



Gates

More details about the Levels Editor of Blade

Masklin, October 2002

English version by Prospero, November 2002

Contents

Introduction.....	3
Portals.....	3
Visualizing Textures In The Editor.....	3
One More One Equal To Three.	5
Texture Assignment.....	13
Default Values.....	17
Improving The Door.....	17
Taking Advantage Of Our Work.	22
Windows.	24
Sun and Moon.	28
Go To The Exterior.	29
Enlarging The Exterior.	31
Speaking Of Roofs.....	33
At The Top.....	35
Slope, Please.....	36
Vertical Textures.	39
Another Farewell.....	41
Credits.	41

Introduction.

In a previous instalment our objective was to overcome the initial fear of the LED. Without too many technical complications we could make our first level.

Too much time has passed since I wrote that first tutorial. The Blade community continues active in spite of the closing of Rebel Act Studios and many are those that venture in the design of maps. But we still don't have a good selection of tutorials about map-making in Blade and few are the brave souls that continue with LED after the first attempt.

Now we will try to advance a little more into the world of level designing with LED. Not too much, only a little more. But even this way, we will have a few problems. Guaranteed.

The initial idea is to create doors and windows in our room. We will see what there is outside and what happens when we work with several sectors. Finally we will play a little with the inclination of floors and roofs.

Portals.

The Blade engine is based on portals, a technique that at the moment has a certain prestige in the world of the 3D programming. A portal is, simply, the union of two sectors. As we saw in our previous tutorial, a sector is a hole that we dug to build our map. When uniting two of these holes, the common area becomes a portal, an area through which light can pass, that is to say, a transparent area.

Contrary to our desires, when uniting two rooms we don't obtain two rooms and a door between the two rooms, but a bigger room. We will see little by little through this tutorial; that is our main objective.

It is important to understand that every time that we unite two sectors, they have an area in common in the height that creates a portal through which the light passes. The engine of Blade, as do all the ones based on portals, use the point of view of the camera as the origin of the light and the engine will carry out the necessary calculations to draw the current sector. Next, it will calculate all the sectors that are contiguous to the first one by means of portals.

It will continue this way until there are no sectors to calculate.

The fewer sectors through which the light passes, the fewer calculations will be needed and quicker it is the visualization. It is a topic that it is outside of the scope of this tutorial, but it is interesting to know that it is desirable to limit the number of visible sectors and therefore Improve the performance of our maps.

Visualizing Textures In The Editor.

When I wrote the first tutorial, we didn't have the bmps associated to the textures of RAS.

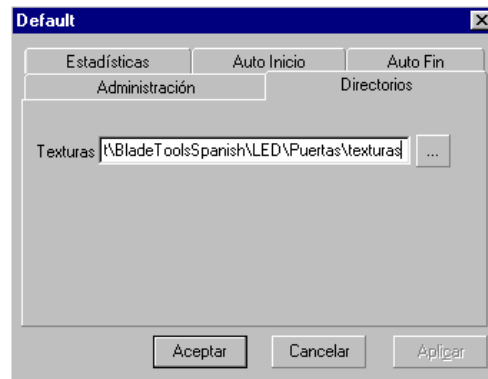
Thanks to Sergio Garces we now have an excellent tool called [mmp_dump](#) to extract all the bmps of an mmp file.

In our working directory make a sub-directory called textures. We copy the executable mmp_dump.exe in this folder and, from a MS-DOS window we cd to this directory. We execute the command:

```
mmp_dump ..\ragnar.mmp
```

and we will see the tool extracts all the bmps associated to the textures contained in the mmp. The names of the bmps are the internal names of the textures.

Now we will tell the LED where to look for the bmps. We choose it the option *File->Properties* of the menu and we enable the Directory tab:



We enter in the *Texture* field the complete path of the directory to which we have extracted the textures of ragnar.mmp

We press *Aceptar* (OK) and we exit the LED after saving the map. Next time that we open our map we will be able to see all the textures by pressing the *palette* button



Also in the property sheet we will see the textures. This will make it easier to select the correct one




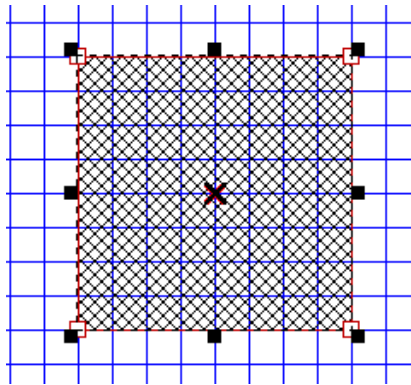
One Plus One is Equal to Three

We begin our map by opening `tonta.mp` with the LED.

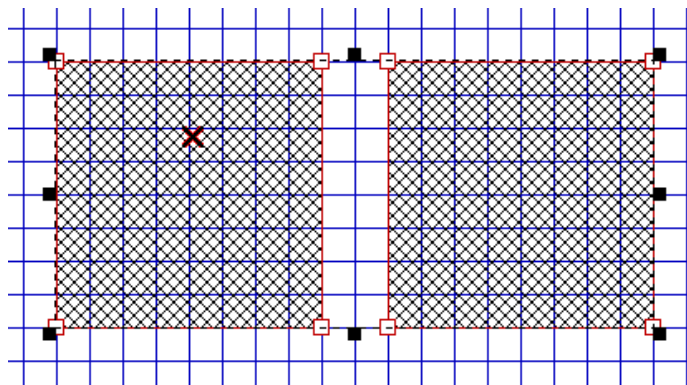
If you don't have it to hand you can grab it from [here](#), although only the `.mp`, you will have to add the level file (lvl) the textures `ragnar.mmp` and `casa_d.mmp`.

Next to the first sector, leaving some separation, we will create a new sector. We have two options, to create it again or to copy the one that we already have. We choose this second option. This way we learn how to copy sectors.

We change the grid so that it has a scale of 1 and we use the *Select Sectors* tool . Then, we click on our sector:

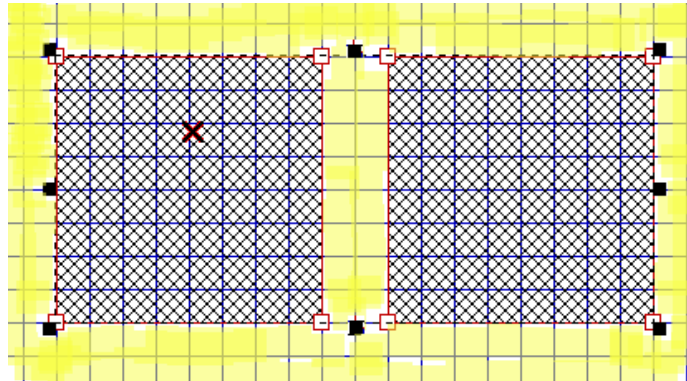



Now we press and hold the Shift key and clicking on the selected sector, drag it to the right, keeping the Shift key down. We will see that the original sector remains while a second sector, identical to the first, moves toward the right. When we release the mouse button we will obtain two identical sectors separated by a space:



The LED doesn't have a simple option of undoing. So, if we make an error the best option is probably to exit the LED and to recover the previous version of the map.

We have left a separation between the two sectors of two units, that is to say, about two meters in game units. What we are seeing now is two identical holes separated by solid land. If we colour the solid area we would have:



Now we save the map and compile. In the 3D view we won't notice any variation with the map. Although the second room exists, it is not visible from the first one. To know easily where this second room is, let us change the texture of the nearest wall. By using the *Select Sectors* tool  we select the right side of the original room. We press V and we change the texture for *grec1*. We set a zoom of 20 and an angle of 90:

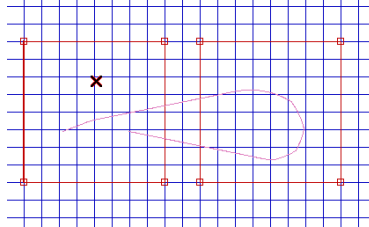


To see the changes we should save and compile. If we have closed the 3D viewer it won't be possible to open it again; you must leave the LED and enter again. If we still keep open the 3D viewer you need to update it by loading the new compiled file (*tonta.bw*). We activate the 3D window and, r-clicking, we select Actualize bw:



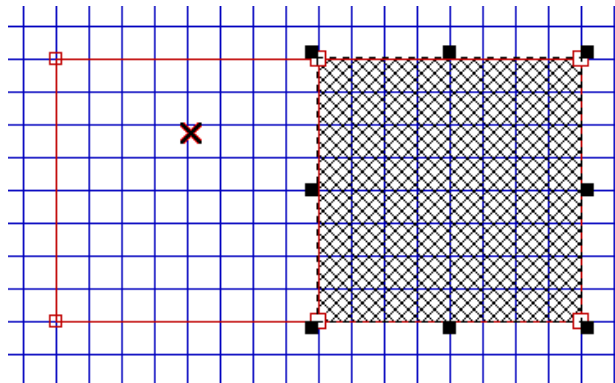
If everything has gone well we will be able to distinguish the wall that belongs to the right side of our first room clearly. With the cursors we place ourselves perpendicular to the wall and we try to cross it. Things will go dark until we pass through the solid space, but soon we emerge in our second room.


It is easy to be disoriented in the 3D view. If we enter in a solid area, it is possible that we will become stuck. I hope that you have not got lost with two rooms, because things will be more complicated when your map has hundred of them, but to help you to navigate, the LED includes an interesting feature. If you place the Editor window and the 3D viewer so both can be visualized simultaneously, then, when moving in the 3D viewer you will see a line in the Editor window that marks your trajectory. Very useful!

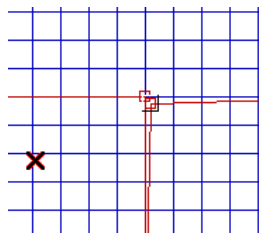


I suppose at this point you will want to join the two rooms to see what happens. Don't expect anything spectacular. The wall that separates the two rooms will simply disappear.

Let us return to the Editor and select the sector on the right. Without holding down the Shift key, drag it with the mouse towards the left until the two sectors are united.




If we have activated the grid  it will be relatively easy to unite both sectors. If we do not have it activated then there is a possibility that the vertexes will not superimpose perfectly.

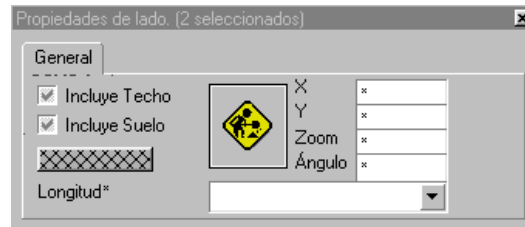


If the vertexes do not superimpose perfectly it could lead to problems in the future. The Portal created will be defective and exhibit strange effects as the light crosses it. Even textures could appear where there should be transparency.


When uniting both sectors the superimposed sides become a portal. There is nothing in the Editor that tells us that this is a portal, we will simply know it and we can check it in the 3D viewer. After saving, compile and reload the map.

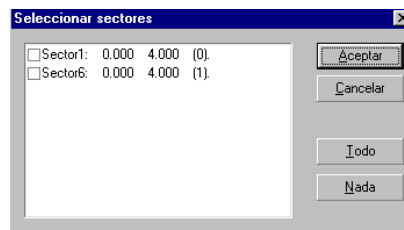
We now have a rectangular room of double size, made by the union of two sectors. The central wall, the portal, is not visible and we can cross it as if it didn't exist.

We can use the *Select walls* tool  to select the portal and visualize the properties palette. We will see that there are no settings displayed:



This is because we are selecting two sides that constitute the portal simultaneously. If we modify an attribute we will change it on both sides.

To select only one of the sides we should double click on them with the *Select walls* tool  enabled. So, the *Select Sectors* window is shown:



In this window we can choose the required side that we want to modify. When selecting a side we will see that the corresponding sector changes colour slightly. If we choose the side associated to the left sector, our original room, we will see that it retains the assigned texture *grec1*.

If we choose the other side, the texture will be the same as the rest of walls, that is to say, *bronce* (brass). And neither of the two textures appears on the wall since it has become a portal. In principle, if the two rooms have the same height the texture assigned to the sides that constitute a portal it is irrelevant.

But if we vary the height... what happens? Does the portal continue to exist? The answer is Yes, but only in the common area between the two sectors. Let us consider a side view of our map now, something that, unfortunately, the LED doesn't offer us.

Again we have coloured in yellow the solid area. The left sector, the original, we have maintained it with a height of 4. The right has gone down to 3. Now the portal is limited to the area marked in green.

To reproduce this situation with the LED, we select the right sector and in the properties window we choose a height of 3 for the ceiling.

When reloading our map we can observe how only the common area is transparent. Now, only this area is our portal.

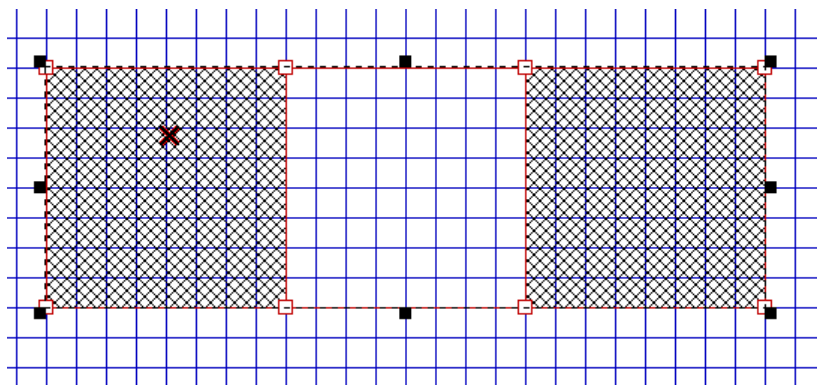


We can also see that the rest of the wall, that is, the part that is not the portal, shows the texture corresponding to the side associated to the original sector: *grec1*.

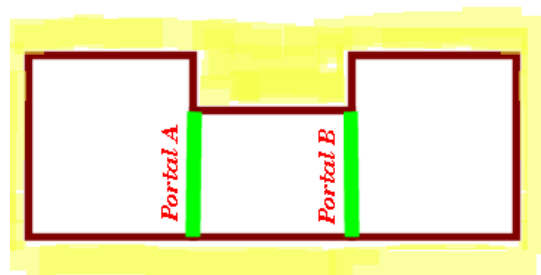
This behaviour is logical since in that area, in that sector, we have assigned this texture. The texture becomes transparent in the area common of both sectors but it shows in the remaining area of the wall.

Now we copy the original sector to the other end and we will have a first step to our two rooms with door.

We select the original sector and, pressing Shift, we drag it toward the right with the grid enabled. So they are in the following arrangement:



With this operation we have three sectors, the central of 3 units of height and two either side of this with heights of 4 units. Vertically the result would be:



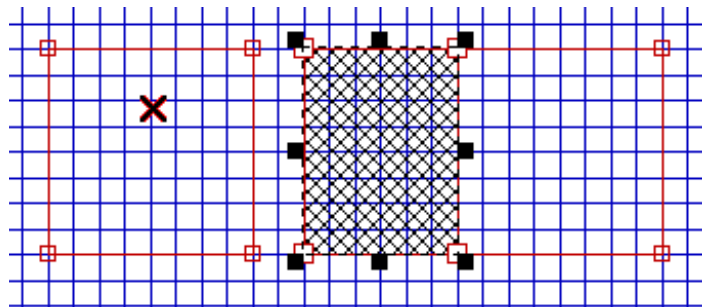
We see clearly the two portals and the effect that has been created in the 3D view. You will see a corridor between the two rooms. As we have copied the first sector, the texture *grec1* appears at the bottom but... what texture is it applied to the upper part of the Portal B? Good, I hope that you have answered *bronze* (brass), that is to say, the same one that is on the left wall of the first sector. If it is not this way, don't worry. We will change it so that you understand better which is the texture that is covering that area.


We double click in the Portal B that separates the outer sectors so that the *Select sectors* window is shown. We choose the side that belongs to the new sector and we change the texture for *cala*. It is a texture with decorations that we will locate perfectly once our map is compiled. Is it exactly where you hoped... honestly?

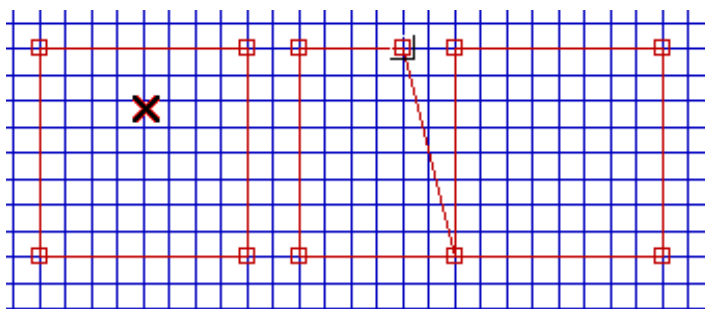
With this operation we have come closer to our objective of uniting two rooms by a door. At the moment it is not a door, but sort of a corridor, but I hope that you have understood that to build our door we need a third sector because the door is also a hole, a sector. If we unite the sectors directly then, the portal that is formed will merge into the wall. For that reason it is necessary to interpose this third sector.

Of course, so that it appears like a door we need it to be narrower and this is our next objective.

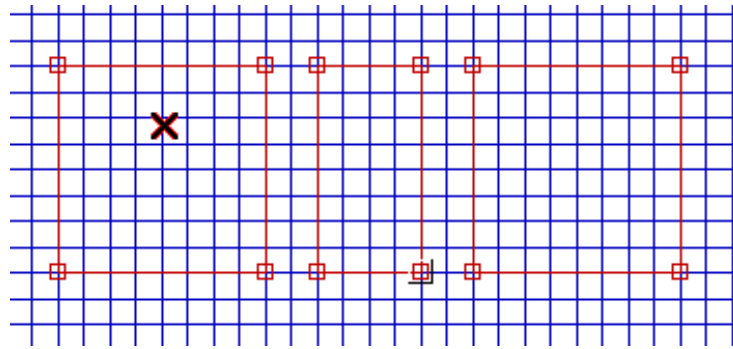
Before doing this we need to learn some additional technical stuff. We begin by trying to separate the central sector from the other two, that is to say, unlinking the portals. There are several methods of doing it. One is to select the central sector and to take the cursor toward the centre of Portal A. When the cursor becomes a horizontal double arrow \leftrightarrow we drag the side with the mouse toward the right. We will see that the side separates from the portal.



We repeat the operation with the Portal B but now using a similar technique but with vertices. We double click on the upper vertex of the Portal B with the *Select vertex* tool  enabled. We choose the vertex belonging to the central sector (remember that the associate sector changes colour) and we drag it toward the left.

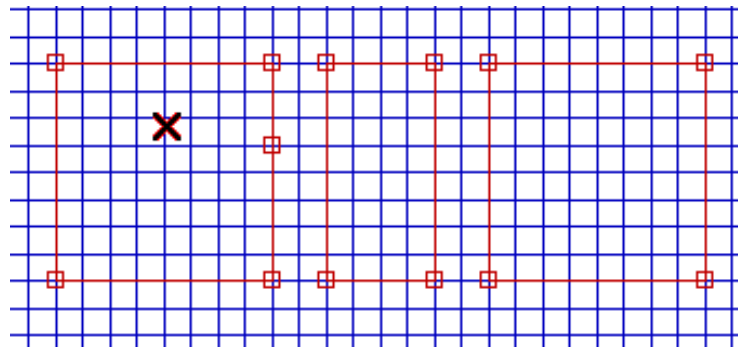


We repeat the operation with the lower vertex and we have two isolated portals.

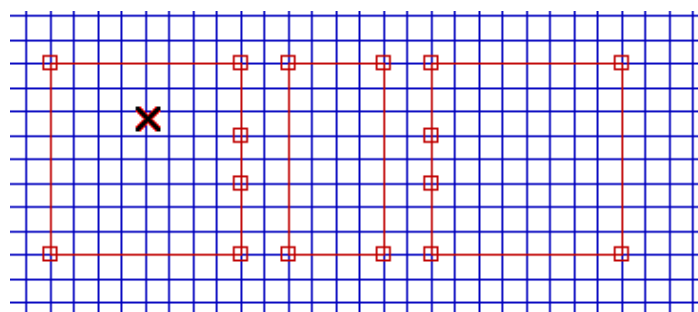


Evidently this second procedure is something slower, but we wanted to show how to separate a vertex of a portal.


The following step is to add vertexes to the side sectors. These vertexes will indicate to the engine of Blade where the sector begins. We locate the cursor in the right side of the first sector, approximately at 2/3 of height, and we r-click. We choose to *Insert vertex* and we will see a new vertex appears.

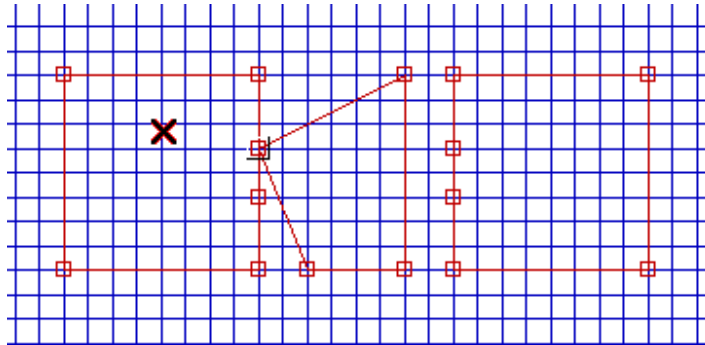


We repeat the operation below and on the left side of the third sector.

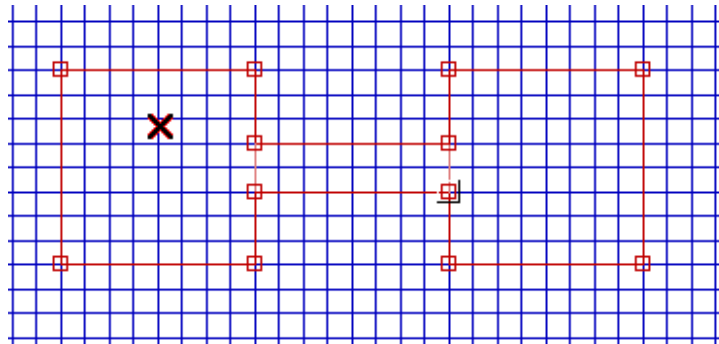


These vertexes will be the points where the new portals will begin.

For that we should choose the *Select vertex* tool  and drag the four vertexes of the central sector toward the four new vertexes that we have just created.



This way we obtain three sectors united by two new portals, but now, they no longer occupy all the width of the wall.



It is time of to compile and to see what it has happened in our map. The result is something very similar to what we are aiming at, although the textures seem to have gone astray.



Conceptually we have arrived at our two rooms united by a door. Maybe it is more of a corridor than a door and the textures need some adjustment, but the first step is taken: we have escaped from our room.

Texture Assignment.

In view of the results that we have obtained it is necessary to know more about the assignment of textures. The basic design of a map is built with a small amount of ability with the LED and a lot of art assigning textures. Not all are endowed with that ability but we can compensate with the magnificent textures that accompany the maps of Blade and a good dose of patience.

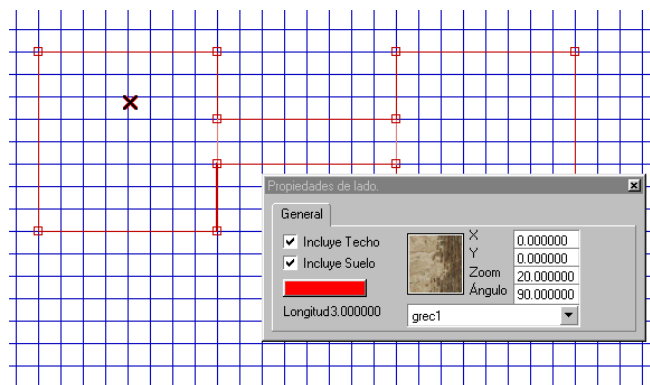
Your map can show a slightly different aspect with different textures to those that are presented here. You can have configured a different default texture. Later on we will explain about it.

For the time being we centre ourselves in getting a healthier aspect for our door. As you already know from our previous tutorial, you can modify the textures with the 3D viewer, but before doing anything we need to know what it has happened.

Whenever one adds a vertex to a sector the original side it is divided in two sides and each of these can be assigned different textures. One of the sides preserves the texture of the original while the other one is assigned the default texture.

Something of this was already seen in the first tutorial, but it is important to understand it because it will be the basis of getting good combinations of textures. Sometimes, we will add vertexes to a sector not to modify their shape, but to be able to assign different textures.

To check it we center ourselves in the portal A, located in our initial sector. We select in the Editor the side located exactly under this portal and in the properties window we assign the texture *grec1*, a zoom of 20, an angle of 90 and X=Y=0.



Most likely, we have not had to change anything because it was already this way. That side inherited the original configuration of the complete side of our First Sector before breaking it by adding vertexes.

Let us make again with the upper side. Surely it is here that we have to carry out changes. Lastly we select the central side. Because it is a portal when double clicking it will show us the Select sectors window. We choose the sector of the left. In the properties window and make sure that the textures and the parameters are the same ones that those of the rest of sides.

We save and we compile and now we will have a better result.



Indeed it does resemble a door, but by playing with the textures we can improve the aspect. We will attempt it from the 3D viewer. As we know, we can use the Select surfaces option (r-click) and the keys of the keypad to assign textures, to move them, to rotate them,... But when selecting any one of the sides that constitute the East wall of our first room we will get a surprise: the three sides are selected as if was an only one side. If we have the properties window visible we can see that indeed there are 3 sides selected.

This behaviour is useful in many situations. If two or more contiguous sides share texture they will be selected together. In other cases, it interests us to assign textures different to each segment.

Then we will have to use the Editor, where we will be able to select them for separate. We can prove some assignments of textures. For example, the texture *ad1* with a zoom of 20 and an angle of 90 gives a better effect for our door.



In this case we have selected a complex texture that is well suited to our needs. With more practice we will be able to use two or more textures that, together, provide the desired effect.

When In the 3D view we select several contiguous sides together the LED has tendency to remember it the close of the session. Although we change the textures in the editor and we assign a different one to each segment we will continue seeing a single texture in the three sides. To solve this it is only necessary to make the changes, to save the map and to reload it in the view 3D.

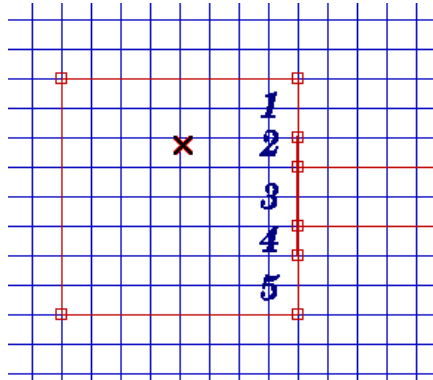
For example, combining the textures *flw* in the lateral segments and the texture *1* (an odd name ☺) in the centre, with zoom of 20, angles of 90 and some adjustment in the central texture can give the following result.



Something worse. This door is even more insipid than the previous one, but it shows us how to assign different textures in an individual wall. Now we can take advantage of this to improve the result.

Indeed, the central texture is, really an arch. What has happened is that it is that only a part of it shows. That is due to the two vertexes that were included to create the door, not to improve it aesthetically.

Now, we add other two vertexes to our wall:



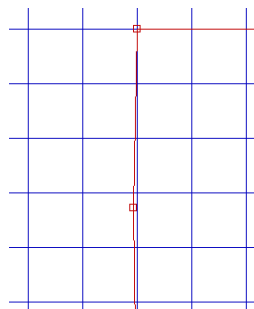
To the external segments, 1 and 5, we assign him the texture *flw* with $x=y=0$, zoom of 20 and angle of 90. To the internal segments, 2-3-4, we assign the texture *1* with $x=-95$, $y=-107$, zoom of 13 and angle of 90.

We save, we compile and we reload the map. The result will be similar to the one shown here:



It looks like a better door, although we have the problem that the pattern on the texture is curved and our door is rectangular. We will solve it at once, but we still need to attend to some details before doing this.

When adding new vertexes it is important to have enabled the grid. If it is not this way, it is quite probable that the vertex divides the side in two segments that are not perfectly aligned. Only with a high level of zoom this defect is perceived:

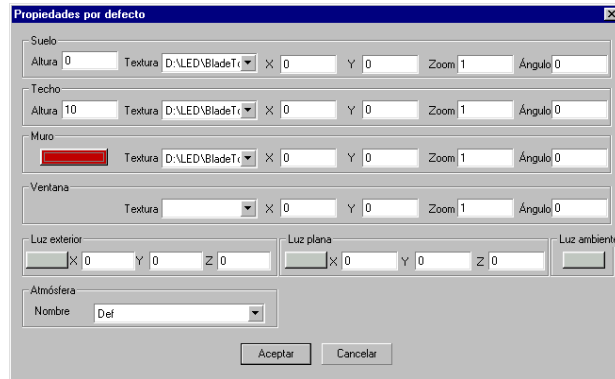


I assure you that it will be source of multiple problems. As we saw shortly in the previous tutorial, all the sectors should be convex and with this defect it can happen that one of the sectors is not (that of the left in the previous drawing is not convex). The assignment of textures and the later behaviour of the map will be a real nightmare. With the grid enabled it won't happen.

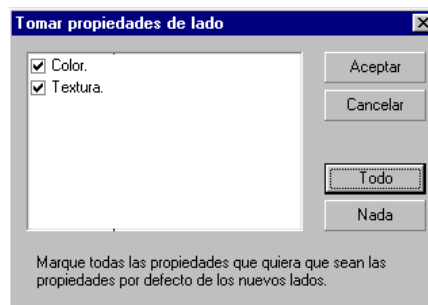
Another important detail to consider is that the zoom level that we have chosen for the grid limits the selection possibilities. If, for example, we have chosen a zoom of 1.0 then to select a side that measures 1.0 exactly or fewer can be a very difficult task. The solution is in diminishing the grid scale, for example to 0.5, to select the side and leave the zoom in 1.0 again. Another possibility is to disable the grid, which will make selection possible. As we have seen before, it is important not to forget activate it again before continuing.

Default Values.

The default values of many properties can be configured by using the option *Tools -> Default Properties*. It will only be possible to access to this option from the Editor, not from the 3D viewer. A window will appear and we will be able to change many of the default attributes.



But usually we won't follow this procedure to assign default values. Instead, for example, to assign the default values of the walls we can select in the Editor a side that seems us appropriate and, r-clicking, we choose to *Put default*. We are then shown a selection window.



In this window, we press *All* so that all the marked properties are copied, and then press the *OK* button. Now the sides that we will make, for example when adding vertexes, will be assigned the texture of this wall. Likewise, we can do this with the sectors, although the properties to copy are many. This way, when we are working in an area with some certain requirements of height, textures, zoom... we change the default properties so the new sectors are all equal. Later on, in the 3D viewer, we can refine the assignment of textures.

Improving The Door.

Our door has a strange aspect because it is rectangular and the assigned texture is an arch. In Blade it is possible to curve the floors and roofs of a sector to get the effect of simulating the aspect of mountains or getting arches, doors or windows in different ways.

Unfortunately the LED does not take care this aspect too well and it is easy to encounter difficulties. We will try to resolve some of them to be able to get as good a result as soon as possible.

In the first place we select the sector that constitutes our door, (really a corridor). We will curve the roof. We show the properties windows and we activate the *Ceiling* tab.



We see that it admits three different shapes: Spherical, Elliptic and Flat. Selecting any one of these shapes we will be able to obtain flat, spherical and elliptic roofs.

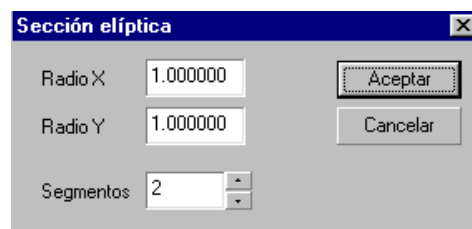


The flat roofs will be the most familiar and they don't need any additional parameter, only the height that we already know. The Spherical roofs don't work well in LED. It seems that the LED have a bug and they don't work correctly. Nevertheless, it is easy to understand that one can simply obtain a spherical roof by making the values of height and width equals.

Spherical roofs/floors work better in caves or any irregular natural features. Any segments that don't work can serve to add to the effect.....Pro.

The Elliptic roofs are the most interesting. The first step is to indicate to the Editor that we want a elliptic roof. You must select the option in the properties window and to press Enter. It is very important this step, as always. If we don't press Enter, the editor is not informed that we are making an elliptic roof.

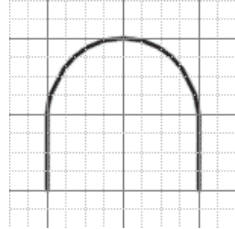
Next we press the small grey button located next to the Elliptic option and a new window appears.



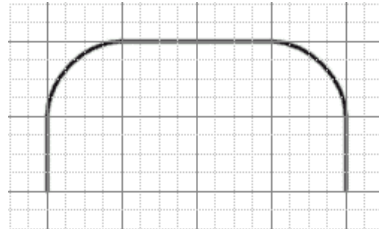
In this window we choose the X and Y radius of the arch. We also choose the number of segments that the engine will use to create our arch. The higher this number, the smoother the curve will be, but the performance will be affected. You can choose 4 or 6 segments.

The meaning of the X and Y radius is better demonstrated with some examples. We have borrowed some of these examples from the excellent tutorial that *SlayerDDD* designed.

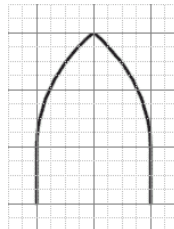
If we choose $X=1$ and $Y=1$ and the width of our sector is 2 units, then the result will be:



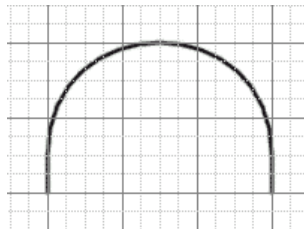
If the width of the sector is 4 units, the same election $X=1$ and $Y=1$ lead to the following arch:



If we make X different from Y, then we will have true elliptic arches. This way, for a sector of 2 width units an election of $X=1$ and $Y=2$ results in the well-known Gothic arch:

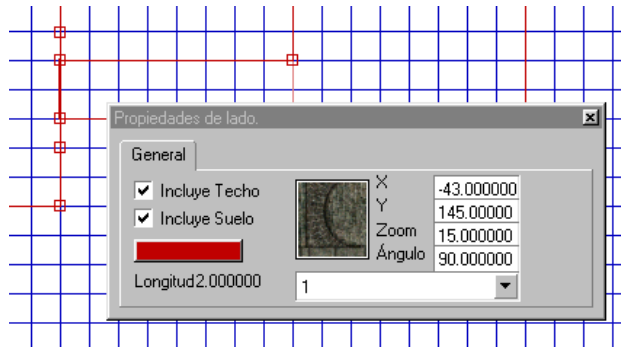


Is more intuitive to use whole units, but it is possible any combination. For example, to get a spherical arch in a sector of 3 width units we should choose $X=Y=1.5$.



It is important to remember that after changing any one of these parameters it is always necessary to press Enter. You already know that the LED always requires a good jog to this key to wake it up.

In our map we have a corridor whose roof has now been curved. The width of this corridor is 2 units. We can discover this information by setting the grid to 1.0 and checking that the side measures two squares. We can also select one of the lateral walls and check their length in the properties window.



Therefore, to get a totally spherical roof we choose $X=1$ and $Y=1$, as well as 4 for the number of segments. After we save and compile, the result should be similar to the following one.

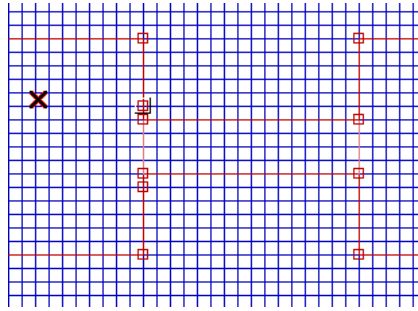


It is not bad, although you can still improve it. In the first place it is possible that the textures have moved. In that case, the easiest solution is to enable the properties window and from the 3D viewer set the parameters of the three segments that compose the arch again. In order that the three segments have the same properties and they are controlled jointly we should select them with the Control key pressed. This way several surfaces can be selected simultaneously. Any change that we make in the properties window will affect to all the selected surfaces. The properties that we maintain without changing will remain unaffected.


In our case it interests us that the three segments have the same attributes, so we set $X=Y=0$, zoom of 15, angle of 90 and the texture at 1. When pressing Enter the three surfaces will become a continuous surface that we will be able to manage as a whole. Now we use the keys of the keypad to choose an appropriate zoom and a position for the texture that is suitable for the arch that we have created. Some appropriate values can be $X = -124$, $Y = -104$, zoom of 11 and angle of 90.

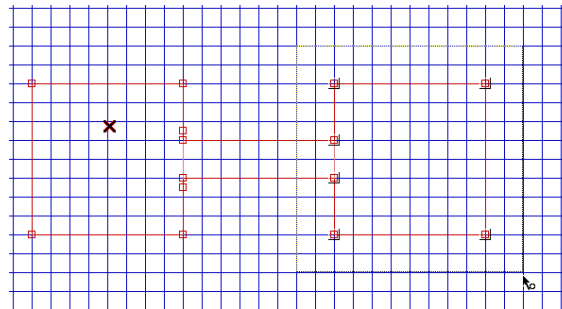
Now, the only significant defect of our door is that the transition between the two textures is not natural. We can take advantage of the columns in the texture of the arch so that they act as a separation between the two textures. We should move the vertexes separating the textures of our door closer.

For example, we can set the zoom of the grid to 0.5 and to bring the two vertexes 0.5 points nearer to the door:

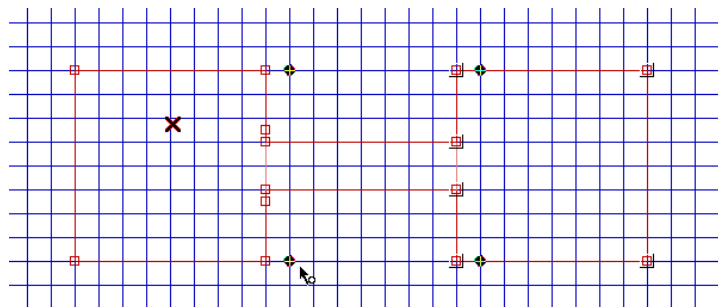


We can also change the texture of the roof of our door for the *bronze* texture, zoom of 20 and angle of 90 so that it looks better. Lastly we can make the door shorter, that is to say, to transform our corridor into a real door.

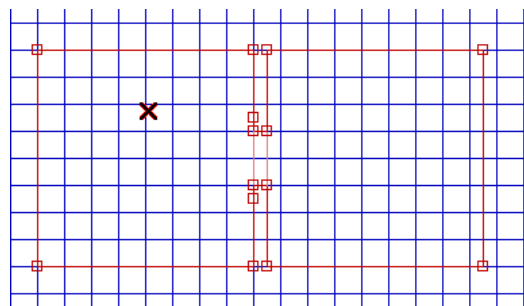
To achieve this we should use the Select vertex tool  and drag with the mouse to select all the vertexes to the right of the corridor:



We enable the grid to 0.5 and we click on one of the selected vertexes, dragging it toward the left. We will see that all the vertexes move in unison.



As a result, the corridor will have been reduced to, let us say, 0.5 units:



Our door will have a final aspect. After saving, compiling and reloading..... quite promising!



Now, the other side of the door is not finished, but that we leave as an exercise, in order that we do not lose forget the lessons of our old professors.

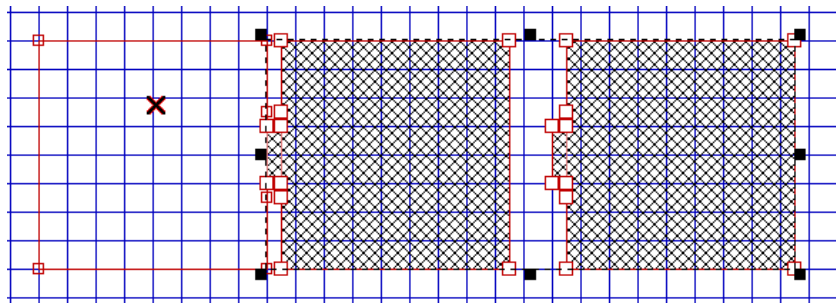
Taking Advantage Of Our Work.

Although with the experience the previous process we will gain a certain fluency, it is important to understand that we can take advantage of part of the realized work to enlarge our designs without necessity of mapping all the rooms one by one.

If the group of two rooms has been interesting, we can simply add a third room by copying sectors.

The process begins by selecting the sector to the right together with the one that constitutes the door. To be able to select both sectors you can to drag the mouse but, probably, it will be more comfortable to press Control and to click on each one of the sectors.

Once selected and keeping pressed the Shift key, drag them toward the right.

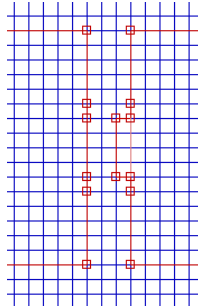


It is important to understand that it is not enough just to bring the new sector to the old one so that the portal is made. To do this properly it is necessary to create the vertexes previously. It is a common error to unite sectors and to think that the vertexes are made automatically.

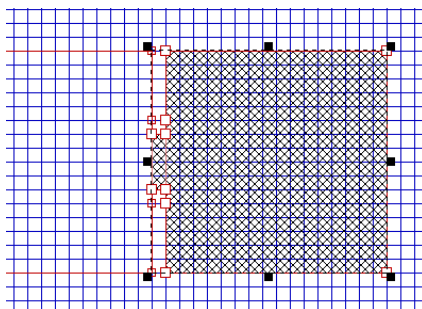
The compiler of the LED is able to add vertexes in an automatic way. Possibly in a complete map of several hundred of sectors the compiler will add some of these vertexes. But it is a practice to avoid. It can create strange situations that it can even affect the behaviour of the enemies.

In the first example there didn't exist vertexes, therefore they were created simultaneously in the two sides that constituted the portal when we insert them. Now, one of those sides already has the vertexes so it is better to add them to the remaining side in advance.

Therefore we choose a grid of 0.5 and we add four vertexes to the right wall of our second room in a similar way to as we have made in other occasions.



These new vertexes will be the anchorage points for the new door and the two vertexes that allow us to assign the textures of arch as we have seen in the previous section. Now we select the two sectors to the right again (the new room and the door) and we drag them until they unite exactly over the new vertexes:




We have finished our new room. Now we should make sure that the textures that are applied in the sides created by the new vertexes are the appropriate ones. Only that, because the rest of textures have been copied.

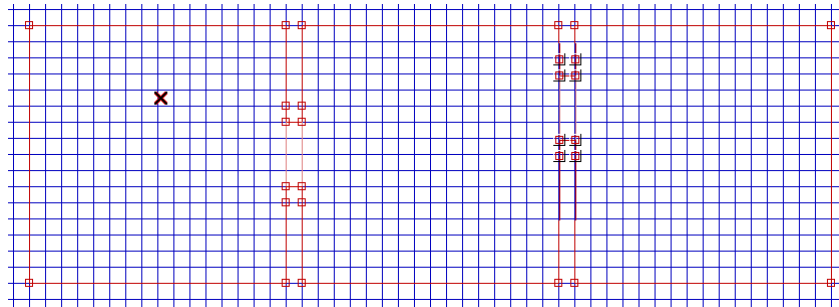
If everything has gone well, when we recompile map will have three rooms joined together:



Windows.


This seems the next logical step. But to create a window doesn't represent a bigger difficulty than that of creating a door. We should only keep in mind that the height of the floor will be higher than that of the floor of the rooms. This way, the hole will be off the ground.

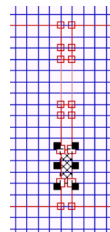
First, let us free a space in our second room to create the window. By enabling the *Select vertex* tool  we drag with the mouse to select the vertexes that constitute the second door. Then we click on anyone of them and, with a grid of 0.5, we drag them up to displace the door a little.



With this process the second door is no longer aligned with the first one and it leaves us space to create our window. If you are curious as to what has happened, check with the 3D viewer, although it is not so difficult to move the door, recompile and check it..

What? The textures have moved! I fear that it is a consequence of our displacement. But you already know how to select surfaces and to move textures in the 3D viewer. You won't find it difficult to correct. If you press and hold the Control key while using the keypad with the 3D viewer the movements will be quicker.

Now we go to for the window. We will create a new sector in the solid area that separates the central room of the room to the right. We simply choose a grid of 0.5 and we insert  the vertexes that we need. Then make the sector by joining the same vertexes. The aspect of our wall will be similar to this:

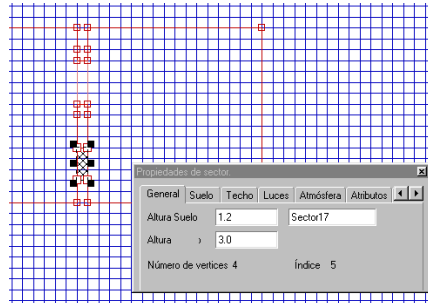


The new sector has been created on the vertexes that existed below. In fact, if we double click on one of these vertexes we will be shown the Select sectors window that indicates that each vertex belongs to two sides. This, as we know, means that a new portal has been created, a portal that soon will be our window.

If we didn't create the vertexes previously but rather we made the sector directly, the portal it will still be created because the compiler takes charge of adding the vertexes that you have missed. But we won't get tired of repeating that this point, as it will cause us problems

if ignored. For example, when later on we attempt the neo-gothic window the result will be unexpected.

What height does this sector have? It is the one that we have as height in the default properties for the sectors. Since our rooms go from 0 to 4, a window could go from 1.2 to 3, for example. We select our new sector and in the properties window we change both heights.

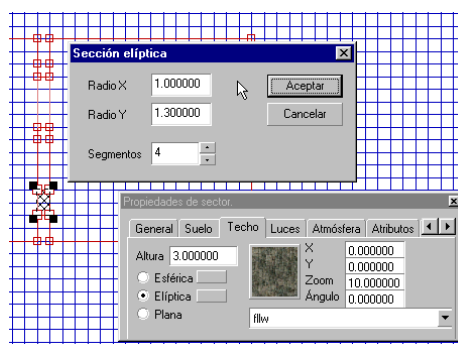


Now we save, we compile and we see the result.



It is not bad hey?. Good. Perhaps we need some adjustment of textures. We have also changed a little the textures in the window borders. On the sides we can use flw with a zoom of 10 and an angle of 90 and an angle of 0 on the top and bottom. This seems more appropriate for a window.

We can convert our window to the Gothic style. Drag two of the vertexes to make it narrower. Let us say that now it will now measure 1 unit. We select the sector and in the Ceiling tab we choose Elliptic (Enter) and then we assign X=1, Y=1.3 and Segments=4.

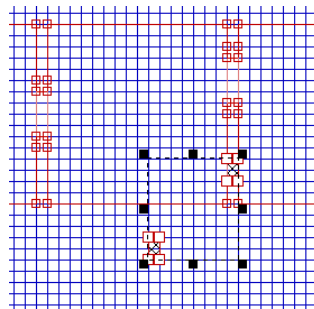


This is the result:



As we see, little by little we are getting results. Again we will take advantage of the existing work. We will copy the window to another wall, for example in the lower side. Of course, it will be necessary to rotate it....

First, we select our window and, with the Shift key pressed, we drag it toward the lower area of the map. We will have a copy of the window.

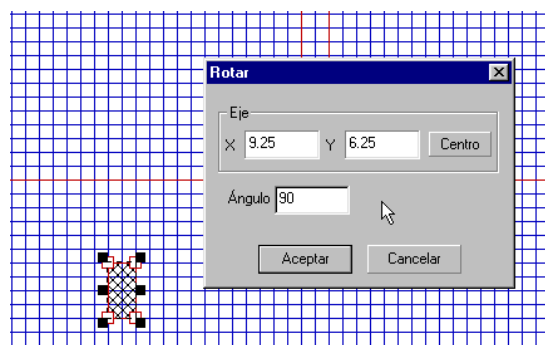


We select the new window so that only this sector is selected and, with the pointer of the mouse inside the sector, we r-click.

A context menu should appear that includes, among other options, *Moving*, *Rotate*, *Scale*,...

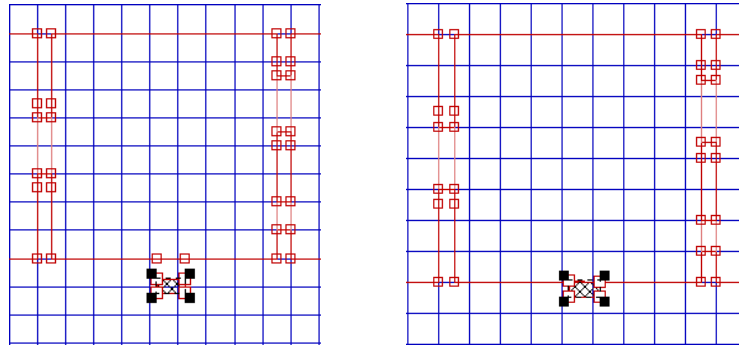
It is difficult to deploy this menu in small sectors. Zooming in or diminishing the grid size can help us.

If we choose the option to *Rotate* a new window appears and we can choose the axis of rotation and the angle. We only change the angle indicating 90 degrees:



This way we are able to rotate the window and now we only have to create the four vertexes in the lower wall and to drag the new window up. But... the lower wall doesn't exist! Indeed, it is not a wall as those that separate our rooms. Rather it is a solid block stretching to infinity, or at least sufficiently wide so that we get tired looking for the other end.

So we will only create two vertexes and to drag the window to them.

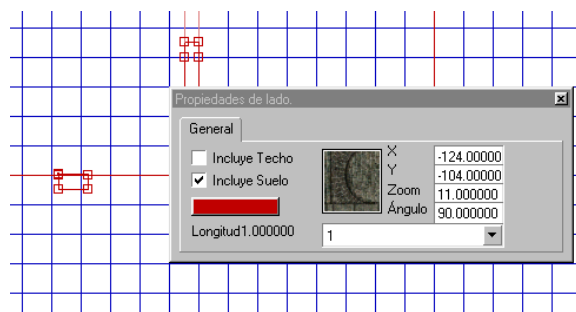


We compile and we see what has happened.

The result is difficult to visualize with an image. Perhaps you hoped for a hole in the wall of a similar shape to that of the original window. That is basically what has happened, but when a portal is not created, the engine of Blade tries to apply the effect of an elliptic roof to this side.

To avoid it we select the side of the new sector that is part of the portal, that is, the upper side of our new window.

In the properties we untick the option *Include Ceiling* and we press Enter.



There, we have fixed the strange curve that formed that side. But the side at the bottom, that is to say the lower side of the window in the map, also has the same problem. You must carry out this same operation for that side to have our complete window.

But, since on the other side we have still not made a sector, we are not able to form the portal so that this window is transparent, like the other window. But we can play with the textures and assign a suitable one for the window area. We select the lower side of our window again and we change the texture for *zad3*, zoom of 10, angle of 0.

Our window now looks like this:



Sun and Moon.

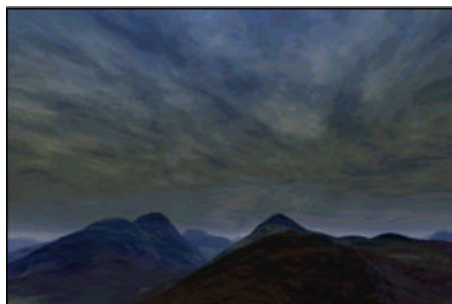
In playing about with windows maybe you want to see the exterior. It is not difficult in Blade to do it, but it is sometimes a little complex to understand.

All the maps load a file with special textures that constitute the celestial vault. As we saw in our first tutorial a line of the type:


`WorldDome -> casa_d.mmp`

The `lvl` file tells the engine of Blade where to find the textures that it will use to create the sky. Inside that file six textures exist (*DomeUp*, *DomeFront*, *DomeLeft*,...) for the six faces of a cube that it will surround our map. Indeed, the sky in Blade is generated from these six textures that undergo a spherical projection to create the impression that we are under a vault.

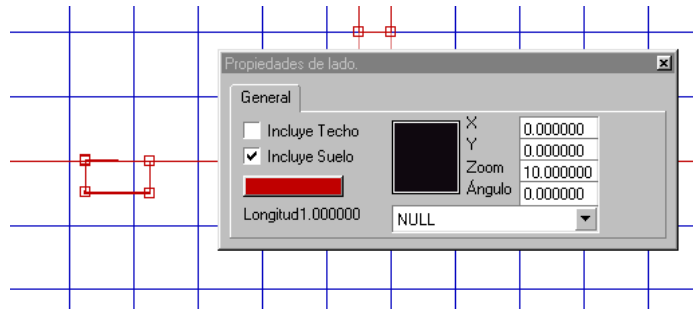
The textures are landscapes with little definition because it is supposed that the player won't be able to get close to them and he will only see them from afar. For example, in the map of characters' selection, the texture *DomeFront* is:



These textures don't appear as such in the LED but rather a special technique is used to assign them: the texture NULL. All the maps should register a NULL texture and it is unimportant which bmp is associated to it. When the engine of Blade is a surface that has assigned to it the NULL texture it will show an appropriate projection of the vault.

From the palette of textures  we can see all the available textures. It is very important that the texture is called NULL and not Null or null or any other combination of uppercase and lowercase. If in our palette of textures it doesn't appear with the name NULL we can change it easily.

We will see it much simpler with examples. Let us return to our map and let us select the lower side of our window again.



Let us change the window texture (*zad3*) for the texture NULL. We compile and we see the result:

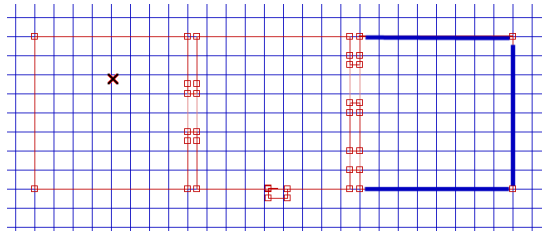


Great! A real window! Yes, very nice. We notice that as we move around the window, the landscape changes as it would in a real window. That is to say, the texture NULL is not a static bitmap but rather the engine of Blade takes charge of showing the appropriate perspective. For that reason, the zoom parameters, angle, etc, don't have effect on the texture NULL.

Although the result is correct, to assign a texture NULL directly to a window is not usual. Is better to create an entire open space that is seen from the window. We will make this soon. A direct assignment of this type can be interesting in windows or holes located far from the area of the player's action. From certain distance it is practically impossible to distinguish among a NULL texture and an open area.

Go To The Exterior.


We will transform one of our rooms into an outdoor area. We return to our map and we change the textures of the three sides marked by the texture NULL.

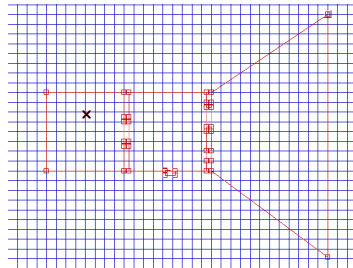


Now, we recompile the map and we will be able to reach the exterior. Good. The exterior is a little strange, it is not usual to have a roof on the landscape. Let us see what has happened.

The sector we changed had a height from 0 to 4 units. In the roof it has the texture *techo* (roof), then the next thing that we will have to do is to change it for NULL. This way the landscape will surround us on all sides.

As we have said before, it is desirable that the textures that show the vault, the NULL textures, be as far away as possible from the action area. We will raise the roof, let us say to 20 units. The texture of the floor we will change for another more appropriate one. In *ragnar.mmp* there is not a lot of choice so we will opt for *sig1* with a zoom of 20.

We will also enlarge our outdoor sector by enabling the Select vertex tool  and dragging the two vertexes on the extreme right until achieving something similar to this:



We have enlarged the area without adding new vertexes and preserving the convex sector.

Let us remember that all the sectors of Blade have to be convex. Otherwise we will have more problems that we could deal with. A sector is convex if, from anyone of its vertexes you can trace a line to all the other vertexes without being left the sector. You have been warned!

We see the result.



It continues being a little strange. Basically for two reasons. The first one is that the outdoor area doesn't surround our construction completely. The second is the height. The walls continue up uninterrupted.

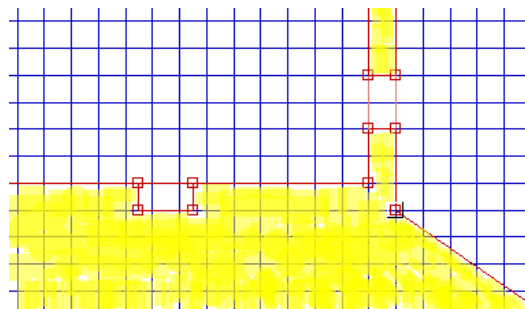
Enlarging The Exterior.

To make the external area has not been very difficult, but all was prepared. We had a sector and we have only changed the textures. When we try to apply the same procedure on the sides of our construction we meet with some conceptual problems. I hope for you they are not problems and everything seems tremendously logical.

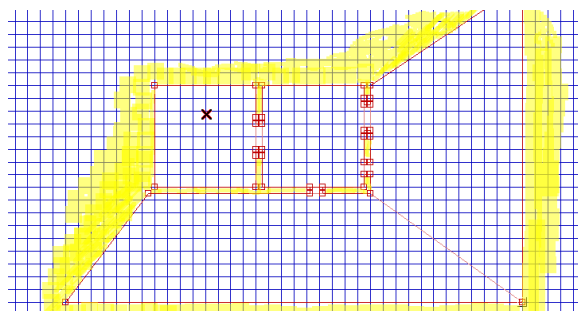
The first thing that we have to remember is that when uniting the sides of two sectors it is always transformed to a portal and they become transparent. Therefore, to create a new external area we must leave a solid area that it will be the wall. When designing this wall, also, we will have to take account of the corners so that the result is convincing.

Let us begin with a corner. We modify our recently created external area by changing to a grid of 0.5 and displacing the left lower vertex a little.

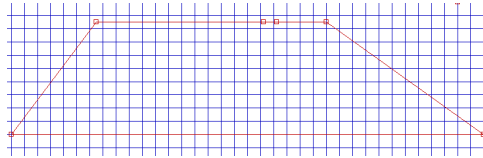
Again, so that it is clearer we have coloured yellow the area that is still solid:



Now already we can create our new sector by occupying the whole coloured area except for the space that will form the wall:



Now you see that the walls at the South and East are formed by not occupying the relevant areas. The separation between the two adjacent sectors constitutes the wall. The doors and windows are also taken into account. It is important to notice that to create the new sector we have not used four vertexes, but six. The additional two corresponding to the South window:

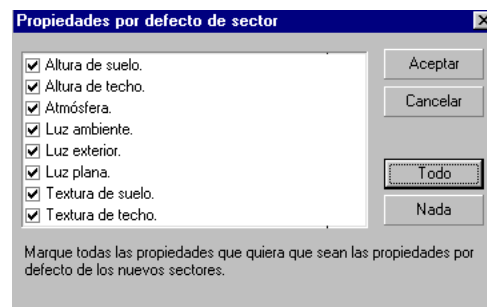


If these vertexes are not included we can have problems with the sector that constitutes the window. By the way, the NULL texture of the lower side of this window is no longer needed, as we have united two sectors and therefore formed a portal. As we know, the assigned textures don't matter. But, in my opinion, it is preferable not to leave NULL textures around, although they are part of portals. When we can change the design of the map we will see NULL textures where we don't want them. It is better to change it for any other texture.

With this new sector we have also created a portal to the right that is the union with the other external sector and that it will allow us to go for a walk in both areas.

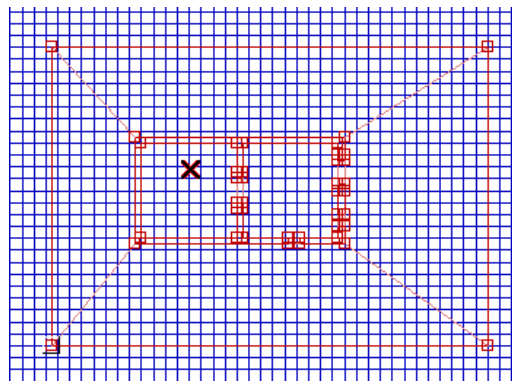
Now, we only have to take care of the textures: NULL to the roof and the lower side, and *sig1* to the floor. The upper side of this new sector constitutes the visible part of the wall of our house. We choose, for example a texture *flw* with a zoom of 20 and an angle of 90. The texture of the left side is unimportant because it will disappear when we create the following external sector. Lastly, to the roof set a height of 20.

To add the two new external sectors, it is comfortable to change the default values of the sectors. We select the recently created sector and we r-click to choose *Put default*. Then we choose *All* and OK.



The new sectors that we create will need the same properties as the last one that we have just done. We won't have to change properties of height, textures,...

The final result will be:



Before or after creating the upper sector we should remember that you must to adjust the right upper corner, like we did with the lower corner.

It is time for compiling....



With some extra work, we can change the textures with the 3D viewer. It is necessary to keep in mind that when changing a texture to NULL in the 3D viewer the landscape it is not shown, it is necessary to recompile.

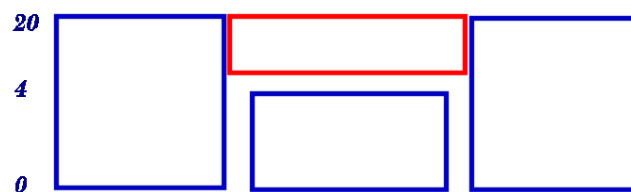
Well, we only have left the problem of the height...

Speaking Of Roofs.

With the roofs, we proceed as we did with the walls: where we create a space they appear. But it is more complex to visualize.

Now we have in our map an external area that surrounds to our construction. The external area has a height of 20 and it is necessary to understand that above that height there is a solid area with no end, even though you have put the NULL texture on it. In fact something similar happened with the NULL texture of our window: although we saw the landscape through it, behind, everything was solid.

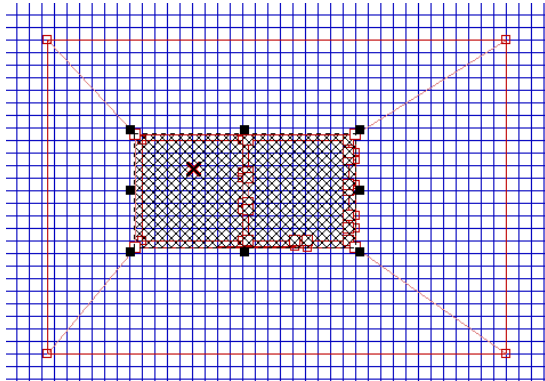
And our house is now performed with two holes of 4 units of height. Above, everything it is solid, up to 20 and further on. For that reason you see that column. Therefore we will dig another hole above our house:



As we see in the graphic the external sectors go up to a height of 20 while the interior ones only go up to 4. Both sectors are separated by the solid area that forms the walls.

Now we will create a new sector above those that constitute our house (drawn in red). The sector will begin at a height of something greater than 4 units, for example 4.2 and it will go up to 20. That solid area of 0.2 units will be our roof. Also, this new sector will join its four sides with the four external sectors to form portals and allow the light to pass freely above our house.

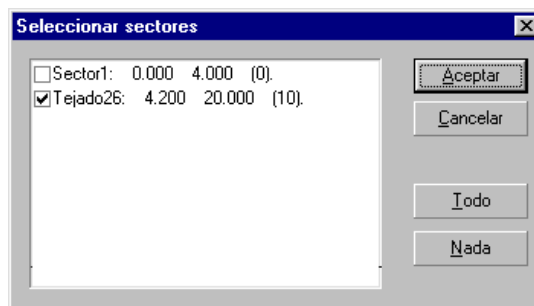
This means, like we see in the drawing, that the new sector extends to the vertexes of the external sectors, to the four corners. Therefore, with four vertexes we will have the newly built sector. We choose a grid of 0.5 and we make it:



The sector will have been created with the default attributes, so we should review it well before compiling. In the first place it is necessary to select it. If now we click on it, we will not only select this sector but also some of those that we had before. By double clicking, we can select the one that interests us.

In the properties window we change the height of the floor to 4.20 and the roof to 20. For the floor we choose the texture *ad4* with a zoom of 20 and angle of 90.

Now we will change the texture of the four walls simultaneously. For this, we select the sector again. In the selection window we will see that next to the names of the sector we see the heights of the ones above or below. This will help us identify the correct sector.



It can also be useful, like we commented in our first tutorial, to assign significant names to the sectors. For example, if our roof sector is called Sector26 we can change the name to Roof26. To append the number ensures that the name of the sector won't be repeated.

Once we have selected the sector, r-click with the pointer of the mouse inside the sector. We choose the option *Select->Edges*. All the sides are selected. From the properties window we change the texture for *flw*. It doesn't really matter which texture you choose because, like we have seen, all the side ones are going to join the sectors that form the exterior so they will be transparent, they will be portals.

Let us see what we have....



Every time I like it more... I wouldn't live there, but it is better than our first efforts. The roof is not very spectacular but we already have many elements already in place. We only lack one important thing: the slopes on the roof. But before continuing it will be as well to learn an additional trick to work with sectors of different heights.

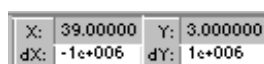
At The Top.

They are many game editors that work directly in 3D. With the LED, the designer has to make an additional effort to visualize in his mind what he is tracing in the Editor. It is difficult and the 3D viewer only helps when the sectors are well defined. If we make a mistake then the result that is displayed in the 3D viewer can be really confusing.

To work with an editor in two dimensions has an added difficulty. From the moment that our design includes sectors of different superimposed heights the confusion increases.

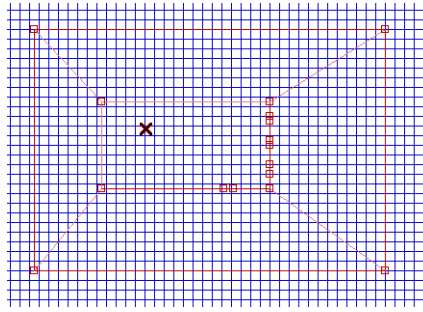
As we have seen, double clicking produces the Select sector window in which the different superimposed sectors and the heights are shown. This is very useful, but it is still very confusing.

To minimize this difficulty, the LED allows to delimit the sectors that are visualized by heights. Indeed, in the toolbar we can see, under the coordinates, two fields that act like a filter of heights:



By default, the values are $-1e005 / 1e005$ that we can consider as an unlimited range of heights. If we change the values to a range of $0 / 4$ in the editor, only the sectors whose settings are between 0 and 4 are shown. That is to say, the roof sector is not visualized because it begins in 4.2. The rest of sectors are completely in that range or they have part of their volume within the limits. (outdoors) Because of this they are shown in the Editor.

On the contrary, if we choose the range 4.2 / 20, our map is reduced to:



That is to say, only the sectors that have part in their volume in this range are visualized. If we choose the range 4 / 20 the map is shown complete, except for the sectors that form the doors and windows, because the sectors that constitute the rooms are part of the range (they conclude in 4).

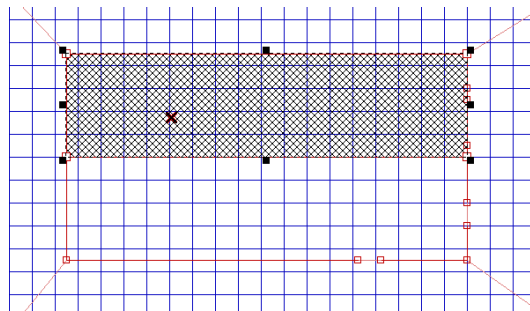
Playing with this system of filters is easy to simplify the design of the map to centre it in a solid area. If the option *View->View All* is enabled, then the filters of height don't work.

It is a very useful tool, but it is necessary to be careful. It is easy to forget that the filter is activated and to believe that an area is limited to the sectors that we see. Especially problematic is the fact that when we add vertexes they should be added to all the sectors. If the sectors are hidden the new vertexes won't be added to those sectors.

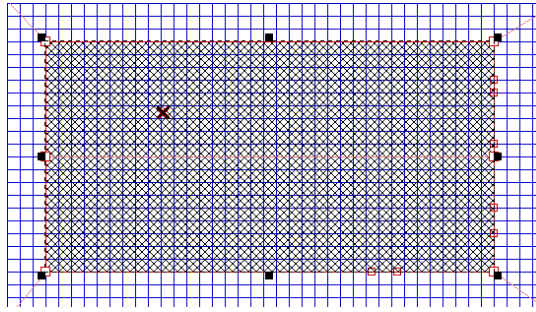
Slope, Please.

We return to our map to try to make a pitched roof. We choose a filter of height of 4.2 / 20 and we activate a grid of 0.5.

On our map we select the roof sector. We will try to divide it in two. We move the cursor to the lower side of the sector until it changes to double arrow \leftrightarrow we can drag it with the mouse to decrease the sector by half.



Now, continuing with the selected sector, we drag it down with the Shift key pressed, so we make an identical copy of it:



Now we have two identical sectors that together carry out the same function as the original roof sector. It is important that the grid is fixed at 0.5 and enabled so that the new sector fits perfectly in the hole left when we decreased the first one.

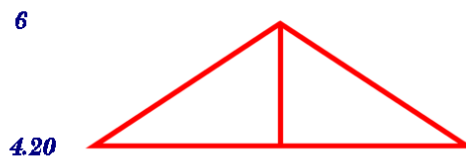
If now we compile we should not notice any changes in the map.

I just crashed the map. The LED has become crazy. It seems that a sector seems is faulty and I cannot continue... I mention this here to remind us to backup our maps. I make copies of the map in an almost obsessive way. They occupy little space, at least at the beginning. It will spare us a lot of grief if we make copies of our map on a regular basis.

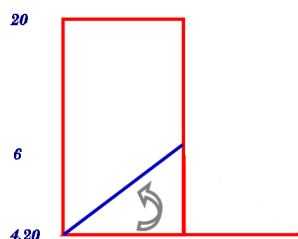
(Wise Words!.....Pro)

(You can tell when two sectors have united correctly when dragging them to a new location by looking at the wall that forms the Portal. It will change to a thin line (black). If it remains thick (red) then move it away and try again. If it still refuses to unite properly then it may be better to create a new sector at the required location. It is a good idea IMHO to work in small steps and in one area of the map at a time. Avoid jumping all over the place in one session. This way if you get a bad sector it is easier to identify it.....Pro)

The following step is to incline the floors of each sector , so a roof of two halves is created:



Each sector can be sloped on the floor and the ceiling. At the beginning this is not very intuitive. For example, in our roof sector what we have to slope is the floor. This may sound slightly strange, but if you think about it, it is logical. Each one of our two roof sectors begins in 4.20 and they end up in 20. What is necessary is to slope the floor from 4.20 up to a height of 6, for example. If we sloped the ceiling, we would in fact be sloping the sky!

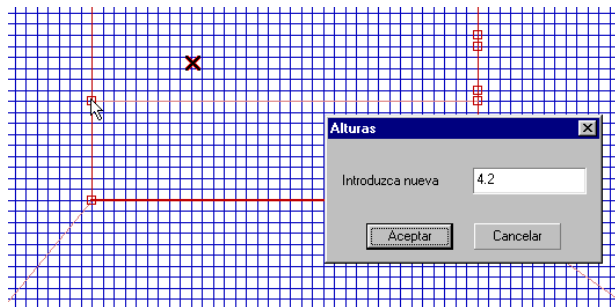


The inclinations are very useful. In some situations we may even want to slope down, that is to say, start at 4.20 and to sink the floor down to 3.80, for example. It is something to keep in mind in our designs.

To prove this with our roof we select the lower side of the lower roof sector. Make sure that you have only selected that side and not the two that constitute the portal (roof - external area).

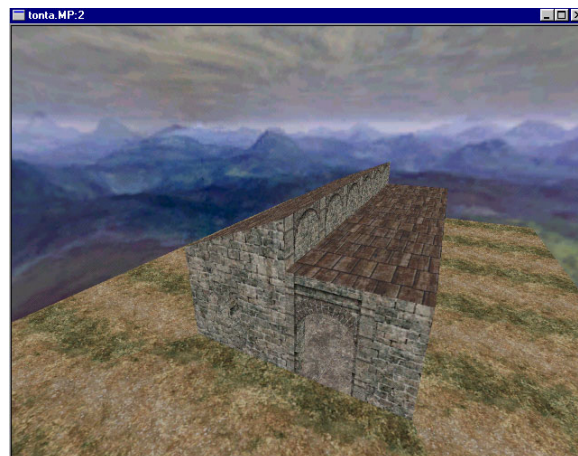
If when deploying the *Slope* menu the options floor / roof are disabled, then you have selected sides of more than one sector.

With this side selected we r-click and we choose *Slope->Floor*. Seemingly nothing happens but the LED is waiting for you to indicate the vertex you wish to slope up to. In our case it can be anyone of the two upper vertexes of the sector. We click, for example, on the left vertex (the one that the mouse pointer is on in the picture):



After clicking a window appears in which we are asked for the new height. By default it gives us the current height of the sector (4.2) and, if we don't change it, nothing will happen. But for our roof we put 6.0 as the new height.

We compile and we can already see our first inclined half:



I trust that is what you were expecting. Now we go to for the other side and to get a new surprise.

We choose the upper side of the upper sector. We r-click and we choose *Slope->Floor*. Click on the same vertex and again we indicate 6 as height.

We compile and we get the result below.



It is what we wanted, but with an unwanted side-effect. The texture of the arch over the door continues up... How to correct it? We will see next.

But before some comments...

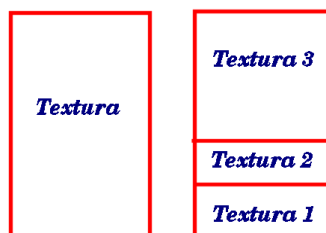
There is no known way with the LED of checking the inclination of a sector. You can only select the sector and to remove the inclination. Also, when reshaping a sector, adding or moving vertexes, we lose the inclination. This means that the recommended method to define slopes is making a previous design in paper. We should note all the data of heights, sides and used vertexes so it is easy to reconstruct the design. A map is a laborious task and after several months working in it is difficult to remember how each one of the areas was built. If we have lost something, or we need to restructure it, the additional information that we have on paper will be very useful.

Vertical Textures.

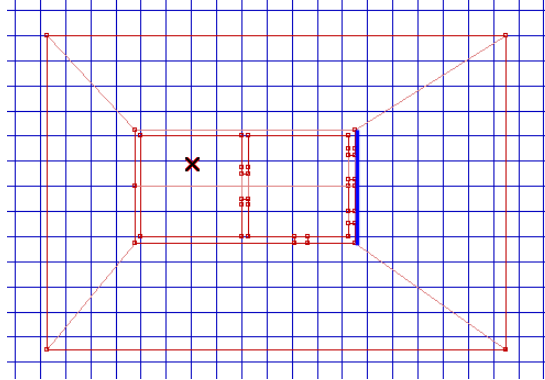
We have already seen how the possibility to add vertexes in a side allows us to place different textures on an individual side. This way a wall can belong to a single sector but show several textures alongside each other. It is like how we have built the doors.

Now we need something on the same principle but vertically. And here it is more difficult, because you cannot add vertexes since the editor works in the horizontal plane. You must add sectors

Indeed, if we have a sector that goes from 0 to 20 we will only be able to apply one texture. If the sector is divided in three, we will be able to apply three different textures.

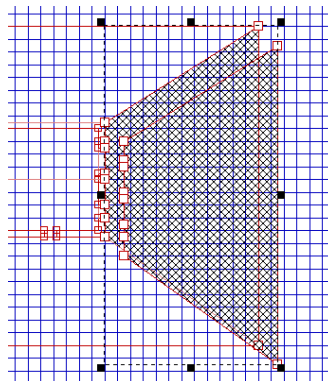


Now remember what texture you are seeing over the door. It is the texture of an arch, 1, that is assigned to the external part of the wall. It is not a unique assignment, but rather, as we know, each vertex delimits a different texture. This way we get the effect of an arch flanked by two columns and the rest simple wall.



Here is the method to change the texture vertically. It is to add a new sector. We will make that by dividing the external sector in two superimposed identical sectors: one of 0 to 4 and the other one of 4 to 20. As the rest of attributes are identical we will proceed faster using the copying of sectors method that we have used previously.

It is important to reset the height filter or we won't see all the sectors. We activate a grid of 0.5 and we select the sector. With the Shift key pressed we drag it a little toward the right until it is clear of the original sector.



We now select only the original sector and we reduce the height of the ceiling from 20 to 4.

When we have several selected sectors, when clicking in one of them nothing happens. To select only one we should click in a different sector and then select the one that interests us again.

Next we select the new sector and we assign a height to the floor of 4 and the roof of 20. Lastly we drag it until it is superimposed perfectly on the old one. For this purpose the activated grid is, as always, indispensable.

What have we got? We will see it better if we compile and we show the 3D view. You can now check if you can select the upper triangle that forms the roof independently from the lower part (the one that goes from 0 to 4). Indeed, there are five textures making that triangle because we have several vertexes to divide it in different areas. The simplest thing is to select them all, holding down the Control key, until the

triangle behaves like a single surface. Then, in the property window, change the texture to *pvolcan*, with zoom of 10, angle of 90, x = -155 and y=0. The result is.



I hope that your house looks like this. Better still, I hope that you have improvised something, a change of textures, added a second window perhaps and made a better house. I would also like this tutorial to seem too elementary to you now. Then I believe it that we would have succeeded in our objective. And if you have still not understood it, don't worry, it is not easy but you can attempt it again other time. If you want to go faster you can download the complete map from [here](#) and take a look. For sure you will find your problem.

Another Farewell.

I admit that the result is a not too attractive house, but that was not the objective. We leave in your hands the final decoration, because we believe that if we have been able to persist to this point, you already have many skills to enable you to build true maps. They are still more topics to explain, some related to the LED and many, perhaps too many, with the programming with python.

Somebody will explain it. For the moment we conclude here.....

Credits.

Today, it is difficult to remember how one has discovered the ins and outs of the LED. SlayerDDD continues being a reference, although he makes a lot that we don't see.

BirdMadBoy brought us a good handful of answers from RAS and we will never be able to thank him enough.

People of MoriaMod have always been an invaluable support and to see how others, such as Prospero or CrazyDwarf, went aboard in extraordinary projects is the perfect incentive to want to make it a little better.

The BODArena forums continue being a good resource and there is more information in the Blade Forever forums. Indeed, in this last forum this tutorial was born. To see new people, such as Khamen, Deladar, esfumato, Luthor, Silver, CK, Panzer_II, 7318... and many more that I cannot now remember, racking their brains to learn

how to manage the LED, has been the fundamental reason that this tutorial sees the light.

Again, Prospero has helped me a lot to complete a second dream.

Thanks to all of them.

@ Masklin – Octubre 2002 – Revisión 0.2

@ Prospero – November 2002 – English version

Footnote:

In the original Spanish, Masklin refers to our first map as '*La Caja Tonta*', a term which literally translates as '*The Silly Box*'. As I could not equate this phrase to a suitable English expression, I substituted the phrase '*First Sector*', which refers to the map that we have used to learn on.

¡Usted no tiene que ser tonto hacer esto, pero ayuda!

.....Pro