3Wave level requirements

For my own reference also I have found on the Internet the old mapping instructions for Quake 3 Threewave CTF, one of the best mod for Quake 3 Arena. Since these instructions have been very hard to found I decided to put a copy of the original text on my webpage. I think these are greats instructions for mapping a CTF map for Quake 3 Arena.

ThreeWave level requirements

Cross platform compatibility – Q3CTF – CCTF – TA gametypes

Version 1.2 updated: Oct 4, 2001 3:40 AM PST

ThreeWave level requirements
Cross platform compatibility – Q3CTF – CCTF – TA gametypes
Version 1.2 updated: Oct 4, 2001 3:40 AM PST
Author – Casey

Special thanks to Paul Jaquays, Johnny Law, Mr. Elusive, Ark,
and Robert Duffy for assistance or additions to this material.

FOR COMPATIBILITY USE NO TEAM ARENA TEXTURES, SOUNDS OR ASSETS IN YOUR LEVELS!

CTF Entity support

Each level should have the following:

1 worldspawn message – mymapname – 3W
1 *info_player_start*
1 *info_player_intermission*
1 *team_CTF_redflag* (gametype: ctf,oneflag)
1 *team_CTF_blueflag* (gametype: ctf,oneflag)
1 *team_CTF_neutralflag* (middle of level) (gametype: oneflag)
1 *team_neutralobelisk* (middle of level) (gametype: oneflag,harvester) !!!!!

1 *team_redobelisk* (in base at flag spot)
1 *team_blueobelisk* (in base at flag spot)
16 *team_CTF_redplayer* (These should all be in or near the red base. Initial spawn point)
16 *team_CTF_blueplayer* (These should all be in or near the blue base. Initial spawn point)
10 – 16 *team_CTF_redspawn* (These should range from the red base to 75% of the way to blue base)
10 – 16 *team_CTF_bluespawn* (These should range from the blue base to 75% of the way to red base)
10 – 16 *info_player_deathmatch* (In case people DM on the map, or future use, and just because)
target_location entities as detailed further below

Optional Level items:

CCTF
weapon_nailgun (same entity as TA)
ammo_nails (same entity as TA)
item_armor_green

TA:
item_guard
item_doubler
item_ammoregen
item_scout
weapon_nailgun (same entity as CCTF)
ammo_nails (same entity as CCTF)
weapon_chaingun
ammo_belt (chaingun ammo)
weapon_prox_launcher
ammo_mines
holdable_kamikaze
holdable_invulnerability

There is more detailed information about CCTF and Team Arena items further below.
Note: Do not use more than 16 of each type of spawn point, the game doesn't recognize any more than the first 16 you put in your level

---

Gametypes Separations

Team Arena and the 1.2x and onward Quake 3 EXE support the marking of almost any entity for use in any specific gametype, or all gametypes. You can add different weapons

and items to different gametypes etc. You have to specify what games each entity is used for if it is only in certain gametypes. Entities in all gametypes don't require these keys.

You need to mark the gametypes to spawn some of the items specific to Team Arena gametypes, these are shown in brackets in (red) beside entities that require them in the above entity support list.

In the above section you will find all the entities needed in the game, also there is a Team Arena mapping manual here, and more specifically the page you'll want to read is this one. Note that we aren't following all the procedures for TA mapping, as we have our own custom decals for many of the Team Arena items and some other changes.

ffa – g_gametype 0
tournament – g_gametype 1
single – g_gametype 2
team – g_gametype 3
ctf – g_gametype 4
oneflag – g_gametype 5
obelisk – g_gametype 6
harvester – g_gametype 7

In the entity editor you would use:
Key: ***gametype***
Value: ***ctf,oneflag,ffa,obelisk*** (seperated,by,comma's,and,no,spaces)

If you read the TA mapping manual carefully you'll see that in TA, the *redobelisk* will automatically spawn in CTF, and OneFlag as the base for the red flag. The same goes for the blue and neutral obelisk's. If you place the items in a simple manner, with the flags and the obelisks etc in the same place together in each base and the middle of the map, you will have no problems. If however, you feel that your map would lend itself better to having the obelisk for overload in a different spot than the flag, you can specify the gametypes, and place multiple copies of each entity. The list above should make it easy to place all the items in your level and set the gametypes as long as your flags and obelisks are all in the same place in the bases and the middle of the level. However, if you want to place them differently, go through all the gametypes on this page and place the items for each gametype, and mark them for those gametypes in the entity properties.

Extended Gametypes

Any *entity* that you don't want to spawn in Team Arena, or in Quake3Arena or Classic Capture the Flag can be removed by using any of the following *keys* set to a *value* of **1**

The Keys are: **notta notq3a** **notcctf**

Additionally, CCTF has one more key that you can use to include an entity specifically in CCTF.
That *key* is: **notq3ayescctf**

The use of this key on an *entity* for CCTF only would include the following keys and values:
*key: **notq3a** value: **1** key: **notta** value: **1** key: **notq3ayescctf** value: **1***

Here is a chart showing all the possible combinations of Extended Gametypes.

notq3a	notta	notcctf	notq3ayescctf	Show in Q3CTF?	Show in TA	Show in CCTF?
0	0	0	0	yes	yes	yes
0	0	1	0	yes	yes	no
0	1	1	0	yes	no	no
0	1	0	0	yes	no	yes
1	1	0	1	no	no	yes

1	0	0	1	no	yes	yes
1	0	0	0	no	yes	no
1	1	0	0	no	no	no
1	1	1	0	no	no	no
1	1	1	1	no	no	NO!
0	0	0	1	YES!	yes	yes

The bottom values in red indicate values that are basically invalid, do not use them in those combinations. Items that are set to *notq3a* will **not** show up in CCTF unless you also set the *notq3ayescctf* key as well. *notq3ayescctf* will **not** work in Q3A to remove items, it is only to add items to CCTF that have already been removed from Q3A by using *notq3a.*

Detailed CCTF Support

green armor (*item_armor_green*)
nailgun (*weapon_nailgun*) (Same entity as Team Arena)
nailgun ammo (*ammo_nails*) Same entity as Team Arena!!

NOTE: If you use the nailgun and nails, they will appear in TA and CCTF, but not in Q3CTF, make sure you put them in an inconspicuous spot, so that they don't look like something is missing in Q3CTF! You can use the *notta* and *notq3a*keys to have different weapons in Q3CTF than in TA and CCTF. Also, you could make the weapon spawn pads disappear in Q3CTF by making them a func_static and giving them the *notq3a* key.

CCTF has a key similar to Q3A and TA to remove things when played in CCTF mode. That is: Key**:** ***notcctf*** Value**: 1** One example of where you might want to use this is for bouncepads, or for a Q3 weapon that you want to remove so you can add the nailgun.

CCTF comes with an *entity.def* file that contains the CCTF specific items. Contact me if you have difficulty with this.

Danger: If your level has a fog pit, or pit of death, or is a space type level, you must read the section in fit and finish about "Fog pits, pits of death, and space maps etc". Make sure you design with the grapple in mind. you will need to block access to areas that normal players can't usually reach.

Detailed Team Arena Support

Tier 1 (mandatory)

Gametype entities:
– harvester machine (*team_neutralobelisk*) (middle of level)
– OneFlag entity (*team_CTF_neutralflag*) (middle of level)
– blue and red obelisk (*team_redobelisk* + *team_blueobelisk*) (in base at flag spot)

If your level has middle area that doesn't lend itself well to evenly placing the neutralflag and harvester machine, you should consider rebuilding a small portion of the middle of your map using *func_static* and enabling the geometry to disappear in those gametypes. This is quite easy to accomplish, and there is an example in the *sample.map* in the Assets Pak. One difficulty that you may run into is that geometry that is turned into *func_static* will not block vis. Also, I would assume that it probably isn't a good idea to build too much stuff using *func_static*, I'm not sure if there are any speed penalties for this type of geometry, but it can't be marked by weapons fire and can look odd if misjoined with normal geometry. Do not use *func_static* geometry to fully enclose or block off the entities in the middle of your map, as the nature of the bot .aas file will restrict them from ever seeing the middle entities even in the appropriate gametype. If you must surround the entities, it can only be accomplished using a *func_train* as detailed in the next paragraph.

Using *func_train* as a *func_static* that bots can see through in oneflag and harvester. I originally tried *func_door* with a lip equal to the height of the door, it worked fine, but it still

makes the door sound which is undesirable. I then tried*func_plat* with a height key of 0, that also worked fine, except when clients load the map, they are spammed with a warning that the plat sounds don't exist, also, if players are able to get on this object it will beep because the sounds are missing. While it would be possible to include silent plat sounds with the level, I didn't want to do this because of the possibility of a higher named pk3 file eventually having real plat sounds in it, which would be undesirable. I finally tried*func_train*, which is much more difficult to set up and tweak, but works. For the train, you need two *path_corner* entities, as the train will not appear of you only use one. The first *path_corner* should be set to wait 999999999999999 so the train will conceivably never move on to the second *path_corner*. The second *path_corner* should be near the first one, but not too close, or the func_train will not appear in the level. The train itself should be set to 'speed 9999' this way, in the event that a level runs long enough that the wait at the first path_corner times out, the train will quickly move to the next stop and back, almost appearing as a visual glitch to a player that might happen to see it. Aligning textures on a func_train can be difficult unless you read this post by Johnny Law on the Quake 3 World messageboard. **Note:** func_train appears draw no matter where you are in the level, so if you use this method make sure it's _very_simple_geometry_!!!

Tier 2 (recommended)

Techs. Think carefully about how play on your level will change with them added. Don't forget that the Q3CTF Haste and Regen etc will still show up in your level in TA mode. You can hide them with the *notta* key if you wish. Remember that TA techs **MUST** be marked as BLUETEAM or REDTEAM **!!!!**

– Guard
– Doubler
– Ammo-regen
– Scout

We have custom Pads for under the TA Techs since their *mapmodels* cannot be removed using the *gametype* key nor the *notq3a* key. Look for them in the Assets Pak as *prefabs,* and they can be seen in the sample map.

Tier 3 (optional)

– nailgun (*weapon_nailgun*) (same entity as CCTF)
– chaingun (*weapon_chaingun*)
– proximity launcher (*weapon_prox_launcher*)

– nails (*ammo_nails*) (same entity as CCTF)
– chaingun ammo (*ammo_belt*)
– prox mines (*ammo_mines*)

– Kamikaze thingy (*holdable_kamikaze*)
– Invulnerability thingy (*holdable_invulnerability*)

NOTE: If you use the nailgun and nails, they will appear in TA and CCTF, but not in Q3CTF, Make sure you use the *notta , notq3a and notcctf* to balance the weapons that will not appear in Q3A TA or CCTF. You can remove weapon spawnpads used for Team Arena weapons by making them a *func_static* and setting them to notq3a. The other option in this case would be to spawn a Q3A weapon there, and set the *notta* key for it.

You may need to get the TA.def file to add the TA entities to your level. This comes included with the TA version of GtkRadiant. You don't have to use GtkRadiant to add the entities to your map if you don't want to, simply append the TA entities to your own entity.def file.

Fit and Finish

There are a few standards of quality which you should adhere to before your levels are released.

– Flags must be set using the direction so they can be well seen when they aren't rotating. (See below)
– Bouncepads must have working circle-pulse effect (see below)
– Bouncepad decal's mush have a working shadow layer (see below)
– Weapon pad decals must have a working shadow layer (see below)
– Curves must be tested at 999 subdivisions as well as 4 (look for holes around them)
– Level should be checked for freak HOM's using *r_fastsky 0* and *r_clear 1* together to find them.
– Clip brushes wherever necessary
– All id Q3A standard limitations are to be followed. *meminfo*, imagelist, r_speeds,
– Curve T-Junctions (sparklies) need to be resolved. Use r_fastsky 0 and r_clear 1 together to find them.
– Player spawn points shouldn't face walls :)
– The 'Angle' must be set for all entities that don't rotate. See below for flags.

(You must be running in +set sv_cheats 1 in your shortcut, and using devmap to use r_showtris and r_clear 1)

ThreeWave replacement textures:

The levels should have a somewhat uniform look. We have some nice custom textures and decals to pull all the levels together. Grab the custom assets here and put them in your levels. A shader file is provided, and a sample.map and bsp. Don't forget to add

ctf_unified.shader and ctf_q3w.shader to your shaderlist.txt to see our assets in the editor. You don't have to 'overuse' the team logos and decals.

– Custom Blue and Red Bouncepad decals.
– Custom weapon spawn location decals.
– Red and Blue Base decals
– Base Area Plaques (similar to Q1)
– Base direction decals
– Odds and ends and steam

We have a custom decal for under weapon spawn points, similar to what is used in Team Arena. Place these under your weapons, and remove any 'ow' floors if you want. If you have a weapon that appears in TA and not in Q3A or vice-versa , make the brush that has the weapon spawn decal on it into a func_static, then set one of the following keys to a value of 1 : notta or notq3a

Flag direction angle

It is very likely that we will stop the flags from rotating and bobbing in CTF (as this is just really annoying). That being done, there must be an angle set for the flag so that it can be seen easily, and that the logo on the flag is facing the right way from that direction.

If you have placed the Overload entities in your level and you know that they are facing the right way, then you can set the flag entity based on the angle of the overload (obelisk) entity. Here's the angles:
Obelisk 0 (360) then flag = 270
Obelisk 90 then flag = 0 (360)
Obelisk 180 then flag = 90
Obelisk 270 then flag = 180

Looking at the flag as if you were attacking, set it so that if you were looking at the flag and facing say north, in the editor the pointer arrow sticking out of the flag points to your left or west. In oneflag, the orientation doesn't matter as long as the arrow faces left or right when approaching it. Meaning that you will see one side of the flag cloth no matter which base you approach it from.

Fixing bouncepads and weaponspawn pads

This method 'usually' works with multilayer shaders like bouncepads, bouncepad decals, and weaponspawn pads. Copying these items from one place to another in your level is usually the beginning of the problem. I'm not entirely sure why but often the first one is ok,

and the copied ones end up with problems, such as the bouncepad missing flash. I do know that these problems are more likely to occur if the face deviates from being very simple. Very simple being that the brush face is the exact same size as the texture, the brush face is aligned to the same grid of 128 as the texture (meaning the texture was applied and never needed to be aligned) the face is flat and parallel to the grid in all 3 dimensions, the texture was applied to that brush manually (as opposed to the brush being a copy of another brush with that texture on it.

Method A [usually works]
1) select the offending brush face
2) re-select the texture
3) re-align the texture as required using the arrow keys, or fit to face

Method B)
1) turn off texture move lock
2) select the offending brush face
3) re-select the texture
4) drag the entire brush around on the grid until the texture is aligned
5) turn texture move lock back on
6) move the brush back into position

Method C) [best for odd angled bounces]
1) replace the offending face with a simple patch mesh
2) re-select the texture
3) re-align the texture as required using the arrow keys, or fit to face

Method D) [best for odd angled bounces]
1) make the brush a func_static

Sadly, If none of the above methods work, you might be out of luck. If it's possible, try moving your bouncepad so that it lies on the 128 grid naturally, which sometimes can mean moving your entire level, or if it's an angled pad, try making it an even 45degree angle.

Aligning patch caps (This will also fix patch caps that the texture is scaled down tiny on)

A lot of levels don't have the patch caps aligned, and often they are scaled or distorted when in reality it's not that difficult to fix. Here's how to make patch caps blend perfectly with your surrounding curve and brush faces.

1) Zero (0,0) the horizontal and vertical alignment of the surrounding textures on normal brushes. Do this by selecting the texture from the texture window and reapply it to all faces. Or use the Bobtoolz plugin's amazing 'Reset Texture' feature to set all instances of that texture to 0,0 if it's a simple texture like concrete, pewter, rust etc.

2) Select the patch cap (if it's grouped you can use <SHIFT>+<ALT>+<LEFTCLICK> repeatedly to cycle until you get just one patch cap)
3) Re-select the texture from the texture window.
4) Re-cycle the patch cap <CTRL>+<SHIFT>+<N> until the texture is natural and blends with surrounding brushes.

target_location entities

All levels must have working target_location entities, and those entities must be tested so that they are all working as intended, but are not overused.

I would appreciate it if you could add the 'count' key to all target_locations with the corresponding color code depending on where in the level they are (middle=white=0 , red=1 , blue=4).
Key: count value: (0 or 1 or 4)

From the Radiant Manual:
Design Tips: The target locations are "line-of-sight." If a player can "see" it, and that target is the closest to the player, then that location message is displayed. Be conservative in placing location markers. Because of the way the location code is implemented, it causes a large amount of data to be transmitted regularly. It has been explained that the game server has to constantly keep track of, and update to the clients, a data table equal to all the location markers multiplied by all the players. Not only that, but it must calculate line of sight from players to location markers and calculate the distance from the location marker. Place the target locations in such a way that the entity can be "seen" from most, if not all, the positions that a player can stand in when it is inside that area. Fewer are better, even if it means that occasionally, a team mate is in an unknown location.

Note from Johnny Law Re Above:
The Radiant manual is incorrect about the way target_location entities work. If you examine the gamecode source you can see that Q3 just selects the closest target_location in the Potentially Visible Set. Line of sight is not involved. I just about pulled my hair out trying to get target_locations to work predictably before I discovered that.

Designing for the Grapple

Keep in mind when you're building that people using the grapple on your level will change the dynamics of play. You will need to ensure that you have clipped off appropriate areas so that grappling flag carriers can't run off and hide in places that are extremely difficult to reach to kill them. Another consideration is making some surfaces un-grappleable to prevent people from hanging out there. I use a special clip texture with 'surfaceparm noimpact' added to stop people from grappling. Pits of death, fog, and space maps are also a special case in light of the grapple and are described below.

Designing for the Grapple #2

In some cases you might want to remove the bouncepads in CCTF so that players are forced to use the grapple, or to get the bouncepad out of the way to make grappling more convenient. This can be accomplished by making the bouncepad out of a func_static and marking it with the *notcctf* key to make it dissapear.

Fog pits, pits of death, and space maps etc.

In CCTF and other grapple gametypes it is possible for people using the grapple to take advantage of the space between the top of the hazard and the and the *trigger_hurt* at the bottom which should kill them. Also, by grappling they could repeatedly play the falling.wav sound and disrupt play on the server. In order for maps that contain any of these types of hazards to work properly in CCTF and other grapple gametypes you must add an additional *trigger_hurt* filling the hazard from roughly 64 units under the surface of the fog to the bottom of the pit or space. This *trigger_hurt* should be set to *dmg* 10 , with No_protection and Slow enabled.

At the same time, you should set the *trigger_multiple* and *target_speaker to: key**: notcctf** value**: 1** so that they will be disabled in CCTF.*

Happy working bot support

All levels will have working bot support that has been tested for problem areas and such. The new q3map and bspc add enhanced bot support to Quake 3 and the 1.2x EXE's. You should be using the new tools to process the level and bot files. Currently as far as I know, the only way to get the new map compilers in binary is by installing the GTK Radiant TA beta. I have then copied *q3map.exe*, *bspc.exe*, *glib-1.3.dll*, *iconv-1.3.dll* to the place I usually keep q3map etc. Save copies of the old ones, as you may need them at some point. For some reason the GTK compilers require those dll's to be present in order to work.

Wrapping it All Up

Readme File

Name your readme.txt with your own name. i.e. g1zm0.txt

Credit **must** be given in your readme file to anybody who contributed assets such as textures, sounds, shaders, models, etc.

Custom Assets

Only include new textures, models, sounds, etc. if you are absolutely certain that the original creator intended them for redistribution. With textures made from altered or manipulated Quake 3 Arena textures there is no problem.

Shader file Uniformity

It's imperative that we keep things organized and tidy as far as shaders go. Use unique texture directory names that won't cause problems. If your level has been released in multiple versions in the past, you should rename the shader file, create a new and unique directory from which your textures and shaders will be loaded to prevent any conflicts with prior versions. During beta testing we'll have to do a little extra testing to make sure that we don't have shader loading/unloading problems on servers that switch between id maps and non id maps as is occasionally seen.

ThreeWave Legacy Textures

Legacy ThreeWave textures and shaders (from the old twpak0.pk3) **must not** be referenced in your map or shaders, the new assets pak includes the threewave textures in their new dir, and a new shader file for working with them where need be. Do not rely on the twpak0.pk3 for texture support. This includes all of these textures which appear as id textures in radiant. These have all been moved to the ctf_q3w directory, and the shaders which originally resided in the twpak0.pk3 ctf.shader have been rewritten and are included in the ctf_q3w.shader in the assets package

id mapmedia.pk3 Textures

If your level uses textures from the id mapmedia.pk3, I would highly recommend that you copy the specific textures that you use into your own unique texture directory and replace them with these in your level. Not many people have the mapmedia.pk3, and will not necessarily see them in your level.

Testing your level

When you believe that your level is complete, and ready to ship, the one most necessary step is to test your level in a .pk3 in an absolutely bare bones copy of Quake3. If your level is intended to be played by people who will already have the threewave paks, then you should only have id's pak0.pk3 – pak6.pk3, and the threewave paks that start with "Q3W" . Removal of the twpak0.pk3 is necessary to ensure that your level doesn't contain any of our legacy textures. Once you have your .pk3 in a clean install of Quake3, load your level up, and examine the console for any errors. If you see "Trying textures….." or other texture or shader related errors, these will need to be corrected.

Level Loading Screen Level Name + Author Name

We only use levelshots of a size of 128 x 128. The Threewave Portal system reies on images being that size, and may not give desired results with larger images. In order to recieve credit for your level, and be 100% compatable with the Threewave MOD, the following should be included in your .arena file:

author “Scancode”
quote “slurp, slurp, slurp” (This is a quote by you, such as a favorite saying, or brief comment about your level. This should not be too long!)