

progs_dump

A mapping devkit for Quake by dumptruck_ds
lango.lan.party@gmail.com
version 1.1.0 Release Candidate 3
2019-31-5

This is a Quake mod compilation designed for mappers. Some of this code is from 2019 other bits date back to 1997! Please read the credits section for more detailed info and links.

NOTE: If you use this DevKit, your project should be released as a stand-alone mod and installed into its own folder in the Quake directory.

To get started, unzip *mod_template.zip* into your Quake directory and rename the *my_mod* folder to the name of your mod. You should not include the *progs_dump* sample maps with your mod.

progs_dump is intended to give Quake mappers more creative options than “vanilla” Quake while retaining the look, feel and gameplay of the original. Mappers can use the custom *progs.dat* and a matching *.fgd* to add the following features to their mapping projects:

- **trigger spawned monsters**
- **custom sounds** (ambient and triggered)
- **custom models and sprites with animation playback**
- **multiple targets, targetnames and killtargets**
- **visual effects like explosions, lightning and particles**
- **enhanced trains** (spawnflag to start and stop via trigger)
- **enhanced triggers** (is_waiting flag to delay triggering)
- **enhanced plats** (toggle, start at bottom, etc.)
- **trigger_setgravity** (on player and / or monsters)
- **trigger_usekey** (trigger volume that requires a key to fire)

This progs also fixes the Rotfish monster kill count bug in addition to making Rotfish non-solid as soon as they are killed. Other minor bugs have been addressed as well. See the changelog for info.

There are sample maps included with this package that are meant to show what is possible with this project. See *progs_dump-1.1.0-RC3-README.txt* for more details.

Features marked **NEW** are new in 1.1.0 and **RC3** are additions for this release.

Acknowledgements

Thanks to the following people for their assistance and generosity. I could not have compiled this mod without their guidance either directly, through tutorials, mapping, code comments or forum posts:

Qmaster, RennyC, c0burn, ydrol, Preach, Joshua Skelton, Spike, Khreathor, Shamblernaut, ericw, metlslime, necros, negke, Baker, sock, G1ftmacher, NewHouse, Johnny Law, iJed, ionous, McLogenog, Danz and many others on func_msgboard.

You can inquire about progs_dump on the Quake Mapping Discord:

<https://discordapp.com/invite/j5xh8QT>

dumptruck's Quake videos:

<https://www.youtube.com/c/dumptruckds>

What's New in 1.1.0

1.1.0 Release Candidates 1 & 2

- * Added reset_items key to worldspawn entity; resets players items on map load (no need for Rune hacks for custom start maps!)
- * Added classnames to "bonus item fell out of level" and "monster stuck in wall" errors when developer is set to 1.
- * Add entities for Thunder, Water and Sky sounds. Use these when you want to hand place these instead of having VIS.exe add them automatically.
- * Fixed door unlock sound not playing (id software bug)
- * Added ladders
- * Added breakables
- * Added spawn_angry key/value for use with Trigger Spawned monsters
- * Added meat_shower (regular or large, triggerable, silent or random gib sounds)
- * Added trigger_void
- * Added trigger_shake (earthquake)
- * Renamed dtsounds.qc to dtmisc.qc
- * Added deadstuff entities
- * Added ritem (items respawn)
- * Added trigger spawning to items
- * Added items suspended flag
- * Fixed trigger_ladder conflict with trigger_setgravity
- * Added pain_target and pain_threshold
- * Added func_togglewall, func_counter, func_oncount and func_particlefield from Hipnotic
- * Added pd_ladders test level
- * Added pd_zombies and pd_meat maps
- * Added Persistent Key behavior (cnt = 1) to func_door and trigger_usekey
- * Added candle.mdl from Rogue and light_candle qc from Honey

1.1.0 Release Candidate 3

- * Fixed Megahealth respawning issue from RC1 / RC2
- * Fixed incorrect item_health bounding box from RC1 / RC2
- * Fixed Attenuation issue on custom sound entities
- * Added Khreathor's self.particle_offset to spawning items for better tfog
- * Added custom health to monsters (except zombies)
- * Added trigger_take_weapon (a.k.a. axe start)
- * Added weapon_shotgun pickup
- * Added RennyC's func_fall - trap brush that falls out from under the player
- * Added Hipnotic version of trap_spikeshooter (with lavaball and rockets)
- * Added Voreballs, Grenades to trap_spikeshooter
- * Added Rubicon2 "retriggerable" spawnflag and custom sounds to func_train
- * Added gibs to trap_spikeshooter
- * Added lightning trail entities from DOE (for traps and visual effects)

- * Added visual effect entities:
 - play_explosion
 - play_tbabyexplode
 - play_lavasplash
- * Added c0burn's switchable light styles
- * Added custom sound option to play_lavasplash
- * Added custom sound option to func_fall
- * Added Zerstrorer misc_particle_stream
- * Added silent option to func_rotate_door (sounds 4)
- * Added DONT_FADE spawnflag to func_fall
- * Added elec22k.wav for looped lightning sound effects
- * Added pd_lasers test level

Features

Trigger Spawned Monsters

Setting spawnflags to 8 allows monsters to be hidden in place until triggered. If you use the included .fgd file, the spawnflag *Trigger Spawn* is selectable on all monsters. This makes spawning monsters much easier than in vanilla Quake.

The *delay* key allows you to add a custom delay to each trigger spawn. Normally, multiple targets will spawn simultaneously. If you want to stagger the time each monster enters the map add a delay, use the drop down menu to select some predefined values or enter a custom value in seconds if you need a specific time set.

Setting the *wait* key to 1 will disable the teleport visual effects and sounds. In the included .fgd can use the drop down menu to select *Spawn Silently*.

NEW Set *spawn_angry* to 1 to have monsters immediately attack the player.

RC3 Monsters can have a custom *health* levels. Use this with care! You should warn the player with a message or some other communication if you change any of these dramatically. e.g. "These grunts seems pretty tough!"

pain_threshold & pain_target

NEW When a monster's health drops below it's *pain_threshold*, it's *pain_targets* are triggered. You can use this to call in reinforcements mid-battle or spawn items or other triggers when a monster reaches a certain level of health. You can also trigger things upon a monster's death as always. Default values for monster health have been added to the FGD for reference.

Multiple targets, targetnames and killtargets

Most entities can now trigger up to four separate targets at once (target, target2, target3 and target4). They can also have multiple targetnames (targetname, targetname2, targetname3 and targetname4). Mappers can also create setups with killtarget and killtarget2. In addition, mappers can use target and killtarget in the same entity. This is not possible in vanilla Quake.

Multiple triggers can be used in nearly any combination or order. For example: target3 can trigger targetname2 in a different entity.

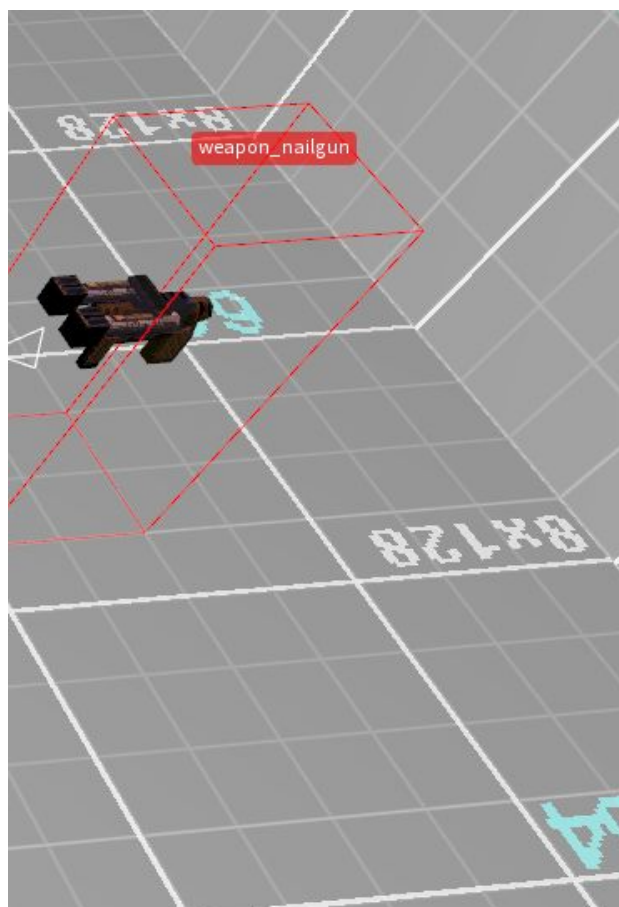
IMPORTANT: When using path corners or other similar entities, use the primary target and targetname fields for navigation only. The additional numbered fields may not function as expected in these cases. The Quoth mod has the same feature and the rule of thumb there applies here. As Preach states on the Quoth tutorial site: *A recommended structure is to use the original targetname field to give entities unique identifiers, and use the remaining fields for group triggers.*

Items

NEW Most items have enhanced capabilities in progs_dump. This includes ammo, weapons, keys and power-ups. Items can be suspended in mid-air via a spawnflag or trigger spawned just like monsters. Set a targetname for the item and select the *Trigger Spawned* spawnflag. Like monsters, they can spawn silently or with the “tfog” teleport visual and sound effects.

Items can also be set to respawn. Setting the *ritem* key value to 1 will cause the items to respawn. Items will respawn based on the default settings for a deathmatch game. You can set a custom respawn time using the *respawndelay* key and control how many times an items respawns with the *respawncount* key | value. By default items will display the “tfog” effects when respawning. You can mimic deathmatch respawns with the *Respawn DM Style* spawnflag. This skips the “tfog” effect and plays a more subtle sound effect.

In the example below, the nailgun is trigger spawned when the player presses a button targeting “t2”. After the player picks up the nailgun it will respawn in 45 seconds but only 3 more times.



Key	Value
classname	weapon_nailgun
origin	-400 -656 -0
respawndelay	45
spawnflags	64
targetname	t2
respawncount	3
ritem	1
delay	
killtarget	
killtarget2	
message	
target	
target2	
target3	
target4	
targetname2	
targetname3	
targetname4	

+ - ☒ Show default properties

<input type="checkbox"/> 1	<input type="checkbox"/> Not on Easy	<input type="checkbox"/> 65536
<input type="checkbox"/> 2	<input type="checkbox"/> Not on Normal	<input type="checkbox"/> 131072
<input type="checkbox"/> 4	<input type="checkbox"/> Not on Hard	<input type="checkbox"/> 262144
<input type="checkbox"/> 8	<input type="checkbox"/> Not in Deathmatch	<input type="checkbox"/> 524288
<input type="checkbox"/> 16	<input type="checkbox"/> 4096	<input type="checkbox"/> 1048576
<input type="checkbox"/> Spawn silent	<input type="checkbox"/> 8192	<input type="checkbox"/> 2097152
<input checked="" type="checkbox"/> Trigger Spawned	<input type="checkbox"/> Respawn DM style	<input type="checkbox"/> 4194304
<input type="checkbox"/> Suspended in air	<input type="checkbox"/> 32768	<input type="checkbox"/> 8388608

RC3 weapon_shotgun

This is a new entity that should be used when you want a player to spawn with only an axe. There are two models to choose from. Spawnflag 2 (the default) selects an unused model from Rubicon 2 by metlslime and spawnflag 4 is from Slapmap and has been used in a few mods.

Custom Sounds

play_sound_triggered

play_sound

ambient_general

IMPORTANT - New sound files used with these entities must be in the SOUND folder of your mod (or a sub folder under that SOUND folder.) There is no need to add “sound” in the “noise” path. (e.g. *boss2/sight.wav*) Most Quake source ports require a mono sound file for custom sounds. Do not use stereo files in your mod.

A note on the “speed” key (a.k.a attenuation factor) in sound entities. Attenuation in Quake means the reduction of a sound over a distance. Here’s a table of what the different speed keys mean in progs_dump.

Speed	QuakeC name	Attenuation effect
-1	ATTN_NONE	heard everywhere
1	ATTN_NORM	fades to zero at 1000 units
2	ATTN_IDLE	fades to zero at 512 units
3	ATTN_STATIC	fades to zero at 341 units

play_sound_triggered

Play a sound when triggered. Most of these key / value pairs can be left to their defaults. Can be looping or a “one off” sound.

NOTE: Looping sounds that are triggered ON will NOT play after the player loads a saved game. They will have to be triggered OFF then ON again. Also, you may encounter problems triggering sounds that are far away from the player. If you do, move the trigger closer.

- toggle (spawnflags): sound can be stopped and started when triggered
- volume: how loud (1 default full volume)
- noise: path of the sound to play (e.g. *blob/sight1.wav*)
- impulse: sound channel 0-7 (0 automatic is default)
- speed: attenuation factor (default recommended)

play_sound

Plays a “one off,” non-looped sound at a random interval. Like thunder or a monster sound.

IMPORTANT: Do NOT use looped sounds with this entity. For looped sounds see *ambient_general* below.

- volume: how loud (1 is default full volume)
- noise: path of the sound to play (e.g. *boss2/sight.wav*)
- wait: random time between sounds (default 20)
- delay: minimum delay between sounds (default 2)
- impulse: sound channel 0-7 (0 automatic is default)
- speed: attenuation factor

ambient_general

Plays a custom looped sound. Cannot be toggled off or triggered.

- noise: path of the sound to play (e.g. *ambience/suck1.wav*)

ambient_thunder

NEW Plays thunder sound at a random interval.

ambient_water1

NEW Swirling water sound effect. Usually this is added automatically to maps with water when you run VIS. If you want to place these in your map by hand, you can run VIS with the -noambient command line switch.

ambient_wind2

NEW Howling wind sound effect. Usually this is added automatically to outdoor sections of maps with sky textures. If you want to place these in your map by hand, you can run VIS with the -noambient command line switch.

Custom Models and Sprites

misc_model

A point entity for displaying models and sprites. A frame range can be given to animate the model.

- mdl: The model to display. Can be of type mdl, bsp, or spr.
- frame: Single frame to display. Can also be used to offset the animation.
- first_frame: The starting frame of the animation.
- last_frame: The last frame of the animation.
- speed: The frames per second of animation. Divide 1 by the fps for this value.
- angles: pitch roll yaw (up/down, angle, tilt left/right)

IMPORTANT: Set the angle value to 0 if using angles key to rotate mdls (see gib_ section below)

Enhanced Triggers

This mod has some enhancements to triggers that allow some to be started off or even toggled off and on. See the table below for more information.

trigger_changelevel

NEW On triggers that point to a hub or start map, the *Use info_player2_start* spawnflag will spawn the player on the info_player_start2 entity when the map changes. **You'll need an info_player_start2 on the map you are changing to!** Use this to skip skill selection when completing an episode as in the original game. Or you can return the player to a different part of a hub map.

trigger_push_custom

This can be used to create traps, jump pads, water currents and more.

If *Start Off* spawnflag is set the entity will not trigger until targeted. This can be targeted and toggled off and on. If the *Silent* spawnflag is set it won't make the standard "windfly" sound. Use *Custom Noise* spawnflag and the noise key/value together to use a custom push sound. Custom sounds should be "one off" sounds NOT looping sounds. A good way to simulate a water current is to have the trigger_push_custom under the surface of your water brush by about 32 units. You can see an example in the pd_gallery.map

trigger_monster_jump

If *Start Off* spawnflag is set the entity will not trigger until targeted. This can be targeted and toggled off and on. So monsters can be attacking from a distance and then be triggered to jump. **NOTE:** The way *trigger_monsterjump* works requires a monster to be "awake" and "angry" at the player before the jump is activated. Keep this in mind when using this new functionality.

RC3 trigger_take_weapon

This will remove the shotgun from the player's inventory and all shells. Place this over an `info_player_start` to have the player start with only the axe... or use this trigger to surprise the player in some devious way. Make sure and place a **weapon_shotgun** in your map for the player to get eventually!

trigger_setgravity

If *Start Off* spawnflag is set the entity will not trigger until targeted. This trigger changes the gravity on a player or monster that touches it. The trigger itself can be toggled on and off.

NOTE: the amount of gravity can only be changed by touching *another* `trigger_setgravity` with a different setting. The *gravity* key defaults to 0 which is normal gravity. Lower numbers equal lower gravity. Setting 100 is also normal gravity. Numbers above 100 will make the player "heavier", i.e. harder to jump.

trigger_shake

NEW Earthquake trigger - shakes players in it's radius when active. Strength of tremor is greatest at the centre.

dmg is strength at center (default is 120.) *wait* duration of shake (default is 1.) *count* effect radius (default is 200.) *noise* path of sound to play when starting to shake. *noise1* path of sound to play when stopping. *targetname* must be triggered. The *VIEWONLY* spawnflag shakes the view, but player movement is not affected.

trigger_usekey

Variable sized single use trigger that requires a key to trigger targets. Must be targeted at one or more entities. Use the *message* key to create a custom message for this. e.g. "Bring the Gold Key here mortal!" This trigger cannot start off or be toggled. **NEW** Setting *cmt* to 1 will not remove the key from the player's inventory, which mimic's the key behavior of Doom. Make sure and add this key | value to all doors and / or let the player know the default key behavior has changed. e.g. Perhaps a pickup message on the keys that reads: "This key works on many doors."

trigger_void

NEW Use this for a 'void' area. Removes monsters, gibbs, ammo, etc... also kills player. Spawnflags can be used to protect players or monsters.

is_waiting

If this value is set to 1, certain triggers will do nothing until another trigger activates it. The FGD provides a dropdown selection or you can enter the value by hand. The following table shows which triggers use *is_waiting* 1:

trigger	is_waiting (start off)
trigger_once	yes
trigger_multiple	yes
trigger_teleport*	yes (use targetname2)
trigger_changelevel	yes

*In order to use *is_waiting* on a trigger_teleport, make sure and use *targetname2* instead of *targetname*.

Enhanced Platforms

func_new_plat

This entity adds new capabilities to plats. It uses spawnflags to dramatically change the behavior of the entity. As with the standard plat, build your plat in the raised position so the entity will be lit correctly when you compile your map.

NOTE: You must use one of the following spawnflags with func_new_plat. Even though they use the same entity name, each spawnflag creates a very different plat.

Spawnflag 1: Setting the *Start at Top* spawnflag creates a plat that starts at the top and when triggered, goes down, waits, then comes back up. *health* = number of seconds to wait (default 5)

Spawnflag 2: Setting *Toggle Plat* creates a plat that will change between the top and bottom each time it is triggered.

NOTE: You must use the *height* key when *Toggle Plat* is used. Use a negative height number to start the plat off in a lower position.

Spawnflag 16: *Plat2* creates a plat in the bottom position, just like the standard plat. If a plat2 is the target of a trigger, it will be disabled in the lowered position until it has been triggered. *Delay* is the time before the plat returns to original position.

IMPORTANT: *Plat2* can be finicky so it's advised to create your plat the exact height you need it to travel (as opposed to having parts sticking into the ground or in hollow pockets below the plat for cosmetic reasons.) You can set a height to tweak the amount of lip needed. See *The Gallery* map for an example.

Misc Entities

trap_spikeshooter

trap_shooter

trap_shooter_switched

RC3 The original trap_spikeshooter shot only nails and lasers. All three of these entities can now shoot lavaballs, rockets, Voreballs, grenades or gibs. Set the spawnflag accordingly. Use the silent spawnflag if needed. Refer to the table below for specifics on how to trigger these.

Entity	Details
trap_spikeshooter	use a trigger_multiple to fire
trap_shooter	fires continuously (killtarget to stop)
trap_shooter_switched	toggle on and off with triggers, buttons

func_counter

NEW This is used to trigger things in a series. With this entity you can do some amazing new game play setups. Make sure and take some time to play with this one!

TOGGLE causes the counter to switch between an on and off state each time the counter is triggered. *LOOP* causes the counter to repeat infinitely. The count resets to zero after reaching the value in *count*. *STEP* causes the counter to only increment when triggered. Effectively, this turns the counter into a relay with counting abilities. *RESET* causes the counter to reset to 0 when restarted. *RANDOM* causes the counter to generate random values in the range 1 to *count* at the specified interval. *FINISHCOUNT* causes the counter to continue counting until it reaches *count* before shutting down even after being set to an off state. *START_ON* causes the counter to be on when the level starts. *count* specifies how many times to repeat the event. If *LOOP* is set, it specifies how high to count before resetting to zero. Default is 10. *wait* the length of time between each trigger event. Default is 1 second. *delay* how much time to wait before firing after being switched on. You can see *func_counter* in action when the sarcophagi burst open in pd_zombies.map and when used to animate the particle fields in pd_ladders.map.

func_oncount

NEW For use as the target of a *func_counter*. When the counter reaches the value set by *count*, *func_oncount* triggers its targets. *count* specifies the value to trigger on. Default is 1. *delay* how much time to wait before firing after being triggered. You can see *func_oncount* in action when the sarcophagi burst open in pd_zombies.map and when used to animate the particle fields in pd_ladders.map.

FireAmbient

This is a simple looping sound from the torches. Use this if you are using custom fire sprites or models.

func_door

NEW Setting *cnt* to 1 will not remove keys from the player's inventory, which mimics the key behavior of Doom. Make sure and add this key / value to all doors and let the player know the default key behavior has changed. e.g. Perhaps a pickup message on the key that reads: "This key works on many doors."

func_explobox

NEW An explosive brush entity. Works just like *misc_explobox* but is made from a brush you create as opposed to the default model.

func_fall

RC3 A brush that drops and fades away when touched and/or triggered. Add some spice to your jumping puzzles or other scripted sequences! Monsters will not trigger *func_fall* but will be gibbed if one falls on them. **NOTE:** When a *func_fall* brush touches another brush or entity it will stop, which can look odd in certain situations. *noise* = sound to play when triggered, the default is a switch sound. *wait* = wait this long before falling. Use the *DONT_FADE* spawnflag if desired.

func_togglewall

NEW Creates an invisible wall that can be toggled on and off. *START_OFF* spawnflag means the wall doesn't block until triggered. *noise* is the sound to play when wall is turned off. *noise1* is the sound to play when wall is blocking. *dmg* is the amount of damage to cause when touched. You can see an example of this in the *pd_ladders* example map above the water area.

func_train

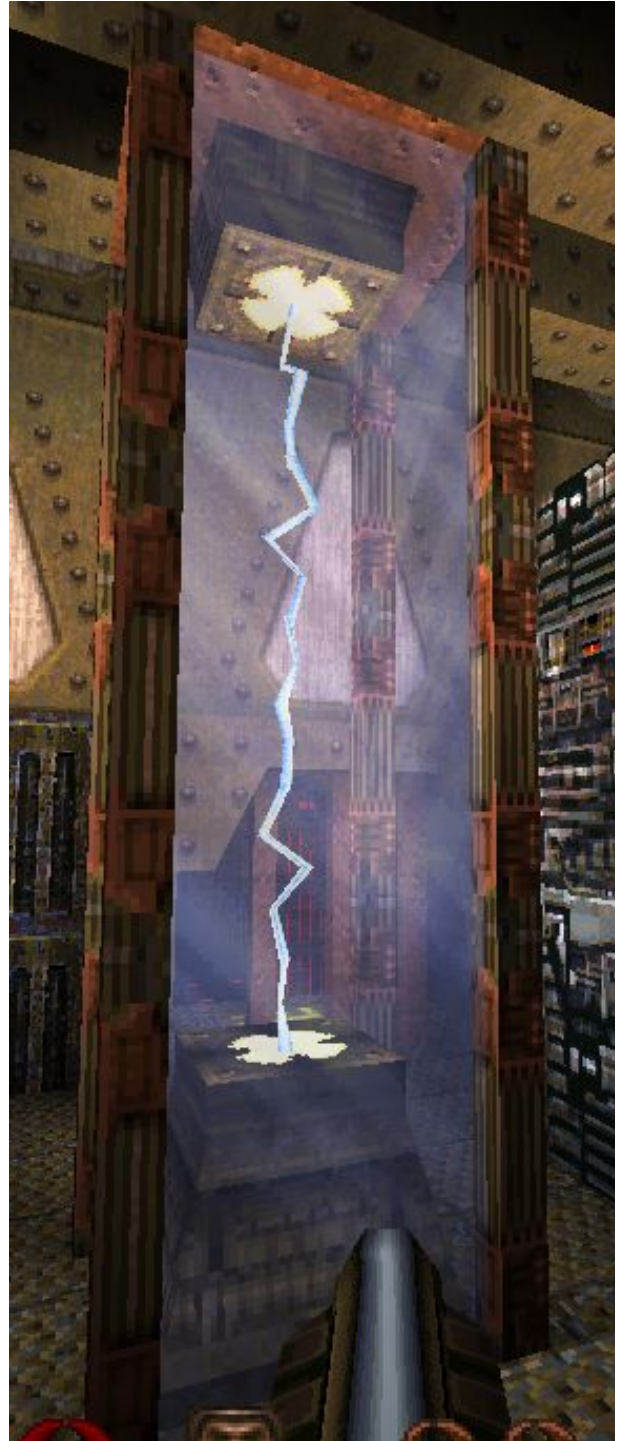
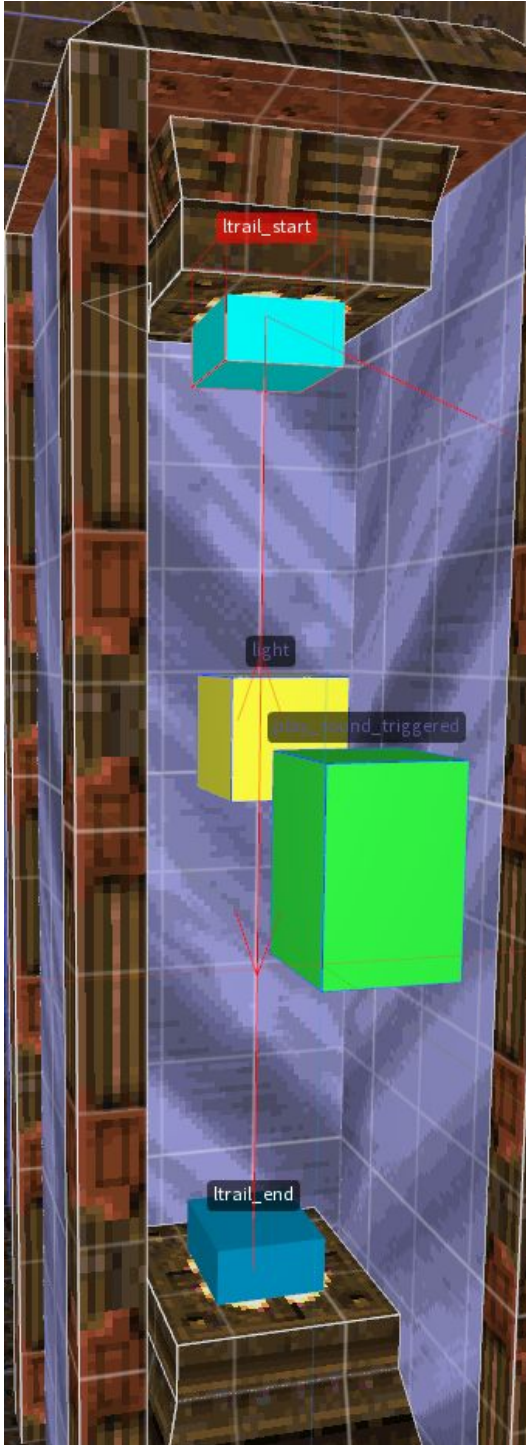
RC3 Just like the standard Quake train entity but with the *RETRIGGER* spawnflag set the train will stop at each path corner and wait to be retriggered before moving again. This will be great for more complicated lifts, doors and of course... trains. Set the *sounds* key to 3 to use custom sounds, then set *noise3* as the start/stop sound and *noise4* for the "in motion" sound.

misc_particle_stream

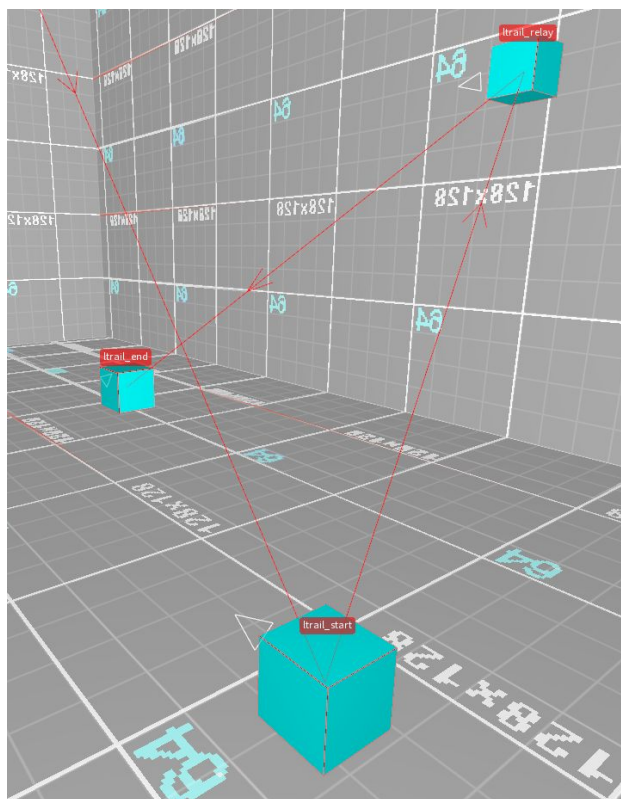
RC3 A particle stream! It appears when triggered. This entity is one end of the stream, target another entity as the other end-point. Usually an *info_notnull*, but you should be able to target anything (like monsters). *target* = This entity's origin is the end-point of the stream *dmg* = 1st Color, use this by itself if you want a single color stream *cnt* = 2nd Color, mixes particles of both colors. *noise* = Sound to play when triggered. Same color chart as *func_particlefield* below. You can see this in action in the *pd_lasers* sample map. FIXME

`ltrail_start`
`ltrail_relay`
`ltrail_end`

RC3 These lightning trail entities can be used for traps, decoration or for other scripted events. For the example below there are two entities. `ltrail_start` and `ltrail_end`, they are targeting each other.



If you want a chain of lightning events you would use a number of `ltrail_relays` between the start and end targeting one to the other, much like you would a `path_corner` with a `func_train`.



NOTE: The key / values are weirdly named in these entities. This is a quirk of QuakeC, where coders try and limit the amount of fields used by “recycling” unused fields to save memory.

ltrail_start Starting point of a lightning trail. Set *currentammo* to amount of damage you want the lightning to do. Default is 25. Set *frags* to amount of time before next item is triggered. Default is 0.3 seconds. Set *weapon* to amount of time to be firing the lightning. Default is 0.3 seconds. Set *sounds* to 1 for no sound. (Yes, it is weird.) Set the *TOGGLE* spawnflag if you want the lightning shooter to continuously fire until triggered again. Set the *START ON* spawnflag to have the lightning shooter start on. Do NOT use both these spawnflags at once.

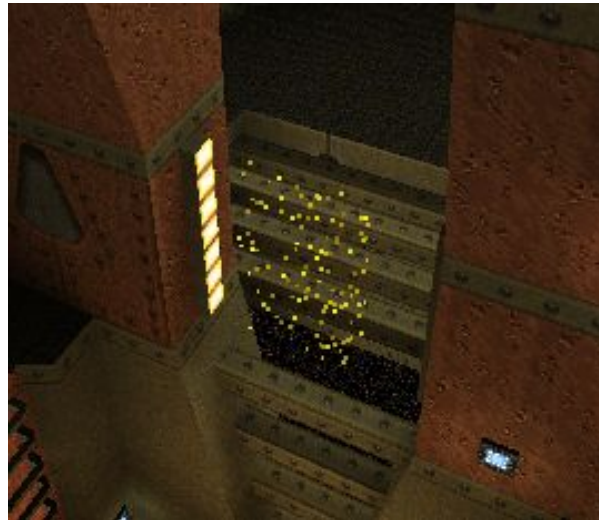
ltrail_relay Relay point of a lightning trail. Set *currentammo* to amount of damage you want the lightning to do. Default is 25. Set *frags* to amount of time before next item is triggered. Default is 0.3 seconds. Set *weapon* to amount of time to be firing the lightning. Default is 0.3 seconds. Unfortunately, `ltrail_relay` entities cannot be set to silent.

ltrail_end Ending point of a lightning trail. Does not fire any lightning. Set *frags* to amount of time before next item is triggered. Default is 0.3 seconds.

NOTE: To have a continuously firing bolt between two points, have a `ltrail_start` and `ltrail_end` targeting each other in a loop and set *frags* to -1. The sound this makes is not ideal, so consider making these silent and use a `play_sound_triggered` with a custom looping sound. This is shown in the `pd_lasers` sample map. In the devkit, `sounds/dump/elec22k.wav` is included for this very reason.

func_particlefield

NEW Creates a brief particle flash roughly the size of the defining brush each time it is triggered. You can see an example of this in the `pd_ladders` example map. In this case, the particle fields are animated in sequence to create a force field effect. *USE_COUNT* when the activator is a `func_counter`, the field will only activate when count is equal to `cnt`. Same as using a `func_oncount` to trigger. *cnt* is the count to activate on when *USE_COUNT* is set. *color* is the color of the particles. Default is 192 (yellow). *count* is the density of the particles. Default is 2. *noise* is the sound to play when triggered. Do not use a looping sound here. *dmg* is the amount of damage to cause when touched.



If you want to use another color for the particle field, refer to the Quake color palette below

NOTE: not all colors will work:

White (0)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Brown (1)	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Light blue (2)	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Green (3)	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Red (4)	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
Orange (5)	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
Gold (6)	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
Peach (7)	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
Purple (8)	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
Magenta (9)	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
Tan (10)	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
Light green (11)	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
Yellow (12)	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
Blue (13)	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
Fire (14)	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
Brights (15)	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

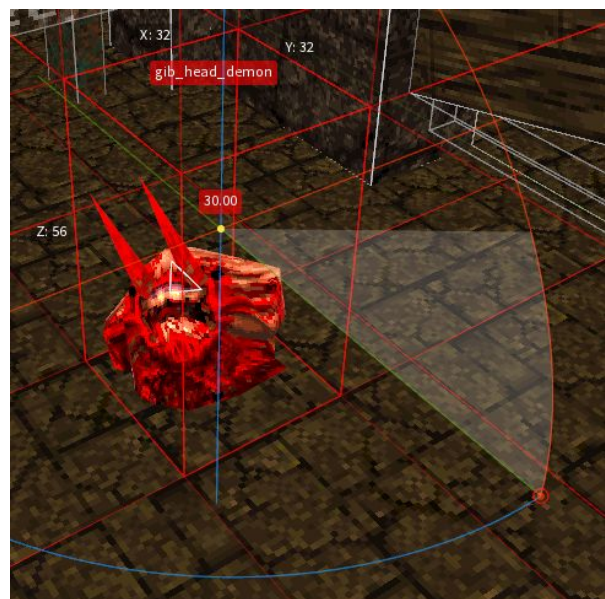
`gib_(classname)`

NEW e.g. `gib_head_player` Easily add these bloody decorations to your map. (Also see `monster_dead_(classname)` below. You can use the SOLID spawnflag to make enable collision on the model but clip brushes will work even better.



If you are using TrenchBroom take extra care when rotating these entities. The way TrenchBroom handles rotations for custom models requires a small work around in some cases. If you want to simply rotate the gib model around the z axis there is no problem. However, if you wish to rotate the model in the X and Y or any combination, you will need to manually type in X Y and Z values *before* using the rotate tool. To do this, use the *angles* key (with an s) and type in something like 0 45 0 as the values. Then you can select the rotate tool and adjust the other values using the widget. Keep in mind the values 0 0 0 will not work. Also the *angle* key (no s) should be blank or set to 0 when using the *angles* key.

Key	
classname	<code>gib_head_demon</code>
origin	<code>97 -163 1</code>
angles	<code>0 45 0</code>
spawnflags	<code>0</code>
targetname	



light_candle

NEW A simple light emitting candle from Mission Pack 2. You can place them into the ground for shorter varieties.

meat_shower

NEW See an example of *meat_shower* used with a *func_counter* in *pd_meat.map* When triggered this entity will spawn a shower of gibs. *style* = 0 is regular gib effect, 1 is more violent *fly_sound* = 0 is silent, 1 plays randomized gib sounds *targetname* = Must be triggered

monster_dead_(classname)

NEW e.g. *monster_dead_ogre* More decorations for your maps. You can use the SOLID spawnflag to make enable collision on the model but clip brushes will work even better. Keep in mind the same issue with rotation mentioned above applies to these models as well.



tele_fog

When triggered, *tele_fog* plays the teleport particle effects and sound. Use this when killtargeting an entity if the player can see it happen. You can see an example of this near the *trigger_use* key entity in *The Gallery* map.

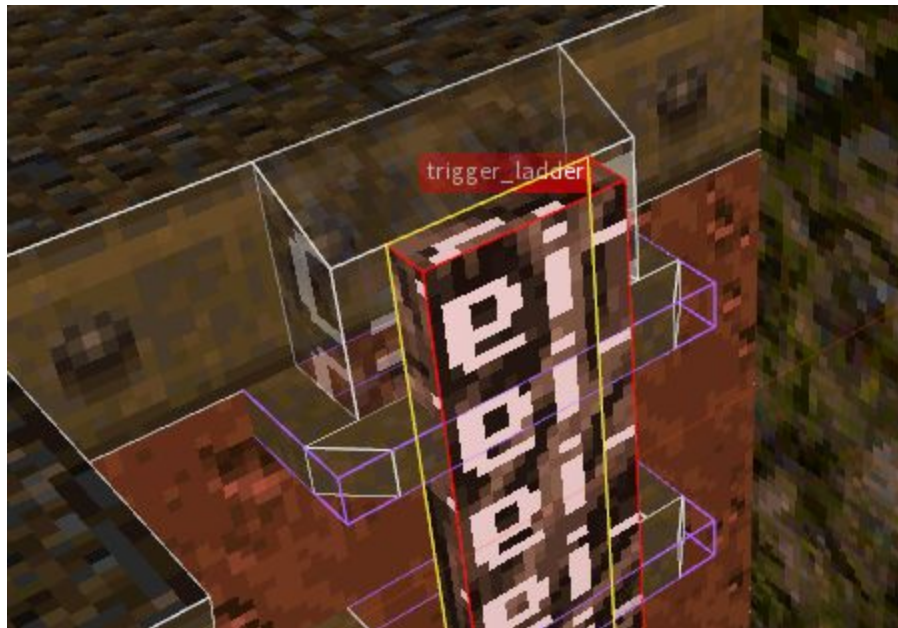
Worldspawn

NEW Added a *reset_items* key. Set to 1 to make the player start with default shotgun and axe.

Ladders

trigger_ladder

NEW IN 1.1.0 Create a small *trigger_ladder* brush covered with the trigger texture. Make sure the outside edge of the brush is flush with your ladder geometry. Set the *angle* key to the direction the player is facing when approaching the ladder. You can use a wedge shaped clip brush to smooth out any “sticky” movements at the top of the ladder as seen below. Please refer [pd_ladders.map](#) for examples.



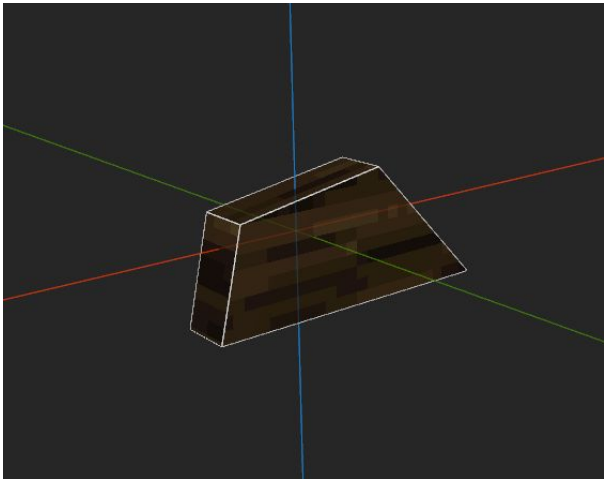
Breakables

func_breakable

NEW Breakables may seem overwhelming to new mappers, however it's not as complicated as it looks. Also, there are two methods to choose from. One is the *Built-in* (easy) method and the other is the *Custom* method (more flexible.)

The Built-in method: Create your brush and make it a *func_breakable*. You can ignore any keys that begin with *brk* or *breakable*. Those are used with the custom method. With the built-in method you will set the *style* to one of three options: 0 = Green Metal, 1 = Red Metal and 2 = Concrete. By default, the breakable will spawn 5 pieces of debris. You can change this amount with the *cnt* key/value. The default *health* of the brush is 20. There is a placeholder sound but you can use the *noise1* key to set a custom sound path. If you give the breakable a *targetname* it will only break when triggered. Use the *Explosion* spawnflag for an explosive brush. Use the *dmg* key to set a custom damage value. You can also use the *No Monster Damage* spawnflag to keep monsters from breaking the brush. I hope to have more “skins” than the default three for the final release of 1.1.0

The Custom Method: This method uses external, custom models (.mdl format) or brush models (.bsp format) instead of the built-in system. You can make small pieces of debris by shaping them in a level editor and compiling them into a .bsp (See *Creating Debris* below.)



You can also use .bsps from other mods (check if you have permission to do so.) In the example below, we are only using one piece and duplicating it when the brush is “broken.” Set the Use custom mdl's or bsp models spawnflag to enable this mode. Then set the path to the .bsp or model in *break_template1*. The *brk_obj_count1* determines how many instances of that bsp will be used. You can have 5 different pieces of debris total (*break_template1-5*) and control how many instances each of those templates spawns with *brk_obj_count1-5*. *noise1* is the path to the sound when breaking. *Style* and *cnt* are not used in this method but *health* and *dmg* are.

Key	Value
classname	func_breakable
break_template1	maps/debris/wood1.bsp
brk_obj_count1	5
spawnflags	4
break_template2	
break_template3	
break_template4	
break_template5	
brk_obj_count2	
brk_obj_count3	
brk_obj_count4	
brk_obj_count5	
cnt	5
dmg	20
health	20
+ - <input checked="" type="checkbox"/> Show default properties	
<input type="checkbox"/> No Monster Damage	<input type="checkbox"/> Not on Easy
<input type="checkbox"/> Explosion	<input type="checkbox"/> Not on Normal
<input checked="" type="checkbox"/> Use custom mdl's or bsp models	<input type="checkbox"/> Not on Hard
<input type="checkbox"/> 8	<input type="checkbox"/> Not in Deathmatch
<input type="checkbox"/> 16	<input type="checkbox"/> 4096
<input type="checkbox"/> 32	<input type="checkbox"/> 8192
<input type="checkbox"/> 64	<input type="checkbox"/> 16384
<input type="checkbox"/> 128	<input type="checkbox"/> 32768

Creating debris

You can create *break_templates* as tiny maps and compile them into bsps. Create one piece at a time as their own map file. Create the debris at the center of the map (origin 0, 0, 0) Compile with qbsp.exe and light. No need to run vis.exe on these. You can add a *light* key/value to the Worldspawn to uniformly light the piece of debris.

Key	
classname	worldspawn
wad	D:/QuakeDev/wads/tir
light	175
_tb_def	external:D:/QuakeC/pr
_sun_manqle	

Place these pieces in your maps folder or a subfolder under maps called debris or breakables and remember to include these when you distribute your map.

Enhanced Zombies

NEW Zombies have more options in progs_dump 1.1.0. First off, there is a motionless and silent version of the crucified decorative zombie. You can also spawn a *sleeping* zombie that will not awaken until triggered. You must target these zombies if the *Spawn Sleeping* spawnflag is set. If you trigger spawn a sleeping zombie into a map you will have to target them a second time to “wake” them up. You can see examples of the new features in the pd_zombies sample map. Spawnflag examples:

<input type="checkbox"/>	Crucified
<input type="checkbox"/>	Ambush
<input type="checkbox"/>	Crucified motionless
<input checked="" type="checkbox"/>	Trigger Spawn
<input checked="" type="checkbox"/>	Spawn Sleeping
<input type="checkbox"/>	32
<input type="checkbox"/>	64
<input type="checkbox"/>	128

Effect Entities

In addition to the **tele_fog** and **meat_shower** effects you can now trigger the following visual effects.

Effect	Details
play_explosion	grenade explosion, causes damage
play_tbabyexplode	Spawn death explosion, causes damage
play_lavalsplash	large particle effect, can have custom sound

Switchable Animated Light Styles

RC3 Normally, if you apply a style to a light (e.g. candle flicker, strobe) those cannot be triggered on and off. Progs_dump now has this feature, borrowed from c0burn's in progress *Slipgate* mod. Just choose a *style2* selection from the dropdown and target the light as normal. Use the *START OFF* spawnflag if needed.

Select the *FADE IN / OUT* spawnflag for a beautiful fade in / out effect on normal lights.

NOTE: Fades will not work on animated lights (e.g. style or style2).

spawnflags	1
style2	2
_anglescale	0.5
_bouncescale	1

+ - ☒ Show default properties

Select a choice option:

0 : Normal

1 : Flicker A

2 : Slow, strong pulse

3 : Candle A

4 : Fast strobe

6 : Flicker B

5 : Gentle pulse

7 : Candle B

8 : Candle C

9 : Slow strobe

10 : Fluorescent flicker

11 : Slow pulse, noblack

12 : Blink on/off

+ - <input checked="" type="checkbox"/> Show default properties		
<input checked="" type="checkbox"/> Start off	<input type="checkbox"/> 256	<input type="checkbox"/> 65536
<input type="checkbox"/> Fade in/out	<input type="checkbox"/> 512	<input type="checkbox"/> 131072
<input type="checkbox"/> 4	<input type="checkbox"/> 1024	<input type="checkbox"/> 262144
<input type="checkbox"/> 8	<input type="checkbox"/> 2048	<input type="checkbox"/> 524288
<input type="checkbox"/> 16	<input type="checkbox"/> 4096	<input type="checkbox"/> 1048576
<input type="checkbox"/> 32	<input type="checkbox"/> 8192	<input type="checkbox"/> 2097152
<input type="checkbox"/> 64	<input type="checkbox"/> 16384	<input type="checkbox"/> 4194304
<input type="checkbox"/> 128	<input type="checkbox"/> 32768	<input type="checkbox"/> 8388608

Rotation Entities

NEW By request, the Hipnotic (Quake Mission Pack 1) entities have been added but are **unsupported**. They should work, but are untested, so use at your own risk! The text below is taken from the Quake C code for the rotation system. Refer to the included sample map for examples. Enjoy!

func_rotate_entity

Creates an entity that continually rotates. Can be toggled on and off if targeted.

TOGGLE = allows the rotation to be toggled on/off

START_ON = whether the entity is spinning when spawned. If TOGGLE is 0, entity can be turned on, but not off.

If "deathtype" is set with a string, this is the message that will appear when a player is killed by the train.

"rotate" is the rate to rotate.

"target" is the center of rotation.

"speed" is how long the entity takes to go from standing still to full speed and vice-versa.

path_rotate (Train with rotation functionality)

Path for rotate_train.

ROTATION tells train to rotate at rate specified by "rotate". Use '0 0 0' to stop rotation.

ANGLES tells train to rotate to the angles specified by "angles" while traveling to this path_rotate. Use values < 0 or > 360 to guarantee that it turns in a certain direction. Having this flag set automatically clears any rotation.

STOP tells the train to stop and wait to be retriggered.

NO_ROTATE tells the train to stop rotating when waiting to be triggered.

DAMAGE tells the train to cause damage based on "dmg".

MOVETIME tells the train to interpret "speed" as the length of time to take moving from one corner to another.

SET_DAMAGE tells the train to set all targets damage to "dmg"

"noise" contains the name of the sound to play when train stops.

"noise1" contains the name of the sound to play when train moves.

"event" is a target to trigger when train arrives at path_rotate.

func_rotate_train

In path_rotate, set speed to be the new speed of the train after it reaches the path change. If speed is -1, the train will warp directly to the next path change after the specified wait time. If MOVETIME is set on the path_rotate, the train interprets "speed" as the length of time to take moving from one corner to another.

"noise" contains the name of the sound to play when train stops.

"noise1" contains the name of the sound to play when train moves.

Both "noise" and "noise1" defaults depend upon "sounds" variable and can be overridden by the "noise" and "noise1" variable in path_rotate.

Also in path_rotate, if STOP is set, the train will wait until it is retriggered before moving on to the next goal.

Trains are moving platforms that players can ride. "path" specifies the first path_rotate and is the starting position. If the train is the target of a button or trigger, it will not begin moving until activated. The func_rotate_train entity is the center of rotation of all objects targeted by it.

If "deathtype" is set with a string, this is the message that will appear when a player is killed by the train.

speed	default 100
dmg	default 0
sounds	
1 =	ratchet metal

func_movewall

Used to emulate collision on rotating objects.

VISIBLE causes brush to be displayed.

TOUCH specifies whether to cause damage when touched by player.

NONBLOCKING makes the brush non-solid. This is useless if VISIBLE is set.

"dmg" specifies the damage to cause when touched or blocked.

rotate_object

This defines an object to be rotated. Used as the target of func_rotate_door.

func_rotate_door

Creates a door that rotates between two positions around a point of rotation each time it's triggered.

STAYOPEN tells the door to reopen after closing. This prevents a trigger-once door from closing again when it's blocked.

"dmg" specifies the damage to cause when blocked. Defaults to 2. Negative numbers indicate no damage.

"speed" specifies how the time it takes to rotate

"sounds"

1 = medieval (default)

2 = metal

3 = base

Credits

QuakeC

misc_model.qc, math.qc by Joshua Skelton

<https://gist.github.com/joshuaskelly/15fe10fbaaa1bf87b341cba6e3ad2ebc>

Trigger Spawned Monsters added via Preach's excellent tutorial:

<https://tomeofpreach.wordpress.com/2017/10/08/teleporting-monsters-flag/>

custents by Carl Glave

<http://www.quaketastic.com/files/tools/windows/quakec/custents.zip>

various .qc from Hipnotic's Quake Mission Pack Scourge of Armagon

Original Code written by Jim Dose and Mark Dochtermann

http://www.quaketastic.com/files/tools/windows/quakec/soa_all.zip

various .qc from Rogue's Quake Mission Pack Dissolution of Eternity

http://www.quaketastic.com/files/tools/windows/quakec/doe_qc.zip

Preach's clean Quake 1.06 source courtesy of Johnny Law

https://github.com/neogeographica/quakec/tree/1.06_Preach

various .qc from Rubicon Rumble Pack Devkit by ijed / Louis

http://www.quaketastic.com/files/single_player/mods/RRP_DEVKIT.zip

Arcane Dimensions code by Simon O'Callaghan

<http://www.simonoc.com/pages/design/sp/ad.htm>

Honey source by czg

<https://www.quaddicted.com/reviews/honey.html>

Zerstörer QuakeC Development Kit - Dave 'Ace_Dave' Weiden and Darin McNeil

<https://www.quaddicted.com/reviews/zer.html>

Rubicon 2 code copyright 2011 John Fitzgibbons.

<https://www.quaddicted.com/reviews/rubicon2.html>

deadstuff version 1.0 - Tony Collen

ftp://archives.gamers.org/pub/idgames2/quakec/level_enhancements/deadstuf.zip

Remake Quake code by Supa, ijed and (?)

<https://icculus.org/projects/remakequake/>

Slipgate by Michael Coburn

<https://github.com/c0burn/Slipgate>

Code snippets and examples from RennyC, Khreathor, Spike and c0burn.

Sample Maps

Ineffable Crown of Darkness

pd_ionous by Ionous

Eigenstate

pd_gravity by Ionous

voice.of.the.nephilim@gmail.com

@voiceovnephilim

Magic River

pd_yoder by Yoder

AndrewYoder@live.com

@Mclogenog

Hello World

pd_old_start by dumptruck_ds

The Gallery

pd_gallery by dumptruck_ds

lango.lan.party@gmail.com

Twitter: @david_spell

Go Map!