



Your Second Dystopia Map

Assumed reading: [Your First Dystopia Map](#).

Let's add some more features to the map. We aren't ready to make the big jump to adding cyberspace just yet - we'll leave that for [Your Third Dystopia Map](#).

In this map, we'll add turrets, buttons, doors and forcefields and who knows what else?

Contents [hide]

- 1 [Step 1: Make a Copy](#)
- 2 [Step 2: Add A npc_turret_ceiling](#)
- 3 [Step 3: Add A func_button](#)
 - 3.1 [Move Direction](#)
- 4 [Step 4: Add a func_door](#)
 - 4.1 [Door Properties](#)
 - 4.2 [Fun With Doors](#)

Step 1: Make a Copy

After making [the first map](#) you should have an enclosed box with two spawns and an objective. File -> Save As -> dys_whateverNameYouWant2

In Hammer, you want to make backups. Lots and lots of backups. If you screw something up, there's no undo key after you compiled, closed Hammer and loaded it up in Dystopia. Any damage is irreversible, so make backups.

We'll work on the new copy.

Step 2: Add A npc_turret_ceiling

Turrets are just another kind of entity.

Add a npc_turret_ceiling by clicking somewhere on the roof.

Double click on it to open its Object Properties.

Name it PunkTurret1.

Set its team to Punks.

If you want, set its health to whatever, or make it invulnerable.

For various reasons, we need a logic_auto invisible-magic-controller-thing to set the turret up properly. Create a logic_auto entity and place it near the turret (for convenience).

Double click on it to open its Object Properties.

Select the Outputs tab.

Click Add.

In "My output named:" type "OnNewGame"

Set "Target entities named:" PunkTurret1

Set "Via this input:" Enable.

This way, if in the first round, the turret is shot down or disabled just before map end, it'll re-enable for the next round. Additionally, I found the turret was invulnerable without one of those.

Press F9 to compile the map, close Hammer, and load up the map in Dystopia. Join as a Punk, make sure the turret is on, and destructible.

Step 3: Add A func_button

We'll add a button to toggle the turret on and off. Buttons are brush entities, not node entities.

Make a very small block - say 4x4x4 units. Put it on wall.

Apply a texture to it.

Double click on it to open its Object Properties.

Change its class to func_button in the pull-down menu.

Set the Name to ToggleTurretButton.

Set its Lip to 1 (or another number less than 4. we want it to move into the wall leaving only 1 inch sticking out).

Set its Speed to 10 (or another high number, we want the button to move fast).

Set the Delay Before Reset to 1 (or a similar low number).

Set the Sounds to something so you have audible confirmation it's working.

Select the Outputs tab.

Click Add.

Set "Targets Entities Named": PunkTurret1.

Set "Via This Input": Toggle.

Set "My output named": OnPressed.

Compile and run the map. Pressing the button as punk should disable or enable the turret.

Move Direction

You may notice that the button moves in a strange direction instead of receding into the wall. By default, all buttons (and doors) move in the positive-X direction (to the right, from default).

You can make it move in any direction - up, down, into or out of a wall, left or right - by setting the Move Direction property. Let's play with it now.

Double click on it to open its Object Properties.

Select the Move Direction field. There will be three numbers - by default, 0 0 0. These are the pitch, yaw and roll. This is easiest to explain in terms of aeroplanes; pitch is when the nose of the plane goes up and down, yaw is when the nose of the aeroplane turns left or right, and roll is when the body of the aeroplane rotates around its longest axis (the wings spin around).

For a normal button or door, we can just set the Yaw value and leave the others zero. Assuming your room is square, set the Yaw value (the second of the three) to a multiple of 90; 0, 90, 180, 270.

Compile and run the map. Pressing the button as punk should disable or enable the turret. Watch which way the button moves. You should be able to see it move in a different direction.

If you want a button (or door) to move up or down, select Up or Down from the pull-down menu.

Set the button to move whatever direction you want; I personally like them to retract into the wall.

Step 4: Add a func_door

A door is a brush entity that moves a fixed distance in a straight line, in some direction.

Obviously, this has applications as a sliding door, but also has many other uses that we will get to.

There is also a func_door_rotating which has a hinged door that rotates some angle about some axis. This too has uses beyond just a door.

Draw a rectangular block the size and shape of your door. Draw two other blocks about the same size either side of it - these will be doorposts or walls (they aren't necessary but they make it easier to visualise).

Select the door block and click ToEntity.

Select func_door from the pull-down menu.

Name it door1.

Give it a few sounds for when it starts and stops moving, or locks, or unlocks.

Click Apply.

Go to the Flags tab, and check Use Opens. This will allow the door to be opened with the E key as well as just by touching it.

Compile and run the map. The door should move in some direction when touched. We want it to slide aside and let us through the doorway.

Re-launch Hammer and set the Move Direction so that the door opens how you want it to.

Compile and run the map to make sure the door works properly now.

The door will move along its entire length in the direction that it is set to move in.

Door Properties

You can set a few other interesting properties of doors.

- block damage, which damages anything caught in the door as it starts to close. The airlock doors on Vaccine have this set to a number approaching infinity.
- reset time. If you set it to -1, the door stays open. Set it to a number >= 0, the door will wait that long and then start to shut itself. Default is 4 seconds.
- lock or unlock. In the Flags tab, a Start Locked flag can be set - the door can be unlocked by a button in meatspace or cyberspace, or when an objective changes status.
- start position. Default is closed, but it can be set to open. If you want, you can set start position to open and tick Starts Locked, then have a button that will unlock, close and relock the door, to seal off an old objective or an alternate route.
- Touch Opens, in the Flags tab. Uncheck this and leave Use Opens unchecked, and you can make a door only controllable by cyberspace or a button.

Fun With Doors

Doors don't need to be attached to walls, floor or ceiling, or can be attached to any combination of the three. Think of them as floating blocks of any shape that move a certain distance in any direction, and then back again.

Sliding doors are just the beginning. Hatches, trapdoors, moving walls, retractable stairways and ladders, sliding platforms and moving pistons are all possible.

Retrieved from "http://www.dystopia-game.com/wiki/index.php?title=Your_Second_Dystopia_Map"

navigation

- [Main Page](#)
- [Community portal](#)
- [Current events](#)
- [Recent changes](#)
- [Random page](#)
- [Help](#)

search

toolbox

- [What links here](#)
- [Related changes](#)
- [Special pages](#)
- [Printable version](#)
- [Permanent link](#)