by Pixtim

for UNITY

QUICKLY CREATE CUSTOMIZABLE GAME LEVELS

CHANGE THE MATERIALS ON YOUR ENTIRE SCENE IN A FEW CLICKS

WITH MONOBLOC UV UNFOLDING CUSTOMIZE YOUR SCENE WITH YOUR OWN TEXTURES

Screenshots UNITY

MORE THAN 130 3D OPTIMIZED ELEMENTS

47 MATERIALS

85 TEXTURES & NORMAL MAPS

First of all thank you for purchasing ***makeYOURlevel***. We spent a lot of time optimizing this package and we sincerely hope that you will be satisfied.

makeYOURlevel is designed to allow you to rapidly design indoor environments while offering the possibility to customize them.

Thus, all 3D models included in the package are designed to be distributed on a 3x3 grid. Whether corridors, rooms or secondary elements, simply arrange them according to this measure.

In addition to a large bank of prefabricated elements, we have added libraries to help you build any customized element you might need. These basic elements are often very simple but are also pre-positioned with an appropriate UV unwrapping.

On the textures, you can not only change all the textures of corridors and rooms in a few clicks (even if you have already created your level),you can also use textures of your creation. To do this you must simply design your tiled textures in a square format.

We ensured that the number of triangles used per element is minimal and perfectly optimized.The textures that we provide are mostly in full size (2048x2048), you are free to reduce them as needed.

To accompany ***makeYOURlevel***, we designed this manual to guide you in its use and provide some advice to those starting with Unity. The process also ensures a quality equal to our sample images.

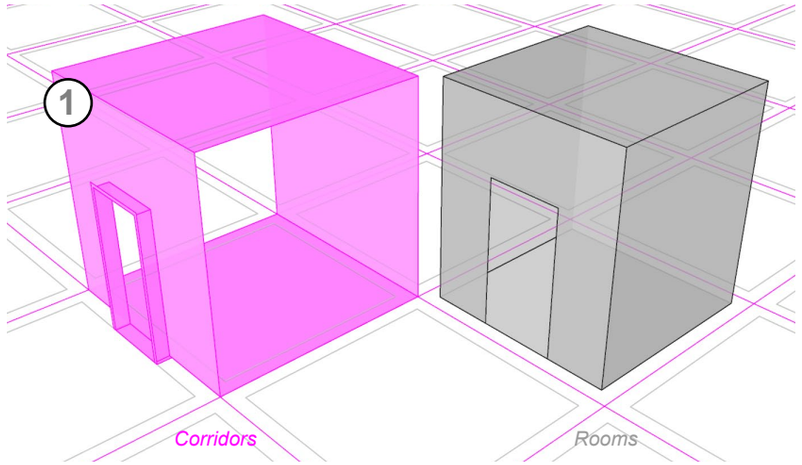
Table of Contents

makeYOURlevel is a package of 3D objects and textures. Some items come with predefined behaviors for user comfort during their tests. These are provided free of charge and shall not in any way be subject to claims.

1. Basic principle

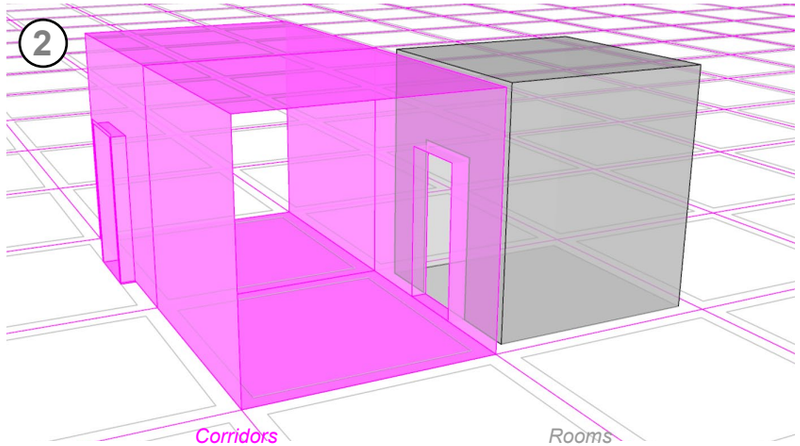
Imagine a virtual grid in your Unity workspace. This grid uses the software's basic unit and each tile measures 3x3 units. All the items we offer in this package will be adapted to this guide.

We thought for a while about integrating a tool to allow an easier distribution of this set of "bricks", but finally, using the software's interface will allow you to more easily include your own items.



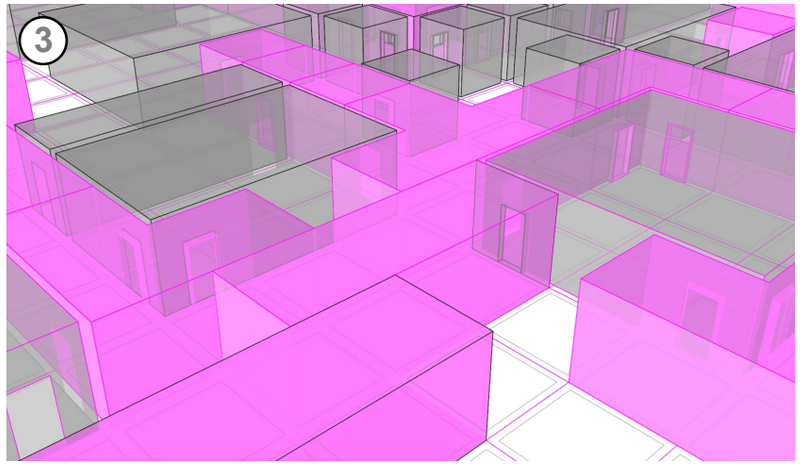
Important: Corridors measure all 3x3 units, but Rooms provide a withdrawal of 0.2 to give the walls thickness.

This does not influence at all the distribution of these two types of elements in 3 x 3 increments. This does mean on the other hand; that if you place two corridors closed on an overlapping side, the separation wall will have no thickness at all.
On this first diagram, you can also see that the corridors are equipped with a junction where there is a door location but not the adjacent Rooms. These items are made to "fit" together.

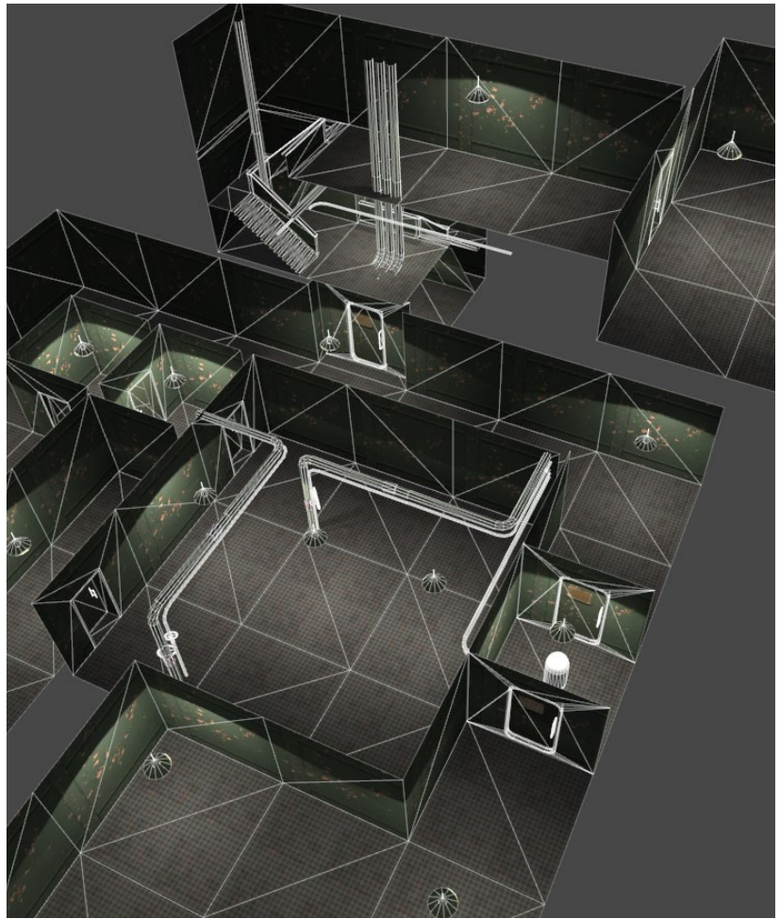


The second graph shows the proper way to position the two types of elements.
Corridor elements are placed side by side with the open sides facing each other, and for the door openings, we place a Room element that has a correctly placed door opening as well.

By positioning the elements in this fashion, and by turning them in 90 °increments to orient them correctly, you will soon obtain complex assemblies.



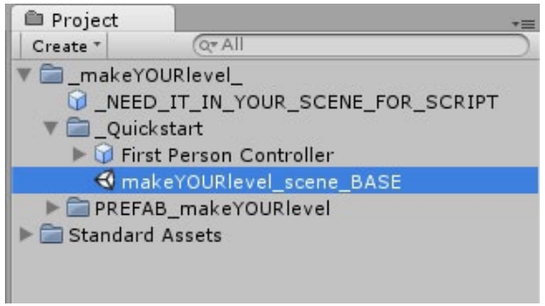
15 Corridor prefabs and 16 Room prefabs are available, but we've also included decomposed elements that will help you make any customized prefabs that you might need. We'll talk about this a little later in the manual.



2. Package Organization

2.1 The Package folders

When you import the ***makeYOURlevel*** package in a blank project, you will get this:



The folder "_makeYOURlevel" includes nearly 130 3D elements, 47 materials and 85 textures & normal maps.

In the sub-folder "_Quickstart", we have included a scene ready to receive items from makeYOURlevel. Nothing complicated, just a classic "First Person Controller" with a few adjustments in behavior and a script allowing it to push the makeYOURlevel prefab doors. Also in the scene, a perfectly vertical directional lamp, which can be used to highlight the contours of normal maps in certain configurations. This scene also includes the element "_NEED_IT_IN_YOUR_SCENE_FOR_SCRIPT".
This file is required in your scene in order to run the scripts we've included. These include a few functions so you can test your game levels directly. These scripts can open the doors and operate an elevator. These scripts are not intended to be kept in your final project, we have graciously integrated them to the package, but we do not guarantee their optimization.

The folder "Standard Assets" also appears. It contains the scripts of the "First Person Controller" and a few particle elements that are used by certain light prefabs.

2.2 The sub-folders

Inside the sub-folder "PREFAB_makeYOURlevel", we organized the item categories by order of use. Then, within these categories, items are organized by type.

3. Design a level with makeYOURlevel

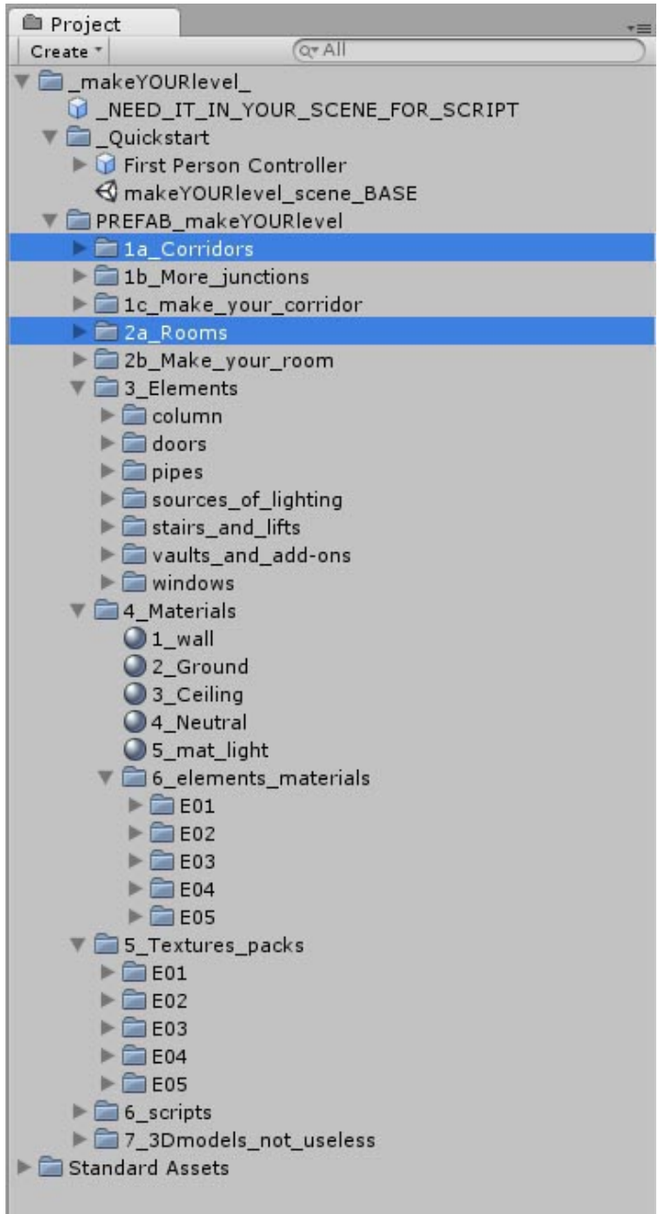
3.1 Getting Started

Duplicate the scene in the folder Quickstart, create a neutral object (GameObject menu) and call it something like "Level_0".
Position it at coordinates x = 0, y = 0 and z = 0.Drop your first Corridor or Rooms elements in this neutral object and position them as explained above.Important: if at this moment, you want to change the textures applied to objects, jump ahead to the "Changing the materials on the entire level" section.

3.2 Additional Connections

While installing the corridors and rooms, you may want to install two rooms side by side, or make a room communicate with the big opening of a corridor. Junction elements are made available in the "1b_More_junctions" sub-folder.

For the simplest usage, select the Corridor or Room object to which you wish to add this junction and place the junction object as child. In this way, Unity will take into account the position of your Corridor or Room object and the position of the junction object will correspond. The last step will be to rotate the junction object to the appropriate angle.



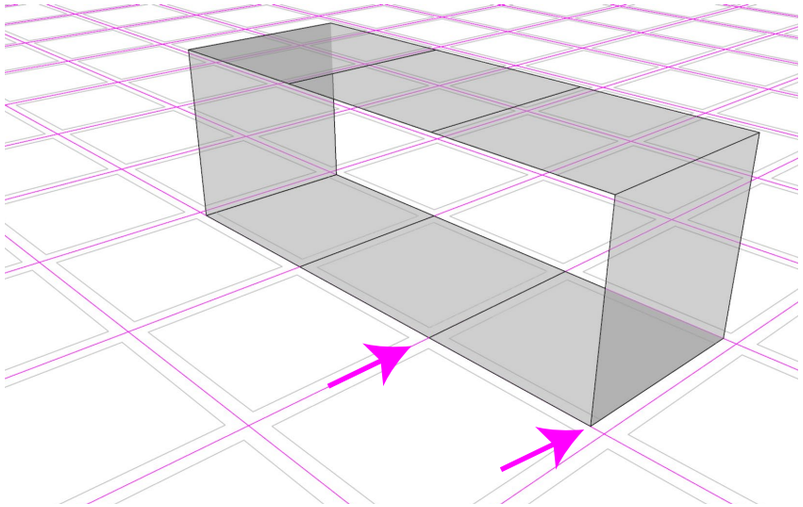
3.3 Making additional Corridors and Rooms

When we started working on ***makeYOURlevel***, we wanted to allow as many customization options as possible to the users.

To do this, we have included in the package elements that may seem simplistic but that will allow you to create Rooms and Corridors of any shape and size,

You can then include them as a prefab to your package.You will find all these "bricks" in the sub-folders "1c_make_your_corridor" and "2b_make_your_room". The elements are also organized by type.
To be able to integrate them easily, our suggestion is to plane a central piece (example : "room_floor_and_ceiling") then drop in all the other sub-elements as children. This way, you will keep the 3x3 proportions, and will easily be able to place them and apply the desired rotation.

Important: The open sides of the Room Bricks are on the edges of the 3X3 boxes while the sides closed by walls are -0.2 units from the border.



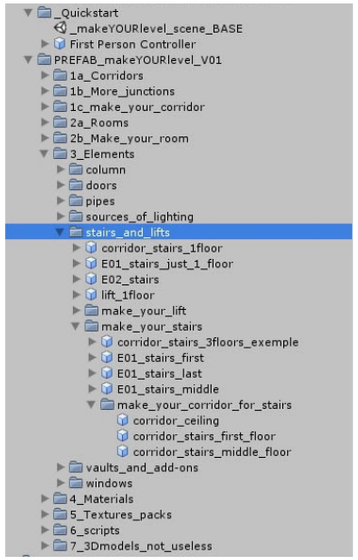
3.4 Integrating floors

First of all, just as you created a null object that we called "Level_0" to contain the entire ground floor, create a null object and name it "Level_01". ***MakeYOURlevel*** elements are inscribed in a cube of 3x3x3 as a matter of proportion.

Important: To make sure that the ground of your floor has a thickness and matches with the stair and elevator prefabs, place the null object "Level_01" at x = 0, y = 3.2. z = 0.
For each sub-sequent level you add, increment this value by 3.2 in y.Thus, the first floor will be 3.2, the second 6.4, the third 9.6, etc. ..

3.5 Use of stairs and elevators

In the sub-folder "3_Elements" you will find different categories including a folder called "stairs_and_lifts".
To create one story-high staircase, start by positioning the element "corridor_stairs_1_floor" in the null object "Level_0" and drop in a staircase element, this one will position itself automatically.



To create stairs that span multiple floors, you will find in the "makes_your_stairs" folder enough elements to create the desired Corridor and the 3 parts of the staircase. One for the ground floor, one that you will multiply by the number of floors, and finally the landing on the top floor.

For the elevator, it's the same principle except that there is no need to create a corridor, elements by type of floor are already made. We advise you to start directly with one of the prefab examples.

Important: the elevators are a bit special because they include a script and colliders.

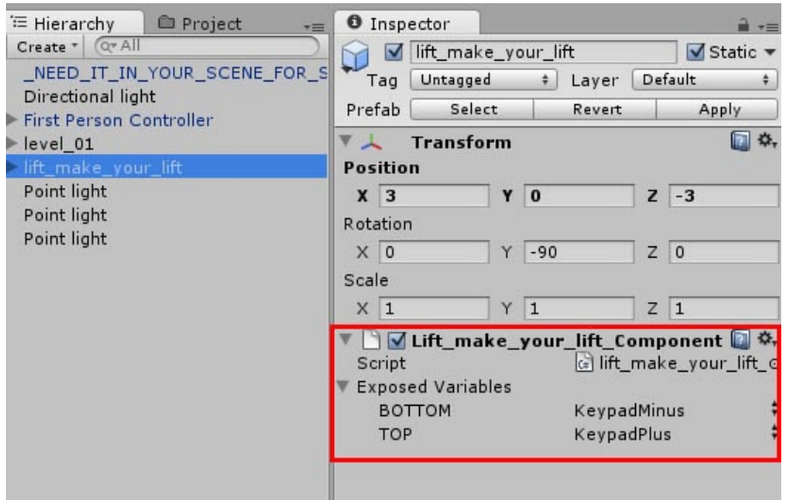
Here is the list of functions and settings:

_ The elevator detects the floor where the player is and will be automatically positioned at the same level as him.

_ The double sliding doors opens when the user approaches it.

_ Keyboard keys + and – (by default) are used to raise or lower the elevator without exceeding the number of floors you have built._If you want to change the keys assigned to control the movement of the elevator, select the lift module, in the Inspector, expand the script window "lift_make_your_lift_Component", and change the assigned keys in the Exposed Variables.

(We remind you that these scripts are delivered for the convenience of testing your game scenes and we do not guarantee proper operation under all circumstances or optimization. To remove them, remove the Components elements and also remove the "_NEED_IT_IN_YOUR_SCENE_ FOR_SCRIPT" element of your scenes)



3.6 Using the secondary elements

Doors, windows, lighting, ceilings, pipes and other types of items are provided in the folder "3_elements".
In the same way as for the elements we have already seen, all these objects are already adapted in size and pre-positioned to fit perfectly.The handling is just as simple, you select the item in which you want to position a light, you drag and drop your prefab lighting as a child and it will be positioned automatically.

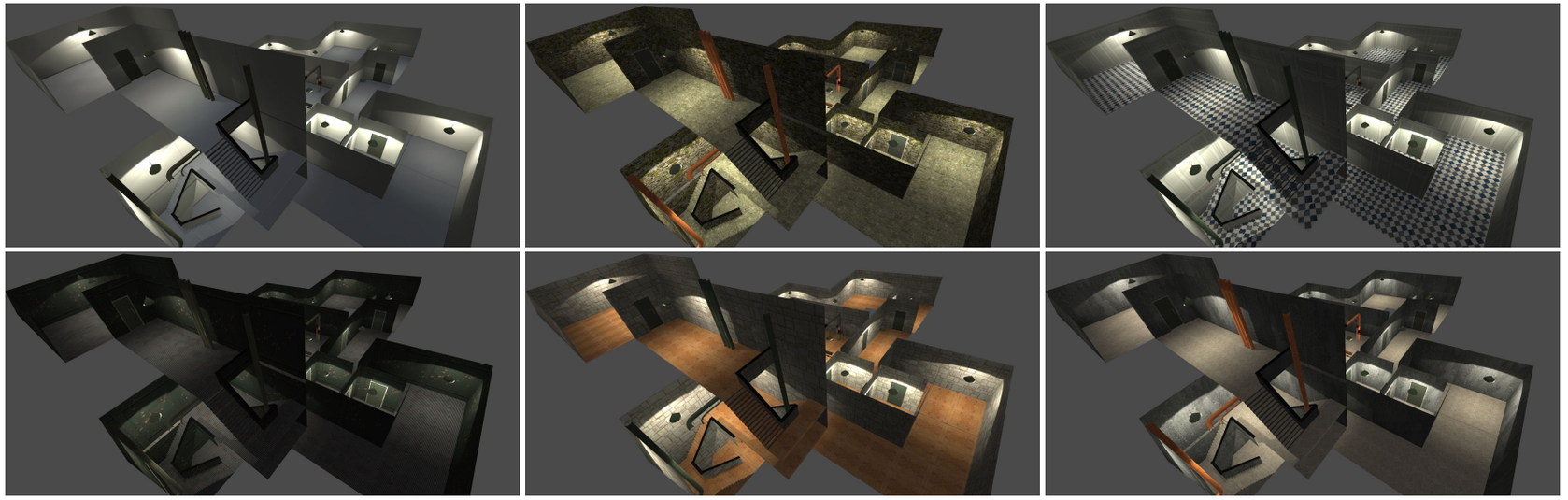
Some types of items require some clarification:

_ Doors: For the door elements to be positioned automatically in the correct position; they best be integrated in a "Room" element.
The doors have physical parameters that come into relationship with all animated objects with a "Push component" script (the First Person Controller included in the package has a "Push component" script).
The door "E04_sliding-door" is the only one that slides, it has the script "active_anim_component" which as its name implies activate animation of the door when the character enters the detection area, and plays the animation backwards when it comes out. This script is also used for the elevator doors.

_ Lighting: Important, certain lighting elements like the torches and the roof-candelabra (E03_light_ceiling) use **particles**. Although each particle generator is limited (15 particles simultaneously for the torches and 4 for each candle of the element E03_light_ceiling), if you use many of these elements, this could lead to decline in performances.

_ Windows: These are to be positioned in the openings created originally for doors.

_ Vaults: The Vault elements are added to your corridors and your room. If you decide to create your entire level with vaulted ceilings, we recommend creating your own customized prefabs in order to save time during the distribution.



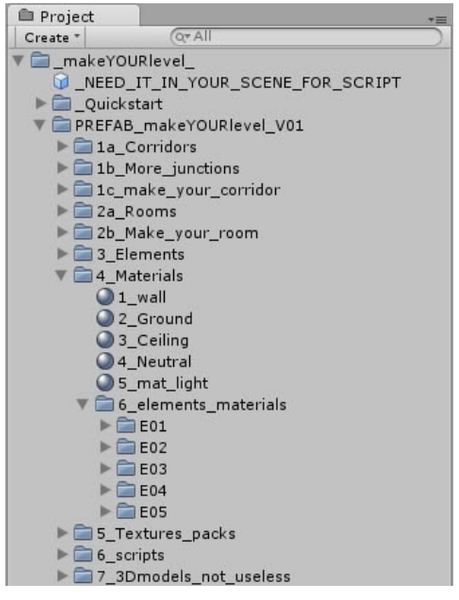
4. Change the materials on your overall level

All the 3D objects in ***makeYOURlevel*** were unfolded with the utmost care. We made sure that the UV unwrapping allows you to apply the tile-able texture of your choice on all objects.

This allowed us also to have only four main materials that are used on all the prefabs, This allows the user to change at any time the textures and effects of these materials on the whole scene.

To create the six screenshots above, it took approximately 2 or 3 minutes to change the entire environment of the level while of course maintaining the lightmap calculated directly in Unity.

Once you have the overall character of your level, you can apply different materials on certain elements to bring variations,



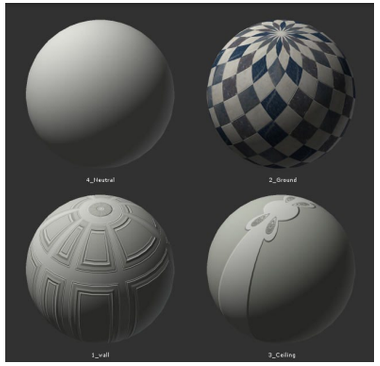
4.1 Method

Let us focus on the materials the most representative of your environment. These would be the walls, ceilings and the ground, a fourth material that we called "neutral" is used on door frames, staircases and other specific elements.
You will find these four materials in the first folder level "04_materials".

To change the textures applied to every wall of your scene, select the material "walls" and give it a new texture and the adapted normal map .. and that's it!

Do the same for the other materials which we enumerated.

We have included in the package 5 types of environments, but what is important is that you can use your own textures to create a game like no other!



Secondary elements such as doors or windows are unfolded so that you can easily change their appearance as well.

Important: the textures that we included in the package are for the vast majority of 2048x2048 pixels. We set the default aniso level to 4.Feel free to change these settings to optimize your game according to the type of media that your project is intended for.

All textures are classified by family in the "5_Textures_packs".

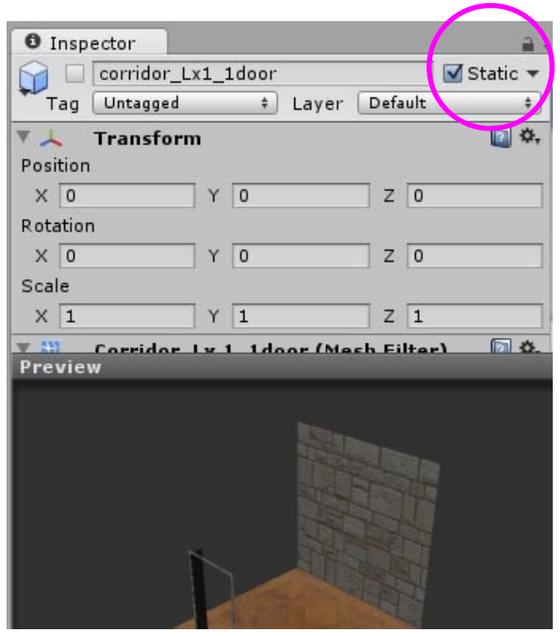
Important: It is entirely possible to choose a single element of your level and assign a unique material to break the loop effect.

5. Generating the lightmap

Here are some tips to generate your lightmap. These indications are to adapt according to your level of requirements, the size of your level of play and the hardware requirements.

Important: The calculation of the lightmap can be long and this time will be directly related to the size of your level, the number of light sources you have placed and their range but also of course to the capabilities of your machine and to the optimization of objects you have integrated in your scene and not from ***makeYOURlevel***.

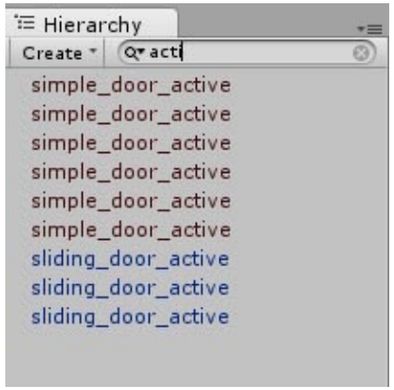
At this point, you have created your level, you have added secondary elements such as doors and light sources along with all the other elements of your scene.



Important: Most ***makeYOURlevel*** elements are intended to be static. They are preconfigured with "Static" checked in the inspector.

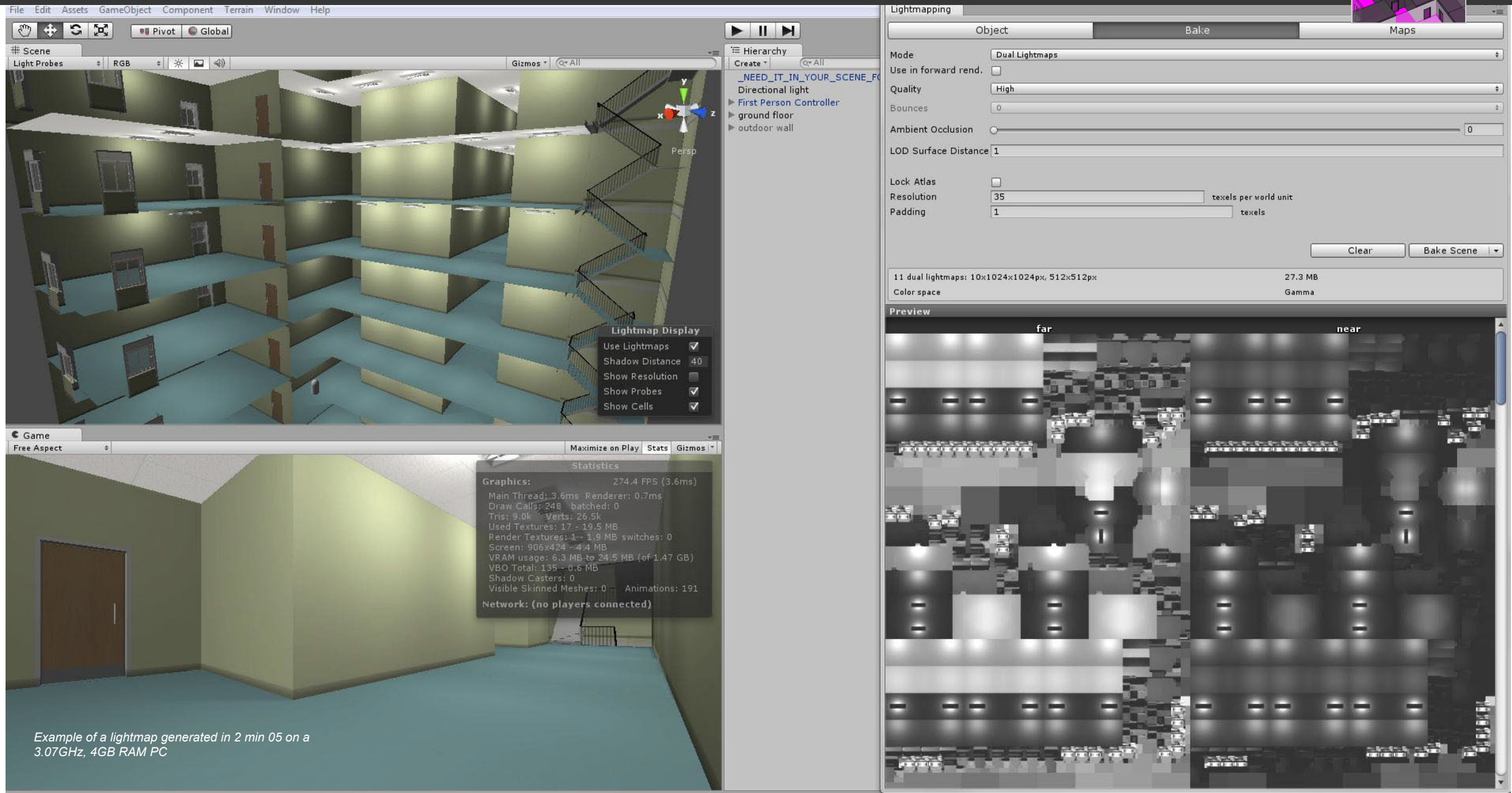
They will therefore be directly taken into account during the calculation of the lightmap.
Other elements such as doors have the possibility to interact so the "Static" attribute is left unchecked.
If we run the calculation of the lightmap in these conditions, the doors will not be taken into account in the calculation.

In our examples, we chose not to bring them dynamic lighting and to integrate them to the lightmap.
All objects intended to be dynamic in the package ***makeYOURlevel*** have the word "active" in their names.
This is done so that if you want to change their "Static" option, you simply need to write "active" in the search field of your scene hierarchy panel, select all of them and check the "Static" option in the Inspector panel.



Proceeding in this manner you can fine tune the lighting (the objects "sources_of_lighting" present in the package all use point light with default names) but we recommend that you change their settings directly in the prefab.

Once you have selected the "static" for all objects that should benefit from the lightmap, open the "lightmapping" window in Unity, make your adjustments and start the calculation. Once you have an outcome that suits you, remember to uncheck the "Static" options for objects that will come into motion.



Example of a lightmap generated in 2 min 05 on a 3.07GHz, 4GB RAM PC

by Pixtim

www.pixtim.com
pixtim@pixtim.com