

Swapping Spawns in Day of Defeat Source by INsane www.dodbits.com

Description: To make a map and have a attack team (allies) and a defending team (axis).

- If the Allies have a successful cap-out of a round the spawns will swap.
- If the Axis cap out (defend) they will remain as the defenders and the spawns will not swap
- Whatever team wins (cap-out) the round... they will become the defending team

Entities used:

- env_global
- logic_auto
- filter_activator_team
- info_teleport_destination
- trigger_teleport

The **other entities** used are the normal ones you will **use in a dod map (flag cap)** ...

- info_player_allies
- info_player_axis
- dod_control_point_master
- dod_capture_area
- dod_control_point

You should know how to set up a normal dod map using the above entities before you attempt to understand the rest of this Tutorial.

I will not place down all the inputs and outputs, you should read this and refer to the input and output tabs in Hammer.

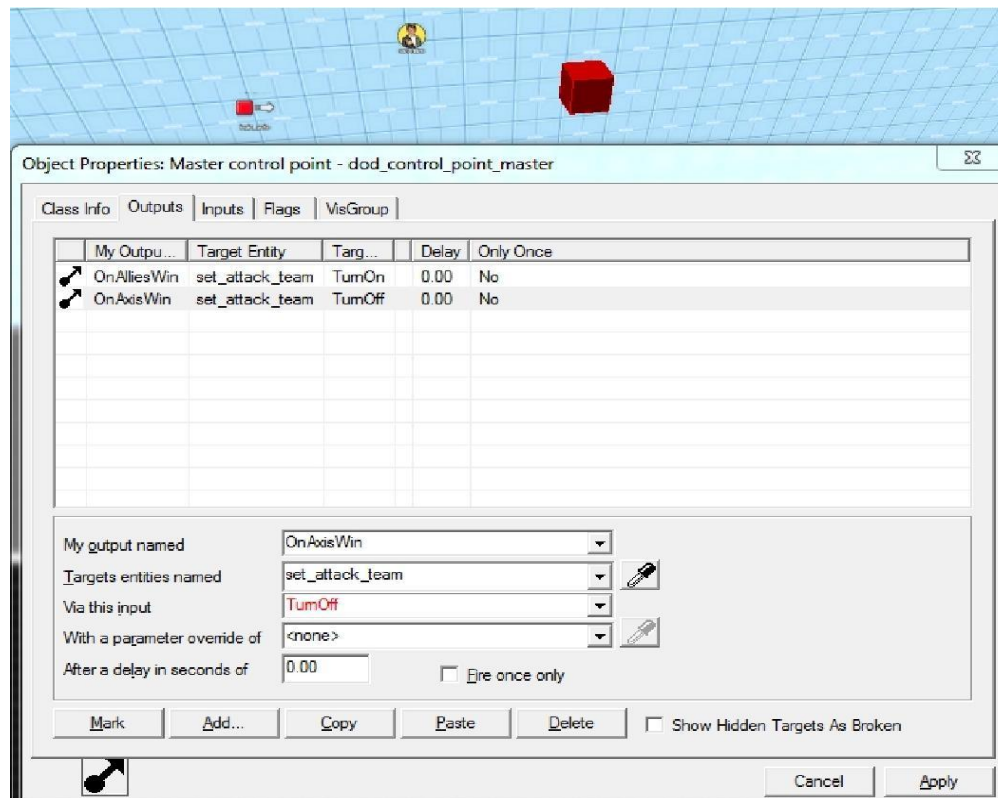
How it works: The main problem is how to set the spawns after a round, this is handled by the env_global entity. *Note that when using the standard FGD for dods this entity will be shown as "Obsolete" this is OK, it will still work and you may see some items in red text in the "Via this input" box, this is also OK. A custom FGD will show this correctly but for this tutorial we are using the standard FGD...* <http://developer.valvesoftware.com/wiki/Fgd>

- Name your env_global , in the key "targetname" call the value "set_attack_team"
- Make a custom value in the key "globalstate"... call the value "swap_spawns"
- Set the Key "initialstate" to "0" (OFF)

Now all that has done is given us a entity that will remember the value "swap_spawns" and hold it into the next round.

We now need to have something set the value "swap_spawns" value in the first round ... and we also need a entity to read that "swap_spawns" value at the next round and fire some outputs.

To set the "swap_spawns" is a matter of using the dod_control_point_master entity outputs...

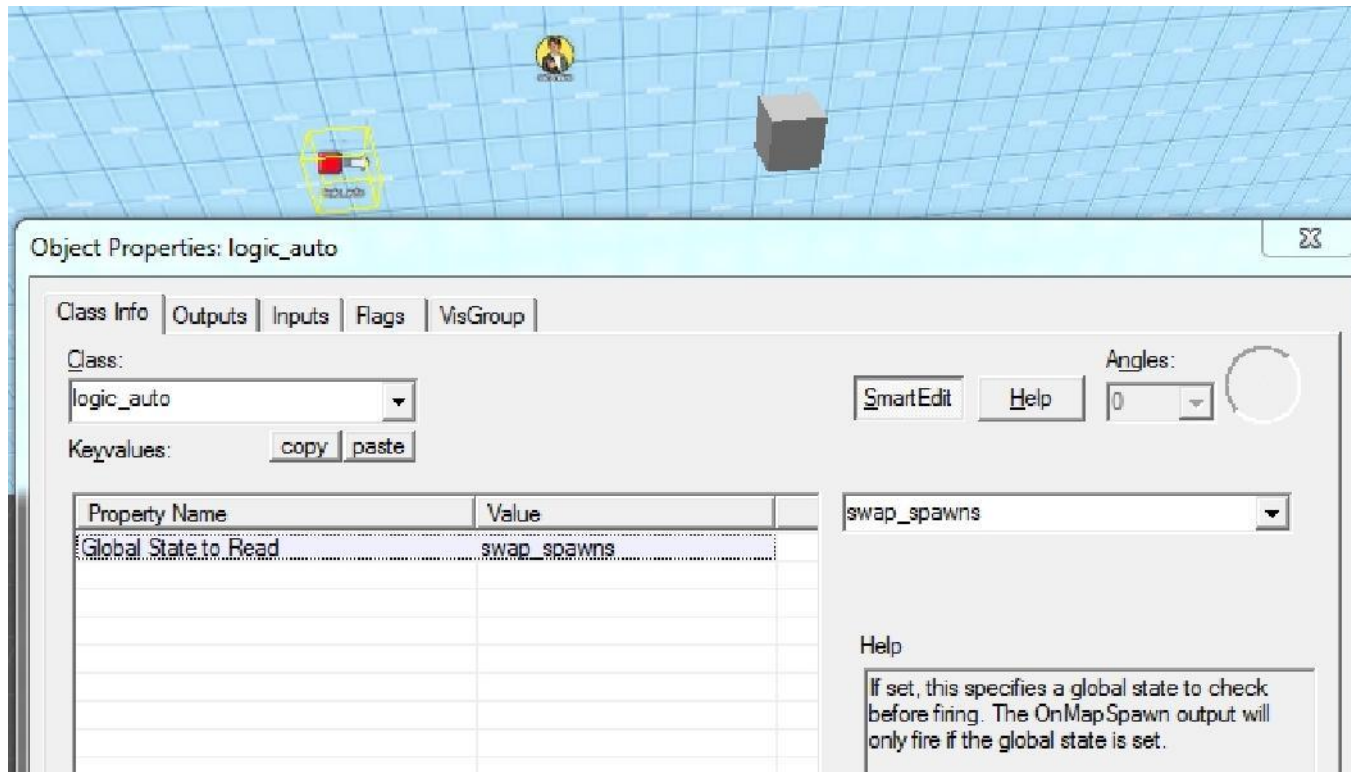


We are telling that env_global entity to remember if ...

- The axis won to set the "swap_spawns" value to OFF (0)
- The Allies won to set the "swap_spawns" value to ON (1)

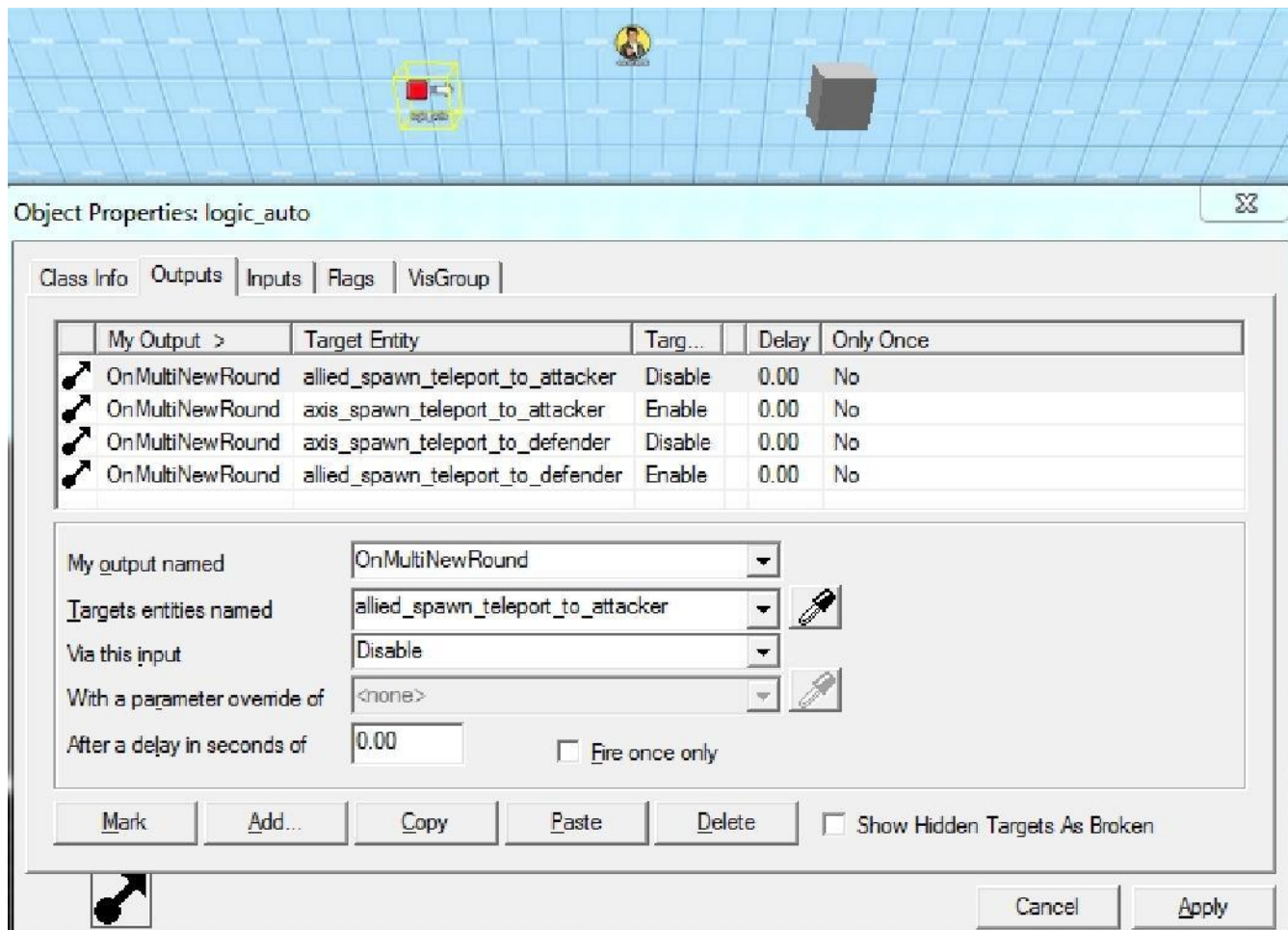
Now we have a simple switch that remembers who won and it will stay that way into the next round.

- We now need to read that information and do something with it, the `logic_auto` entity will be looking for this value "swap_spawns"



The logic_auto entity will read whether it is set at ON "1" or OFF "0"

We can now control some stuff in the new round via the outputs of the logic_auto entity, in this case we are going to teleport the spawns if the value "swap_spawns" is set to ON "1"



That is the first hurdle over, we have some **outputs firing to enable and disable other entities** that control what team spawns where **after a round win**. We now have to work on the entities that receive this information and are affected by it.

- filter_activator_team
- info_teleport_destination
- trigger_teleport
- info_player_allies
- info_player_axis

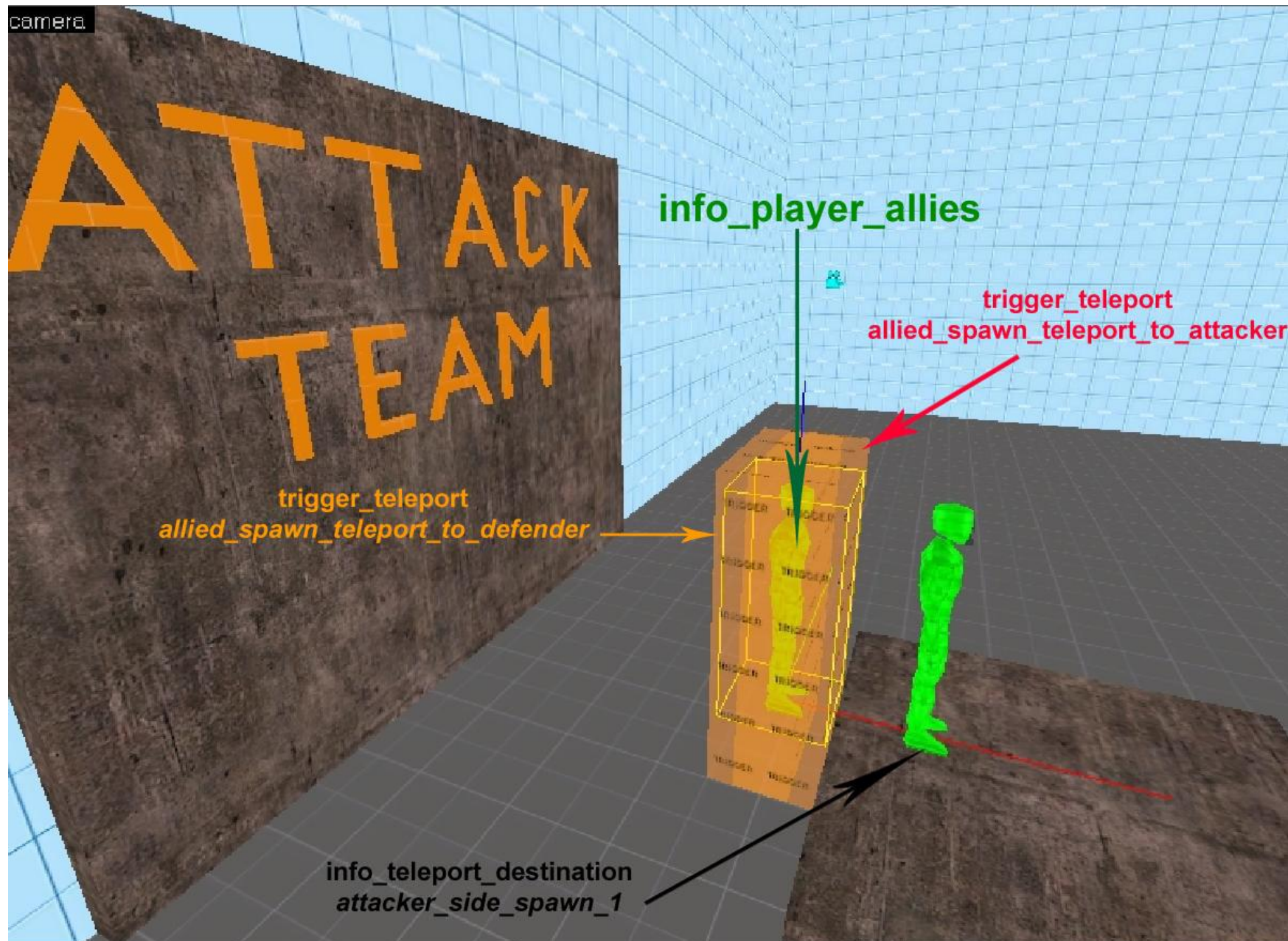
These five entities will control the teams spawn location based on who just won the round. Start with a normal set of info_player_allies and info_player_axis and make two trigger_teleport areas for each spawn point. Now these do not have to be in the spawn area, what matters in the end is the info_teleport_destination is in the spawn areas.

You should make a 16 per side for a real map, it is OK to keep the two names per spawn point the same for all one sides spawn points, however do not make all the info_teleport_destination entities the same name, you will *have problems with stuck players spawning in the same destination*, for now we will just work with a single set per side.

Look at the picture on the next page, you can see that there are two teleports over the allies spawn point .

The trigger_teleport called "allied_spawn_teleport_to_defender" is enabled or disabled by the logic_auto outputs, this is the final act of the swap. This is why you see two trigger_teleport entities per spawn point they go to different places and need to be enabled and disabled.

There is also one more important entity to ensure the allies and axis don't end up in the same spawn after being swapped... that would be fun but bad. more on this later, but for now look at the attack side (allies at the start) and see how these look in hammer.



To ensure the allies and axis don't end up in the same spawn after being swapped.. you will note in the tutorial VMF there are two filter_activator_team entities, one for allies and one for axis. What these will do is control what team can use a particular trigger_teleport.

After the allies are swapped it is possible for the allies to step in to the trigger_teleport on the Defenders side named axis_spawn_teleport_to_defender. But if we apply a filter so only the axis can use that teleport... problem solved. Look at each trigger_teleport, you will note that the **"Filter Name" is filled in with the name of the filter activator team entities that allow only that team to use the teleport.**

So to recap, we have a way to remember a custom value at the start of a new round, if it has been triggered, a logic_auto can control some teleports and deliver the players to the correct spawns based on who won in the last round. What's next?

Ok we only have one spawn point per side! We may need 16... it is not that bad, don't worry. We can leave the control side as it is, the spawn points and the teleports can be copied 16 times per side... but we still have some edits and this is where you may decide to make a special spawn room that is not a part of the real map players see.

Just copy the spawns points and each of their teleports 16 times for each team. Copy Attack and defend side teleport destinations 16 times for each side.

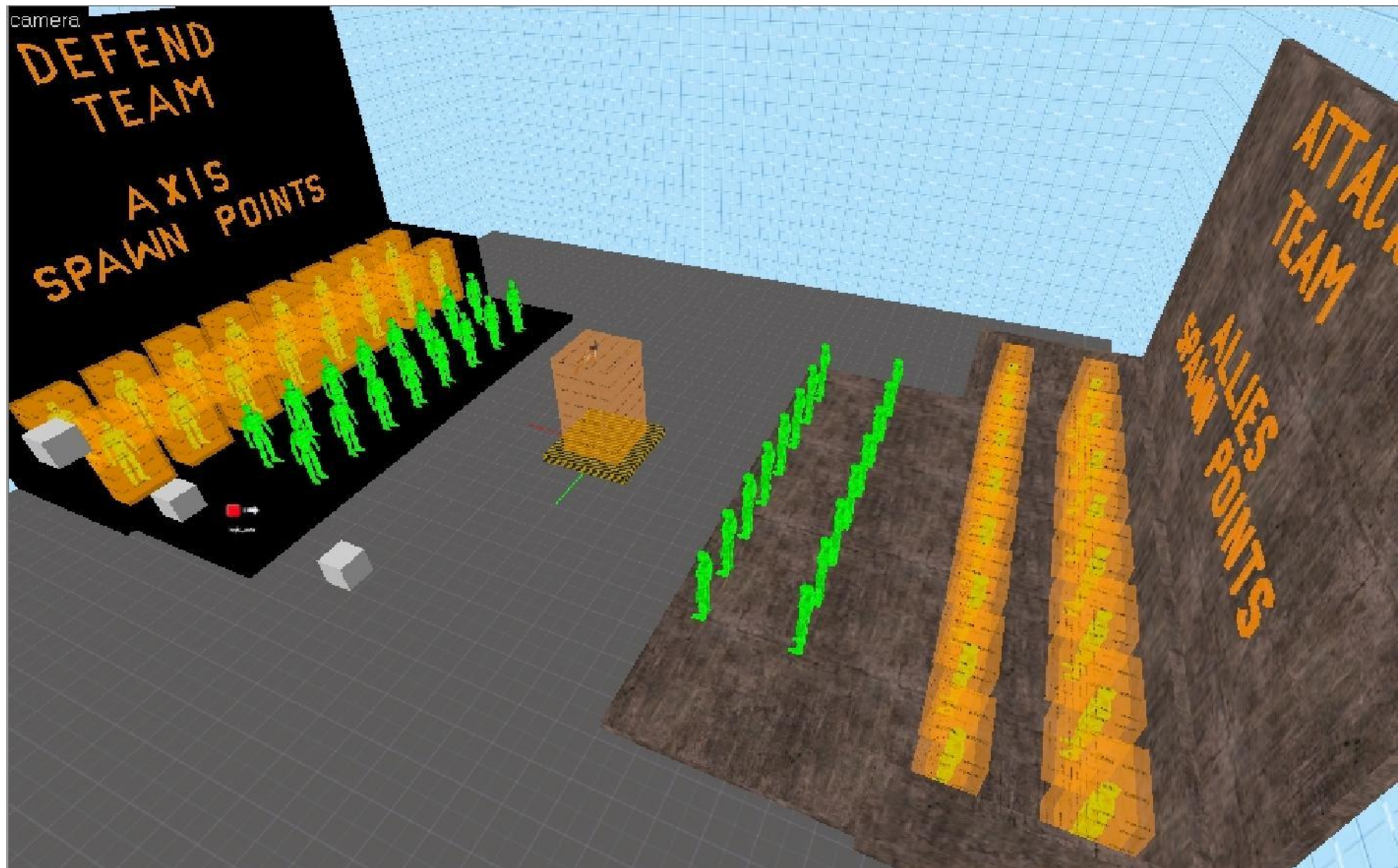
Now rename the teleport destinations with a unique name for each one.

- attacker_side_spawn_1 ... to ... attacker_side_spawn_16
- defender_side_spawn_1 ... to ... defender_side_spawn_16

Go to each of the trigger_teleport entities around all spawn points , go to the **Remote Destination** and pick the correct destination. Be careful and remember each spawn point will go to either the attack side or defend side... just turn the "1" to a "2" then continue on to "16" for each of the teleports. What this does is make each teleport unique, each one is now defined by ...

- The name that the control entities will switch (logic_auto switches **allied_spawn_teleport_to_attacker** and **axis_spawn_teleport_to_defender**)
- The filter that will only allow a certain team to use it
- The Remote Destination, each destination is different

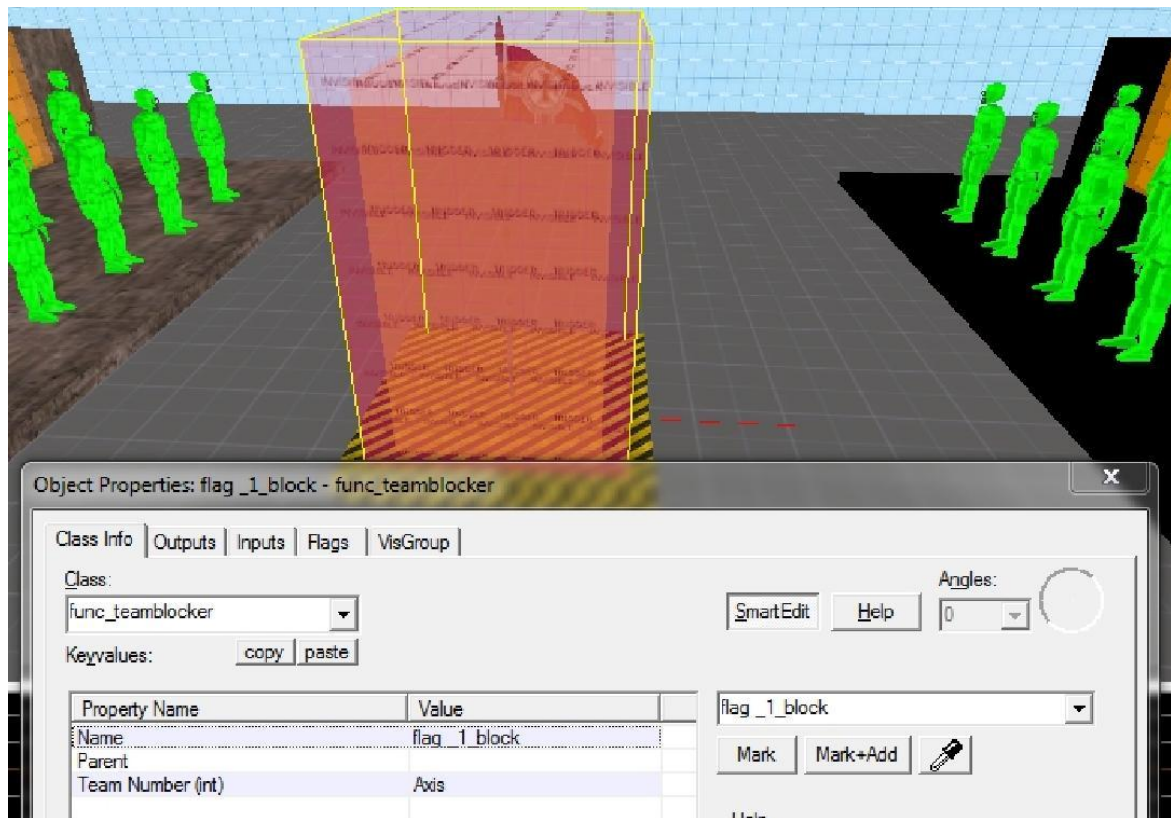
Now we have a set of 32 spawn points each one unique and depending on who won, will go to a individual spawn area divided up between 16 positions per spawn. See the next page of what that looks like. If you like I have made a set attached to this tutorial you can cut and paste.



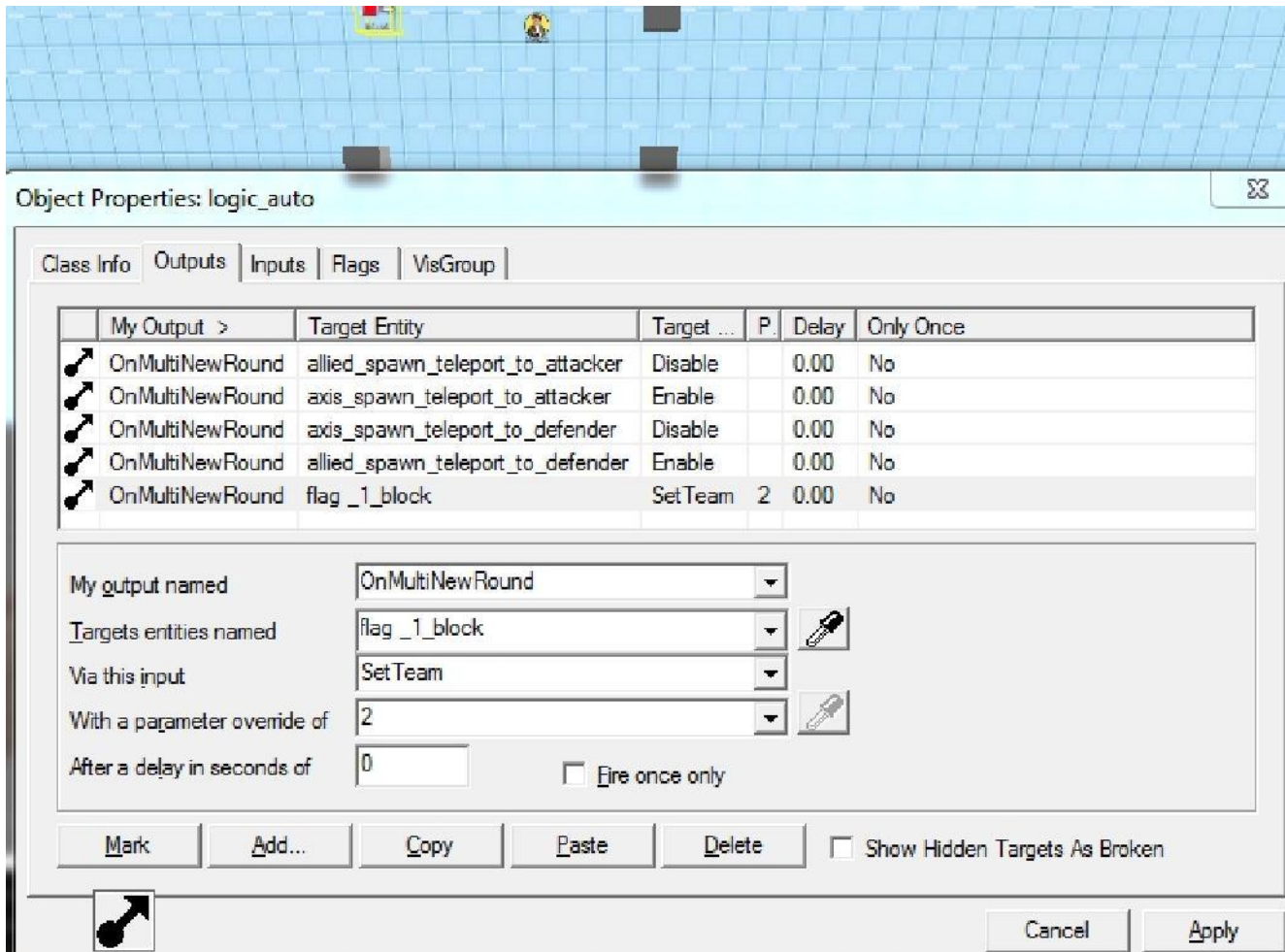
There is more yet. You may have noticed 3 maps in the Download...

- example map single spawns (a plain map with just 2 spawns and the control entities)
- example map 32 spawns (The same as above but with a 32 player set.)
- 32 swap spawns and deny defender access to the flag cap (the next twist in this tale)

Lets now add a function to the flag caps. We will deny the Defending team access to the flag cap. Only the Attacking side, be they allies or axis can cap. There is a YouTube video on this... <http://www.youtube.com/watch?v=81LeMo2jv9U> . We will make a func_teamblocker and place it over the cap zone, fill it out like this...



This will stop the axis in the first round, the allies will be able to reach this. We now need to go to the logic_auto outputs...



Object Properties: logic_auto

Class Info | **Outputs** | Inputs | Flags | VisGroup

	My Output >	Target Entity	Target ...	P.	Delay	Only Once
🔗	OnMultiNewRound	allied_spawn_teleport_to_attacker	Disable		0.00	No
🔗	OnMultiNewRound	axis_spawn_teleport_to_attacker	Enable		0.00	No
🔗	OnMultiNewRound	axis_spawn_teleport_to_defender	Disable		0.00	No
🔗	OnMultiNewRound	allied_spawn_teleport_to_defender	Enable		0.00	No
🔗	OnMultiNewRound	flag_1_block	SetTeam	2	0.00	No

My output named: OnMultiNewRound

Targets entities named: flag_1_block

Via this input: SetTeam

With a parameter override of: 2

After a delay in seconds of: 0 ☐ Fire once only

Mark Add... Copy Paste Delete ☐ Show Hidden Targets As Broken

Cancel Apply

Note that the "With the parameter override of" Value is "2" ? This is important, don't put "Allies" like in the func_teamblocker, it will not work, the teams are defined here like this... Axis = "3" and Allies = "2". Now it will work! The attacking team are the only ones that can cap.

What you have now in the VMF.. " **dod_swap_spawns_cap_deny_16Set.vmf**" in the **32 swap spawns and deny defender access to the flag cap** folder is a template to add to your map, just copy the func_teamblocker and place them over all the flags... or just some, they deny access, but if you leave some out defender can still cap... but never cap-out the round.

Have fun, with the above, you may even think of some twists and turns of your own... well share then stupid!

INsane

September 2010

dodbits.com