2011

idTech4.com

Editor: Simon Wadsworth

[DOOM 3 MULTIPLAYER LEVEL EDITING GUIDE]

Back in the days when Doom 3 was new and shiny Josh ‘b0rg’ Holmes, gaming editor at VIAVGA, created a fantastic guide to getting started with mapping. Its focus was multiplayer, but it’s just as relevant to single player. Sadly, the VIAVGA website seems to have gone, but we archived the content and created this PDF with the whole guide in.

1 INTRODUCTION

So you've played through the DOOM 3 single player campaign, then turned your attention to multiplayer and entertained the idea of creating your own levels. Unless you've spent some time getting familiar with Radiant you're probably wondering where to begin. Part 1 of this guide is designed to help get you started. Throughout this guide we will run through the basics of DOOMEdit, discuss some basic hints and tips that should help you to avoid some big mistakes before you even begin, go through the process of creating a room and connecting it to other areas as well as covering the use of various entities such as lights, triggers and more. By the time you finish reading this guide you should be ready to go forth and grow your own. The first few pages of this guide is pretty heavy on text and light on demonstrative images but that trend will change as you make your way through it.

The best way I've found to describe the process of making maps is to compare it to playing with Lego. It's certainly more complicated than that of course, but the basic principle is the same. The main difference between mapping and creating structures out of Lego is that when it comes to mapping, not only do you get to create the geometry; you also get to paint and light it. The other difference is that instead of using little plastic blocks that click together neatly with your hands, the mouse curser and editing tool functions perform the same functions your hands do with Lego. Thus, familiarizing yourself with the various tools and functions of DOOM 3 editing is paramount (because playing with Lego without hands would be way too challenging).

If you are completely new to the concept of mapping, there are a few terms that you need to familiarize yourself with. In the paragraphs above I mentioned Radiant, which is an evolving level editing program that has seen many incarnations over the years, going back as far as the original Quake. Since its earliest most humble version Radiant has improved out of sight, capable of creating levels for almost all Quake 2 and Quake 3 engine games. Although Radiant does currently support DOOM 3 level editing (in a beta version), this guide focuses on the official DOOM 3 version of Radiant (DOOMEdit), which comes packaged with the game.

The official packaged version is very similar to the build available at the Radiant website, though DOOM 3 itself was created with the former. The official version is also the program I used to create VIAVGA DM1, and thus also the version I am currently most familiar with. For this reason this guide will focus exclusively with the official packaged version of DOOMEdit.

It is fortunate that DOOMEdit is similarly styled to Radiant because there is already a plethora of information available and a lot of it applies to DOOM editing. There are, however, many differences – some subtle and some not so subtle. Where applicable I will try to focus on the differences between Quake engine and DOOM engine editing, which should prove somewhat useful to those of you who have tinkered around with Radiant in the past. Because there is already a wealth of information regarding Radiant, should you encounter a problem that isn't covered within this guide the answer you seek may well be contained in one or more of the many Radiant editing guides that already exist. Check out the links and resources section for a directory to useful forums and guides that are available. The glossary of terms on the Radiant website may also prove useful to you.

2 SETUP AND CONFIGURATION

Opening DOOMEdit is as easy as starting DOOM 3, accessing the console (by pressing CTRL, ALT and the tilde '~' key (right next to the 1 key) and then typing the word 'editor' and then pressing enter. As soon as you do this DOOMEdit should open up and several windows should be displayed.

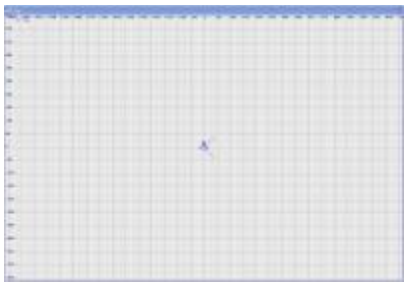
In the event that DOOMEdit doesn't open properly, or you experience any problems (such as an erratic mouse curser) you will probably need to make some changes to the system options within Doom 3. Make sure that you have anti-aliasing turned off. Some people have also experienced problems running the editor whilst DOOM 3 is set at full screen mode as well, so if any problems persist make sure you have switched over to window mode.

If you'd prefer a faster method of opening DOOMEdit (or don't want to keep switching between windowed and full screen mode), you can create a shortcut that will open the editor directly (without starting DOOM 3 first). To do this, create a shortcut of the DOOM 3 exe, right click and select properties, and then add this to the command line: set r_fullscreen 0 +set com_allowConsole 1 +editor

Clicking on that shortcut should now open the editor directly.

If this is the first time you have looked at a 3D level editing program by now you're probably thinking 'what the heck?' The good news is that it is easily possible to ignore some of the windows that pop up by default. There are numerous alternative ways of using DOOMEdit, and my preferred method is to rely on one grid window only (this is where the wire frame representation of your work will be displayed), the 3D camera window and the Inspectors window, which is where you can access and edit entities, materials (textures) and other media such as models and sounds.

By default, there should be 3 main grid windows available to you. One will be shown as 'XY top', another shown as 'XZ front' and another as 'YZ side'. As their name suggests, these windows are where you edit you level at the alternative X, Y and Z axis.



Whilst it is possible to switch between each window as required, my preference is to use only one of these windows, toggling between the XY and Z axis within the one window when required. Doing so will toggle the alternative view angles displayed in the window. This is done by clicking on the 'XYZ' toggle switch on the top toolbar. The reason I choose to edit within one wire frame window instead of 3 is that it allows me to enlarge the window and leave it in one place (no need to drag it around to expose other windows), which makes it easier for me to see what I'm doing. Clicking on the XYZ switch is faster (for me) than dragging windows about in order to expose others. This is particularly handy when editing with only one monitor because there is only so much room on the screen. If you decide to use one window instead of 3, just slide 2 of them out of view and then ignore them forever.

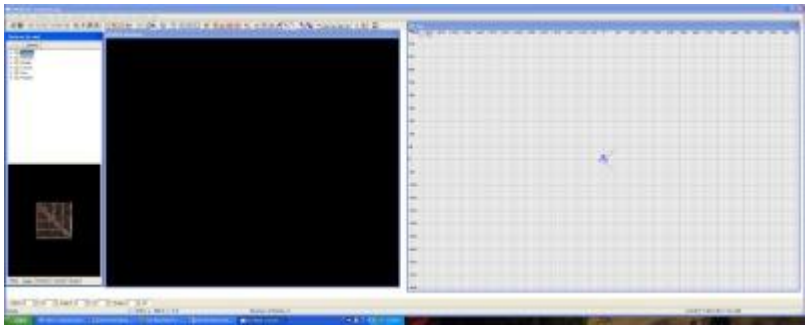
The 'Camera' window is particularly important, because it displays your work in 3D. Unlike previous like-styled editing programs, DOOM 3's camera window allows for real time previewing of your level, which is infinitely

useful. By default the camera window will display a basic 3D rendition of your map (in much the same way Radiant users would be used to).

Not only is the camera window handy for viewing your work as it progresses, it can also be rather handy for selecting brushes. Left clicking on any brush within the camera window will highlight it, which is not always terribly easy to do in the wire frame window if other brushes are overlapping it.

The camera window also takes on new meaning for DOOM 3 editing in that it also allows you to view your level as it would appear in the game. Pressing F3 will switch the camera window to Render mode, which is a genuine preview of your work. Pressing F4 activates Real-time render mode, which shows any recent changes and updates on the fly as you make alterations in real time. Not only is this very useful for experimenting with textures (because they often look different in game than they do in the texture window and camera 3D viewing mode) but it is also infinitely useful for and checking the effect of light entities. This feature is probably the most important difference between DOOMEdit and Radiant and it's significant enough to make it far more attractive (in this writer's humble opinion).

Although it is easily possible to create maps with one monitor (especially if it's a relatively large monitor), it is actually much easier to employ the use of two monitors. Not all video cards support this feature, but if yours does and you are taking the concept of building your own levels seriously, I strongly suggest considering using two monitors. This allows you to spread your working area across both monitors, allowing you to view more windows at once, as well as enlarging them for easier access and navigation.



Whilst 512 MB of RAM seems to be adequate for playing DOOM 3, it would appear that having more than 512 MB for editing purposes is a good idea – especially if you plan to view lots of material textures. DOOM 3 incorporates loads of textures (far more than most games), which means that there are plenty of them to cycle through when seeking the material texture you want. Providing your video card is beefy enough to run DOOM 3, it should be capable of running DOOMEdit without any problems.

3 HINTS AND TIPS – BEFORE YOU BEGIN

Before embarking on your first DOOM 3 map, there are a few hints worth knowing before getting too far ahead of yourself. Assuming that this is the first level you have ever attempted to create, it is good advice to experiment with the basics before getting too ambitious. Although you may wish to recreate a fully scaled version of say, the Death Star, trust me when I say that it is far too ambitious a project for your first attempt. Indeed, it is highly likely that your first attempt won't be fit for public consumption, so don't get too hopeful about releasing your first work to the world and receiving anything but praise. Keeping your first work to yourself and on your own private network can be a good idea. Starting with something basic is good practice, helping you to get your head around the vast learning curve ahead of you. As everything begins to fall into place you will discover various ways and means with which to realize your most ambitious goals.

When and if you reach the stage where you are ready to go forth and build something worthy of maximum exposure it can really pay off to plan ahead. If you have a pretty good idea of what it is you want to create the process is usually a lot easier. If you factor in some basic design principles as you plan it out certain potential problems can be overcome even before they occur. It's a simple matter of knowing what not to do. Overall the basic rules that applied to Quake 3 editing also apply to DOOM 3 although there are a few notable exceptions (covered in more detail further into the guide).

The most obvious tip worth heeding is that you really shouldn't plan on creating a level with massive and highly detailed open areas because DOOM 3 probably won't like them. The problem here is not one of size restrictions as such, but more so the ability to light them effectively. This not to say that it cannot be done (because the Monorail ride in DOOM 3 proves that it can be) but don't expect a smooth flowing deathmatch experience in big open spaces unless lighting is kept to an absolute minimum. So if you were hoping to recreate the Coliseum for example, I would suggest you reconsider.

DOOM 3 seems to like certain areas to be compartmentalized, either sectioned by doors dividing them and or with right-angled (or sharper) corridors and hallways connecting larger areas. This helps to avoid causing the engine to see into certain areas that otherwise cannot be viewed from where you are positioned. With correctly applied optimization methods (discussed in further detail later in the tutorial), engine rendering visibility can be significantly reduced, which is infinitely helpful in keeping frame rate speeds at acceptable levels. Most of these methods are only really effective if the map has been designed with these principles in mind. I assume that this is the reason that the default DOOM 3 levels contain a lot of doors and right angle corridors.

Scale is an important factor when planning or building a level and there are various related issues to consider. The first most important scale issue is that of player size. If you want players to be able to move around your level without getting their head stuck on overhanging structure, obviously it will have to be high enough for them to walk under. DOOM 3 player models stand to a total of 74 game units, so keep that in mind (I'm thinking a crouched player is 64 units high). The width of player characters is 32, so if you plan on including any slim corridors, tunnels or ducts make sure they are at least 32 units wide. Steps for staircases and stairwells need to be 16 units or less, though 8, 10 and 12 units seem to work better for this purpose. Experimenting with scale in the very earliest stages of development will help you to avoid making almost irreversible mistakes later on, so think about the kind of game play you are hoping to inspire within your level.

There is an important tool that can heavily assist in creating the scale you want, which displays the dimensions of the brush you are currently using. By pressing 'Q' any highlighted brush will have dimensions displayed next to it. Alternatively, this can be turned on by going into 'Preferences' by pressing 'P' and ticking 'Paint Sizing Info'.

It is very good advice to create brushes in specific sizes of a power of 2 where possible. Not only will it make it easier to fit and align brushes with ease, it will often make texturing those brushes much simpler. Many material textures are designed to fit faces of 16 x 16, 32 x 32, 64 x 64 or 128 x 128, or various combinations of those dimensions. Thus creating brushes with dimensions of a power of 2 will often make life easier to texture faces cleanly and evenly. More on this as we delve further into logistics.

One of the greatest things about DOOM 3 is that it is possible to open the default multiplayer maps within the editor. By doing so you can see exactly how they were constructed and also how certain features are applied. If you are a first time mapper, I strongly recommend you do this. Having said that, please resist the temptation to copy parts of these levels and pasting them into your own, because not only is it poor form it's also a pathetically lazy way to learn. Besides, even though these are official maps, don't take it for granted that they are perfectly constructed because you might be surprised to learn that they aren't.

To open a default map in DOOMEdit, first you'll have to open the pak000.pk4 file (which is located in the base directory). Winzip or WinRAR will make this possible. To use Winzip, Right click on the file, select 'Open With' and choose Winzip. It will take a while to unzip this file because it is rather large so put on the kettle while it does so. Once unzipped a bunch of folders and cfgs files will be displayed. Look for a folder called maps. Inside is a folder called Game. Double click that and then locate the folder titled mp (multiplayer). Each of the default multiplayer maps are listed here. If you want to be able to open all 5 of these levels, copy every file listed and paste them into a folder called maps, which should be located in your base directory. If it doesn't exist, create it.

Now when you open the editor and select open, each of these maps should be available for you to open and investigate. Beware, however, that if you make any changes to these maps it may well cause problems opening these maps in game later on so whatever you do, don't make any changes and save them.

Speaking of saving, this is something you'll want to do often as your work progresses. Each and every time you make a significant change, save your work under a new name. I stress saving under a new name because if you take an irreversible turn and you don't have any previous versions to fall back on you'll be mighty upset. I'm speaking from experience here, unfortunately.

4 CREATING A BRUSH

Perhaps the most difficult aspect of level editing is getting familiar with the editing program. Rest assured that you don't need to know every little detail before getting started but it certainly helps to know some of the basics. Let's start from the start - creating a brush.

A brush is a chunk of solid geometry. Brushes form walls, floors, ceilings and most structural objects within your map. Think of a brush as a piece of Lego that you can stretch, skew and snip in various ways to form the desired dimensions and shape. There are numerous ways to go about creating them.

So you're staring at a grid window, which is completely blank except for the grid matrix. By default the grid is formed of larger units of 64, each of them containing smaller units of 8. This is a fitting grid setting to begin with, although there will be times where you will wish to alter the grid settings when it comes time to create smaller detail brushes but for now the default settings are ideal.

The simplest method to create a brush is to place your mouse cursor within the grid window, click the left mouse button and then drag it across the grid. As you do this a brush will appear outlined in red. As your mouse cursor moves across the grid the brush dimensions will change dynamically. Providing you have Paint Sizing Info displayed, dimensions of the brush will be highlighted next to it.

In the case of this demonstration map, the very first brush created will make up the XY dimensions of the floor/ceiling of the first room. My intention is to create a central room that connects all other areas of the map. This room will be the focal point of the map and it will be relatively large. The dimensions I have chosen for this room are 1024 x 1024. This size is larger than the default viewable area in the grid window, so to increase the viewable grid I can press the 'Home' key. To decrease the size of the viewable area, press the delete key. This can also be done by the use of the mouse wheel, though I have become accustomed to using the Home and Delete keys primarily.

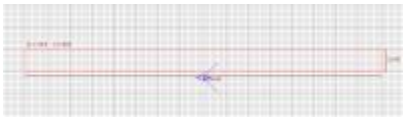
If you wish to move the grid around within the window (and often you will), right click and hold anywhere within the window and move the cursor around until you are happy with the perspective being displayed.

To increase the dimensions of any selected brush what you need to do is place your cursor outside of the highlighted brush outlines, left click, hold and drag the edge to where you want it to be. To decrease the dimensions place the cursor outside of the highlighted brush and push the edge inwards.

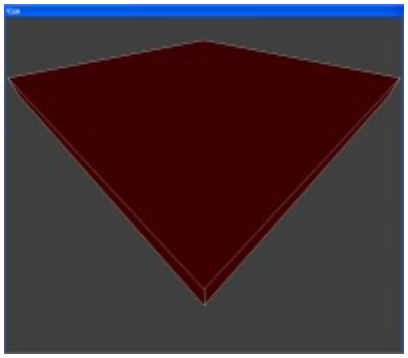
To move the brush around the grid, place your cursor inside the brush, left click, hold and when you move the cursor the brush will follow. Note that I have positioned the brush evenly along the 64 unit grid lines.

Although typically you will want to create a brush manually, there are other means of doing so. Displayed on the top menu is 'Brush' tab. Here you can select to create a brush of varying shapes, including brushes of anywhere between 3 to 9 sided brushes as well as arbitrary brushes of specified sides and primitive brushes which include cones and spheres. If you want to duplicate a selected brush simply press the space bar.

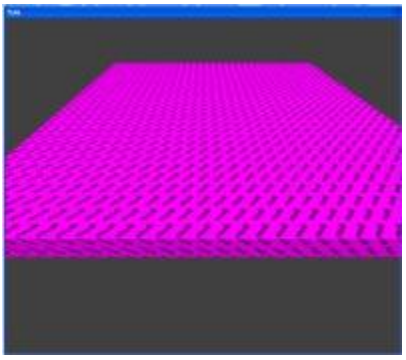
So the example brush is 1024 x 1024 units wide as displayed in the 'XY Top' window. The next step is to make sure the brush is of the dimensions you want in the 'Front XZ' and 'YZ Side' perspectives. Click on the 'XZY' tab on the top menu bar to switch to the 'XZ Front' perspective. Here we are going to make the height of the first brush 64 units, which is a pretty good choice for a floor and or ceiling brush.



You will note that this brush is also displayed in the 3D Camera window. If the brush is selected it will be coloured red. You can move your viewing perspective within the camera window and getting familiar with this feature is very important. First move your cursor into the camera window and left click to highlight it. The 'D' key will move your viewing perspective upwards. The 'C' key will move your viewing perspective downward. The arrow keys will move your perspective forward, backward and turn left or right respectively. The 'A' key will rotate your viewing perspective upwards and the 'Z' key will rotate the perspective downward. Experiment with these viewing perspectives by using those keys to view your newly created brush from alternative angles. Eventually you should become as proficient moving around the camera perspective window as easily and freely as you do in the game itself.



Now turn your attention to the Inspectors window and click on textures. You will note that the displayed texture will be shown as 'emptyname'. This means that the selected brush is not textured. Let's assign a basic texture to the brush. Click on the media tab at the bottom of the Inspectors window and a series of folders will be displayed at the top. Click on the textures folder and then select the textures folder that this exposes. Click the '+' key next to it and a series of folders should appear. These folders are where the many material textures are stored. Select the 'Common' folder and look for the texture called 'Caulk'. As soon as you do this the selected brush should turn into the caulk texture, showing up bright pink with the text 'Caulk' all over it.



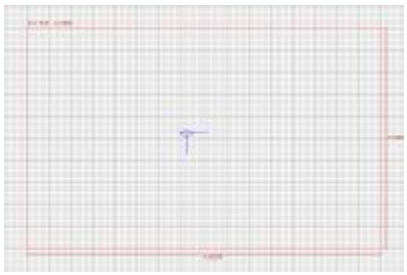
By now you are probably wondering why we have chosen to use the Caulk texture, and this does deserve explanation. The caulk texture is an invisible texture that does not render in game in any way. It is invisible as such, but it is solid. The purpose of using the caulk texture is to tell the engine that there is no need to display any viewable texture. Caulk textures are best used on any surface that will not be seen in game. So your next question will undoubtedly be 'if you can't see the caulk texture, why bother using it?'

The answer is that many brush surfaces will not be visible in game anyway, so it's better to texture those surfaces with an invisible texture to ensure that the engine isn't attempting to render any textures that you otherwise couldn't see anyway. When editing for Quake engine games it was particularly good practice to create all and any brushes with the caulk texture first, and then texture intended visible surfaces with whatever texture you wish to be drawn. Apparently this practice is not as important with DOOM editing because the engine does a better job of culling unseen textures anyway. Having said that, it's not always the case so it's a good idea to employ the same method.

5 MAKING A ROOM

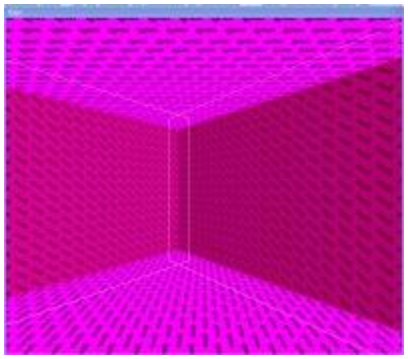
There are quite a few ways you could go about creating the skeleton for your first room - some of these methods will become obvious to you as you become more familiar with the tools. Lets begin by describing the easiest and fastest method, which is to hollow a solid brush. Let's highlight the brush we just created and switch to the Front or side viewing perspective. My intention is to make this first room with a relatively high ceiling. I want to see some rocket jumping going on here, so we'll need a fair bit of height to facilitate it.

With the brush selected in either the front or side viewing perspectives, left click outside the brush outline and drag upward. I'm going to go for a height of 640 units. So now I have one big solid brush that I want to turn into a room. I want the floor, ceiling and walls to be 64 units thick so I click on the 'Grid' option in the top menu bar and select 64 units. You will notice that the smaller 8 unit squares disappear inside the grid, leaving only 64 unit squares. Now with the brush still selected, click on the 'hollow' icon on the top menu bar. By now your solid brush should be hollowed with walls of 64 units. This is good, but we're not finished yet.



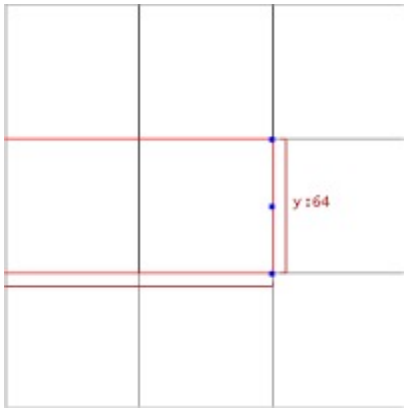
If you look closely, the separate brushes making up the floor, ceiling and walls are overlapping. This is not so good. Ideally, you want to avoid overlapping brushes where possible. In this case there is two ways we could go about it. One way would be to reduce the floor and ceiling dimensions by 64 units so that each edge fits neatly within the walls. The other way would be to reduce the height of the walls so that they fit neatly in between the floor and ceiling, which is the way we're going to do it now.

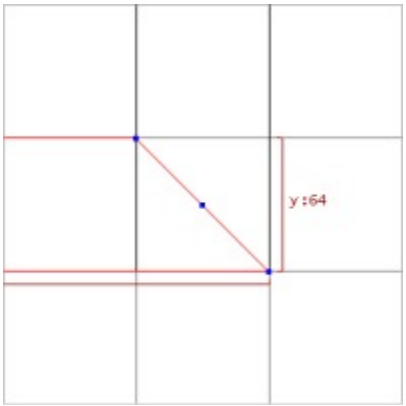
Go to the camera window and move your viewing perspective inside the newly hollowed room and press escape to unselect all of the highlighted brushes. Now hold down shift and then click on each wall brush to highlight them. With each wall brush selected, go to the front or side grid perspective. Place your cursor above the top of the selected brushes and move the cursor down so that the brushes shrink by 64 units to meet the top of the ceiling. Now do the same from the bottom and make the walls fit neatly against the floor brush. By now the walls should not be overlapping the ceiling or floor.

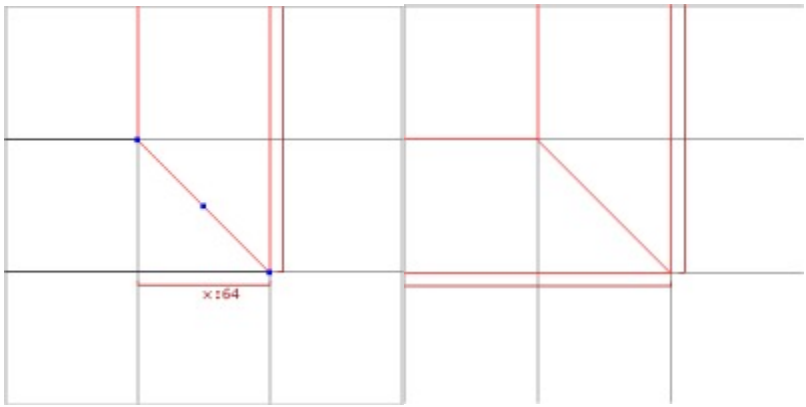


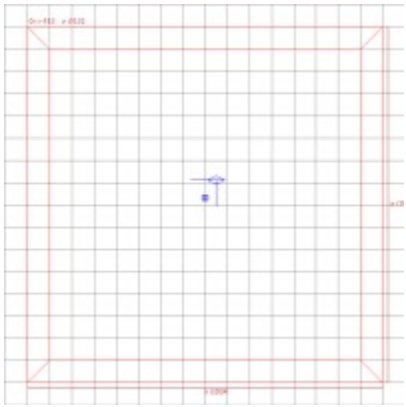
The walls are still intersecting each other though and that's the next thing we're going to fix. There are a couple of ways to do this. One would be to reduce the width of two of those walls by 64 units either side. Often this is sufficient, though there is another way to go about it which can sometimes be more suitable. That is to mitre the edges so that they connect on a 45 degree angle, which is what we're going to do.

Select one of the walls and position the grid window so you are looking at one of its ends. Then press the 'E' key, which will highlight the edges of the brush with small blue dots. Place your cursor on the inside dot and drag it inward by 64 units so that it forms a perfect 45 degree angle interesting the nearest wall brush. Now do the same to the intersected wall brush so that each edge aligns perfectly. Repeat this process until all walls are aligned neatly in the same way. (You may be asking why go to the trouble of doing this and it's a good question. The answer is that by doing so you reduce the amount of external surfaces that are drawn from 3 to 2. It won't make any difference if the outside surfaces of the walls won't be seen, but if they are it's probably worth doing. Here's a graphic example:

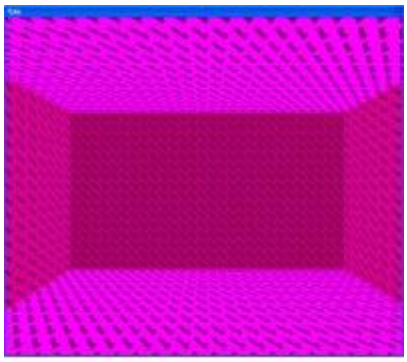




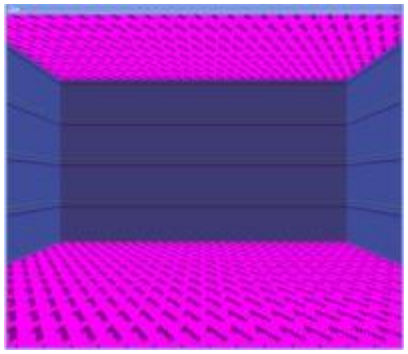




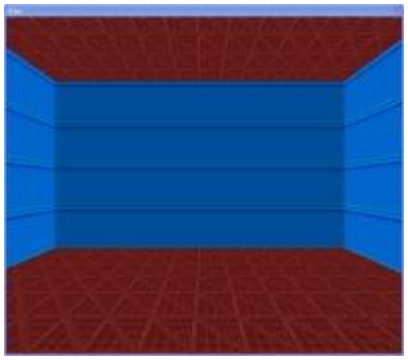
OK – now we have the basic skeleton of our first room. Before going too much further it is probably a good idea to have a look at it in game to ensure the dimensions really are what we were looking for. There are a few things we have to do before that is possible, however. Lets start by texturing the inside surfaces of the room, because if we leave the visible walls caulked we won't see anything. At this stage almost any placeholder texture will do – we can fine tune this later on.



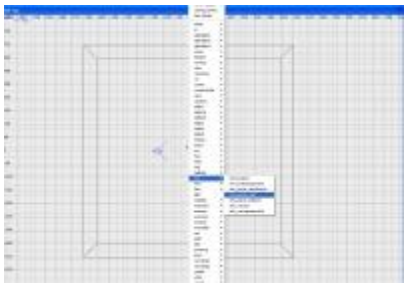
First we'll highlight each wall surface we intend to fit with a texture. Go to the camera screen, press and hold shift, ctrl, alt and left click on each wall surface. Then go back to the media/texture folders (in the Inspector window) and click on the base_wall folder. Clicking on any listed texture should texture the highlighted walls. When satisfied with the placeholder texture, deselect each surface by left clicking each of them or by pressing escape.



Now we want to repeat this process for the ceiling and floor, starting by highlighting their viewable surfaces using shift, ctrl, alt and left click. Minimize the base_wall folder and maximize the base_floor folder. Select a placeholder texture and then deselect the ceiling and floor surfaces. Now were almost ready to check out the room in game. Before we do there are still a couple of things we need to do.

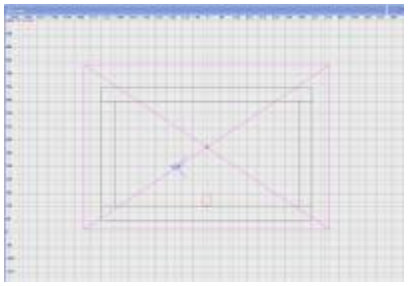


Our next step is to create a spawn location inside the room. To do this, go to the top viewing perspective, right click inside the room in the grid window and select info, and then info_player_start. A red box (32 x 32 units) will appear. Position this wherever you want to spawn. Then switch to the front or side viewing perspective and position the red box so that it is sitting flush or just above the floor. If you make the mistake of positioning it anywhere below the floor you will spawn inside the brush and will be unable to move.



There's one more thing left to do before trying to check out our first room – that is to light it. Unless we create some kind of light source the room will be pitch black. Switch back to the top viewing perspective, right click inside the room and select 'light'. A light entity will appear, represented in the grid window as a small red square surrounded by a larger purple square. Position it so that it is centred within the room. The purple lines represent the strength of the light. By default light entities will project light to a radius of 300 x 300 x 300 units. This can be modified to any dimensions you wish. An equilateral radius of 300 units won't be sufficient to light a room of this size, so for the purpose of testing the first room it will make sense to increase the size of the light radius. Press 'J' to bring up the light entity editor.

Click inside the radius 'X' field and replace 300 with, say, 600. Upon doing so you will note that the light radius displayed in the grid window has increased dramatically. Now switch the viewing perspective to front or side and position the light centrally. Note how the equilateral radius is stronger than it needs to be to illuminate the floor and ceiling. To change this press 'J' again to bring up the light editor window and in the Z field, replace 600 with say, 400. If the dimensions of your room are the same as described above, it should now look something like this:



Deselect the light entity. We're nearly there. Before opening the map in game, now is probably a good time to check out the real-time render mode. Press F4 to activate real-time render mode, which should show you how the map is going to look in game. Now let's have a look in game to ensure the size of the room is about right.

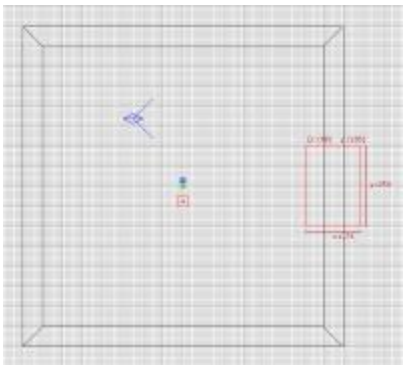
Save your map by clicking 'file' and then 'save as'. Choose a name (test1 for example) for your map. Now click on the BSP tab on the top menu and click BSP. Doing so will compile your map. Now you are ready to check out the first stages of your first room in game. Press F2 and DOOM 3 will start. Go to the console (ctrl, alt and ~) and type 'test1'. All going well, you should now be able to run around inside the room.


100

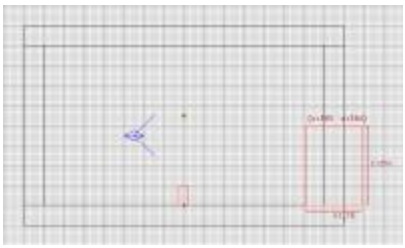
6 MAKING DOORWAY FRAMES

So now we have the bare-bones skeleton of our first room. We're satisfied with the size and scale of it and are now ready to connect it to other areas of the map. It would be possible to start dressing the room up with appropriate lighting and detail, but generally it is better practice to create the framework for the entire level first. If this is done correctly it can often be simpler to start adding eye candy later and that's what we're going to do. Let's start by creating some openings with which the rest of the level will connect to our first room.

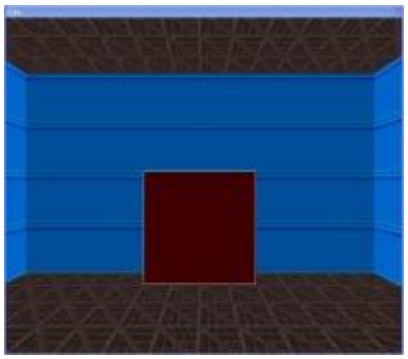
Once again, there are a few ways we can go about doing this. Let's start with the laziest and not so recommended way of doing it, which is to use the CSG subtract feature. Essentially, what we're going to do is to create a brush of the dimensions we want our doorway openings to be, then position them where we want them to be, and then we're going to use that brush to carve out a hole in the wall. In this case, I want the doorways to be 256 units wide (plenty of room) so I make a crush 256 units wide and using the top grid perspective, place it directly over the wall we're going to carve.



Then we go to a side or front grid perspective and make the brush 256 units high.

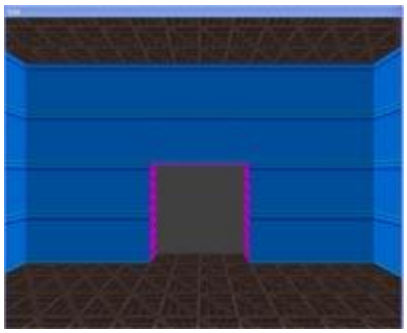


Before going any further let's do a quick visual check in the camera window to make sure the size looks about right.



Here comes the really easy bit. Move your cursor up to the top menu bar and click on the csg subtract tab, pictured below.

Left click and then move your cursor to the edit tab and select delete. The selected brush you used to carve the hole will be deleted and providing you haven't made a silly mistake there should now be a nice even hole in the wall.



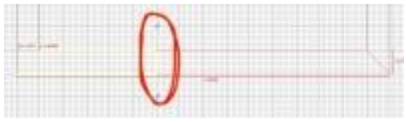
Now that I've shared the csg subtract feature with you, I should also add that this is not always the best way of going about it, even if it is the fastest. Often, this method will work just fine (and seems to cause less problems with DOOM editing than it did for Quake editing), but other times it can cause niggling issues – particularly if you try doing it with complex brushes. Below is a description of another way of going about that I suggest you pay attention to. Let's explore the use of the clip feature, which can be infinitely useful in various ways.

First, let's highlight the wall where we want to place a doorway.

Now place your cursor over the clipping tool tab on the top menu bar. Alternatively you can simply use the keyboard hotkey by pressing 'x'. This will activate the clipping tool.

Now move your cursor back to the grid window. What we want to do is place a clipping point on the grid where we want the side of the doorway to be, like so:

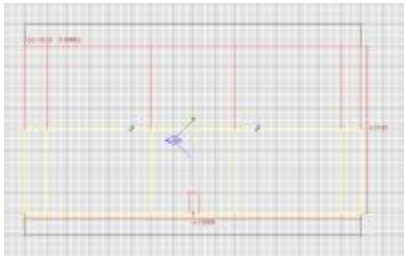
Now we're going to repeat the process on the other side of the wall. As soon as we add this second clip point, part of the brush that we intend to clip will be highlighted in yellow.



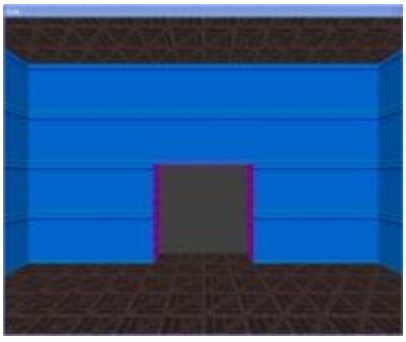
If we were to press enter at this point, the yellow portion of the brush would remain and the red section would be deleted. That's worth remembering, but it's not what we're trying to do now. What we're aiming to do is split the selected brush and to do this we need to press and hold down the shift key and then press enter. This will split the brush between the two clipping points.

Now repeat the process where we want the other edge of the doorway to be. Once done, it should look exactly like this:

We're nearly there. The next step is to create the height of the doorway by clipping it along the top edge. Toggle the grid perspective to a front or side view. Place the clip points evenly 256 units above the floor, press and hold shift and then press enter.



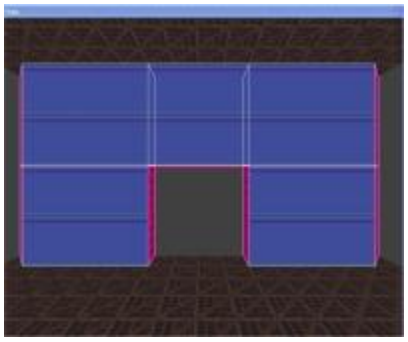
Providing you've followed the process to the letter, the wall brush will now be split nicely. All we need to do now is highlight the remaining brush where we intend for our doorway to be and then delete it (edit, delete). Now it should look much like this:



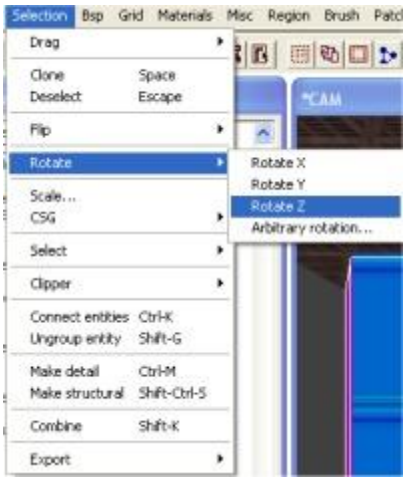
If you compare the results of either method described, you're probably asking yourself why it was worth it to go to the trouble of using the clipping tool when the csg subtract method was actually faster. The answer is that not only is the clipping tool more useful, using it properly also creates brush splits precisely where you want them and you cannot rely on the csg subtract tool to do that.

Ok, so we've created a hole in one wall, but my intention is to create a hole in each of the four walls. One way to achieve this is to repeat the process for each wall. Alternatively, you could always just select each remaining brush in the wall we've just carved and then duplicate them (by pressing the space bar) and then replacing ach of the other solid walls. Here's how.

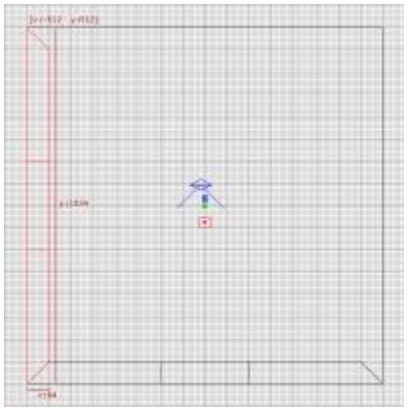
Begin by selecting each of the solid walls we wish to replace and delete them. Then select the wall brushes we want to replace them with.



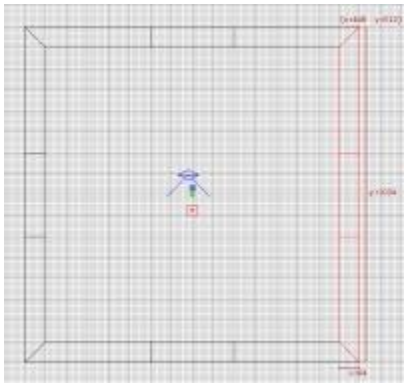
Make sure the grid window is on the top viewing perspective where the selected brushes will be highlighted in red. Then press the space bar. You will note that this will create a duplicate of the brushes that you had selected (which will now be de-selected). Now move your cursor to the 'selection' tab on the top menu bar, click 'rotate' and then click 'rotate Z'.



This will rotate the duplicated brushes 90 degrees. Now move the brushes neatly to fit against the mitred edge of the first wall.



Repeat this process until you have 4 walls again. Once done you should have 4 walls with identically positioned doorways.

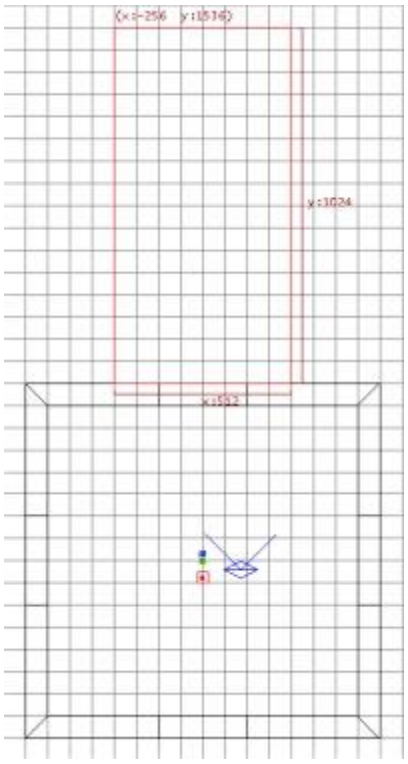


Now we're ready to move on and create some corridors with which we can connect other rooms or areas of the map.

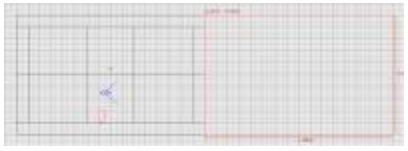
7 MAKING CORRIDORS

Now it's time to extend our skeletal framework beyond our first room. As with many other aspects of level editing, there are many techniques to go about it. One way (and perhaps the most time consuming way) would be to create the floor, walls and ceiling of a corridor individually. I like to cut corners as often as I can, however, so the following is a description of how to do it with a speedier (and equally effective) approach.

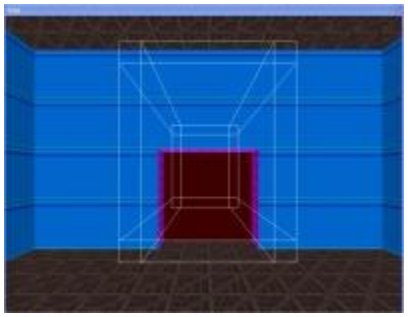
Let's begin by creating a big solid brush that will make up the framework for the first section of our first connecting corridor. Texture it with the caulk texture. My intention is to make corridors that are significantly wider than the doorway opening, so I'm going to make it 512 units wide.



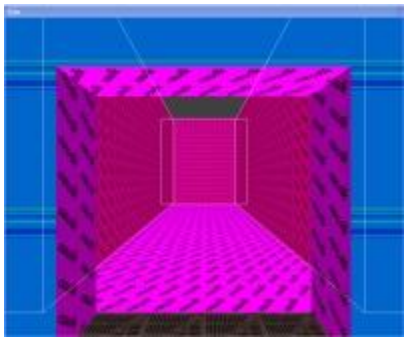
My intention is to place a stairwell that leads upwards in this corridor, so I'm also going to make it significantly higher than the top of the doorway. Your intention may well be different, so if you want a corridor ceiling that meets flush with the doorway, make this brush sit 64 units above the doorway.



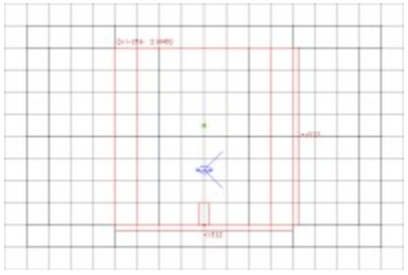
Because we're going to hollow this brush and make the walls, floor and ceiling 64 units thick, let's click on the grid tab in the top menu and select 64 units. Then click on the hollow tab on the top menu bar, which will hollow our brush leaving us with 64 unit thick walls. In the camera window it should now look like this:



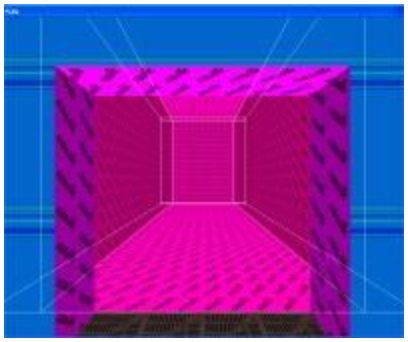
Press escape to deselect the highlighted brushes. Because we've taken a lazy approach and hollowed the brush to form the beginnings of our corridor skeleton, it has left us with an unwanted wall that blocks the doorway entrance. Using the camera window, select this unwanted brush and delete it. We now have a basic corridor, though as with the first room that we also hollowed out, wall, floor and ceiling brushes are overlapping and our next step is to adjust the wall brushes so that they fit evenly against the floor and ceiling brushes. So select the remaining wall brushes.



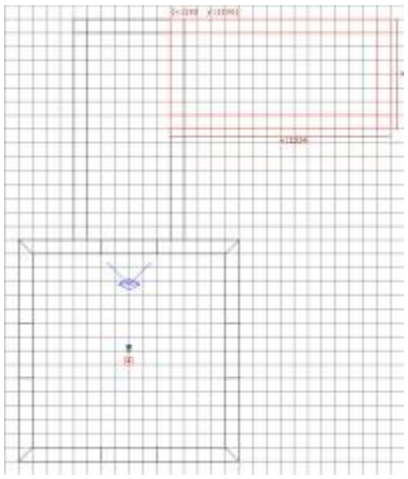
Now go to the grid window and switch to a front or side viewing perspective. Shrink the height by 64 units from the top and from the bottom so that the walls fit neatly between the floor and ceiling.



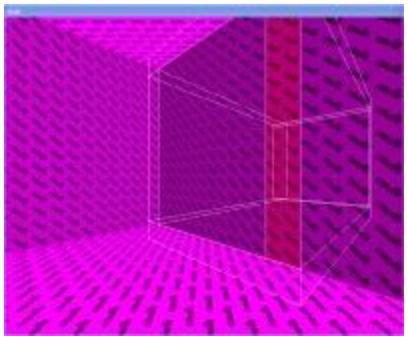
Our corridor is going to make a right angle (remember what I mentioned earlier about compartmentalizing areas for easy optimization?) and we're going to take a simple and easy approach to creating this. Start by selecting the walls, floor and ceiling of the corridor and switch to the top viewing perspective in the grid window. Press the space bar to duplicate them.



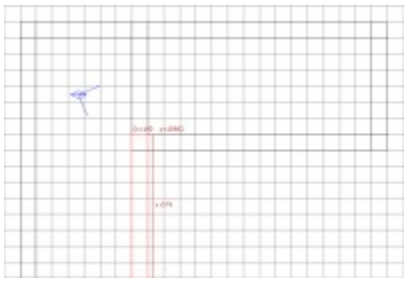
Just as we did before in the previous chapter, we're going to rotate the duplicated brushes by choosing selection, rotate, rotate Z. Now let's fit them neatly against the corridor we created previously, like so:



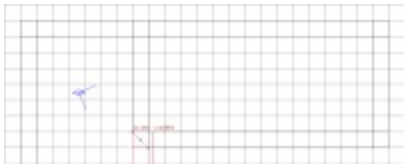
In the camera window it should look much like this:



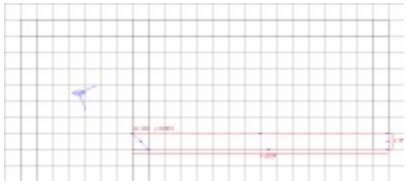
Notice how the right wall of the first corridor section blocks off the entrance to the second section? We're going to sort that out right now. Deselect the highlighted brushes and then select the offending wall brush. Now reduce its size so that it overlaps the brush it intersects.



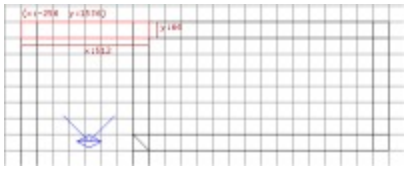
Remember how we discussed mitring the corners of intersecting brushes? Here's a good example of where this technique is well worth doing. Once again, below is a graphic example of how to do it. Press E to highlight the edge corners of the brush and drag the top right point downwards by 64 units.



Press escape to deselect this brush and then select the brush it intersects. Press E to highlight its edge corners and drag the bottom left point by 64 units to the right so that it fits neatly with the angled edge we just created with the brush it intersects.

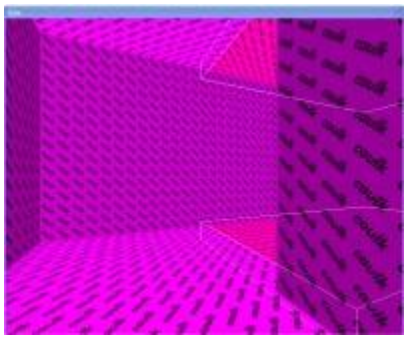


Done? Good – but there's still a few overlapping wall brushes to deal with. First of all, let's delete the now unnecessary wall brush at the end of the first section of the corridor. It's now unnecessary because we can use the wall brush of the newly created section of the corridor to form the entire wall if we wish. Select the wall brush and delete it.



Now select the other wall brush that runs along the same plane on the grid, and drag it to intersect the left hand side wall brush. Then mitre both of the intersecting edges with the technique described above.

One more thing to do before cleaning up our newly created right angle corridor, which is to eliminate the overlapping ceiling and floor brushes. Select both the ceiling and floor brushes of the right section and reduce them in width by 64 units so that they align neatly with the floor and ceiling brushes of the first section. In the camera window it should now look something like this:

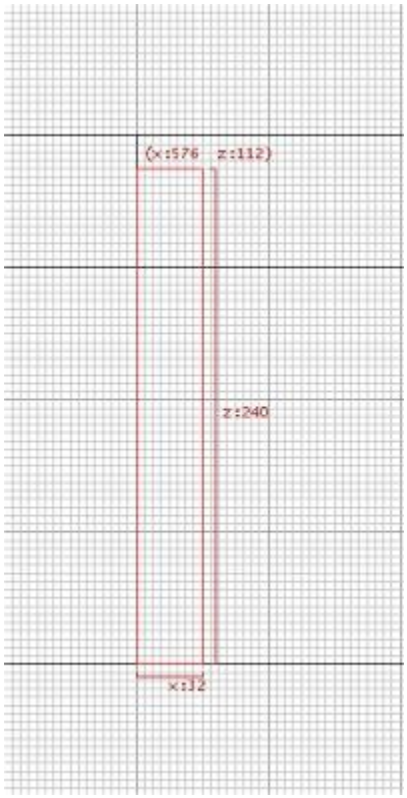


By now you might be starting to get a bit of a sense as to why it can pay to plan ahead (as mentioned at the very beginning of this tutorial). If you haven't put any thought into where this corridor is going, you will probably have to go back and make various adjustments to accommodate what ever your level evolves into. In any case, this is a quick and efficient method of creating a right angle corridor and as also mentioned toward the beginning of the article, this is something you will probably want to do fairly often.

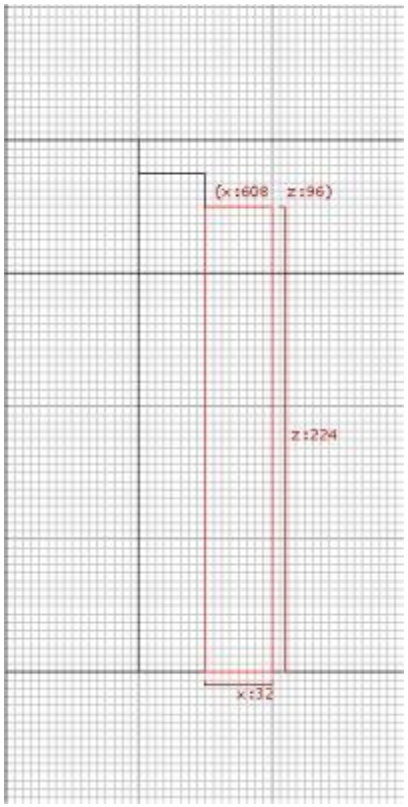
8 CREATING STAIRS

If you think creating stairs should be easy, you're right. There are a lot of different types of staircases and when it comes to creating them with a level editing program there's probably an equal amount of ways to go about doing it. Because I'm going for a fairly uniform look for the VIAVGA DM series of maps (for reasons that will become apparent sometime in the future), the stairs I'm creating for VIAVGA DM2 are similar to those seen in DM1. Here's how I'm going about it.

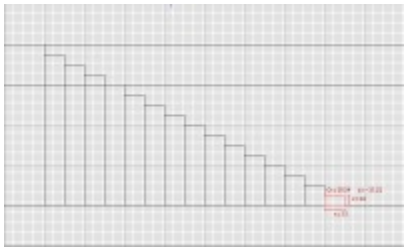
These stairs are to be made of steps of 32 units across and 16 units high. 16 units is the maximum and I'm using this step up height due to the staircases being relatively steep. Let's begin the process by creating the first step by making a brush that is 16 units lower than the upper edge the steps are pressed up against (using the side or front viewing perspective).



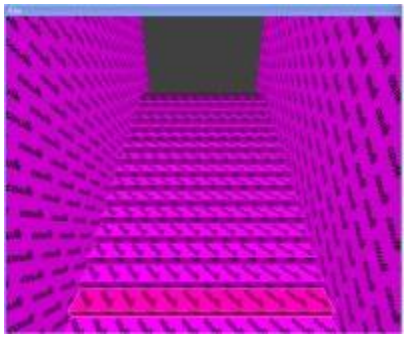
You will notice that the width of this brush is 32 units (if the width was any less, the staircase would likely end up being unnaturally steep). The bottom of this brush is neatly aligned with the floor they are placed upon. Once again I have started off with a brush textured with the caulk texture. Now that I am satisfied with it's potion I simply press space bar to duplicate the first step, then move the new step in alignment with the first. Then I shrink this new step by 16 units.



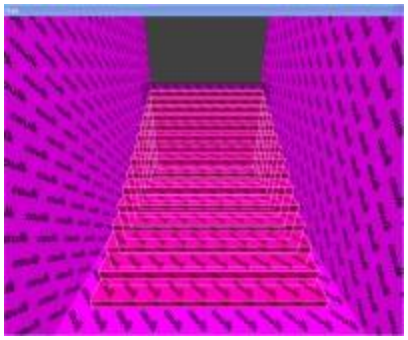
Then I simply repeat this process until there is a single step sitting 16 units from the ground.



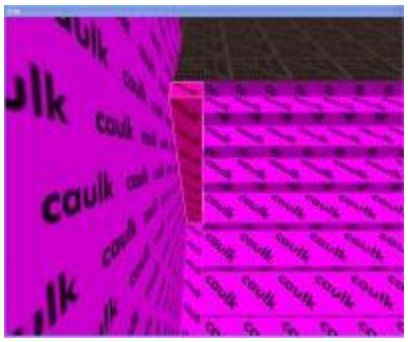
If you were mapping with a total lack of vision or total abundance of laziness, you could just texture it at this point and you'd have a set of stairs. Unfortunately they'd be pretty plain and wouldn't inspire anyone (as you can see by the camera window screenshot shown below), so let's take it a little further.



Let's start by creating a nice trimmed edge for the stairs. Select each step brush in the camera window, go to the grid and toggle to the top viewing perspective. Now reduce their width by 16 units (it's up to you how many units wide you decide to make the trim) each side. This is how it should now look in the camera window:

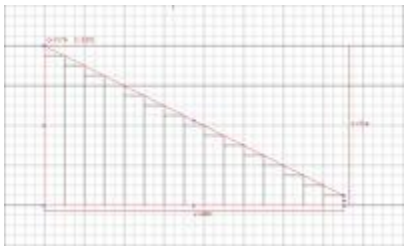


It still looks rather ordinary doesn't it? Our next move will bring us closer to the desired effect. Deselect each highlighted brush, and then reselect the very first step you created. Duplicate it and then shrink its width so that it fits into the 32 unit gap (that has been created by shrinking each step) on the left or right side of the steps. Raise it by 16 units so that it is level with the floor that the steps are leading up to.

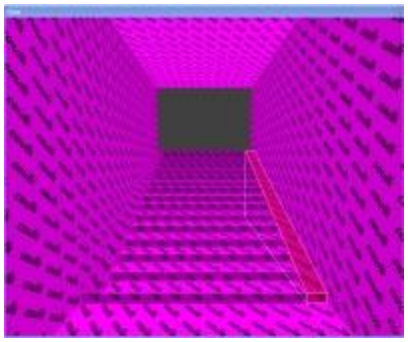


Switch to the front or side viewing perspective (whichever gives you a side on view of the staircase) and then stretch the brush so that it aligns perfectly with the last, smallest step.

Now press E to highlight the edge corners of the selected brush. Drag the upper right point down to the right corner edge of the smallest step, like so:

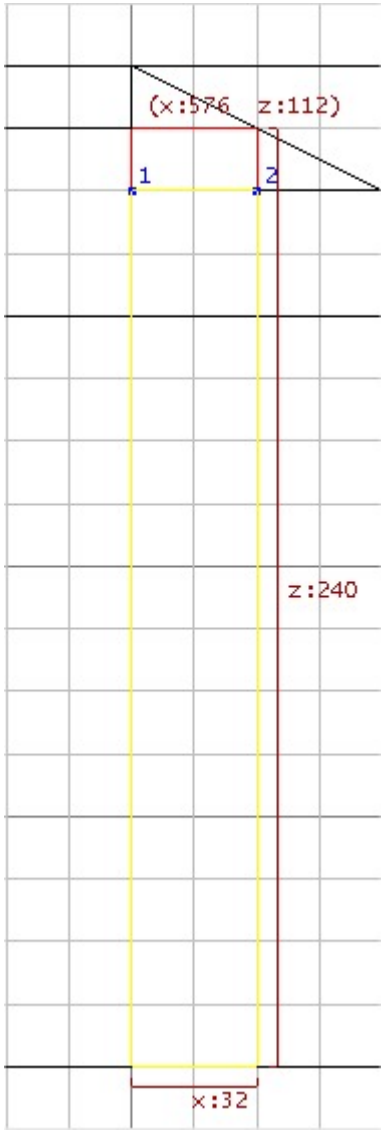


Press E again to remove the highlighted edge corners, and then press space bar to duplicate the brush and then move it into position on the other side. Now we're getting somewhere.

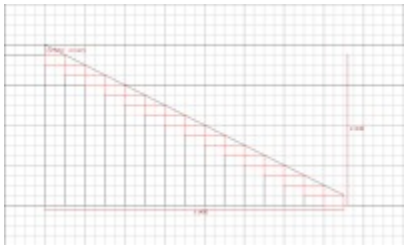


Our next step may seem a little unnecessary, but there is method to the madness. What we're going to do it clip the visible edge of each step so that when it comes time to texturing the front ledge of each step, the texture will only be drawn on the visible surface and not the entire brush (stretching all the way to the floor). Because the smallest step is only 16 units off the floor we won't need to do it to that one.

So, press X to activate the clipper, go to a front or side viewing perspective in the grid window and beginning with the top step, let's start clipping. We've covered this technique already, but just in case you need a refresher already, here's a graphic example:

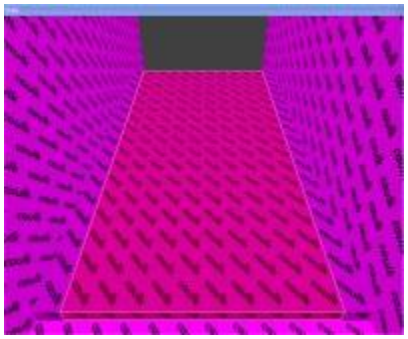


And here's an example of a fully clipped set of stairs (I've highlighted the viewable sections of each step to make it easier to see):

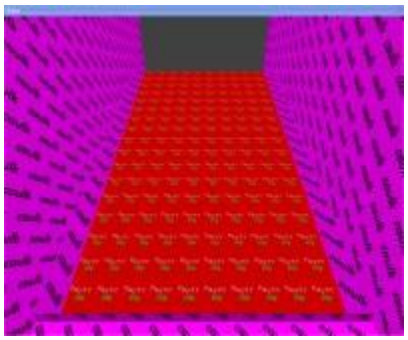


In the next part of the tutorial we'll cover the topic of texturing, and we'll use these steps as our first example. But before we go on to do that there is one final thing to do to make sure the steps function smoothly. That is to create an invisible 'player clip' brush to cover the steps with. In effect, this will make the steps act more like a ramp and doing so makes running up and down them smoother. Failing to do this will result in a noticeably bumpy ride. So here's what we're going to do.

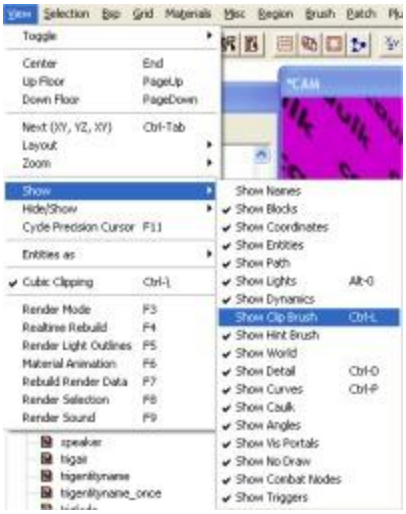
Select one of the side trim brushes we created and press space bar to duplicate it. Move the newly duplicated brush so that it fits neatly against the trim brush we just duplicated it from. Then stretch the new brush so that it fits neatly against the trim brush on the other side, like so:



Leave it highlighted and then go to the texture window, click on the media tab at the bottom, select textures, then textures, and then open up the 'common' folder. Look for a texture called player_clip. Double click it and the brush you just created should change from the caulk texture to the player_clip texture. Press escape to deselect it. You're done!



In the event that the brush you just created seemed to disappear when you deselected it, don't worry – it's still there. By default clip brushes are rendered invisible in the editor just as they are in the game. If you want to be able to view your clip brushes (and from time to time you will), all you need to do is go to the 'view' tab on the top menu bar, click 'show' and create a tick next to 'show clip' brush.

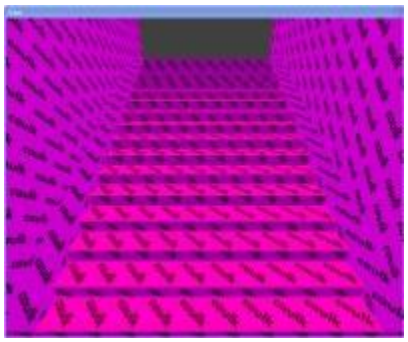

View Selection Bsp Grid Materials Misc Region Brush Patch
Toggle
Center End
Up Floor PageUp
Down Floor PageDown
Next (XY, YZ, XY) Ctrl-Tab
Layout
Zoom
Show
Hide/Show
Cycle Precision Cursor F11
Entities as
Cubic Clipping Ctrl-L
Render Mode F3
Realtime Rebuild F4
Render Light Outlines F5
Material Animation F6
Rebuild Render Data F7
Render Selection F8
Render Sound F9
Show Names
Show Blocks
Show Coordinates
Show Entities
Show Path
Show Lights Alt-0
Show Dynamics
Show Clip Brush Ctrl-L
Show Hint Brush
Show World
Show Detail Ctrl-D
Show Curves Ctrl-P
Show Caulk
Show Angles
Show Vis Portals
Show No Draw
Show Combat Nodes
Show Triggers
speaker
tigair
tigerentityname
tigerentityname_once

9 BASIC TEXTURING

If you've followed my advice and have been making most or all of your brushes out of the horribly ugly caulk texture, by now you're probably starting to feel like you dropped a few tabs of acid and are wondering what the hell I'm trying to do to you. Well, you do get used to staring at the caulk texture if you develop levels like this, but I admit it can start to get on your nerves. Sooner or later the time comes to start texturing visible surfaces and for the sake of the exercise, let's start sooner. But try to remember that as you create new brushes it is good practice to make them with the caulk texture.

Let's begin by texturing the steps we created in the last chapter. A common mistake many first time mappers make is that they try to texture each and every single surface individually, and although often you have to, sometimes you don't. Steps are a good example of this.

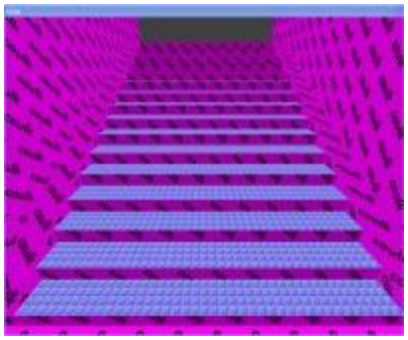
To select a brush surface to texture it, all you have to do is hold and press shift, ctrl and alt and click on a brush surface in the camera window. If you keep these keys pressed and click on other surfaces they will also be highlighted, allowing you to texture each surface highlighted in one swoop. I'm going to begin this demonstration by texturing the top side of each step. Using the method I've just described, here's what it looks like after I've highlighted each surface I intend to texture:



Now I'm going to go forth and choose a texture to put on them. To do this, go to the textures window, click on media, textures, textures, and select a folder to view available textures. Because steps are considered a 'trim' of sorts, I'll probably find a suitable texture in the base_trim folder.

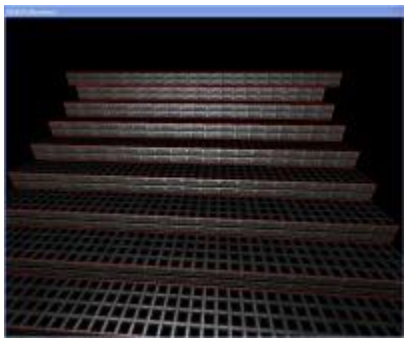
I like the luxury of looking at all available textures in the texture window, and the best way I've found to do this is to scroll down through each texture in the folder I'm investigating until I've scrolled through them all. As soon as I've cycled through each texture I then click on the texture tab (just to the right of the media tab. Now all of the textures I've just cycled through will be displayed in the textures window, and doing it this way is a lot easier than any other method I've experimented with.

There are quite a few textures in the base_trim folder that are well suited for using on the top of steps, though the steps I have created are rather wide and none of the available textures seem to fit too well. I could stretch them to make them fit (we'll discuss this in more detail later) but instead I've decided to use a texture that tiles better, so as to fit the over-sized dimensions of my steps. In the end I've found something that looks to work nicely in the base_floor folder.

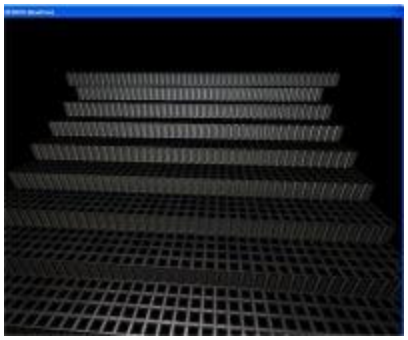


When experimenting with textures it's a really good idea to preview it using the camera render mode (press f3). Unfortunately, if there is no light source nearby all you're going to see is blackness. Therefore it pays to create a placeholder light entity to illuminate the area so you can get a feel for how well the texture works where you have placed it. To do so you can either create a light entity and place it nearby, or otherwise just copy an existing light entity (select and press space bar) and move it nearby your newly textured surface/s. Here's how the texture I've selected looks after putting a placeholder light source nearby.

I may or may not change this texture later on. Perhaps I'll stumble upon something more suitable, or otherwise decide my first choice wasn't so good after all - but it'll do for now. The next step is to texture the front surface of each step by repeating the process as described above. I like my chances of finding a suitable texture for this surface in the base trim folder, as the front surface of the steps is smaller than the top surface. Just as I suspected, there are a bunch of textures here that would work pretty well. I decide upon this one for now:



But I'm not done with this texture yet. I have a sneaking suspicion that it will actually fit better if I rotate it by 90 degrees. To do this, click 'S' to bring up the surface inspector tool. A window will pop up with a variety of useful options. The option we're concerned with at the moment is the rotate field. By clicking on either of the up/down arrows any selected texture will rotate by a predetermined increment (15 units by default).

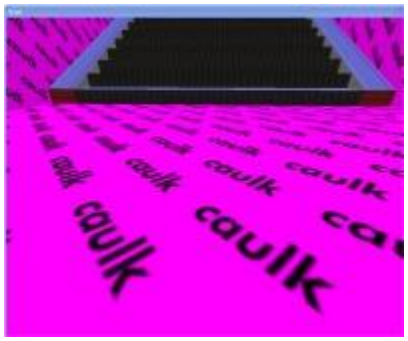


My suspicions were well founded. Note how the groves of the rotated texture now match up to the edges of the texture of the top surface texture? The more you can achieve this, the more natural the placement of textures will appear.

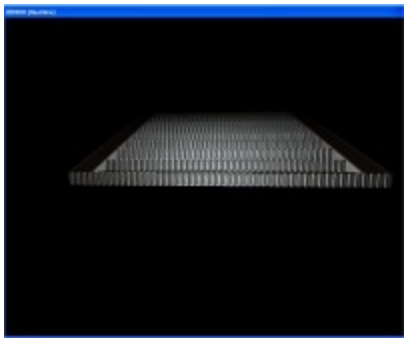
Now that I am satisfied (for the time being) with the textures I've chosen for the steps, lets move on to the trims on either side of the steps. First, press and hold shift, ctrl and alt and left click on the top surface of each trim. Then find the texture you want to use. I've decided to use bluetex3k2, which is located in the base_trim folder. Once again I will have to use the rotate feature to rotate the texture by 90 degrees.

Once rotated it looks pretty good. Now its time to move onto the side of trims. It's going to make sense to use a plain texture here (with no detail lines), so once I've highlighted the visible surfaces of the trim sides I select the texture stesilverswatch which is also located in the base_trim folder. It works nicely. Now there's only one more thing to do before the steps are fully textured.

The front edges of the trims are still caulked, so I have to find a suitable texture to place here.



It's a pretty easy choice because these surfaces align perfectly with the front surface of the bottom step. Therefore it makes sense to texture it with the same texture.

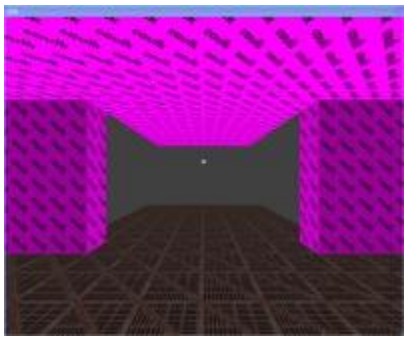


It works for me. The steps are now fully textured. That was pretty easy wasn't it? Texturing can get more complex than that, however, so we'll discuss some more advanced texturing features and techniques in the next chapter of the tutorial.

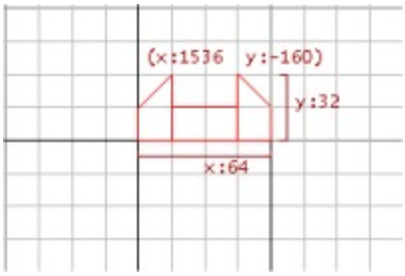
10 MAKING A DOOR

One level feature that DOOM 3 seems to really like are doorways because when created correctly, they act as suitable vis blockers that go a long way toward optimizing frame rates within the map. Placed with some level of foresight, they prevent the engine from seeing into levels that the player cannot (until the door is opened, that is). In any case, most maps will likely feature a doorway of some kind and the chances are high you're planning to put at least one of them into your own level. Below is a description on how to go about doing it.

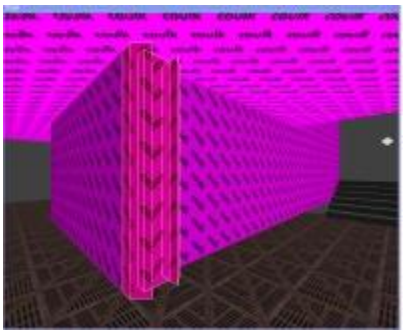
Before getting too far ahead of ourselves, let's first create a frame for the door. Here's a picture of the entrance I'm planning to put my first door.



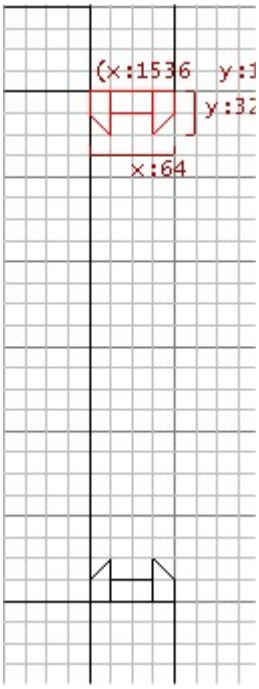
There are a few ways you could go about making a frame for the door. You could simply create it so that the frame meets the edge of the door flush on both sides. That method works fine, though I'm planning on having this particular door indented into the frame, which I think looks a little more realistic. The process for doing this is rather simple, and if you've followed the basic steps of this guide up until this point you would know everything you need to know to make your own. Below are some images that show the step by step process of building my frame. Here's the top down view of one side of my frame.



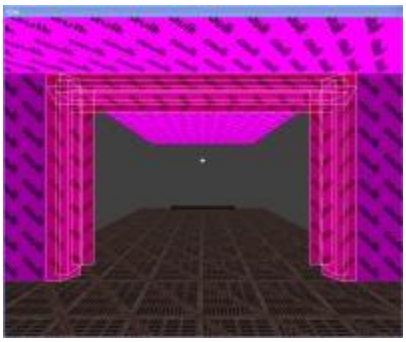
Here's how it looks in the camera window:



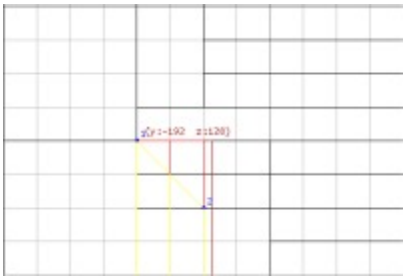
Then by pressing space to duplicate the left side, I then rotate the brushes on a Z axis twice until it mirrors the left side. Then I position the duplicated brushes on the opposite side of the intended doorway, like so:



I also want this frame to skirt the top of the doorway as well, so to achieve this I start by duplicating the right side of the frame and then rotate the newly duplicated brushes on the X axis 3 times. Then I position the top section of the frame flush against the ceiling and stretch the brushes so that they interest both left and right sides of the frames. The result of doing this looks something like this:

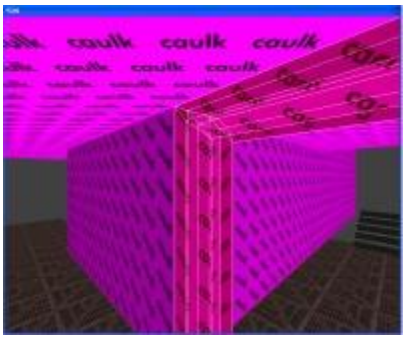


There's a slight problem with this picture that we have to address, which is that each of the brushes are intersecting, when in fact it is much better practice to have these brushes neatly pressing flush against each other. Here's another example of where the clipping tool can be invaluable. To fix the problem I highlight each brush on the left side of the frame and then clip the top end of them on a 45 degree angle.

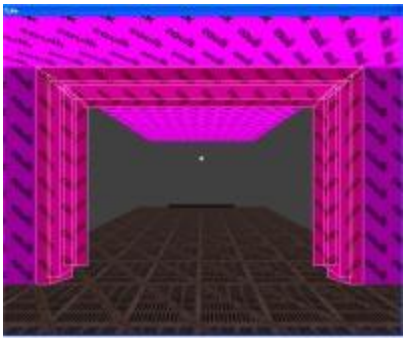


Now repeat this process on the top section of the frame so that both clipped edges meet each other flush on a 45 degree angle, like so:

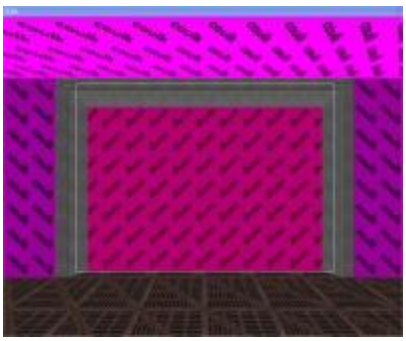
I've highlighted the newly clipped frame brushes on either side to make it easy to see how it should now look:



Now I'm going to repeat the process on the other side of the frame.

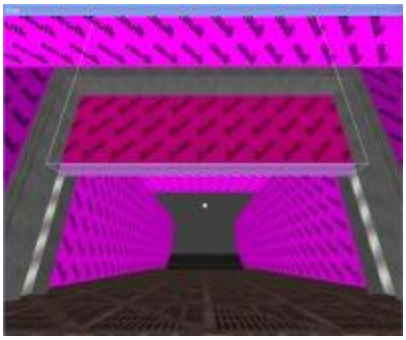


Now we have the basis for our doorframe, but before moving on I'm going to texture it with a placeholder material (which may or may not remain) quickly. I've used a couple of trim textures (from the base_trim folder) for this purpose. Ok, so we have a basic door frame. Now let's make the door to fit inside it. First create a brush that fits neatly between the floor, ceiling and either side of the inner door frame inserts. In the case of my doorway, it looks much like this:



Not all surfaces of this particular door need to be textured as the top and sides won't actually be visible; therefore I'll leave them caulked. The underside of the door will become visible when the door is opened,

however, so my first step in texturing the door will be the bottom surface. To do this I move the brush upwards, texture the bottom surface and then move it back into place.

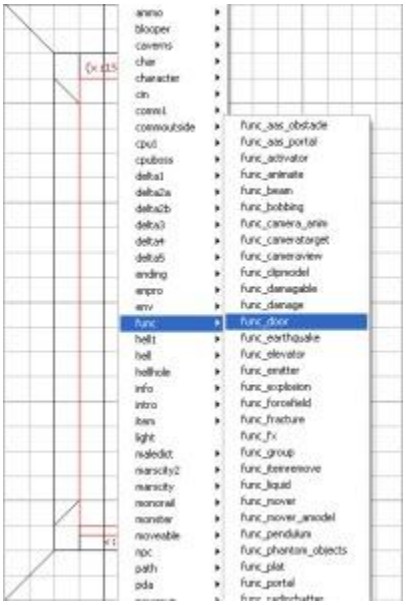


Now to texture both sides of the door itself. Fortunately, almost all of the most suitable door textures can be found in the base_door folder. Depending on the size of the door you are making, this can be either simple or complex. If the door is relatively small, you would typically be able to find a texture that should fit the door neatly, without tiling the texture or splitting the door into separate brushes. This particular door is relatively large, so I'm going to split the door into two separate brushes (top and bottom halves) and texture each brush surface separately. Once I've done this (using the basic techniques described in the previous chapter), previewed it using the real-time render mode feature and am satisfied with how it looks, it's then time to move onto the important bit - making the door open and close.

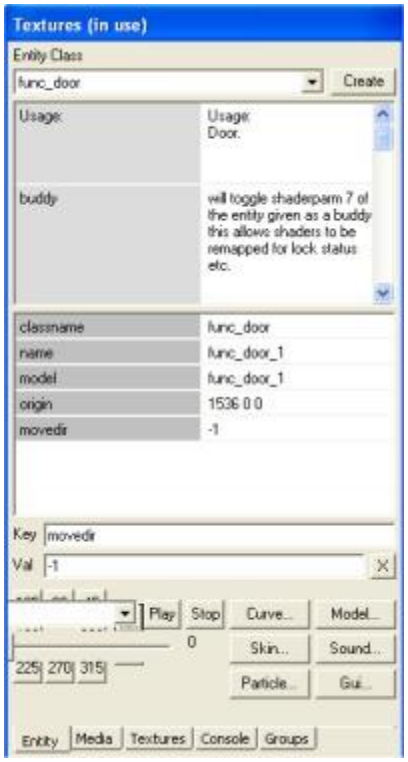
There are quite a few ways in which doors can be activated, and how they actually open up. For example, doors can be activated by switches (which we'll describe in more detail further down the track) or they can open automatically when a player moves close to them. They can also be opened when shot by a weapon. Which type of opening and closing features you want will be dictated by the kind of effect you want to achieve. Typically, automatically opening doors usually (but not always) work best in multiplayer scenarios, because in most situations players prefer to be concentrating on combat instead of fiddling around with switches and buttons. For that reason, this door will open automatically.

Doors can also open upwards, downwards, sideways or consist of multiple parts that open in various directions. I'm going to keep this door simple, moving in an upward direction, and the following is a description on how to make it happen.

First of all, select the brush/s that is to be the moving part/s of your door. Go to the grid window using the top viewing perspective, right click and then select func and then func_door.



Once you've done this, press the 'N' key, which should bring up the entity class window. There are quite a few options available from the entities class window, and it is here that you will specify which direction and method that your door will open. Because I want this door to move upwards, I click on the 'up' tab, which will create a key 'movedir' and value of '-1'.

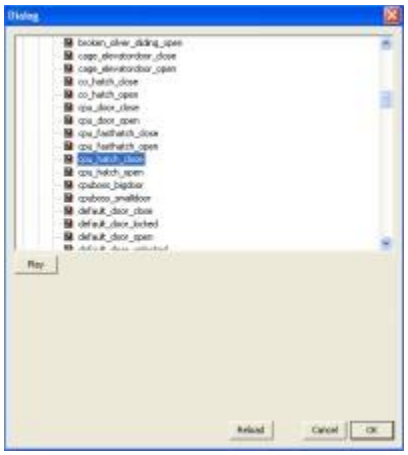


There are a few other settings I need to specify for this door to work in the way I want. One is to specify how much of the door will extend out from the ceiling when the door is in its full open position. The frame of my door extends by 32 units from the ceiling, and I want the full open position to be placed just underneath the frame (by 16 units). To create this effect, all I need to do is create a key called 'lip' in the key field and a value of 16 units.

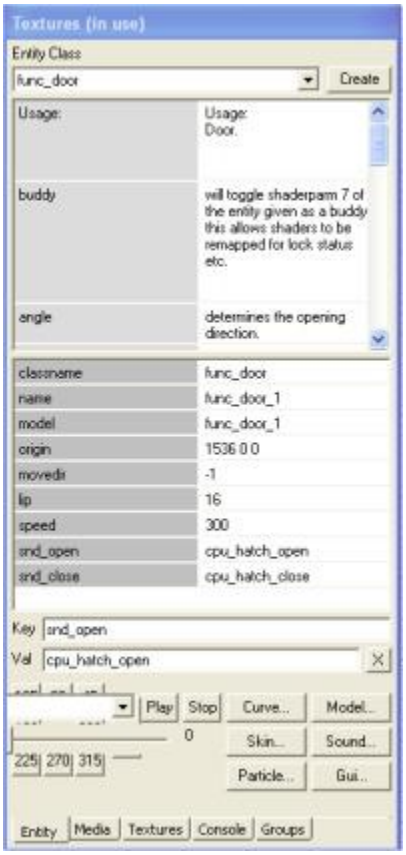
Next I want to specify how fast the door moves upwards. To do this I type 'speed' in the key field, and then the speed value in the 'value' field. The correct speed to use will differ, depending largely on how far the door has to move to reach its open position. Because this is a large door, I'm going to go with a speed of 300. I may tweak this speed value a little later after testing it in game, but for now I'm using an educated guess that 300 will work nicely.

One more thing to do before considering the job done - to give the door a sound. At the bottom right corner of the entity class window is a tab labelled 'sound'. By clicking on this tab a dialog box will open with two directories. One is labelled Sound Shaders and another called Wave files. I've found the greatest selection of sound files for doors are found in the Sound Shaders directory, under sound/doors.

Door sounds typically have an open and close sound and although not absolutely required it's a good idea to use the open sound for the opening sequence and the closing sound for a closing sequence. I've chosen to use the sounds 'cpu_hatch_open' and 'cpu_hatch_close' for the open and close sounds for this door. You can test the listed sounds by clicking on the 'play' tab and select them for use by clicking on the 'OK' tab.



Note than by clicking OK, they will then be displayed in the entity class 'key' field as snd_shader. To specify them to be used for their intended open and closing movements. To specify a particular sound for the opening movement, type 'snd_open' in the key field when the open sound is shown in the 'value' field. Use 'snd_close' to specify a closing sound. When all is said and done, the entity class window should look much like this:

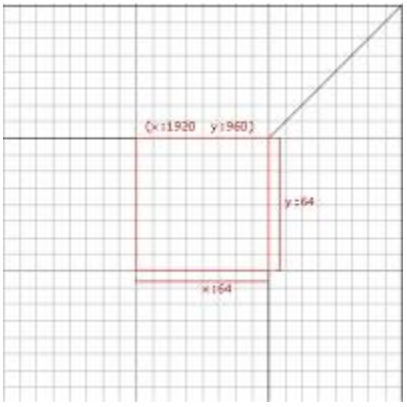


Now my door should work in much the way I want it to. There are other options worth investigating and experimenting with, and these are all described in the top field of the entity class window. You are well advised to spend some time having a look through the various options available.

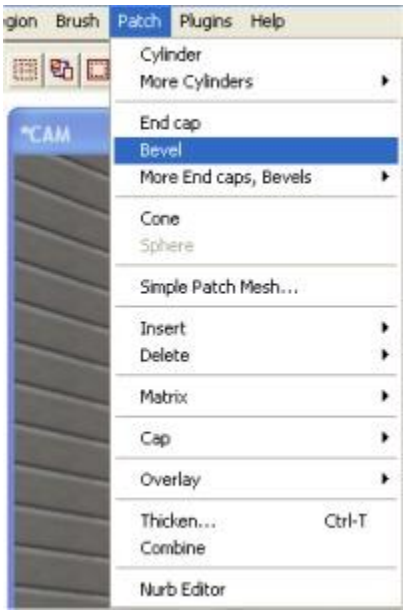
11 BASIC CURVED SURFACES

Curved surfaces, often referred to in level design as patch meshes, add a nice touch to the look and feel of a map and this chapter discusses some of the basics that go into their use and application. Before I continue with any demonstrations, however, I want to give you a little word of warning concerning using curved surfaces as vis blocking structure or to form part of the hull of your map. Don't. Not only do curved surfaces not block engine visibility, they also cannot be used as part of the outer hull of your map, otherwise it will leak and won't load. If you want engine visibility to be blocked, you will need to put a vis blocking brush behind it. I will provide a demonstration of it in this chapter and cover it in further detail in the optimization section.

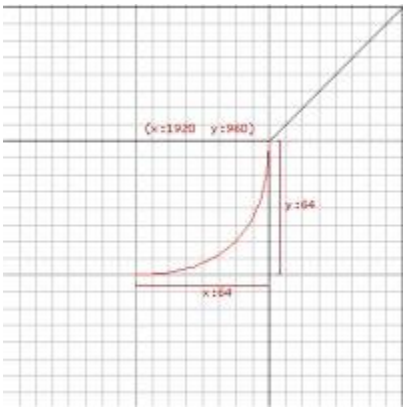
The first demonstration I'm going to give will show how to create some curved trimmings to make basic corridors look a lot less boxy. I'm going to do this by using a bevel curve. To begin lets first create a brush 64 x 64 units wide in the grid window using the top viewing perspective.



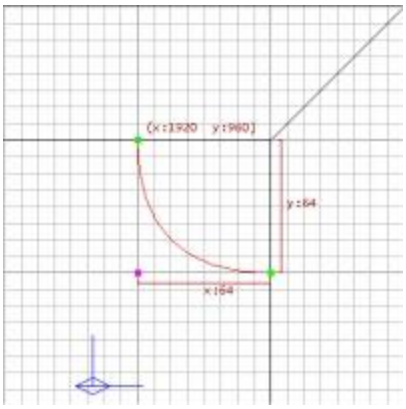
Then go to the 'Patch' tab on the top menu bar and then select 'Bevel'.



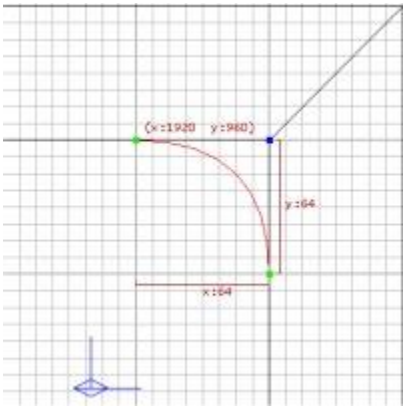
This will magically transform the brush into a bevel, like so:



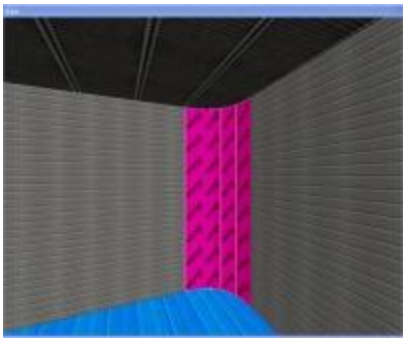
Unfortunately the level editor isn't quite intuitive enough to know exactly what angle I want the bevel to be facing, so to position it where I want to I have to rotate the bevel once on the Z axis. The next order of business is to morph the bevel into the shape I want. Currently the curved surface is facing outward and I actually want it facing inward. Do this by pressing 'V' to highlight the vertex points of the bevel.



Then click on the lower left vertex point so that it turns from pink to blue. Once blue you can now move the vertex point to manipulate the shape of the bevel. In this case we're going to drag it into the opposite top right corner, which will morph the bevel into the desired shape.

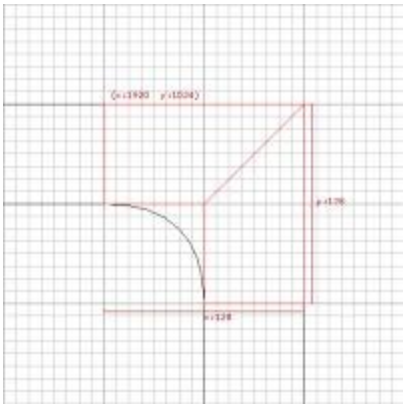


Next it would be a good idea to stretch the bevel into the desired height. This is a simple matter of switching the viewing perspective to the front or side options and then stretching it to the height you want. For now I'm going to stretch it so that it meets the ceiling and floor.



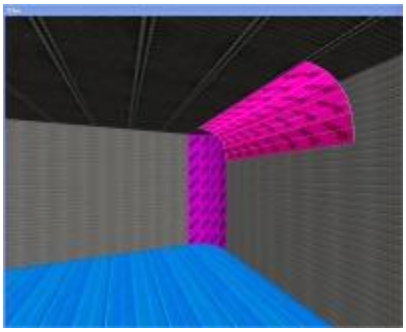
At this point I could just texture the bevel and I would have transformed an otherwise boring boxy room or corridor edge into a smoother looking curved one. We're going to go a little further than that, however, but before I go on I want to share a little optimization technique with you.

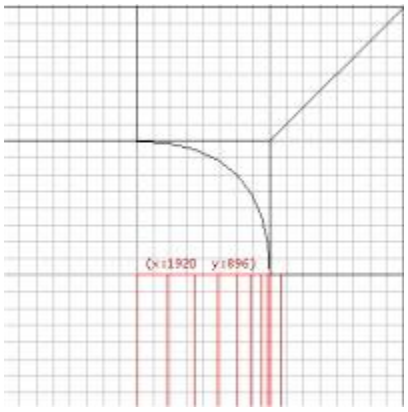
What we're going to do is eliminate any unseen textures from behind the bevel. I assume that by now you are becoming pretty handy with the clipping tool and if my assumption is correct, the following diagram will clearly show what I'm doing.



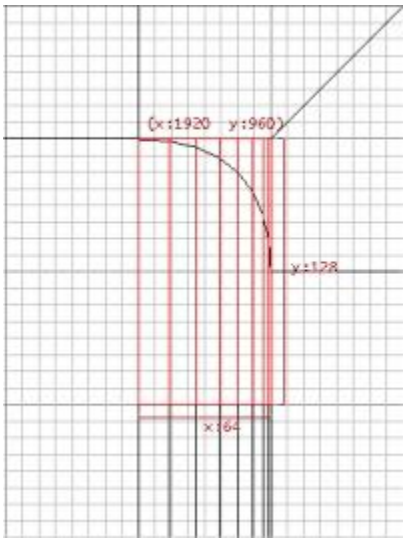
Once the brushes behind the curve have been segmented from the other wall brushes they should be fully textured in the caulk texture. What we've just done is not absolutely necessary but it is good practice and you are well advised to get into the habit of it.

Now let's further enhance what we've started, by adding a curve trim to run along the edge of the ceiling, which will later join up nicely on the corner trim we've just created. Let's begin by being lazy and duplicating (press 'space') the bevel we just made. Rotate it on the X axis 3 times and then position it so that the top corner meets the bottom of the ceiling.

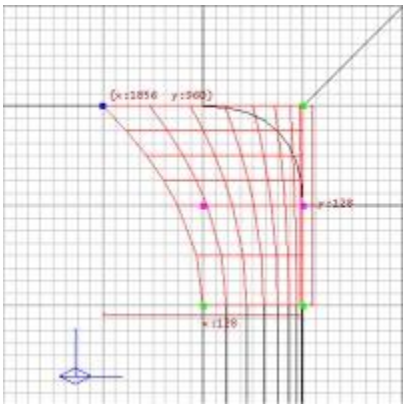


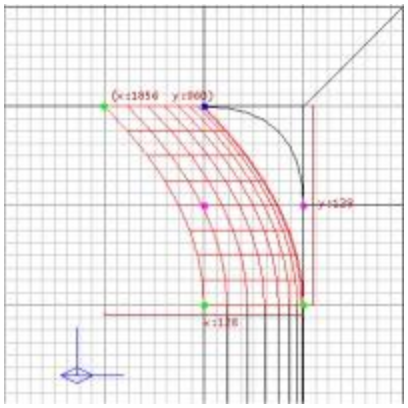


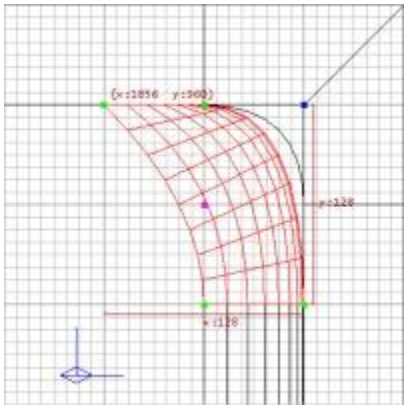
Now things are going to get a little tricky to perform and even harder to describe, but the end result is worth it, so bear with me. Reduce the height of the corner bevel by 64 units. Then Duplicate the bevel you just made and position it across the top of the corner bevel.

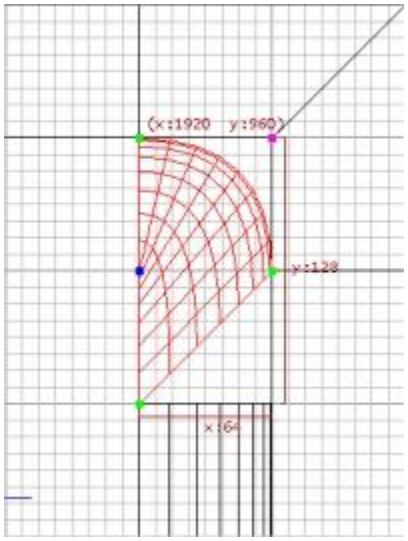


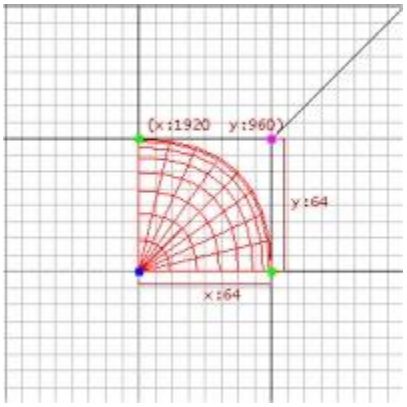
Now follow the diagrams below, which show each step that I take to create a joining trim that will be used to connect 3 other bevel brushes. The blue dots shown in each image are the ones you need to pay most attention to:



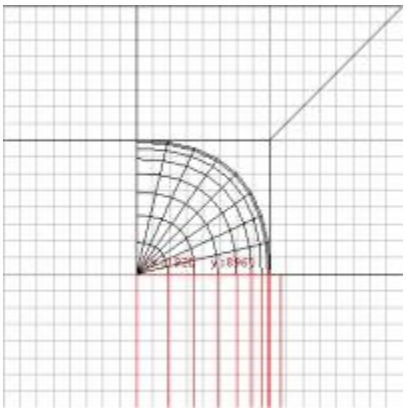




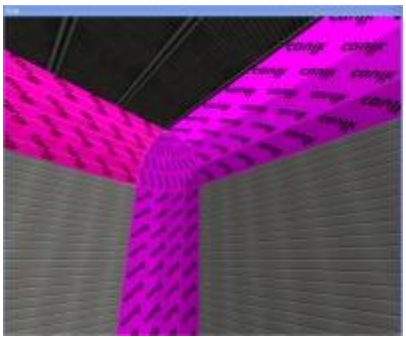

(x:1920 y:960)
y:128
x:64



Still with me? Good. Now stretch the edge of the ceiling bevel so that it joins with the connecting trim we just created, like so:



Now duplicate that ceiling bevel and then rotate it on the Z axis three times. We're going to use this particular bevel to make up the ceiling trim on the other side of the corner, which when all is said and done, should look much like this:



The next step is to texture the bevels we have created, and the bad news is that this is just as challenging, if not more so, that creating the bevels themselves. Once you start trying you'll see what I mean. Having said that, texturing curved surfaces is often relatively simple - but we've created a not so simple patch mesh, which means we need to employ a high degree of creativity to texture the surfaces we've made in such a way that looks natural. This all comes down to experimentation in finding a texture (or a combination of textures) that work in conjunction. Tiling or fitting a texture upon patch meshes that have been so heavily manipulated (such as the corner fitting piece we just created) is easier than it sounds. The next chapter will go into further detail on this aspect.

The trickiest part lies within finding a texture that works with both the connecting trim texture and those that connect to it. After spending some time experimenting with various textures and previewing them with the real-time preview feature in the cam window, I've found that the base_wall/lfwall3a texture works nicely with the same texture used for the corner trim below it as well as the base_wall/lfwall3 texture, which I've used for the ceiling trims that connect to it. Further experimentation may influence me to change this later on, but for now it seems to work pretty well. I've placed a place holder light entity nearby to use as a means to get a rough idea of how it will look in game, and you can see the results of this in the image below:

We'll discuss some more advanced uses of curved surfaces and methods of texturing them in further detail in the next chapter within the guide.

12 LEVEL OPTIMIZATION

You may be thinking to yourself that you should just build your map and then optimize it best you can towards the end of the development process. To a degree, this isn't such a bad idea, but for most level optimization practices for the DOOM 3 engine, it really does pay to think ahead and optimize as you go.

Previously we mentioned that every brush should be made with the caulk texture, and only visible surfaces should be applied with the actual texture you want drawn. The same applies for Quake 3 editing (and a similar technique is also used when mapping for the Source engine).

We have also made mention that the DOOM 3 engine likes levels to be compartmentalized, and that the layout of your map should facilitate this with doors and right angled corridors and hallways connecting specific areas. This bit is really important, because it will allow you to use visportals to help optimize your map, and these can help a lot. Let's discuss them in more detail.

Visportals

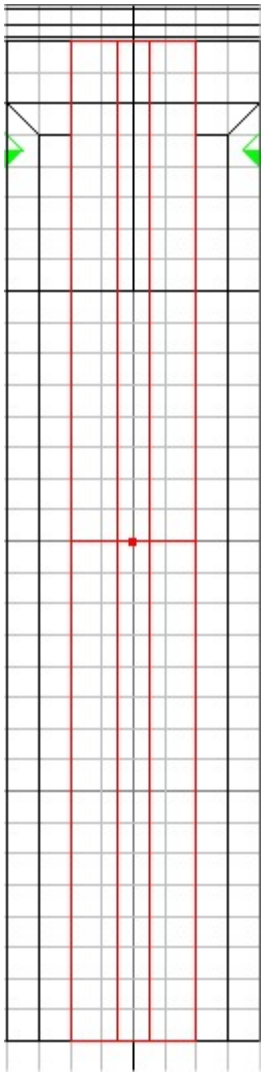
Visportals are invisible walls that block engine visibility in certain areas in certain situations. When used correctly, visportals create walls that the engine cannot see past from certain perspectives - i.e.: around corners and through doors. They won't work in situations where the player can see into an area from their current perspective but if used correctly they will work when the player otherwise cannot see into a particular area. Before giving you a couple of examples, let's first discuss how to create them.

To make a visportal all you need to do is create a brush and paint it with the common/no_draw texture. Deselect it, and then select one surface (the surface you want to block visibility). Texture that surface with the editor/visportal texture. Note: when you do this and then deselect the brush, it may disappear from view. If so, it is because visportals are flagged as hidden. If you want to view your visportal/s, go to view/show and place a tick on visportals, like so:

To function correctly, visportal brushes must neatly align with the top, bottom and sides of the surrounding structure. If there is any gap between the visportal brush and surrounding structure, the visportal won't work so take care to get this part right.

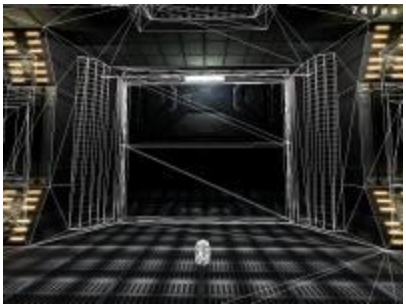
Using visportals inside of doors to block visibility to other parts of the map is relatively easy. Basically, all you need to do is create your visportal brush as described above and then place it inside the door brush. Providing that it is placed correctly, the door will now block visibility from one area to the next whenever the door is

closed. Below is an image of a highlighted visportal brush placed inside a highlighted door. Note how the top and bottom of the visportal brush meet the top and bottom surfaces perfectly. The same must apply on each side as well.



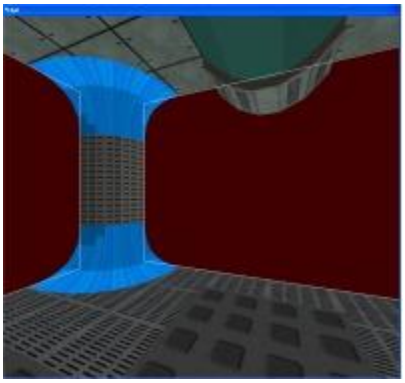
Below are two images that show the difference between a door without a visportal placed inside and one that has a correctly applied visportal. Using the 'r_showtris 3' console command in game, all viewable brushes will be highlighted. Here's what it looks like without a visportal.

Note how there is a tonne of brushes that the engine can see past the door. This is bad, because it is rendering brushes that it won't need to until the door is opened. Here is an image that shows what happens after you place the visportal inside the door:

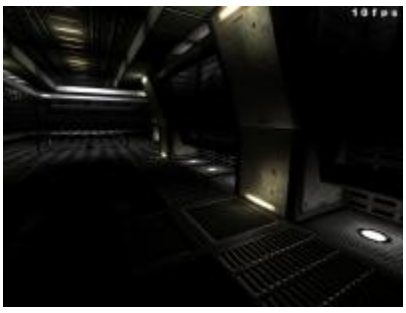


It's a big difference isn't it? Not only will it help improve frame rate speeds in game, it will also improve compile speed times - and that's one good reason to make them as you go.

Another, and perhaps more important example, is the correct use of visportals outside of doors. These can be used liberally and are best positioned at specific areas. They are particularly effective for sealing off right angled parts of the map. Below is an image that shows 2 highlighted visportal brushes. Each of them seals off the ends of a right angle in the map.

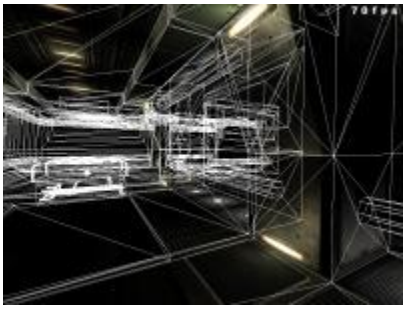


It's important to realise that if the player was standing in that position, the visportals won't block anything that is viewable. But if the player is positioned where a certain area cannot actually be seen, visportals will prevent the engine from trying to draw the unseen area. It sounds confusing, but the more you use them the more you will learn how effective they can be. The following images demonstrate just how much difference visportals can make. The first image simply shows the scene with which the example is given.



Now we go to the console and type 'r_showtris 3' to show us which brushes the engine is really drawing.

Wow. Notice how practically the entire map can be seen by the engine from that perspective, even though it is hidden from view to the player? It's a total waste of rendering resources, which is why visportals can be so effective. The image below shows the same scene with visportals correctly placed.



The images speak for themselves. Note the frame rate speeds on each screenshot (top right) to get an idea of practical performance difference. Visportals can make a massive difference, so it pays to get familiar with them. Remember that they can be used both horizontally and vertically. Don't be afraid to experiment with them and don't be shy in using them wherever you suspect they might make a difference. To check if any visportals you place are doing their job properly, use the 'r_showtris 3' console command.

In the event that you notice some rendering anomalies after placing visportals, it may well be because one or more of them aren't positioned properly or don't like where they have been placed.

Lighting

The per-pixel lighting technology embraced by the DOOM 3 engine is a particularly demanding method for illumination, and for this reason level designers have to be very careful with how they are used. We'll discuss the use of lighting in more detail in the next chapter, though we will first discuss some basics in optimizing them here.

The most important things to remember is that you want to avoid the radius of light entities overlapping as much as possible, and that you don't want too many light sources illuminating the same brush. To these ends there are several things to keep in mind. First of all, never make light entities cast any more light than you need. Secondly, be wary of creating large brushes that will be affected by numerous lights.

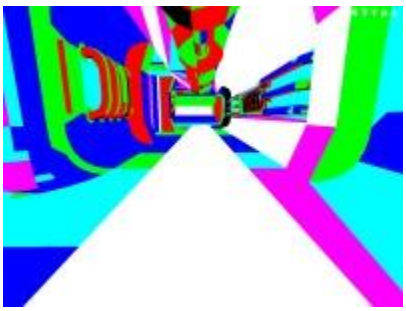
Long corridors, for example, can easily be made with long brushes making up the floors, ceilings and walls. With Quake engine editing, wherever it was possible to use 1 brush instead of many, it was good practice. The same is true of DOOM 3, but only in circumstances where there are few light sources affecting the brushes. That is rarely the case in the given example of a long corridor.

If you are planning on using a lot of light entities in a particular scene you would be well advised to construct the scene in segments, instead of using large brushes. On the other hand, you could always make a scene with larger brushes and then split them into segments using the clipping function.

Before providing some examples, I'd like to introduce another console function that can help you determine how light sources are affecting specific areas of your map. By typing 'r_showlightcount 1' into the console, all brushes will show up in brightly lit colours, being black, red, green, dark blue, light blue, pink and white. If a brush surface shows up as black, it means that there is no light source affecting it. Red means that only one light source is touching, and this is ideal (though not always possible). Green represents 2 touching sources, and this is more than acceptable. Dark blue means 3 light sources are affecting a surface and this is still acceptable in most cases. Light blue surfaces are also acceptable, but only if there are few of them (and there aren't any or many pink or white surfaces showing up. Pink and white show surfaces that are being affected by too many light sources. Where possible you need to minimize how many pink and white surfaces show up in any particular scene.

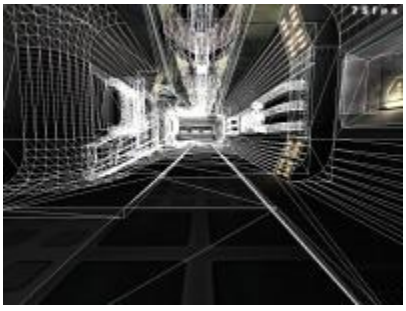
So here's an example that demonstrates how splitting brushes can really help. This first image shows a long corridor with a floor brush that spans the full length.

Now here's an example of the same scene with 'r_showlightsource 1' turned on.

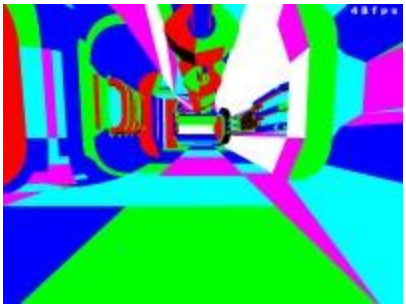


Notice how the floor brush is highlighted in white - this is bad, and it's the result of using one big long brush instead of segmenting it into numerous brushes.

So after I go forth and slice up the single brush into multiple brushes, and using the 'r_showtris 3' command, here's what it looks like now:



And here's how the same scene looks using the 'r_showlightcount 1' command.

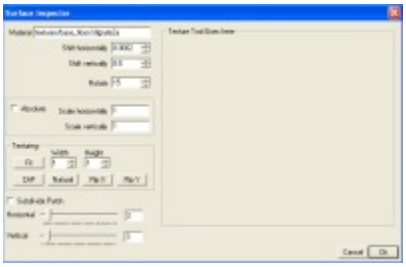


Note how the single white brush has been replaced with more acceptable colours. This will result in smoother frame rates.

If you're really observant, you would have noticed that I used a different texture for each segment. I did this for two reasons. The first (and least important) reason is that the two textures I used work well together, and help to create a more detailed appearance. The second and much more relevant reason is that if the same texture is used with precisely the same alignment on segmented brushes, the engine will connect them and treat them as one single brush in game.

Using alternate textures isn't the only solution to this dilemma. If you want to use the same texture across multiple brushes, but want to avoid having the engine treat multiple brushes as one, there is a little trick that you can use, which is to offset the texture alignment on every second brush ever so slightly. So long as the textures aren't perfectly aligned, the engine won't assume that multiple aligned brushes can be treated as one.

To do this, highlight the surfaces that you want to offset and then press 'S' to bring up the surface inspector. In one of the shift fields, change the .5 value to .0002.



Then shift the highlighted surface textures a click or two. The texture will shift ever so slightly across the surface and now the engine will treat those separate brushes as separate brushes. The textures will still appear perfectly aligned and you'd have to look very closely to detect any difference, providing that you shift the texture with such a small increment.

We'll discuss more facets of DOOM 3 lighting in the next chapter. Until then, remember everything I've tried to explain in this chapter, because it will make a huge difference to how well your map runs.

13 LIGHTING

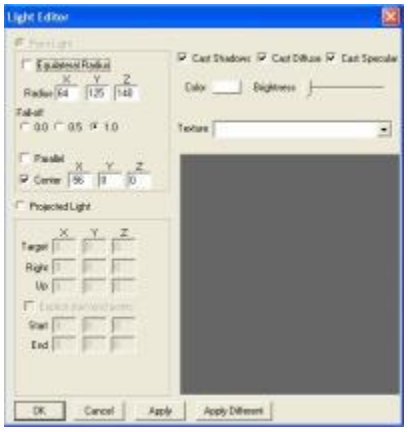
One of the most impressive attributes of the DOOM 3 engine is per-pixel lighting. Lightmaps are no longer used to illuminate the game world environment or the objects and characters within it (and so are the ridiculously long compiling times along with it). These have been replaced with a unified per-pixel lighting system that lights all surfaces in exactly the same way from the same light sources, which offers a more convincing and consistent appearance.

There are, however, a few downsides to the per-pixel lighting technology used in the DOOM 3 engine. Most obvious is the fact that per-pixel lighting is particularly demanding on system resources and only relatively capable PCs are capable of running it smoothly. Because of this level designers have to be particularly thoughtful about how they light specific areas - especially in multiplayer maps.

The other downside is that they work quite differently to the various lighting methods that could be applied to lightmaps, which means a whole new learning curve for level designers who are new to DOOM 3. Without question, the most challenging aspect of creating DOOM 3 levels is lighting them sufficiently and efficiently. Be prepared to experiment and whenever that fails, be prepared to experiment more.

Having said that, the flipside is that the new methods of lighting in DOOM 3 are rather interesting and numerous effects can be achieved with a bit of patience and creativity. If patience isn't one of your better virtues, keep reading to get some handy tips.

As mentioned earlier, to place a light source within your map it's a simple matter of placing the mouse cursor over the 2D grid window, right clicking and selecting light. The next step is to position it where you want it. In the rarest of circumstances, it really is that easy. More often than not, however, you'll want to get a bit more creative. To do that, press the 'J' key (whilst the light entity is selected) to bring up the Light Editor window.



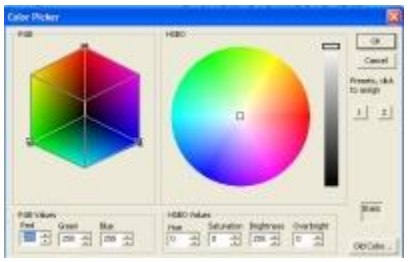
You will note that there are two kinds of light sources: Point lights and projected lights. The former work similarly to those of the Quake engines (with some notable differences) and the latter is quite new and worth getting familiar with. Let's talk about point lights first, however.

Point Lights

There are several options available to you when editing point lights. The first and most obvious is the radius. By default this will be set at 300, which is typically brighter than you'll want it to be. A radius of 300 will be displayed in the X radius field. Alter this to whatever radius you think is suitable and then deselect the light. Select it again if you want to manipulate any of the X, Y or Z axis to get the spread of illumination you are trying to achieve.

Alternatively, there is another way to create the radius of light and in some ways it's easier than setting values in the X, Y and Z axis fields. To do this, create a brush where you want the light to take effect and whilst it is selected, press the right mouse button and select light. This will automatically create a light entity that emits light to the exact dimensions of the brush you just created.

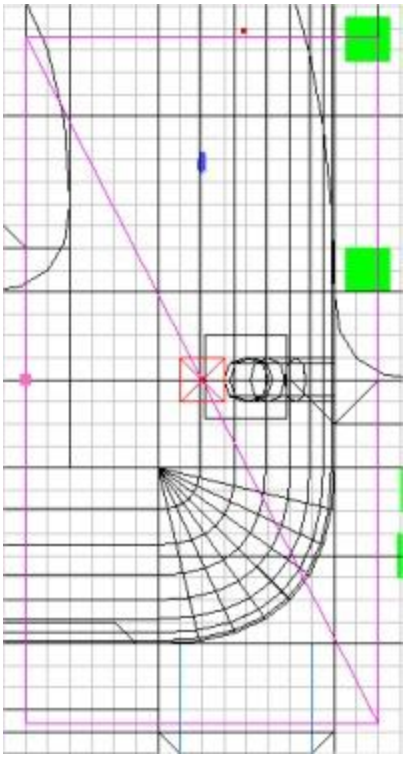
Now that your light entity has been created there are a few more things you can do to change its appearance. Light entities will emit white light by default, although this can be modified by clicking on the Colour tab, and then selecting a shade of colour that suits. Ignore the brightness tab next to the colour tab because it doesn't actually do anything. You can adjust light brightness by using the saturation slider in the colour picker window, however.



Above the colour and brightness tabs are 3 options worth noting. They are 'cast shadows', 'cast diffuse' and 'cast specular'. Unticking the diffuse and specular boxes will alter the way the light interacts with textures, and this can help to increase frame rates in some situations. I haven't yet found these to make a big enough difference to bother with, however, and prefer the appearance of diffuse and specular features turned on. There are various instances where I have been compelled to turn off the 'cast shadows' option. Sometimes to save on processing calculations to increase frame rates and other times to prevent some lights from casting shadows, either because of the way they effect the surrounding environment, or how they cause the player to cast shadows in multiplayer games.

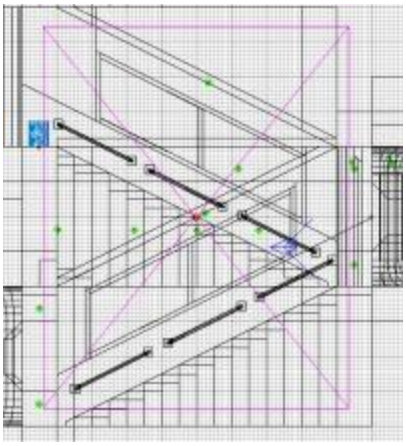
The Fall-off options don't actually do anything from the Light editor either, so don't bother messing around with them. The centre option is worth paying attention to though, so let's discuss this for a moment. There will be times where you will want to simulate light emitting from a specific texture. Sometimes a problem can arise where the centre origin point of the light will show up too clearly, which looks a bit weird. Here's an example:

An effective way to address this problem is to move the centre origin point of the light. To do this, place a tick into the centre tab in the light editor window. You will notice a small pink dot appear near your selected light entity. Now deselect the light and then select it again so you can adjust its position. Click on the pink dot and drag it around from the centre to achieve a more realistic appearance. In this case I am moving the origin point away from the texture in question.

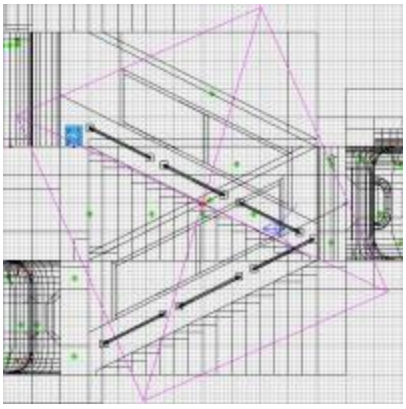


And here's the effect it has, which certainly looks a little more realistic:

Before I move on to talk about projected lights, I have one more tip on point lights, which is to mention that light entities can be rotated on any of the 3 axis, allowing you to create an angled lighting effect. I've found this useful for instances where the surface that I want to simulate light from is angled, as opposed to straight up vertical or horizontal. In the following example I have a default point light positioned under an angled light source, which causes an odd looking illumination.

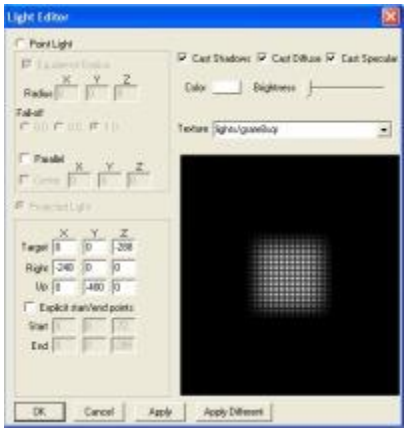


But if I press the 'R' key to rotate the light entity, I can more closely match the angle of the light source resulting in a more realistic looking effect.



Projected Lights

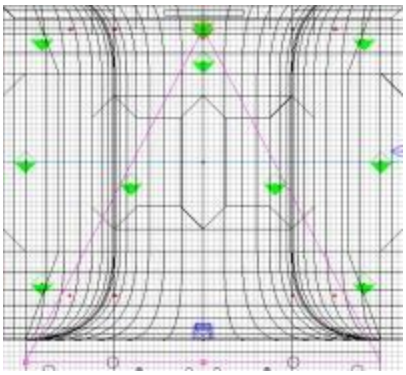
Projected lights project a light texture from an origin point as opposed to emitting a radius of light as point lights do. These can add some very cool effects to your level and are well worth experimenting with. To create a projected light, first create a basic light entity and position it where you want the origin point of the projected light to be. Then press 'J' to bring up the light editor window and place a tick in the projected light tab.



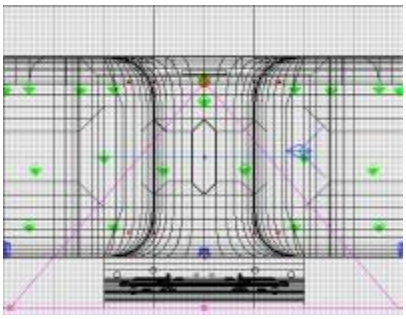
Bingo - you have just created a projected light. But there are still a few things left to do with it before moving on. First you'll want to select a texture to project. Do this by clicking on the down arrow in the texture tab.

Here you can cycle through all of the available textures for projected lights. When you find a texture you're satisfied with, deselect the light.

It's likely that you will want to edit how far or wide the texture is projected. Re-select your newly created projected light. You will note that there are 2 pink dots connected to the projection indicator.



Left click on these pink dots to adjust how far and wide the texture is projected. In the example below, you can see that I've lowered how far the light is projected, as well as the width.



And here's the end result in game:

Adjustments can be made in each of the axis viewing perspectives, although be warned that it can get a little fiddly to get the right angle, so be patient if you are trying to project a texture at an odd angle.

Don't forget to read up in our Level optimization section for some tips on making sure your application of lighting doesn't slow down frame rates too much.