User Manual

Table of Contents

Section 1. Initial Setup

Before you can get started creating a camera, you will need to set up a few things in Unity's tag manager.

1) On the main taskbar go to *Edit -> Project Setting -> Tags*

2) In the tags foldout add two tags "*HorizontalWall*" and "*VerticalWall*"

3) Under any one of the User Layer's add "*Wall*" and "*InvisibleBound*"

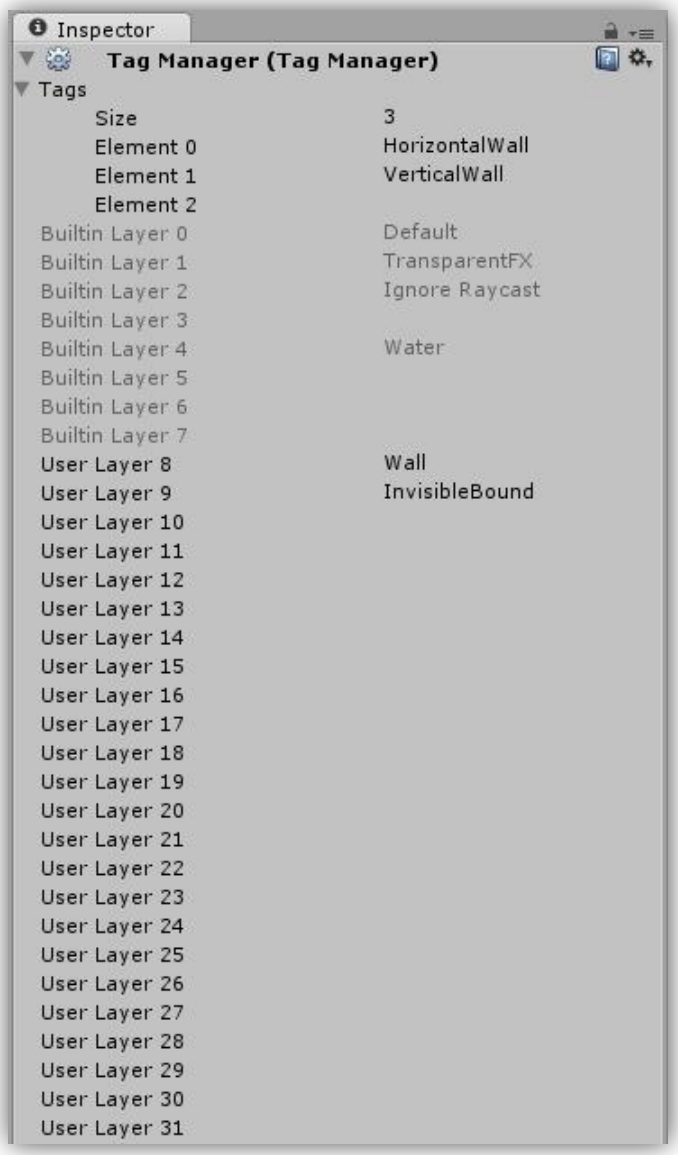


Figure 1 Unity's Tag Manager

Section 2. Creating a Camera

UniVenture uses a custom Editor to make setting up your camera quick and simple. First open up the custom Editor by going to

GameObject -> UniVenture -> Create Camera

You should see this window open.

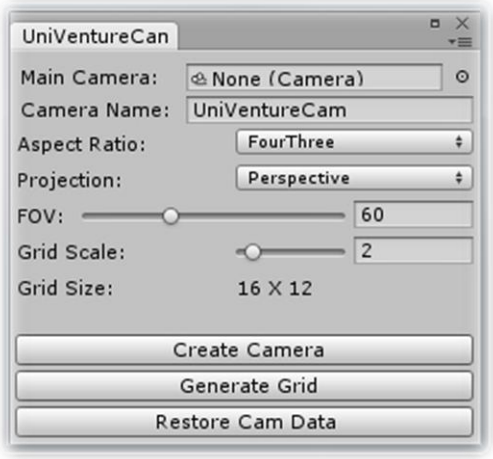


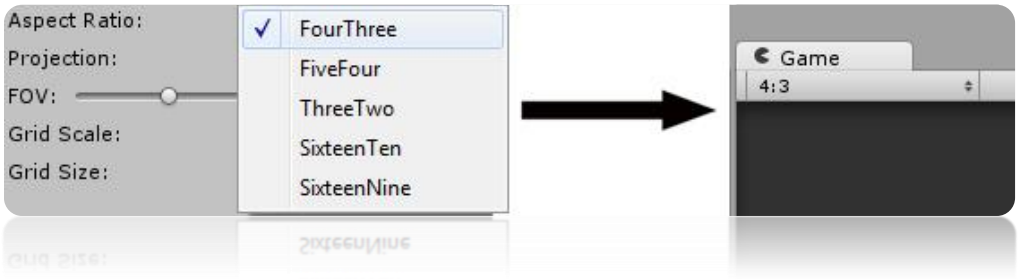
Figure 2 Create Camera Window

2.1 Referencing the MainCamera

Every new scene in Unity has a *MainCamera*. To reference this camera simply drag the *MainCamera* from the Hierarchy into the Object Field labeled Main Camera. Alternatively you can click the circle to the right and select *MainCamera* from the list. You can name the camera by typing your desired camera name into the Camera Name field. By default this is set to *UniVentureCam*.

2.2 Aspect Ratio

In the Aspect Ratio field use the drop down menu to select your desired aspect ratio. This will be used in calculating the camera's height so it is important that you ensure the aspect ratio Unity is using matches this one.



2.3 Projection

Unity's camera two different methods of projection: Perspective and Orthographic. Perspective will render object with a field of view, giving objects perspective while Orthographic with render objects uniformly with no perspective. UniVenture's Camera system supports both projections.

2.3.1 FOV (Field of View)

If Perspective is enabled, you may select a desired FOV in the range of [1, 179]. It is recommended that you use the default of 60 degrees however 30, 45, or some other multiple of 15 also works quite well.
(This option is removed if Orthographic projection is enabled)

2.4 Grid

Unity has a built-in grid system that UniVenture utilizes to facilitate development. Here you can specify grid size by using the Grid Scale slider. This slider will increase the size of the grid by a multiple of the aspect ratio. The Grid Size label displays the current grid size (Width X Height) and updates according to the Grid Scale.

2.5 Finishing Up

2.5.1 Creating Camera

Once you're satisfied with the settings you can click on the "Create Camera" button at the bottom of the window. At this point the custom Editor will apply your settings.

2.5.2 Generating a Grid

After creating the camera you can generate a *HelperGrid* to facilitate level development. By clicking on the "Generate Grid" button the custom Editor will use the information you provided about the camera to generate a large grid to use as a visual reference in building and designing environments.

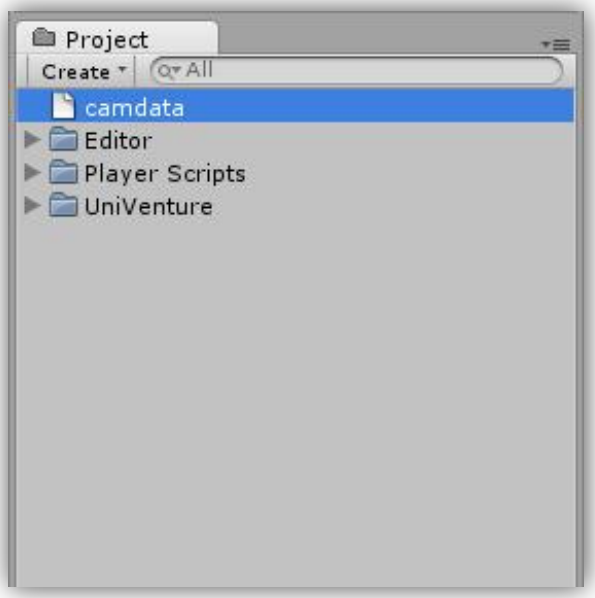
2.5.3 Restoring CamData

In the event that something happens to the "*camdata*" asset file, you can restore the file by re-applying your settings and clicking "Restore Cam Data."

Section 3. Setting up the Camera

Now that you've created your camera, there are just a couple more things you need to do before it's ready for use.

3.1 CamData



When you clicked on "Create Camera" a file called "*camdata"* was created. This asset file stores data based on your settings that tell the camera how to behave. However, our camera doesn't currently know this file exists, so we first need to reference it.

To reference the *camdata* asset, simply click on the foldout tab for "UniVentureCam" (or whatever you named the camera) and select the first object in the list "_MainTrigger". In the Inspector window you should see the "Main Trigger Script" already attached to the camera. In the second field labeled "Camera Data" drag and drop the camdata asset file into this object field.

3.2 Player Script

Now our camera needs a reference to our player's movement script so that it may deactivate this script during transitions to ensure a smooth transition from room to room. To reference the player simply locate your player character in the Hierarchy and drag and drop it into the third field labeled "Player Script." This will automatically return the first script attached to your character so it is important to make sure that the first script you attach to your player is the movement script. By default this only works with rigidbody characters and will not work with CharacterControllers. See Section 4 for setting the camera up to work with CharacterControllers.

3.3 Camera Modes

1) **Dungeon –** In Dungeon mode the camera will follow the player if the room is larger than the visible area. Camera movement will still be limited to the bounds of the room.

2) **Overworld –** In Overworld mode the camera will stay fixed to the visible area until the player leaves the bounds of the visible area.

3.4 Camera Styles

1) **Modern –** Camera will transition into the next room at a rate defined by **Camera Move Time.**

2) **Classic –** Camera will jump to the next room with no transition.

3.5 Camera Move Time

Camera Move Time is the amount of time (in seconds) that it takes the camera to transition to the next room. This only applies to the **Modern** camera style.

3.5 Debug Lines

The camera uses raycast's to gather information about the room. It is important when designing levels that these raycast's hit the walls that define each room. If you are concerned that these rays are not making contact with the walls you can check **Draw Debug Lines** which will draw the path that each ray follows (indicated by red and blue lines) in the Scene view.

Section 4. Setting Camera Up To Work With CharacterController

By default the camera is set up to work with rigidbody characters only. However, we can easily extend the functionality to work with CharacterControllers.

In the project folder under UniVenture click the foldout toggle for Camera Trigger Scripts and open up the **MainTrigger.cs** file.

On line 7 change the visibility of the **playerScript** field from **public** to **private**.

Scroll down to lines 99 – 101 and you should see the following at the top of the Start() method:

```
97      void Start ()
98      {
99          //if using player with character controller delete comment and replace "MovementScriptNameHere"
100         //with the name of your player movement script.
101         //playerScript = GameObject.FindWithTag("Player").GetComponent<MovementScriptNameHere>();
102
```

On line 101 delete the comment and inside the **<>** replace “MovementScriptNameHere” with the name of the player movement script attached to your player. It is important to make sure that the script you’re referencing is the script that is actually receiving input from the user to move the player.

Section 5. Grid Editor

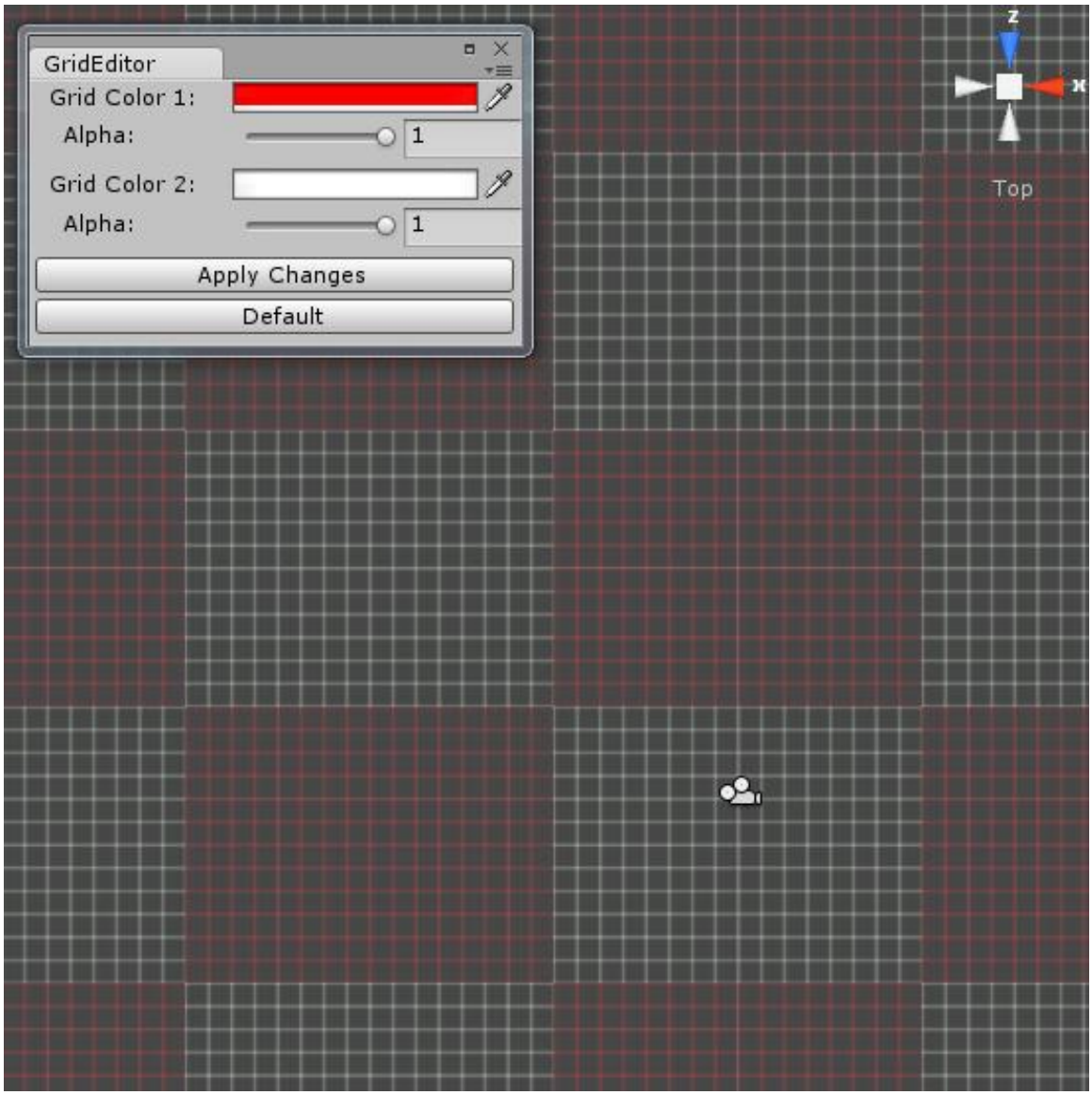


Figure 3 Grid Editor with HelperGrid

The Grid Editor is a very simple tool that allows you to change the color and alpha value of each section of grid on the *HelperGrid*. Each block of color represents the camera's visible area. Clicking in the color field will bring up the Color Picker. The alpha values range from 0 to 1 with 0 being completely invisible and 1 being fully visible. Once you are satisfied with your changes click "Apply Changes." Clicking "Default" will restore the default colors of red and white and alpha values of one.

Section 6. Toolset

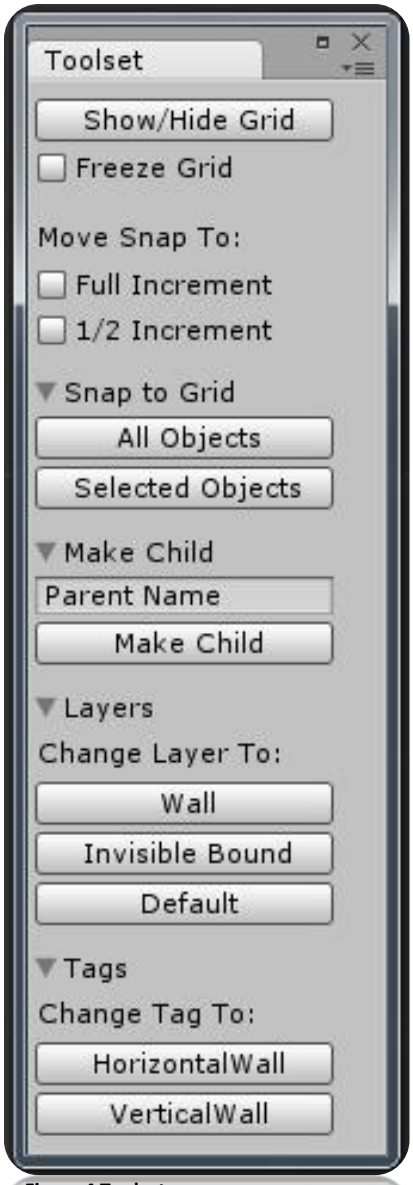


Figure 4 Toolset

6.1 Grid Options

1) **Show/Hide Grid –** Toggles the *HelperGrid* so you can hide it when you don't need it.
2) **Freeze Grid –** If checked, it will prevent you from accidently selecting the *HelperGrid* or any of its children while working in Scene view. (Toolset must remain open to keep the grid frozen)

6.2 Move Snaps

1) **Full Increment –** Moves selected object in increments of 1
2) **½ Increment –** Move selected objects in increments of 0.5

6.3 Snaps

1) **All Objects –** Snaps every object in the scene to grid
2) **Selected Objects –** Snaps only selected objects to grid

6.4 Make Child – Makes every selected object the child of an empty game object. You can name the parent object by typing the desired name into the text field.

6.5 Layers

1) **Wall –** Changes all selected objects to the "*Wall*" layer
2) **Invisible Bound -** Changes all selected objects to the "*InvisibleBound*" layer
3) **Default -** Changes all selected objects to the "*Defualt*" layer

6.6 Tags

1) **HorizontalWall -** Changes all selected objects tags to "*HorizontalWall*"
2) **VerticalWall -** Changes all selected objects tags to "*VerticalWall*"

Section 7. Resources

If you have any questions please visit the UniVenture Camera forum thread on the official Unity forums:
http://forum.unity3d.com/threads/101197-UniVenture-Camera-A-Zelda-Style-Camera-for-Unity

For tutorial vids that cover everything in this manual and more please visit:
http://www.youtube.com/playlist?list=PL73E38DEB7B3F8DD0&feature=plcp