

# **2.5D Toolkit**

VB Game Studio

By Valter Buccinà

## User Manual

V2.03

[Website](#)

[Facebook](#)

[Youtube channel](#)

## Table of Contents

Note: Version 2.03 .....	4
Note: Version 2.02 .....	5
Introduction .....	6
Watch videos .....	8
How to use .....	9
General options .....	12
General options .....	12
Settings.....	15
Gizmos setting .....	15
Areas color.....	16
Background image .....	16
2.5DTK Env.....	18
Navigation .....	24
Prepare NavMesh .....	24
View navigation .....	26
Objects .....	27
Place object .....	27
Mesh .....	29
Make mesh .....	29
Advanced.....	30
Move whole environment.....	30
Save data .....	30
2.5D Toolkit folders.....	31
Tips.....	32
Inline help.....	32
Mouse functions .....	33
General tips .....	34
A way to draw a simple mesh.....	36
Demo scene: Mainframe.....	39

Demo scene: IBM.....	47
Demo scene: Cableway Orthographic .....	51
Demo scene: Cableway Perspective .....	55
Demo scene: NicoHome.....	59
Demo scene: RPG .....	63
Demo scene: Pools.....	67
Demo scene: Salon .....	69
Demo scene: Villa (Multilevel).....	73
A way to use Meter and Cube .....	76
Add shadow.....	81
Move whole environment.....	88

# Note: Version 2.03

---

In the 2.03 version Meter option has been improved and Sprite Auto Positioning option has been added.

- [2.5DTK Env. Tab / <Enable Meter>] **<Automatic meter>** Use **<Save width>** and **<Save depth>** to enter width and depth measurements of the scene. From now on these values will update according to the changes you will make to camera and floor.

**<Hide lines>** Hide/Show reference lines.

**<Reset values>** Reset settings.

**<Show label>** Show/Hide the summary label of the width and depth measurements at the top left of the screen.

**<Label font size>** Set label size.

Watch video: [v2.03 - 2.5D Toolkit: Improved Meter](#)

- [Objects tab] **<Auto position>** To be used with sprites of the same size as the background image. This mode allows to place sprites in seconds with just a click.

In orthographic projection sprite will be overlapped with the utmost precision on the background image from which it was cut, in perspective projection with the zoom at maximum it will show a difference in scale in the order of a few ten thousandths due to resizing. This millimeter difference will not be absolutely visible in the game but if you wish you can correct it with the 2.5D Toolkit controls.

**Note: sprite's pivot must be set to Center.**

**<Hide pivot>** Hide/Show sprites pivot.

Watch video: [v2.03 - 2.5D Toolkit: A way to place sprite with just a click in one second](#)

Manual does not show these new changes.

# Note: Version 2.02

---

**Version 2.02 involves support for Tilemap.**

A new Tilemap tab has been added.

By dragging Grid and Tilemap into their fields you can create areas on the Tilemap.

Note that with the default settings it will not be possible to use Unity Navigation.

**<Rotate 90°>** option will rotate camera and tilemap 90° allowing areas to be baked, and unchecking it will set the default environment again.

You can build Tilemap with the Tiles before rotating it or you can first rotate it and then start working.

Keep in mind that areas will not rotate so draw them with the settings you want to use.

It also adds three new options:

- [Navigation Tab] **<Show line degrees>** It shows line angle while you are drawing. Angle is shown on 360° and every 90°, useful for quickly finding angles like 30/45/60 without having to subtract from 360.  
Values are shown in the inspector but it is possible to enable a summary label which will follow the mouse (**<Near mouse>**) or will be fixed at the Game View top (**<Fixed on top>**).  
Label is resizable with **<Label font size>**.
- [Navigation Tab] **<Keep line straight>** This option can only be used with Tilemap or environments set in orthographic projection. By selecting it you can draw a straight line horizontally or vertically. Mouse distance from the line can be set according to your needs through **<Line offset>**.
- [General options] **<Draw wire spot>** Draw semitransparent spots.

Manual does not show these new changes.

Watch [How to use Tailmap with 2.5D Toolkit](#) video which shows in a few minutes steps to follow.

## Introduction

---

### Relevant changes since version 1.05

- It is no longer required to enter Play Mode to create areas and place objects. Now whole process can be done directly in Edit Mode. This new feature will speed up work and allow to use zoom tool of Game View to better place objects.
- Whether you work in Orthographic or Perspective projection the right camera size or distance will be kept (this function can be disabled). This way your sprites clipped from the background image won't need to be resized.
- Each area can be deleted and converted from walkable to non-walkable and vice versa. Vertices can be moved, added, deleted both while you are drawing and once area has been closed.
- You can choose to work with a Plane as floor or use 2.5D Toolkit tool that doesn't require it. This tool cannot manage multilevel environments, if your scene requires floors at different heights you have to work with plans.
- If you work in Orthographic projection you can take advantage of the new tool to resize character to get an even more accurate overview of your scene.
- With the new and powerful Meter and Cube tools you can now set your environment even more realistic.
- Scene building always takes place starting from the coordinate (0, 0, 0). Once work is complete, you can move whole environment wherever you want.
- ... and many other options that will make your job more enjoyable.



Gizmos enabled in game view is required

2.5D Toolkit allows you to easily and quickly create and manage 2.5D environments for your games (it can even be used just to create meshes of any shape and complexity).

Use 2D backgrounds with sprites and 3D objects to create adventures like The Longest Journey, Broken Sword, Syberia and other great games.

Draw your navigation areas without the slightest effort by switching from Game View to Scene View and vice versa according to your needs. Draw as many areas (walkable and non-walkable) as you want. Bake your navmesh or create and save a mesh from it. You can use multiple plans to build a multi-level environment to be used directly in your scene or create meshes you can save and import later.

Place 2D and 3D objects in Game View exactly where you want them to be. Move, scale and rotate them thus easily obtaining environment you want to build. Build your scene just how you want others to see it.

### **What You See is What You Get.**

You can work in Scene View and manage everything yourself or use the powerful tools provided by 2.5D Toolkit.

You can choose to work with a background image in scene or use a Background Camera.

Both perspective and orthographic cameras are supported.

2.5D Toolkit is able to calculate the right camera size or distance so that your sprites will not need to be resized making your work even easier.

Once completed, whole environment can be moved anywhere in 3d space.

You may export in a file all objects data to create a new project from scratch.

If you encounter problems while developing an environment and you don't know how to proceed contact us.



Please read manual before you start using 2.5D Toolkit. You will find useful information and tips for best result with minimum effort.

# Watch videos

---

## YouTube channel

Version higher than 2.0

- [v2.03 - 2.5D Toolkit: A way to place sprite with just a click in one second](#)
- [v2.03 - 2.5D Toolkit: Improved Meter](#)
- [v2.02 - 2.5D Toolkit: How to use Tailmap](#)
- [v2.01 - 2.5D Toolkit: A way to use Meter and Cube](#)
- [v2.0 - 2.5D Toolkit: How to place 2D and 3D objects](#)
- [v2.0 - 2.5D Toolkit; Demo overview](#)
- [v2.0 - 2.5D Toolkit: Tutorial](#)

Version up to 1.05

- [Up to v1.05 - 2.5D Toolkit: Recreating a short Broken Sword scene](#)
- [Up to v1.05 - 2.5D Toolkit: How and why to use more meshes in one scene](#)
- [Up to v1.05 - 2.5D Toolkit: Multiple walkable areas](#)

Note: 2.5D Toolkit is continuously updated and improved. Some options have been removed, others added or moved. Older videos might show different profiles from the current ones. It is still recommended to watch them.

Take time with them before you start working, you will thus be able to create great environments for your 3D characters.

# How to use

---



Required: Scene name, Main Camera, Character and \*Floor

\*just if you don't use the 2.5D Toolkit Environment tool or Tilemap

1. If you have not done it yet, Import **2.5D Toolkit**. Create a new scene. Click [MainMenu/Tools/VB Game Studio/2.5D Toolkit]. Two new GameObject will be created into Hierarchy, VB25DTK and VBAreaTK.
2. Save scene. Scene name is required as 2.5D Toolkit will use a folder with this name to save some data.
3. Place side by side Scene View and Game View and enable Gizmos in Game View.
4. To create meshes of any shape and complexity, [see this example](#).
5. Background image:
  - 5.1.) Select your background image and set its Texture Type field to Sprite (2D and UI).
  - 5.2.) If you work with image in scene, drag it to the Hierarchy and reset it to (0, 0, 0). By dragging your image into [Tab Settings] <Use image in scene> field a popup will suggest you to set a correct Aspect Ratio in Game View (do not use "Fixed Resolution" for images that are a bit large as the gizmos line may not be clear enough) and asks if it should set right camera size or distance (according to the camera projection). Two new buttons will be enabled: <Reset to initial setting> and <Apply camera ortographic size> or <Apply camera distance>. While searching for the correct camera setting this tool will center and rotate the background image taking into account the new parameters. In Orthographic projection it will assign the right camera size, in Perspective projection the correct distance (according to FOV setting). Your background image should never be moved (whole environment can be moved at the end of the work). Of course you can ignore this feature and manage everything yourself.
  - 5.3.) If you use Background Camera: click [Tab Settings] <Create background camera> to create new camera called VBBGCamera. A new layer "VBBackground" will be created. Note that it will not be removed if you

delete the Background Camera (some settings of the Main Camera will be changed and they will be restored if camera background is deleted through 2.5D Toolkit). Drag background image into its field. A popup will suggest you to set a correct Aspect Ratio in Game View (do not use “Fixed Resolution” for images that are a bit large as the gizmos line may not be clear enough) and asks if it should set right camera size or distance (according to the camera projection). Two new buttons will be enabled: **<Reset to initial setting>** and **<Apply camera orthographic size>** or **<Apply camera distance>**. While searching for the correct camera setting this tool will center and rotate the background image taking into account the new parameters. In Orthographic projection it will assign the right camera size, in Perspective projection the correct distance (according to FOV setting). Of course you can ignore this feature and manage everything yourself.

6. Main Camera and character are required to unlock 2.5D Toolkit features. Drag the character (you can use your own or just for your test what you'll find in Demo/Resources folder) into Hierarchy then into its field. Drag Main Camera into its field. Floor (Plane in scene) is only required if you don't use [Tab 2.5DTK Env.] **<Use 2.5D Toolkit environment>** tool (works with Background Camera). This tool cannot manage multilevel environments, if your scene requires multiple floors at different heights you have to work with Plans.
7. For all your 2D images: Select them one by one in Project and in the Inspector change Texture Type field from **Default** to **Sprite (2D and UI)**. Set Pivot on Bottom (do not set it for background image leaving it on Center). Check Read/Write Enabled. Apply change. **Note: If you want to use <Auto position> option pivot must be set to Center.**
8. Next steps will depend on the way you work. In general, building a 2.5D environment involves placing one or more Plans in front of the background image (you may drag on them the Glass material available in the Materials folder to get a semi-transparent). They must be moved and resized to always cover the entire walkable area of your background image taking into account camera settings. Camera settings in turn depend on the image and perspective the character must have in the scene. It is therefore essential, when real dimensions are not available, to be able to evaluate at least some sizes of the image and using the very useful **Meter** and **Cube** tools you can recreate an environment proportionate with background image.

9. When environment is ready you can build navigation areas and place 2D and 3D objects.

**Note:**

Vertices and edges of areas must not touch each other. You can use zoom in Game View to get as close as possible (when needed) but without overlapping them.

If areas are structured on several overlapping levels, they may not be assembled correctly. In this case it is possible to draw more than one mesh in a scene.

Watch [2.5D Toolkit: how and why to use more meshes in one scene](#) video.

Please read manual before you start using 2.5D Toolkit. You will find useful information and tips for best result with minimum effort. Many tools (e.g. Meter, Cube) have been developed to help create a realistic environment starting from a 2D image. Knowing how to use them is essential.

If you don't have time to read manual at least take a look at the [tips](#) section. You will find a summary of the most important functions to know.

Give all your GameObject different names as some tools move, delete, recreate objects using Parent property.

Now you can start working.

**Note:**

When some options are enabled it will not be possible to change tabs.

[tab Navigation]

- *Current state* in Editing mode
- *A selected area vertex*

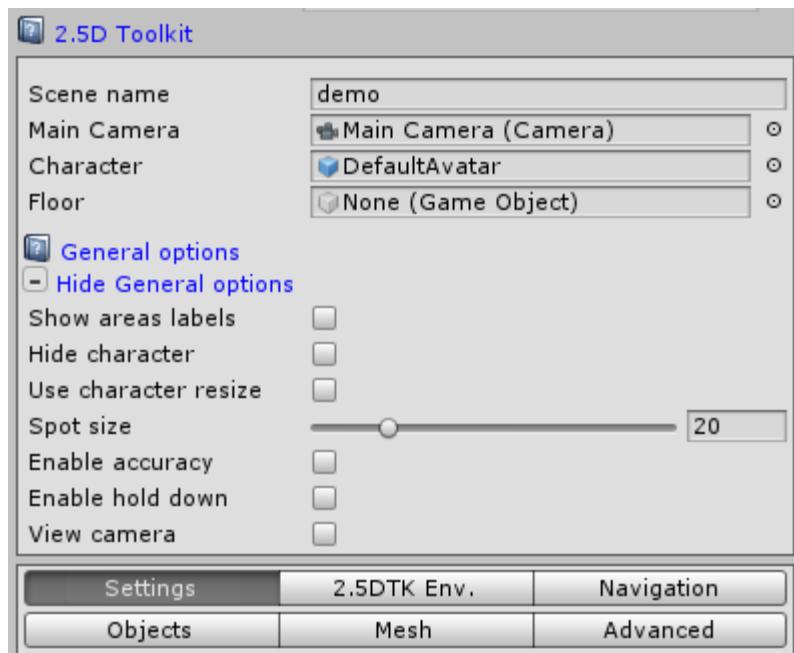
[tab Objects]

- **<Object>** field not empty

[Click Help icon to view a summary of each available function](#)

# General options

Shared options always accessible.



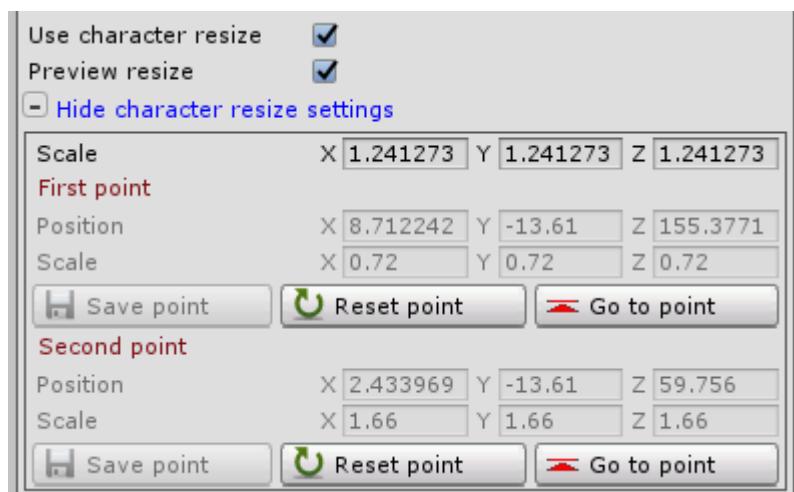
## General options

**<Hide General options>** Hide/Show whole options area.

**<Show areas labels>** Hide/Show labels. Useful for finding area number when drawing contains many of them.

**<Hide character>** Hide/Show character.

**<Use character resize>** Available in Orthographic projection.



It allows to assign two different scales to the character at two different coordinates in 3D space so that it resizes itself according to these parameters.

Place the character at a point in the scene where you can take a good reference (e.g. a door, a window, a table), scale it and save the point. Find another point further away and repeat these steps.

As can be seen from the screenshot, a scale of 0.72 was assigned at the furthest point and 1.66 at the closest point. About halfway the character is scaled to 1.24.

Script also takes care of changing character speed when it moves away from camera.

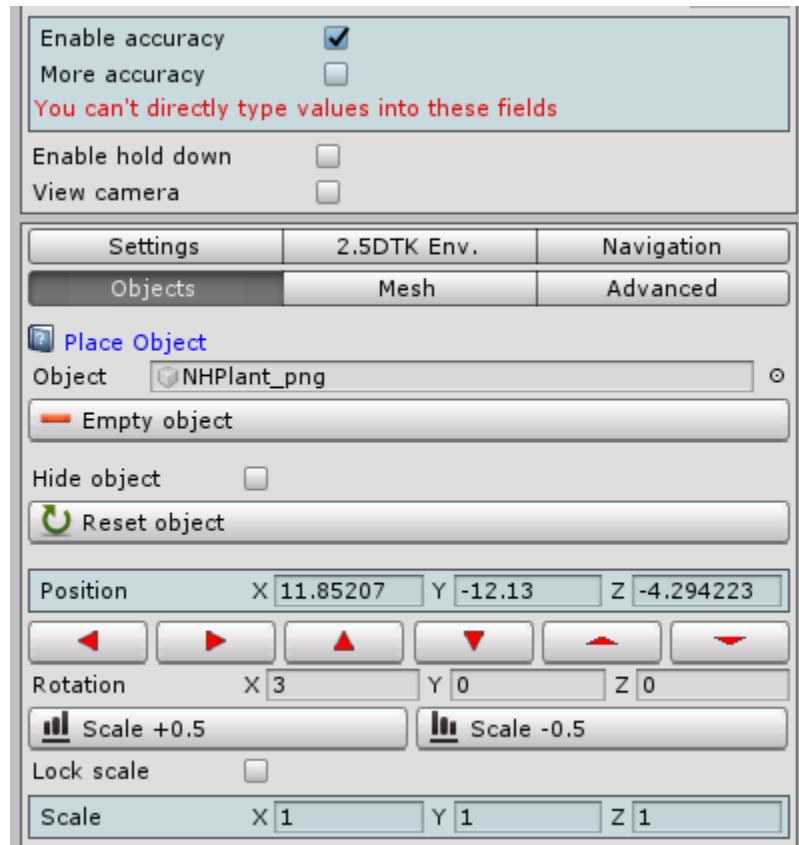
This tool only accepts two parameters as it is intended to provide an overview of the scene. Several assets for creating adventures offer more complete features.

**<Preview resize>** Working with the advanced tools of 2.5D Toolkit: while moving the character with the mouse it will automatically be resized to get a preview.

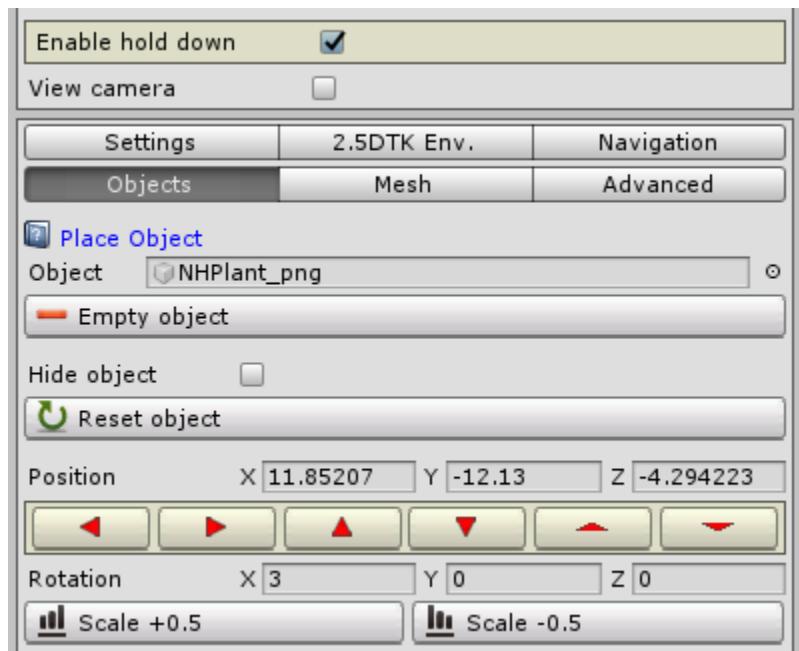
**<Hide character resize settings>** Hide/Show whole resize area.

**<Spot size>** Adjusts spots (vertices) size of the areas.

**<Enable accuracy>** Available for some fields.



By enabling this option fields involved will be highlighted. Settings increase in precision. **<More accuracy>** Improve further. With this feature enabled it will not be possible to type directly in the fields. It is very useful especially for placing sprites.  
**<Enable hold down>** Available for some buttons.

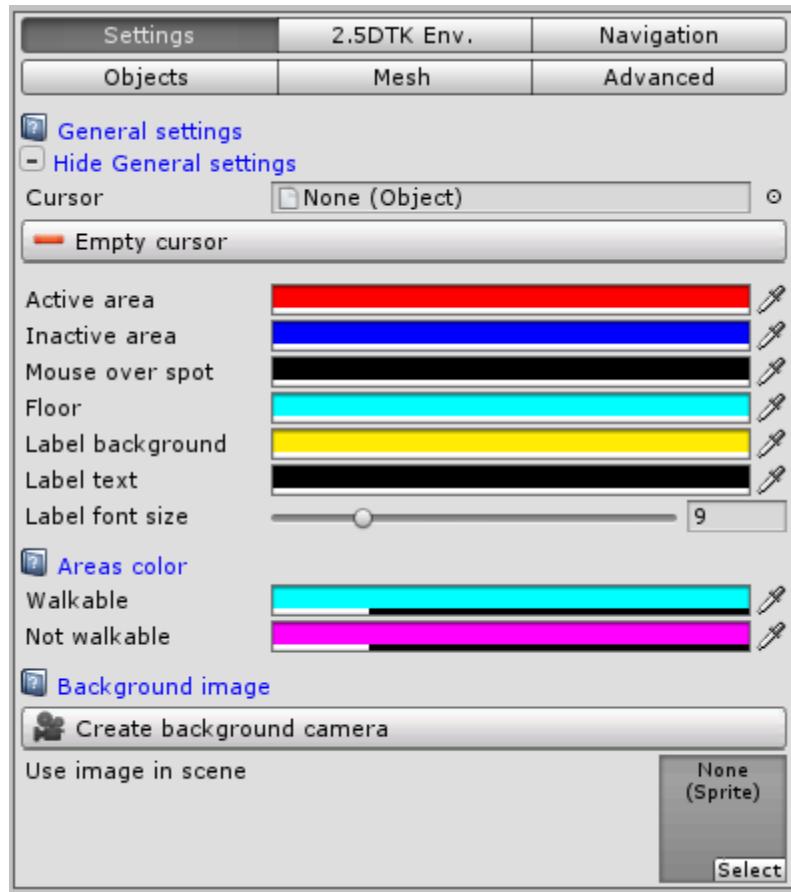


By enabling this option fields involved will be highlighted. You can hold the button down instead of pressing it repeatedly to moving an object.

**<View camera>** Hide/Show camera frustum.

# Settings

First 2.5D Toolkit settings.



## Gizmos setting

**<Cursor field>** You can drag here one of the cursors that you will find in the 2.5dToolKit/Cursor folder or use the default one leaving the field empty.

**<Empty cursor>** Empty **<Cursor field>**.

**<Active area>** Line color during drawing or a selected area. Each new area will be marked with labels which can be shown or hidden.

**<Inactive area>** Area not selected.

**<Mouse over spot>** Spot color on mouse over.

**<Floor>** Floor color.

With **<Label background>**, **<Label text>** and **<Label font size>** you can change labels appearance.

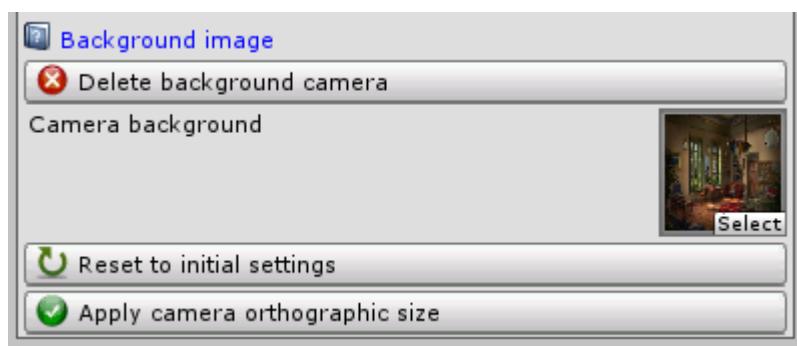
## Areas color

Area color can be set at any time with **<Walkable>** and **<Not walkable>**, before, during and after it has been drawn. If the color is changed make sure to set Alpha channel to about 20 percent to maintain a good overall view of background image.

## Background image

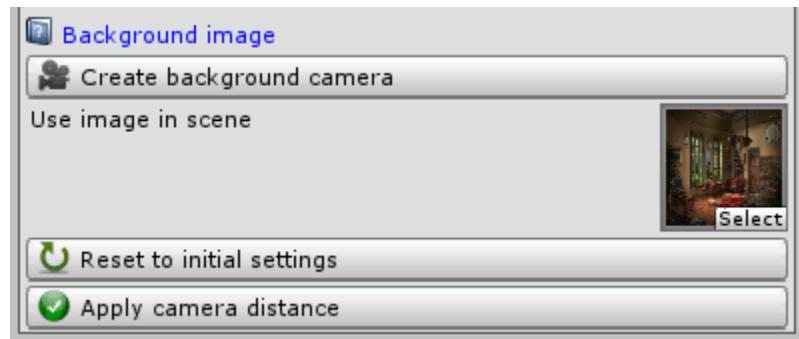
You can create a second camera that renders the background.

Click **<Create background camera>** to create new camera called VBBGCamera. A new layer “VBBackground” will be created. Note that it will not be removed if you delete the Background Camera (some settings of the Main Camera will be changed and they will be restored if the camera background is deleted through 2.5D Toolkit). Drag the background image into its field (set its Texture Type field in **Sprite (2D and UI)**), a popup will suggest you to set a correct Aspect Ratio in Game View (do not use “Fixed Resolution” for images that are a bit large as the gizmos line to draw your mesh may not be clear enough) and asks if he should set right camera size or distance (according to the camera projection). Two new buttons will be enabled: **<Reset to initial setting>** and **<Apply camera orthographic size>** or **<Apply camera distance>**. While searching for the correct camera setting this tool will center and rotate the background image taking into account the new parameters. In Orthographic projection it will assign the right camera size, in Perspective projection the correct distance (according to FOV setting). Of course you can ignore this feature and manage everything yourself.



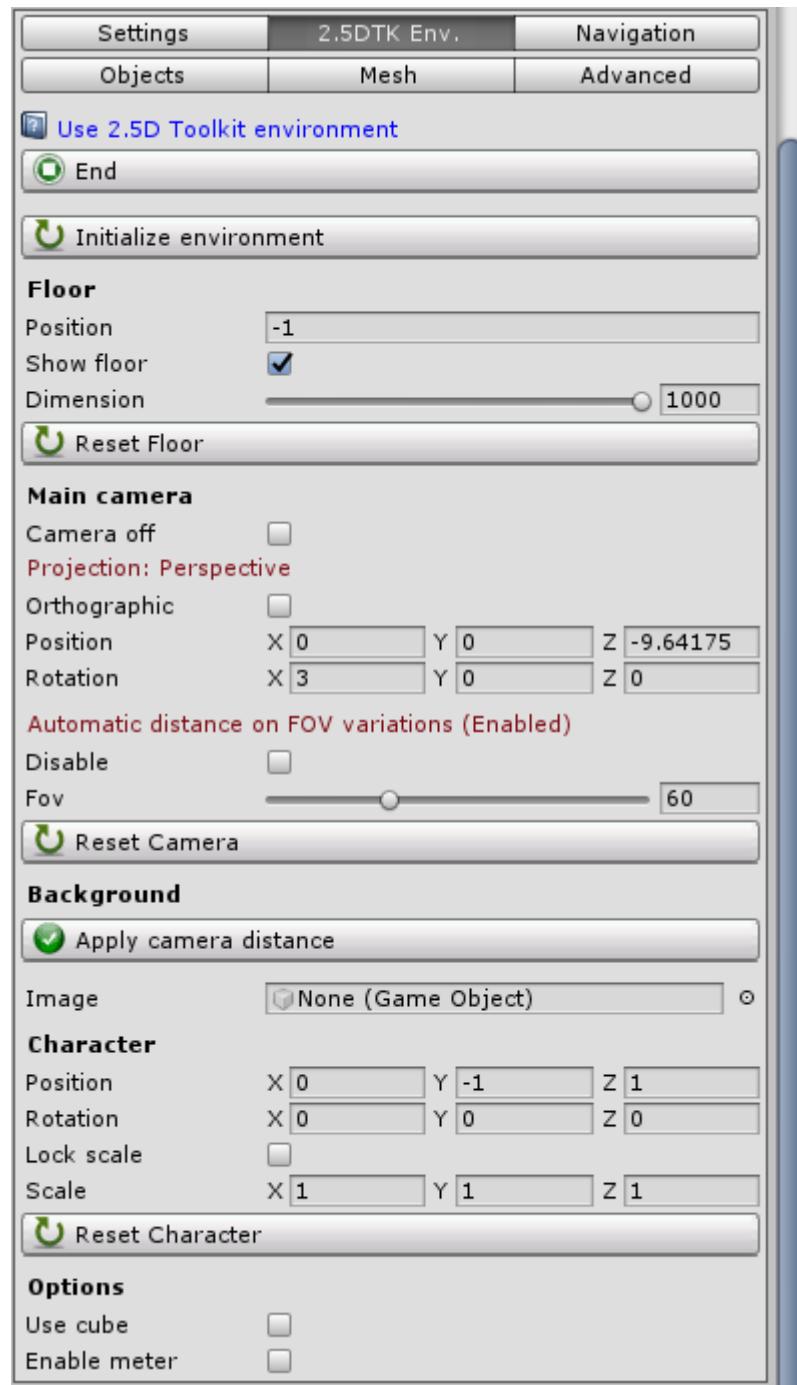
**<Use image in scene>** By dragging your image into this field a popup will suggest you to set a correct Aspect Ratio in Game View (do not use “Fixed Resolution” for images that are a bit large as the gizmos line to draw your mesh may not be clear enough) and asks if he should set right camera size or distance (according to the

camera projection). Two new buttons will be enabled: **Reset to initial setting** and **<Apply camera orthographic size>** or **<Apply camera distance>**. While searching for the correct camera setting this tool will center and rotate the background image taking into account the new parameters. In Orthographic projection it will assign the right camera size, in Perspective projection the correct distance (according to FOV setting). Your background image should never be moved. Of course you can ignore this feature and manage everything yourself.



## 2.5DTK Env.

Allows you to build environment without the need to use a Plane in Scene View.



It cannot manage multilevel environments, if your scene requires floors at different altitude you have to work with plans.



Background Camera is required. No Plane needed.

**<Start/End work>** Enable/disable this tool. Caution: if areas have been created they will be deleted by disabling the tool.

**<Initialize environment>** Reset the environment to its initial working state. Caution: if areas have been created they will be deleted.

## Floor

**<Position>** Being the work floor infinite it will be useful only to move it along the Y axis. The further it is from the camera the more scene width increases (character get smaller) and vice versa. Using the [Meter](#) tool and referring to hypothesized or known dimensions, we can set the right distance. Keep in mind: by changing camera rotation previous measurements will change, always repeat them after each new setting.

**<Show floor>** Hide>Show floor.

**<Dimension>** Change the number of squares displayed. Floor dimension is always infinite.

**<Reset floor>** Set floor to -1 on the Y axis.

## Main Camera

**<Camera off>** Turn off/on camera.

**<Projection>** Show camera projection.

**<Orthographic>** Change camera projection. Caution: reset the environment to its initial working state. If areas have been created they will be deleted.

**<Position>** Set camera position. Note: In perspective projection it will not be useful to move camera on Z axis as the distance is automatically applied by setting FOV or by clicking **<Apply camera distance>** button. However, it is possible to disable the automatic distance setting dependent on the FOV and manage these settings yourself (do not click anymore **<Apply camera distance>** button).

**<Rotation>** Set camera rotation.

**<Automatic distance on FOV variations (Enabled/Disabled)>** Shows whether the setting is turned on or off.

**<Disable>** Enable/Disable automatic camera distance setting dependent on the FOV.

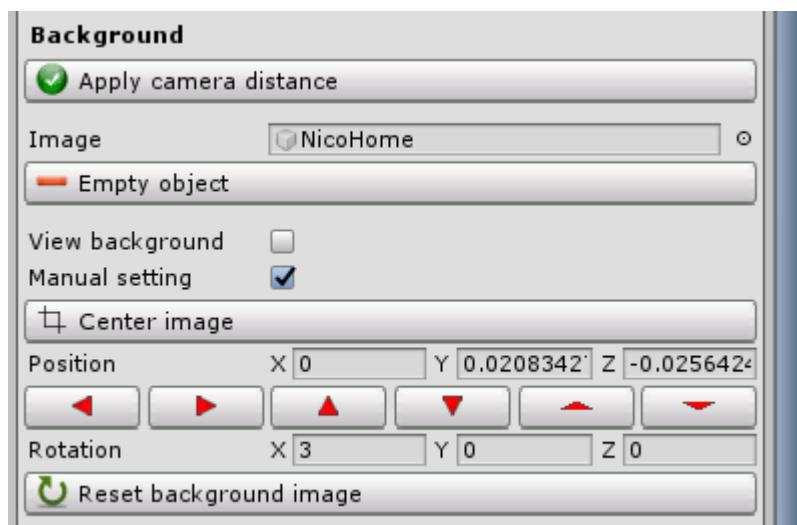
**<FOV>** Set FOV. If the automatic distance setting option is not disabled camera's Z axis will automatically change based on this value.

**<Reset Camera>** Set camera position to (0, 1, -10) and Rotation to (3, 0, 0).

## Background

According to the projection of the camera **<Apply camera distance>** or **<Apply camera size>**. Generally they are not needed as these values are automatically applied by the 2.5D Toolkit. If changes have been made in Scene View by mistake, you can use these buttons to restore the correct values.

**<Image>** This field will probably never be needed. By dragging the image used for the Background Camera in the Hierarchy and then in this field it will be possible to select **<Manual setting>** which will show a subsection with which you can manage the background image manually. This portion of the code was helpful during development and we thought to leave it for anyone who wants to experiment and modify source.



## Character

**<Position>** Set character position.

**<Rotation>** Set character rotation.

**<Lock scale>** Enable to change scale on three axes simultaneously

**<Scale>** Set character scale.

**<Reset character>** Set character position to (0, 0, 0), Rotation to (0, 0, 0) and Scale to (1, 1, 1).

## Options

**<Use cube>** Create a Cube with colored sides in the scene (disabling option the Cube will be deleted). [Using Cube is useful for defining the amount of camera rotation.](#) Character is needful for estimate scene depth but often it does not help to find the right rotation. By placing the Cube near an object in our background image, we will be able to evaluate camera rotation more precisely by comparing the perspective of the two objects. If sides colors get confused with the background image you can change them from Resources/CubeUI/Materials.

**<Place cube>** Move Cube with left mouse button. Since the same button is used by the Meter tool, activating one disables the other and vice versa.

**<Transparent cube>** Makes cube semi-transparent to better overlay it on objects.

[It can be used to try to guess FOV and camera rotation.](#)

**<Don't keep proportion>** Useful in perspective projection.

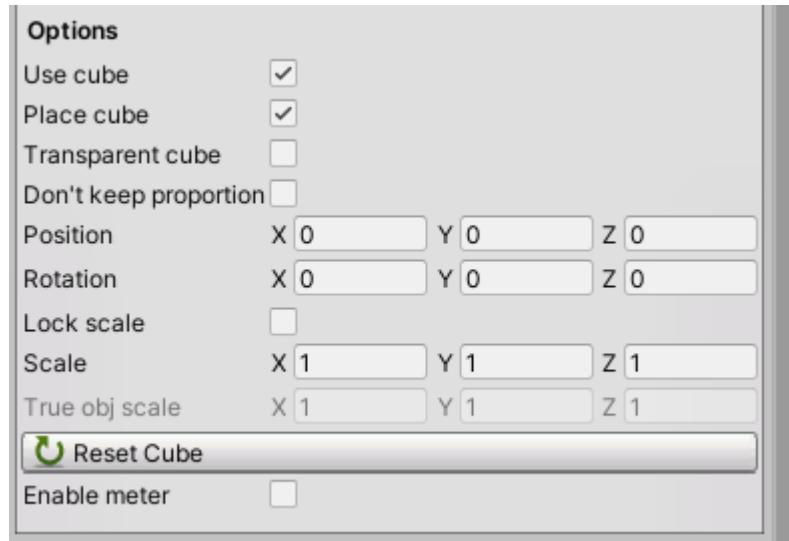
**<Position>** Move cube. If needed it can be moved on the Y axis but by clicking with the mouse will bring it back to the floor.

**<Rotation>** Set Cube rotation. Assigning to cube the same rotation of the object you want to compare helps in the evaluation.

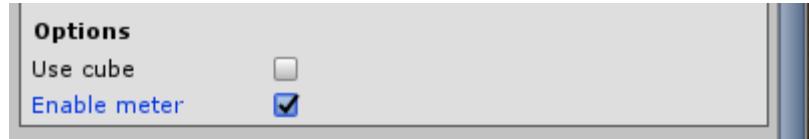
**<Lock scale>** Change Cube size on the three axes.

**<Scale>** [It is often useful to give cube a flat shape in order to be able to compare it with the floor.](#) So you can define camera rotation based on it.

**<Reset Cube>** Set cube to scale (1, 1, 1).



**<Enable meter>** Very powerful tool. With it you can finally find the setting parameters closest to the real ones. It should only be used to measure the floor or objects on it (measuring at floor level).

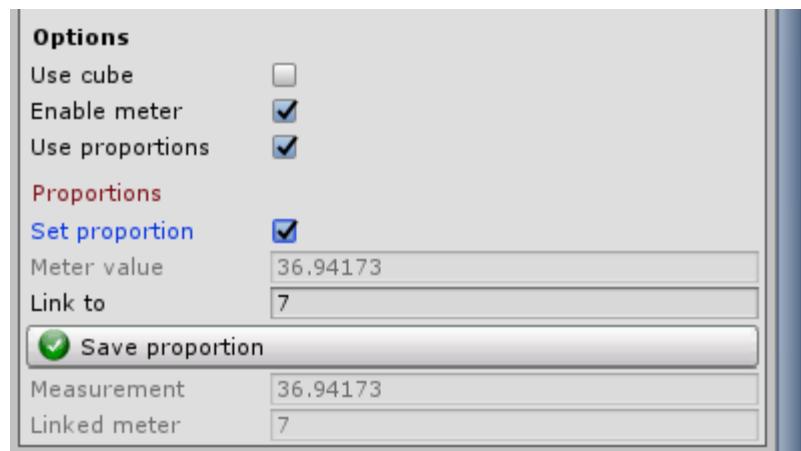


Steps to follow:

1. Look carefully at your background image. If you already have the real measurements you can skip some steps.
2. How do you see image environment? Square, longer than wide, wider than long?
3. How wide can your scene be? And how long? Use the objects in the image to evaluate possible measures.
4. Once you have defined the dimensions you can start to create scene.
5. Start by measuring the width. Enable Meter tool, click (in Game View) in the leftmost part and then in the rightmost part. The measurement will be printed in the Console (we are working in meters).
6. Change Floor position until you read a measurement close to the established one.
7. Measure the depth now. Change camera rotation to find the desired measurement.
8. Keep in mind that changing one will result in a change in the other. Work on both to obtain values close to the defined ones.
9. When you think you have found the best settings you can do a further test: take an object in your image to which you are able to give a dimension (a door, a chair, a car). If your settings are correct this object will have measures very close to the real ones. Remember, to be reliable, measurements must always be made at floor level.
10. If you work in perspective projection by assigning a right scale to your character at a given point you can verify that moving it in the scene will maintain correct proportions.

11. When your scene is ready you can draw walkable and non-walkable areas and place objects.

If you work in Orthographic projection Meter tool would need to be set up. You can use it as it is but it is advisable to configure it.



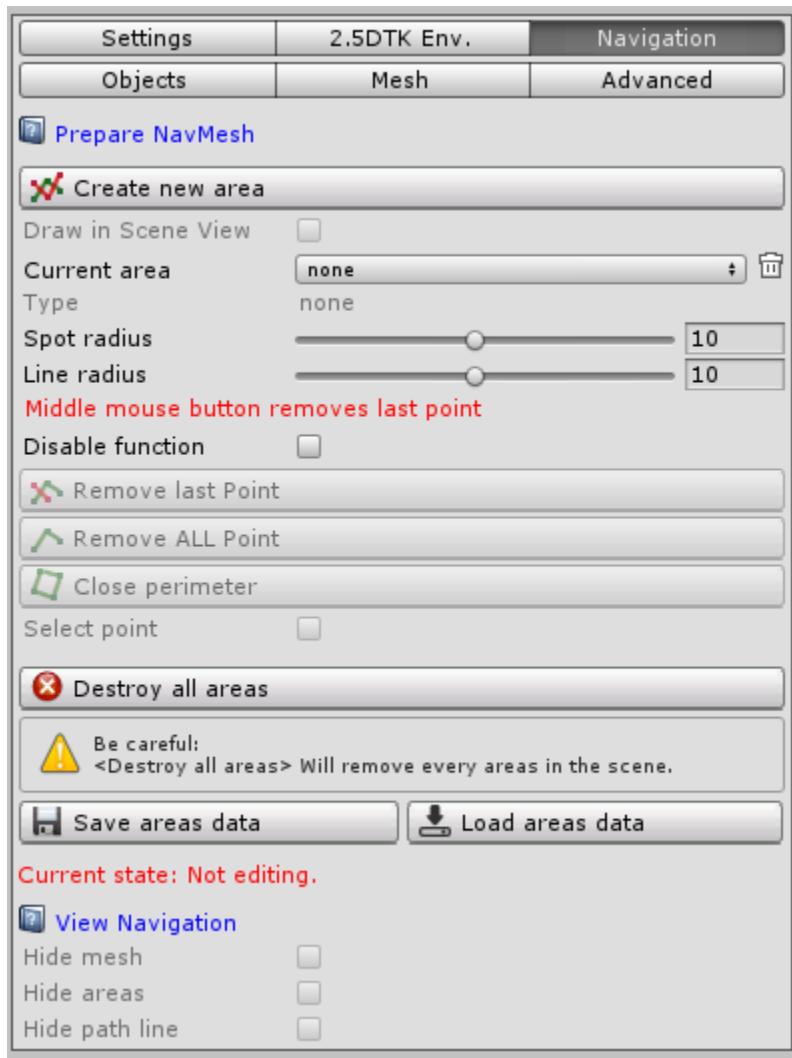
Click **<Set proportion>**, click (in Game View) in the leftmost part and then in the rightmost part. The measurement in meters will appear in the non-modifiable **<Meter value>** field. In the **<Link to>** field type the measure you want to correspond to the one just taken (e.g. 36.94 meters must correspond to 7 meters), finally click on **<Save proportion>**. From now on every measure taken will be redefined according to this proportion.



Meter tool: It should only be used to measure the floor or objects on it (measuring at floor level).

# Navigation

From this tab you can get your navigation environment.



## Prepare NavMesh

**<Create new area>** Initializes a new area to draw. You can create several separate walkable areas but each non-walkable area (holes) must be within a walkable area. It is advisable not to overlap the edges of the various areas.

Each area can be deleted by selecting it through **<Current area>** and clicking on the trash icon.

**<Draw in Scene View>** By enabling this option you will be able to draw directly in Scene View. Available only if **<Use 2.5D Toolkit environment>** option is not used.

This option is very useful in several cases.

If your environment is very "deep" you may have some trouble drawing some areas, e.g. small non-walkable areas. In this case, after placing your sprites in Game View, you can draw your areas around them in Scene View.

**Note:** Scene View does not support overlapping vertices to close areas. You will need to use **<Close perimeter>** button.

It could also happen that you have to draw in areas not visible to the camera.

Again this option will be convenient for you.

To move from Game View to Scene View and vice versa click the right mouse button to hide the line and click it again to grab the line when you are in the other scenario.

**Remark:** when this option is enabled the mouse buttons will be managed by the script while you are on the Plane.

So to change the view of Scene View make sure you are outside the Plane.

If the zoom allows you to see only the Plane you need to disable this option to move the view or use the middle mouse wheel to move the scene further away.

This option is enabled only in "Editing" state.

Watch the explanatory [2.5D Toolkit: how and why to use more meshes in one scene](#) video.

**<Current area>** Increments for each new area created and allows to select area to work on.

**<Type>** Define whether the area will be walkable or non-walkable. You can change area type at any time, even after it has been closed.

**<Spot radius>** When you hover over a vertex of the area it will change color. By clicking on it you can select/grab it according to what you are doing. You can change spots (vertices) size using **<Spot size>** in the **General Option** but spot sensitivity is independent of its size. If the spots are very close you may want to decrease the radius for greater precision (e.g. when both spots are selected simultaneously).

**<Line radius>** When you hover over an edge of the area it will become thicker. By right clicking you will insert a new vertex, clicking with the middle mouse button on it you will delete the spot. As with **<Spot radius>** you can set the sensitivity range.

**<Disable function>** If you use the Game View zoom to move around the scene you have to use the middle mouse button but at each click the last inserted vertex will be deleted. There are two possible functions to avoid it: select **<Current area>** to None or enable **<Disable function>** (**Middle mouse button doesn't removes last point**).

**<Remove last Point>** Delete last vertex entered in the selected area. **Middle mouse button allows the same operation**. If area was closed it will be deleted.

**<Remove ALL Point>** Delete all inserted vertices of selected area. If area was closed it will be deleted.

Clicking on **<Close perimeter>** area will be completed by connecting the last vertex inserted with the first one. Area can also be closed by clicking on the first vertex or by dragging the last spot on the first. It will be possible to close an area only after inserting at least three vertices.

**<Select point>** This option can be useful when due to the settings required by your scene you cannot drag with mouse a vertex exactly where you would like to place it. When you activate this tool, nothing else can be done until you close it (only **<Current area>** will be active, which if selected on zero will automatically close it). Click on the vertex you want to move.



You can now move it with the buttons in the four directions. You can jointly use **<Enable accuracy>** to increase precision and **<Enable hold down>** in the **General Option**.

With **<Confirm position>** and **<Restore position>** you will save or restore position.

**<Destroy all areas>** Remove all areas.

**<Save areas data>** You can save all created areas (e.g. if you want to try arranging areas differently but don't want to lose what has been done so far).

**<Load areas data>** Restores previously saved area configuration.

**Caution: only one configuration is supported. Saving again will overwrite the previous configuration.**

## View navigation

**<Hide mesh>** Hide/Show mesh (only the default one called VBMeshTK). If you have saved mesh with a different name it will not be hidden.

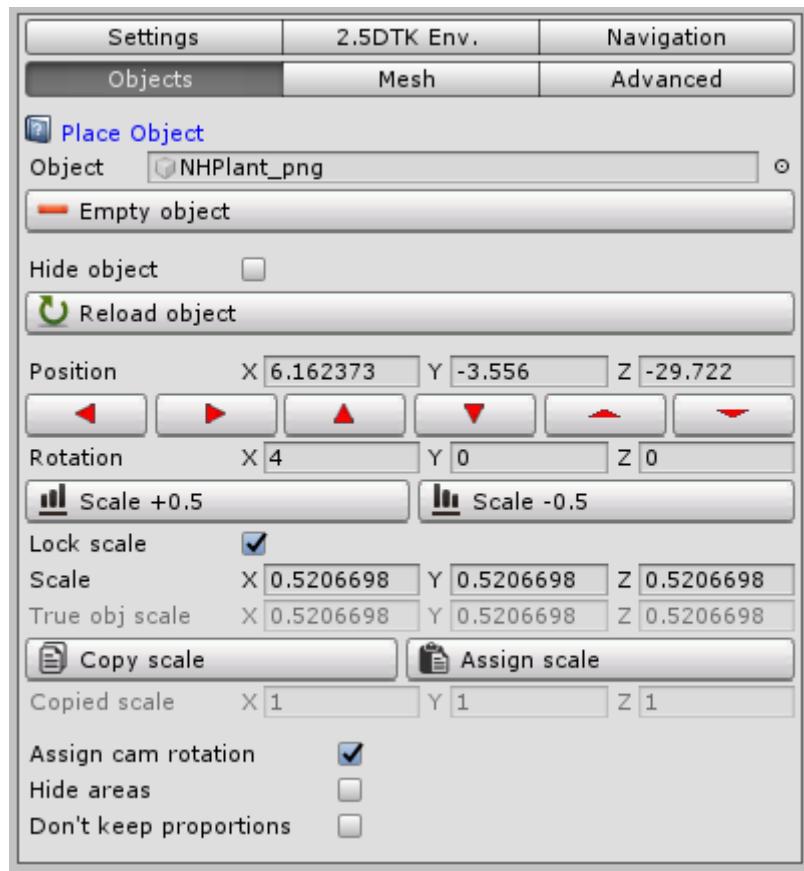
**<Hide areas>** Hide/Show all areas and Floor.

**<Hide path line>** Hide/Show character path line.

Note: available only in Play mode.

# Objects

Placing objects in a 3d space is a very demanding task that you can avoid by working on a 2D environment.



Often it will be enough to simply place the objects in Game View, in some cases it may be useful to have a look in Scene View.

## Place object

**<Object fied>** Drag from Hierarchy a new scene object here. It can be a 2D or 3D object.

**<Empty object>** Empty <<Object fied>>.

**<Hide object>** You can hide/show object. Especially useful with 2D objects to ensure it is perfectly overlapped to the background image from which they were cut out. Check and uncheck several times looking at the image carefully. If there are no differences sprite size/position is right.

**<Reload object>** Using this button you can delete 2D or 3D object from the Hierarchy and load it again from its folder with its original size. This is very useful when an object has been resized multiple times or a scale has been assigned more than once through **<Assign scale>** and it is difficult to give it the right size. Without having to drag the object manually it will be replaced with the original one.

With **<Position>** and **<Rotation>** you can move and rotate object.

Position buttons are critical. Through them you can place sprites with maximum precision. Use them in conjunction with **<Enable accuracy>** to increase precision and **<Enable hold down>** in the **General Option**.

**<Scale + 0.5>, <Scale - 0.5>** If object is very large or very small it may be useful to use this buttons to resize it more quickly.

**<Lock scale>** Allow you to scale object on the three axes simultaneously.

**<Scale>** Scale object. Z axis does not matter with 2D objects.

**<Copy scale>** If for some reason you are not using the automatic Size/Camera distance setting you may want to use a certain scale for all objects. Once you set your personal scale on an object copy it and assign it to all other objects with **<Assign scale>**. Caution: do not assign scale to the object more than once. If this happens, reload the object with **<Reload object>** to start again with its original size.

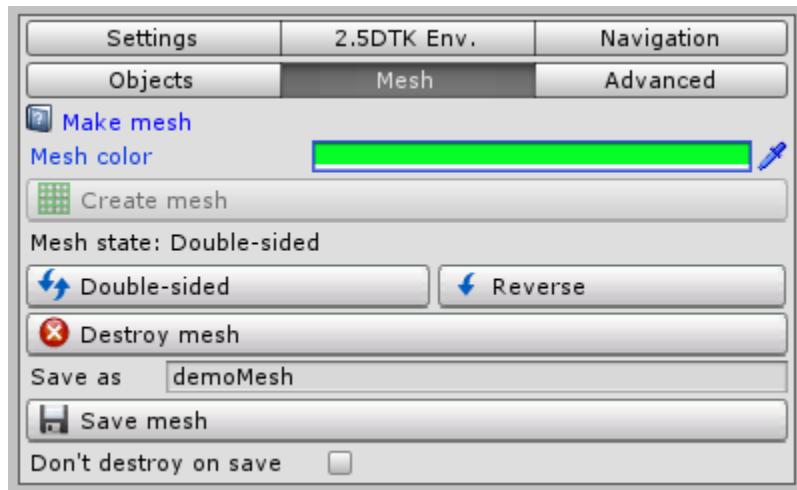
**<Assign cam rotation>** Assign camera rotation values to the object. Will be done automatically on 2D objects when you drag it into its field.

**<Hide areas>** Hides all areas/mesh to free view and better place objects. The mesh must be the default one called VBMeshTK.

**<Don't keep proportions>** Enabled in Perspective projection. Normally the object keep its proportions by moving it along the Z axis, selecting this option the object will change size according to the distance from the camera.

# Mesh

This option will be active if at least one walkable area has been created.



## Make mesh

**<Mesh color>** Assign a color to the mesh before it is created.

**<Create mesh>** Mesh will be created by default double-sided. With **<Double-sided>** and **<Reverse>** you can switch from one mode to another. If mesh is not visible in reverse mode simply click same button again.

**<Destroy mesh>** Clear mesh.

**<Save as>** You can name the mesh to save. By default it will be "(name scene)Mesh".

This is needed when you want to create more meshes in the same scene.

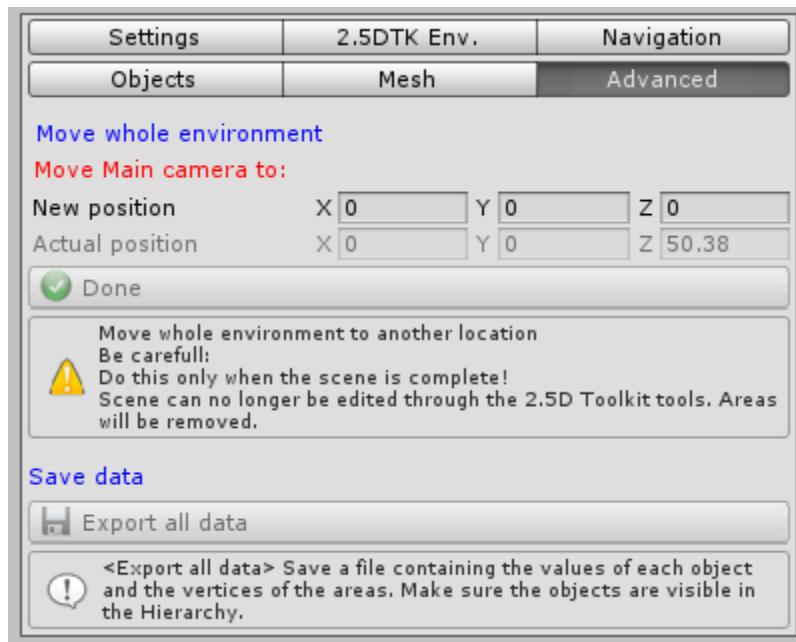
**<Save mesh>** Mesh will be deleted from Hierarchy and saved in 2.5dToolKit/Resources/[scene name] folder in two formats: .asset and .prefab. Follow [this tutorial](#) to get the shadows.

**<Don't destroy on save>** This option is used in conjunction with **<Save as>**. Normally when you save the mesh it will be removed from the Hierarchy. Checking this box the mesh will not be deleted. By keeping the mesh in the scene you can use it as a reference to design another area.

Watch the explanatory [2.5D Toolkit: how and why to use more meshes in one scene](#) video.

## Advanced

Move whole environment to another location. Save environment data.



### Move whole environment

Depending on whether you are using background image in scene or Background Camera you can define a new position in the 3D space for your environment by setting a new Main Camera or Background Image position. All Gameobjects in the Hierarchy will move accordingly. Make sure they are visible. If you want to avoid moving some objects you can hide them.

Caution: do this only when scene is complete. After that, you will no longer be able to edit scene with 2.5D Toolkit tools. Areas will be deleted.

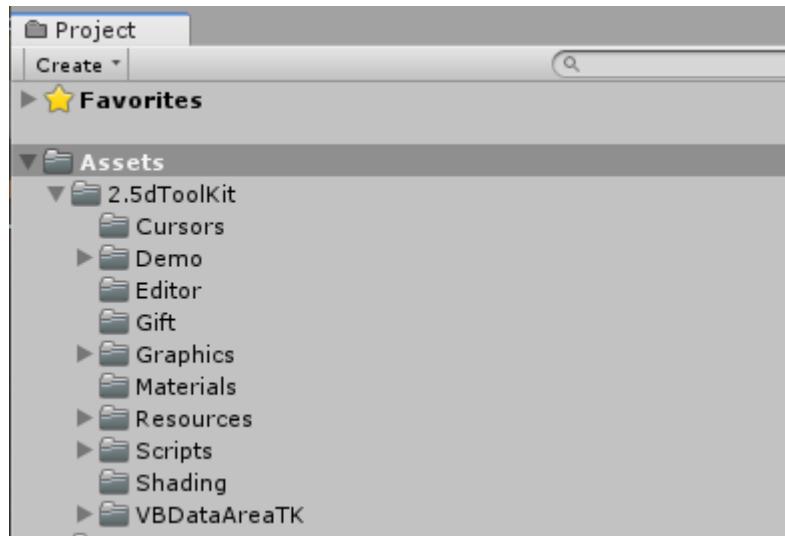
**Keep in mind that if you use a GameObject as a folder initially set to (0, 0, 0) its transform will be modified and the objects contained in it will be processed as children**

### Save data

**<Export all data>** Save a file containing the values of each object into Hierarchy and areas vertices. Make sure all objects are visible. You can use it to recreate scene in a new project.

## 2.5D Toolkit folders

---



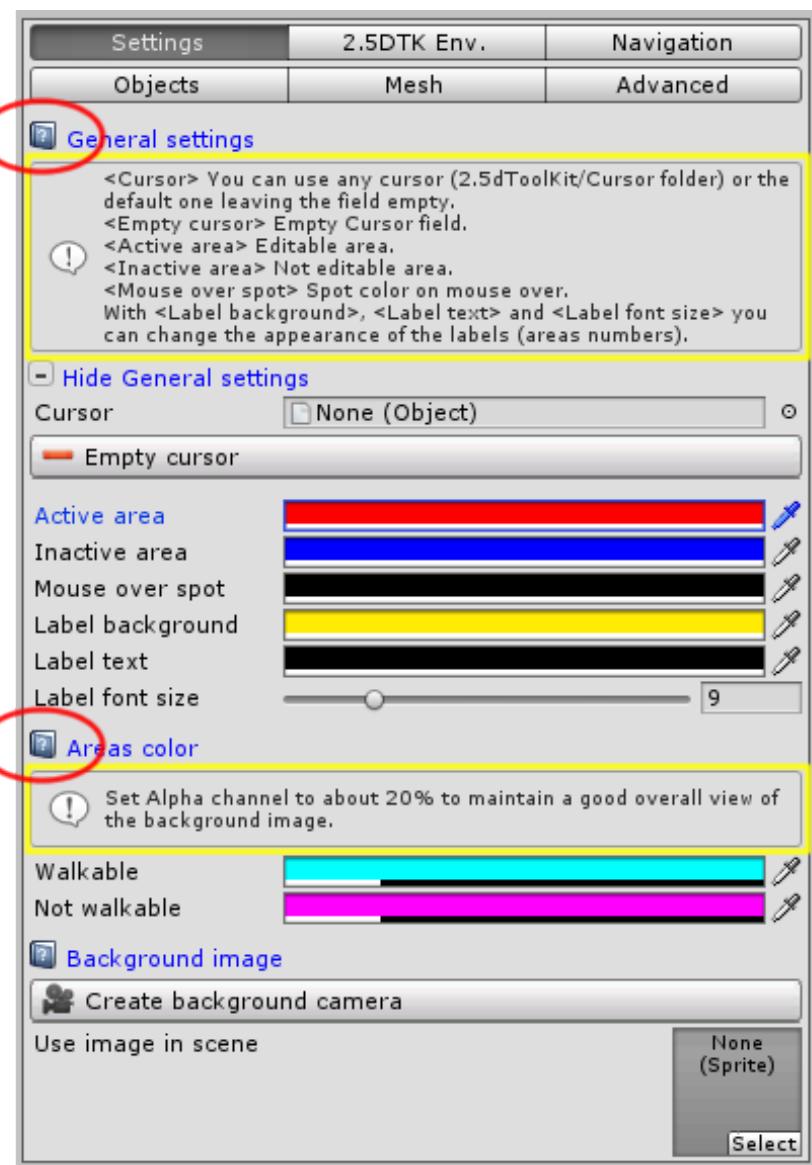
- *Cursors*] Some cursors to use for drawing.
- *Demo*] Several demos to learn about 2.5D Toolkit. It contains default character and its script.
- *Editor*] Editor scripts.
- *Gift*] Contains useful gifts for game development.
- *Graphics*] Icons used by the editor.
- *Materials*] Contains semi-transparent materials for Plans.
- *Resources*] Meshes will be saved in this folder. It contains Background Camera and Cube prefab.
- *Scripts*] 2.5D Toolkit scripts.
- *Shading*] Material and shaders to get shadows.
- *VBDataAreaTK*] Areas data will be saved in this folder.

# Tips

---

## Inline help

Click Help icon to show inline help.



# Mouse functions

Note: to draw you do not need to hold down left mouse button.

## Creating areas

- Left click
  - Drawing: insert a new vertex.
  - Over spot (vertex): move it / select it.
- Right click:
  - Drawing: show/hide tracking line.
  - Over area edge: insert a new vertex.
- Middle click:
  - Drawing: remove last inserted vertex, if area is closed it will be removed.
  - Over spot (vertex): remove spot.

## Creating environment

- Left click
  - [Tab 2.5DTK Env]: place Cube / use Meter.
  - [Tab Objects]: move object.
- Right click:
  - [Tab 2.5DTK Env]: move Character.
  - [Tab Objects]: move Character.

# General tips

## Objects

- For all 2D images cropped from background image: select them one by one in Project and in Inspector change Texture Type field from **Default** to **Sprite (2D and UI)**. Set Pivot on Bottom (do not set it for background image leaving it on Center). Check Read/Write Enabled. Apply change. Check that both background image and sprites have same **Pixels Per Unit** otherwise the sprites will have a different size.
- All 2D and 3D objects can be moved away from the floor by setting the Y value (e.g. if you want to hang a picture on the wall or place an object on a table).  
[Look at this example.](#)
- By repeatedly modifying an object (e.g. by changing its scale or using **<Assign scale>** more than once on the same object) it can happen that it is no longer possible to get its right dimensions. Rather than wasting a lot of time trying to restore it, click **<Reload object>** which will reload object with its original size.
- Whenever you place a object remember to click **<Empty object>**. If you casually click in Game View you will have to reposition it.

## Camera

- A 2.5D environment generally **does not require a high FOV**. Moreover, many images will need camera rotation which with a high FOV will contribute to character distortion on the screen sides. A good value can be around 20/25.
- **To reduce the depth of the scene**, increase the rotation of the camera (taking into account the environment where the character moves, you can use [Cube](#) to define a right camera rotation).
- Use automatic configuration of camera size/distance. This will allow to always have sprites (cut from the background image) of the right size and will not need to resize them.
- When using background image in scene with automatic camera size or distance, **never move or rotate image**. It will be handled by the script.  
You will then be able to move whole environment once scene is complete from [tab Advanced] **<Move whole environment>**.

## Scene

- Give all your GameObject different names as some tools move, delete, recreate objects using Parent property.
- If you use [Tab 2.5DTK Env.] **<Use 2.5D Toolkit environment>** use [Meter](#) to create a realistic environment and when needed [Cube](#) to define a right camera rotation.
- Once you have created your environment and assigning a walking speed to character always **compare the number of character steps with the distance he travels.**

Example: One step generally covers 65/80 centimeters. The character travels a distance of 10 meters with 15 steps. We divide 10 by 0.70 and we get around 14. It is fine. If the difference is minimal we can slightly adjust speed that will change steps number but if the difference is excessive we must consider reshaping the scene.

## Areas / Mesh

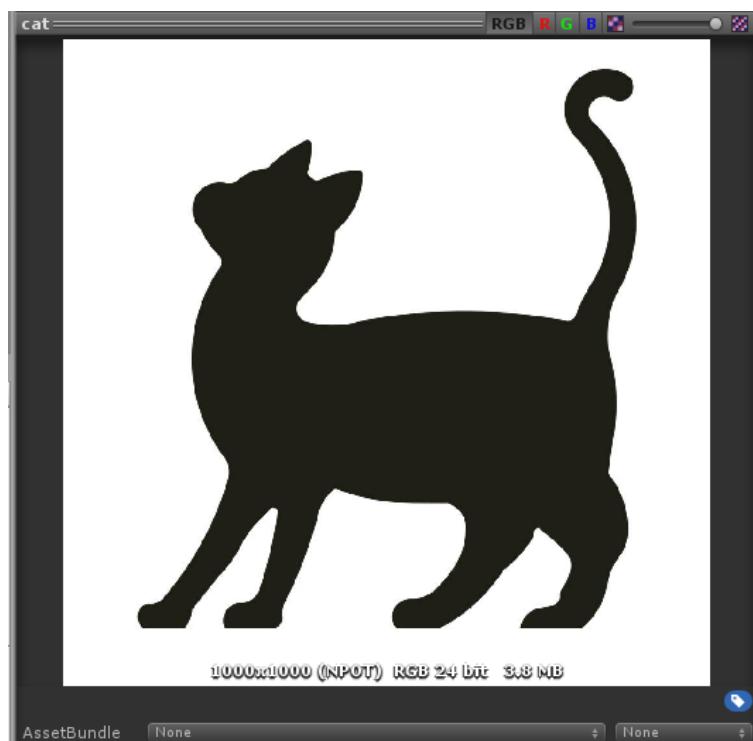
- When creating areas make sure vertex or edge of one **does not touch** those of another area.
- If you use the Game View zoom to move around the scene you have to use the middle mouse button but at each click the last inserted vertex will be deleted (is a default 2.5D Toolkit feature). There are two possible functions to avoid it: select **<Current area>** to None or enable **<Disable function>** (**Middle mouse button doesn't removes last point**).
- Always save created areas [Tab Navigation] **<Save areas data>** so you can reload them in case of need.
- Save new mesh before modifying its material otherwise Unity raises an error trying to save it later.

# A way to draw a simple mesh

---

You will find this example in Demo folder and in the first part of the [Tutorial Ver2.0](#) video.

We want to create a mesh from this image:



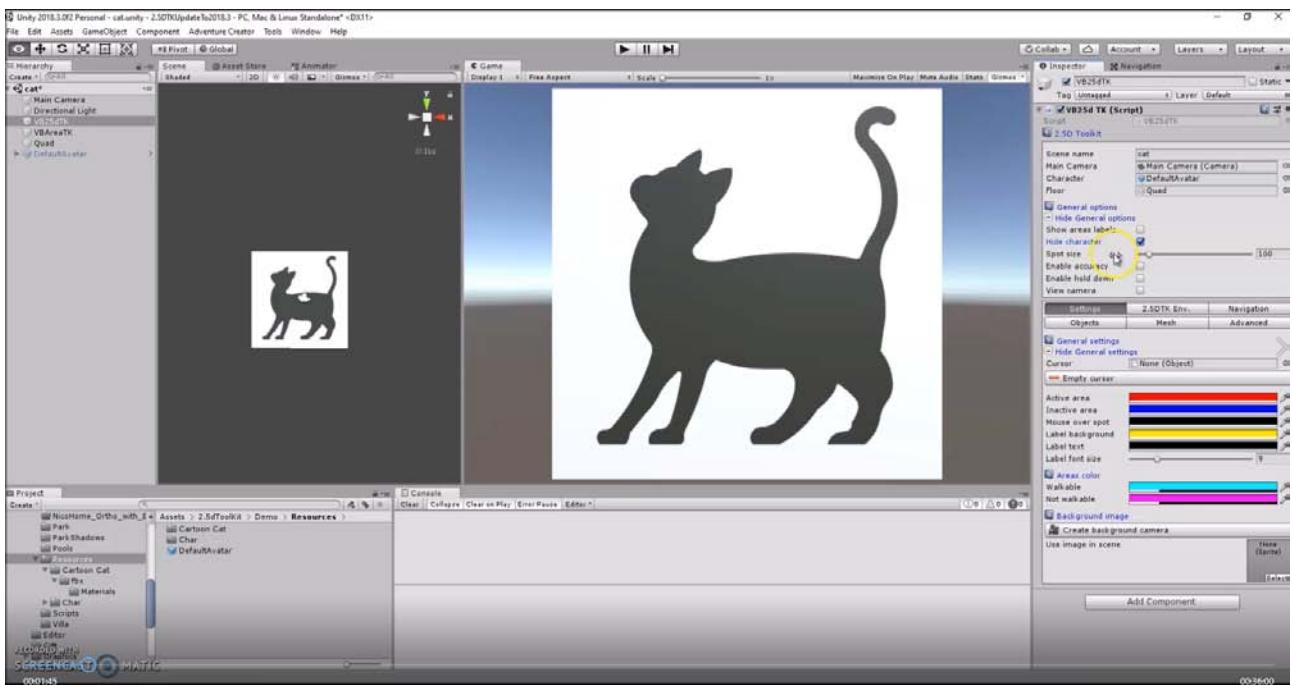
Create a new scene. Click [MainMenu/Tools/VB Game Studio/2.5D Toolkit]. Two new GameObject will be created into Hierarchy, VB25DTK and VBAreaTK.

Save scene with name. Scene name is required as 2.5D Toolkit will use a folder with this name to save some data.

Create a new Quad and drag on it image you want to convert to mesh. According to your needs you can set Aspect Ratio in Game View (1000:1000 for this image).

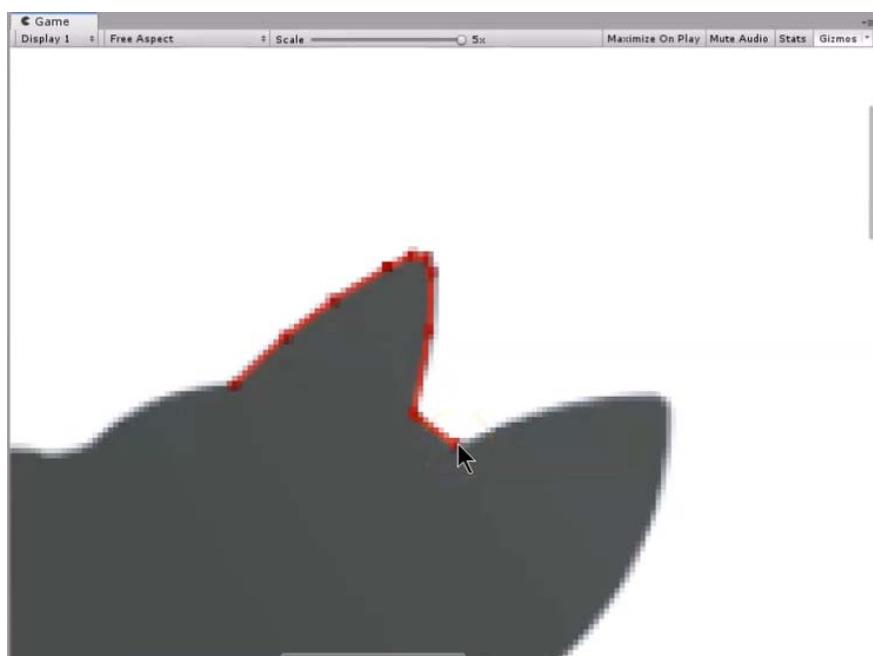
Drag Main Camera, Character and Quad in their fields. Even if you don't need it Character is required to unlock 2.5D Toolkit. Hide it with <Hide character> in the **General Option**.

Set Quad and Camera to position (0, 0, 0) then move camera away until your image fills Game View.



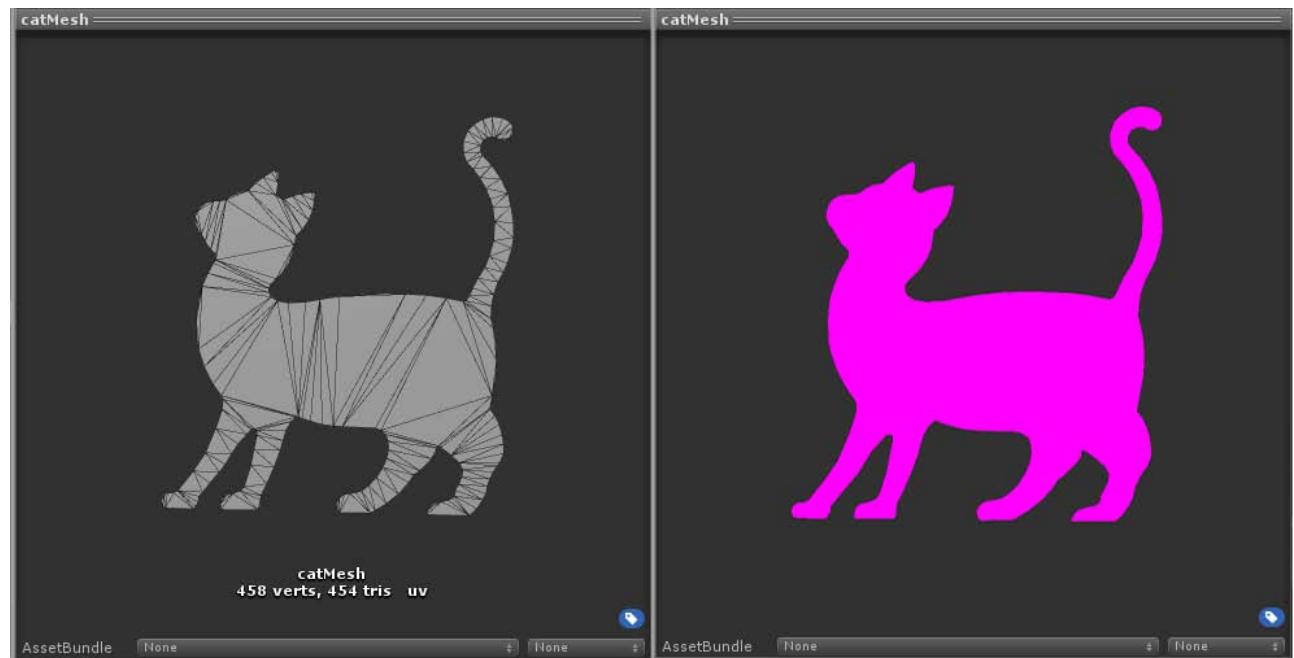
To increase the precision you could move the zoom to the maximum then move along the drawing with the middle mouse button. **Caution:** if you use the Game View zoom to move around the scene you have to use the middle mouse button but at each click the last inserted vertex will be deleted (is a default 2.5D Toolkit feature). There are two possible functions to avoid it: select **<Current area>** to None or enable **<Disable function>** (Middle mouse button removes last point).

Go to the Navigation Tab, click **<Create new area>** and start drawing.



Once finished (area must be closed) go to Mesh Tab, click <Create mesh> then <Save mesh>.

Mesh will be saved in Resources/[scenename] folder as .asset and .prefab.



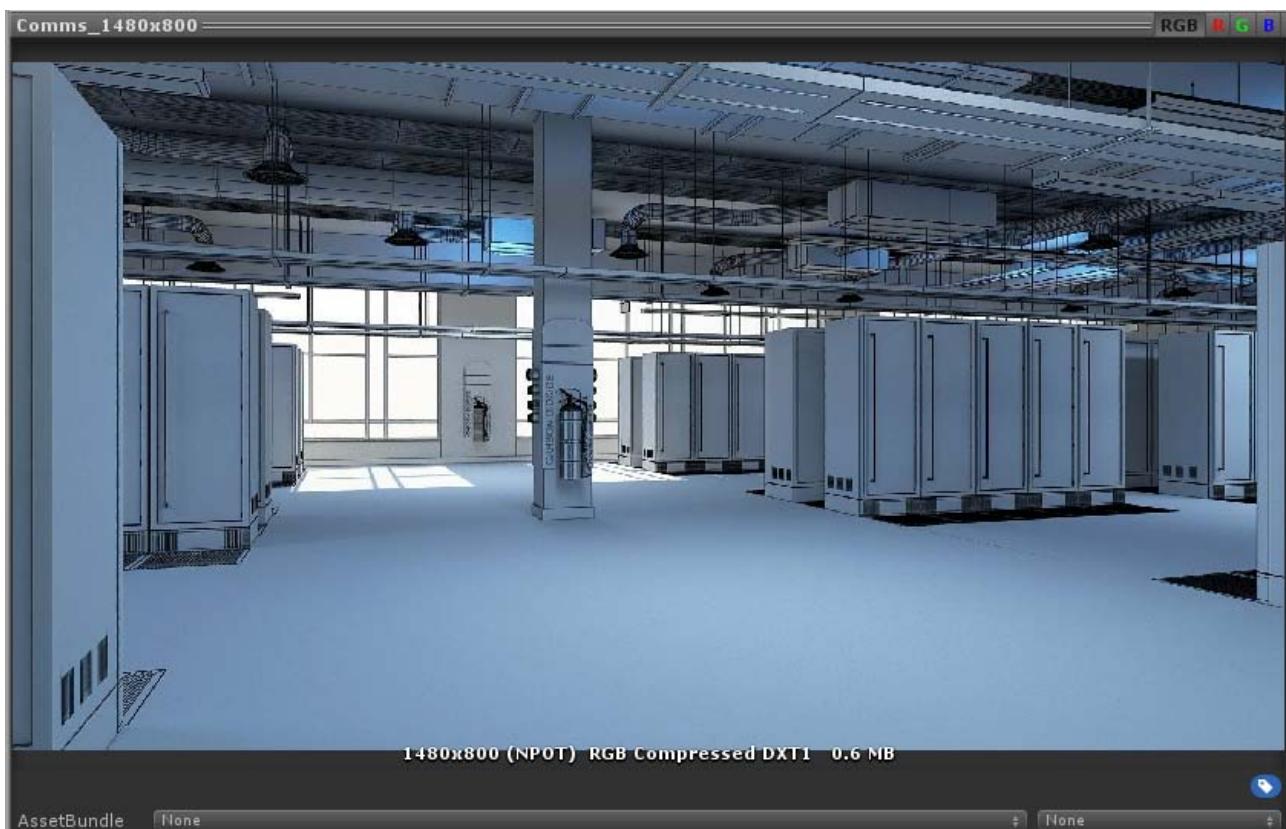
# Demo scene: Mainframe

---

You will find this example in Demo folder.

It uses 2.5D Toolkit environment tool which requires Background Camera.

Projection is Perspective, Camera distance is set as automatic, Aspect Ratio is set to 1480:800, Character walk: 1.7, Character run: 3.85.



Let's analyze the image:

- How do you see image environment? Square, longer than wide, wider than long?
- How wide can your scene be? And how long? Use the objects in the image to evaluate the possible measures.

The first thing we can notice is that probably won't need to rotate camera (a minimum rotation can also be useful to decrease the depth of a scene but as we will see this is not the case). Tool always initializes a rotation of 3. We set it to 0.

We leave the camera's Y value at 1, if needed we will change it later.

Let's decrease slightly the FOV.

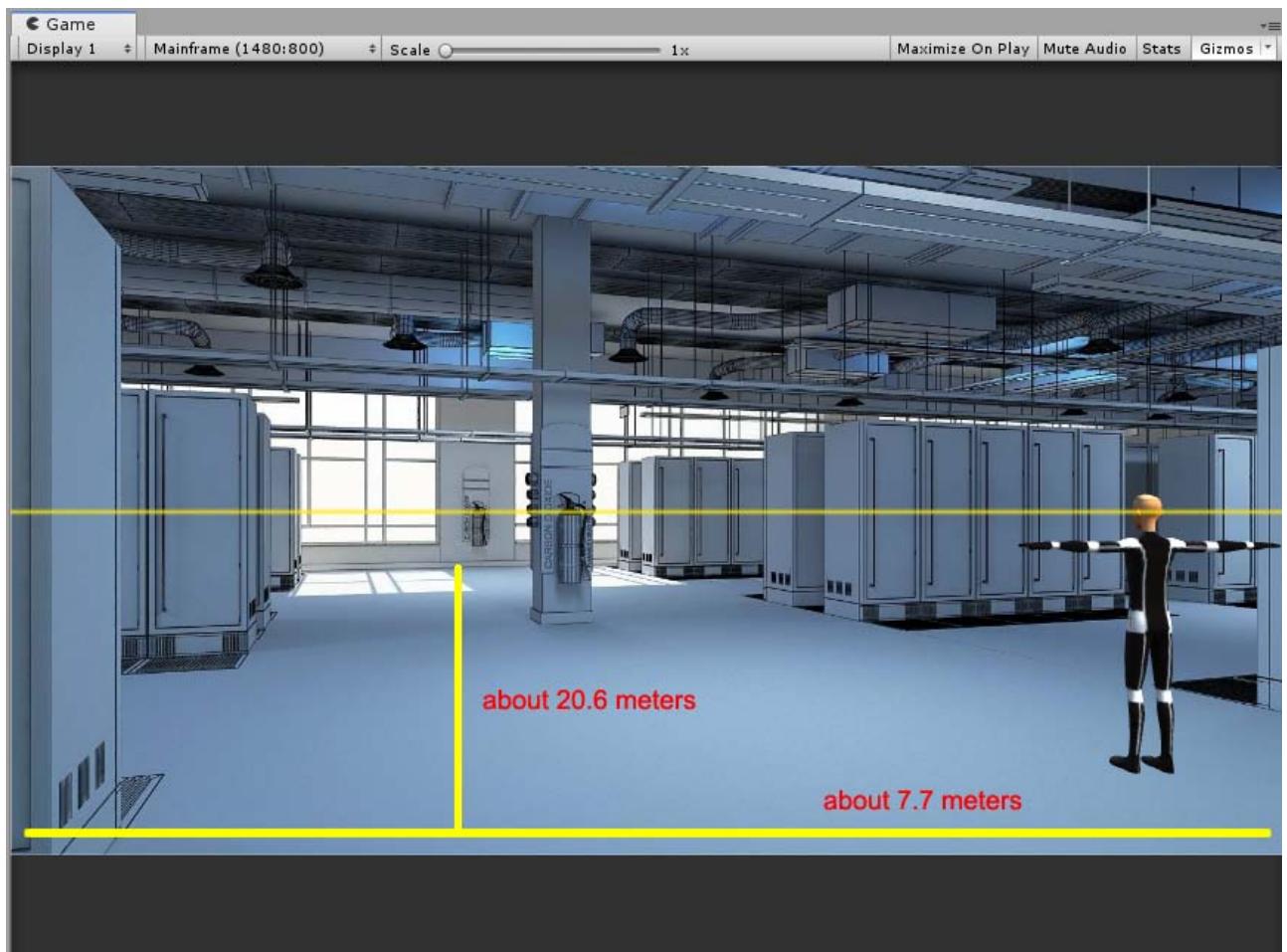
A 2.5D environment does not require high values and when there is a camera rotation a high FOV causes an undesired character distortion on the two screen sides. In this case we have no camera rotation and we can keep it even a little higher than usual (a standard size could be between 20 and 30).

For now let's try to set it to 51, we can always change it later.

Let's start by measuring width and depth with Meter tool.

To see how to use Meter: [Tutorial Ver2.0](#)

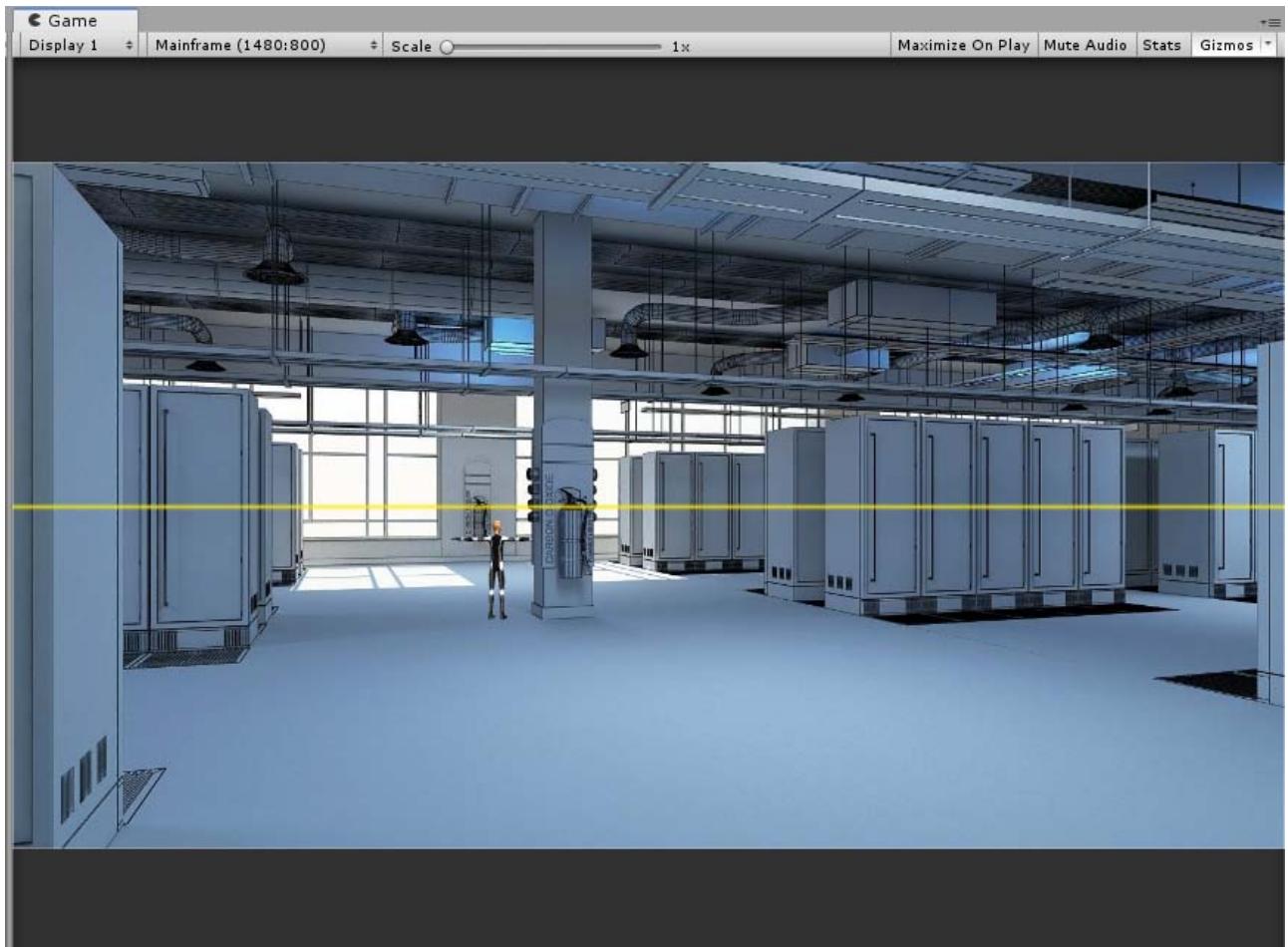
Note that we are in perspective projection and if we are evaluating the width of the scene at the point closest to us we will have to use meter at that point and no, e.g.. halfway through the scene.



About 8 meters x 21 meters. We are lucky as the two sizes already appear proportionate without having to make major changes.

We choose to slightly reduce the measures by moving the floor to -0.71.

To adjust character height we move it near the central column and we rely on the fire extinguisher hanging on it.



A good compromise would be to have the character's head just above the fire extinguisher. At this point its scale should be 1.24.

By keeping the left mouse button pressed, we can move the character around the scene, checking that it maintains a correct proportion with the objects closest to it.

We are lucky again as at the farthest point of the scene we see another fire extinguisher placed at the same height as the first.

If our scene is set correctly by moving the character to that point we should find the same proportions.

Here, too, the character's head should be just above the fire extinguisher.



Great!

As already said, if adjustments were needed, they can be obtained by small camera rotation.

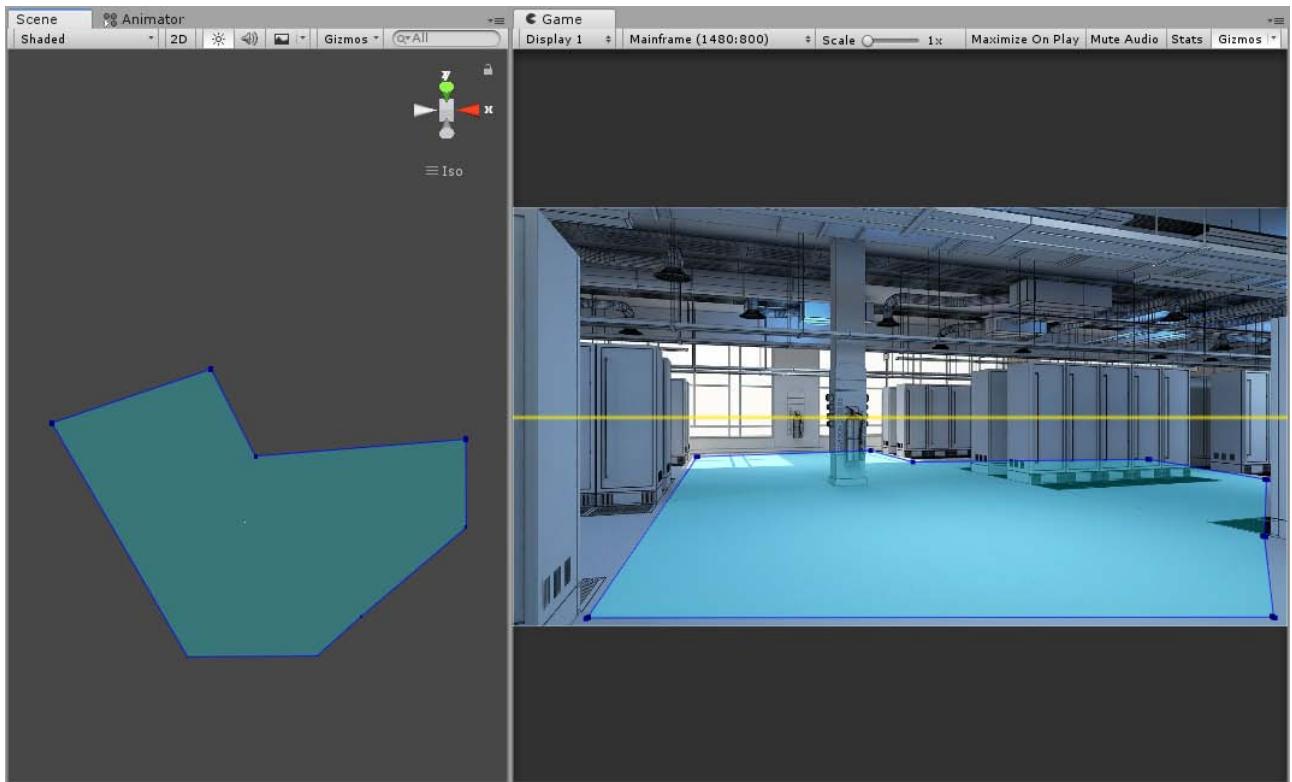
Now from the Navigation Tab **<Create new area>** we can draw our first walkable area.

It is useful to keep a top view in Scene View.

When working with 2D images in a 3D world it is essential using imagination to try to see our image from above and from all sides.

Our first area is most of the time the walkable area which will contain several holes where we will place sprites or even 3D objects.

We would like the character to walk between the two mainframe groups on the right so we will design our area taking this into account.



Now if you want you can bake area and try your scene.

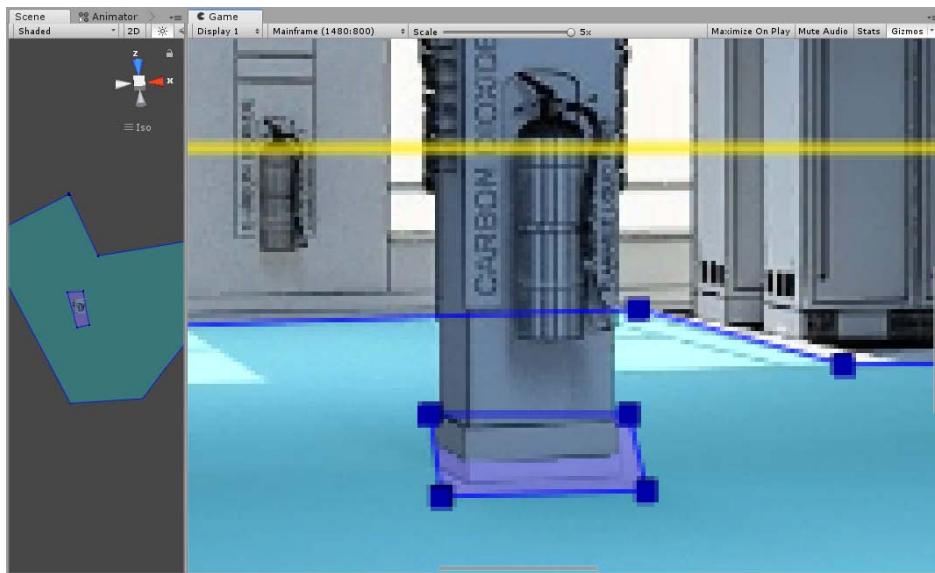
We now need to create two non-walkable areas, one for the column and one for the mainframe group.

Navigation Tab, change **<Type>** to *NotWalkable*, **<Create new area>**. Draw a square around the column. As you can see in Scene View what appears square in Game View is actually a rectangle in Scene View and this because of perspective.

Where is the column relative to our scene? Midway, closer to us, further away? We can also use the main walkable area as a reference.

Leave area as it is and drag your column sprite to Hierarchy (cropped from the background).

Objects Tab, drag sprite in its field. Grab it with the left mouse button and drag it to the newly drawn area. Look in Scene View. Do you think it is positioned correctly in relation to the environment? When position is right, enlarge Game View and use its zoom.

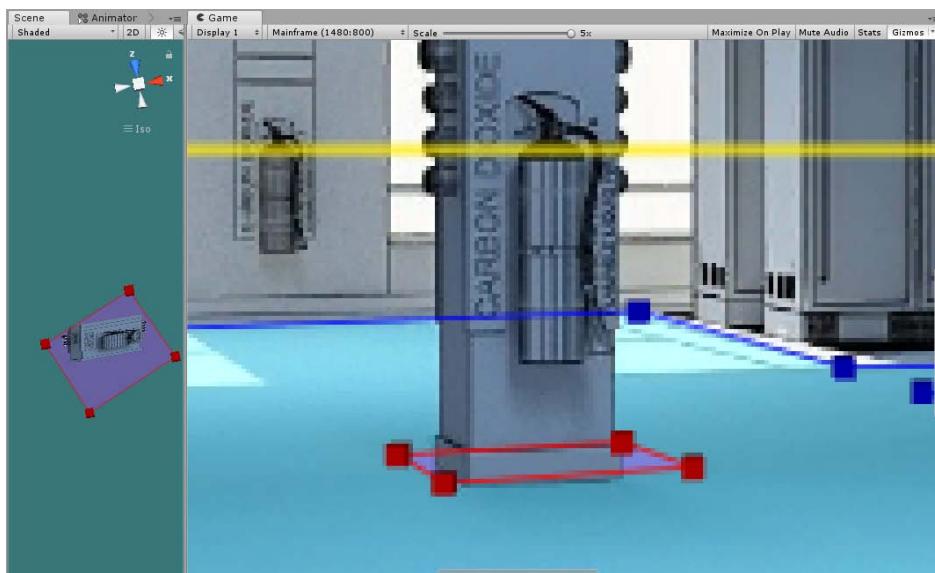


Using the positioning buttons, overlap sprite on the background image.

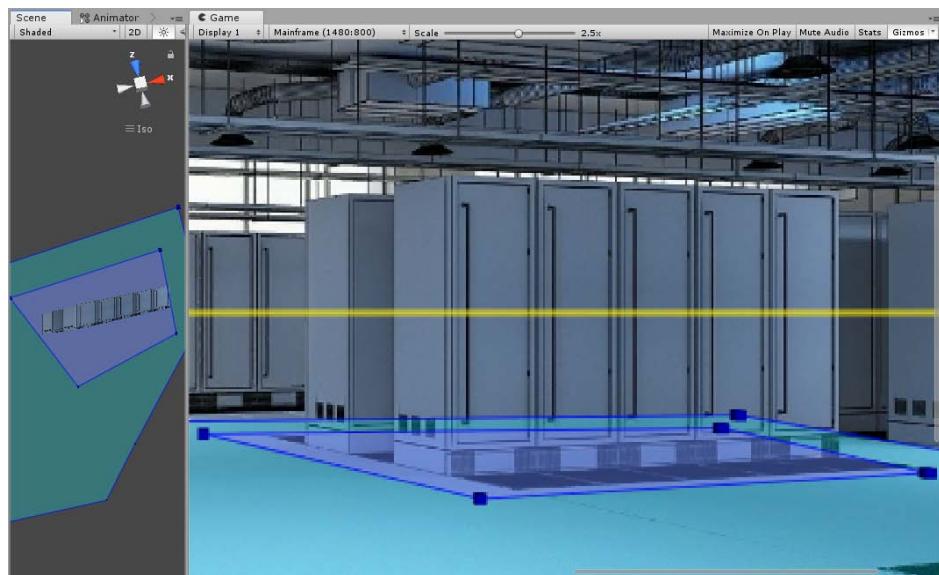
You can also use **<Enable accuracy>** to increase precision and **<Enable hold down>** in the **General Option**.

Use **<Hide object>** to hide and show sprite to check it is best placed.

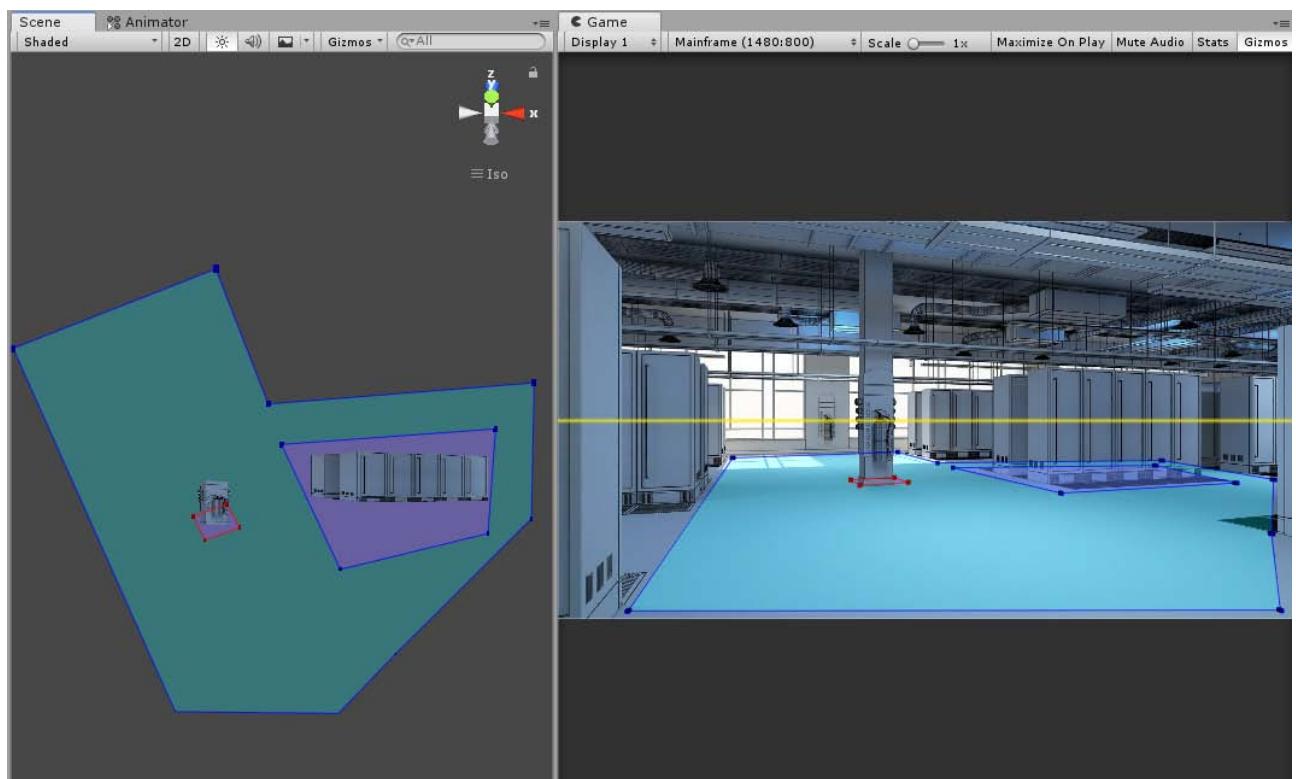
Finally fix area around it. You can use **<Select point>** option to move vertices more precisely.



Repeat same steps to place mainframe group.



Scene is ready



Bake and test it.

In this scene shadow has been added to character by creating a mesh once the settings are complete.

**Tip:**

Another consideration you can make is to count the number of steps.

Once you have assigned to character the most suitable walking speed for your scene, check how many steps it takes compared to the number of meters.

In this demo we have a depth of about 18 meters. One step generally covers 65/80 centimeters. The speed assigned to the character is 1.7 which seems right for the scene. Character travels the entire area with 21 steps. We divide 18 meters by 0.80 and we get around 22. It can be fine. By slightly adjusting all our values, we can achieve even greater precision (if the difference is minimal we can avoid reshaping the scene and varying the walking speed slightly).

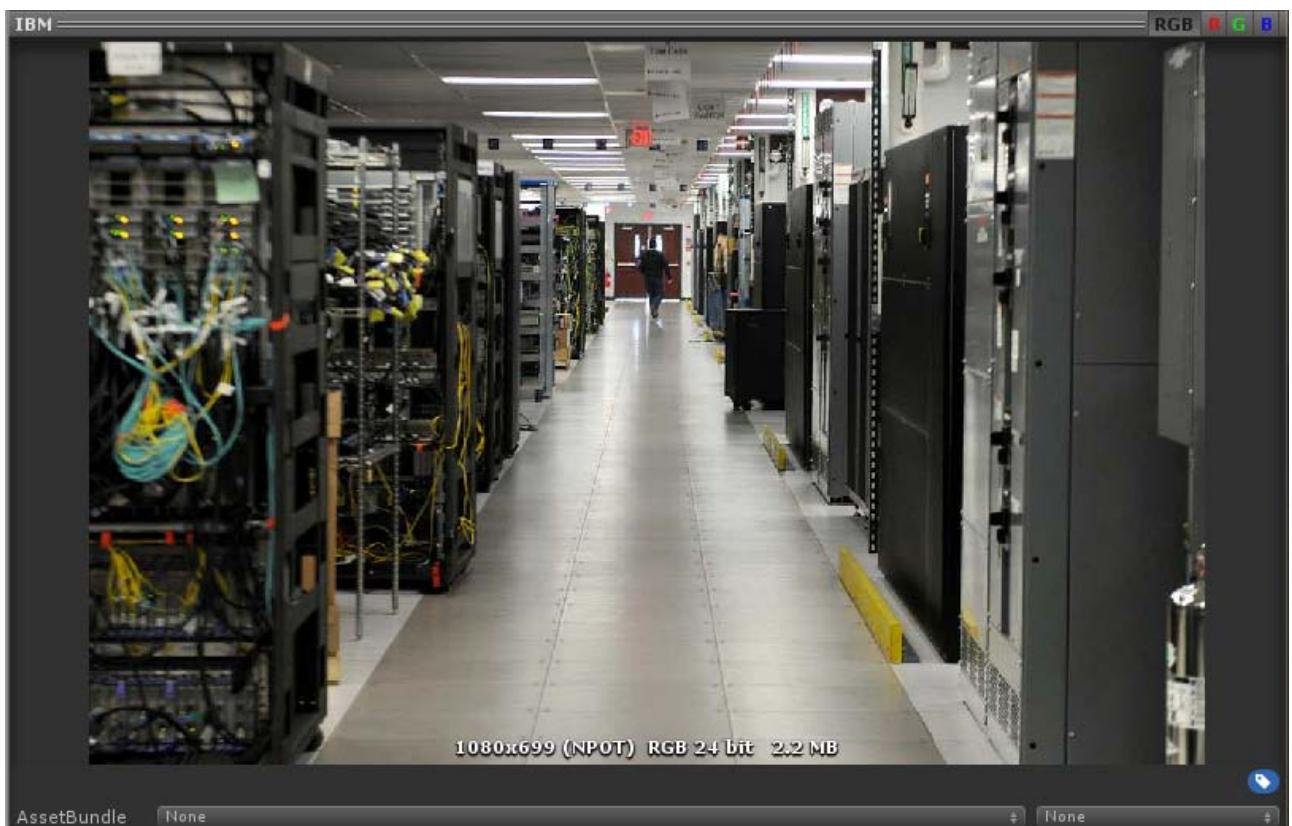
# Demo scene: IBM

---

You will find this example in Demo folder.

It uses 2.5D Toolkit environment tool which requires Background Camera.

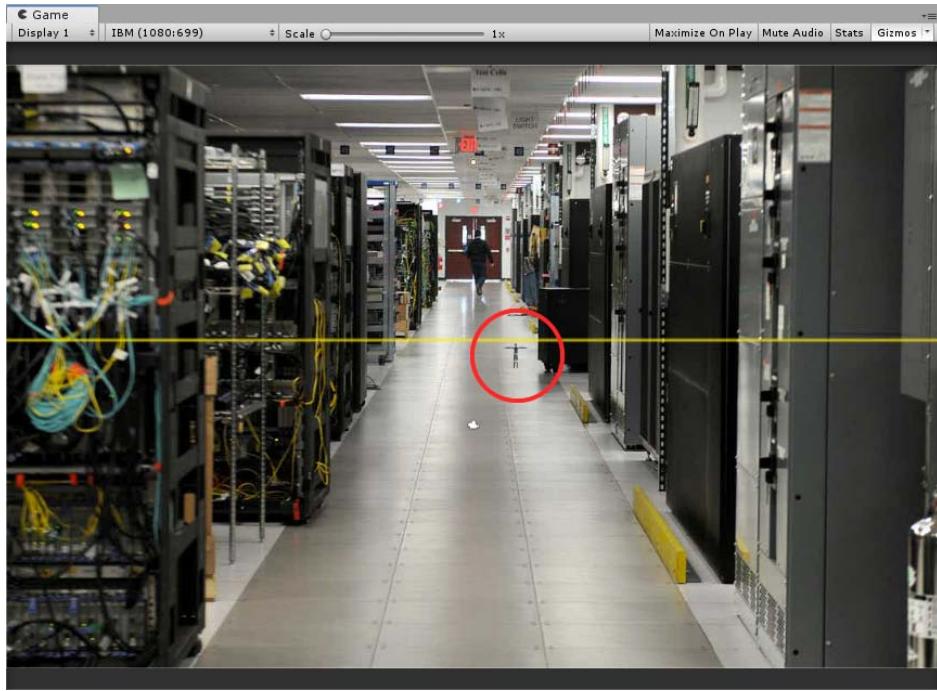
Projection is Perspective, Camera distance is set as automatic, Aspect Ratio is set to 1080:699, Character walk: 1.3, Character run: 4.



Steps to follow are the same as in the previous demo. However, there is some small difference.

Let's look at the picture.

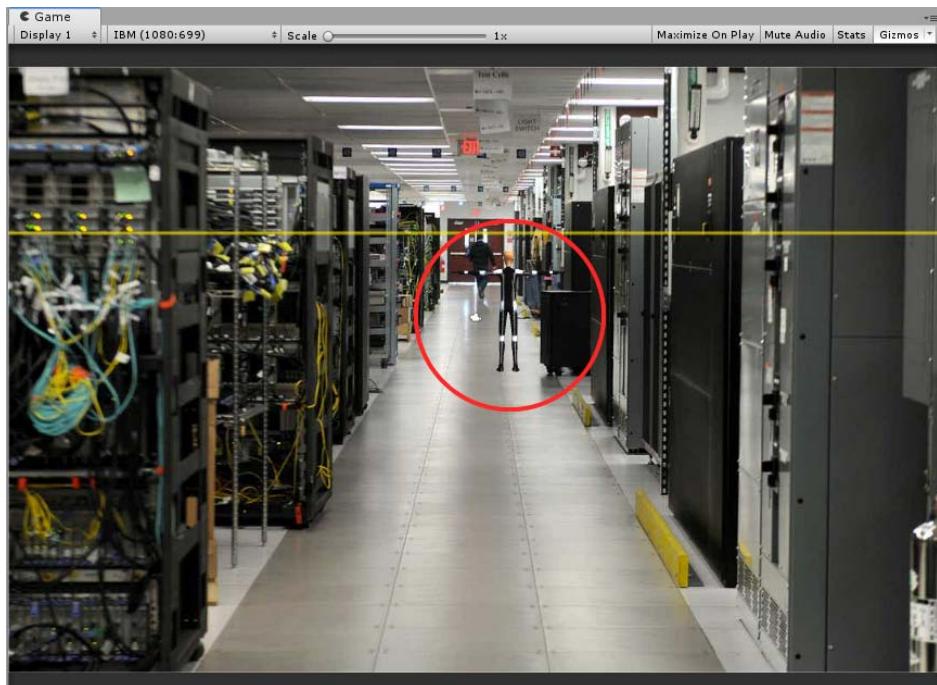
The first important thing to note is that with the default settings the horizon line is below the furthest point of the area that we have to create. To raise the horizon line, you can lower the FOV or increase camera rotation.



Since we know that a 2.5D environment does not require high FOV values we start with lowering it and we stop at 26 which could be a good value. It could go down more but if needed it will be done later.

Now moving the character (with the right mouse button) in depth we see that it scale excessively.

We therefore increase the camera rotation to about 6. If we now move the character again we will see that its scale has improved considerably.



Furthermore, the horizon line is now beyond the end of the area that we need to create.

Now that we have given a first initial setting to the scene we can start working with the measures.

The floor is covered with square tiles that will help us get started. How much could a tile measure? Let's assume 60 centimeters which give us a total width of 1 meter and 80 centimeters.

By measuring the width of the three tiles with the meter we get about 2 and a half meters.

The height of the floor is set to -1 by default. We raise it to reduce the size. After some tests we find that 0.5 gives us a width of about 1 meter and 80 centimeters which is the measure we were looking for.

By measuring the depth we get about 34 meters which would seem a little too much. We increase the rotation of the camera by setting it to 6.6.

Now the depth is about 25 meters which appears to be realistic.

Finally we move the character close to the person at the bottom of our scene and resize it to the same height. Use the zoom in Game View for more precision and if you want you can activate **<Enable accuracy>** in the **General Option**.

A scale of 0.71 seems ideal.

Now move the character throughout the scene, next to the shelves or the various objects of the image and check that it respects the right proportions.

When you are satisfied you can draw the walkable area.

**Tip:**

Another consideration you can make is to count the number of steps.

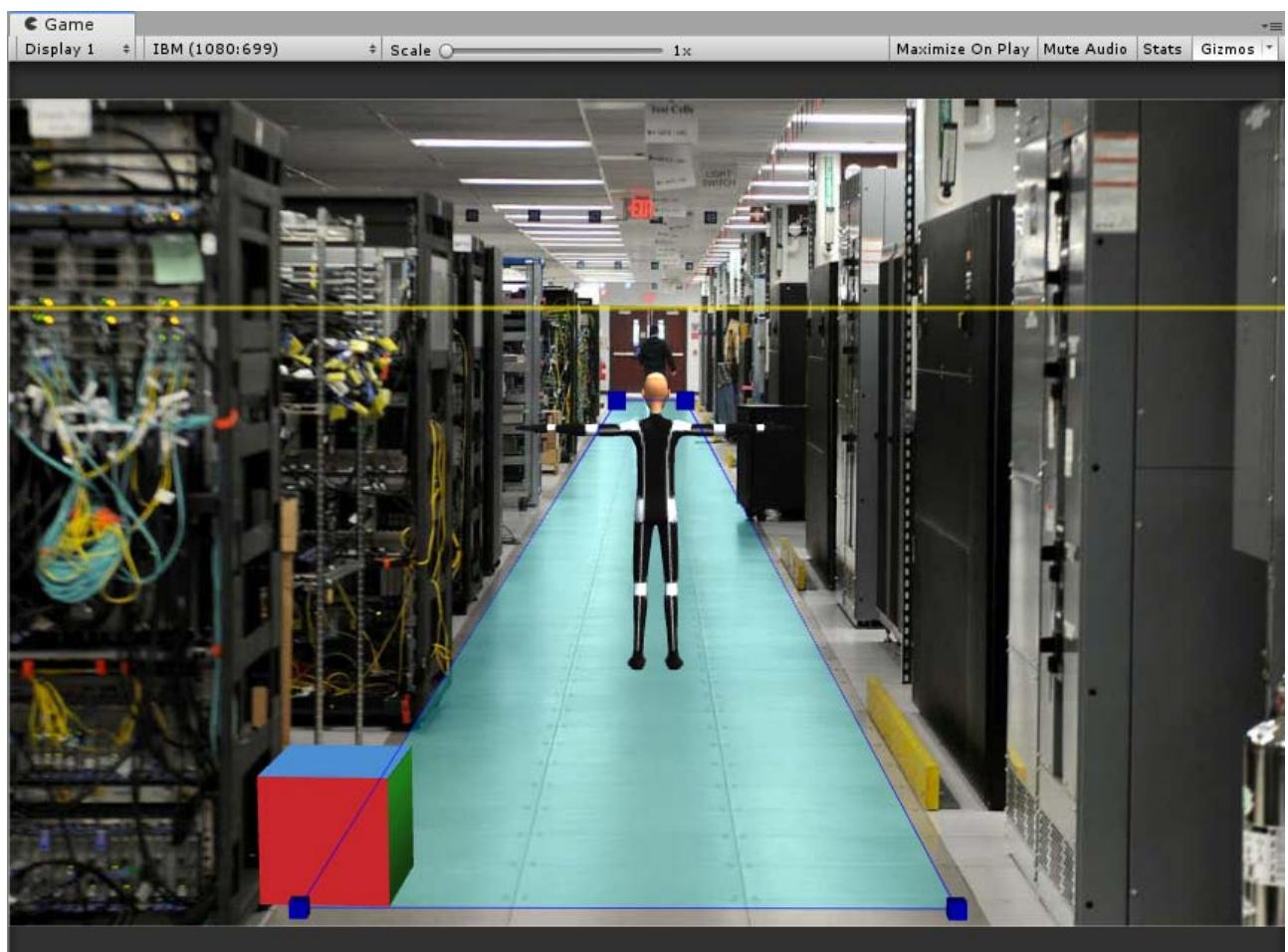
Once you have assigned to character the most suitable walking speed for your scene, check how many steps it takes compared to the number of meters.

In this demo we have a depth of 25 meters. One step generally covers 65/80 centimeters. The speed assigned to the character is 1.3 which seems right for the scene. Character travels the entire area with 37 steps. We divide 25 meters by 0.70 and we get around 35. It can be fine. By slightly adjusting all our values, we can

achieve even greater precision (if the difference is minimal we can avoid reshaping the scene and varying the walking speed slightly).

Using Cube is useful for defining the amount of camera rotation. Character is needed for estimate scene depth but often it does not help to find the right rotation. Placing Cube in scene will make it easier to guess if camera rotation matches the environment.

Could the upper face of the cube be the natural continuation of the shelf surface?

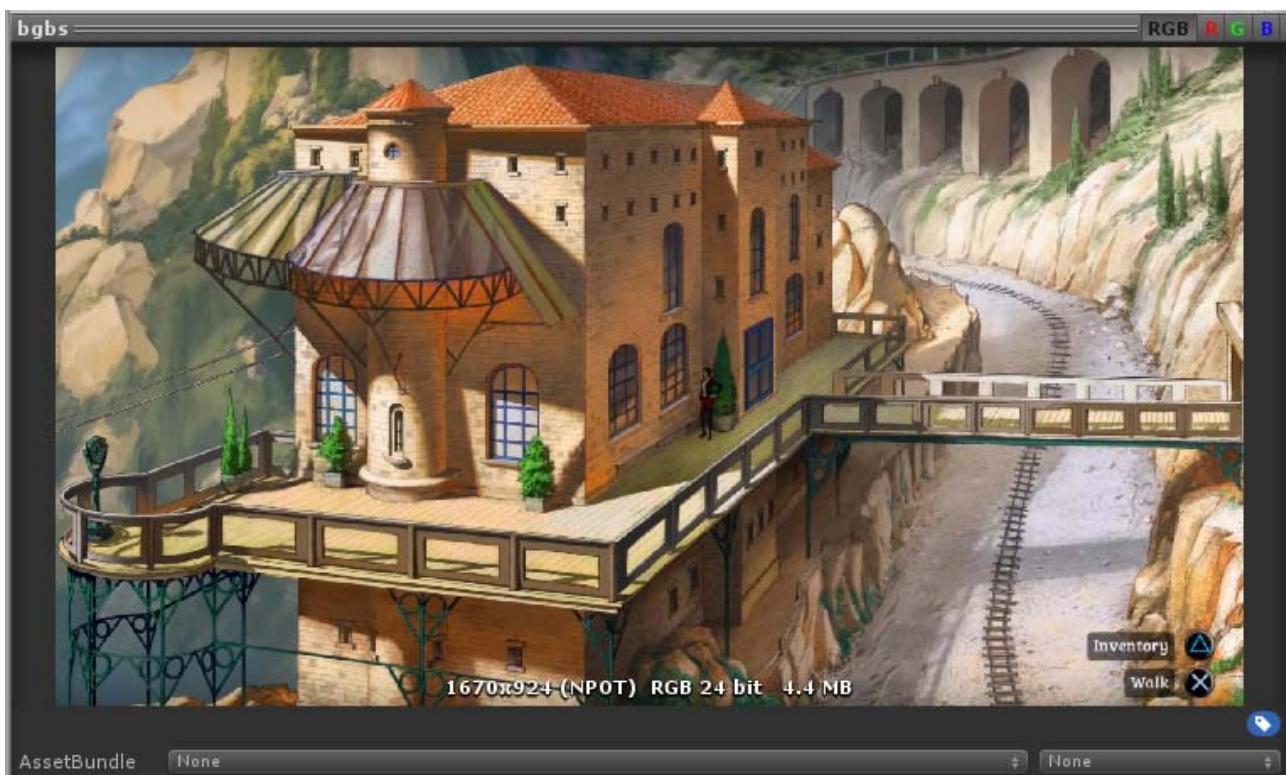


# Demo scene: Cableway Orthographic

You will find this example in Demo folder.

It uses Background Camera and Plane.

Projection is Orthographic, Camera size is set as automatic, Aspect Ratio is set to 1670:924, Character walk: 1.2, Character run: 2.5 (Animator Controller).



As is evident, this image is a drawing and however very beautiful it does not fully respect the proportions. It requires more attention than a photograph.

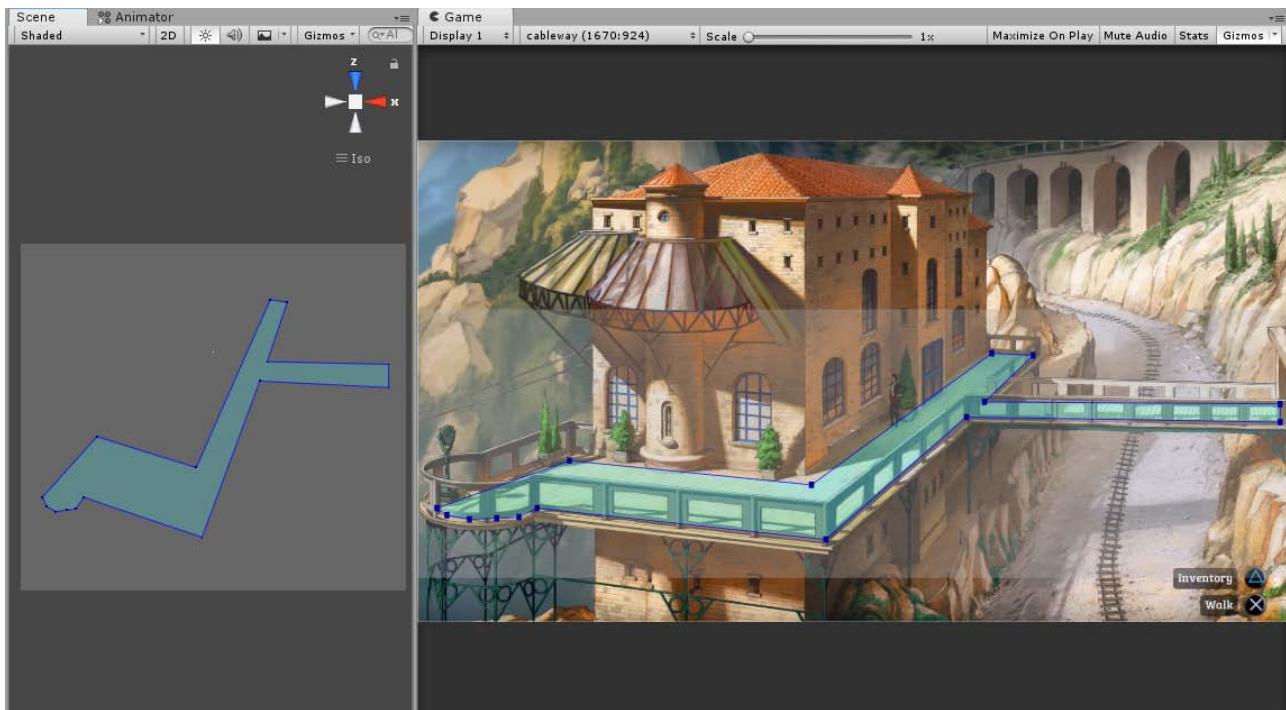
We do not dwell on all the steps as in the previous demos (watch this video that explains how to place a Plane with the Background Camera [Tutorial Ver2.0](#)) but we will make some important considerations.

If you try to build the environment a bit quickly you will find that walkable area does not exactly match the design (seen from above with imagination).

The deeper part of the area will be longer than it should be.

Not being able to use the meter tool (which you can have in the "Use 2.5D Toolkit environment" mode) you can rely on the number of holes in the railing.

You have to get a fair compromise between the various sides of the railing. By working on Plane position and on camera setting you can obtain a satisfactory result.



As you can see from the screenshot, the furthest part of the area is about a third of the part closest to the camera (the two parts divided by the bridge).

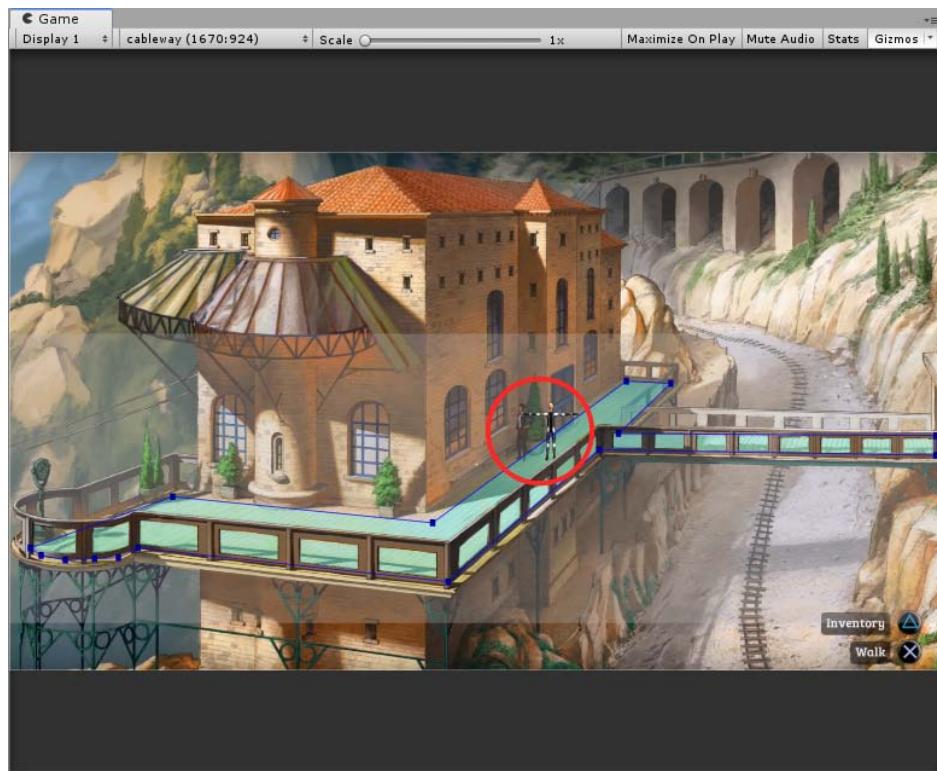
Overall, the area appears proportionate with the image, but remembering that it is a drawing, perfection cannot be achieved.

After building our environment we have to set character resizing as we are in orthographic projection.

Tool provided is very simple and is useful just to have a preview of our scene. Many assets for creating adventure games provide more accurate tools that allow you to change the height of the character in more than two places in the same scene. In the current scene, character should probably further increase his size in the leftmost part of the image, near the telescope.

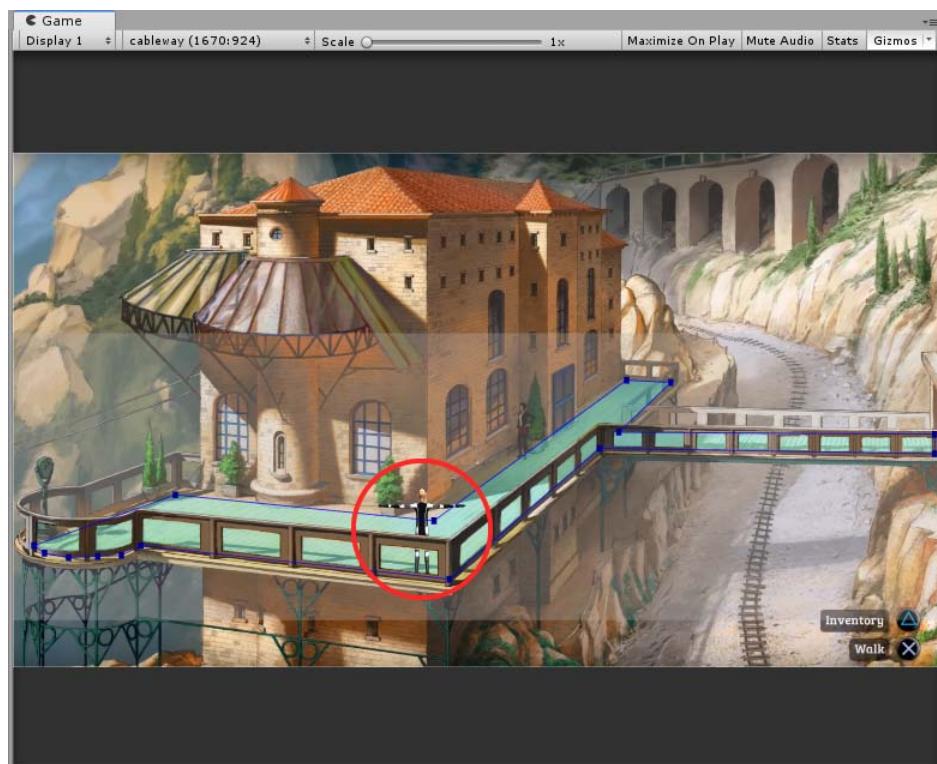
Enable **<Use character resize>** in the **General Option**. If there is a previous configuration, reset both points

Move the character close to Nico, the female figure you see in the corner, and resize him through this tool.

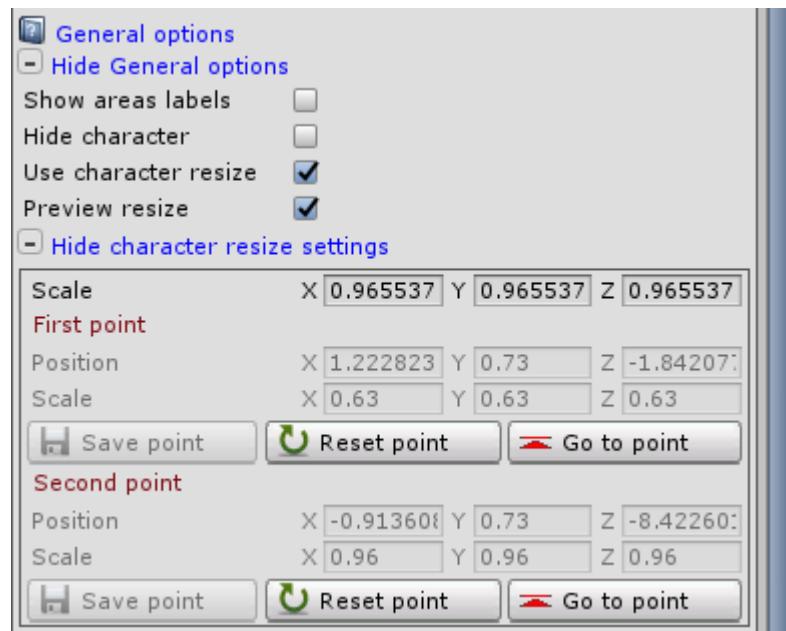


Click <Save point>.

Now move character to farthest point or closest point to the railing and resize it again.

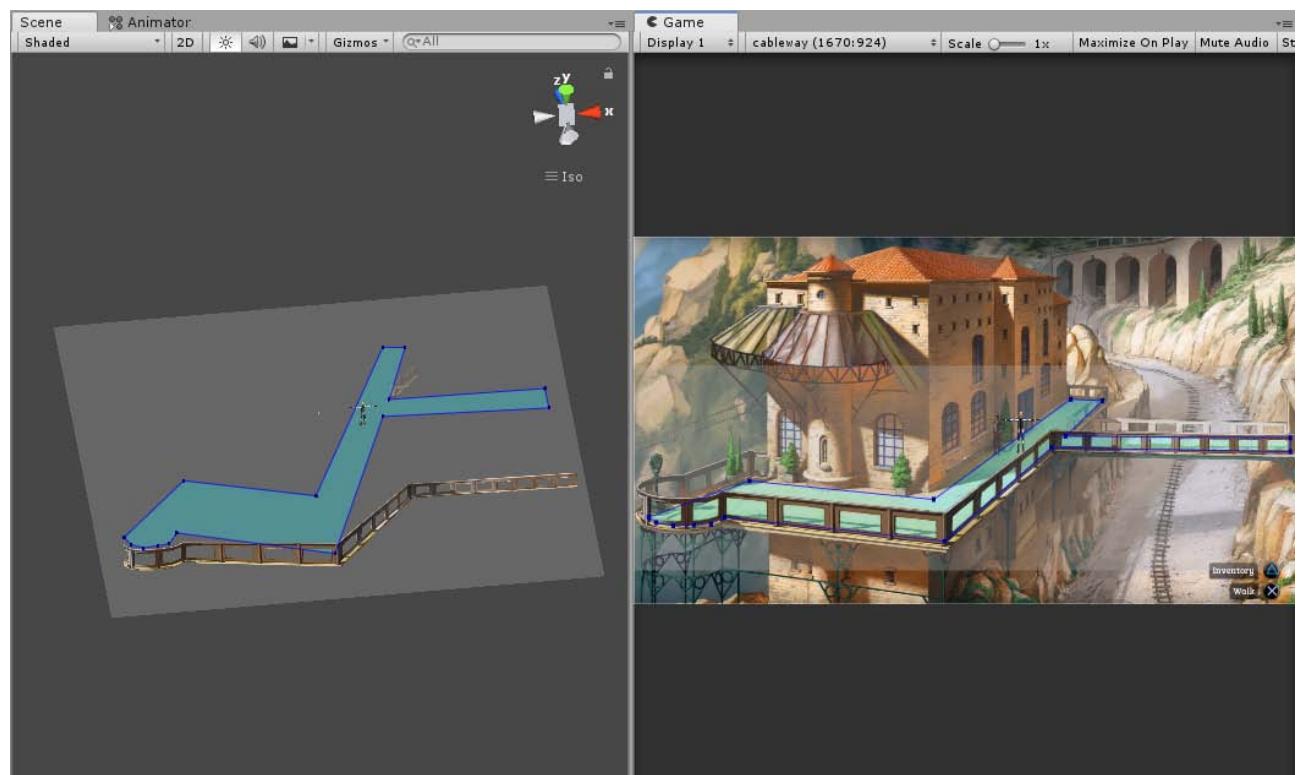


You should have a configuration like this



With <Preview resize> in the **General Option** enabled, move the character in the scene with the right mouse button and check its resizing.

Now all that remains is to place the sprites. This is the ready scene:



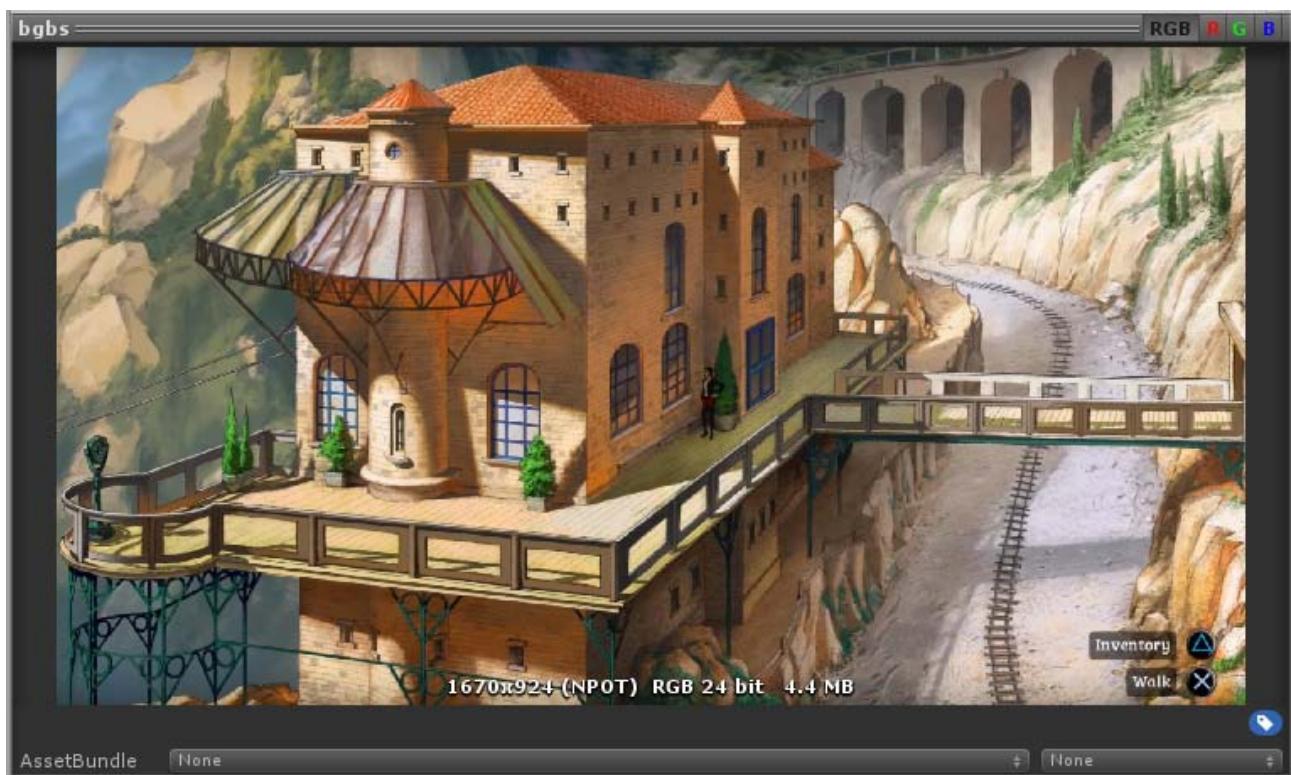
# Demo scene: Cableway Perspective

---

You will find this example in Demo folder.

It uses 2.5D Toolkit environment tool which requires Background Camera.

Projection is Perspective, Camera distance is set as automatic, Aspect Ratio is set to 1670:924, Character walk: 1, Character run 2 (Animator Controller).



Although it would not seem this image in Perspective projection involves many difficulties. As said in the previous demo, it is a drawing and does not respect real proportions. We will use the Meter tool to create the environment but we cannot be precise. If the image were a photograph, we could rely on the holes of the railings and assuming that their measurement is 1 meter we could build an environment quite precisely. However, if we try this way with this design we would never be able to complete the work.

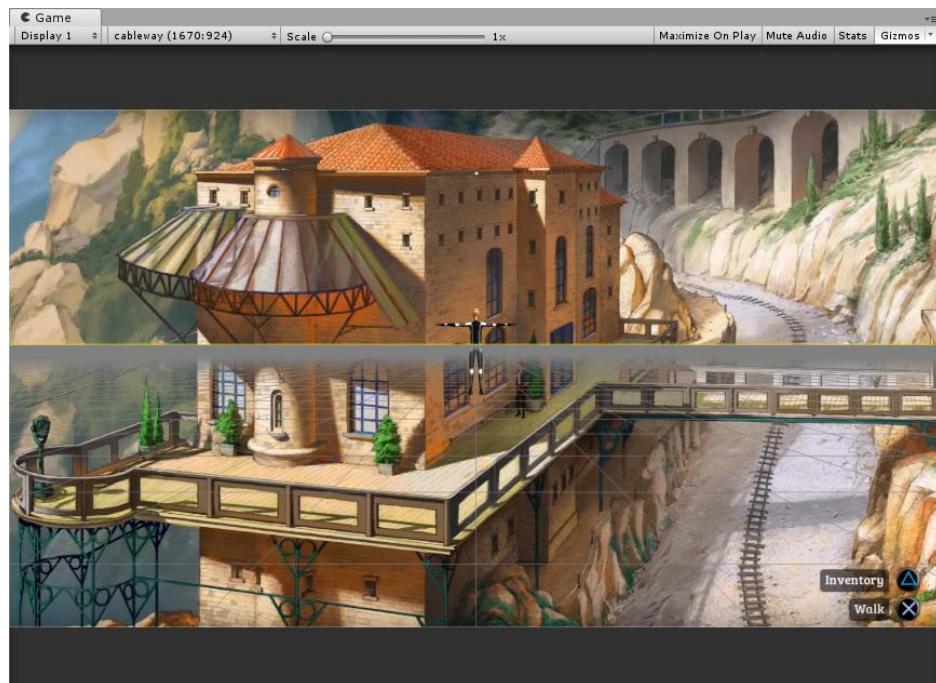
We will have to settle for partial results.

The part closest to the camera has about 5 and a half holes then 5 and a half meters.

The part that goes deep has about 10 holes then 10 meters.

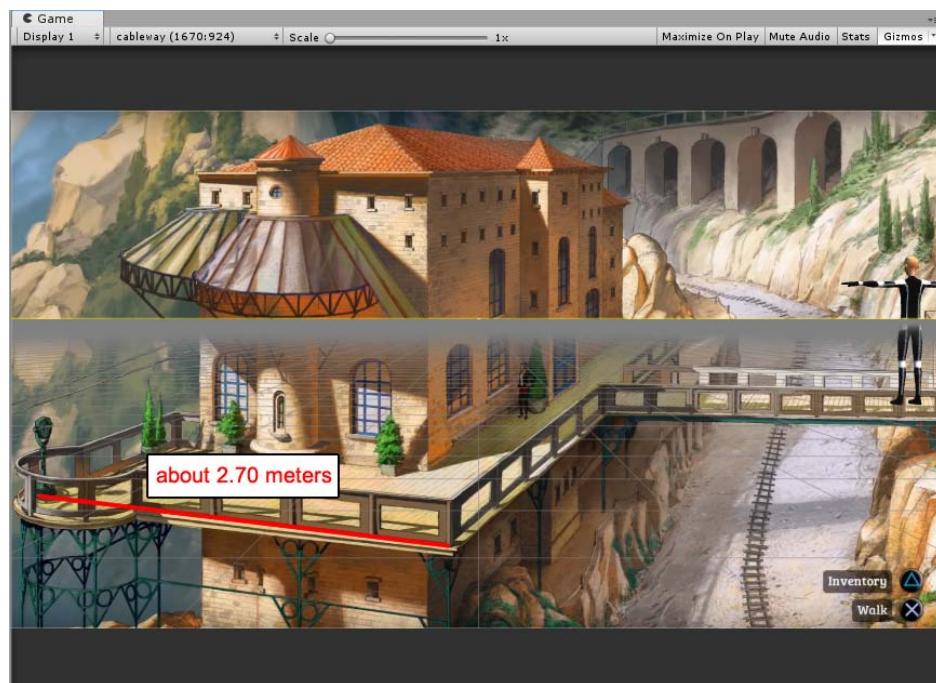
Our area should therefore be about twice as deep as it is wide.

We immediately see that our horizon line is below the furthest point of the area that we will have to create.



Let's start by lowering the FOV to 30. Now the horizon is above that point.

Now let's measure the width of the point closest to us.



2.7 meters while it should be 5.5 meters

Setting Floor to -2 we get about 5.5 meters.

With these settings if we measure the depth we will have about 96 meters while we should have 11 meters.

We then rotate the camera and see that with a value of 8.8 we get about 11 meters.

But now our first measure has changed from 5.5 to 3.

We therefore continue to modify the position of the floor, the rotation and if needed also the FOV to find that combination we are looking for, that is the depth as double the width. At the same time it is also required to move the character to verify that the new settings do not involve an unsuitable resizing (this does not happen in Orthogonal projection as you will define the resizing according to your needed).

After trying all the possible combinations it will be evident that this design does not allow to obtain a realistic setting.

With a floor at -4.37, a FOV at 35, a camera rotation at 15.5, a character scale at 0.57, an acceptable configuration is obtained.

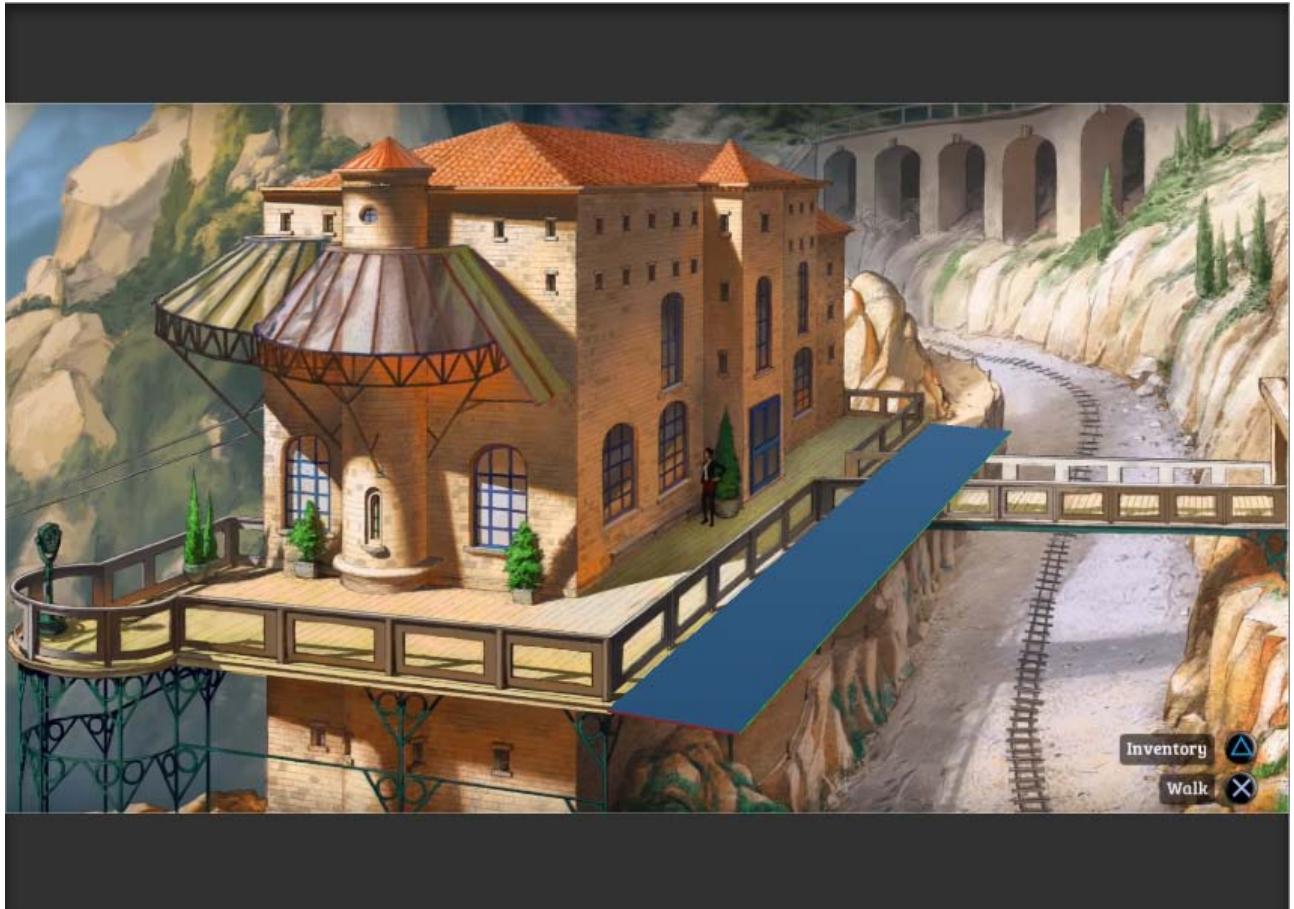
In this case, however, the character will undergo a deformation on both sides of the screen.

To reduce distortion we need to lower the FOV at least to 25.

With a floor at -5.6, a camera rotation at 14.64, a character scale at 0.56, another acceptable configuration is obtained.

This is another way of using the cube. By making it similar to the floor you can define a fairly reliable camera rotation.

Very often this solution allows to automatically obtain the right character resizing.



If number of steps did not correspond with the distance traveled a possible work around could be to change the walking speed according to where the character is located. This will increase/decrease steps numbers giving a more realistic look to the scene.

# Demo scene: NicoHome

---

You will find this example in Demo folder.

It uses background image in scene and Plane. Projection is Orthographic, Camera size is set as automatic, Aspect Ratio is set to 3374:2161, Character walk: 6.8, Character run: 12 (Animator Controller 2).



This tutorial [Tutorial Ver2.0](#) shows all possible configurations allowed by 2.5D Toolkit using this image.

We added a detail that is not present in the video.

A 3D cat on the chest showing how to give the impression that a 3D object is placed on an object in the background image.

After dragging the 3D object as usual from the Hierarchy in its field in Tab Objects move it with the mouse in front of the point where you have to place it.

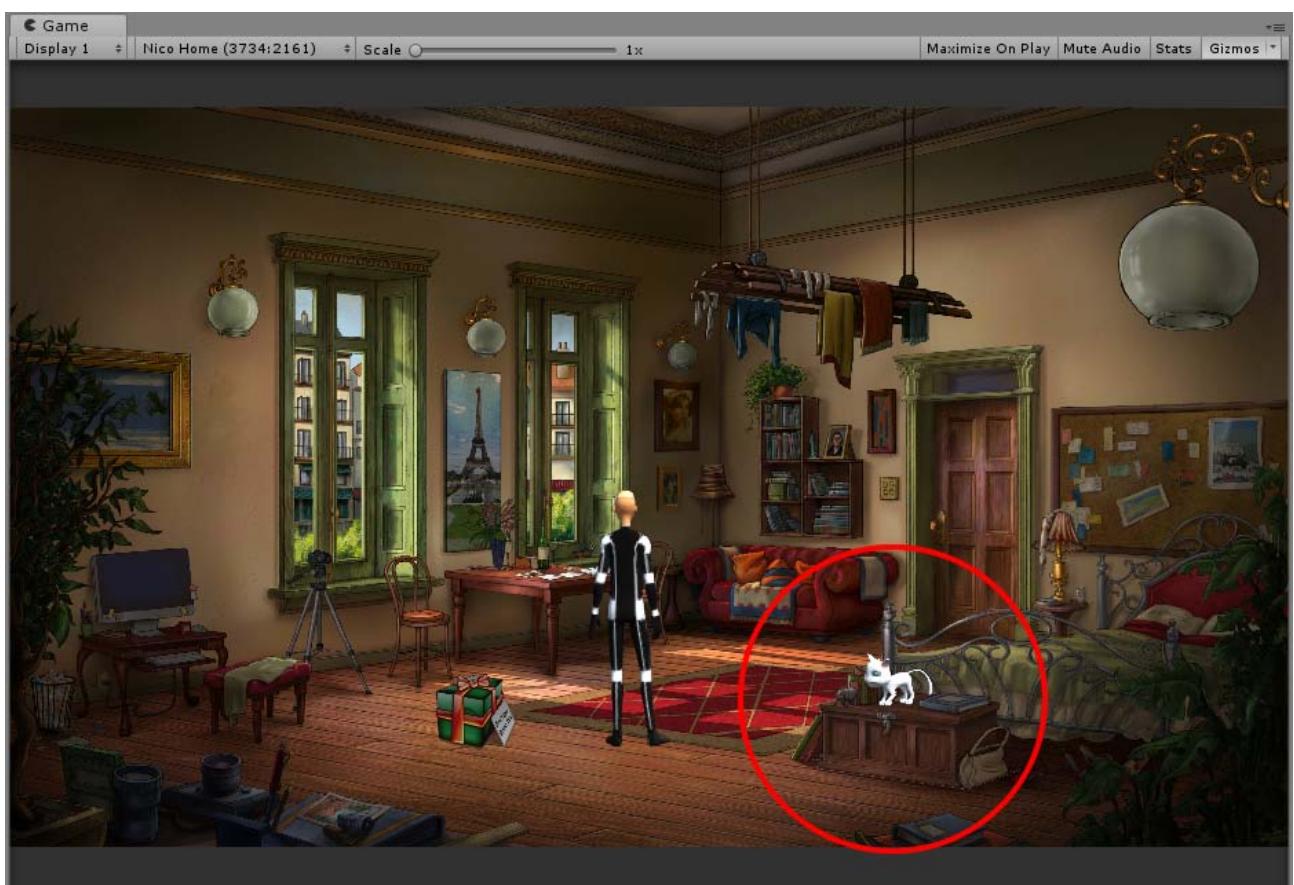
In our case it will be in front of the sprite of the chest but the procedure is the same even if the point is a part of the background image.

Moving object on the Y axis raises it to the desired height and if needed adjust its position on both the X and Z axis but always using the 2.5D Toolkit controls.

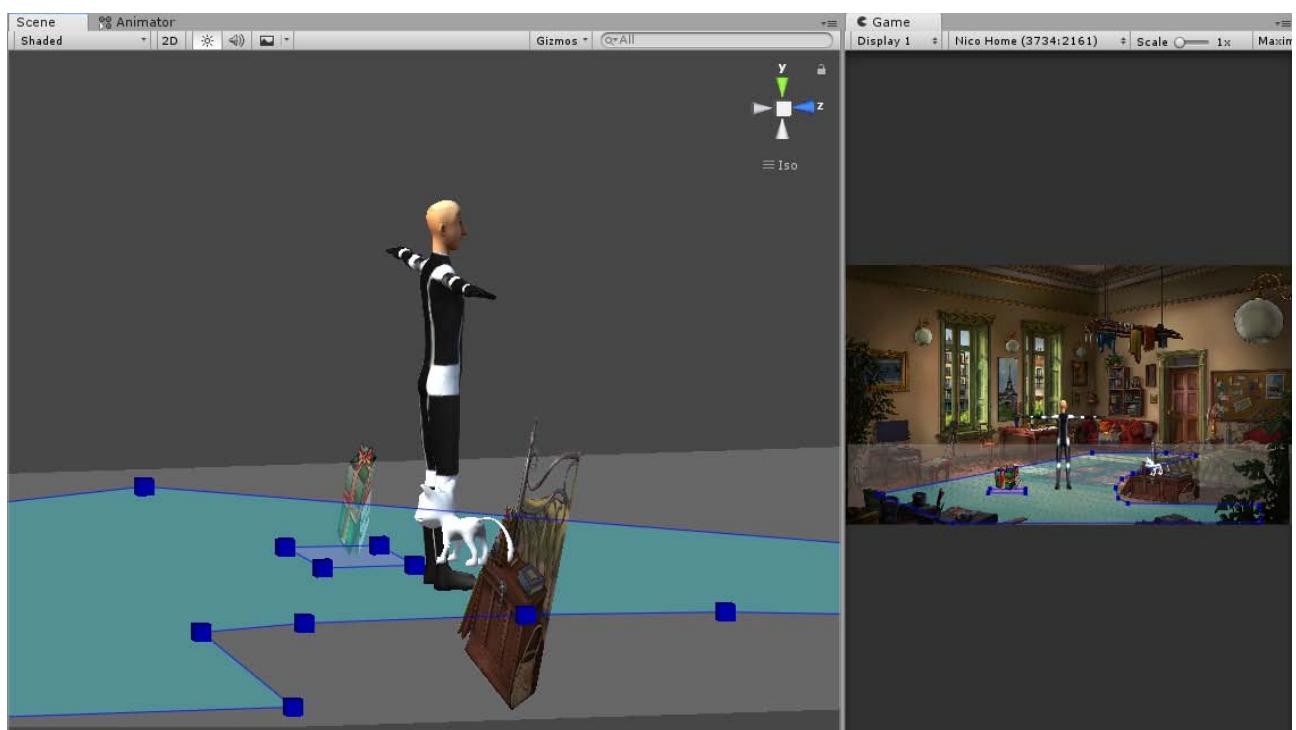
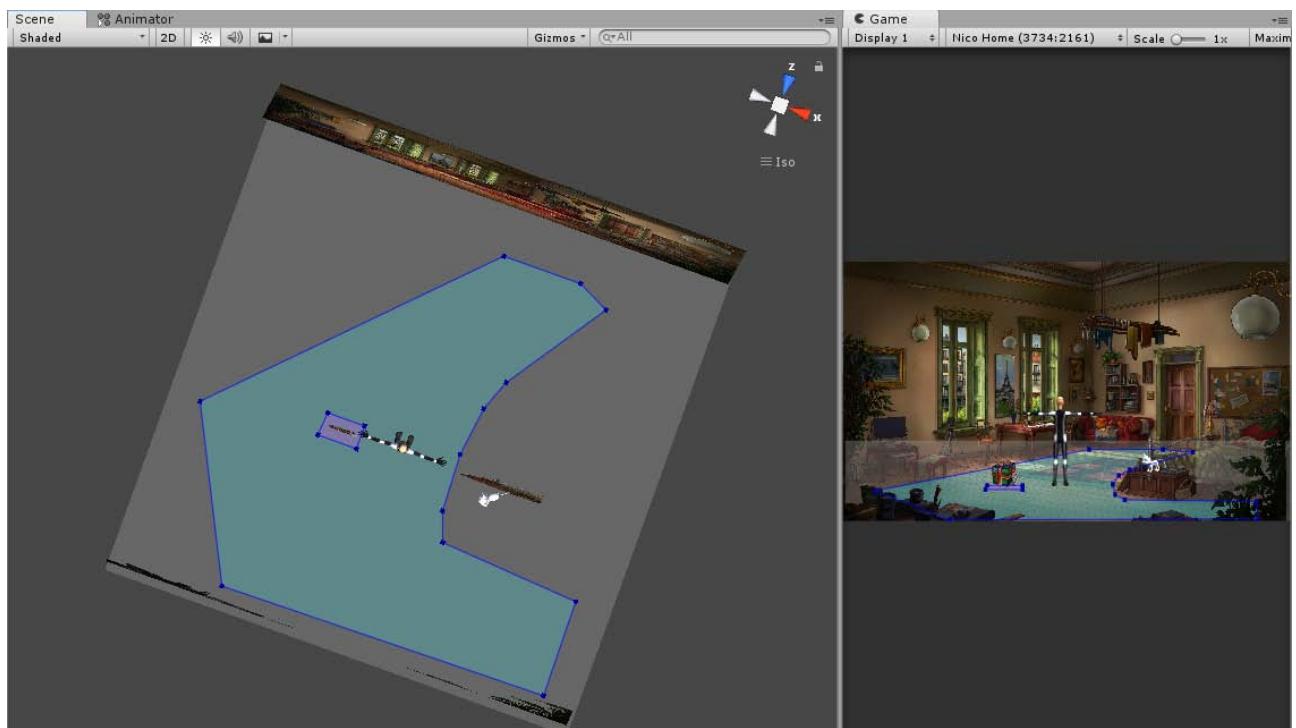
By clicking with the mouse in the scene you will bring it back to the floor.

When you are finished remember to always click on <Empty object> or it could happen that by clicking in the scene you have to place it again.

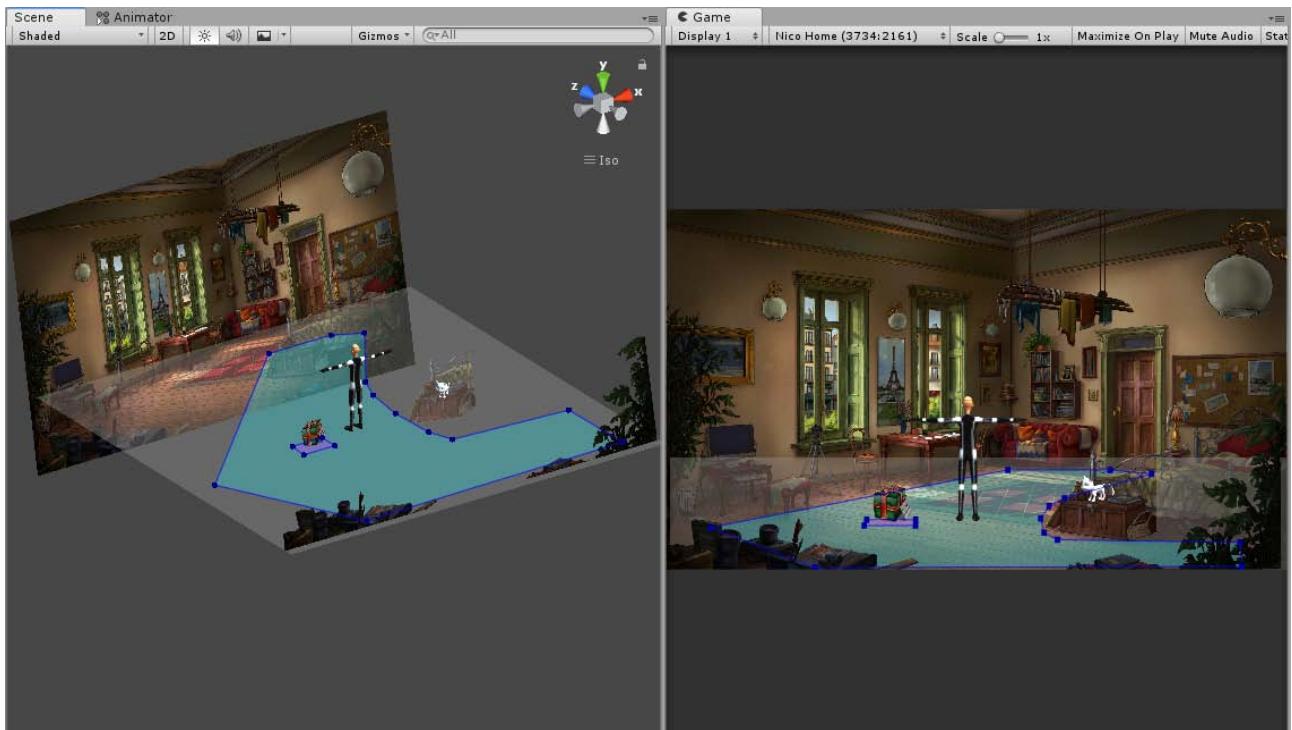
This is how it will look:



And this is its real position:



Finally, this is the completed environment:



## Demo scene: RPG

---

You will find this example in Demo folder.

It uses 2.5D Toolkit environment tool which requires Background Camera.

Projection is Orthographic, Camera size is set as automatic, Aspect Ratio is set to 650:430, Character walk: 0.7, Character run: 1.5 (Animator Controller 3).

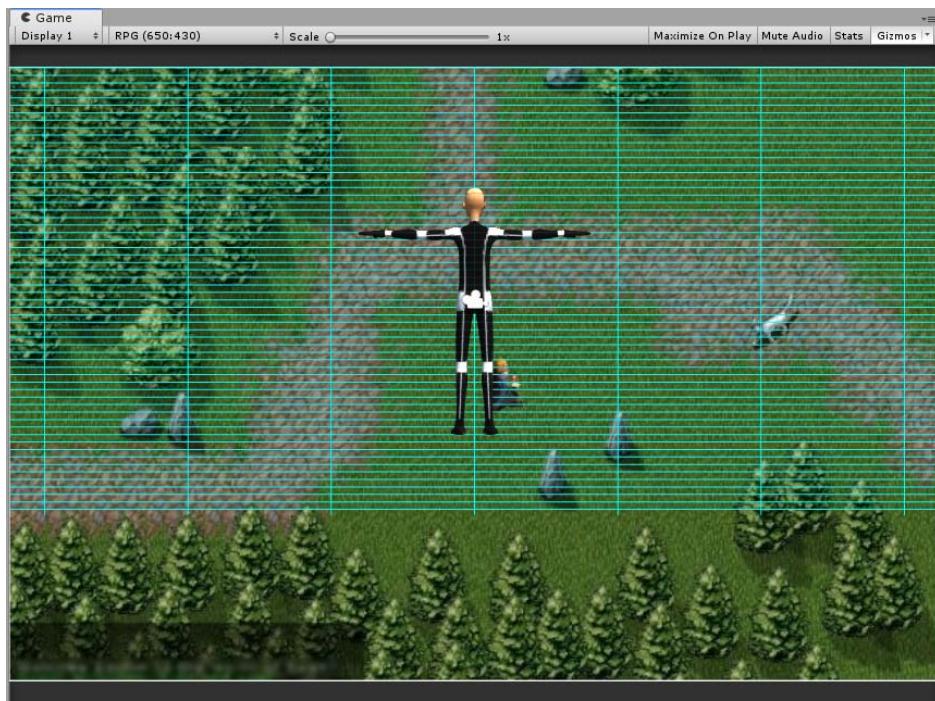


As is evident, this scene will require a high camera rotation.

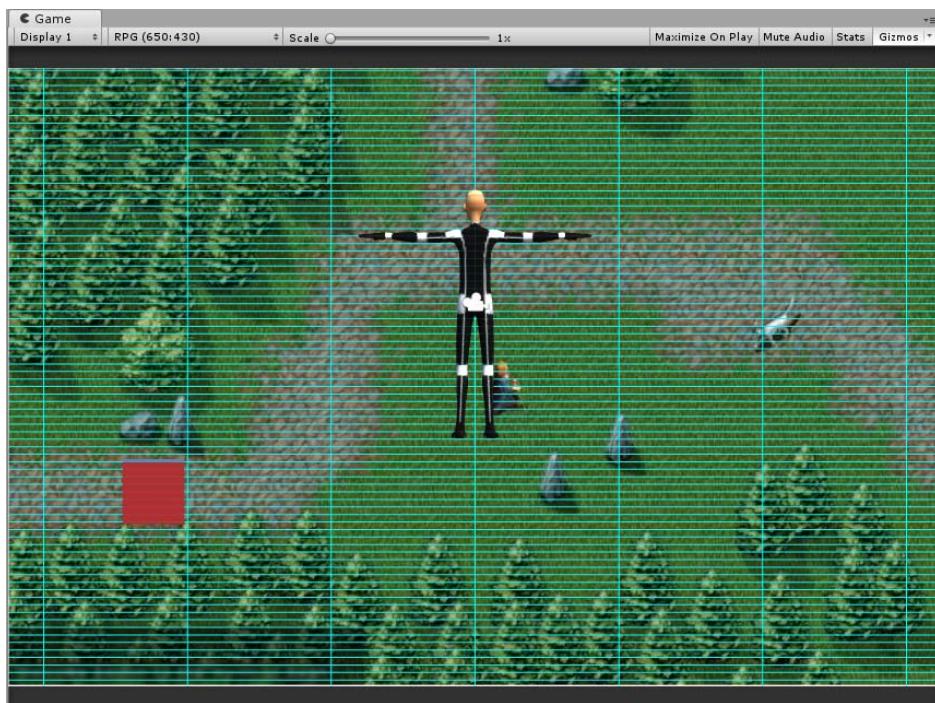
In addition to the perspective the character seen from above will assume, we can also help ourselves with the Cube tool.

Observing the arrangement of the cube faces will give us a better idea of the angle of view.

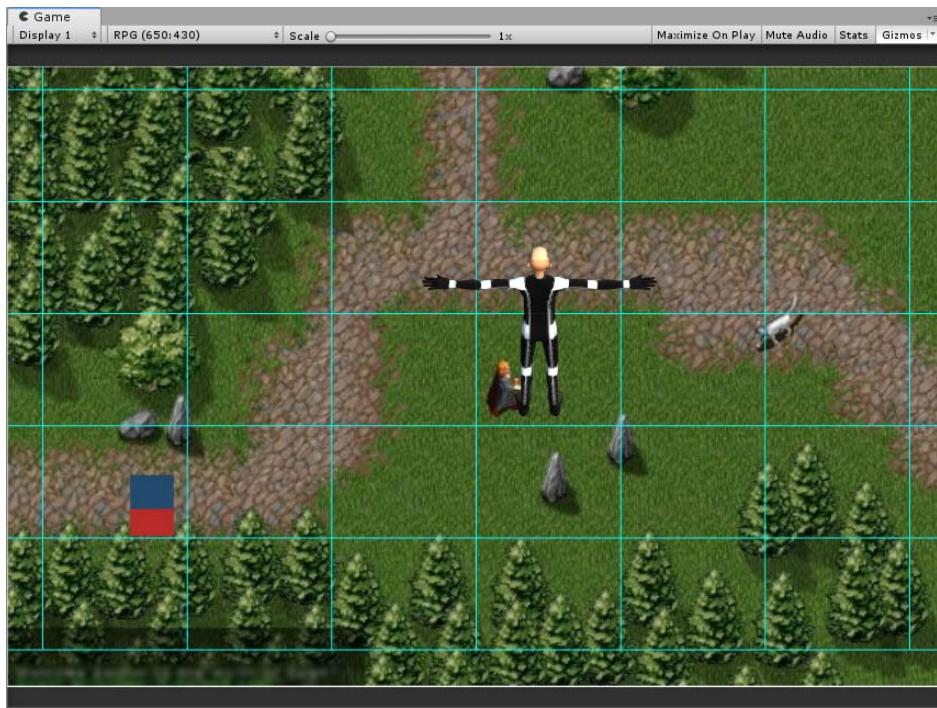
This is the scene in its initial setting:



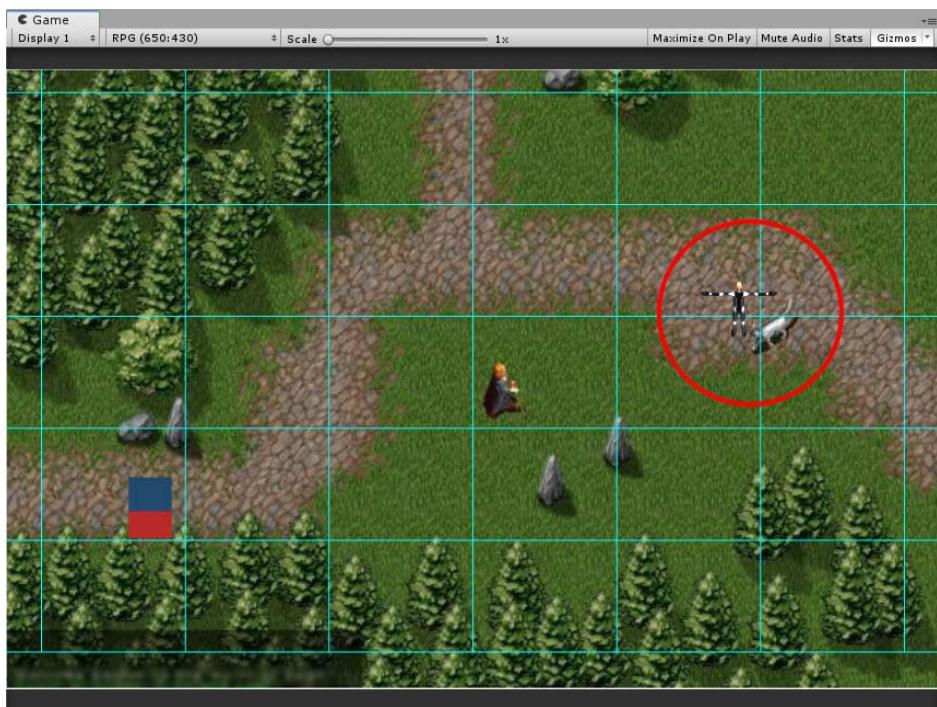
Let's start by lowering the floor to cover the entire area of the background image, then **<Use cube>** , **<Place cube>** , and resize it a little.



Now rotate the camera until you get a good perspective.



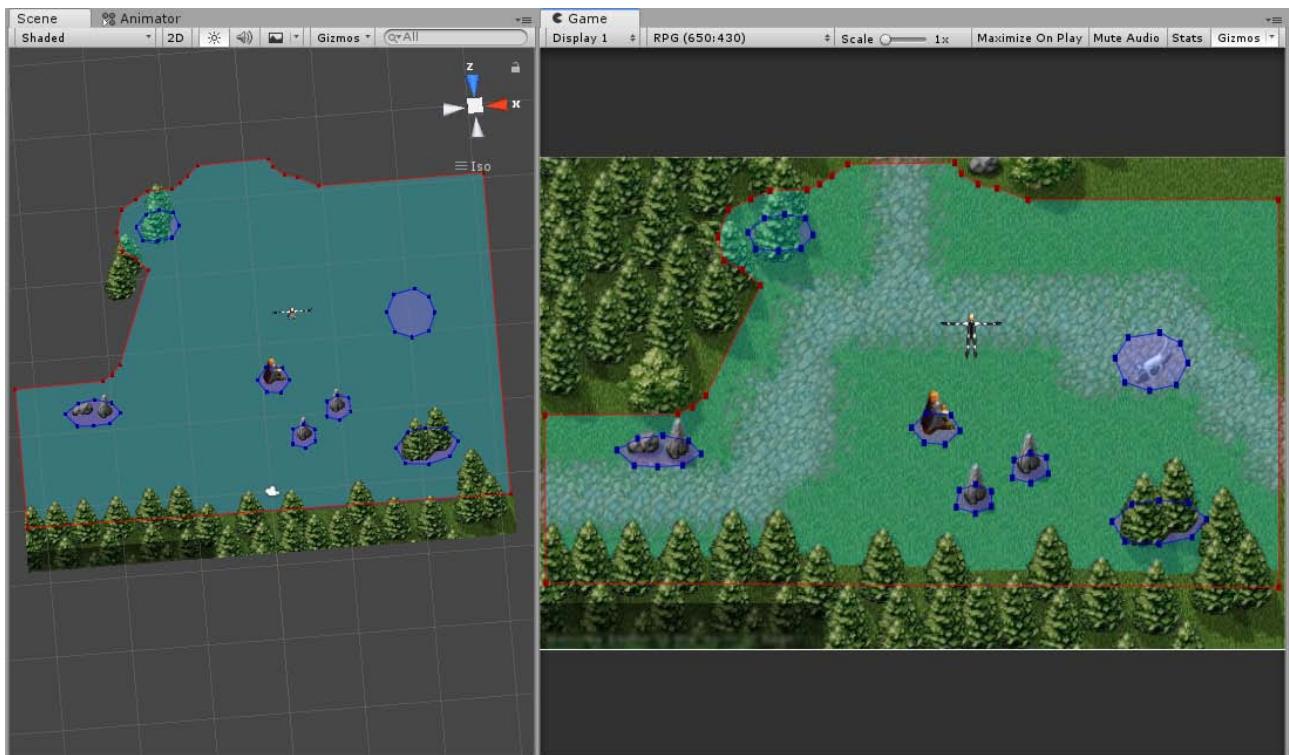
A rotation of 50 would look fine. Resize the character and move it along the scene to check its perspective. It can help to bring it closer to the dog.



Now if you want you can draw a temporary area that covers the whole image, bake it and try in Play Mode.

When you are sure settings are right, you can create areas and place sprites.

This could be an ended configuration:



## Demo scene: Pools

---

You will find this example in Demo folder.

It uses 2.5D Toolkit environment tool which requires Background Camera.

Projection is Perspective, Camera distance is set as automatic, Aspect Ratio is set to 770:577, Character walk: 2.1, Character run: 5.1 (Animator Controller IBM).



This will be a fairly easy environment to build.

We evaluate the width of the scene in the part closest to the camera of about 6/7 meters and a depth between 25 and 30 meters.

Let's lower the FOV to 40 for the moment.

We see that by moving the floor to -2.5 we get measurements very close to the chosen ones. Default camera rotation is 3 but we can reduce it a little, down to 2.

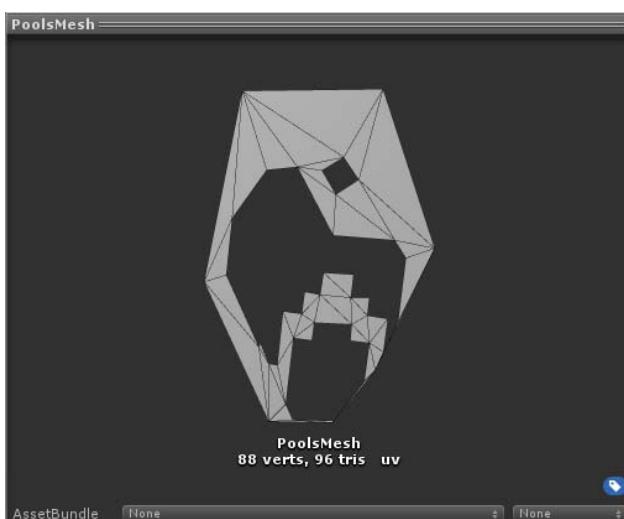
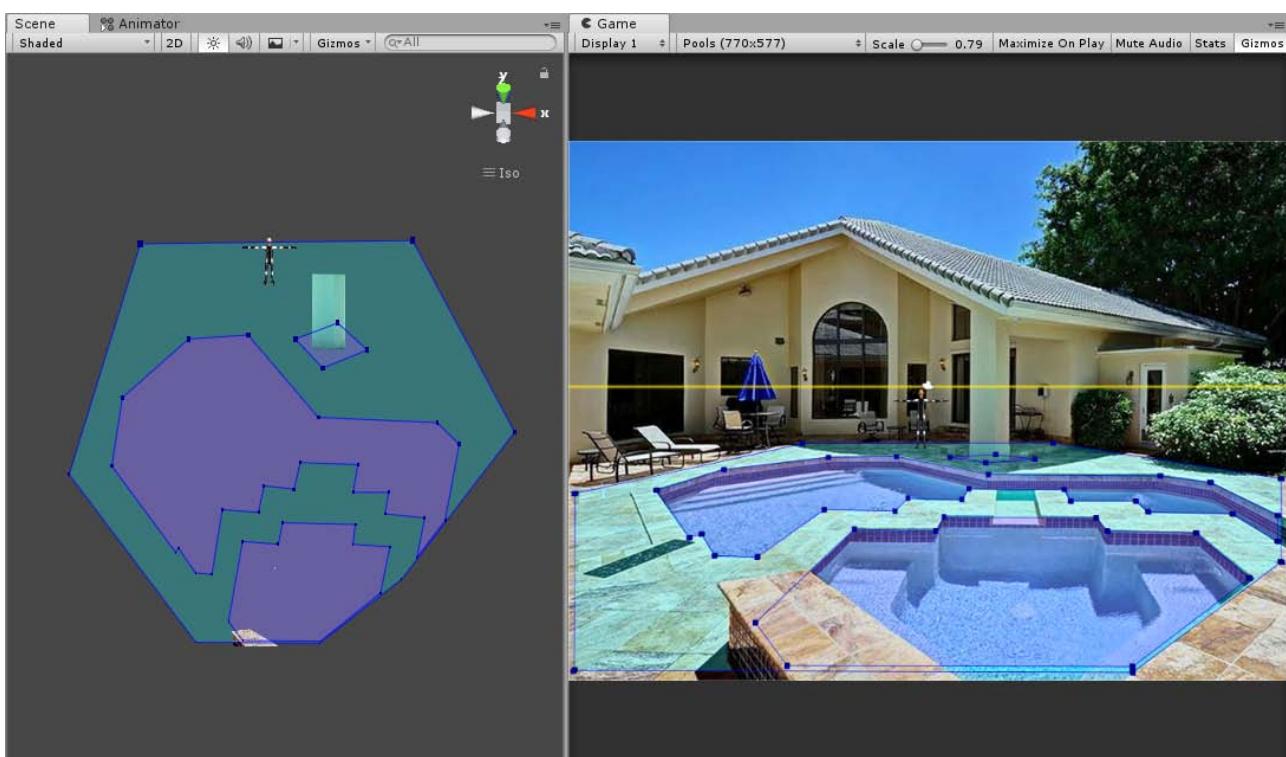
With such a low rotation, a high FOV will not cause character distortion on the sides of the screen.

We move the character to the bottom of the scene, near the chairs or the door on the right. The door has a step and we will have to take it into account when resizing it.

Now moving the character along the scene we will see that it respects the right proportions with the environment, even when it is close to the deck chairs.

Very quickly we found that with floor to -2.49, FOV to 40, camera rotation to 2 and character scaling to 1.33, our scene is ready.

Let's create the areas, place the two sprites cut out of the background image (pillar and wall) and work is ended.



Double-sided mesh has:  
88 verts, 96 tris

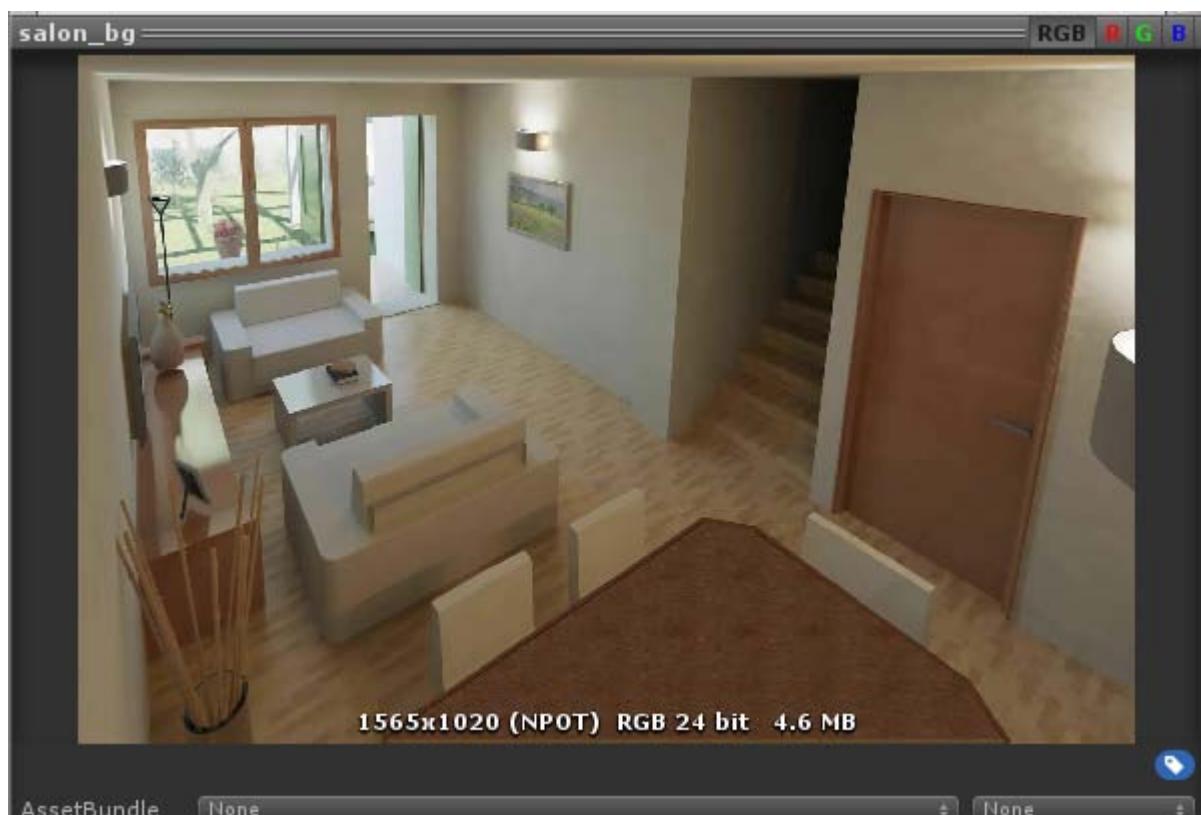
## Demo scene: Salon

---

You will find this example in Demo folder.

It uses 2.5D Toolkit environment tool which requires Background Camera.

Projection is Perspective, Camera distance is set as automatic, Aspect Ratio is set to 1565:1020, Character walk: 1, Character run: 2.3 (Animator Controller).



A similar environment is presented by way of example only, we should avoid making the character walk in very narrow points as we will do.

This image will require a little more attention. To make the scene realistic, the character will have to be aligned with the vertical lines as much as possible, in the rightmost part it will have to be slightly tilted on the right, same behavior on the left.

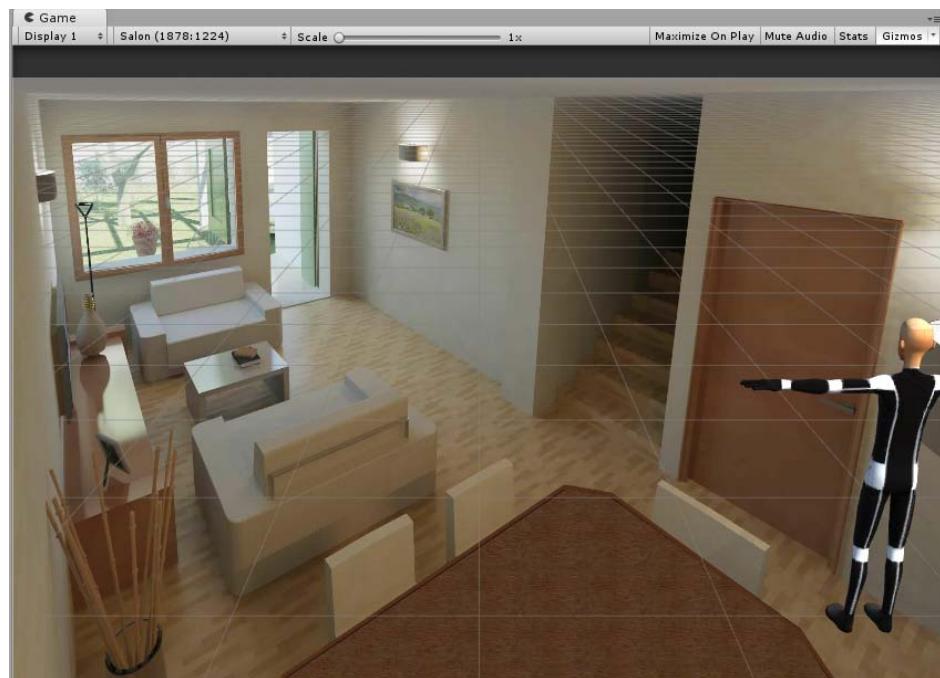
In previous demos we talked many times about FOV and distortion it tends to cause on the sides of the screen when camera rotation is high.

With this image we need this distortion.

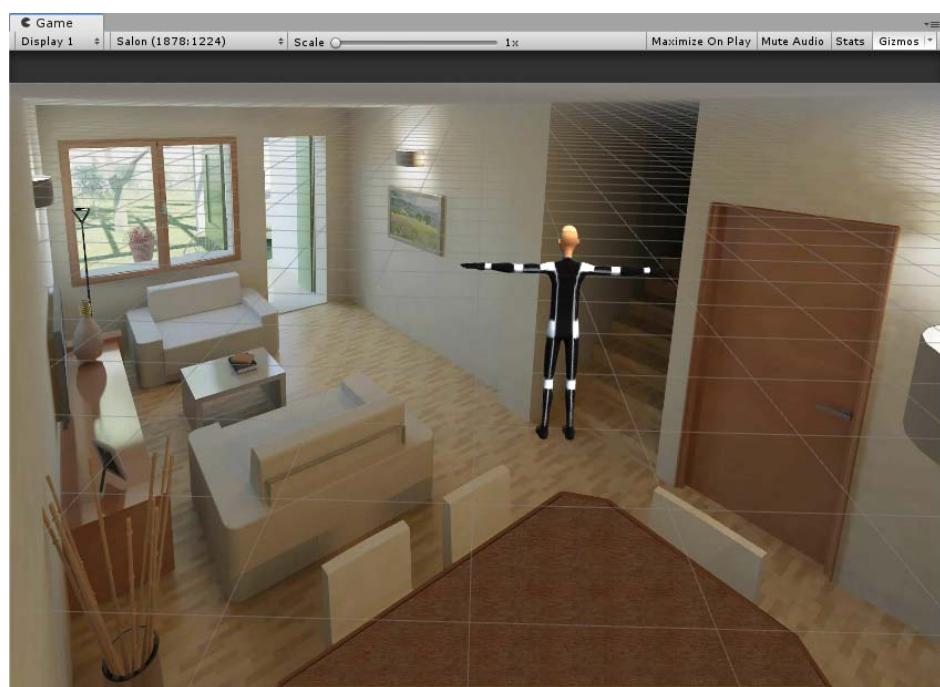
We then work on all the settings by moving the character throughout the scene and always keeping the room measurements under control.

The following could be a good solution:

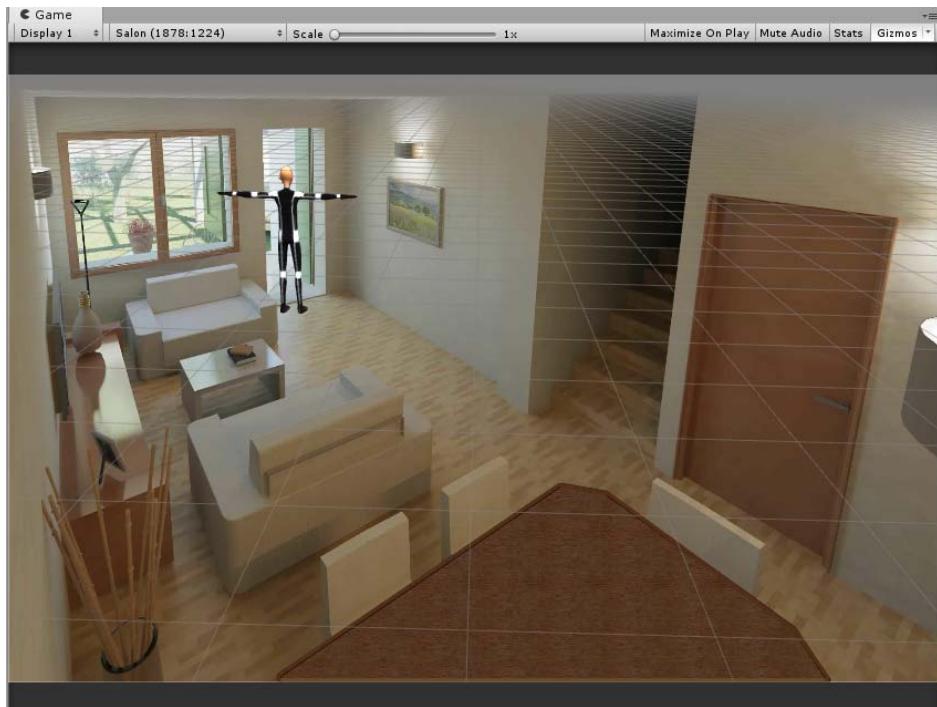
Character has a correct height and is aligned with the door



Character is aligned with the wall



Character is aligned with the exit door



Up to now we have always used only camera rotation on the X axis.

This scene required a minimum rotation on the Z axis which allowed to obtain the best alignment of the character with the vertical lines.

As always, when all the settings are right, we can proceed to create areas and to place sprites cut from the background image.

We will suddenly find ourselves faced with a strange behavior: perhaps due to an error in cropping and saving the images or because of the general settings our sprites will not overlap those of the background image.

We will therefore have to resize them manually.

In this case, the Tab Objects **<Copy scale>** and **<Assign scale>** functions helps.

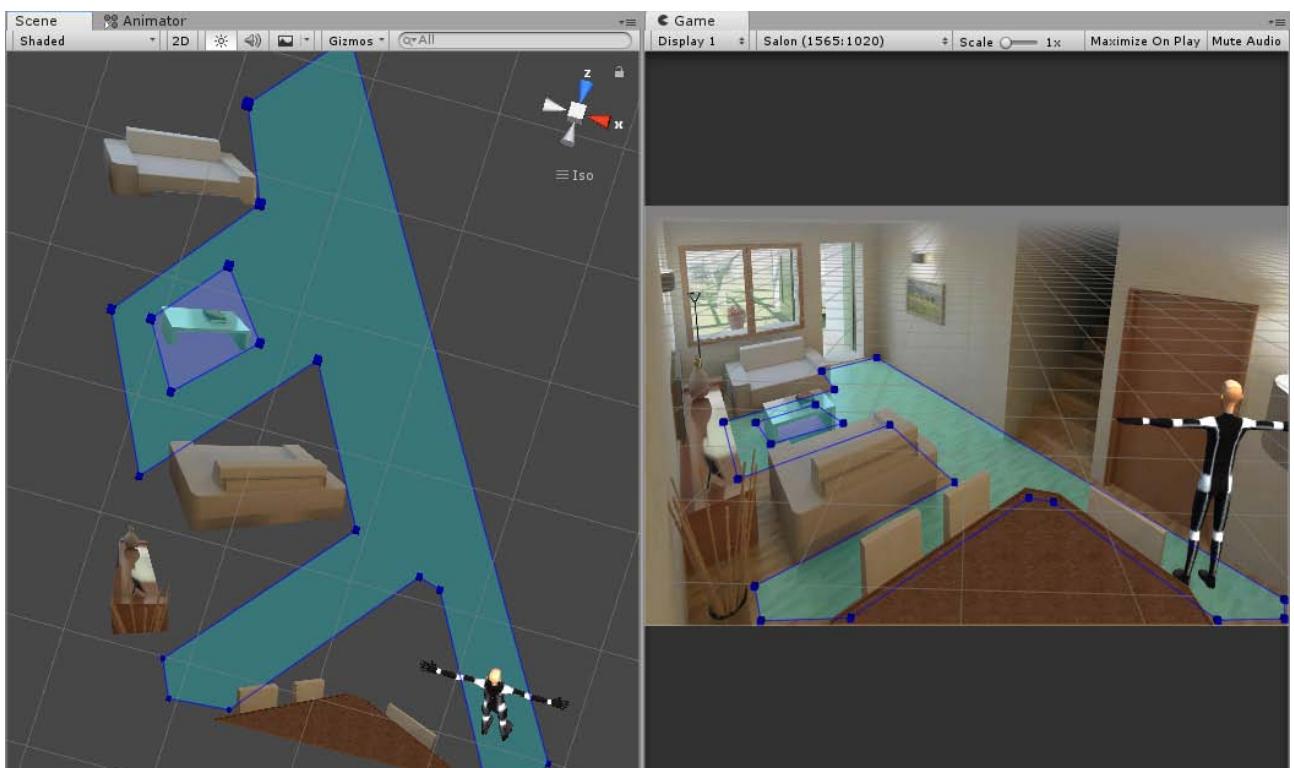
Once an object is resized with respect to the background image, click **<Copy scale>** to store the scale used.

From now on you will use the **<Assign scale>** button for all the other sprites you need to place and you will not need to search for the exact size for each one (if sprites were cut from the same image and with same Pixels Per Unit).

Due to the high rotation of the camera, placement of objects will require greater attention.

Checking in Scene View with top view, sprite must be placed in the center of the non-walkable area and then through the 2.5D Toolkit controls moved on the X and Y axis until it overlaps with the background image. Sprite must not invade the walkable area otherwise character will seem to pass through it.

This is the completed scene



Following the same procedure but using two plans in scene instead of the 2.5D Toolkit environment tool (which involves the use of only one floor) you could place an inclined plane on the stair to make the character walk above it.

By cutting out a sprite from the background image the character would disappear behind it.

## Demo scene: Villa (Multilevel)

---

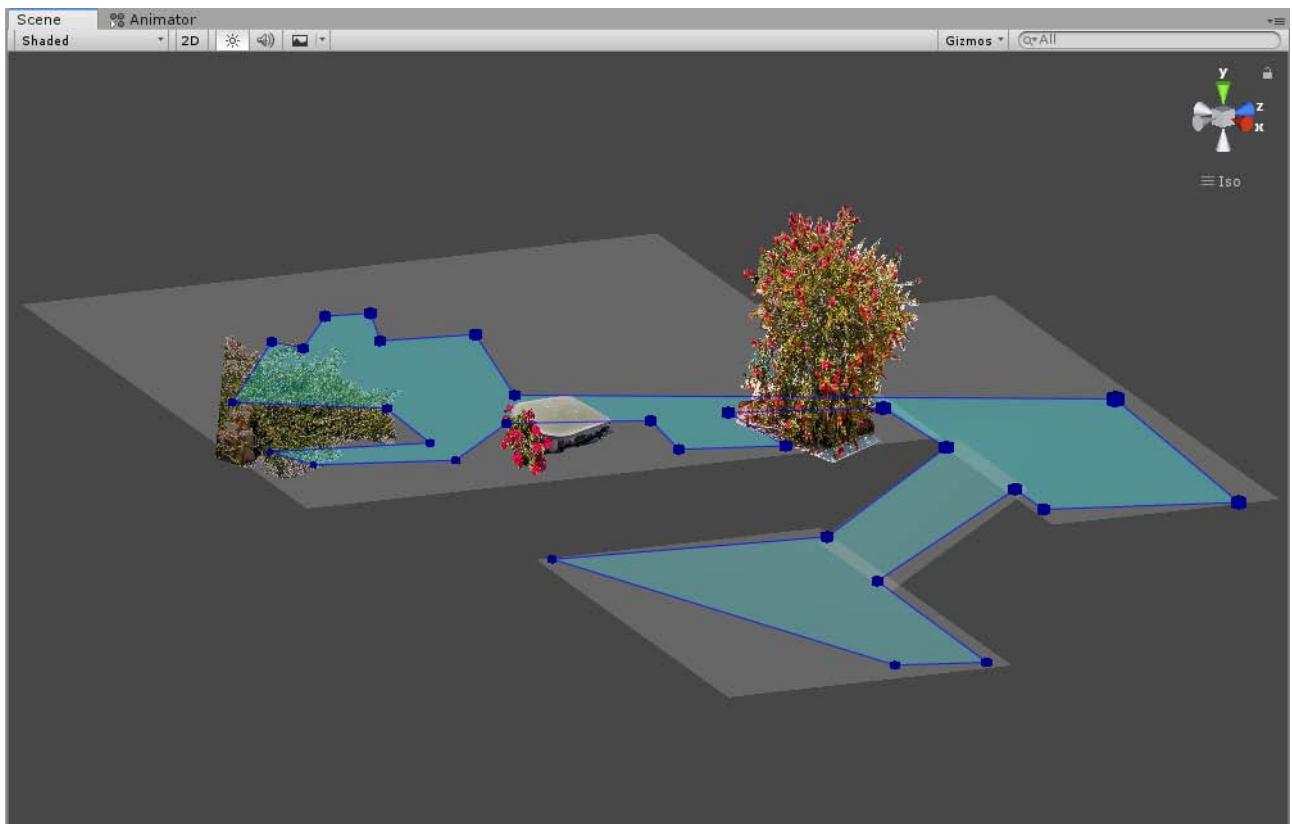
You will find this example in Demo folder.

It does not use any 2.5D Toolkit tools. Background image is in scene, four Plans were used to create walkable area working in Scene View. Projection is Perspective, Camera distance is free, Aspect Ratio is set to 4200: 2800, Character walk 1.8, Character run 3.3 (Animator Controller).

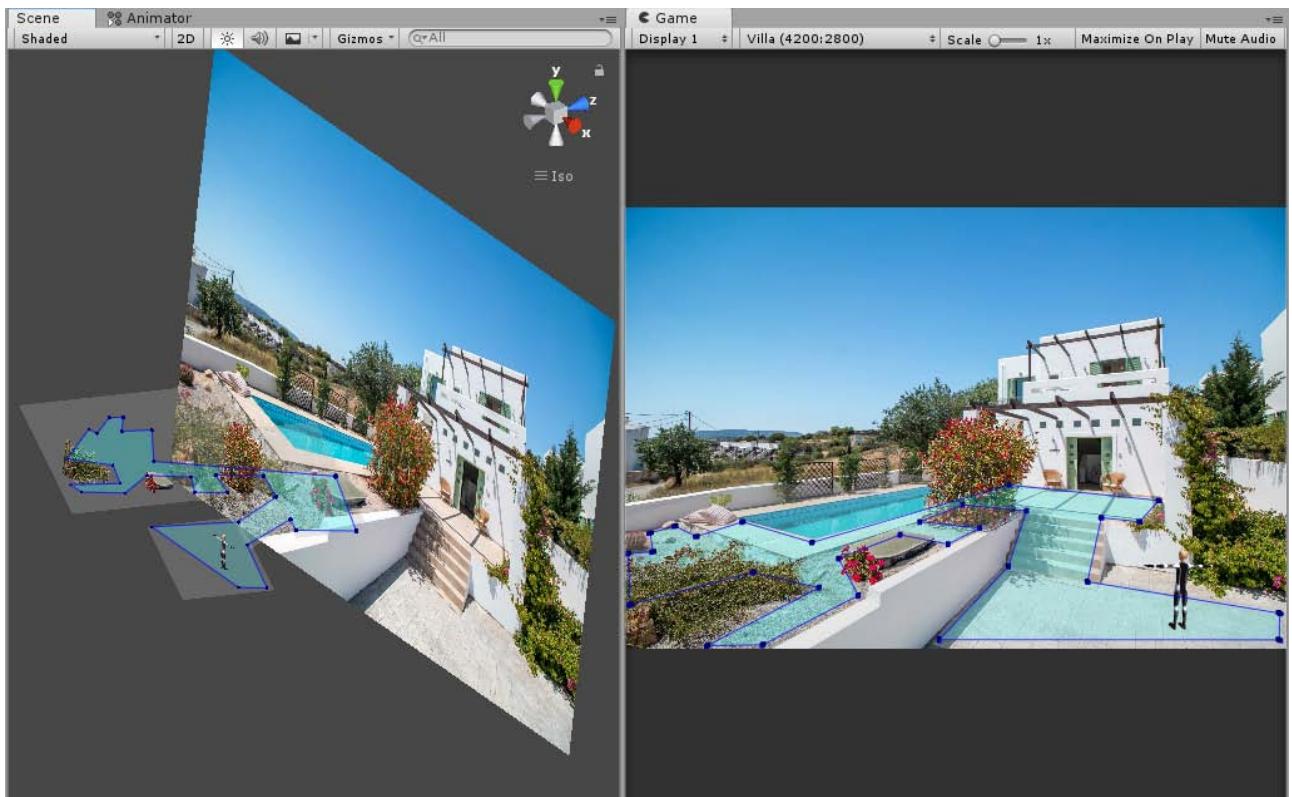


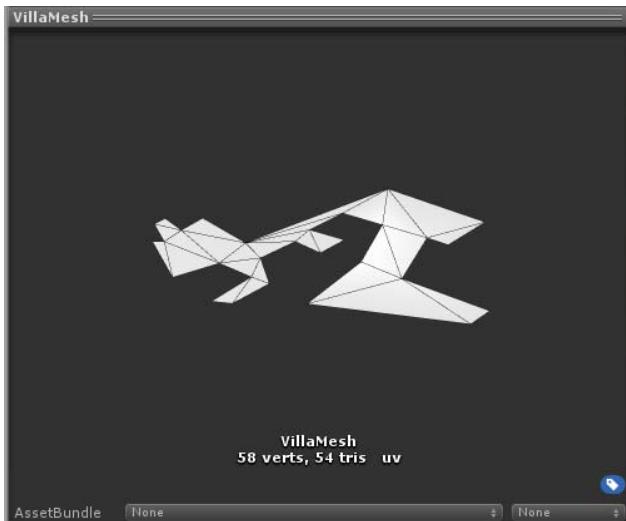
The Meter tool has not been used and all settings have been defined by trial and error.

Not having used automatic camera distance, sprites requested to be resized.



This is the end result:





Double-sided mesh has:  
58 verts, 54 tris

# A way to use Meter and Cube

---

Cube can be used in different ways depending on whether orthographic or perspective projection is used.

We will see how to use it to try to guess FOV and camera rotation of a 2D image.

[This video shows steps to follow.](#)

This is the image we will use:



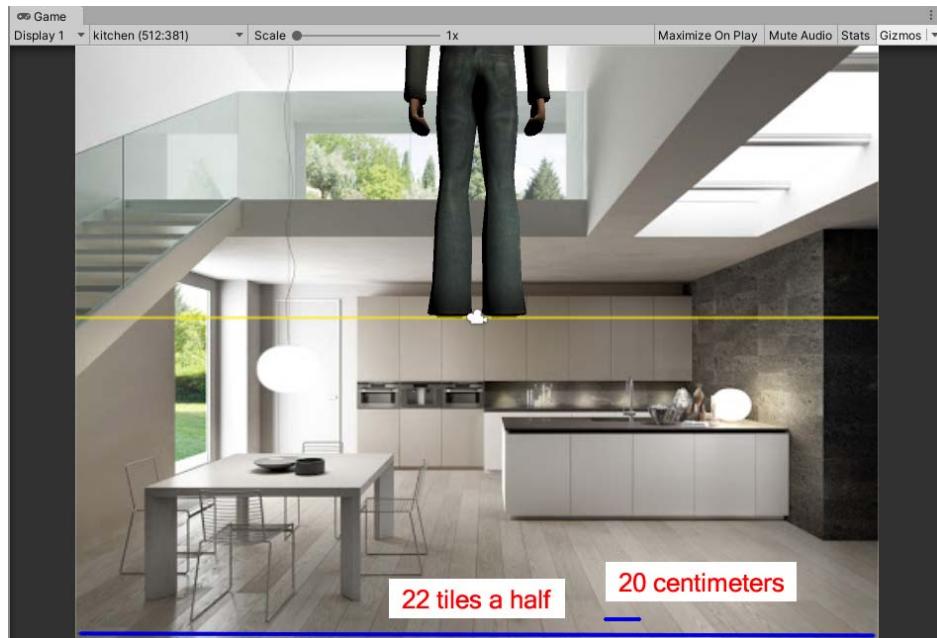
Camera distance is managed by 2.5D Toolkit so sprites cut from background image will not require to be resized.

Let's look at the picture: kitchen, seen from the top, would appear square so depth should be very close to its width.

Floor is covered with tiles that are generally about 20 centimeters wide and we will use this measure to define scene width in the area closest to the camera.

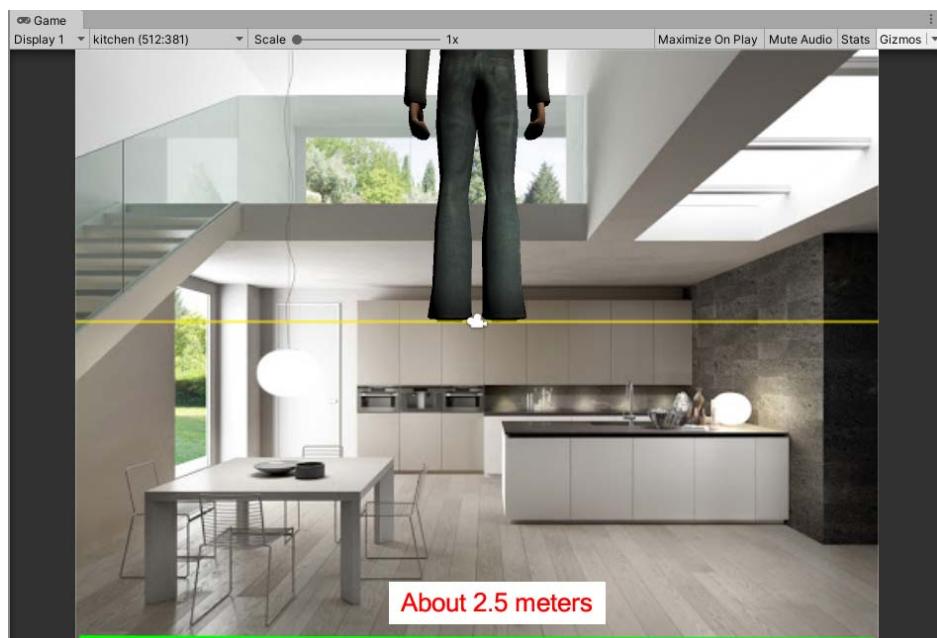
Tiles are 22 and a half so scene width should be 4 meters and 50 centimeters.

$$22.5 \times 20 = 450.$$



Scene default settings are: Floor -1, FOV 60, camera rotation 3.

First of all let's take a first measurement

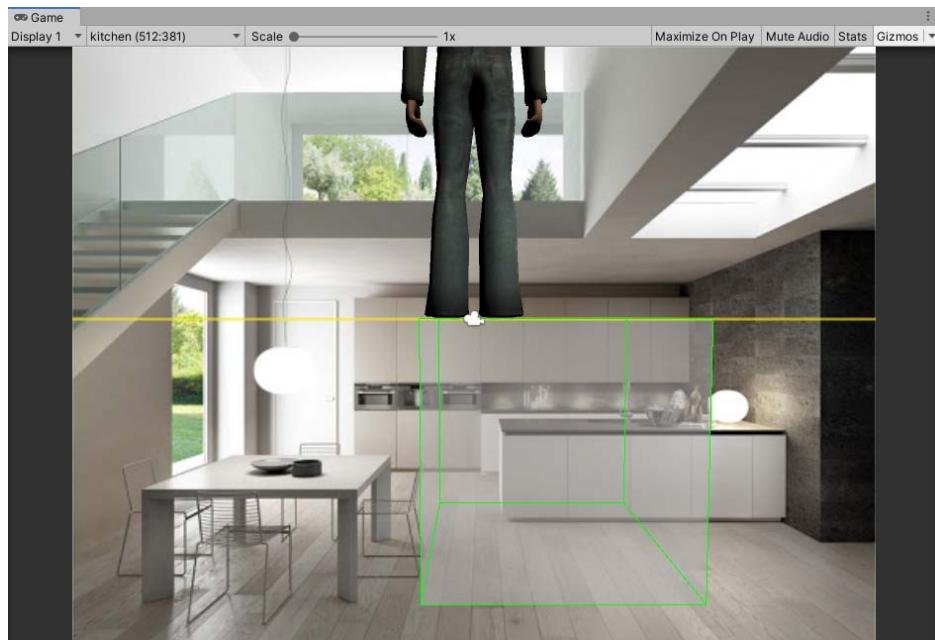


About 2.5 meters while it should be 4.5 meters. We lower floor.

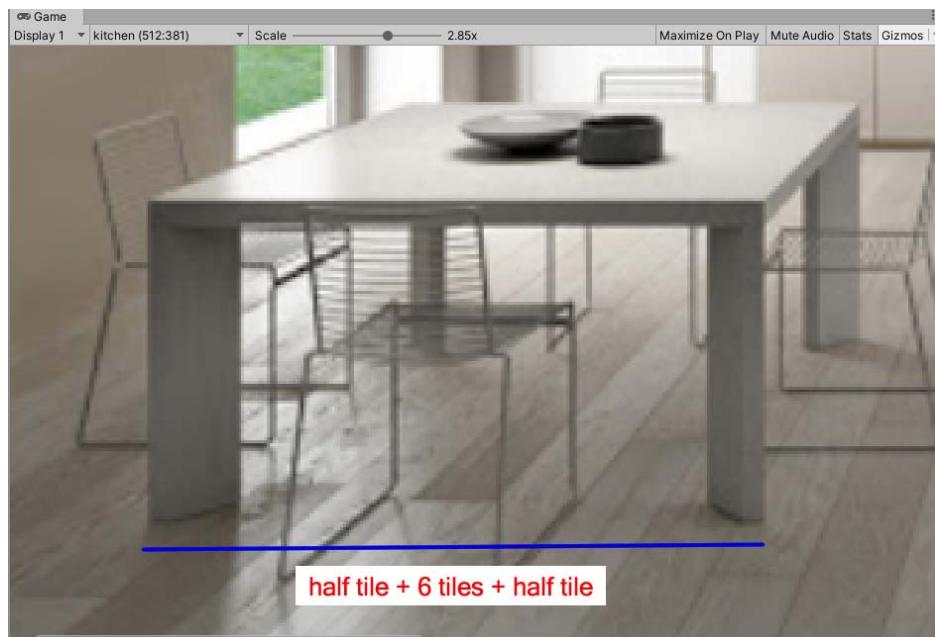
With a floor set at -1.8 we have reached the desired size. It may change with the next settings but this way we start with reliable values.

Now click on **<Use cube>** and **<Place cube>**. We will use table as a reference object.

Click **<Transparent cube>** and **<Don't keep proportion>**. This way we can overlay cube on table trying to guess right settings for the camera.

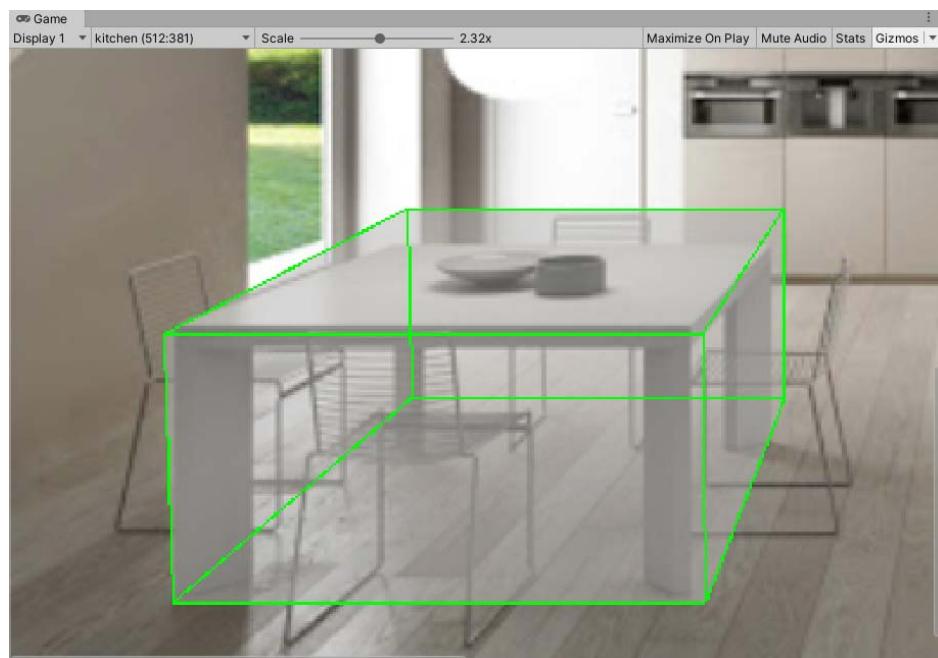


We know that each tile measures 20 cm. and zooming image we see that table has a width of 6 tiles + half tile for each leg.

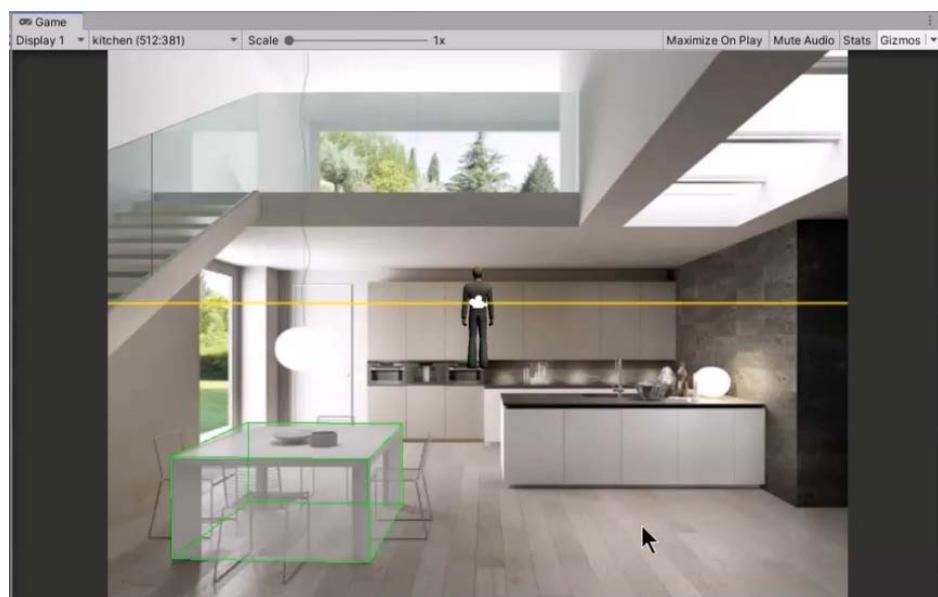


Table, which appears square, has a size equal to 140 centimeters per side ( $6 * 20 + 10 + 10$ ). We resize cube on the X and Z axis to 1.4 and give it a height equal to that of the table (approaching cube to the table it will be easy to set its height).

With the current setting we see that cube does not follow the edges of the table, so we will change FOV and camera rotation.



After some tests we get a satisfactory overlap



You don't need maximum precision, a minimum distortion of the image will be enough to not allow to get a perfect overlay.

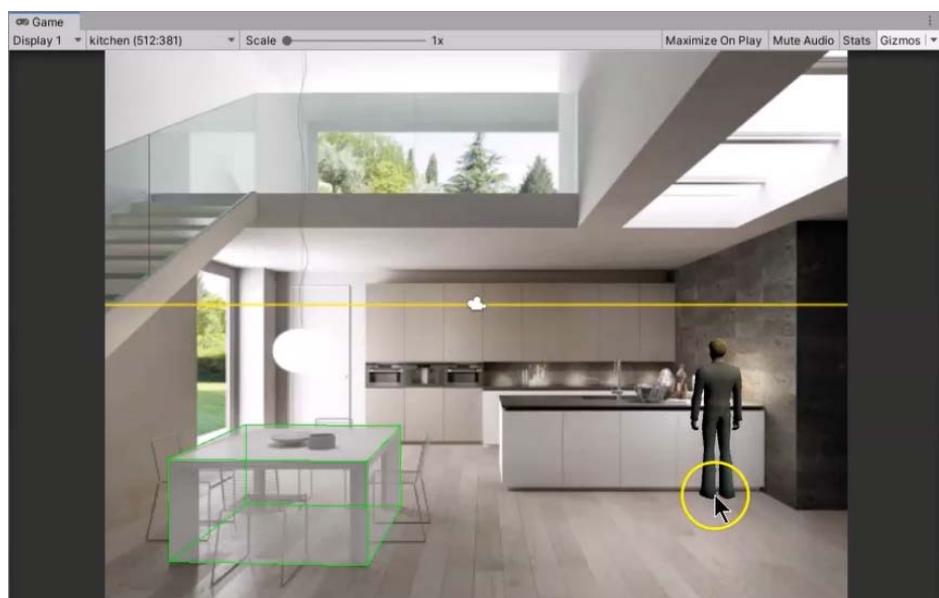
If we now measure width and depth of the scene again we will get about 4.50 meters for both which are the established sizes.

If scene width has changed too much and ratio between width and depth is right, just move floor.

The last thing to do is to resize character.

Bring it close to the table or any other object that you can take as a reference.

If FOV and camera rotation are reliable, character will maintain a correct proportion in every point of the scene.



Now you can build areas and place sprites.

If you want to try setting this scene you will find this image in Demo/Graphics folder.

# Add shadow

---

To get shadow, two different configurations are possible according to the environment used.

If you use Background Camera it will be very simple.

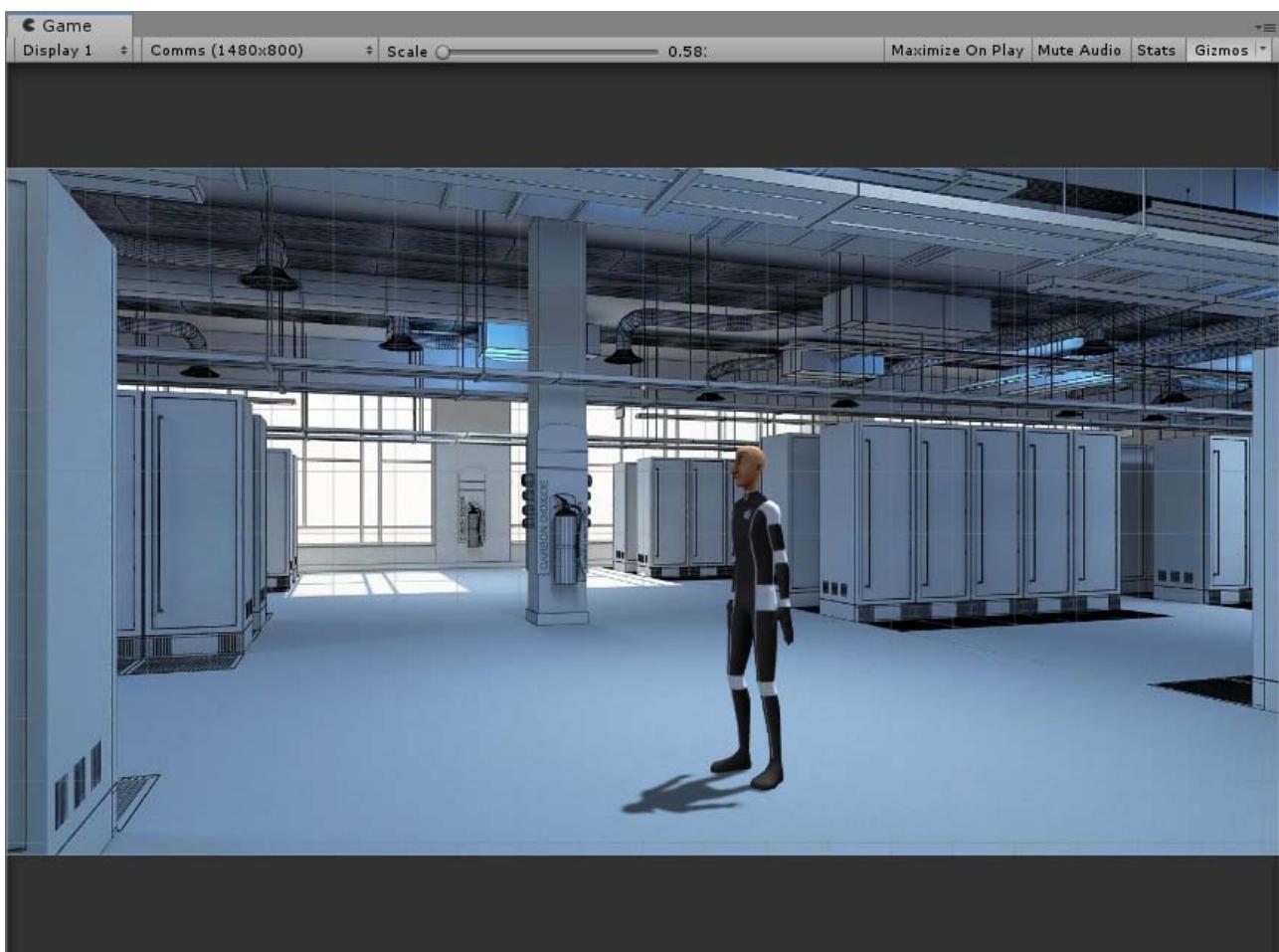
Once completed the scene creates a mesh [Tab Mesh] <Create mesh> then <Save mesh>. Drag new mesh from 2.5dToolKit/Resources/[scene\_name] in Hierarchy.

Select mesh and drag from 2.5dToolKit/Shading:

- MatteShadow material to Element 0 of Mesh Renderer
- CustomTransparentShadowCollectorVal on Shader of MatteShadow

For maximum performance select Directional Light and set Resolution of Realtime Shadows on Very High Resolution.

(See below for screenshots).



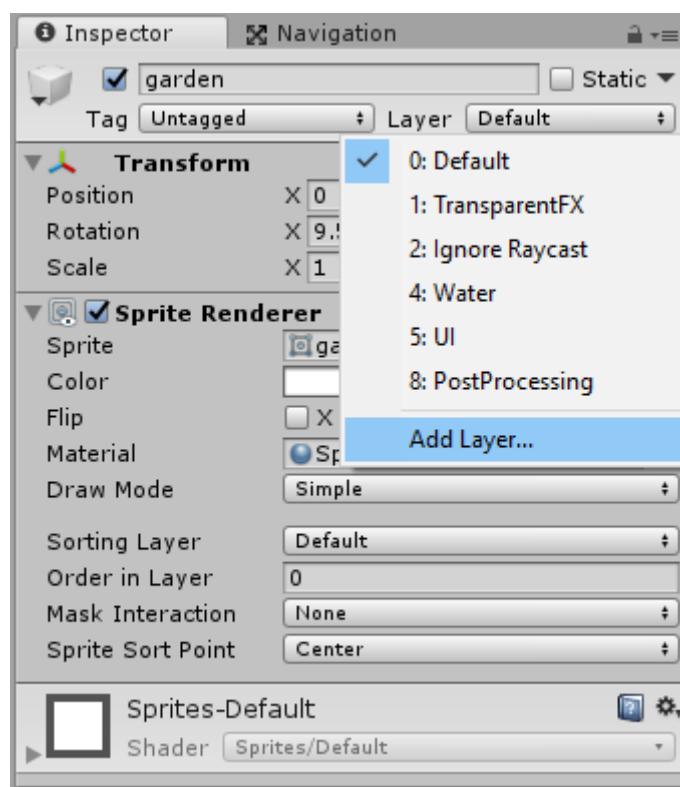
**Save mesh before changing its material otherwise Unity raises an error trying to save it later.**

If you use the background image in scene, you need to create a second camera and follow a somewhat long process.

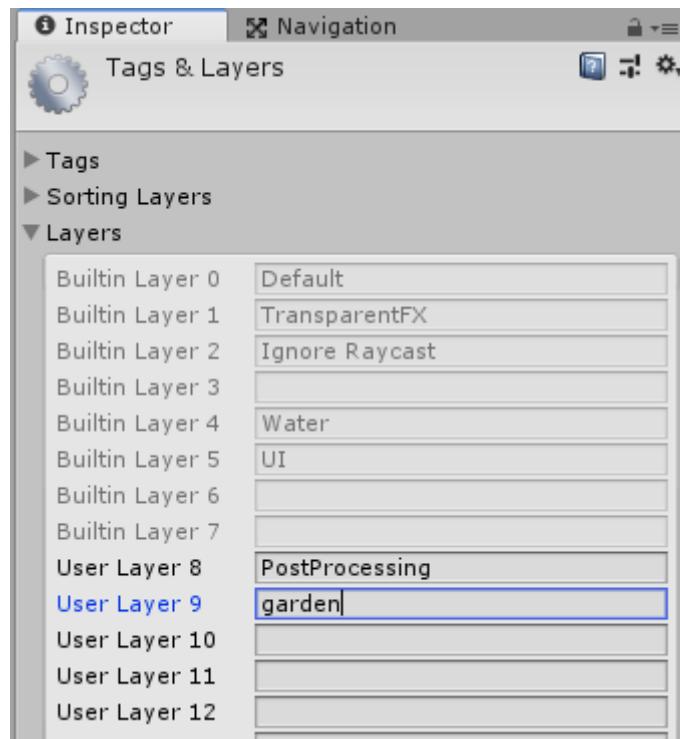
Create a new camera (Hierarchy > Create > Camera) and name it “Background Camera”.

Select background image (in this example it is called “garden”) in Hierarchy and in Inspector click on the Layer dropdown menu.

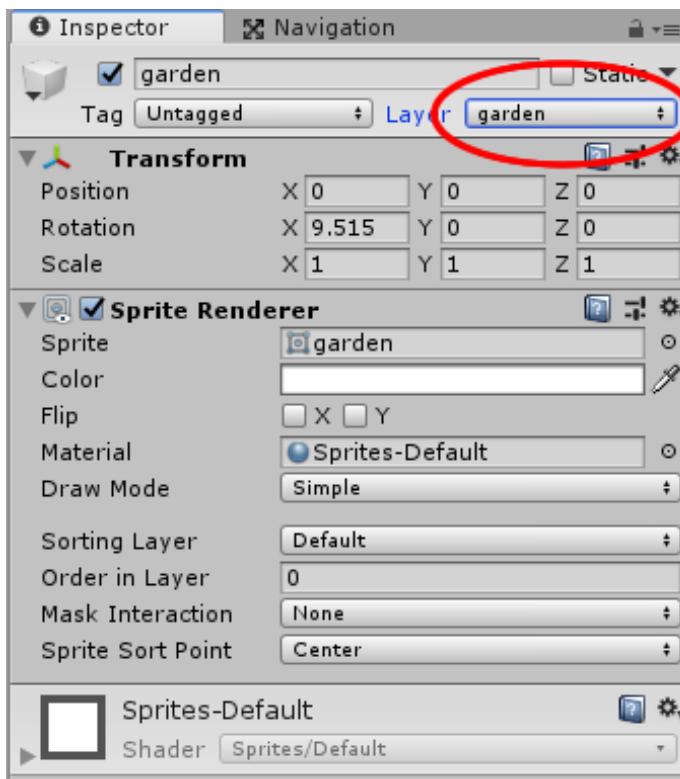
Select Add Layer.



In the first free User Layer slot, type "garden"

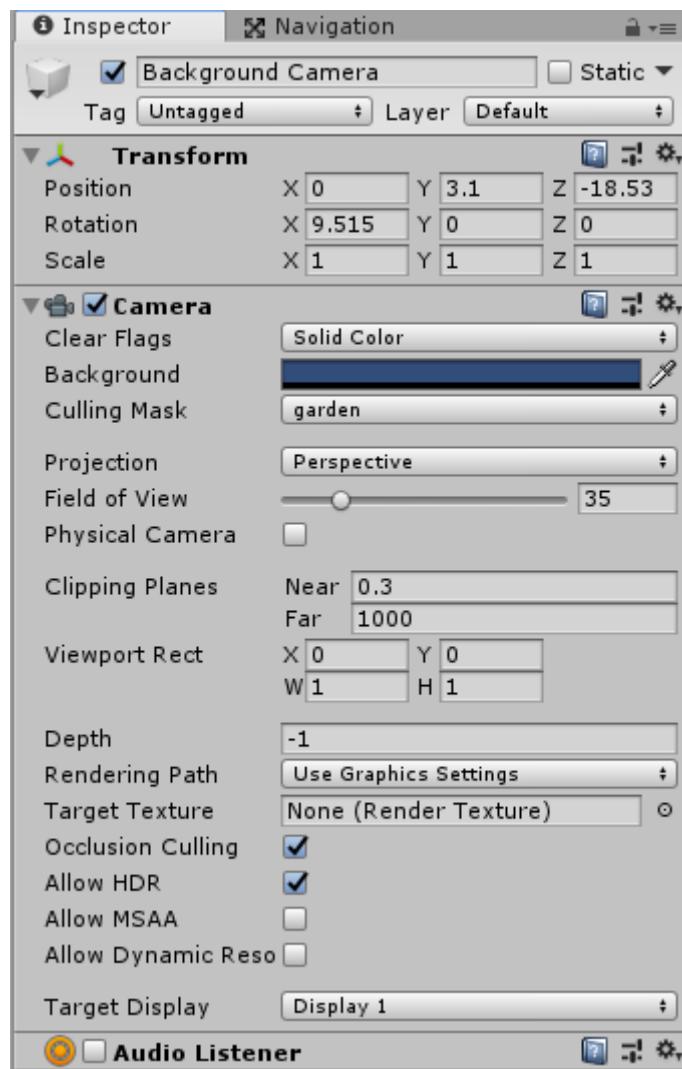


select the image again and assign this layer to it.



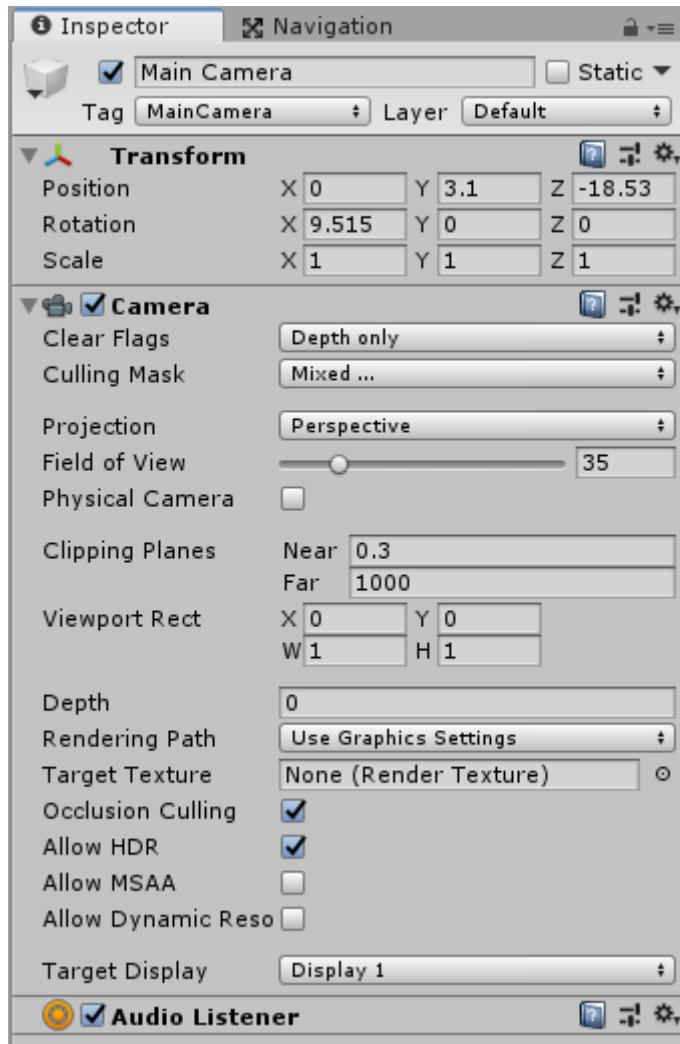
Now select your Background Camera and change settings:

- Set same values as the Main Camera (you can use Copy / Paste Component)
- Uncheck Flare Layer, Audio Listener and GUIlayer (Legacy in Unity 5)
- Set Clear Flags to Solid Color
- Set Culling Mask, first to Nothing then to “garden”
- Set Depth to -1



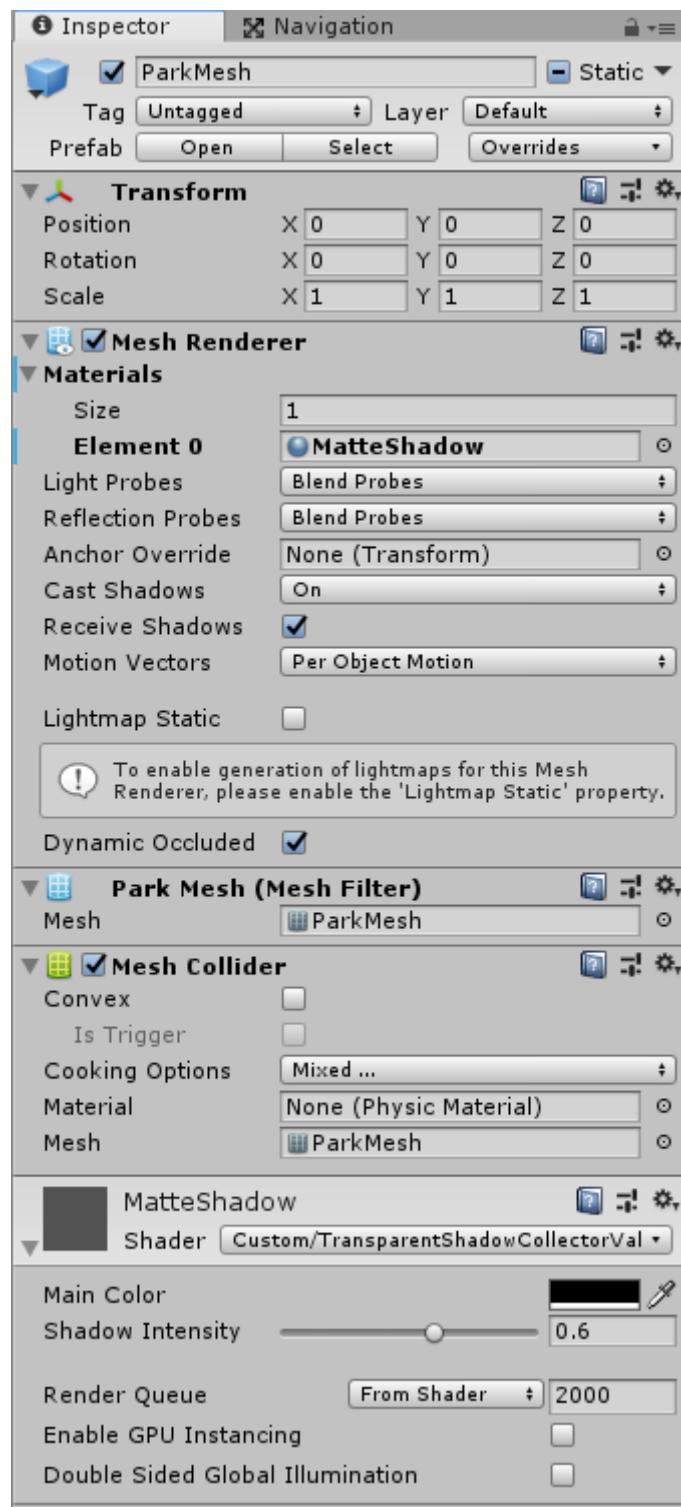
Select Main Camera.

- Set Clear Flags to Depth Only
- In Culling Mask uncheck “garden”
- Set Depth to 0

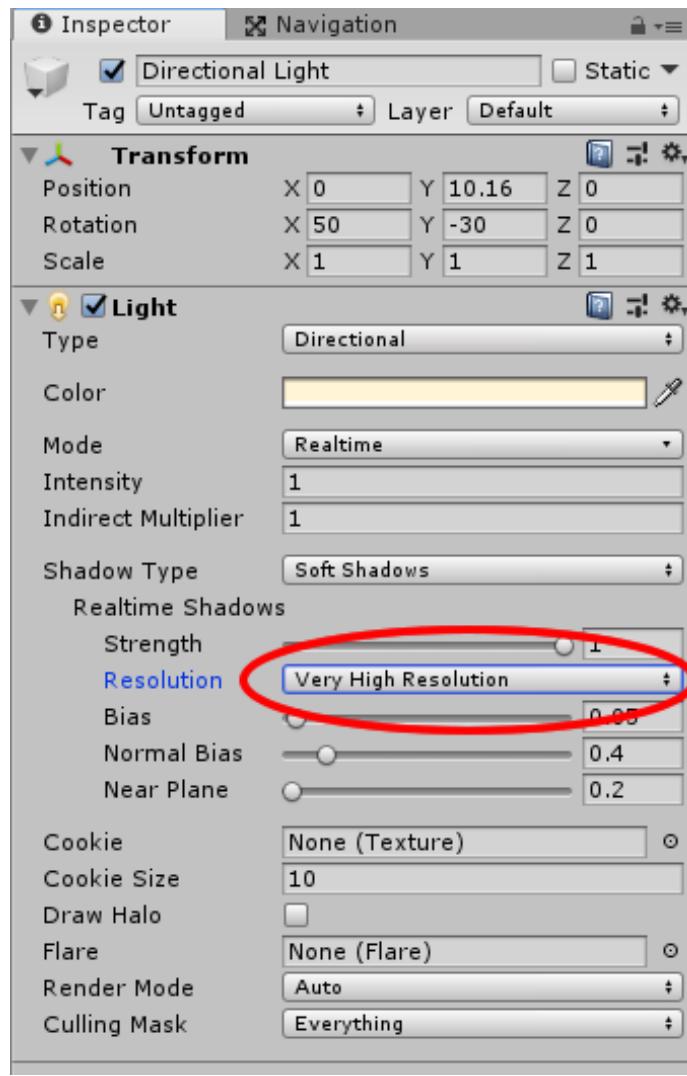


Select mesh in Hierarchy and drag from 2.5dToolKit/Shading:

- MatteShadow material to Element 0 of Mesh Renderer
- CustomTransparentShadowCollectorVal on Shader of MatteShadow



For maximum performance select Directional Light and set Resolution of Realtime Shadows on Very High Resolution.



Move light to get lighting you prefer.

**Save mesh before changing its material otherwise Unity raises an error trying to save it later.**

If shader does not work you can try the others.

Keep in mind that materials for obtaining shadows are for demonstration purposes only. You can search for better ones or write your personal ones.

# Move whole environment

---

Depending on whether you are using background image in scene or Background Camera you can define a new position in the 3D space for your environment by setting a new Main Camera or background image position. All Gameobjects in the Hierarchy will move accordingly. Make sure they are visible. If you want to avoid moving some objects you can hide them.

**Caution:** do this only when scene is complete. After that, you will no longer be able to edit scene with 2.5D Toolkit tools. Areas will be deleted.

**Keep in mind that if you use a GameObject as a folder initially set to (0, 0, 0) its transform will be modified and the objects contained in it will be processed as children (their new position will therefore be relative to the Parent).**

