

Blow up that multiplayer objective :: Bjarne Grønnevik

There are a bunch of tutorials on blowing objectives... they mostly tell you half the story. I will try to be more complete.

This tutorial requires a bit of scripting and setting parameters and configuring entities, so if you have no idea what those words mean, try to learn that first.

Oh yeah, you'll probably want to put the objective somewhere, so I recommend building a map, made up only of one small room (To minimize compile times while you are learning). All red words below are keywords that should not be changed, the rest you can change as much as you like.

A multiplayer objective destruction is made out of a number of parts:

- 1: [Making a bomb.](#)
- 2: [A trigger for that bomb.](#)
- 3: [Something to blow up.](#)
- 4: [Explosion radius.](#)
- 5: [A chain reaction explosion FX.](#)
- 6: [Debris from the explosion.](#)
- 7: [Leftovers after the explosion.](#)
- 8: [Some scripting in your .scr file.](#)

So lets start from the top.

(1) Making a bomb

Create a script_model (with nothing selected, right click on the map grid and select script -> model).

Give it these key-value pairs:

#exploder_set 1 // The ID that connects all the parts of the objective. To add entities to this exploder give them the key/value #set/1.

\$explosion_fx models/fx/fx_explosion.tik // What explosion effect to activate when the bomb blows (optional).

\$killarea groundzero // Target name of a trigger_multiple brush that defines the area where you will be killed by the bomb blast (optional, if left out a default radius from the bomb will be used). See 4

\$trigger_name panel_trigger // Target name of a trigger_use brush that defines the area where you can plant the bomb.

model models/items/pulse_explosive.tik // What model to use for the bomb (optional, you can make your own bomb and skip this key).

targetname panel_bomb // A name used by other entities to target this entity.

By now you should have a nice black/yellow bomb before you.

(2) Making a trigger for the bomb

Create a trigger_use (with nothing selected, right click on the map grid and select trigger->use).Give it these key-value pairs:

targetname panel_trigger // A name used by other entities to target this entity (the bomb above uses it).

Reposition the trigger so that it sits around your bomb. Done.

(3) Something to blow up

I made a control panel out of a single brush, textured with the norway/nor_panelXXX textures, but you can use any number of brushes or a model, in the same way as we set a model for the bomb (the models/statweapons/flak88turret.tik is apparently popular).

Select all the brushes you wish to blow up and make them a script_object (right click on the map grid and select script->object).Give it these key-value pairs:

#set 1 // The ID that connects all the parts of the objective to the bomb.

targetname exploder // Tells the exploder system that this is an item that shall be destroyed.

By now you should have a doomed panel.

(4) Blast radius calibration

Shouldn't soldiers nearby be affected by the explosion too? Well yes, the default behavior, like damage and blast radius is set in the .scr file below, but we can be more specific.

Create a trigger_multiple (with nothing selected, right click on the map grid and select trigger->multiple).Give it these key-value pairs:

targetname groundzero // A name used by other entities to target this entity (the bomb above uses it).

Reposition the trigger so that it sits around the area where the bomb should hurt /kill people. Done.

(5) Fire and brimstone shall rain!

We want some wicked fire and sound FX, right? You know you want it... So lets create a chain-reaction explosion effect:

Create two script_model's (with nothing selected, right click on the map grid and select script -> model).Place them close to the bomb and give both of them these key-value pairs:

#set 1 // The ID that connects all the parts of the objective. To add entities to this exploder give them the key / value **#set** / 1.

model models/animate/fx_explosion_tank.tik // What model to use for the explosion effect.

targetname exploderfire // Tells the exploder system that this is an item containing special FX.

Give one of the script_model's this value:

#pause 0.5 // Delays the explosion FX to 0.5 seconds after the bomb goes off.

By now you should have a funky, two stage explosion.

(6) Throwing debris out of the explosion.

We are not talking debris from the explosion FX here, but the thing you are blowing up breaking apart and getting hurled in all directions. It will not break apart by itself; you will have to make the debris by yourself (or select a model).

I copied the panel and split the copy in four pieces.

Select all the brushes **one by one** and make them script_object's (right click on the map grid and select script->object).

Give them these key-value pairs :

#set 1 //The ID that connects all the parts of the objective to the bomb.

targetname exploderchunk // Tells the exploder system that this is an item that shall be hurled when the bomb is blown. Place the pieces a bit outside the panel (don't worry, the exploder system will hide them until the bomb blows).

Where to throw the debris?

Create a script_origin (with nothing selected, right click on the map grid and select script -> origin) for each piece where you want it to be hurled (Remember: It will start in that direction, and then be drawn towards the ground as gravity takes over).

Unselect everything, select **first** the piece **then** the script_origin, then press CTRL+k (this connects the entities. Order is important). You can also set the targetname of the

script_origin and then set the target key of the piece to the same value (using CTRL+k gives targetnames like t3, and that can be hard to debug).

You should now have some debris that point a blue arrow towards one red box each.

(7) Something left after the explosion?

Sometimes there is a need for something to remain after the explosion. A twisted Flak88, a nice wall turns into a smoking hole...

Construct what remains after the explosion. Select the blown up remains and make them a script_object (right click on the map grid and select script->object). Give the selection these key-value pairs:

#set 1 // ID that connects all the parts of the objective to the bomb.

targetname explodersmashed // Tells the exploder system that this is an item that shall replace the exploder object after it has been destroyed.

That's what will replace the panel after the explosion.

(8) Last but not least: some scripting

Here is the script for the objective map. The objective related stuff in red. Explanations / comments in green:

```
// Custom objective test map
// ARCHITECTURE: Bjarne Grønnevik
// SCRIPTING: Bjarne Grønnevik
```

main:

```
setcvar "g_obj_alliedtext1" "Destroy the control"
setcvar "g_obj_alliedtext2" "panel"
setcvar "g_obj_alliedtext3" "" setcvar "g_obj_axistext1" "Defend the control"
setcvar "g_obj_axistext2" "panel"
setcvar "g_obj_axistext3" ""
setcvar "g_scoreboardpic" "none"
```

level waittill prespawn

```
exec global/DMprecache.scr
level.script = maps/obj/cust_obj_test.scr
exec global/ambient.scr cust_obj_test
```

```
thread global/exploder.scr::main // Initialize the exploder subsystem
```

level waittill spawn

```
level.defusing_team = "axis" // Axis like the bombs unplanted  
level.planting_team = "allies" // Allies will try to plant the bombs  
level.targets_to_destroy = 1 // Number of targets in this map  
level.bomb_damage = 200 // Default damage of the bomb  
level.bomb_explosion_radius = 2048 // Default radius of bomb blast
```

```
// Set the parameters for round based match  
level.dmrespawning = 0 // 1 or 0 (0=no respawn)  
level.dmrroundlimit = 5 // round time limit in minutes  
level.clockside = axis // set to axis, allies, kills, or draw
```

```
// level waittill roundstart // Comment out this line using '/' before it to be able to set the  
bomb when alone on the map ( just for testing )
```

```
$panel_bomb thread global/obj_dm.scr::bomb_thinker // "panel_bomb" is the targetname set  
on the bomb in MOHRadiant  
thread allies_win_bomb // Start the win check thread for allies  
$panel_bomb thread axis_win_timer // Start the win check thread for axis  
end // end of main
```

```
// Allied victory test
```

```
allies_win_bomb:  
while(level.targets_destroyed < level.targets_to_destroy) // While undestroyed objectives left  
waitframe // chill out  
teamwin allies // No objectives left allies win  
end // end allied victory test
```

```
// Axis victory test
```

```
axis_win_timer:  
level waittill axiswin // At the end Axis win  
end // end axis victory test
```

That's all you need to know to blow most of the world into tiny bits, in thy mercy...

[[Download the Example map](#)]

Need any help : Ask in the [Forum](#)