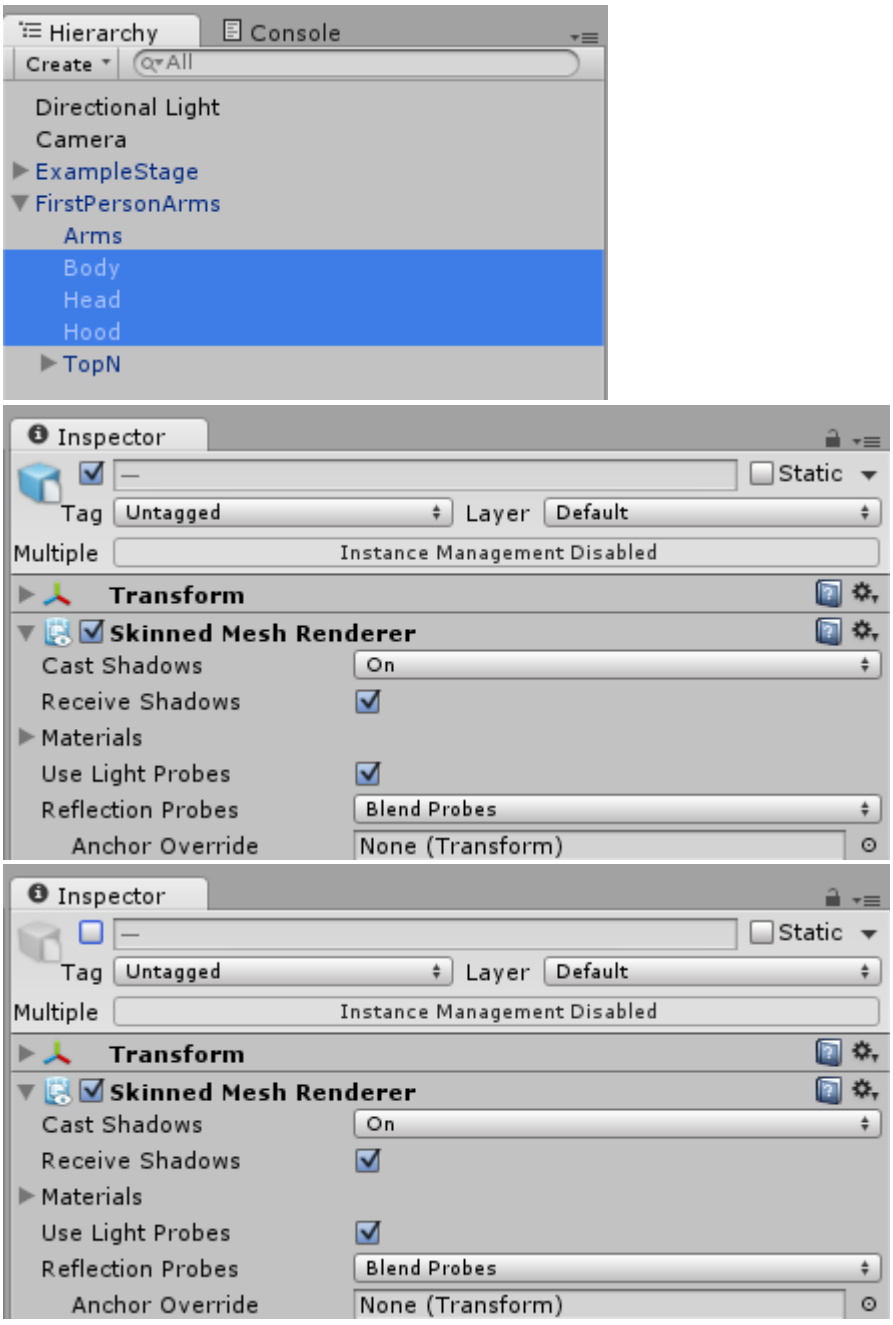


First Person Mode

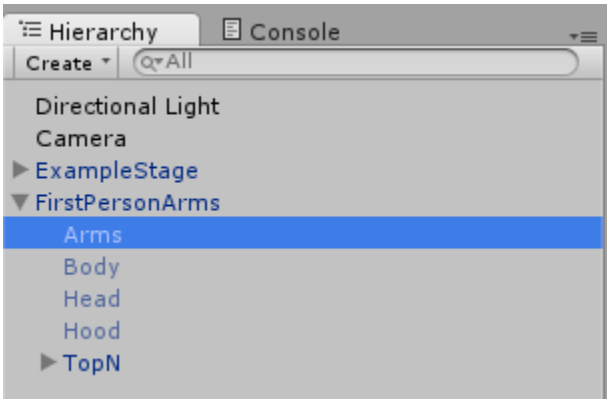
1. Select your player in the “Hierarchy” tab (since we are setting up the First Person game mode, we will be using “FirstPersonArms” as the player).

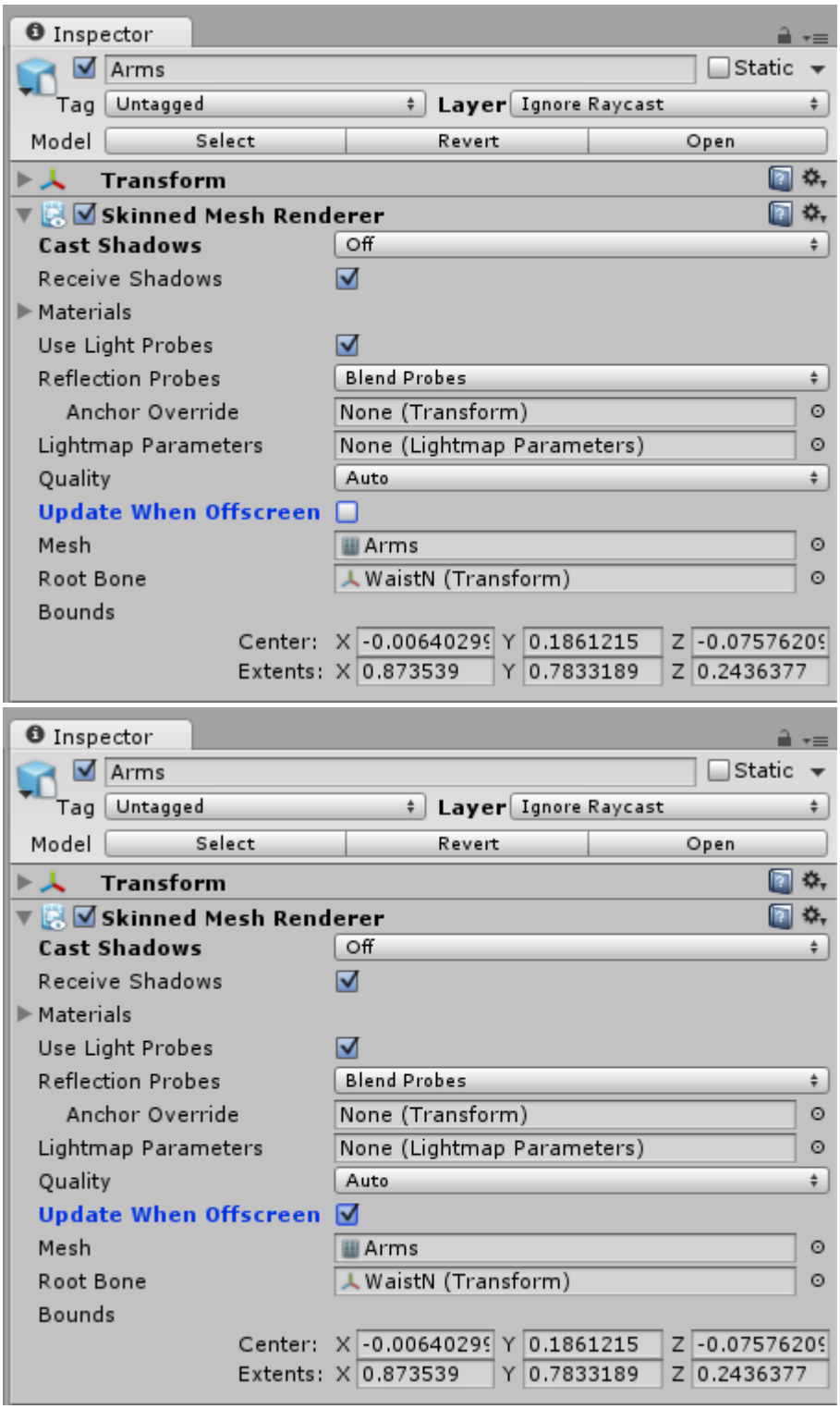


2. Disable/hide whichever parts of the player model you do not want to be seen while in First Person Mode.

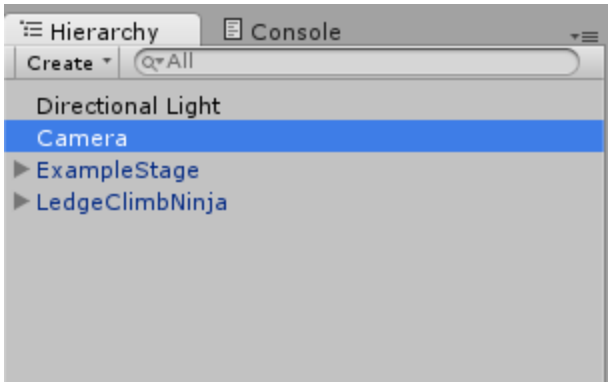


3. Select whichever parts of the player model you do want to be seen while in First Person Mode, and enable the "Update When Offscreen" option inside of their Mesh Renderer component.

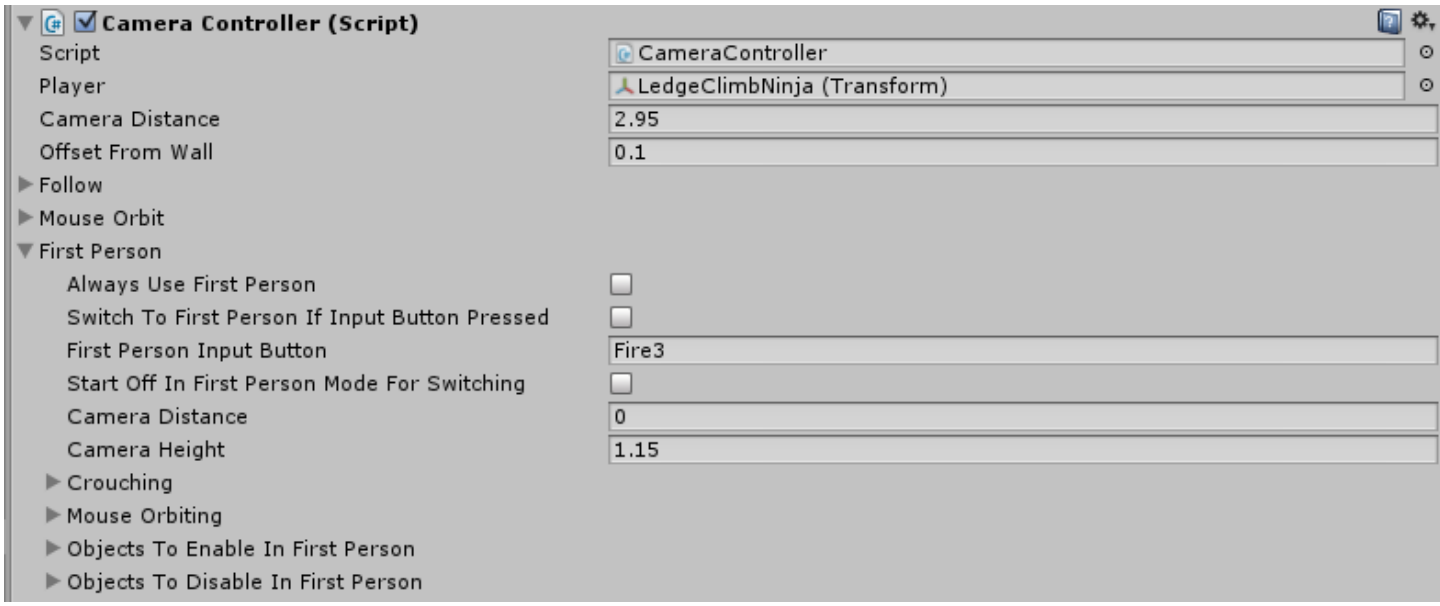


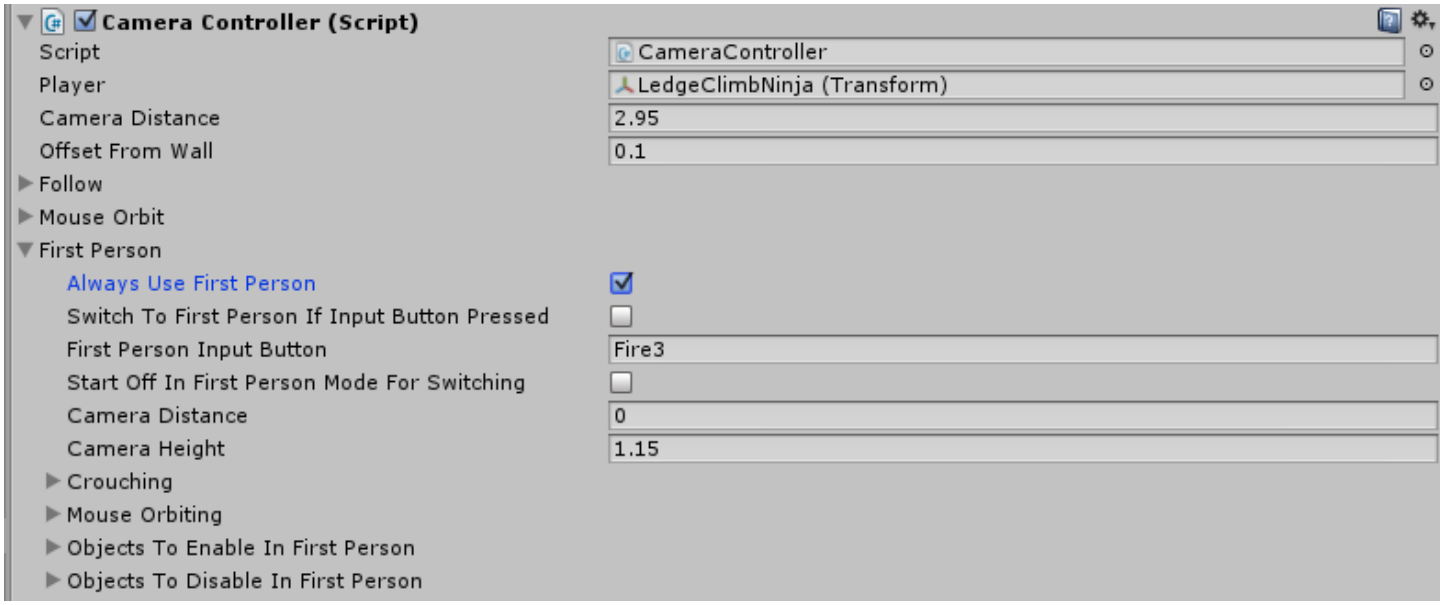


4. Select your camera in the “Hierarchy” tab.



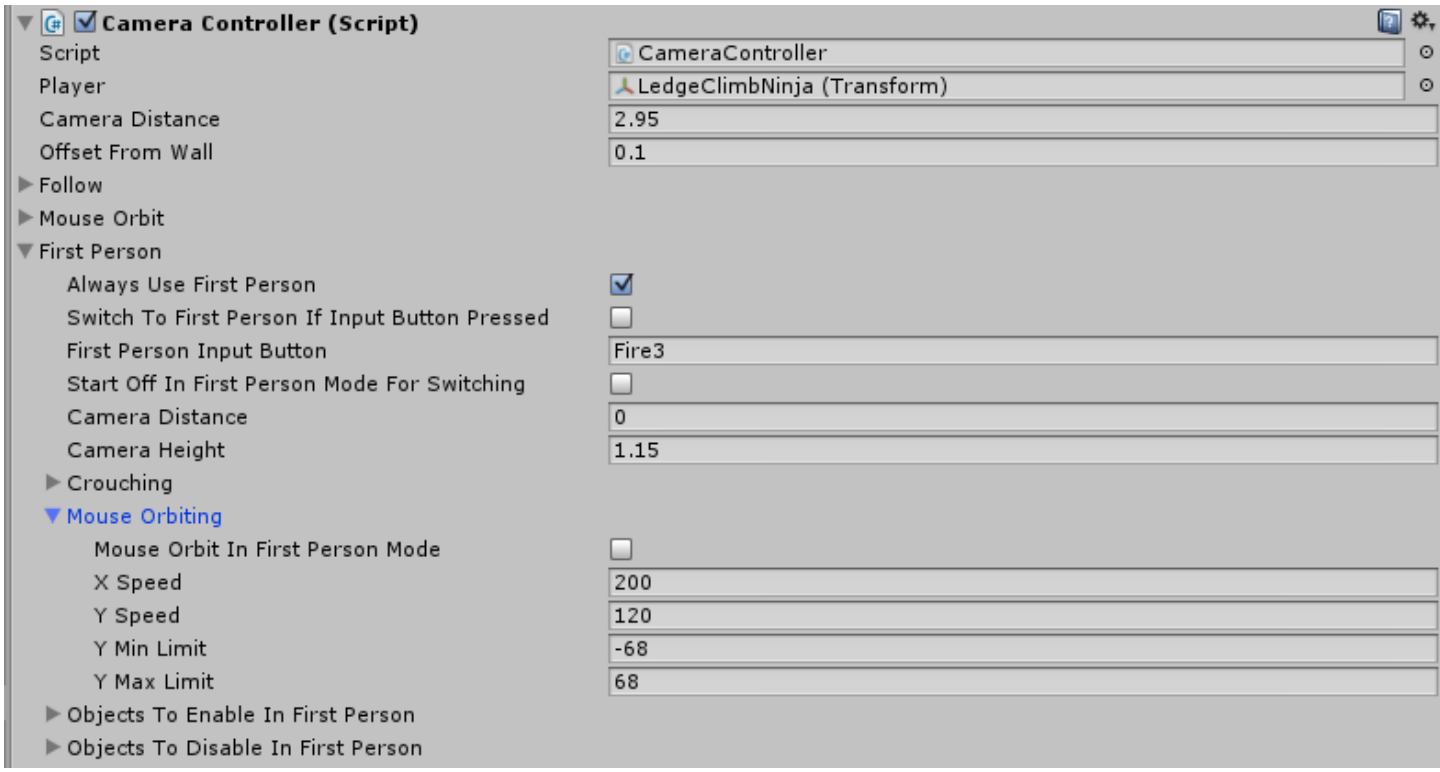
5. In the “Camera Controller” script, check the box that says “Always Use First Person” (or, if you would like to switch to mouse orbiting by pressing a button instead of always mouse orbiting, check “Switch To First Person If Input Button Pressed,” and leave “Always Mouse Orbit” unchecked).

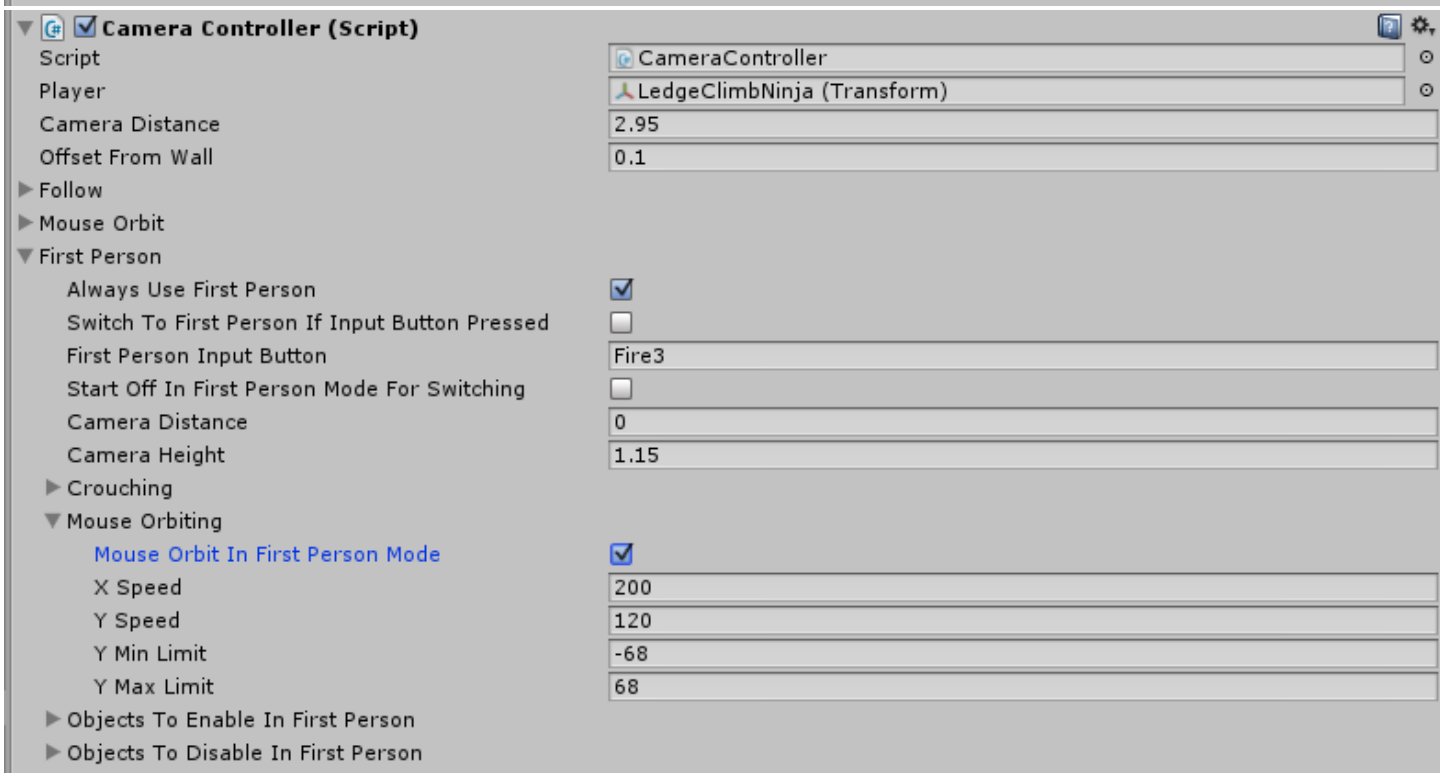




6. (OPTIONAL)

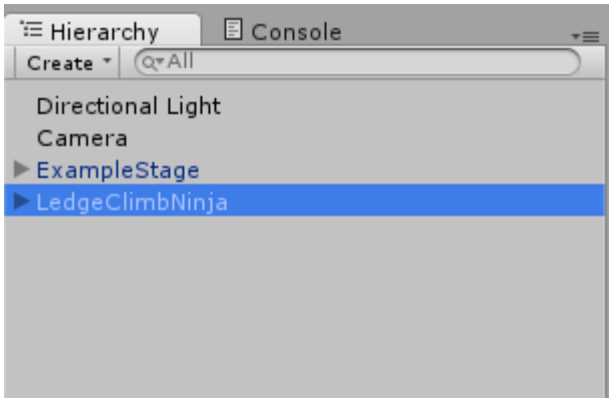
If you would like camera to mouse orbit in first person mode, expand the “Mouse Orbiting” section of the “First Person” category, then check “Mouse Orbit In First Person Mode.”



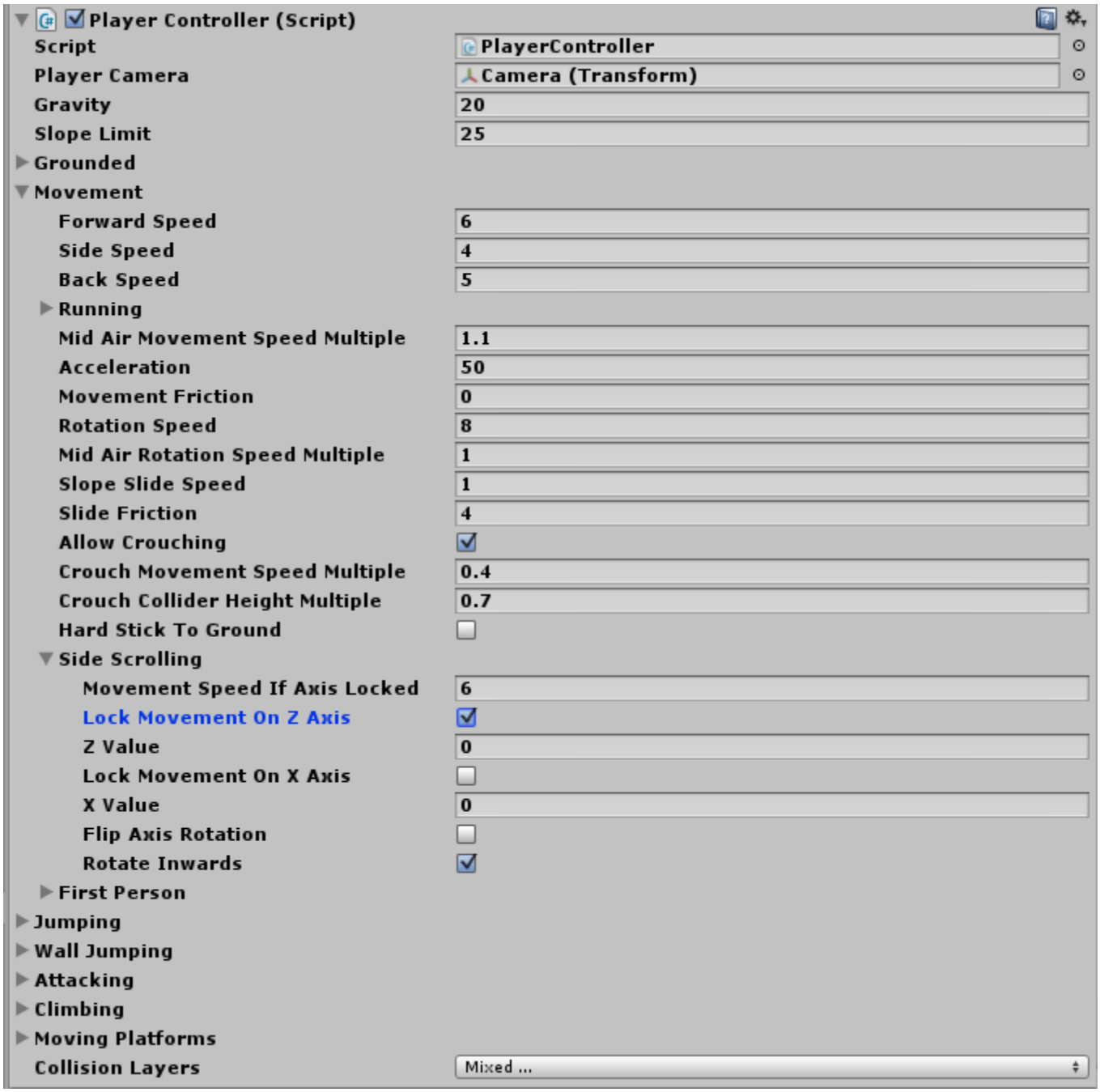


Side-Scrolling / 2.5D Mode

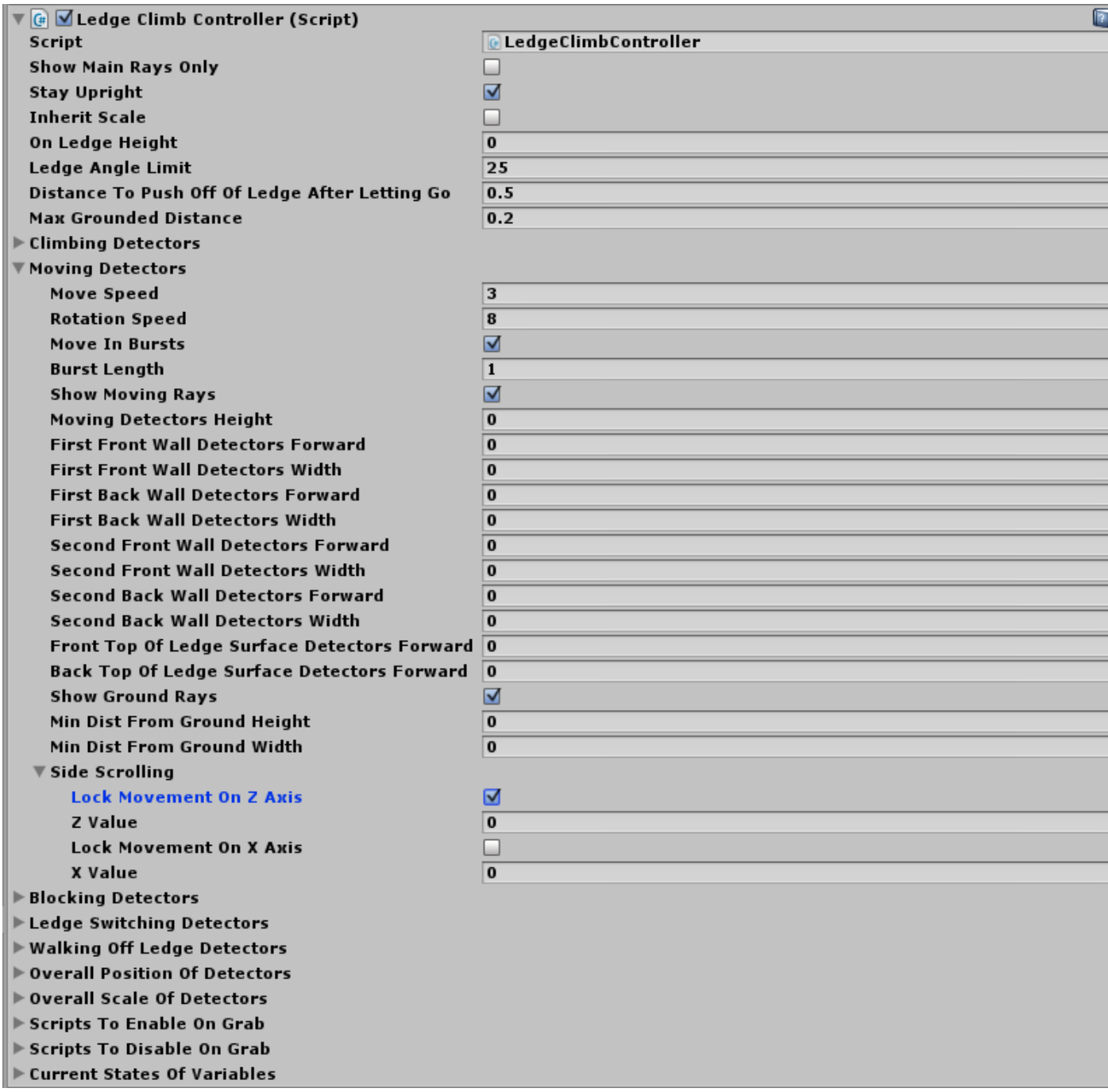
1. Select your player in the “Hierarchy” tab.



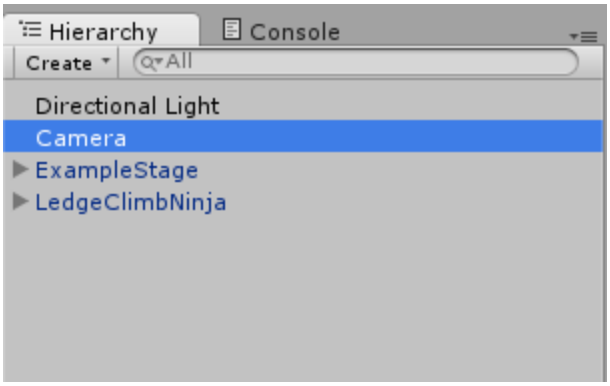
2. In the “Player Controller” script, check the box that says “Lock Movement On Z Axis” (or if you would like to lock the x-axis and only move on the z-axis instead, check the box that says “Lock Movement On X Axis” instead).



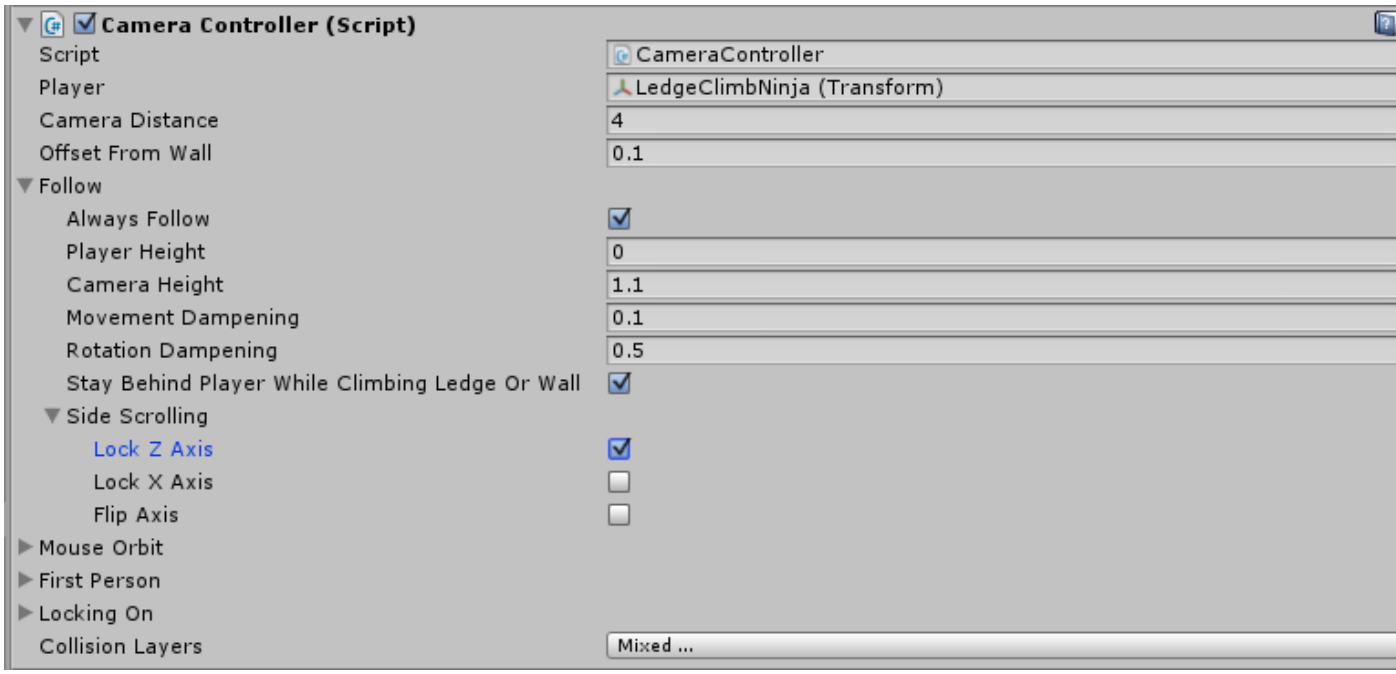
3. In the “Ledge Climb Controller” script, check the box that says “Lock Movement On Z Axis” (or if you would like to lock the x-axis and only move on the z-axis instead, check the box that says “Lock Movement On X Axis” instead).



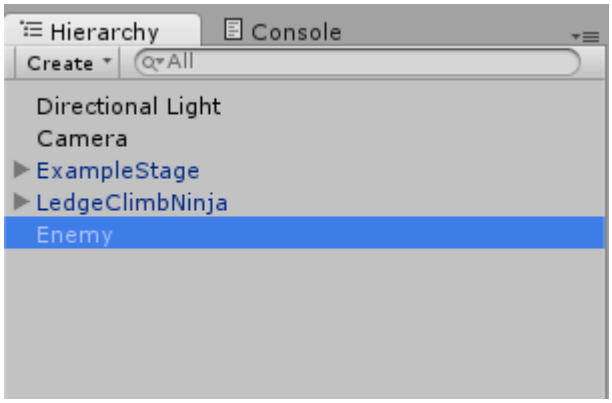
4. Select your camera in the “Hierarchy” tab.



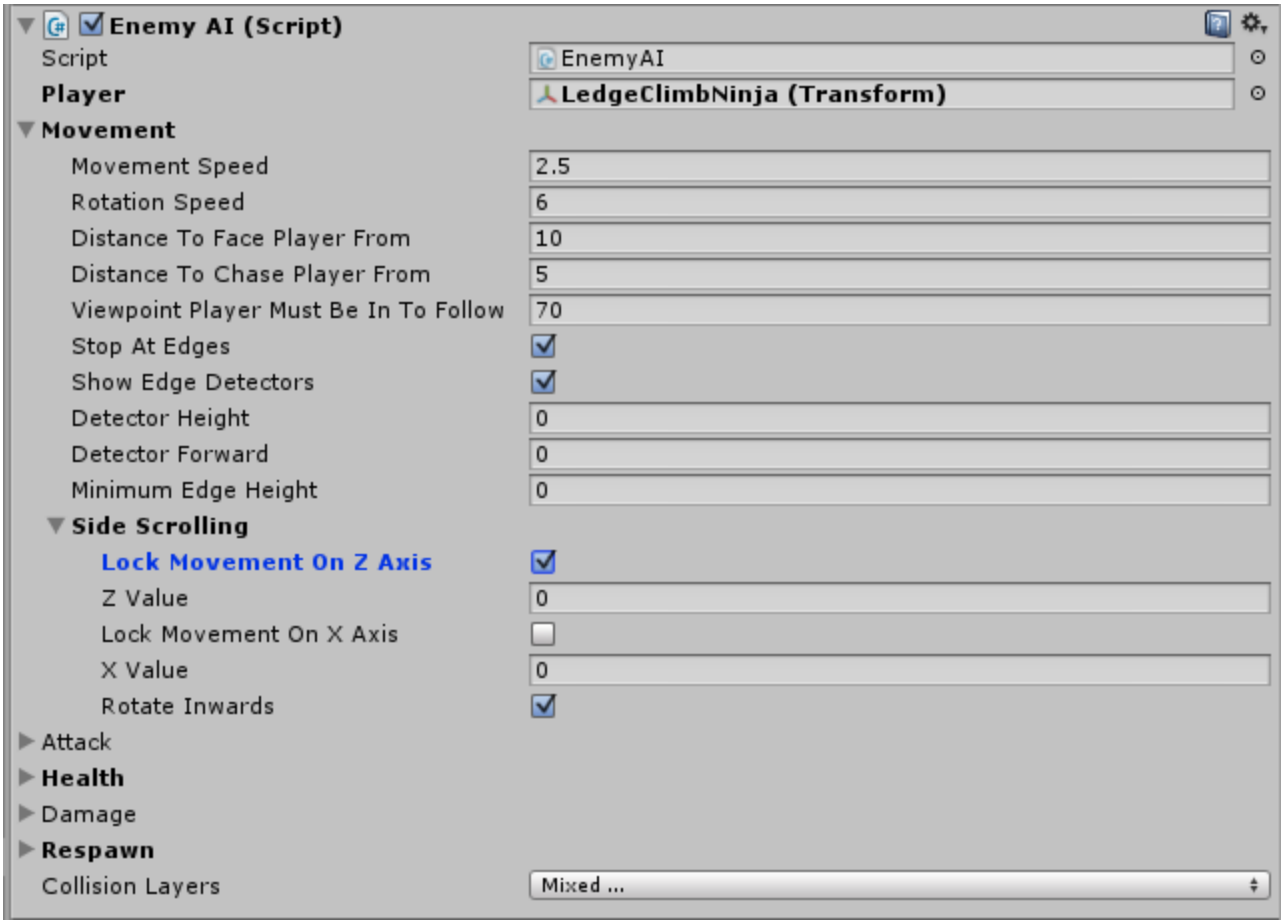
5. In the “Camera Controller” script, check the box that says “Lock Z Axis” (or if you would like to lock the x-axis and only move on the z-axis instead, check the box that says “Lock X Axis” instead).



6. Select your enemy in the “Hierarchy” tab.

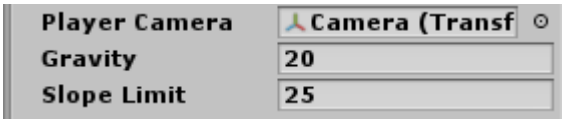


7. In the “Enemy AI” script, check the box that says “Lock Movement On Z Axis” (or if you would like to lock the x-axis and only move on the z-axis instead, check the box that says “Lock Movement On X Axis” instead).



All in One Game Kit - ELC Character System Variables and Features

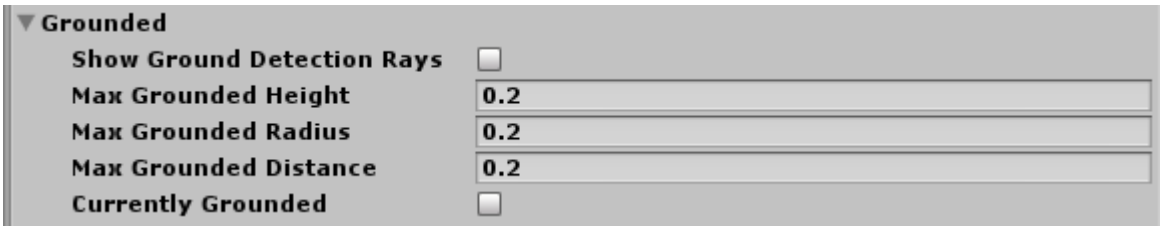
PlayerController.cs



Player Camera – the camera set to follow the player

Gravity – the amount of downward force, or "gravity," that is constantly being applied to the player

Slope Limit – the maximum angle of a slope you can stand on without sliding down



Grounded – detectors that determine whether the player is grounded or not

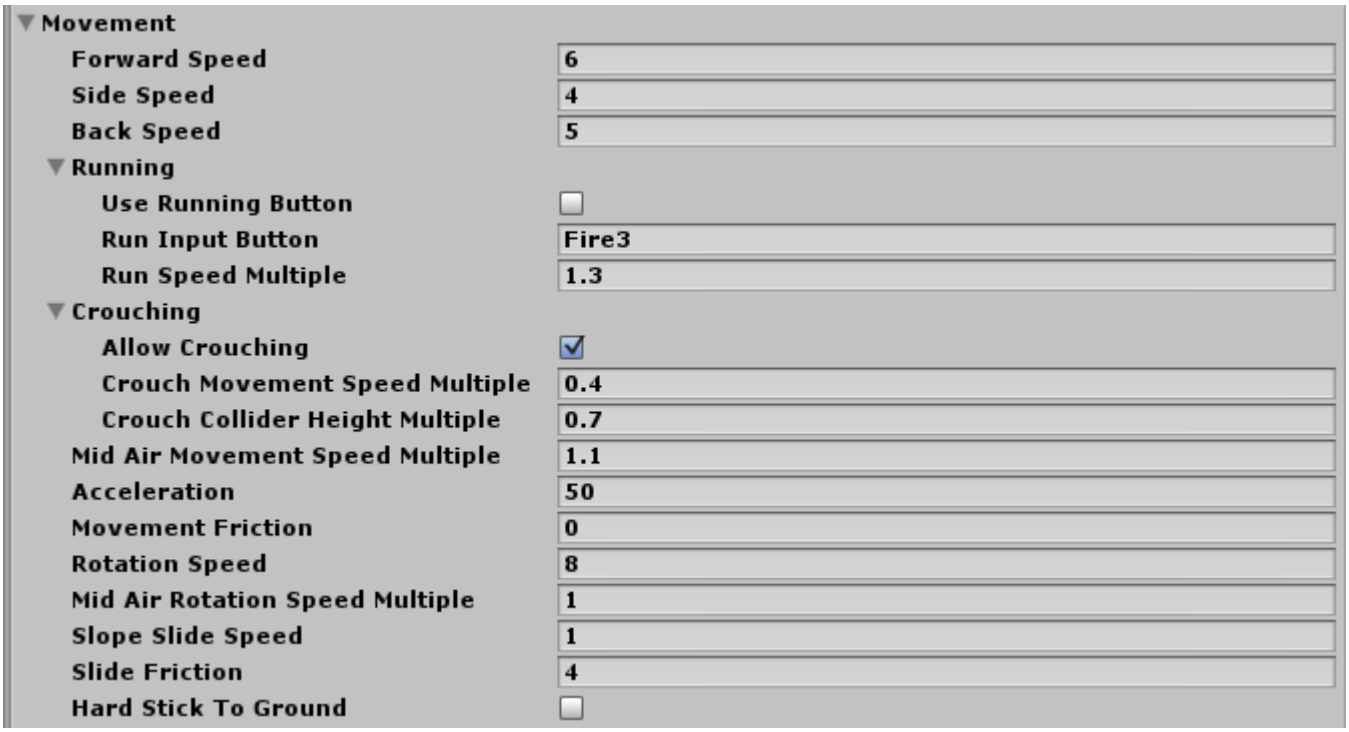
Show Ground Detectors – shows the rays that detect whether the player is grounded or not

Max Grounded Height – the maximum height of the ground the ground detectors can hit to be considered grounded

Max Grounded Radius – the maximum radius of the area ground detectors can hit to be considered grounded

Max Grounded Distance – the maximum distance you can be from the ground to be considered grounded

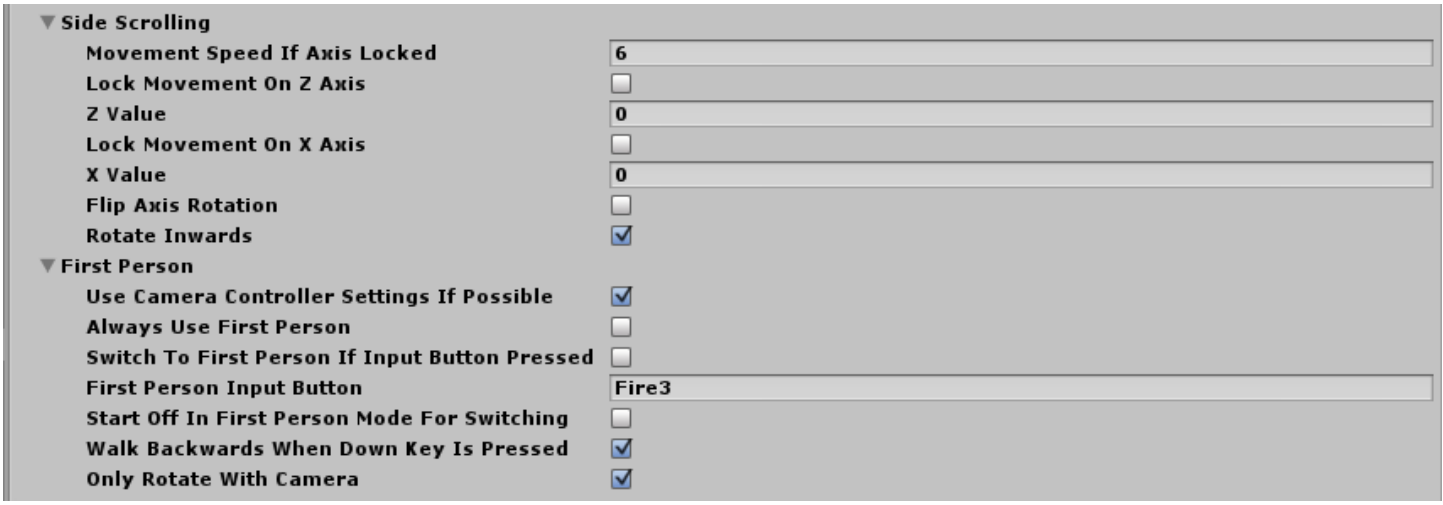
Currently Grounded – determines if the player is currently grounded/on the ground



Movement – variables that affect the player's movement

- Forward Speed – player's speed when running forward
- Side Speed – player's speed when running sideways
- Back Speed – player's speed when running backwards
- Running – variables that determine whether or not the player uses a running button to run
  - Use Running Button – allows the player to multiply his movement speed when the run button is pressed
  - Run Input Button – the button (found in "Edit > Project Settings > Input") that is used to make the player run
  - Run Speed Multiple – player's movement speed while the player is running/the run button is held down (multiplied by move speed
- Crouching – variables that determine whether or not the player can crouch
  - Allow Crouching – determines whether or not the player is allowed to crouch
  - Crouch Movement Speed Multiple – player's movement speed while crouching (multiplied by move speed)
  - Crouch Collider Height Multiple – what to multiply the player's collider height while crouching by
- Mid Air Movement Speed Multiple – player's movement speed in mid-air (multiplied by the player's current movement speed)
- Acceleration – how fast the player will reach their maximum speed
- Movement Friction – the amount of friction applied to the player's movement
- Rotation Speed – player's rotation speed
- Mid Air Rotation Speed Multiple – player's rotation speed in mid-air (multiplied by Rotation Speed)
- Slope Slide Speed – how quickly you slide down slopes
- Slide Friction – the amount of friction applied to the player from sliding down a slope

Hard Stick to Ground – by using a raycast, this option sets the position of the player to the position of the ground under him

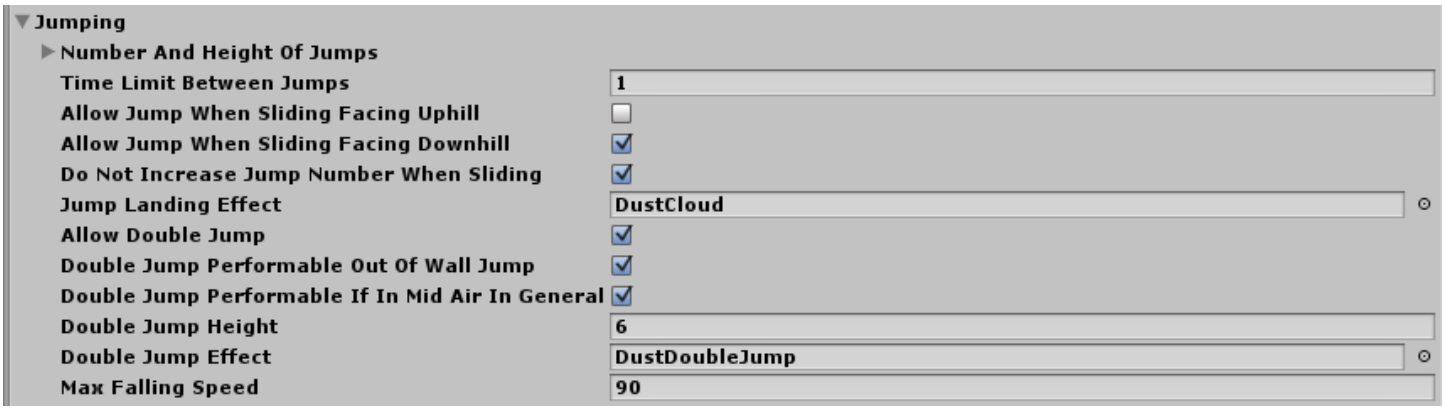


Side Scrolling – variables that determine whether or not the player uses 2.5D side-scrolling

- Movement Speed If Axis Is Locked – the move speed of the player if one of the axis are locked
- Lock Movement On Z Axis – locks the movement of the player on the z-axis
- Z Value – the permanent z-value of the player if his movement on the z-axis is locked
- Lock Movement On X Axis – locks the movement of the player on the x-axis
- X Value – the permanent x-value of the player if his movement on the x-axis is locked
- Flip Axis Rotation – flips the player's rotation on the non-locked axis (it adds 180 degrees to the player's rotation)
- Rotate Inwards – when the player rotates from side to side, he rotates inward (so that you see his front side while he is rotating)

First Person – variables that determine whether or not the player uses first person mode

- Use Camera Controller Settings If Possible – if the player camera has the script: "CameraController.cs" attached to it, the player will use the same first person settings as the camera
- Always Use First Person – allows the player to always stay in first person mode
- Switch To First Person If Input Button Pressed – switches to first person mode and back when the "firstPersonInputButton" is pressed
- First Person Input Button – the button (found in "Edit > Project Settings > Input") that is used to enter first person mode
- Start Off In First Person Mode For Switching – if the player is allowed to switch to first person mode, start off in first person mode instead of having to switch to it first
- Walk Backwards When Down Key Is Pressed – allows the player to walk backwards (instead of turn around) when the down key is pressed
- Only Rotate With Camera – does not allow the arrow keys to change the player's direction; only allows the player to rotate to the direction that the camera is facing



Jumping – variables that affect the player's jumps

Number And Height Of Jumps – the number of jumps the player can do and the height of the jumps (the elements)

Time Limit Between Jumps – the amount of time you have between each jump to continue the jump combo

Allow Jump When Sliding Facing Uphill – determines whether or not you are allowed to jump when you are facing uphill and sliding down a slope

Allow Jump When Sliding Facing Downhill – determines whether or not you are allowed to jump when you are facing downhill and sliding down a slope

Do Not Increase Jump Number When Sliding – only allows the player to perform their first jump when sliding down a slope

Jump Landing Effect – optional dust effect to appear after landing jump

Allow Double Jump – determines whether or not you are allowed to double jump

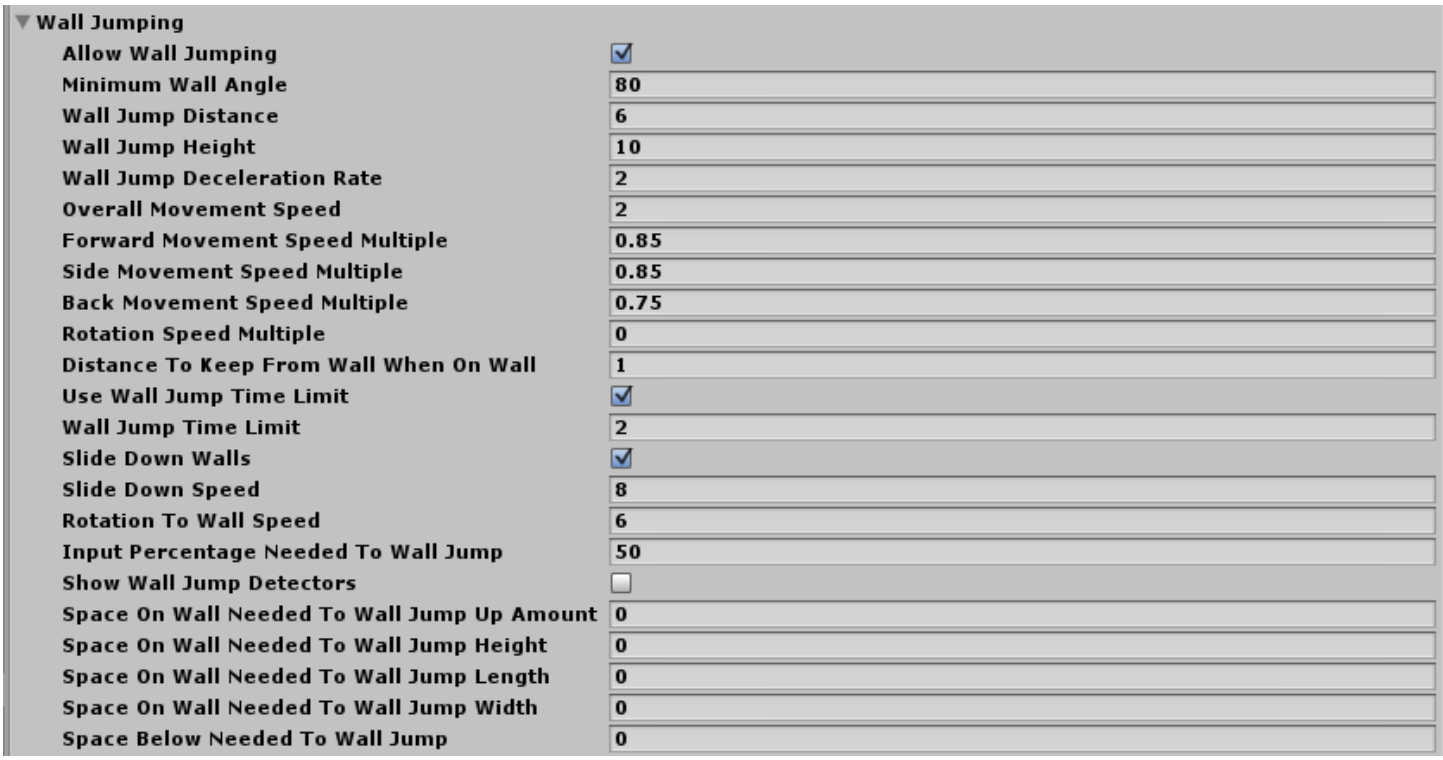
Double Jump Performable Out Of Wall Jump – (if "Allow Double Jump" is true) determines whether or not the player can perform their double jump if they are in mid-air as a result of wall jumping

Double Jump Performable If In Mid Air In General – (if "Allow Double Jump" is true) determines whether or not the player can perform their double jump simply because they are in mid-air (instead of having to be in mid-air as a result of jumping)

Double Jump Height – height of double jump

Double Jump Effect – optional effect to appear when performing a double jump

Max Falling Speed – the maximum speed you can fall



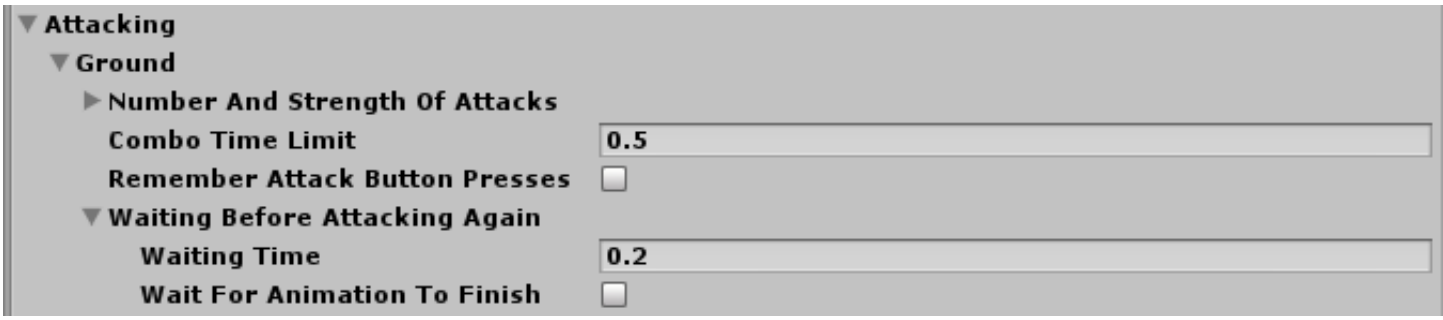
Wall Jumping – variables that affect the player's wall jumps

- Allow Wall Jumping – determines whether or not the player is allowed to wall jump
- Minimum Wall Angle – the minimum angle a wall can be to wall jump off of it
- Wall Jump Distance – distance of wall jump
- Wall Jump Height – height of wall jump
- Wall Jump Deceleration Rate – how quickly the momentum from the wall jump stops
- Overall Movement Speed – player's movement speed in mid-air
- Forward Movement Speed Multiple – player's speed when moving forward in mid air (multiplied by Overall Movement Speed)
- Side Movement Speed Multiple – player's speed when moving sideways in mid air (multiplied by Overall Movement Speed)
- Back Movement Speed Multiple – player's speed when moving backwards in mid air (multiplied by Overall Movement Speed)
- Rotation Speed Multiple – player's rotation speed in mid-air (multiplied by Rotation Speed)
- Distance To Keep From Wall When On Wall – the distance the player keeps from the wall he is currently stuck to
- Use Wall Jump Time Limit – allows the use of a time limit to wall jump when on walls
- Wall Jump Time Limit – the amount of time you can stay on a wall before falling
- Slide Down Walls – allows player to slide down if on a wall
- Slide Down Speed – the speed at which the player slides down walls
- Rotation To Wall Speed – how quickly the player rotates onto a wall for a wall jump
- Input Percentage Needed To Wall Jump – the amount of input needed to be applied to the joystick or key in order to stick to a wall for a wall jump
- Show Wall Jump Detectors – determines whether to show or hide the detectors that allow wall jumping
- Space On Wall Needed To Wall Jump Up Amount – moves the rays that detect the amount of open space on a wall up and down
- Space On Wall Needed To Wall Jump Height – changes the height of the rays that detect the amount of open space on a wall

Space On Wall Needed To Wall Jump Length – changes the length of the rays that detect the amount of open space on a wall

Space On Wall Needed To Wall Jump Width – changes the width of the rays that detect the amount of open space on a wall

Space Below Needed To Wall Jump – changes the minimum distance from the ground you must be in order to wall jump



Attacking – variables that control the player's attacks

Ground – variables for the player's ground attacks

Number And Strength Of Attacks – the number of attacks the player can perform and the strength of each one (the elements)

Combo Time Limit – the amount of time you have to wait between each attack to continue the ground attack combo

Remember Attack Button Presses – allows the player to press the attack button multiple times, then continue the combo based on the number of times the button was pressed (before the combo resets)

Waiting Before Attacking Again – variables that determine how long the player must wait before attacking again

Waiting Time – the amount of time you have to wait (after each attack) before you can continue the attack combo

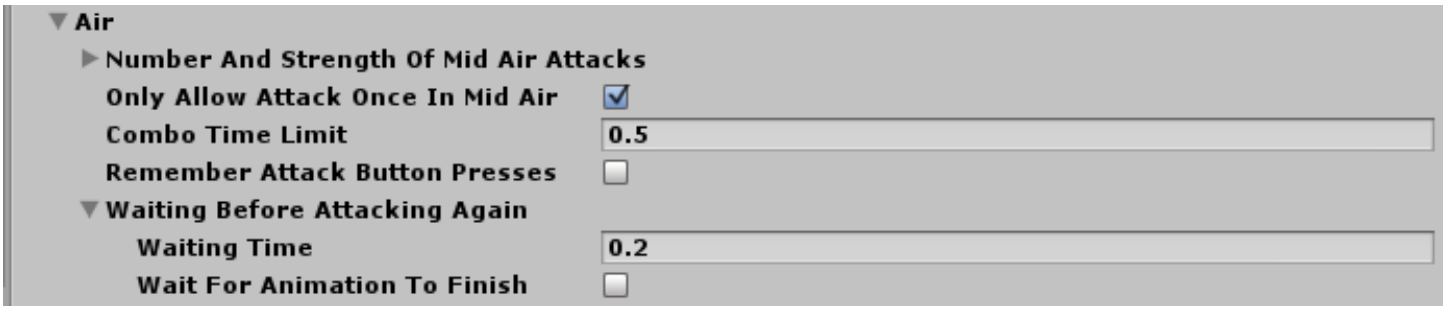
Wait For Animation To Finish – waits for the current attack animation to finish before continuing the combo

Crouch – variables for the player's crouch attacks

Allow Crouch Attack – allows the player to attack while crouching

Crouch Attack Strength – the strength of the player's crouch attack

Time Limit Between Crouch Attacks – the amount of time you have to wait between each attack to crouch attack again



Air – variables for the player's mid air attacks

Number And Strength Of Mid Air Attacks – the number of maximum attacks the player can perform and the strength of each one (the elements)

Only Allow Mid Air Attack Once In Mid Air – only allows the player to use his mid-air attack once while in the air

Combo Time Limit – the amount of time you have to wait between each attack to continue the mid air attack combo

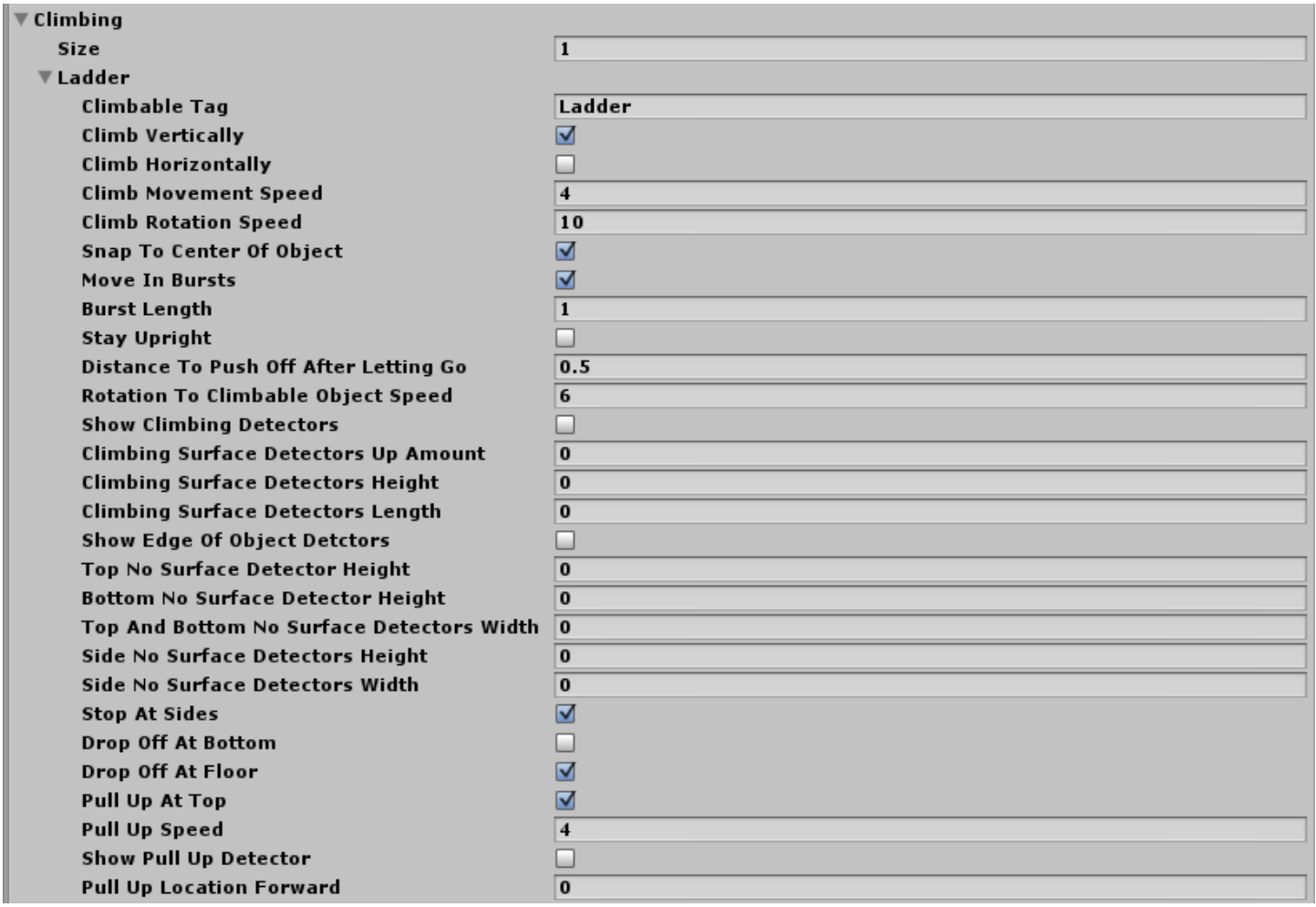
Remember Attack Button Presses – allows the player to press the attack button multiple times, then continue the combo based on the number of times the button was pressed (before the combo resets)

Waiting Before Attacking Again – variables that determine how long the player must wait before attacking again

Waiting Time – the amount of time you have to wait (after each attack) before you can continue the attack combo

Wait For Animation To Finish – waits for the current attack animation to finish before continuing the combo

Attack Input Button – the button (found in "Edit > Project Settings > Input") that is used to attack



Climbing – variables that control the player's ladder and wall climbing

Climbable Tag – the tag of a climbable object

Climb Vertically – determines whether or not the player is allowed to climb vertically

Climb Horizontally – determines whether or not the player is allowed to climb horizontally

Climb Movement Speed – how quickly the player climbs on walls

Climb Rotation Speed – how quickly the player rotates on walls

Snap To Center Of Object – snaps the player to the middle (along the x and z-axis) of the climbable object (most useful for ladders)

Move In Bursts – move in bursts (while on a climbable object)

Burst Length – the amount of time a movement burst lasts

Stay Upright – determines whether or not the player can rotate up and down

Distance To Push Off After Letting Go – the distance the player pushes off of a ladder/wall after letting go

Rotation To Climbable Object Speed – how quickly the player rotates onto a wall to climb

Show Climbing Detectors – determines whether to show or hide the detectors that allow climbing

Climbing Surface Detectors Up Amount – moves the rays that detect the surface of a wall up and down

Climbing Surface Detectors Height – changes the height of the rays that detect the surface of a wall

Climbing Surface Detectors Length – changes the length of the rays that detect the surface of a wall

Show Edge Of Object Detectors – determines whether or not to show the detectors that determine where the top and bottom of a climbable object is

Top No Surface Detector Height – the height of the detector that determines if there is no surface detected at the top of the climbable object, so that the player can pull up or stop before climbing any higher

Bottom No Surface Detector Height – the height of the detector that determines if there is no surface detected at the bottom of the climbable object, so that the player can drop off or stop before climbing any lower

Top And Bottom No Surface Detectors Width – the width of the detectors that determines if there is no surface detected at the top and bottom of the climbable object

Side No Surface Detectors Height – the height of the detectors that determines if there is no surface detected at the sides of the climbable object

Side No Surface Detectors Width – the width of the detectors that determines if there is no surface detected at the sides of the climbable object

Stop At Sides – keeps player from climbing any further sideways once he has reached the side

Drop Off At Bottom – allows player to drop off of a climbable object once he has reached the bottom

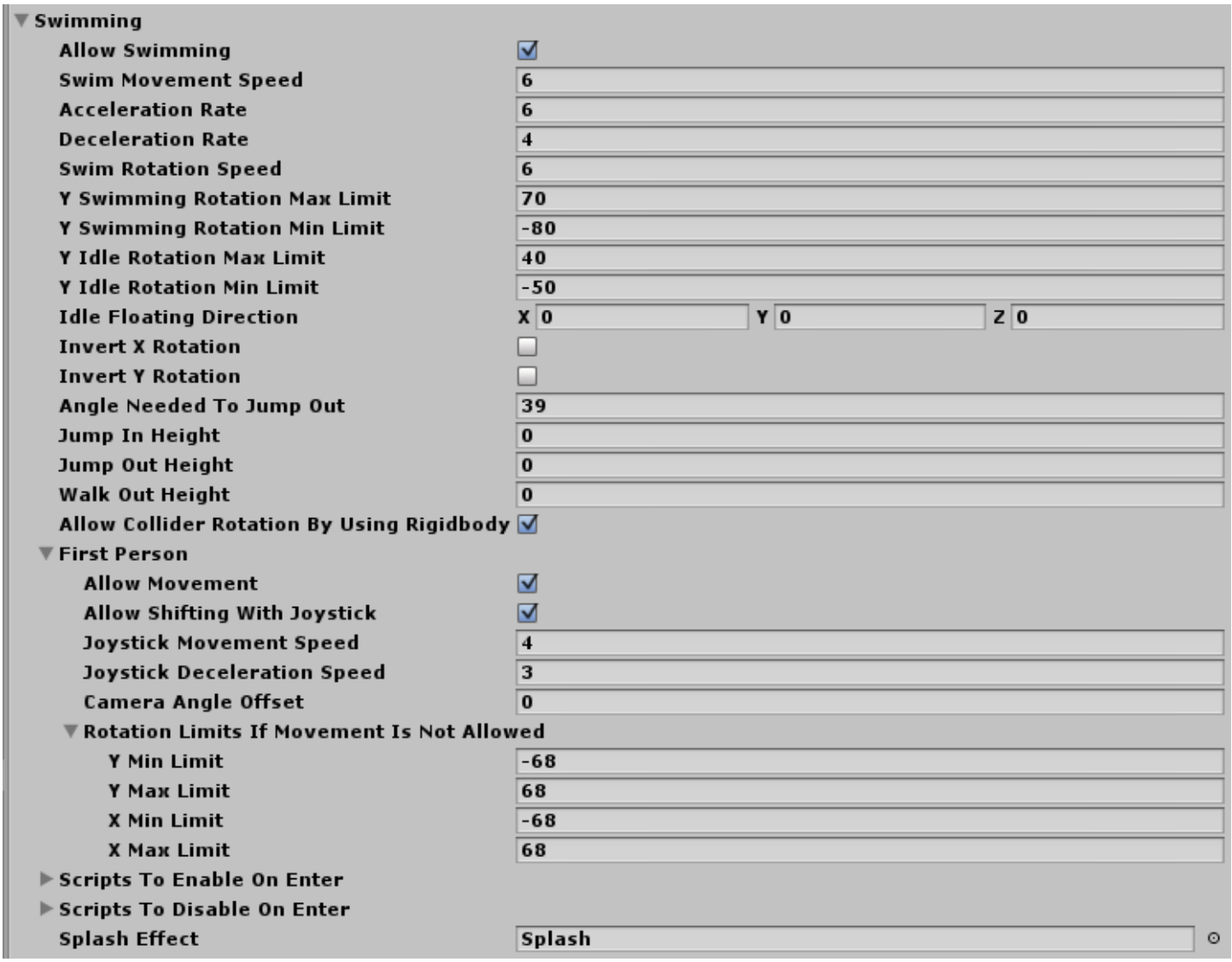
Drop Off At Floor – allows player to drop off of a climbable object once he has reached the floor

Pull Up At Top – allows player to pull up and over a climbable object once he has reached the top

Pull Up Speed – the speed the player pulls up and over ledges once he has reached the top of a climbable object

Show Pull Up Detector – determines whether or not to show the detector that determines where the player pulls up to

Pull Up Location Forward – the forward distance of the detector that determines where the player pulls up to



Swimming – variables that control the player's swimming

- Allow Swimming – determines whether or not the player is allowed to swim
- Swim Movement Speed – the player's movement speed while swimming
- Acceleration Rate – how fast the player will reach their maximum swimming speed
- Deceleration Rate – how fast the player will reach their minimum swimming speed (0 speed)
- Swim Rotation Speed – the player's rotation speed while swimming
- Y Swimming Rotation Max Limit – the maximum y value the player can rotate to while swimming vertically
- Y Swimming Rotation Min Limit – the minimum y value the player can rotate to while swimming vertically
- Y Idle Rotation Max Limit – the maximum y value the player can rotate to while not swimming vertically
- Y Idle Rotation Min Limit – the minimum y value the player can rotate to while not swimming vertically
- Idle Floating Direction – the direction the player floats in while he is not swimming
- Invert X Rotation – inverts the player's left and right swimming rotation
- Invert Y Rotation – inverts the player's up and down swimming rotation
- Angle Needed To Jump Out – the minimum angle the player must be rotated to jump out of the water
- Jump In Height – the height the player must sink before entering the water

Jump Out Height – the height the player must swim before jumping out of the water (the player cannot swim any higher once he reaches this point)

Walk Out Height – the height the player must swim before walking out of the water, and on to the surface (note: the player cannot swim any higher than the jump out height)

Allow Collider Rotation By Using Rigidbody – allows the collider of the player to rotate while swimming (by adding a Rigidbody+CapsuleCollider, and disabling the CharacterController temporarily, since CharacterController colliders can only stay upright)

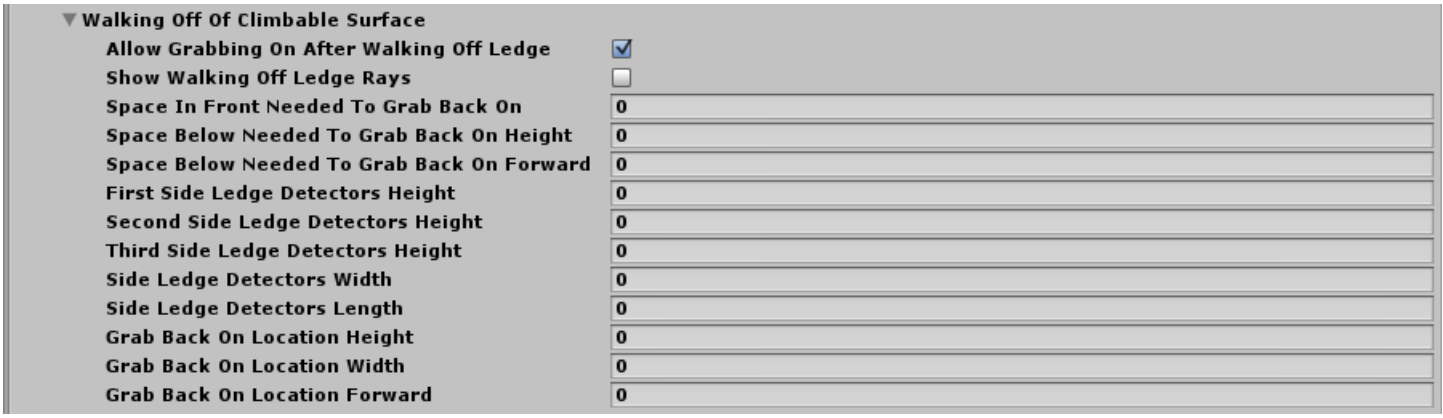
First Person – variables that determine how the player swims in first person mode

- Allow Movement – allows the player to move while in first person mode
- Allow Shifting With Joystick – allows the player to rotate with the camera, and move left, right, up, or down with the joystick, while mouse orbiting in first person mode
- Joystick Movement Speed – the speed the player moves left, right, up, or down (with the joystick) while mouse orbiting in first person mode
- Joystick Deceleration Speed – the speed the player stops moving left, right, up, or down (with the joystick) while mouse orbiting in first person mode
- Camera Angle Offset – the angle of the player's head upon entering first person mode
- Rotation Limits If Movement Is Not Allowed – variables that limit the camera's (and in turn, the player's) first person rotation while swimming
  - Y Min Limit – the minimum y value the camera can orbit to
  - Y Max Limit – the maximum y value the camera can orbit to
  - X Min Limit – the minimum x value the camera can orbit to
  - X Max Limit – the maximum x value the camera can orbit to

Scripts To Enable On Enter – scripts to enable when the player enters the water (scripts disable when the player exits the water)

Scripts To Disable On Enter – scripts to disable when the player enters the water (scripts enable when the player exits the water)

Splash Effect – optional splash effect to appear after landing in water



Walking Off Of Climbable Surface – variables that detect whether the player has walked off a ledge and can grab on to a ladder

- Allow Grabbing On After Walking Off Ledge –allows the player to grab on to a climbable surface under the ledge that he is walking off of
- Show Walking Off Ledge Rays – shows the rays that detect if the player is walking off of a ledge

Space In Front Needed To Grab Back On – the amount of space in front of the player needed to grab on to a climbable object under a ledge

Space Below Needed To Grab Back On Height – the height of the detectors that determine the amount of space below the player needed to grab on to a climbable object under a ledge

Space Below Needed To Grab Back On Forward – the forward distance of the detectors that determine the amount of space below the player needed to grab on to a climbable object under a ledge

First Side Ledge Detectors Height – the height of the first set of detectors that determine if there are ledges to the side of the player keeping him from grabbing on

Second Side Ledge Detectors Height – the height of the second set of detectors that determine if there are ledges to the side of the player keeping him from grabbing on

Third Side Ledge Detectors Height – the height of the third set of detectors that determine if there are ledges to the side of the player keeping him from grabbing on

Side Ledge Detectors Width – the width of the detectors that determine if there are ledges to the side of the player keeping him from grabbing on

Side Ledge Detectors Length – the length of the detectors that determine if there are ledges to the side of the player keeping him from grabbing on

Grab Back On Location Height – the height of the detectors that determine where the player will grab on to

Grab Back On Location Width – the height of the detectors that determine where the player will grab on to

Grab Back On Location Forward – the forward distance of the detectors that determine where the player will grab on to

Scripts To Enable On Grab – scripts to enable when the player grabs on to a wall (scripts disable when the player lets go of a wall)

Scripts To Disable On Grab – scripts to disable when the player grabs on to a wall (scripts enable when the player lets go of a wall)

Push Against Wall If Player Is Stuck – if the script considers the player stuck, the player pushes himself away from the wall until he is free

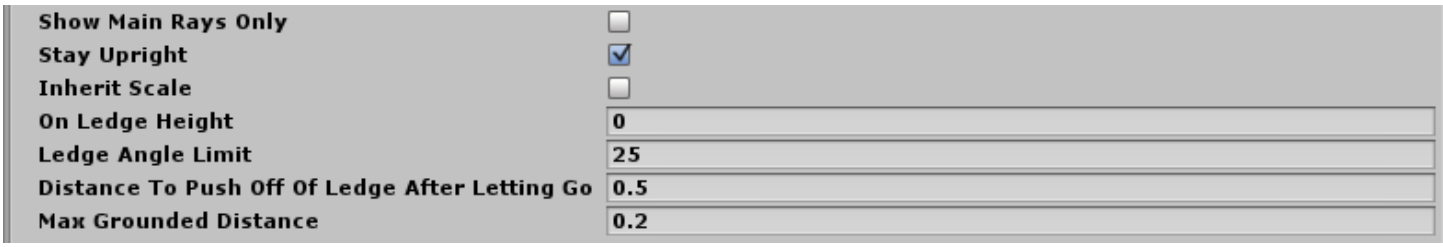
Moving Platforms – variables that determine whether the player moves with moving platforms or not

Allow Moving Platform Support – determines whether or not the player can move with moving platforms

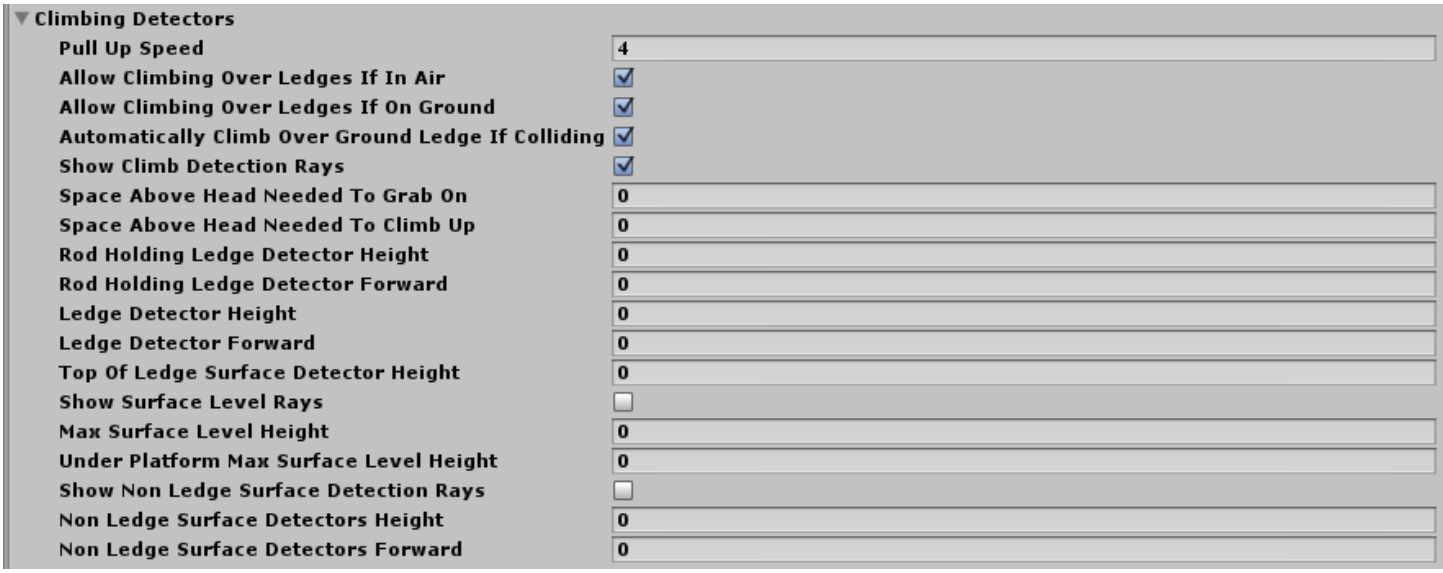
Moving Platform Tag – the tag of the moving platform objects

Collision Layers – the layers that the detectors (raycasts/linecasts) will collide with

LedgeClimbController.cs



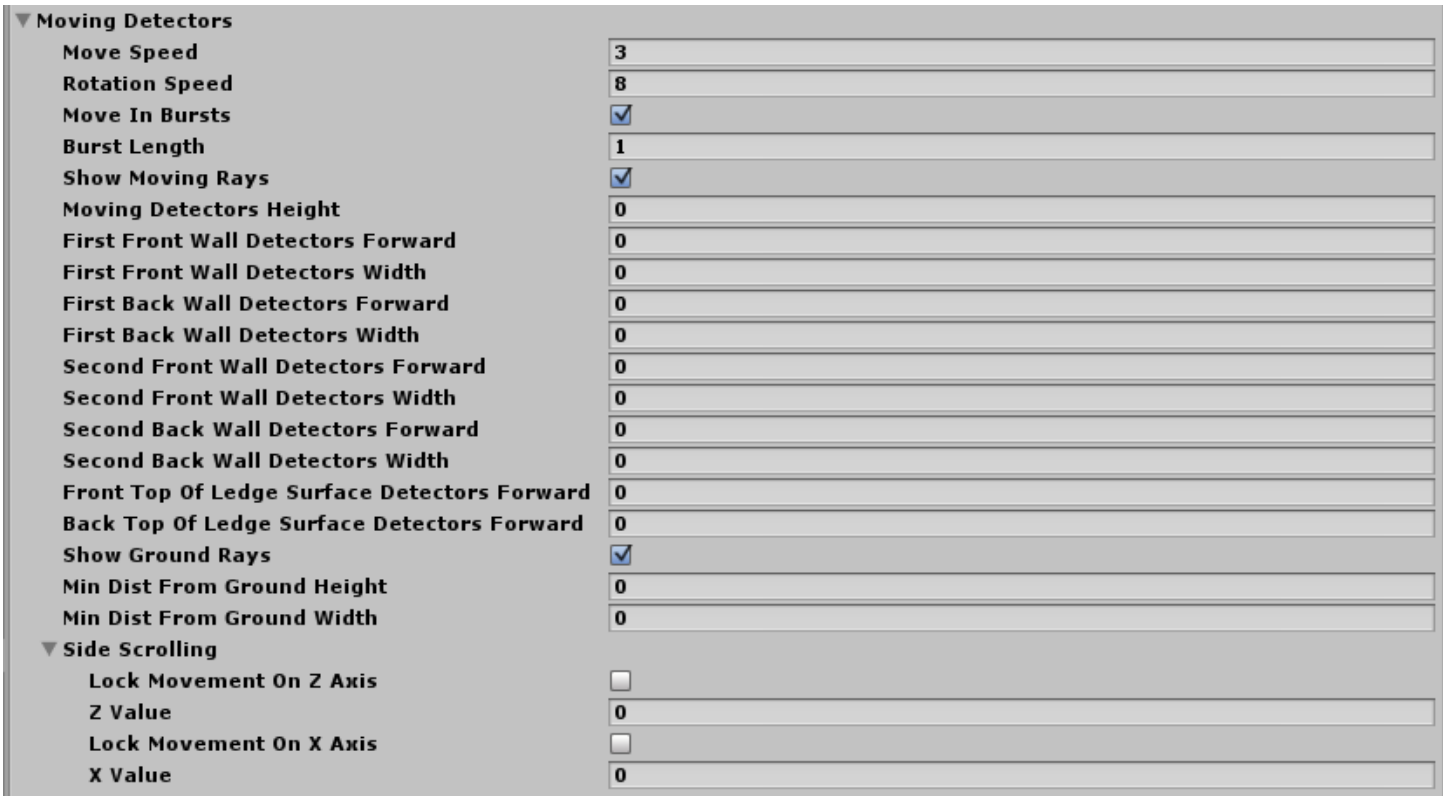
Show Main Rays Only – show only the main detector rays
Stay Upright – only rotate to the left and right, not up and down
Inherit Scale – set the scale of the rays to the scale of the script holder
On Ledge Height – determines the player's up distance/height while grabbed on to a ledge
Ledge Angle Limit – the maximum angle of a ledge you can grab on to
Distance To Push Off Of Ledge After Letting Go – the distance the player pushes off of a ledge after letting go
Max Grounded Distance – the maximum distance you can be from the ground to be considered grounded



Climbing Detectors – detectors that determine if climbing on to or grabbing on to a ledge is possible

- Pull Up Speed – the speed the player pulls up and over ledges
- Allow Climbing Over Ledges If In Air – allows player to climb over a ledge that he is currently grabbed on to in mid-air
- Allow Climbing Over Ledges If On Ground – allows player to climb over a ledge if he is on the ground
- Automatically Climb Over Ground Ledge If Colliding – (if "Allow Climbing Over Ledges If On Ground" is true) allows player to automatically climb over a ledge on the ground if he is colliding with and moving toward it (as opposed to having to stop and move again if colliding)
- Show Climb Detection Rays – shows the rays that detect whether a ledge can be climbed or not
- Space Above Head Needed To Grab On – the amount of space above the player's head needed to grab on to a ledge
- Space Above Head Needed To Climb Up – the amount of space above the player's head needed to climb up and over a ledge
- Rod Holding Ledge Detector Height – the height of the rod that holds the ledge detector
- Rod Holding Ledge Detector Forward – the forward distance of the rod that holds the ledge detector

Ledge Detector Height – the height of the ledge detector (the ray that detects ledges)
Ledge Detector Forward – the forward distance of the ledge detector (the ray that detects ledges)
Top Of Ledge Surface Detector Height – the height of the detector that determines where the top (surface) of a ledge is
Show Surface Level Rays – shows the rays that detect whether a ledge is low enough to be climbed or not
Max Surface Level Height – the maximum height (or level) of a surface that the player can grab on to
Under Platform Max Surface Level Height – the maximum height (or level) of a surface that the player can grab on to when that surface is under a platform/another collider
Show Non Ledge Surface Detection Rays – shows the rays that detect whether a surface is actually a ledge or not (instead of something else such as a slope/hillside)
Non Ledge Surface Detectors Height – the height of the detectors that determine whether a surface is actually a ledge or not (instead of something else such as a slope/hillside)
Non Ledge Surface Detectors Forward – the forward distance of the detectors that determine whether a surface is actually a ledge or not (instead of something else such as a slope/hillside)



Moving Detectors – detectors that determine where and how the player will move when on a ledge
Move Speed – the speed the player moves left and right (while grabbed on to a ledge)
Rotation Speed – the speed the player rotates (while grabbed on to a ledge)
Move In Bursts – move left and right in bursts (while grabbed on to a ledge)
Burst Length – the amount of time a movement burst lasts
Show Moving Rays – shows the rays that detect where and how the player will move when on a ledge
Moving Detectors Height – the height of the rays that detect where and how the player will move when on a ledge
First Front Wall Detectors Forward – the forward distance of the first set of detectors that detect walls in front of the player

First Front Wall Detectors Width – the width of the first set of detectors that detect walls in front of the player

First Back Wall Detectors Forward – the forward distance of the first set of detectors that detect walls behind the player

First Back Wall Detectors Width – the width of the first set of detectors that detect walls behind the player

Second Front Wall Detectors Forward – the forward distance of the second set of detectors that detect walls in front of the player

Second Front Wall Detectors Width – the width of the second set of detectors that detect walls in front of the player

Second Back Wall Detectors Forward – the forward distance of the second set of detectors that detect walls behind the player

Second Back Wall Detectors Width – the width of the second set of detectors that detect walls behind the player

Front Top Of Ledge Surface Detectors Forward – the forward distance of the detectors that determine where the top (surface) of a ledge in front of the player is

Back Top Of Ledge Surface Detectors Forward – the forward distance of the detectors that determine where the top (surface) of a ledge behind the player is

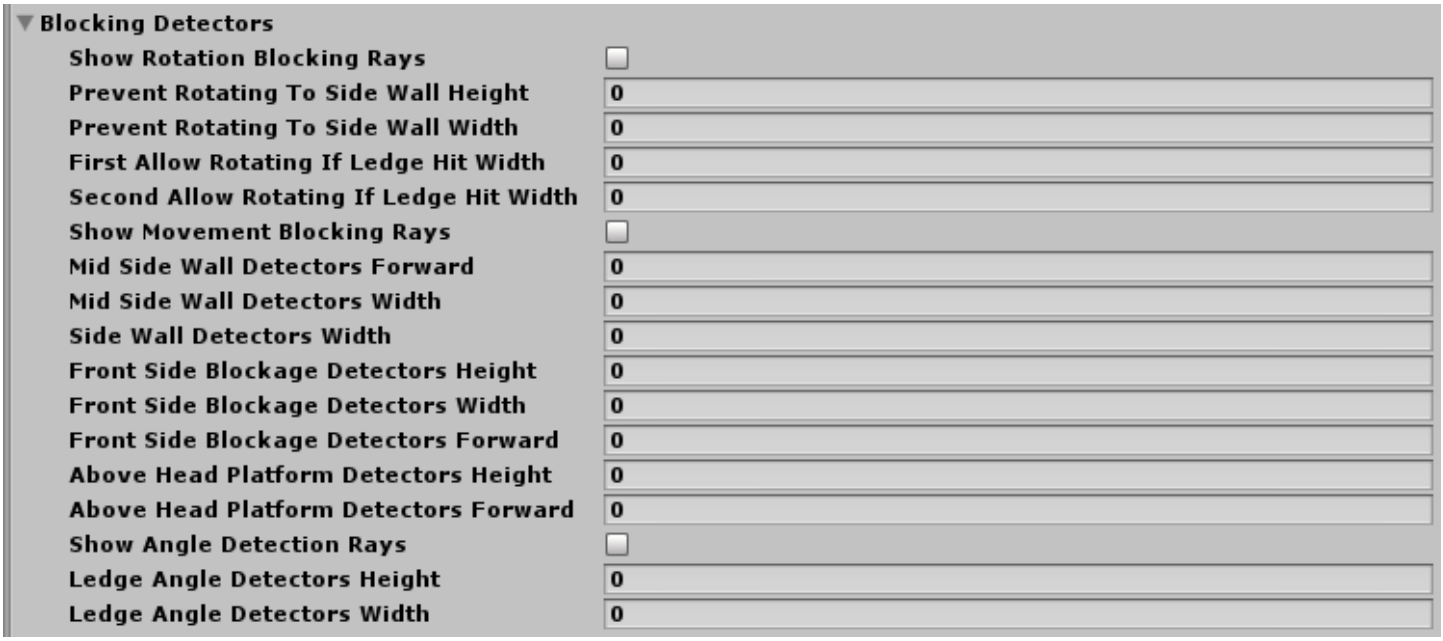
Show Ground Rays – shows the rays that detect whether the player is too close to the ground to move or not

Min Dist From Ground Height –the minimum distance (or height) the player must be from the ground in order to grab on to or move on a ledge

Min Dist From Ground Width – the width of the rays that determine the minimum distance (or height) the player must be from the ground in order to grab on to or move on a ledge

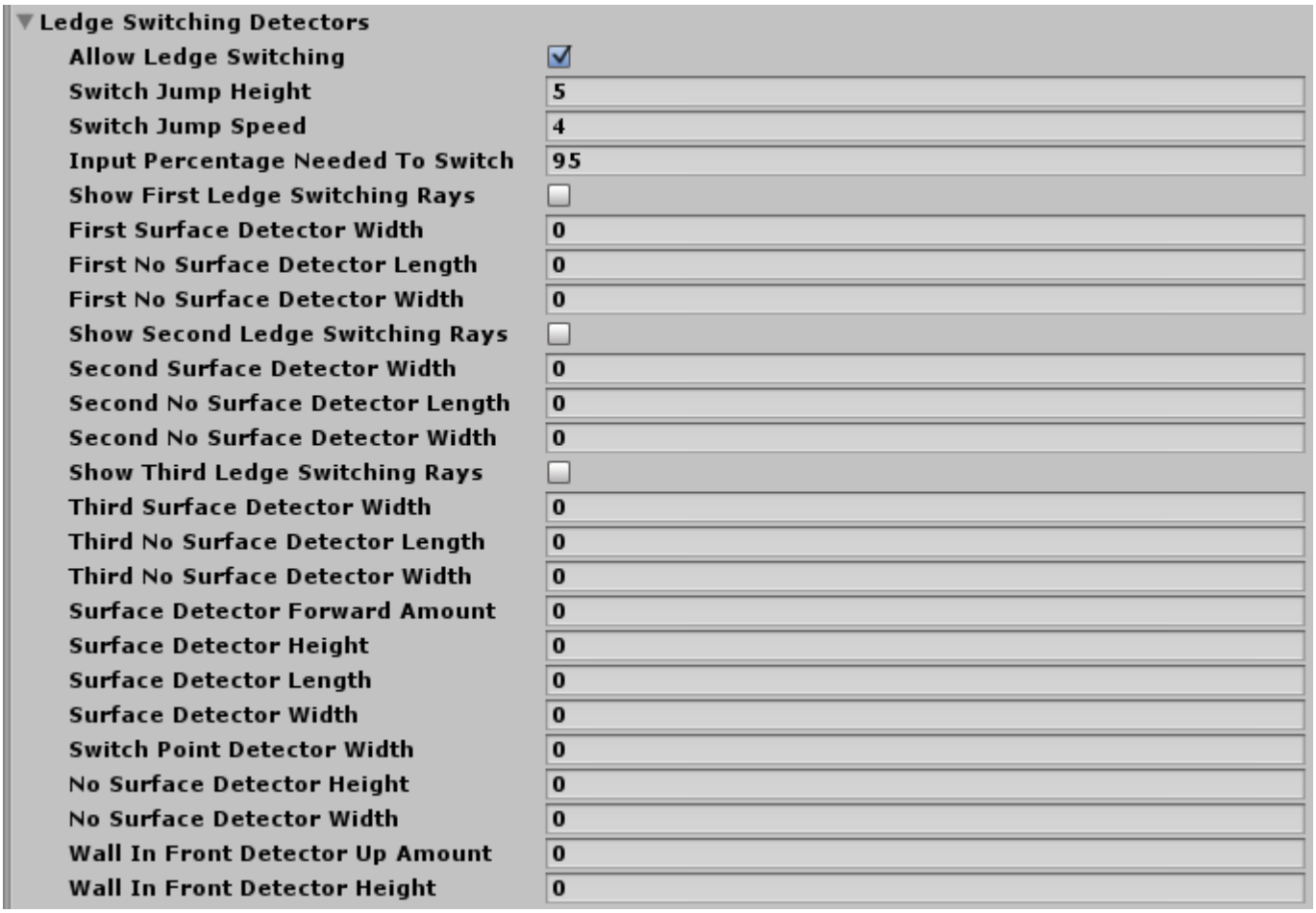
Side Scrolling – variables that determine whether or not the player uses 2.5D side-scrolling

- Lock Movement On Z Axis – locks the movement of the player on the z-axis
- Z Value – the permanent z-value of the player if his movement on the z-axis is locked
- Lock Movement On X Axis – locks the movement of the player on the x-axis
- X Value – the permanent x-value of the player if his movement on the x-axis is locked



Blocking Detectors – detectors that determine whether the player is blocked from moving on a ledge or not

Show Rotation Blocking Rays – shows the rays that detect whether the player is blocked from rotating on a ledge or not
Prevent Rotating To Side Wall Height – the height of the detectors that determine if something to the side of the player (such as a wall) cannot be rotated to
Prevent Rotating To Side Wall Width – the width of the detectors that determine if something to the side of the player (such as a wall) cannot be rotated to
First Allow Rotating If Ledge Hit Width – the width of the first set of detectors that determine if something to the side of the player (such as a ledge) can be rotated to
Second Allow Rotating If Ledge Hit Width – the width of the second set of detectors that determine if something to the side of the player (such as a ledge) can be rotated to
Show Movement Blocking Rays – shows the rays that detect whether the player is blocked from moving on a ledge or not
Mid Side Wall Detectors Forward – the forward distance of the detectors that determine whether a wall in front of and to the side of the player is blocking the player's movement
Mid Side Wall Detectors Width – the width of the detectors that determine whether a wall in front of and to the side of the player is blocking the player's movement
Side Wall Detectors Width – the width of the detectors that determine whether a wall to the side of the player is blocking the player's movement
Front Side Blockage Detectors Height – the height of the detectors that determine if there is something (a collider) in front of the player blocking his movement
Front Side Blockage Detectors Width – the width of the detectors that determine if there is something (a collider) in front of the player blocking his movement
Front Side Blockage Detectors Forward – the forward distance of the detectors that determine if there is something (a collider) in front of the player blocking his movement
Above Head Platform Detectors Height – the height of the detectors that determine if there is a platform/collider above (and slightly in front of) the player's head blocking his movement
Above Head Platform Detectors Forward – the forward distance of the detectors that determine if there is a platform/collider above (and slightly in front of) the player's head blocking his movement
Show Angle Detection Rays – shows the rays that detect whether a ledge's angle is too high to be climbed/moved on or not
Ledge Angle Detectors Height – the height of the detectors that determine the angle of a ledge
Ledge Angle Detectors Width – the width of the detectors that determine the angle of a ledge



Ledge Switching Detectors – detectors that determine whether switching from one ledge to another is possible or not

Allow Ledge Switching – allows the player to switch between ledges (jump from one ledge to another)

Switch Jump Height – the height the player jumps when switching ledges

Switch Jump Speed – the speed the player jumps between ledges

Input Percentage Needed To Switch – the amount of input needed to be applied to the joystick or key in order to switch ledges

Show First Ledge Switching Rays – shows the first set of rays that detect whether switching from one ledge to another is possible or not

First Surface Detector Width – the width of the first set of detectors that determine how large of a surface a ledge must be in order for the player to switch to it

First No Surface Detector Length – the length of the first set of detectors that determine if there is no surface blocking the player from switching ledges

First No Surface Detector Width – the width of the first set of detectors that determine if there is no surface blocking the player from switching ledges

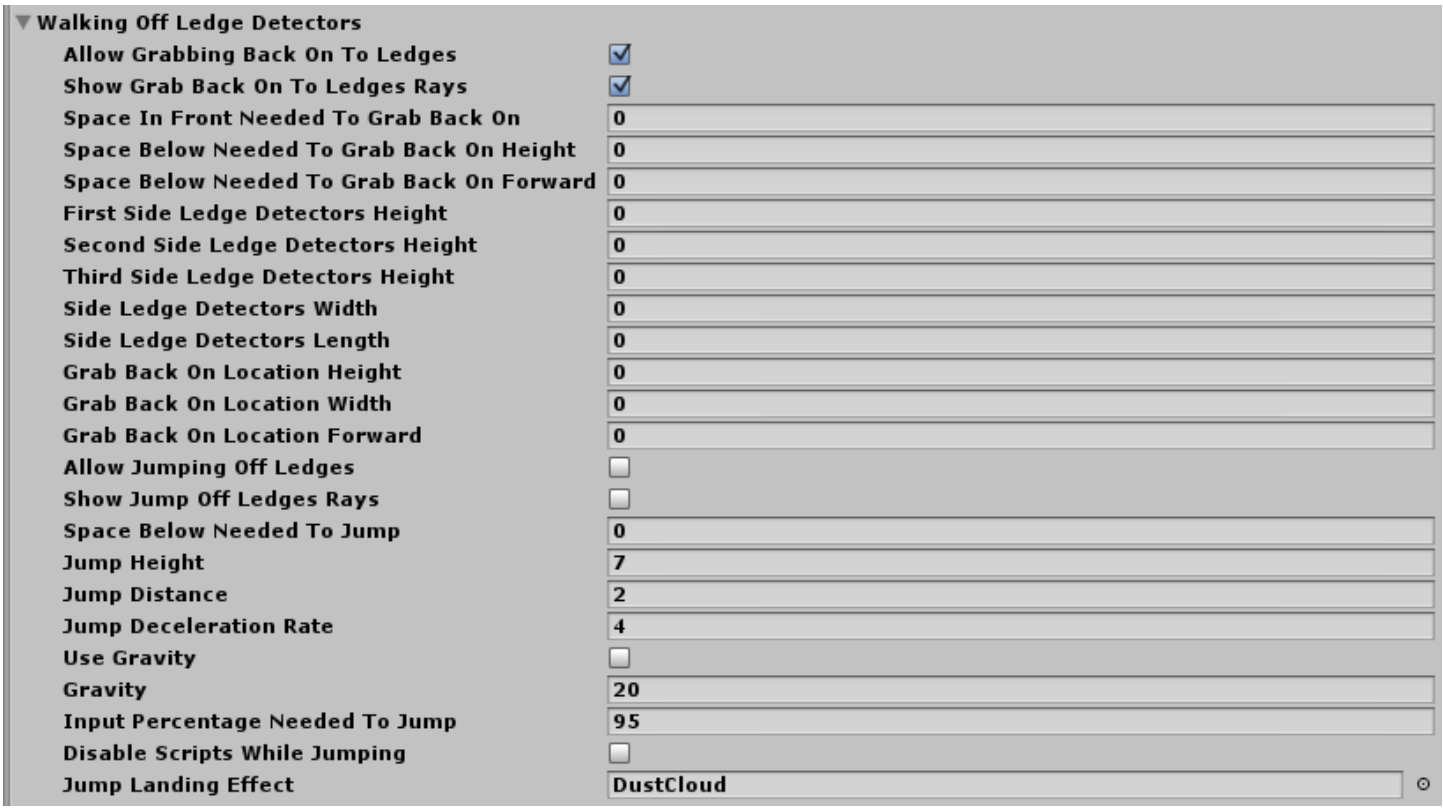
Show Second Ledge Switching Rays – shows the second set of rays that detect whether switching from one ledge to another is possible or not

Second Surface Detector Width – the width of the second set of detectors that determine how large of a surface a ledge must be in order for the player to switch to it

Second No Surface Detector Length – the length of the second set of detectors that determine if there is no surface blocking the player from switching ledges

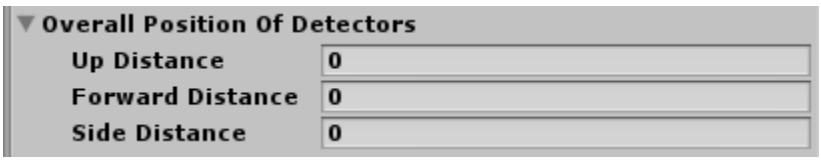
Second No Surface Detector Width – the width of the second set of detectors that determine if there is no surface blocking the player from switching ledges

Show Third Ledge Switching Rays – shows the third set of rays that detect whether switching from one ledge to another is possible or not
Third Surface Detector Width – the width of the third set of detectors that determine how large of a surface a ledge must be in order for the player to switch to it
Third No Surface Detector Length – the length of the third set of detectors that determine if there is no surface blocking the player from switching ledges
Third No Surface Detector Width – the width of the third set of detectors that determine if there is no surface blocking the player from switching ledges
Surface Detector Forward – the forward distance of the detectors that determine how large of a surface a ledge must be in order for the player to switch to it
Surface Detector Height – the height of the detectors that determine how large of a surface a ledge must be in order for the player to switch to it
Surface Detector Length – the length of the detectors that determine how large of a surface a ledge must be in order for the player to switch to it
Surface Detector Width – the width of the detectors that determine how large of a surface a ledge must be in order for the player to switch to it
Switch Point Detector Width – the width of the detectors that determine where the player will switch to
No Surface Detector Height – the height of the detectors that determine if there is no surface blocking the player from switching ledges
No Surface Detector Width – the width of the detectors that determine if there is no surface blocking the player from switching ledges
Wall In Front Detector Up Amount – the up distance of the detectors that determine if there is a wall in front of the player keeping him from switching ledges
Wall In Front Detector Height – the height of the detectors that determine if there is a wall in front of the player keeping him from switching ledges



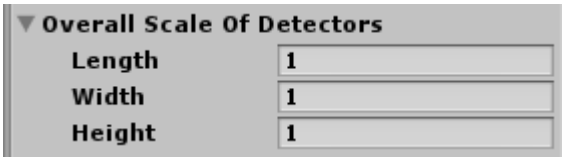
Walking Off Ledge Detectors – detectors that determine if and what to do when the player walks off of a ledge

Allow Grabbing Back On To Ledges – allows the player to grab back on to a ledge that he is walking off of
Show Grab Back On To Ledges Rays – shows the rays that detect if the player is walking off of a ledge
Space In Front Needed To Grab Back On – the amount of space in front of the player needed to grab back on to a ledge
Space Below Needed To Grab Back On Height – the height of the detectors that determine the amount of space below the player needed to grab back on to a ledge
Space Below Needed To Grab Back On Forward – the forward distance of the detectors that determine the amount of space below the player needed to grab back on to a ledge
First Side Ledge Detectors Height – the height of the first set of detectors that determine if there are ledges to the side of the player keeping him from grabbing back on
Second Side Ledge Detectors Height – the height of the second set of detectors that determine if there are ledges to the side of the player keeping him from grabbing back on
Third Side Ledge Detectors Height – the height of the third set of detectors that determine if there are ledges to the side of the player keeping him from grabbing back on
Side Ledge Detectors Width – the width of the detectors that determine if there are ledges to the side of the player keeping him from grabbing back on
Side Ledge Detectors Length – the length of the detectors that determine if there are ledges to the side of the player keeping him from grabbing back on
Grab Back On To Location Height – the height of the detectors that determine where the player will grab back on to
Grab Back On To Location Width – the height of the detectors that determine where the player will grab back on to
Grab Back On To Location Forward – the forward distance of the detectors that determine where the player will grab back on to
Allow Jumping Off Ledges – allows the player to automatically jump off of a ledge that he is walking off of
Show Jump Off Ledges Rays – shows the rays that detect if the player can jump off of a ledge
Space Below Needed To Jump – the amount of space below the player needed in order to jump off of a ledge
Jump Height – the height the player will jump off of a ledge
Jump Distance – the distance the player will jump off of a ledge
Jump Deceleration Rate – the rate at which the player's jump will decelerate
Use Gravity – determines whether or not to use "gravity" while jumping
Gravity – the amount of downward force, or "gravity," that is constantly applied to the player while jumping
Input Percentage Needed To Jump – the amount of input needed to be applied to the joystick or key in order to jump off of a ledge
Disable Scripts While Jumping – determines whether or not to disable the “Scripts To Disable On Grab” scripts (and enable the “Scripts To Enable On Grab” scripts) while jumping
Jump Landing Effect – optional dust effect to appear after landing jump

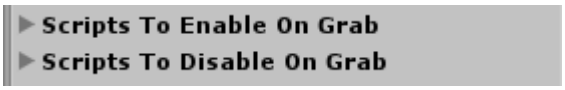


Overall Position Of Detectors – the overall position of every detector in this script

Up Distance – the up distance (or height) of every detector
Forward Distance – the forward distance of every detector
Side Distance – the side distance of every detector



Overall Scale Of Detectors – the overall scale of every detector in this script
Length – the length of every detector
Width – the width of every detector
Height – the height of every detector



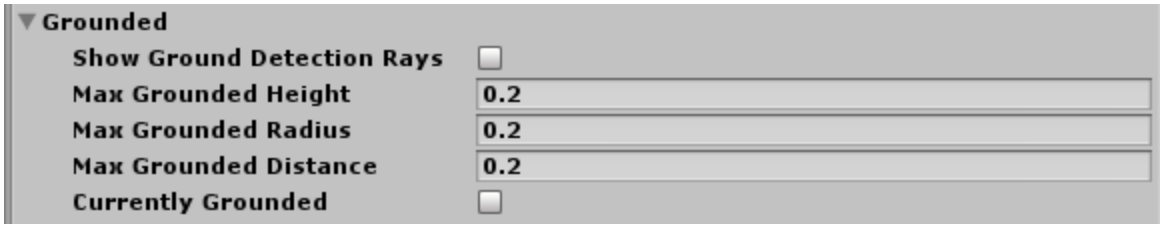
Scripts To Enable On Grab – scripts to enable when the player grabs on to a ledge (scripts disable when the player lets go of a ledge)
Scripts To Disable On Grab – scripts to disable when the player grabs on to a ledge (scripts enable when the player lets go of a ledge)

Current States Of Variables – the current states of the main variables
Ledge Grab Possible – determines if grabbing on to a ledge is currently possible
Climb Up Possible – determines if climbing up and over a ledge is currently possible
Left Movement Possible – determines if movement to the left (while grabbed on to a ledge) is currently possible
Right Movement Possible – determines if movement to the right (while grabbed on to a ledge) is currently possible
Currently On Ledge – determines if the player is currently grabbed on to a ledge or not
Grounded – determines if the player is currently grounded/on the ground or not

Dust Effect – optional dust effect to appear when finished climbing up and on to a ledge
Collision Layers – the layers that the detectors (raycasts/linecasts) will collide with

WallClimbController_Standalone.cs

(This script is the exact same as the "Climbing" section of the PlayerController.cs script, only this is a standalone version of the section)



Grounded – detectors that determine whether the player is grounded or not

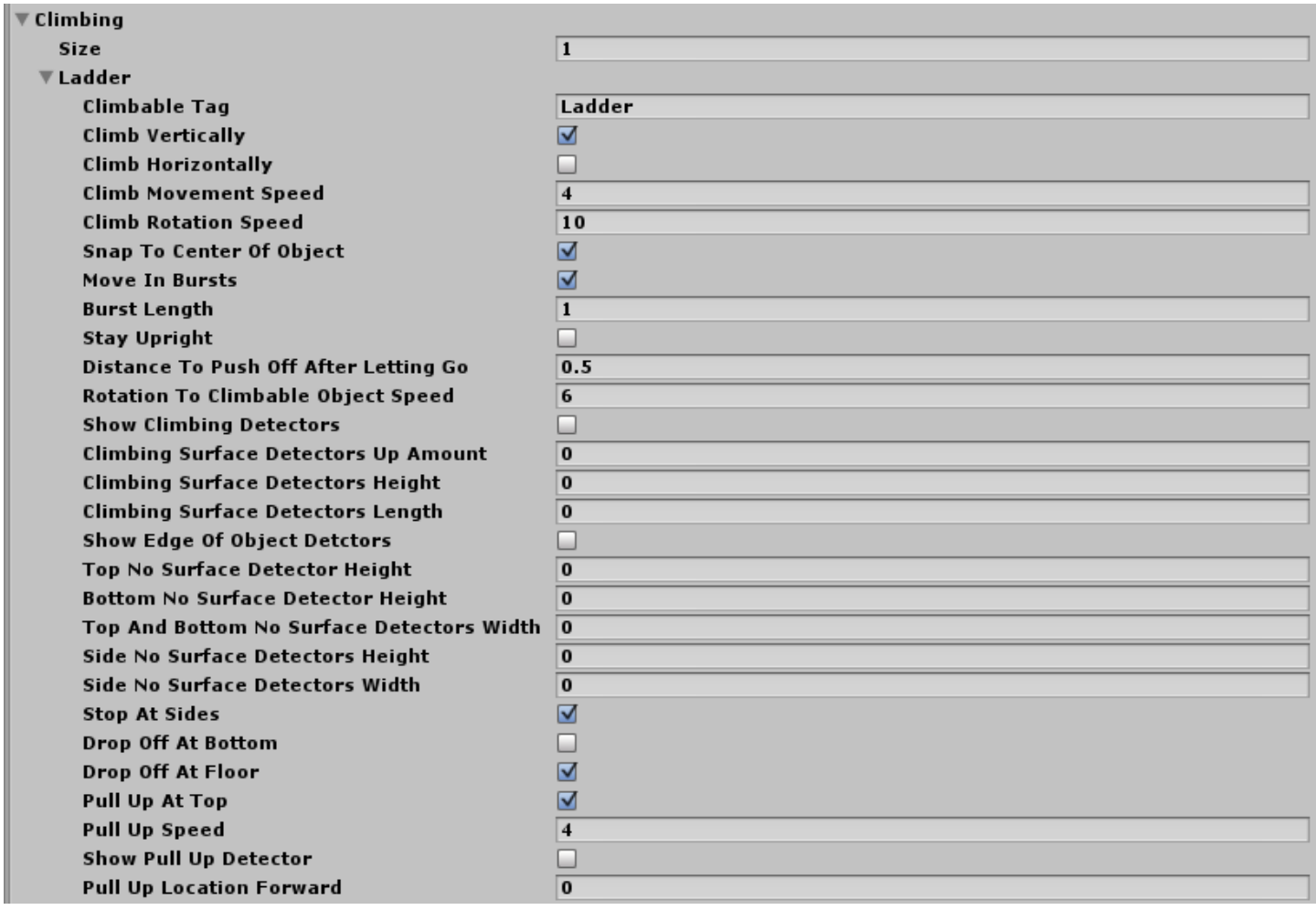
Show Ground Detectors – shows the rays that detect whether the player is grounded or not

Max Grounded Height – the maximum height of the ground the ground detectors can hit to be considered grounded

Max Grounded Radius – the maximum radius of the area ground detectors can hit to be considered grounded

Max Grounded Distance – the maximum distance you can be from the ground to be considered grounded

Currently Grounded – determines if the player is currently grounded/on the ground



Climbing – variables that control the player's ladder and wall climbing

Climbable Tag – the tag of a climbable object

Climb Vertically – determines whether or not the player is allowed to climb vertically

Climb Horizontally – determines whether or not the player is allowed to climb horizontally

Climb Movement Speed – how quickly the player climbs on walls

Climb Rotation Speed – how quickly the player rotates on walls

Snap To Center Of Object – snaps the player to the middle (along the x and z-axis) of the climbable object (most useful for ladders)

Move In Bursts – move in bursts (while on a climbable object)

Burst Length – the amount of time a movement burst lasts

Stay Upright – determines whether or not the player can rotate up and down

Distance To Push Off After Letting Go – the distance the player pushes off of a ladder/wall after letting go

Rotation To Climbable Object Speed – how quickly the player rotates onto a wall to climb

Show Climbing Detectors – determines whether to show or hide the detectors that allow climbing

Climbing Surface Detectors Up Amount – moves the rays that detect the surface of a wall up and down

Climbing Surface Detectors Height – changes the height of the rays that detect the surface of a wall

Climbing Surface Detectors Length – changes the length of the rays that detect the surface of a wall

Show Edge Of Object Detectors – determines whether or not to show the detectors that determine where the top and bottom of a climbable object is

Top No Surface Detector Height – the height of the detector that determines if there is no surface detected at the top of the climbable object, so that the player can pull up or stop before climbing any higher

Bottom No Surface Detector Height – the height of the detector that determines if there is no surface detected at the bottom of the climbable object, so that the player can drop off or stop before climbing any lower

Top And Bottom No Surface Detectors Width – the width of the detectors that determines if there is no surface detected at the top and bottom of the climbable object

Side No Surface Detectors Height – the height of the detectors that determines if there is no surface detected at the sides of the climbable object

Side No Surface Detectors Width – the width of the detectors that determines if there is no surface detected at the sides of the climbable object

Stop At Sides – keeps player from climbing any further sideways once he has reached the side

Drop Off At Bottom – allows player to drop off of a climbable object once he has reached the bottom

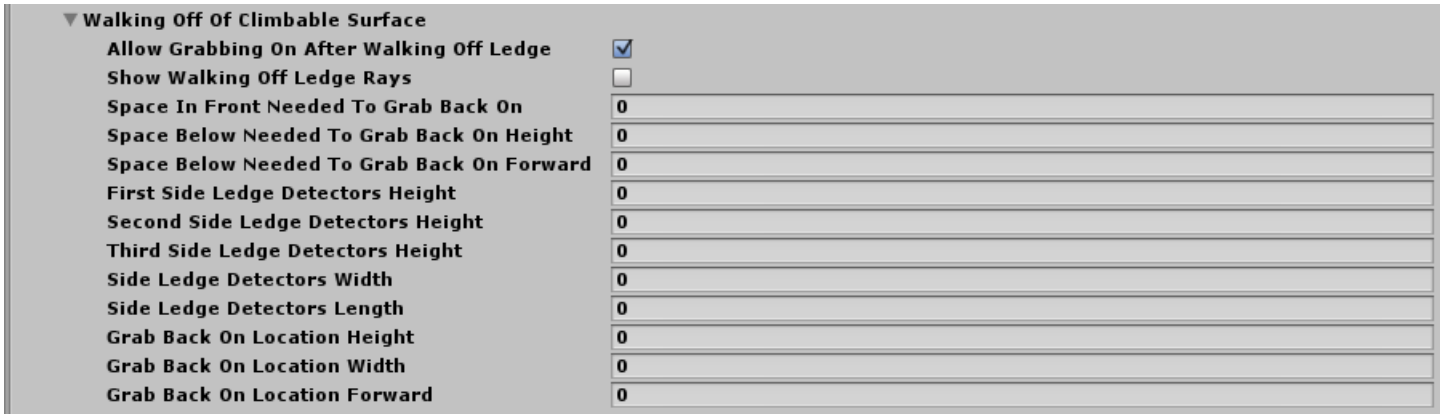
Drop Off At Floor – allows player to drop off of a climbable object once he has reached the floor

Pull Up At Top – allows player to pull up and over a climbable object once he has reached the top

Pull Up Speed – the speed the player pulls up and over ledges once he has reached the top of a climbable object

Show Pull Up Detector – determines whether or not to show the detector that determines where the player pulls up to

Pull Up Location Forward – the forward distance of the detector that determines where the player pulls up to



Walking Off Of Climbable Surface – variables that detect whether the player has walked off a ledge and can grab on to a ladder

Allow Grabbing On After Walking Off Ledge –allows the player to grab on to a climbable surface under the ledge that he is walking off of

Show Walking Off Ledge Rays – shows the rays that detect if the player is walking off of a ledge

Space In Front Needed To Grab Back On – the amount of space in front of the player needed to grab on to a climbable object under a ledge

Space Below Needed To Grab Back On Height – the height of the detectors that determine the amount of space below the player needed to grab on to a climbable object under a ledge

Space Below Needed To Grab Back On Forward – the forward distance of the detectors that determine the amount of space below the player needed to grab on to a climbable object under a ledge

First Side Ledge Detectors Height – the height of the first set of detectors that determine if there are ledges to the side of the player keeping him from grabbing on

Second Side Ledge Detectors Height – the height of the second set of detectors that determine if there are ledges to the side of the player keeping him from grabbing on

Third Side Ledge Detectors Height – the height of the third set of detectors that determine if there are ledges to the side of the player keeping him from grabbing on

Side Ledge Detectors Width – the width of the detectors that determine if there are ledges to the side of the player keeping him from grabbing on

Side Ledge Detectors Length – the length of the detectors that determine if there are ledges to the side of the player keeping him from grabbing on

Grab Back On Location Height – the height of the detectors that determine where the player will grab on to

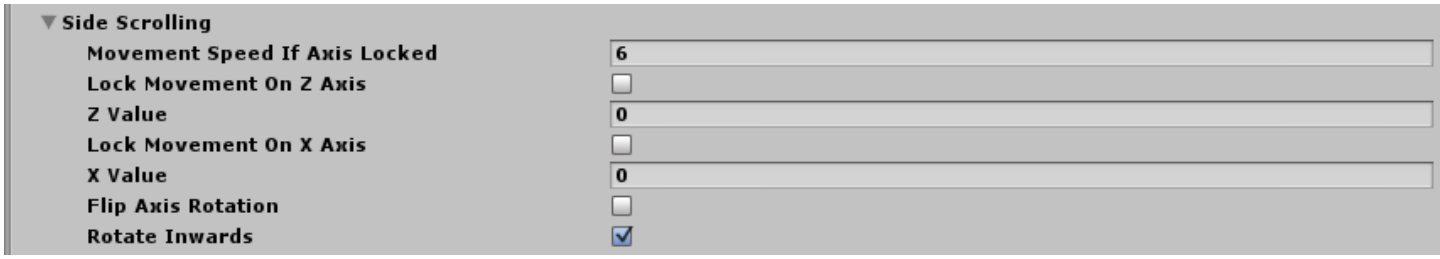
Grab Back On Location Width – the height of the detectors that determine where the player will grab on to

Grab Back On Location Forward – the forward distance of the detectors that determine where the player will grab on to

Scripts To Enable On Grab – scripts to enable when the player grabs on to a wall (scripts disable when the player lets go of a wall)

Scripts To Disable On Grab – scripts to disable when the player grabs on to a wall (scripts enable when the player lets go of a wall)

Push Against Wall If Player Is Stuck – if the script considers the player stuck, the player pushes himself away from the wall until he is free



Side Scrolling – variables that determine whether or not the player uses 2.5D side-scrolling

Movement Speed If Axis Is Locked – the move speed of the player if one of the axis are locked

Lock Movement On Z Axis – locks the movement of the player on the z-axis

Z Value – the permanent z-value of the player if his movement on the z-axis is locked

Lock Movement On X Axis – locks the movement of the player on the x-axis

X Value – the permanent x-value of the player if his movement on the x-axis is locked

Flip Axis Rotation – flips the player's rotation on the non-locked axis (it adds 180 degrees to the player's rotation)

Rotate Inwards – when the player rotates from side to side, he rotates inward (so that you see his front side while he is rotating)

MovingPlatformController_Standalone.cs

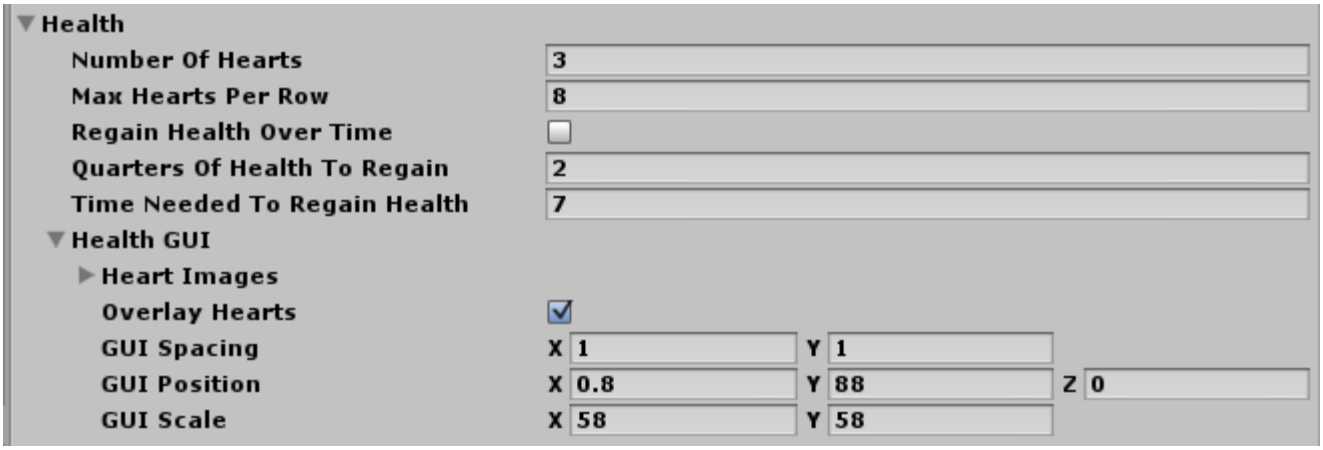
(This script is the exact same as the "Moving Platforms" section of the PlayerController.cs script, only this is a standalone version of the section)

Allow Moving Platform Support – determines whether or not the player can move with moving platforms

Moving Platform Tag – the tag of the moving platform objects

Health.cs

Player Camera – the camera set to follow the player (automatically assigns to Camera.main if not set)



Health – variables that control the player's health

Number Of Hearts – the number of full hearts the player has

Max Hearts Per Row – the maximum number of hearts per row

Regain Health Over Time – allows the player to regain his health over a certain amount of time

Quarters Of Health To Regain – the quarters of health that the player regains after a certain amount of time

Time Needed To Regain Health – the amount of time before the player regains health

Health GUI –variables that control the player's health GUI

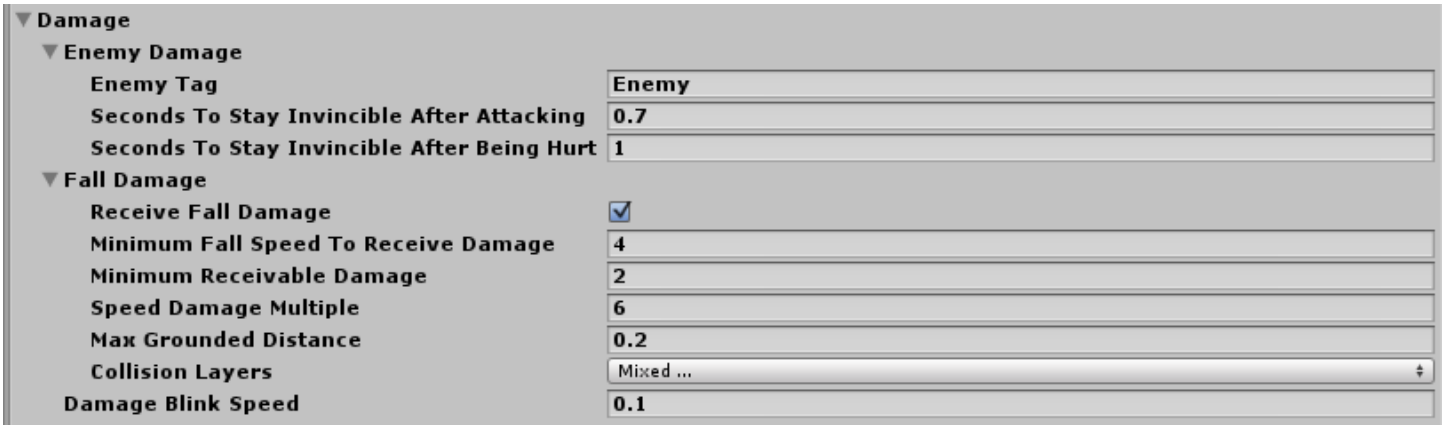
Heart Images – images of the hearts being used to represent the player's health; in order from no health (0 quarters) to full health (4 quarters)

Overlay Hearts – makes each new heart overlay the last one

GUI Spacing – the horizontal and vertical spacing between each heart image/GUI

GUI Position – the position of the heart images/GUI

GUI Scale – the width and height of the heart images/GUI



Damage – variables that control the player's damage

Enemy Damage – variables that control the player's reactions to damage received from or delivered to an enemy

Enemy Tag – the tag of the enemies in the scene

Seconds To Stay Invincible After Attacking – the amount of time the player stays invincible after attacking an enemy

Seconds To Stay Invincible After Being Hurt – the amount of time the player stays invincible after being hurt

Fall Damage – variables that control the player's fall damage

Receive Fall Damage – allows the player to receive damage from falls

Minimum Fall Speed To Receive Damage – the minimum speed the player must fall in order to receive fall damage

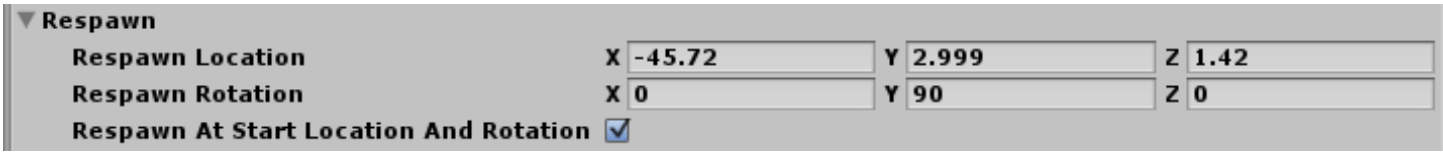
Minimum Receivable Damage – the minimum damage (in quarters of hearts) the player receives from a fall

Speed Damage Multiple – the amount of damage the player receives increased/multiplied by the fall's speed

Max Grounded Distance – the maximum distance you can be from the ground to be considered grounded

Collision Layers – the layers that the grounded detectors (raycasts/linecasts) will collide with

Damage Blink Speed – the speed at which the player blinks (becomes invisible and visible again) after being hurt



Respawn – variables that control the player's respawn

Respawn Location – the location at which the player respawns

Respawn Rotation – the rotation at which the player respawns

Respawn At Start Location And Rotation – enables the player to respawn at the same location and rotation he started with

Health Items – items that give the player health

Health Item Tag – the tag of an item that gives the player health

Quarters Of Health To Regain – the amount of health you regain from the health item (measured in quarter hearts)

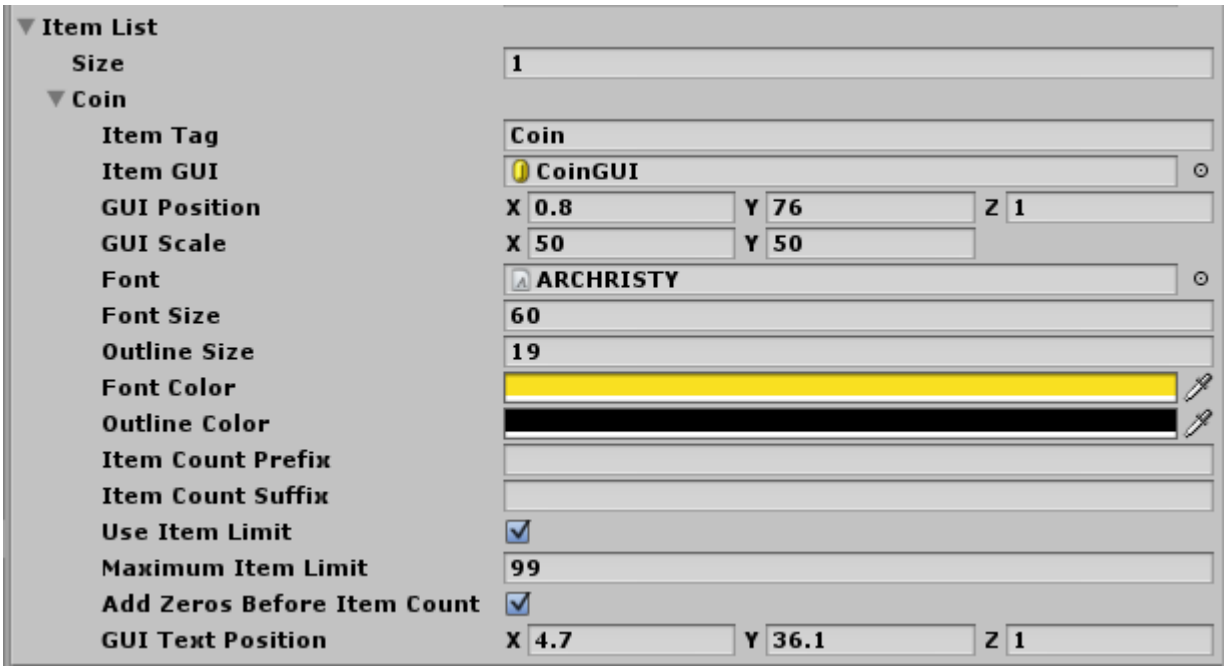
Damage Items – items that take the player's health

Damage Item Tag – the tag of an item that takes the player's health

Quarters Of Health To Lose – the amount of health you lose from the damage item (measured in quarter hearts)

ItemManager.cs

Player Camera – the camera set to follow the player (automatically assigns to Camera.main if not set)



Item List – the items that the player can carry

Item Tag – the tag of the item

Item GUI – the GUI of the item

GUI Position – the position of the item GUI

GUI Scale – the scale of the item GUI

Font – the font of the text

Font Size – the size of the text font

Outline Size – the size of the text outline

Font Color – the color of the text

Outline Color – the color of the text outline

Item Count Prefix – prefix that comes before the item count

Item Count Suffix – suffix that comes after the item count

Use Item Limit – determines whether or not limit the amount of items you can have

Maximum Item Limit – the maximum amount of items you can have

Add Zeros Before Item Count – adds zeros before the item count (the number of zeros added is relative to the maximum item limit)

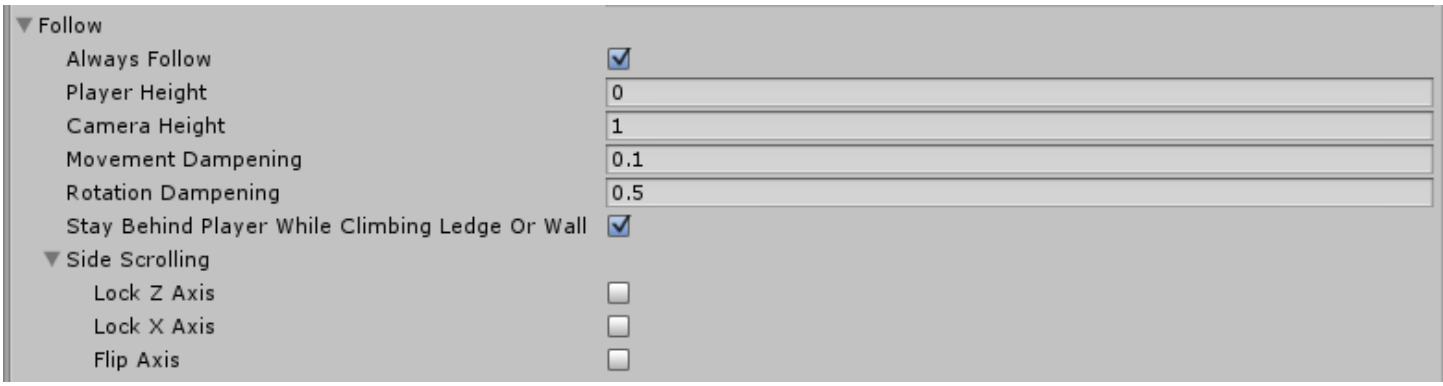
GUI Text Position – the position of the GUI item count text

CameraController.cs

Player – the player set for the camera to follow

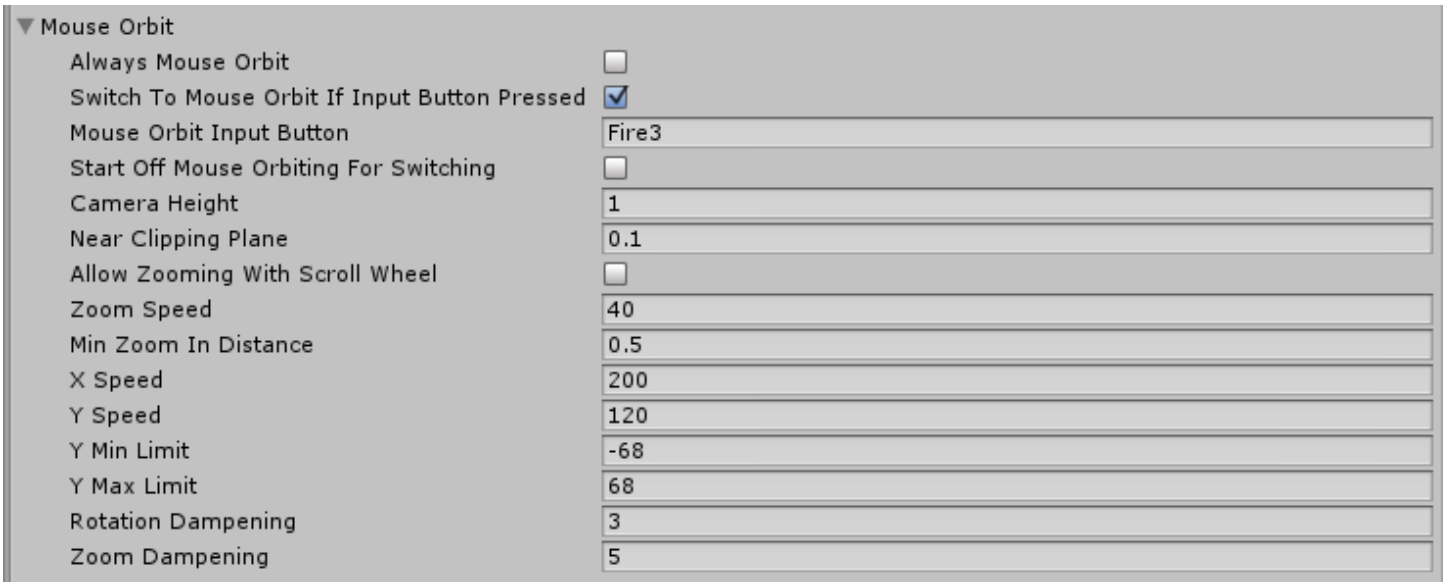
Camera Distance – the distance between the camera and the player

Offset From Wall – the distance between the camera and the wall if they are colliding



Follow – variables that determine the way the camera locks on

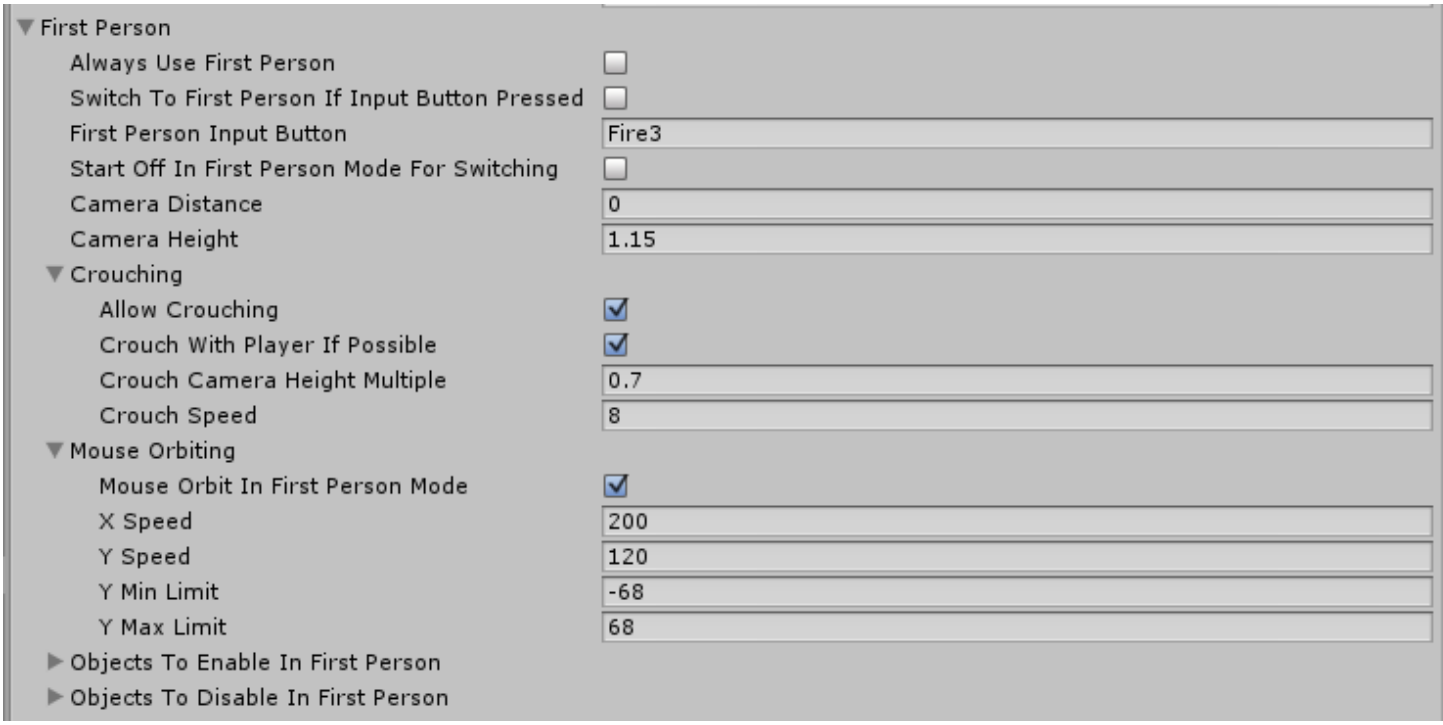
- Always Follow – allows the camera to always follow the player
- Player Height – the height of the player
- Camera Height – the height of the camera
- Movement Dampening – the amount of dampening applied to the movement of the camera
- Rotation Dampening – the amount of dampening applied to the rotation of the camera
- Stay Behind Player While Climbing Ledge Or Wall – makes the camera stay behind the player if he is climbing a ledge or wall
- Side Scrolling – variables that allow the camera to follow the player as a 2.5D side-scroller
  - Lock Z Axis – does not allow the camera to follow the player's z-axis movement
  - Lock X Axis – does not allow the camera to follow the player's x-axis movement
  - Flip Axis – flips the rotation of the non-locked axis (adds 180 degrees to the rotation)



Mouse Orbit – variables that determine the way the camera orbits the player

- Always Mouse Orbit – allows the camera to always orbit the player (without the use of a button), as long as the camera is not following the player
- Switch To Mouse Orbit If Input Button Pressed – switches to mouse orbit mode and back when the "mouseOrbitInputButton" is pressed
- Mouse Orbit Input Button – the button (found in "Edit > Project Settings > Input") that is used to mouse orbit
- Start Off Mouse Orbiting For Switching – if the camera is allowed to switch to mouse orbiting, start off mouse orbiting instead of having to switch to it first
- Camera Height – the height of the camera

Near Clipping Plane – the nearest possible distance between the camera and the player (when the camera is pushed against a wall)
Allow Zooming With Scroll Wheel – allows the user to zoom in between the "distance" and "minZoomInDistance" values (by using the mouse's scroll wheel)
Zoom Speed – the speed the camera zooms in and out when using the scroll wheel
Min Zoom In Distance – the nearest possible distance to the player that the camera can zoom in to
X Speed – the camera's orbit speed (look sensitivity) on the x-axis
Y Speed – the camera's orbit speed (look sensitivity) on the y-axis
Y Min Limit – the minimum y value the camera can orbit to
Y Max Limit – the maximum y value the camera can orbit to
Rotation Dampening – the amount of dampening applied to the rotation of the camera
Zoom Dampening – the amount of dampening applied to the zooming of the camera (this zoom is applied when the camera collides with a wall, and zooms in or out to avoid it)



First Person – variables that determine if the player can enter first person mode
Always Use First Person – allows the camera to always stay in first person mode (as long as the camera is not following or mouse orbiting)
Switch To First Person If Input Button Pressed – switches to first person mode and back when the "firstPersonInputButton" is pressed
First Person Input Button – the button (found in "Edit > Project Settings > Input") that is used to enter first person mode
Start Off In First Person Mode For Switching – if the camera is allowed to switch to first person mode, start off in first person mode instead of having to switch to it first
Camera Distance – the distance of the camera from the player
Camera Height – the height of the camera
Crouching – variables that determine if the camera can crouch (lower down) with the player while in first person mode
Allow Crouching – determines if the camera can crouch (lower down)

Crouch With Player If Possible – allows the camera to crouch (lower down) if the player has the script, "PlayerController.cs," and is crouching

Crouch Camera Height Multiple – what to multiply the height of the camera by while crouching

Crouch Speed – the speed at which the camera crouches down and uncrouches

Mouse Orbiting – variables that determine if the camera can mouse orbit while in first person mode

Mouse Orbit In First Person Mode – allows the camera to mouse orbit while in first person mode

X Speed – the camera's orbit speed (look sensitivity) on the x-axis

Y Speed – the camera's orbit speed (look sensitivity) on the y-axis

Y Min Limit – the minimum y value the camera can orbit to

Y Max Limit – the maximum y value the camera can orbit to

Objects To Enable In First Person – gameObjects to enable when the camera is in first person mode (gameObjects disable when the camera exits first person mode)

Objects To Disable In First Person – gameObjects to disable when the camera is in first person mode (gameObjects enable when the camera exits first person mode)

Locking On – variables that determine how the camera locks on

Allow Locking On – determines whether or not to allow locking on

Lock On Input Button – the button or axis (found in "Edit > Project Settings > Input") pressed to lock on

Bar Texture – the texture that appears on the top and bottom portion of the screen while locking on

Bar Coverage – the percentage of the screen that the bar texture covers

Lock On Speed – the speed at which the camera locks on behind the player

Locking On To Tags – variables that determine which objects to lock on to

Tags To Lock On To – the tags of the objects that can be locked on to

Lock In Distance – how close you have to be to the tagged object to lock on

Lock Out Distance – how far you have to be from the tagged object (if you are already locked on to it) to lock out

Arrow Pointer – the object (arrow) that appears over the tagged objects head when you are able to lock on to him

Arrow Pointer Height – how high the arrow pointer is above the tagged objects head

Arrow Pointer Bounce Height – how high the arrow pointer bounces

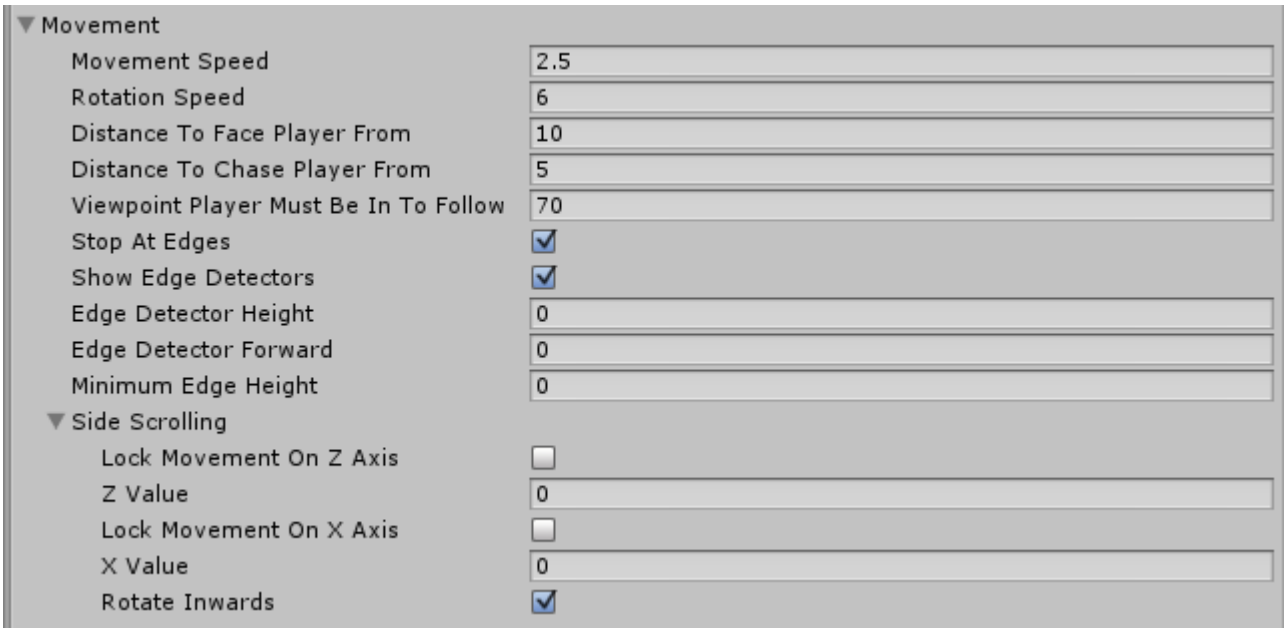
Arrow Pointer Bounce Speed – the speed at which the arrow pointer bounces

Target Same Object While Locked On – stays locked on to the same target, even if another target moves closer

Collision Layers – the layers that the detectors (raycasts/linecasts) will collide with

EnemyAI.cs

Player – the player in the scene

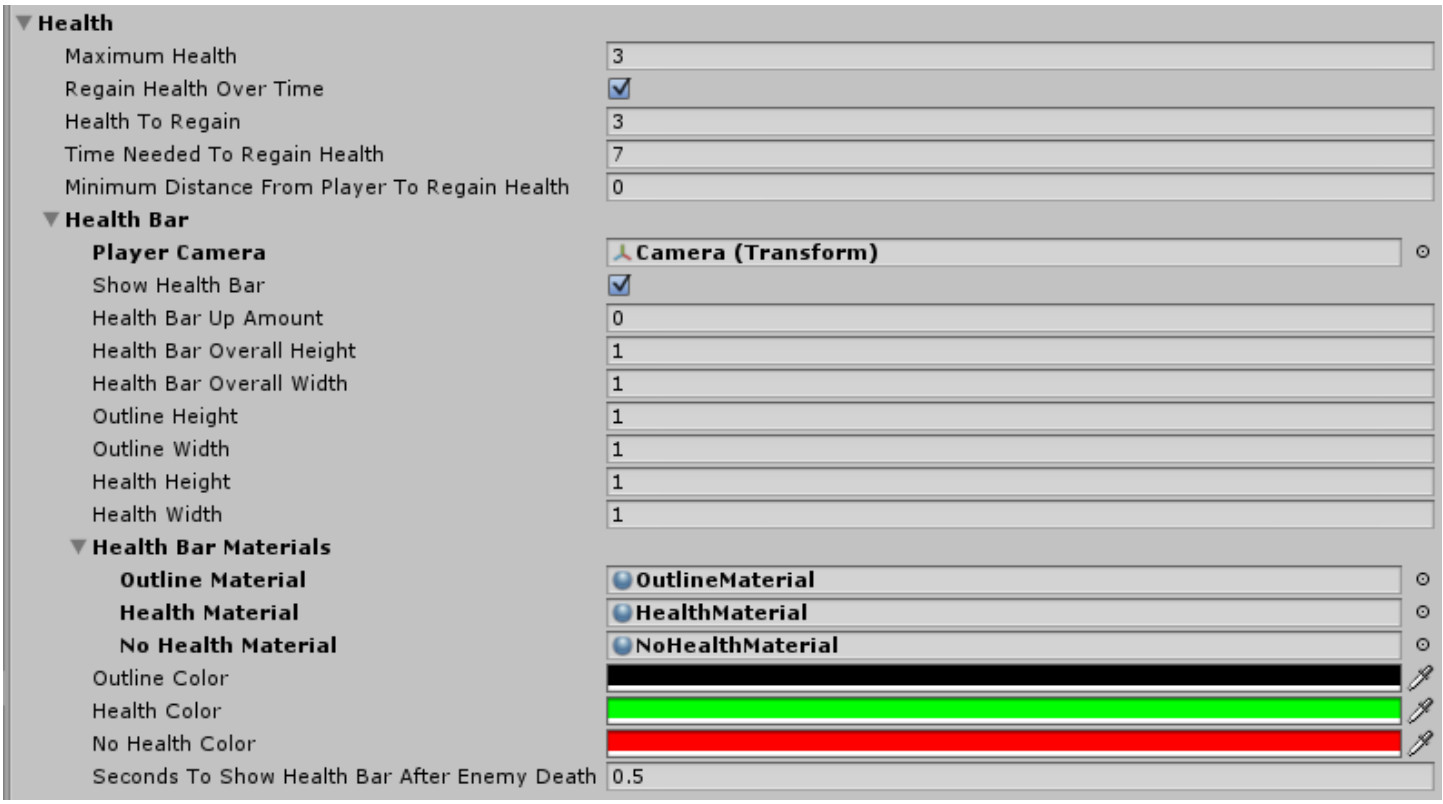


Movement – variables that determine the enemy's movement

- Movement Speed – the enemy's movement speed
- Rotation Speed – the enemy's rotation speed
- Distance To Face Player From – the maximum distance the player can be for the enemy to face/look at the player
- Distance To Chase Player From – the maximum distance the player can be for the enemy to chase the player
- Viewpoint Player Must Be In To Follow – the size of the enemy's viewpoint; if the player is inside the viewpoint, the enemy fill follow him
- Stop At Edges – stops the enemy before he walks off the edge
- Show Edge Detectors – detectors that determine where the edge is located
- Edge Detector Height – the height of the edge detectors
- Edge Detector Forward – the forward distance of the edge detectors
- Minimum Edge Height – the height of the edge detector that determines if there is surface below the player
- Side Scrolling – variables that determine whether or not the enemy uses 2.5D side-scrolling
  - Lock Movement On Z Axis – locks the movement of the enemy on the z-axis
  - Z Value – the permanent z-value of the enemy if his movement on the z-axis is locked
  - Lock Movement On X Axis – locks the movement of the enemy on the x-axis
  - X Value – the permanent x-value of the enemy if his movement on the x-axis is locked
  - Rotate Inwards – when the enemy rotates from side to side, he rotates inward (so that you see his front side while he is rotating)

Attack – variables that determine the enemy's attack

Attack Power – the amount of damage you inflict to the player on collision (measured in quarter hearts)



Health – variables that determine the enemy's health

Maximum Health – the health of the enemy

Regain Health Over Time – allows the enemy to regain his health over a certain amount of time

Health To Regain – the health that the enemy regains after a certain amount of time

Time Needed To Regain Health – the amount of time before the enemy regains health

Minimum Distance From Player To Regain Health – the distance from the player that the enemy must be to regain health

Health Bar – variables that determine whether the enemy has a health bar or not

Player Camera – the camera set to follow the player (automatically assigns to Camera.main if not set)

Show Health Bar – determines whether or not to show a health bar above the enemy

Health Bar Up Amount – the up distance of the health bar above the enemy

Health Bar Overall Height – the overall height of the health bar above the enemy

Health Bar Overall Width – the overall width of the health bar above the enemy

Outline Height – the height of the health bar's outline

Outline Width – the width of the health bar's outline

Health Height – the height of the health bar's health

Health Width – the width of the health bar's health

Health Bar Materials – the materials of the health bar

Outline Material – the material of the health bar's outline

Health Material – the material of the health bar's health percentage

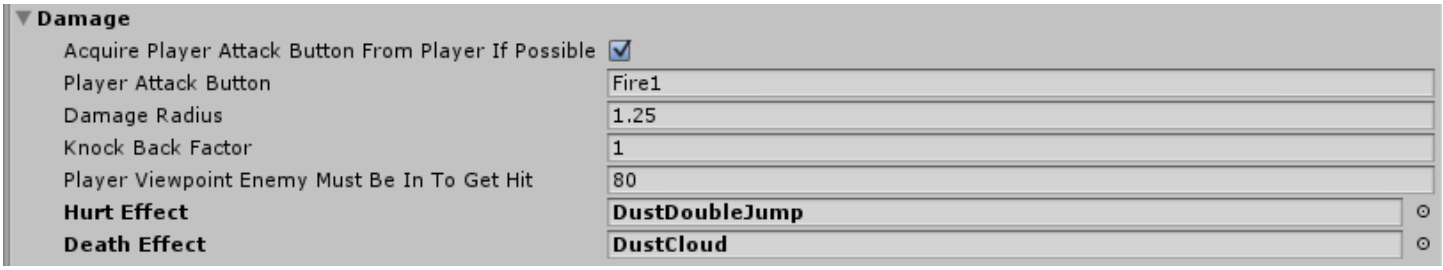
No Health Material – the material of the health bar's no health percentage

Outline Color – the color of the health bar's outline

Health Color – the color of the health bar's health percentage

No Health Color – the color of the health bar's no health percentage

Seconds To Show Health Bar After Enemy Death – the amount of time that the health bar stays visible after the enemy has been killed



Damage – variables that determine the enemy's damage

Acquire Player Attack Button From Player If Possible – if the player has the "PlayerController.cs" script attatched to him, aquire the player attack button from that script

Player Attack Button – the button (found in "Edit > Project Settings > Input") that the player uses to attack the enemy

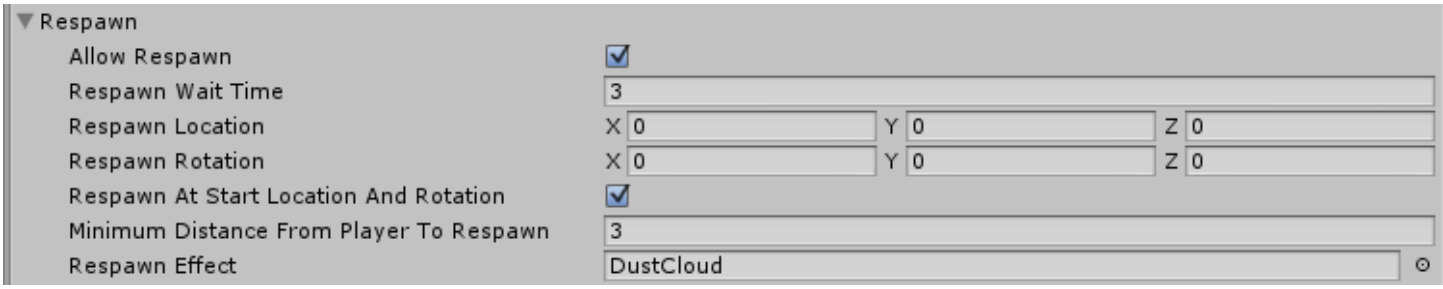
Damage Radius – the radius the player must be in to hurt the enemy

Knock Back Factor – the distance that the enemy gets knocked back after getting hit

Player Viewpoint Enemy Must Be In To Get Hit – the size of the player's viewpoint; if the enemy is inside of the player's viewpoint (and the player is inside of the enemy's hit area) and the player attacks, the enemy will receive damage

Hurt Effect – optional effect to appear when the enemy gets hurt

Death Effect – optional effect to appear when the enemy gets killed



Respawn – variables that control the enemy's respawn

Allow Respawn – determines whether or not the enemy can respawn after being killed

Respawn Wait Time – the amount of time to wait, after the enemy has been killed, to respawn

Respawn Location – the location at which the enemy respawns

Respawn Rotation – the rotation at which the enemy respawns

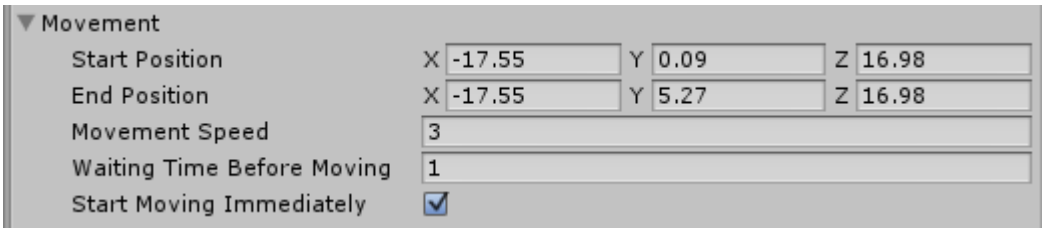
Respawn At Start Location And Rotation – enables the enemy to respawn at the same location and rotation he started with

Minimum Distance From Player To Respawn – the minimum distance the player must be from the respawn location in order to respawn

Respawn Effect – optional effect to appear when the enemy respawns

Collision Layers – the layers that the detectors (raycasts/linecasts) will collide with

MovingPlatform.cs



Movement – variables that determine the movement of the platform

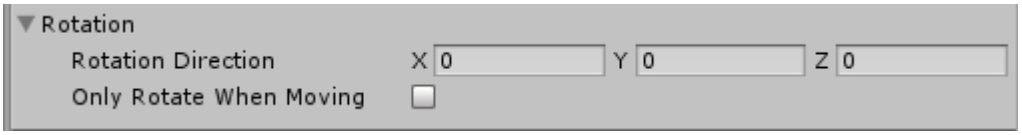
Start Position – the Vector3 that the platform starts at (before moving to the end position)

End Position – the Vector3 that the platform ends at (before moving back to the start position)

Movement Speed – the speed of the moving platform

Waiting Time Before Moving – the amount of time the platform waits (once it has reached the start or end position) before moving again

Start Moving Immediately – for the first time that the platform starts moving, do not wait



Rotation – variables that determine the rotation of the platform

Rotation Direction – the direction to rotate towards

Only Rotate When Moving – only allows the platform to rotate if the platform is already moving/not waiting to move