

# Mapping for Tremulous

## Introduction

Many people have asked about mapping for tremulous. They especially seem to be having trouble with the install and their first couple of maps. Since the troubleshooting of the install is diverse I'll leave that up to the mapping wiki (link to mapping wiki <http://tremmapping.pbwiki.com/>) and the forums (direct link to mapping <http://www.tremulous.net/phpBB2/viewforum.php?f=5> and troubleshooting forum <http://www.tremulous.net/phpBB2/viewforum.php?f=4>). What I hope we'll be doing in this thread is creating a step by step tutorial of the actual mapping. I'll be doing this the way I'm doing FCP. There are other ways, but well, this is mine. Remember that tremulous is capable of much more than what is shown here but you are beginners so everything is down to its basics except for the gameplay mechanics guide which should be understood from the very beginning if you want to make fun and challenging maps.

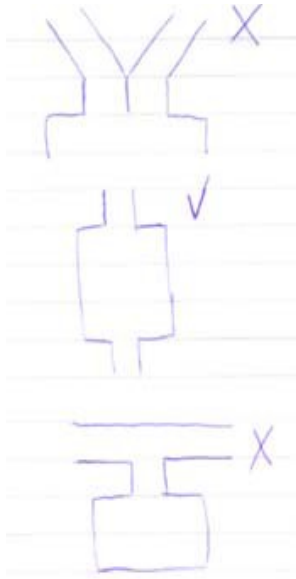
Mapping is about making maps; 3D levels which players can walk through and play in. The play in part is the most important and something most people forget about when they start mapping. It's pretty easy to create a room which you can walk in or a level you can explore. But making something that is actually playable takes a little preparation. You won't find this in any mapping tutorials on the internet because Tremulous' gameplay mechanics are uncommon.

## Gameplay mechanics

What I usually do is make a general gameflow chart, this is applicable for any multiplayer game but for each game the requirements differ since the style of play differs.

## Entrances

The most important thing to keep in mind with Tremulous is bases. Especially a starting base should always have at least 2 entrances. If it doesn't have 2 entrances it will very fast become easily defendable. You might want to make such situations in small maps to lengthen gametime but not in the larger maps. If a room has 2 entrances it does not mean that it automatically means the engagement are has two entrances. Examples follow.



Also important in gameflow is the distance between the two entrances of engagement areas without going through the engagement area itself, a short distance means 2 things, a team can easily hold in the other, but if the other breaks out the attackers of the other entrance suddenly have 2 fronts. With long distances this is not the case but someone is more likely to break through the blockade and be able to do damage to the attacker's base.

## Team advantages

Other things to keep in mind are advantageous situations towards a certain team. I will describe several situations and which team they give the advantage. You can choose to avoid these but by preference include several of them to give a team a tactical advantage in certain locations. Especially the conditions that change at stages are very interesting to incorporate into maps. Too many however make the map overcomplicated and unliked. They will be mostly called gimmicky.

Long, straight, very wide corridors and long, straight very narrow corridors both give the advantage to the humans with their long range line of sight weapons.

Winding corridors give the advantage to aliens in human stage 1 but they're drawn together with the advent of the helmet at human stage 2.

Height difference is a definitive advantage to all aliens except the tyrant and normal granger. Any other alien is far more mobile on the z-axis than any human could ever hope to be. Good examples of this are high, dark ceilings, stacked crates and drops like the one on arachnid. These drops are very nice tactical situations. Most aliens can flee up them and humans might be able to trick tyrants into falling down them. Small drops are also of tactical importance when it comes to human and alien basebuilding since most equipment-variations and aliens will have a hard time going up a defended drop. Think of the elevator room in Karith.

Another well known and loved situation are the vents, providing easy access and retreat for the smaller aliens while providing alternate routes to any human not wearing a battlesuit they also limit mobility of anyone entering. When you keep in mind that you can select which aliens you want to be able to enter the vent by the way of their hitboxsize in correlation to the vent size these are very nice additions to maps when not overused. Remember that a long vent which only allows aliens smaller than the advanced basilisk will get massacred by any long range weapon the humans have. Try to twist them or make them have several access points to

alleviate this problem. There is one other thing you have to remember with vents but it's mentioned in the map optimization chapter.

The marauder's walljump capability might also be used to limit anything dragoon and up from crossing a gap. Wall climbing aliens can climb along the walls, the marauder can walljump from the left to the right wall continuously so if you make the gap not too wide and longer than the advanced dragoons maximum pounce distance you have effectively created a dragoon block. Since the advanced dragoon pounces farther than the normal dragoon you could also open up a path for heavy aliens at alien stage 3 where the dragoon can't get past put the advanced dragoon can by keeping the gap shorter than the adv goon's max pounce but longer than the normal goon's max pounce.

The last important part is about map size. Humans by default need to reload their weapons and health in base. Aliens don't, they just need time. So small maps favour humans since they can return to base more easily and aliens don't get a real chance to flee and heal. Large maps put the favour at odds since it still allows for good humans to hunt aliens down. But ultimately I'd say large maps favour aliens.

## Player-defined Advantages

As you have noticed most of these situations have to do with the limitations of the ability of certain classes. The previously mentioned ones all were exclusive. You can predict who will have the advantage. The other category is player-defined advantages which do not depend on the team. These situations depend on who controls a certain element. This might be such situations as having control of a certain door lock like in pulse, being able to deny the other team of accessing a certain path like controlling the centre in uncreation (which limits the other team to having to pass through a certain guarded intersection), or having a situation which is exceptionally suited for a certain class or weapon like a Mass Driver in a long straight vent. When used correctly these add to the available tactics.

Someone might be able to deny aliens the second access point to an engagement area with the Mass Driver in the vent which is the second access but if this is far from base the Mass Driver will have to reload and relinquish his hold to the aliens. This works best if two opposite situations are right next to each other. This could be a marauder friendly environment at the human exit of the vent. As such it will not be easy for the Mass Driver to regain control from the marauder nor will it be easy for the marauder to enter his preferred area. This also invokes base moves since if the Mass Driver is able to lock it off while not having to relinquish control to the aliens because his base is in the marauder friendly area for reloading you might be able to design a situation where a single human can defend the second access (vent) while the other is covered by the bases own defences. You should keep all of this in mind when you're creating maps which are fun and challenging.

Also nice tactical spots are the good base optimized spots which encourage moving base. A good but grossly overpowered example is the elevator room in Karith where the first team to relocate to it gains an advantage in defence.

## Gameflow analysis

We'll take a look at several simple maps that are already out and I hope you are already familiar with them.

First up, ATCS:

ATCS is a straightforward map which I'll use to explain the basics of preparation.

ATCS



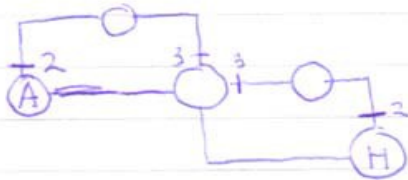
You can see how I drew the engagement areas as circles and the paths leading between as simple lines. This is the most basic way of drawing a gameflow chart. It consist solely of important areas and the paths between them. Sometimes you will want to mark certain areas with either letters or numbers for clarity purposes. For now a circle with an A as the alien starting location and a circle with an H for the human starting location is the closest I can show you.

ATCS also exhibits something very important, the 2 entrance to base aspect which can be seen by the two paths leading to the bases.

Second up, Uncreation:

Many people find Uncreation unfair, and it actually is in the first two stages when we look at the gameflow chart.

UNC REATION



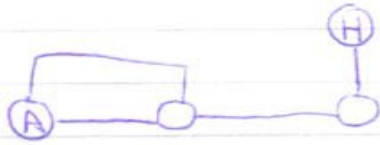
You can see the mapper used trigger\_stage (marked with their respective numbers), these are triggers that can activate certain things like doors, for creating a good gameflow in stage 3. Because of this Uncreation is a map which should make both bases easily defendable in the first 2 stages while opening up favorable offensive gameplay with 2 entrances to each base in stage 3.

Third up is an example using the two betas of Gloom2:

I'm using this one because it shows a simple development common in mapping.

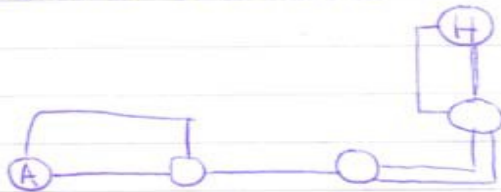
Gloom2Beta1

## GLOOM 2 BETA 1



### Gloom2Beta2

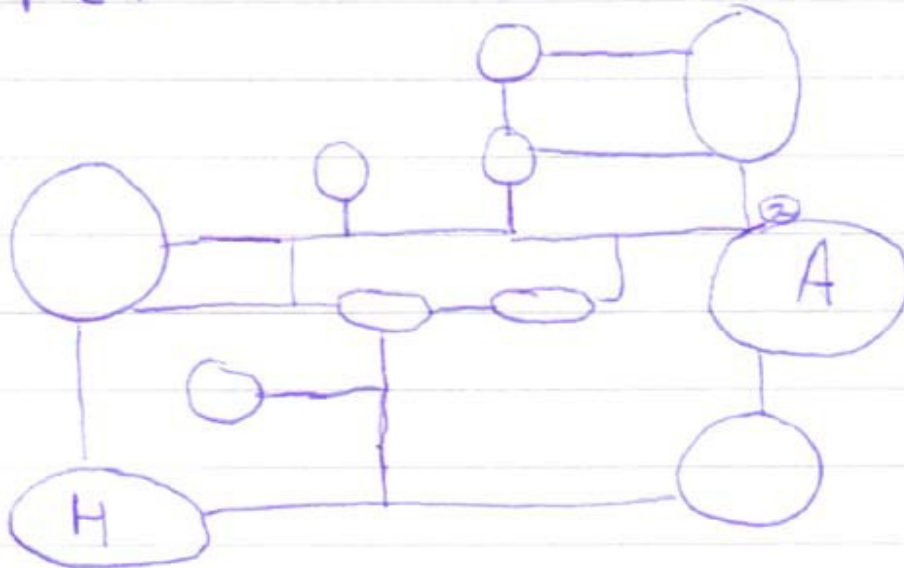
## GLOOM 2 BETA 2



You can see that Beta 1 had a continually easy to defend human base throughout the entire game while aliens had the more easily attackable 2 entrance system. The mapper solved this in Beta 2 by splitting one engagement area in two and adding a second entrance to the engagement area holding the human base.

Example of a more complex map, FCP.

## FCP

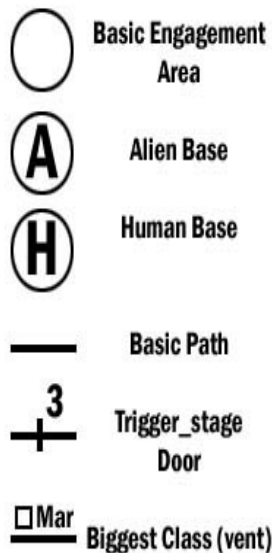


## Addendum

These were the very basics of gameflow preparation, don't be afraid to draw it big. You might want to add or change things and you need the room for that. And as mentioned there are

some things which you might not think about while playing the game but which can definitely add some strategic depth like the stage triggered doors, drops or doors through which certain aliens can or cannot fit.

What now follows is a short list of markings for use in making these maps. This is purely for consistency's sake. I hope all new mappers use them so people can help each other out and understand what someone is saying when they propose changes to a map. It isn't complete since most changes are only basic.



---

## Mapping Guide

Most of you will have skipped the first boring but essential part. Shame on you. But since I can't stop you, this is the part you've been waiting for. The Tremulous specific mapping tutorial.

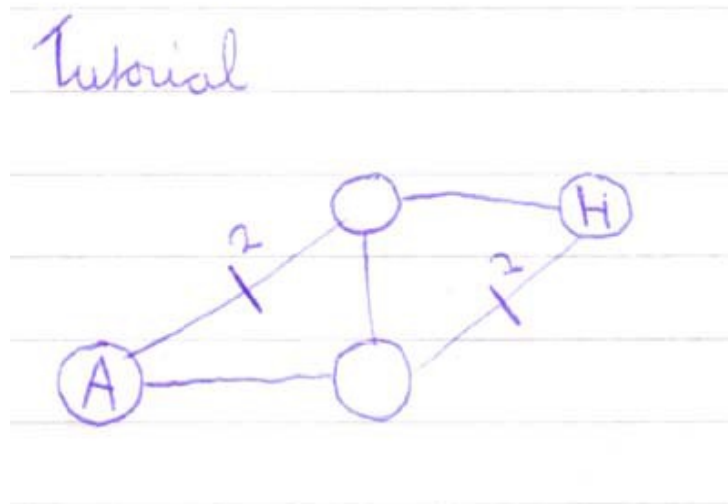
### Installation Directions

First get GTKradiant (direct link [http://www.qeradiant.com/?data=files&files\\_dir=1](http://www.qeradiant.com/?data=files&files_dir=1) and maybe store on tremulous.info in case it goes down?). I myself work with version 1.2.13 since it's an older but tried and tested version and has everything you need for tremulous, you can do it with newer versions but they don't offer any real gain for tremulous mapping as far as I know. Install it according to the guide on the mapping wiki(direct link <http://tremmapping.pbwiki.com/>). Some options might differ however from my version if you choose another but all essential ones have remained the same.

### Planning

You should start with thinking about the type of environment you'll be working in. Most common and easy is a simple human base. We'll be using that for the tutorial. Other things to think off at this stage are that when you want to add something special into it you might need new textures, models, sounds and effects which all require extra work, some of which you might not be able to do yourself.

Next up is the gameflow chart.



We see that we'll obviously have the 2 base locations and 2 interconnected basic engagement areas. Since we have so little engagement areas and paths this map isn't really meant for relocation of bases. I've also added 2 trigger\_stage entities into the gameplan which will trigger at stage 2 and prevent early pincer movements.

Now comes the real part, the mapping.

(next part if I decide to post it in forum, not for tremulous.info)

Note well, if you are having trouble with setting up GTKradiant this is not the topic to ask questions about it. Make a new thread (direct link to mapping forum

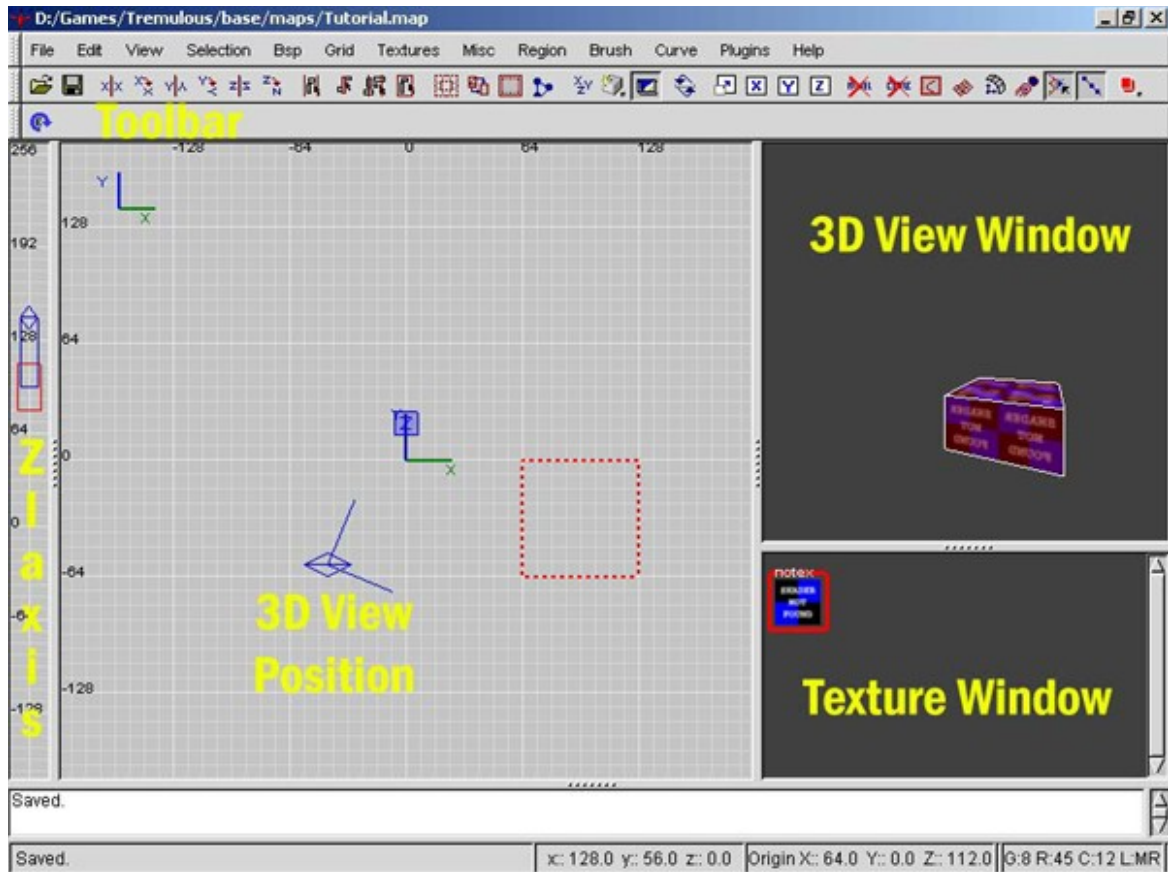
<http://www.tremulous.net/phpBB2/viewforum.php?f=5>) or check out the mapping wiki (Link to mapping wiki <http://tremmapping.pbwiki.com/>).

## Introduction to the User Interface

First up, a simple explanation of the screen you'll be doing most of your work in.

The Main Screen:

Some options I might name are in yellow.



You can see at the top left where you are saving your map to.

Mapping in 3D is done by manipulating a brush along 3 axis.

You can create a brush by clicking and dragging your mouse pointer in the gridded part of the screen. In the top left of this part of the screen you can see the two axis you are looking down upon. To the left is the Z-Axis which controls the height of the brush. You can switch the axis you look down on by pressing ctrl+tab but the Z-axis bar will always be on the left.

When you have created a brush you can either move it around by holding the left mouse button down on it and dragging it across the grid, delete it by pressing backspace or alter it's dimensions by holding the left mouse button down outside the selected brush and dragging it. The same applies to the Z-Axis. You can deselect a brush by pressing escape and reselect one by holding shift and clicking on it. The amount of movement for now will be controlled by the grid size which can be selected by either the numbers at the top of the keyboard, the menu or the numpad.

The 3D view window displays what is seen by the 3D view position.

It's standard controls are:

up = forward      down = back  
left = turn left    right = turn right  
A = Look up        D = Move up along the Z-axis  
Z = Look down      C = Move down along the Z-axis

You can also use free look by right clicking on the 3D View Window

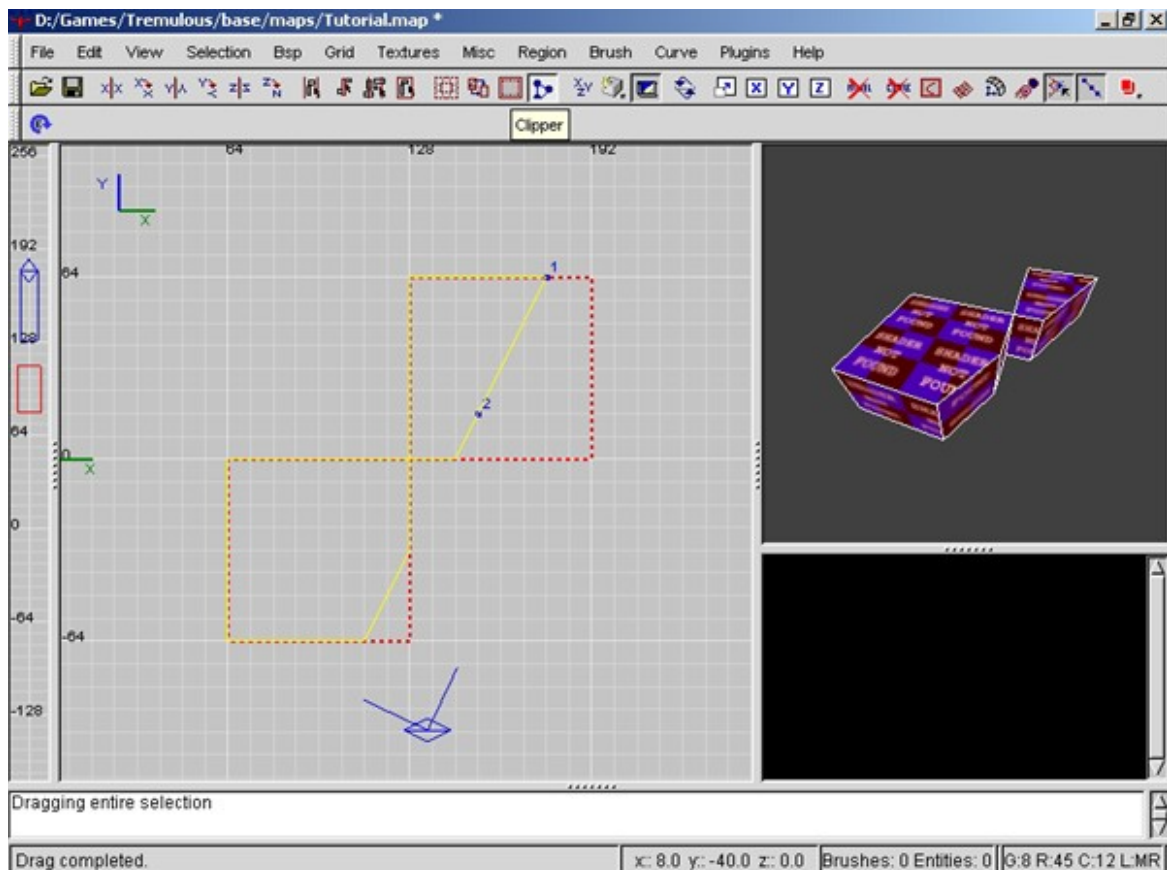
## 2-Point Clipping



Now we are getting closer to your first room. But there are still several ways to alter a brush. I will only explain the clipper now. Pressing either V (vertices) or E (edges) accesses two different ways of manipulating brushes and patches which you can explore yourself.

The clipper is an important tool so we're going to learn how to use it before we start with the room. Only 2 point clipping will be explained here.

The clipper tool is activated by either pressing X or selecting it in the toolbar. In the screenshot I'm hovering my mouse over it.

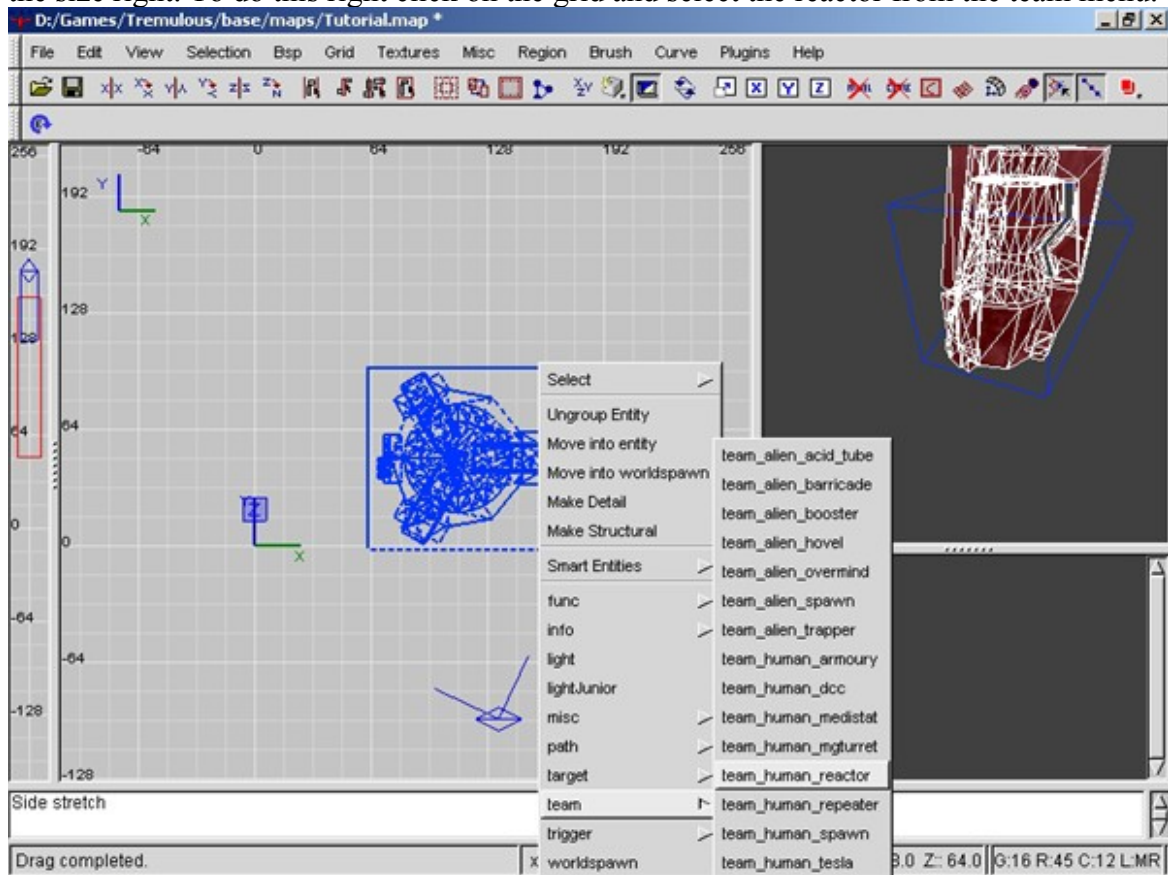


Select a brush or multiple brushes and activate the clipper. The first click will fix a point 1 on the grid and your second click a point 2. Through these two points goes an infinite plain along the axis you are looking down. The part that will remain when you press enter is yellow on the grid. In the 3D View Screen you can also see the result of your action. If you want to change the selection so the other side stays and the current yellow one gets deleted press Control+Enter to switch the selection. Pressing Shift+Enter will result in a cut with both sides of the brush remaining. Press enter to make the change permanent. You can do this many times to a brush.

## Your First Room

Now a simple room consists of four walls, a floor and a ceiling. An important thing to remember when building all rooms and hallways are that no entities should connect to the void. The void is the space not enclosed by brushes. First we will make a floor. You can first

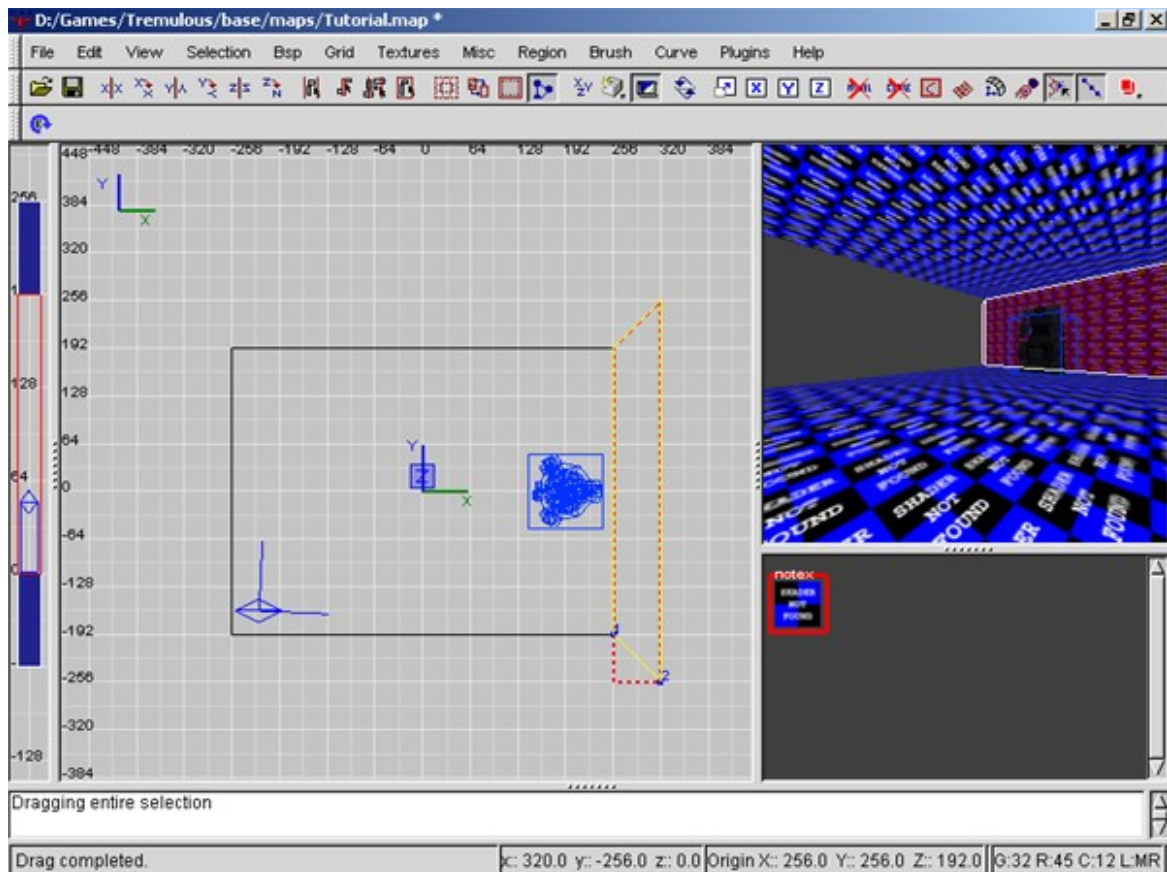
put in a temporary scale help by entering a reactor into the map. This will help you in getting the size right. To do this right click on the grid and select the reactor from the team menu.



Now since we're doing this the simple way we'll make the floor at editor ground level. Better known as Z-Axis 0.

For reasons which are hard to explain and not at benefit to the tutorial always try making walls and floors at least 32 gameunits (gu) thick. This is as thick as a single block at the grid size selected by pressing 6. I usually make mine even thicker.

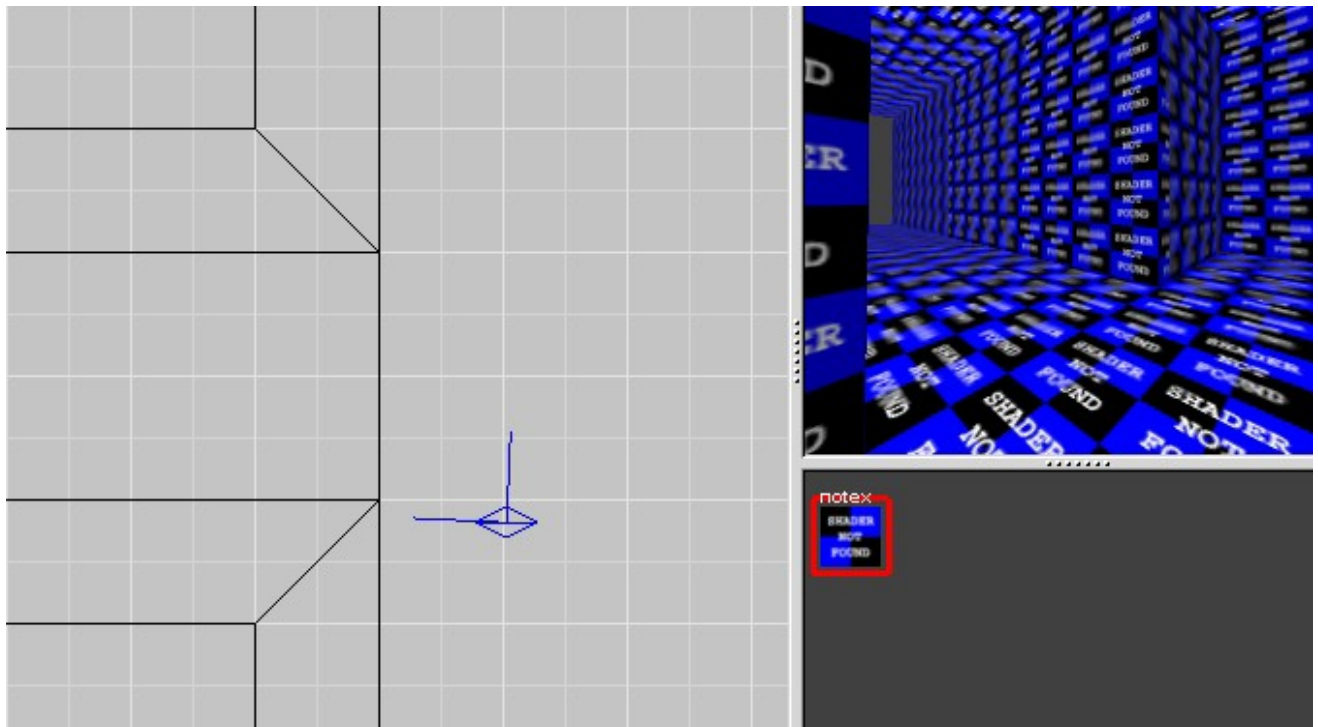
For an easy ceiling you can clone the floor. Have the floor selected and press the spacebar to clone the floor. Now drag it upwards for about a difference of 192 gu using the Z-Axis and put it back right on top of the floor. Now we are going to make 1 wall and clone it three times. But we will use the clipper to get it at its most efficient way, why this is efficient will be explained in the map optimization part of the guide. This is important in larger maps where people can complain about fps and it can prevent certain light problems. It only needs to touch the corners of the floor and ceiling but if there is even a very small opening the map is open to the void and can't be played.



When you are done clone it and turn it 90 degrees over the Z-axis by pressing the Z-axis Rotate button in the toolbar. Adjust the length to fit the other side of the floor. Select the first wall you made again, clone it, but this time press the X-Flip button in the toolbar. Drag it across to the other side. You can do the same with the second wall you made but using the Y-Flip button in the toolbar this time. If you have done this all your first room is done, but it isn't playable yet. But that isn't so bad since I'm not done yet either. The gameflow chart showed we had 2 entrances to the human base engagement area. Next up is the first hallway.

## Connecting Hallways

Your first hallway will be made by splitting the left side wall using the clipper and making an opening that will be about as wide as the reactor. Now use the clipper again to make a sharp corner and add the walls and ceiling of the hallway.



Now you can add a basic engagement area which can consist of several rooms or just a single room in the same way as explained above. Feel free to experiment with height of the room. Keep doing this in the same way until you've got the entire layout done. This means there are no decorations or lights, only hallways and rooms for now.

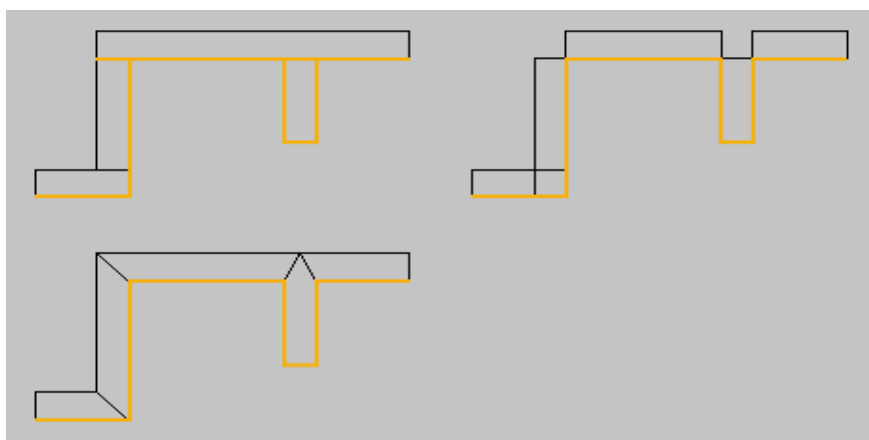
## Map Optimization

Map optimization is a separate chapter of this guide which could be skipped but that will result in less than optimal maps. I urge you to at least read and understand its principals.

Why do we clip the walls like that?

We clip walls like that because of two things. Ease of texturing and optimization.

Let's look at some ways of connecting walls. The textured faces have been coloured to show what I mean in the explanation.



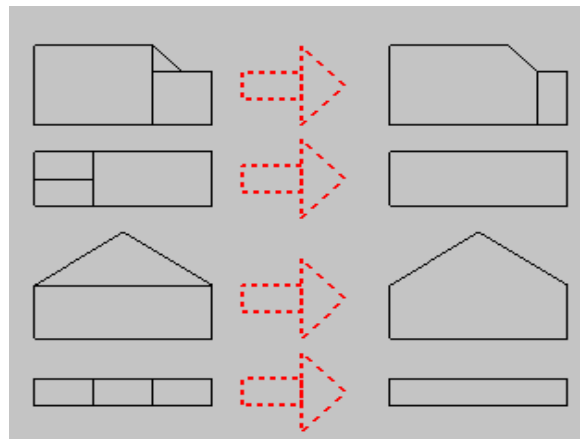
The upper left example shows the most common way a new mapper would connect them. This results however in parts of textures being drawn by the engine but never being seen by the player. As such they are a waste of resources.

Also notice how one wall has been put at a ninety degree angle at the top right of this example. This might seem nice and all but it will result in light bleeding. Light bleeding is light passing through walls. This is the case here since lighting is calculated on faces of brushes. But the face of the top wall is at both sides of the perpendicular wall. Thus light will be calculated on that face and pass from one side to the other. This looks like crap. Also notice the bottom left corner in this example. You can see you have a problem with the continuing texture because that wall consists of one large face and one small one. If you want to have complex walls this is a no-no.

Now the top left example alleviates some of the problems. But it looks disorganised already and this is only a few walls. This is why in my opinion the bottom left example is the best way to work.

### Efficient Brush Building

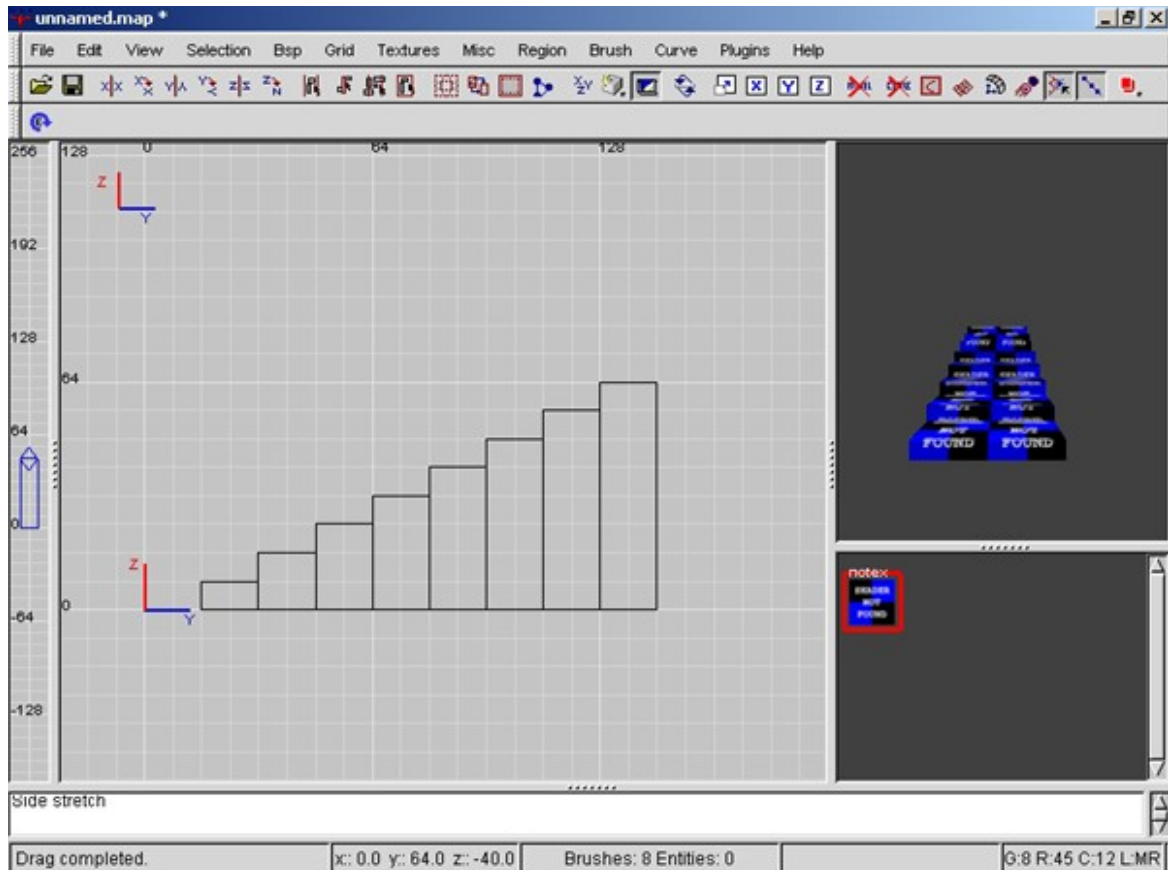
Unless you want to apply different textures to neighbouring brushes always try to merge them with `csgmerge`. This makes the brush and polycount lower which reduces resource usage. For several examples look at the below image.



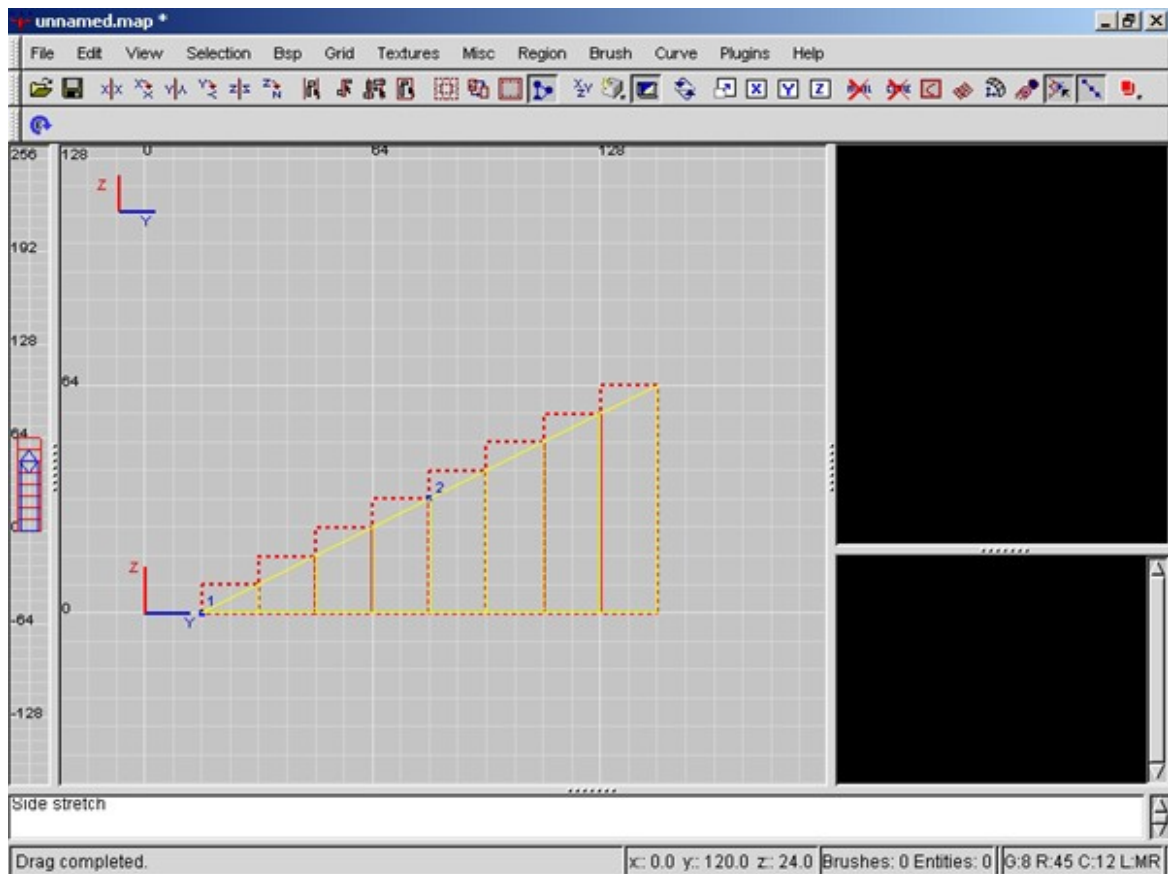
Optimized Stair

We'll now make your basic stair. This is best explained in a series of images than on text. The clipper is used in this part.

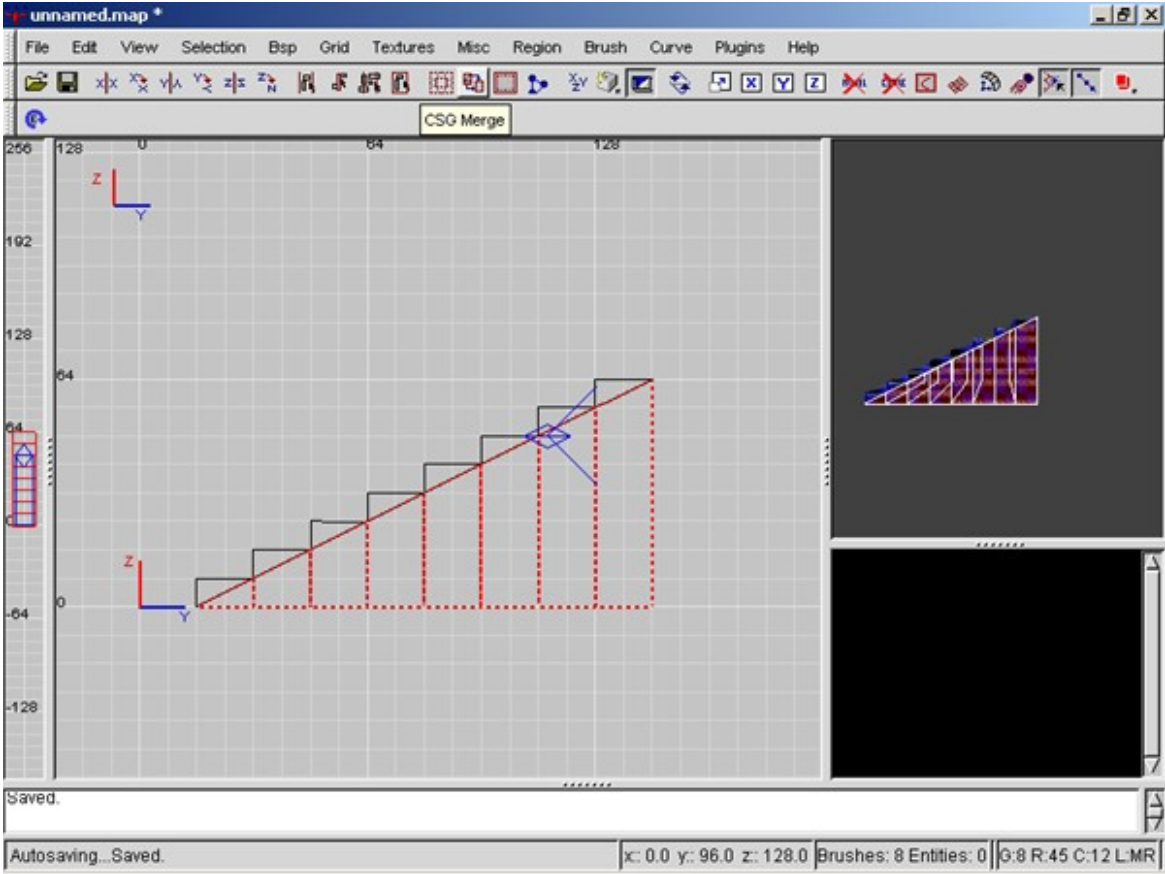
The first step, step depth and height are custom but try not to make stairs too steep since players will bounce off it instead of walk. Also steps higher than 18 gu can't be used in a stair. Note that I used `ctrl+tab` to switch to a side view in this screenshot.



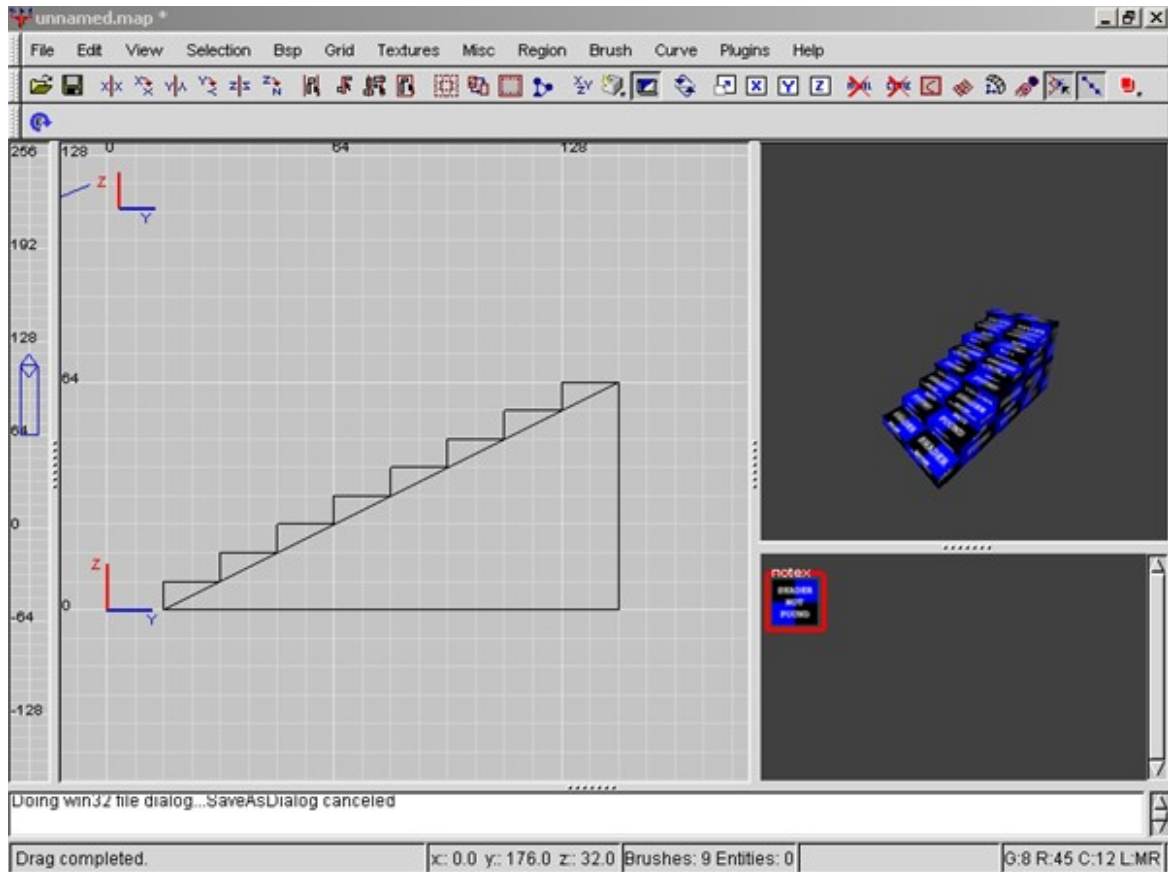
I used shift+enter during the clipping stage here for an easy next step.



We will use CSG-merge now to combine the lower half into one easy brush which is efficient resourcewise.



And the end result would be:



Don't forget to apply caulk to the unseen faces of the stairs during the texturing stage to actually make it worthwhile!

## Vents

Vents are nice, 45 degree turns are not. This is because of simple math. Bounding boxes or hitboxes don't turn when the player turns. They stay aligned to the main axis. This results in the appliance of Pythagoras. A dretch can fit in a 35 gu wide vent aligned to the x-axis but if the vent makes a 45 degree turn it needs to be at least the square root out of  $(30 \times 30)(30 \times 30) = 42,42 = 43$  gu wide. Remember this when making such turns.

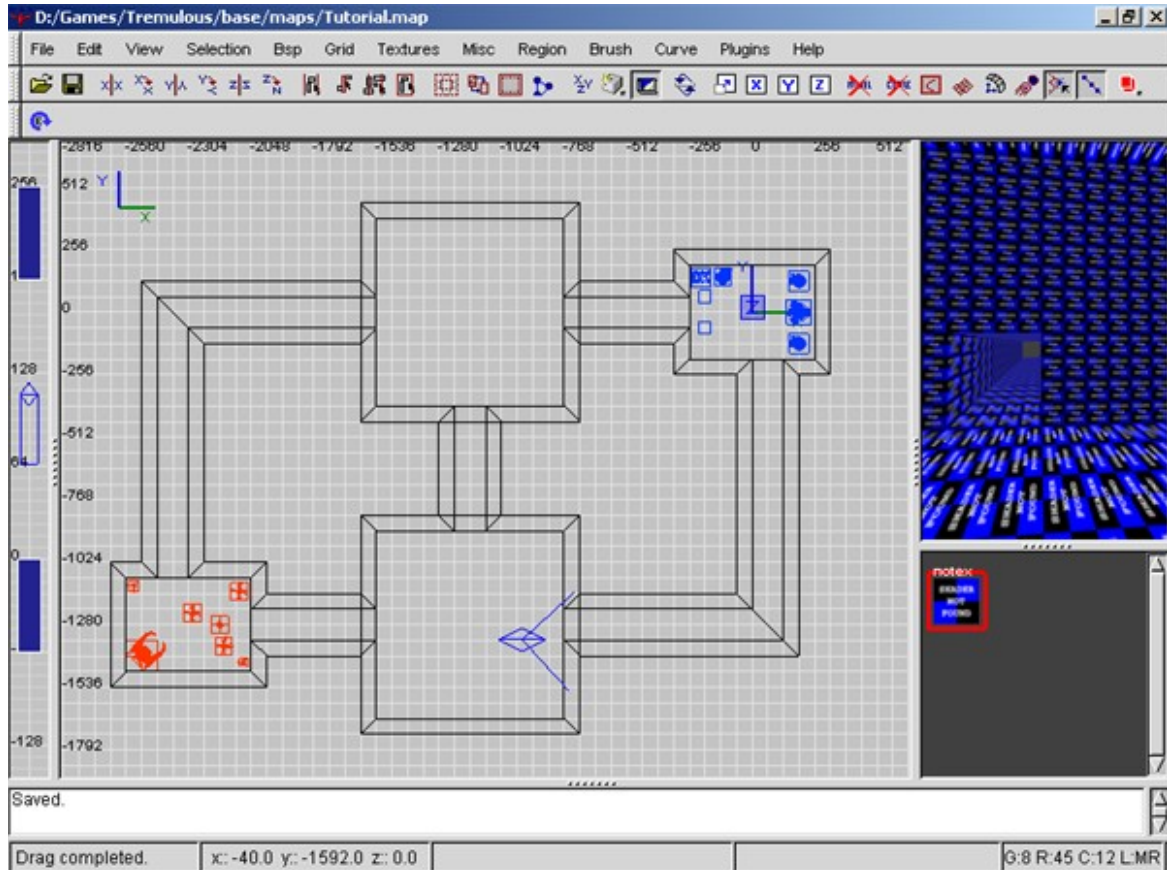
## Continuation of Tutorial Inserting Bases

Next up we will add our bases. Keep in mind that you want to start with only stage 1 buildings. You can do it the same way you entered the reactor at the beginning of this guide. You can also select a building and use the angle keys to make them face a certain direction. This is important for people spawning with their face away from the wall and making turrets face the right way at the start. If you enter a turret you don't see a model, only a bounding box. Don't worry, this has to do with the dynamic nature of the turret ingame. The angle keys still point it in the right direction. You might also want to put some thought in cover for certain buildings which we will put in later. Also important is that the structures should be a bit above the ground or else they will not spawn. Right click on the grid where you want your building and select it from the team submenu. Remember that this is a small symmetrical



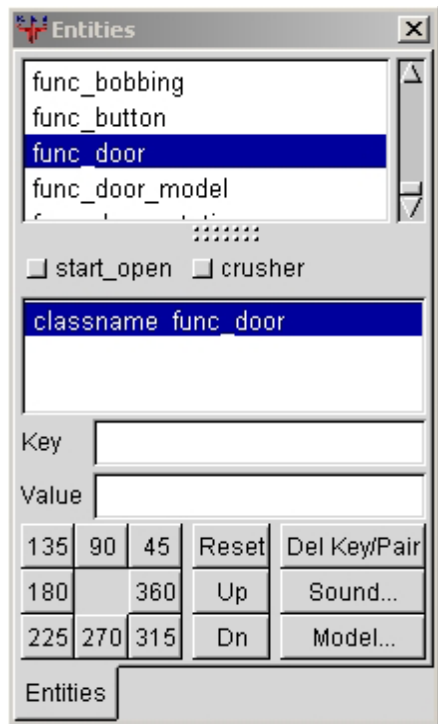
map. Any larger non symmetrical maps needs to include the gameflow guide when placing the bases.

It should look something like this now.



## Making a Defective Door

Now the gimmick in the map; the stage 2 activated doors. We'll start with the simple door and then add the stage 2 activator. A door is an entity. It can move up (-1), down (-2) or in divisions of 360 degrees around the Z-axis (0 to 360). It can not go diagonal upwards. For that you will have to use a platform and advanced mechanisms which will not be discussed here. Start by making a brush in the shape of the door in it's closed position. When this is done keep it selected and right click on it. Choose `func>func_door`. Now it is a defective door. We'll proceed to entering the data that will make it work. While having the door selected press N.



## The Entity window (n)

This is the entity window. It applies to just about everything in the right click menu. For now I'll give you a short description.

In the upper part you can select the entity you want the selected brush to be.

The regular brush is worldspawn but we will be using func\_door.

Below that is a readme part concerning the selected class. You might have to drag the double dotted line to get it to show up.

Then there usually are a series of buttons you can read about in the readme part.

Below that comes the actual data of the entity in the form of keys and values which can be entered below. These are usually explained in the readme part as well.

## Defective to Working Door

We will enter angle as a key and -1 as a value here, this will make the door go up. Now press enter to have the door use this data. Alternatively you could use the lowerleft corner to set angle. Next we want the door to open slowly, the usual speed has a value of 100 but we want 30% of that. Replace angle with speed and -1 with 30 and press enter to enter this second set of data. Now it is a functioning door.

## Including a Trigger

For having a door activated by a button or trigger a door needs a targetname. We need this for the stage 2 trigger so enter targetname as a key and stage2door1 as the value. Press enter to have the door assume this name.

Now we will add the trigger. Deselect everything and right click, now select trigger>trigger\_stage. This will make a small box appear. Make sure to not let it be located in the void since it is an entity. Press N to get to its data. Now enter the following data.

Key: target                      Value: stage2door1

Key: team                      Value: 1 (for the alien side door, use 2 for the human side door)

Key: stage                      Value: 1

This will make all entities by the name of stage2door1 be activated when the aliens level up to stage 2. Repeat this for the other door using the opposite team and another name for the door like stage2door2.

## Decoration and Detailing

Next we will start adding some decorations and supports. Decoration should not block the vis field because it results in very long compile times.

The vis field is a strange subject to explain. Basically when your map is compiled it goes through the VIS stage (hopefully). This calculates what can be seen from every spot in the map and divides it into sections defined by vis portals which mostly have the same view, you can alter this with hinting but that will be discussed another time. The vis as such is blocked by brushes. If the map has a lot of vis portals it needs to load a lot of vis fields which is not efficient since it is wasting processing power. You might want to have some easy overlapping in this situation.

This is why most decorations should be made detail brushes. Detail brushes do not block the vis and thus allow for simpler vis fields. This can be done by having a brush selected and pressing Ctrl+M. This will make the selected brush detail. By pressing Ctrl+D the present detail brushes are toggled on and off in the radiant view. This is very handy with high detail work in your later maps. Remember to never make a brush connected to the void detail since it will result in a leak and the map will not compile. Decorations usually include lamps, trimmings and anything else that is smaller than a player at the least and only do this to regular brushes, not entities! Please note that with all the detail brushes hidden it looks exactly like the hull we had previously. It doesn't need to look exactly like your hull but it should look very simple.

Don't forget that Tremulous' alien radar in its current state is affected by the vis. You don't want to give humans the advantage of a radar free corner since that isn't how the game is supposed to work. Detail allows radar while structural brushes block it.

### Patches

There is a certain class of elements known as patches. Cylinders, endcaps, bevels and the simple patch mesh are the most known ones. I will only shortly explain their use since they are not that hard. You make them by making a brush the size of the patch you want. While you have it selected you click the curve menu and select what you want. The brush will be converted to a patch. You can alter this patch in several ways but you can't use the clipper on it. Also remember that patches are not structural. If a patch separates an entity from the void you will get a leak. In this case you might want to patch the hole behind the patch with a caulk wall. To deform a patch you can use bend (b) or edges(e) but the one with the most control is vertices(v). Vertices gives you several vertice points you can play with to achieve what you want like bending a pipe or patch. Try it out and get a hang for it. It's the best way to get rounded objects in maps.

## Texturing

Now comes the graphical part, texturing and lighting. We'll start with texturing using ATCS's texture set. But first load up the common set by using the texture menu at the top and simply select common. The common texture set is the Swiss-army knife like texture collection. Most will be explained in tutorials around the internet so I won't go into all of them now.

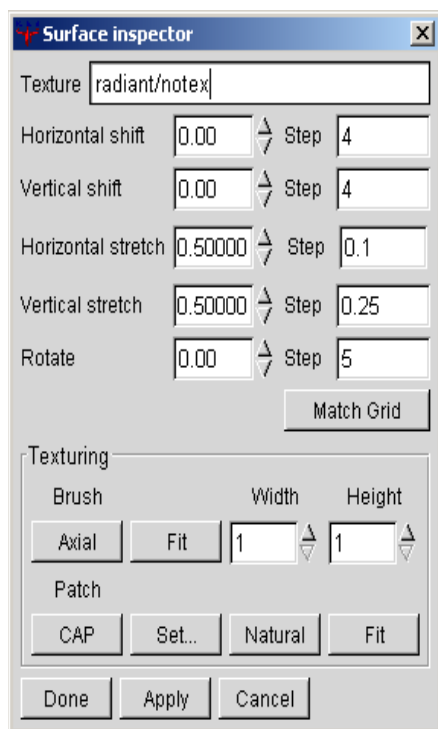
Now deselect everything and press shift+A. If you haven't textured anything yet everything using the notex texture should be selected. Now select caulk in the texture window. Caulk is a texture that isn't drawn ingame by tremulous thus it has no impact on the performance. When caulk is visible to a player however you get what is known as a Hall of Mirrors effect, very ugly and unwanted so make sure you don't leave any caulk open to the player's eyes.

There are two ways to texture and several things to look out for. You can select a brush in the normal way and apply a texture; the texture will be on all faces of the brush. This is usually not wanted. Better is using ctrl+shift+left click; this will select a single face of a brush. If you keep ctrl+shift+left click down and drag it you can select multiple surfaces which lessens the workload.

One thing to look out for are that textures aren't spread too large so they appear ugly in game. Another thing is to be careful with the textures surrounded by a white border in the texture window. These are shaders. They can range from simple light emitting textures, image sequences to deforming surfaces. Be sure to use the white bordered lights since others might simply not emit any light. You can now select the ATCS texture set or any other through the texture menu at the top. Now go forth and texture your map.

## Surface Inspectors (s & shift s)

Now you might have noticed that texturing is off on some points. The texture is misaligned or repeated more then you want. This is where the surface inspector comes in at the press of the S key.



Texture points to the texture to be used.

The 5 following should be pretty clear with the first box being the current value and the step size the amount it changes when using the arrows. You can directly alter both values.

Texturing>Brush

Axial sizes and orients it to the axis.

Fit fits the texture the specified amount of times to the right on the selected face(s).

Texturing>Patch

Cap is best applied to caps of cylinders and such.

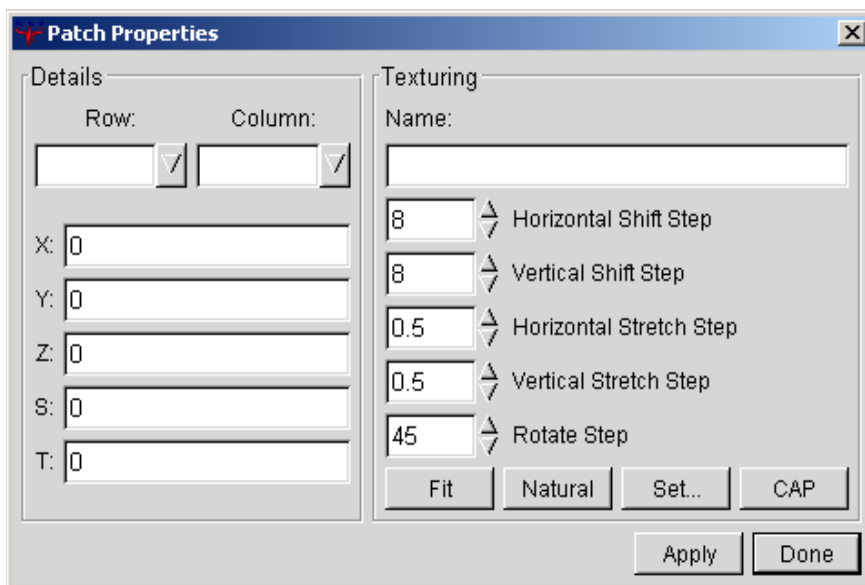
Set is not often used.

Natural can be used to correct awry displaying of textures

Fit fits

Now there is also a surface inspector for patches activated by pressing shift+S.

This one is better to use on patches, bevels and end caps.



Most things are the same except that you'll want to use natural a lot because of the curved nature of most patches.

## Lighting

Final point before coming to a complete map is lighting. This is really not a matter that can be expressively taught or explained. I'll go over the basics and the settings but where to put them and which settings to use strongly depends on everything else in the map and the mood you're trying to create so I can't help with that.

There are two main sources of light. The point lights and the light emitting textures. Also available for an overall light is the ambient key with a value entered in any worldspawn.

The point lights can be selected by right clicking and selecting either light or lightJunior.

Light is the normal light originating from one point.

When you select light it will ask for a value, don't worry, you'll change it anyway. Just click ok. Press N to bring up the entity window.

Light specifies the strength of the light.

\_color specifies the color by a 256 point system, you simply divide 1 by the R, G and B value each to get the color values you enter.

Target allows for the making of a spotlight.  
The rest of the settings are not important for starting mappers.

lightJunior is almost entirely the same as light, the only difference is it only affects entities.

Light emitting textures are actually shaders and their creation will be explained in the shader section of this manual.

Now come some tips to lighting.

By preference use light emitting textures. They usually look better in place then light coming from nowhere.

If you use point lights don't make the values too big since you will get ugly spots, it's better to use several small lights.

Don't let to many lights overlap, this will increase compile times significantly.

## Some little important tidbits

When loading a map in tremulous the game checks for the info\_intermissions. The least a game needs to load is the info\_player\_intermission from the right-click>info menu. This is the spectator start view when you enter the map. The info\_alien\_intermission is the spectating point for the alien team when dead and the info\_human\_intermission is for the human team. The direction settings for them are as follows. They cannot currently be aimed using a target\_position. You need to enter an angles key to aim them in a polar-like coordinate system. Input: Angles (x in degrees) (y in degrees, -degrees is up) 0 (roll).

The position of the alien and human intermission usually is aimed at the starting base. The player\_intermission should be aimed at either a map trademark or the most important engagement area.

## Compiling your Map

Now we are going to compile the map for the first time. I usually compile it several times before this to get the gameflow right. I do this before applying caulk to all brushes.

You can compile with the compiler that came with GTKradiant but it's better to use the newest q3map2 and q3map2toolz (Could Megabite upload the latest versions to tremulous.info since the official shaderlab is gone?). Since there are a lot of options for compiling I can't go through all of them, nor should all be used. I will sum up the essential switches and an explanation why.

BSP: Binary Space Partitioning, Just do it

Vis: Calculates the visibility from the portals, Performance wise important

Light: Calculates the light, pretty obvious

Custinfoparms (custominforparms): Needed for the nobuild textures.

-meta: I don't know exactly what it does but it's needed for advanced compiling. Use it

-patchshadows: Shadows from patches, was previously not available.

-patchmeta: Advised for preventing patches going awry. Not necessary but advised

-bounce is a common lightbleeding switch if you don't know how to use them so don't at the start.

## Testing your Map

You can try your map out now by loading up tremulous, pulling down the console by pressing ~ and entering /sv\_pure 0 and /devmap mapname.

Now correct the mistakes in gameflow and construction and try again, several times. Don't worry, your first map will be crap, so will your second probably but practice makes perfect.

Now when the blatant mistakes are removed the alpha stage is complete. You can try and submit it to tremulous.info to get it on the derelict server as a beta but only maps which have some work and potential behind them will get on. I advice to never release or even mention maps in alpha stage unless you are having problems you can't find the answer to on the internet.

## Naming conventions

First important thing is a clear naming system which I'll try to set an example for now. The system I will be proposing is the one used by soubok which allows for many versions with short names. It goes like this. 'mapname\_version.bsp'. Example: 'mymap\_34.bsp'. Save the .map for that version with the same name and if you alter it up the version number.

To make a map in the .pk3 format you can use the mapacker included in q3map2toolz. This is the easy way and I recommend it, rename it to 'Map-mapname-status-tremulous version number.pk3'. Example: 'Map-mymap-beta2-1.1.0.pk3'  
Status is either Alpha or beta, when a map is finished the status section should be removed.

## Layout of the .PK3

The other way is by hand. A .pk3 is simply a .zip renamed to .pk3 but manually selecting all textures you used is a tiresome job. I will simply describe the internal components of the .pk3 and what they do. I suggest using the naming conventions above none the less.

Mapname.pk3

    Readme.txt

    Textures\mapname\_version\ (Your textures here)

        Env\ (Skybox usually)

    Levelshots\ (Screenshot for loading screen in the form of 'mapname\_version.jpeg')

    Maps\mapname\_version.bsp (your .bsp here)

    Scripts\

        mapname\_version.arena (Information file)

        mapname\_version.shader (Shader file for texture effects and special surfaces)

        mapname\_version.particle (Particle file, advanced mapping)

        mapname\_version.trail (Trail file, advanced mapping)

    Sounds\ (Your sounds here)

In the readme you mention the following

- Mapname, version, date.
  - contactinfo for contacting the mapper, usually an e-mail address or site (you).
  - any notes and extra information you'd like to add.
  - people whose textures, shaders, particle systems etc you used.
- If you want your map to be included in an official release you are also required to release it under the following license and include the .map file in the .pk3.
- 

## LICENSES:

=====

This map and all medias this map use are under creative commons license Attribution-ShareAlike 2.5 ( <http://creativecommons.org/licenses/by-sa/2.5/> )

Attribution-ShareAlike 2.5

You are free:

- to copy, distribute, display, and perform the work
- to make derivative works
- to make commercial use of the work

Under the following conditions:

- Attribution: You must attribute the work in the manner specified by the author or licensor.
  - Share Alike: If you alter, transform, or build upon this work, you may distribute the resulting work only under a license identical to this one.
- 

## .Arena

I will now describe what needs to be in the .arena file and after that some simple shader examples like water, glass, light emitting textures and the ladder.

Mapname.arena

```
{
  map      "mapname_version"
  longname "Displayed name in loading screen"
  type     "tremulous"
}
```

## .Shader and shader introduction

Now the shaders. These are sections of code that go into the .shader file to define the effects. The orange(colour change megabyte?) bits are comments and not necessary in the shader file but for clarification purposes. If a certain keyword has already been mentioned I won't mention it in the following shaders unless it has a different function. Check out the shadermanual (direct link <http://www.heppler.com/shader/shader/index.html> put on tremulous.info?) for all information. I include these because people might want to use them or know how they were done.



## LIGHT EMITTING SHADER

First up the used .jpgs

Largelamplight.jpg



Largelamplight.blend.jpg



```
textures/mymapname/largelamplight //Defines when to load if selected from texture menu
{
    qer_editorimage textures/mymapname/largelamplight.jpg //image displayed in radiant
    q3map_lightimage textures/mymapname/largelamplight.blend.jpg //emitting pattern
    surfaceparm nomarks //This makes sure no shotmarks cover the lamp
    q3map_surfacelight 1250 //This is the setting that makes the texture emit light
    {
        map textures/mymapname/largelamplight.jpg //Texture displayed in map
    }
    {
        map $lightmap //maps lightmap in
        blendfunc filter //filters dark parts
        tcGen lightmap //generates lightmap
    }
    {
        map textures/mymapname/largelamplight.blend.jpg //Overlay source, kinda
        blendfunc add //Adds previous image ignoring the darker parts.
    }
}
```

## MIRROR

To make a mirror you apply the following shader to one side of a brush and put a misc\_portal within 64 gameunits of it that doesn't target a camera

```
textures/mymapname/mirror
{
    qer_editorimage textures/mymapname/mirrorbackground.tga

    noPicMip //Doesn't get bad quality render at distance
    noMipMaps //Doesn't get bad quality render at distance
```

```

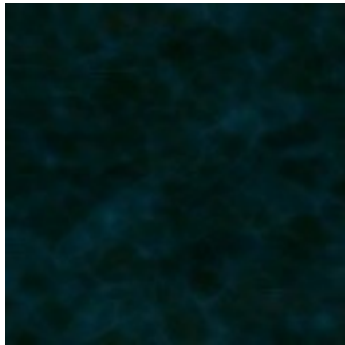
surfaceparm nolightmap //Not affected by lightmap
surfaceparm playerclip //Player can not walkthrough, not really necessary.
portal //Makes the mirror functioning
{
    map textures/mymapname/mirrorbackground.tga //Black works best
    blendFunc GL_ONE GL_ONE_MINUS_SRC_ALPHA // Advanced blendfunc
    depthWrite //Advanced
    alphaGen portal 256 // Sets fade distance of mirror in gu
}
}

```

## GLASS

//Created by Louie "KnightBK" Doulias for Black Knights Productions, Prisoners of War Q3 Mod.

//Please check out our mod site at <http://www.og-world.com/bkp/>



```

textures/mymapname/plainglass
{
    qer_editorimage textures/mymapname/plainglass.jpg
    surfaceparm nolightmap
    surfaceparm trans //Doesn't block visibility calculation
    surfaceparm solid //Player cannot pass through
    surfaceparm nomarks
    {
        map textures/mymapname/plainglass.tga
        tcGen environment //This shader is environment mapped
        blendfunc GL_ONE GL_ONE //Makes transparent
    }
}

```

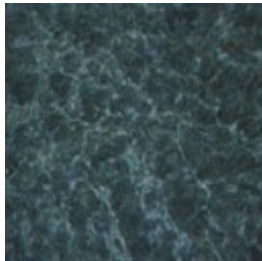
## LADDER



```
textures/common/ladder
{
    qer_trans .60 //Defines editor transparency
    qer_editorimage textures/common/ladder.jpg
    surfaceparm ladder //Makes a brush textured with this function as a ladder
    surfaceparm nodraw //Not seen ingame
    surfaceparm nolightmap
    surfaceparm nonsolid //Player can walk through brush (if alien in this case)
    surfaceparm trans
    surfaceparm nomarks
    surfaceparm noimpact //Projectiles pass through
    surfaceparm playerclip //Players can't walk through
}
```

WATER (taken from pulse, copyright Soubok)

To make water apply this texture to all sides of the brush you want to be water. This brush may intersect with other brushes without z fighting. You must use the white bordered texture to make it work.



```
textures/pulse/water
{
    qer_editorimage textures/pulse/water.tga
    qer_trans .5

    entityMergable

    q3map_globaltexture //Corrects problems with the tmod scale function

    surfaceparm nobuild
    surfaceparm trans
    surfaceparm nonsolid
    surfaceparm water //Defines water properties

    surfaceparm fog //Adds fog property for lowering visibility of water
    fogparms ( 0.01 0.05 0.05 ) 1024 //fog settings (R,G,B) Distance to 0% Transparency

    cull disable //Removes cull, cull makes a texture be applied to only one face of a side.
```

```
tesssize 128 //tesssize defines the amount of triangles for deformation.
deformVertexes wave 100 sin 1 2 1 .1 //Brush deformation settings
```

```
{
    map textures/pulse/water.tga
    blendfunc GL_ONE GL_SRC_COLOR //Advanced blendfunc
    tcMod scale .03 .03 //Scale texture
    tcMod scroll .01 .001 //Moves texture
}
{
    map textures/pulse/water.tga
    blendfunc GL_DST_COLOR GL_ONE //Advanced Blendfunc
    tcMod turb .1 .1 0 .01 //Turbulence effect to distort the repeating texture effect.
    tcMod scale .5 .5
    tcMod scroll -.02 .1
}
{
    map $lightmap
    rgbGen identity //Colors specified to be white (1.0 1.0 1.0)
    tcGen lightmap //Soubok comment? Dunno
    blendfunc filter
}
}
```

HINTSKIP (later used with hint, advanced mapping)



```
// ydnar: skip works like quake 2 hint: it doesn't generate bsp splits
// use on sides of hint brushes where you don't want bsp splits or portals
// add this to your common.shader file
```

```
textures/common/hintskip
{
    qer_trans .60
    qer_editorimage textures/common/hintskip.jpg
    qer_nocarve //Not affected by csg subtract
    surfaceparm nodraw
    surfaceparm nonsolid
    surfaceparm structural //Advanced, bsp breakpoint
    surfaceparm trans
    surfaceparm noimpact
    surfaceparm skip //Skipped in vis process
}
```

## Addendum

Some other important things to remember:

Playersizes:

Aliens

<b>Class</b>	<b>Size</b>
Granger	40x40x40
Adv Granger	50x50x50
Dretch	30x30x30
Basilisk	36x36x36
Adv. Basilisk	42x42x42
Marauder	50x50x38
Adv. Marauder	52x52x42
Dragoon	52x52x55
Adv Dragoon	64x64x67
Tyrant	64x64x92

Humans

<b>Stance</b>	<b>Size</b>
Standing	30x30x56
Crouching	30x30x40

### Common Start Errors

Can't find map maps/mapname.bsp

You haven't made your map .pk3 yet or didn't set /sv\_pure to 0. Do either and try again.

My map doesn't show up in the map list.

Check if your .pk3 includes a correct .arena file. Non .pk3 maps do not show up in the map list.

One of the team wins as soon as the maps load.

The losing team has no eggs or telenodes due to them simply not being included or because they are partially inside a brush.

My map is too dark/bright when I load it.

This is a combination of the right /r\_gamma setting (online guide value is 1.5) and the light intensity you entered in radiant. Adjust /r\_gamma to 1.5 and/or change the values in radiant accordingly.

## Mapping Commands Ingame

Useful mapping commands and what they do, set them to 0 to disable them again:

~ : brings down the console where you can enter the following commands.

/sv\_pure 0 : allows non .pk3 maps to be loaded

/devmap mapname : loads specified maps with cheats enabled for the following commands

/noclip : you can move anywhere in the map:

/r\_clear 1: Makes the HOM effect in void pink. Also useful for spotting caulk.

/r\_gamma 1: 1.5 is a common aim for the lighting but 1 is real. Adjusts light intensity.

/r\_showtris 1: You can see the rendered vis fields from your position. Optimization analysis.

/r\_speeds 1: Displays rendering speeds. Do not let them get too high. 30k is border case.

/cg\_drawFPS 1: fps counter on screen. Optimization analysis.

/cg\_draw2D 0: removes hud from screen, use with cg\_drawgun 0 for levelshot.

/cg\_drawgun 0: removes gun from screen, use with cg\_draw2D 0 for levelshot.

## Game Altering Commands

/g\_Alienbuildpoints xxx : sets the alien's buildpoint level

/g\_Humanbuildpoints xxx : sets the human's buildpoint level

/g\_Humanstage2threshold x: Human stage 2 threshold kill count (triggers trigger\_stage)

/g\_Humanstage3threshold x: Human stage 3 threshold kill count (triggers trigger\_stage)

/g\_Alienstage2threshold x: Alien stage 2 threshold kill count (triggers trigger\_stage)

/g\_Alienstage3threshold x: Alien stage 3 threshold kill count (triggers trigger\_stage)

/g\_HumanStage x : 0 for stage 1, 1 for stage 2 and 2 for stage 3 humans. No trigger\_stage

/g\_AlienStage x : 0 for stage 1, 1 for stage 2 and 2 for stage 3 aliens. No trigger\_stage

/give funds xxx : xxx is a specification of the amount of credits/ evolves you will receive.

/g\_gravity xxx : gravity setting, 800 is default.

This concludes this guide to mapping. If you are getting certain errors not mentioned here try reading this (link <http://quake3world.com/forum/viewtopic.php?t=2418> maybe index and add to tremulous.info megabite? It will stay up as long as quake3world stays up and that's always a surprise.)