

---

# **Eureka Documentation**

**Wesley Werner**

**Jan 14, 2019**



---

## Contents

---

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	License . . . . .	3
<b>2</b>	<b>Installation</b>	<b>5</b>
2.1	Requirements . . . . .	5
2.2	Windows . . . . .	6
2.3	GNU / Linux . . . . .	6
<b>3</b>	<b>User Interface</b>	<b>7</b>
3.1	Panning and Zooming . . . . .	8
3.2	The Camera . . . . .	8
3.3	The Grid . . . . .	9
3.4	Rendering Mode . . . . .	10
3.5	Find and Replace . . . . .	15
<b>4</b>	<b>Project Management</b>	<b>17</b>
4.1	Create a new WAD . . . . .	17
4.2	Open a map . . . . .	18
4.3	Add a new map . . . . .	19
4.4	Rename a map . . . . .	20
4.5	Copy a map . . . . .	21
4.6	Delete a map . . . . .	21
4.7	Export a map . . . . .	21
<b>5</b>	<b>Mapping Concepts</b>	<b>25</b>
5.1	Vertices . . . . .	25
5.2	Linedefs . . . . .	26
5.3	Sectors . . . . .	28
5.4	Things . . . . .	29
5.5	Multiple selections . . . . .	30
5.6	Deselecting . . . . .	30
5.7	Transformations . . . . .	30
5.8	Textures . . . . .	36
<b>6</b>	<b>Building Basics</b>	<b>43</b>
6.1	Adding a room . . . . .	43
6.2	Joining rooms . . . . .	47

6.3	Adjusting ceiling height . . . . .	49
6.4	Texture alignment . . . . .	51
6.5	Auto alignment . . . . .	53
<b>7</b>	<b>Cookbook . . . . .</b>	<b>55</b>
7.1	Stairs . . . . .	55
7.2	Curved Stairs . . . . .	58
7.3	Sky . . . . .	69
7.4	Sacrificial Altar . . . . .	72
7.5	Toxic Pool . . . . .	80
7.6	Doors . . . . .	84
7.7	Teleporters . . . . .	96
7.8	Lifts . . . . .	99
7.9	Traps . . . . .	103
7.10	Prefabs . . . . .	112
<b>8</b>	<b>Cheatsheet . . . . .</b>	<b>125</b>

*Manual for Eureka v1.21*



# CHAPTER 1

---

## Introduction

---

Eureka is a map editor for the classic DOOM games, and a few related games such as Heretic and Hexen. It supports Linux, Windows and OS X. Eureka was created by Andrew Apted and lives at <http://eureka-editor.sourceforge.net>

This manual was written by Wesley Werner. If you find a mistake in this manual [please report it here](#). If you want to contribute, [fork the repo](#) and send a pull request.

There is also a [downloadable PDF version of this manual](#), and a [printable cheat sheet](#).

## 1.1 License

This manual is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#).





### 2.1 Requirements

A IWAD file, or “Internal WAD”, contains the default maps, textures, sounds and other auxiliary data. You can use the *DOOM.WAD* or *DOOM2.WAD* files if you own those.

Alternatively you can download the [Freedoom](#) WADS, Freedoom is liberally licensed under the BSD license and provides free levels, artwork, sound effects and music compatible with any classic Doom source port. Download and extract Freedoom to an accessible location of your choice. If you are a GNU / Linux user you may extract the Freedoom WADS into `~/.eureka/iwads/` where they are automatically detected for your convenience.

---

**Note:** A PWAD file, or “Patch WAD”, provides additional levels and other resources that replace those in the IWAD. When making levels in Eureka we will be working with PWADs - any mention of a WAD file implies the PWAD unless otherwise noted.

---

#### 2.1.1 Doom Engine

You will need a Doom source port to play-test your maps. Here are a couple of options you can browse:

- [Chocolate Doom](#) - accurately reproduces the experience of Doom as it was played in the 1990s. A safe option for testing your maps.
- [PrBoom](#) - an advanced engine with fine-grained control over options controlling compatibility levels. This port is the standard for recording and playing of demos.
- [Crispy Doom](#) - a minimalist engine with few extra features.
- [GZDoom](#) - an advanced and moddable engine with many features, including newer and non-standard features. You should not base your map’s compatibility off this port alone.

## 2.2 Windows

To get the latest Eureka for Windows, go to the [Eureka project page](#) and click the big “download” button. Extract the zip somewhere memorable and run *eureka.exe*.

## 2.3 GNU / Linux

Your package manager might contain Eureka, albeit if outdated you can opt to build from source.

- Locate the latest version on the [download page](#)
- Get and extract *eureka-\*-source.tar.gz*
- Install the build dependencies:

```
$ sudo apt-get install \  
build-essential \  
libfltk1.3-dev \  
libxft-dev \  
libxinerama-dev \  
libfontconfig1-dev \  
libjpeg-dev \  
libpng12-dev \  
zlib1g-dev \  
xdg-utils
```

- make the binary

```
$ cd eureka*  
$ make
```

- install Eureka

```
$ sudo make install
```

---

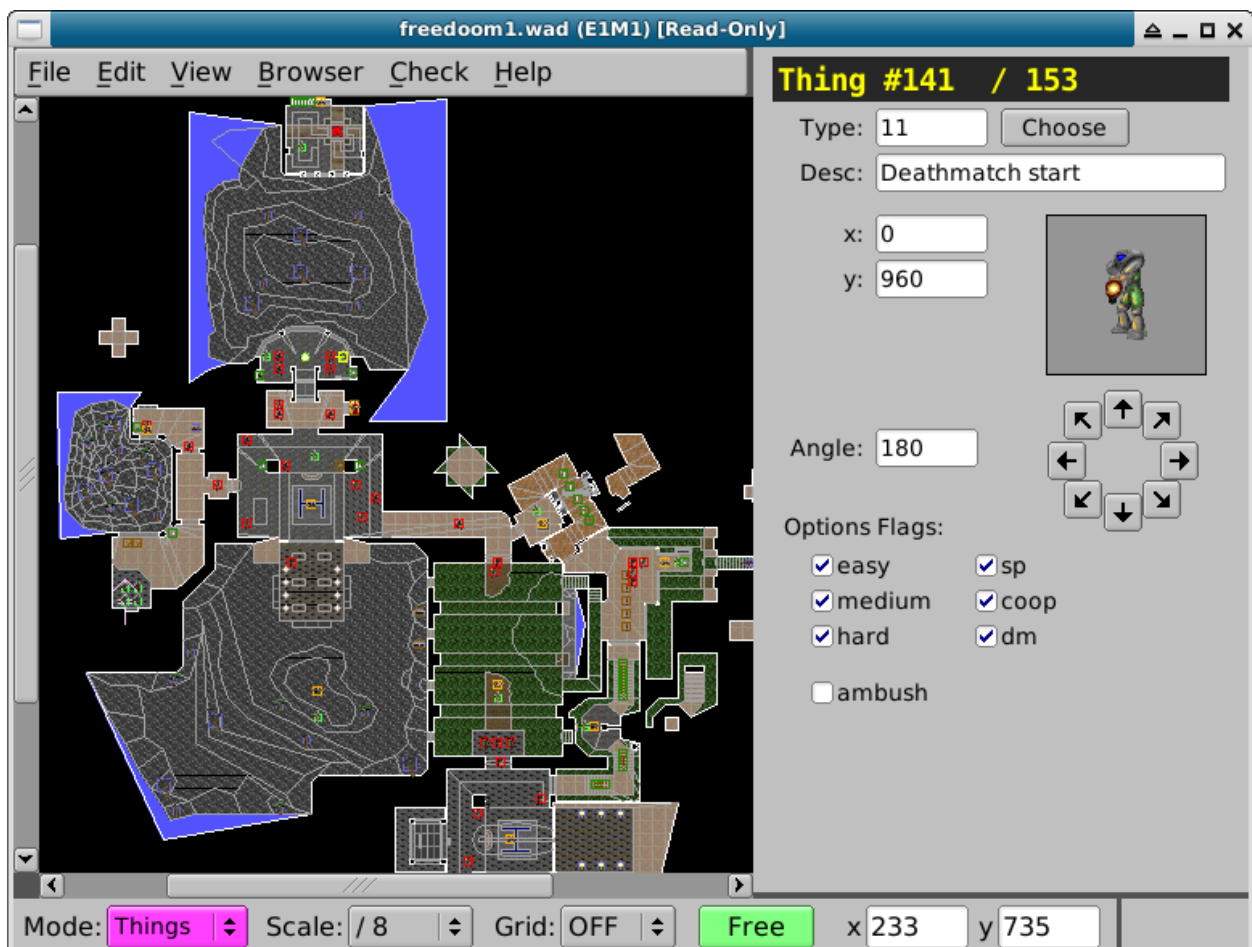
**Note:** See file INSTALL.txt included with the Eureka source for detailed build instructions.

---

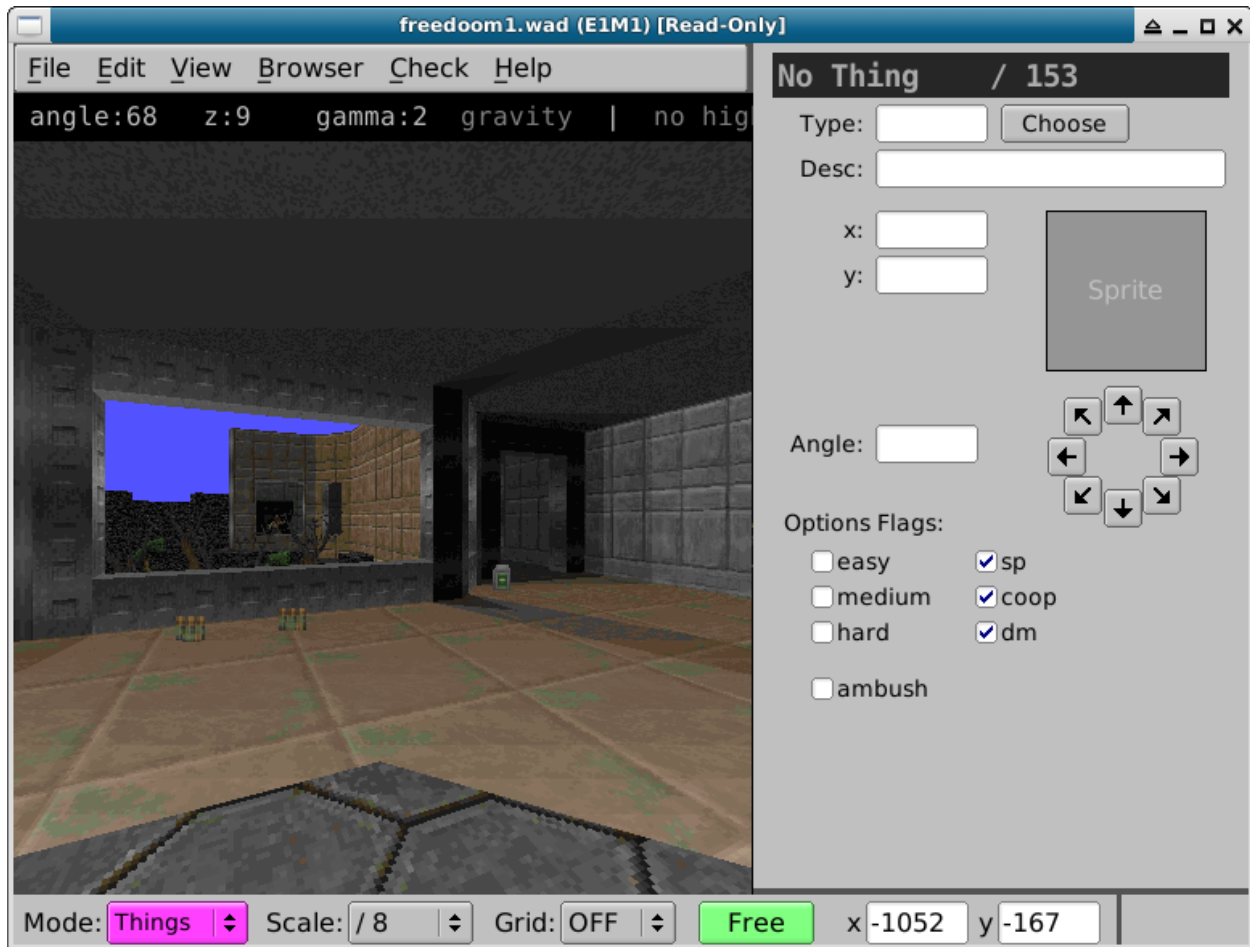
## CHAPTER 3

### User Interface

Eureka features a 2D orthogonal view and a 3D view. Switch between the two views with the `tab` key.



*The 2D view is used to construct walls and place things on the map*



*The 3D view is used to preview textures and make ceiling, floor and light adjustments*

## 3.1 Panning and Zooming

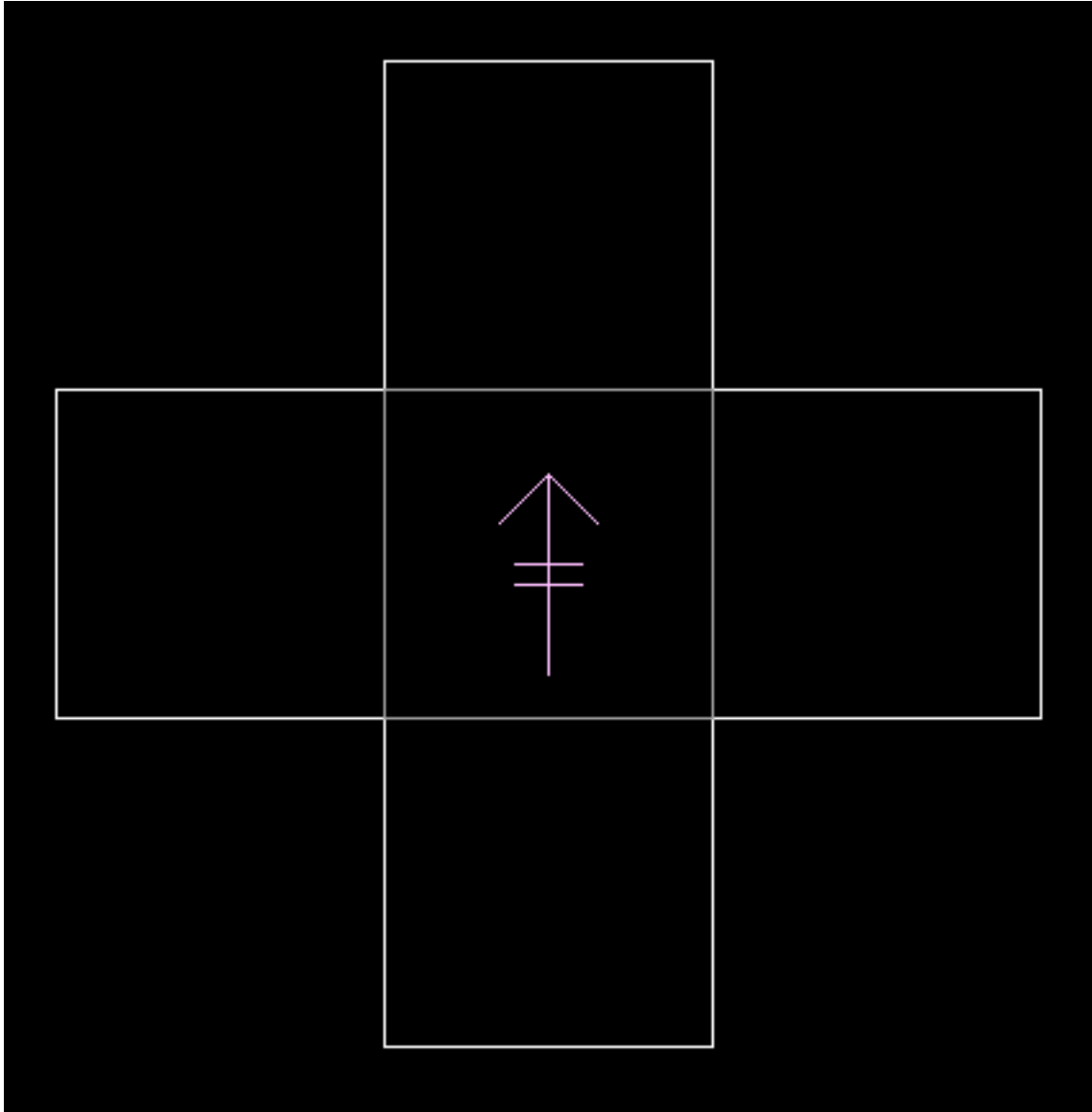
To move around the 2D view:

- Roll the mouse wheel to zoom
- Click and drag with the middle mouse button (mouse wheel) to pan
- Press `home` to zoom the whole map into view

## 3.2 The Camera

This arrow in 2D view indicates the current position of the 3D camera.

- Press `'` (single quote) in the 2D view to position the 3D camera at the location of the mouse cursor.
- Press `end` in 2D view to center the map on the camera position.



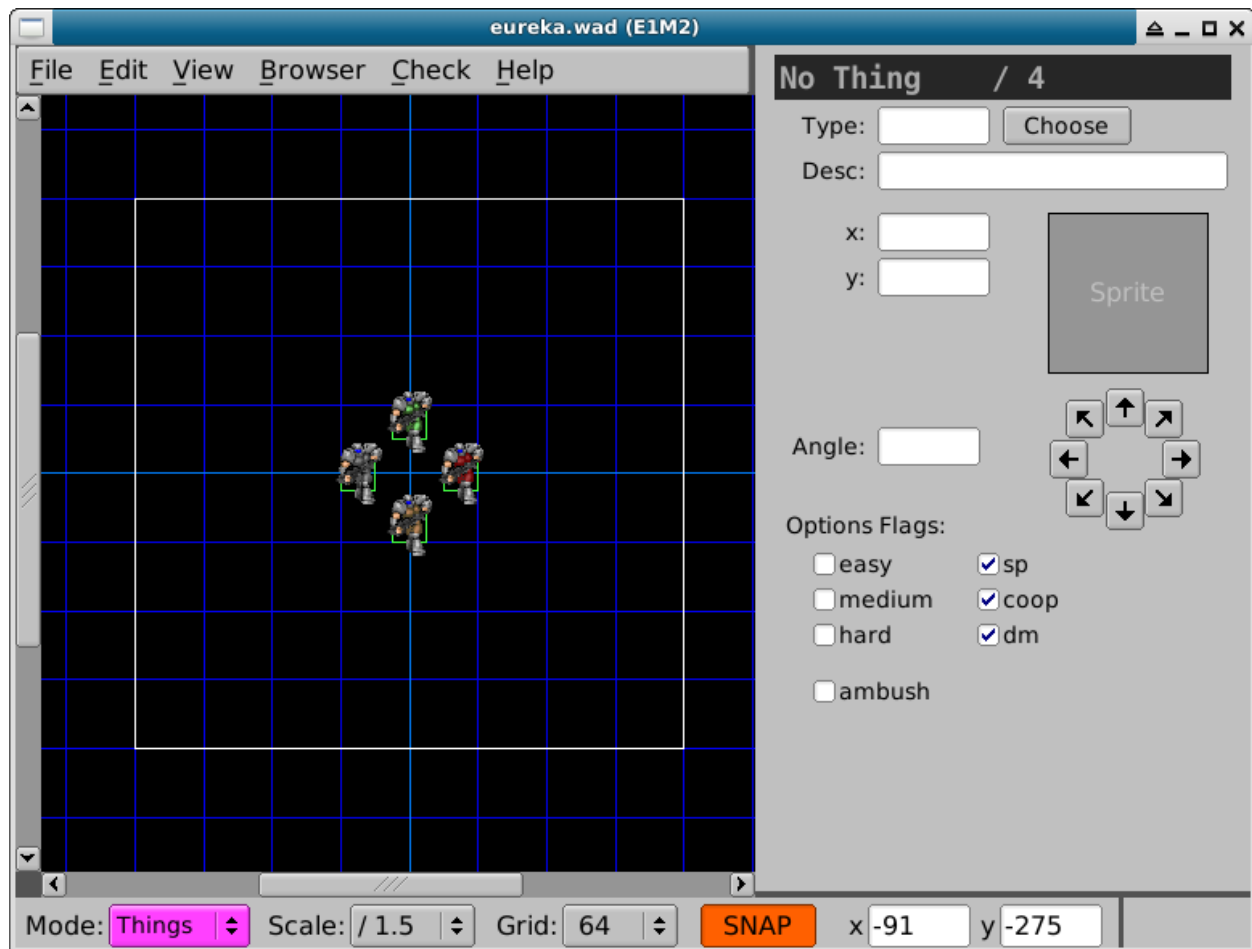
When in the 3D view you can control camera movement with the following controls:

- Roll the mouse wheel to move forward/backward
- Click and drag left/right with the middle mouse button to rotate the view
- Click and drag up/down with the middle mouse button to raise/lower the view

### 3.3 The Grid

Toggle the grid in 2D view with the **Grid** dropdown box (located on the bottom status bar), or by pressing `g`. You can quickly change the grid size with the `0-9` keys.

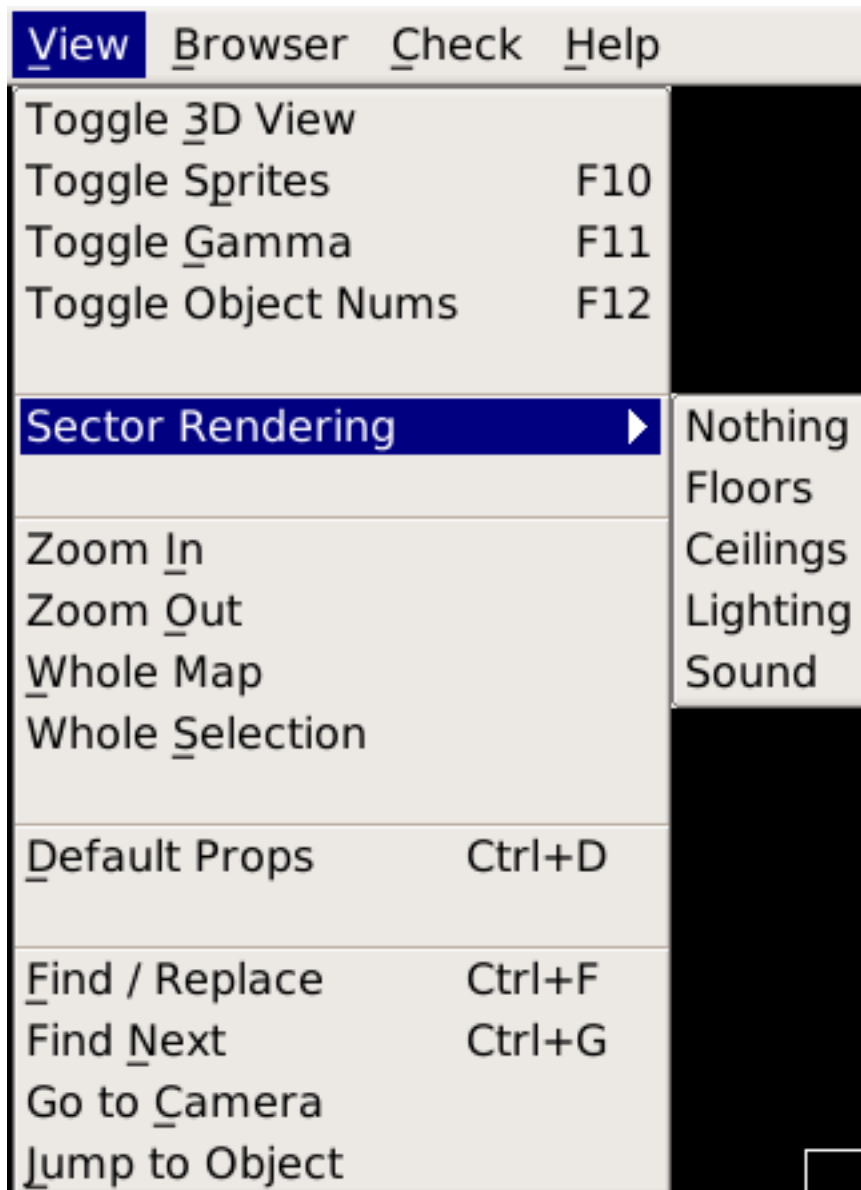
Toggle free mode / grid snapping with the `£` key.



**Note:** If you encounter lag while panning a large zoomed-out map, disable grid rendering with `g` while panning.

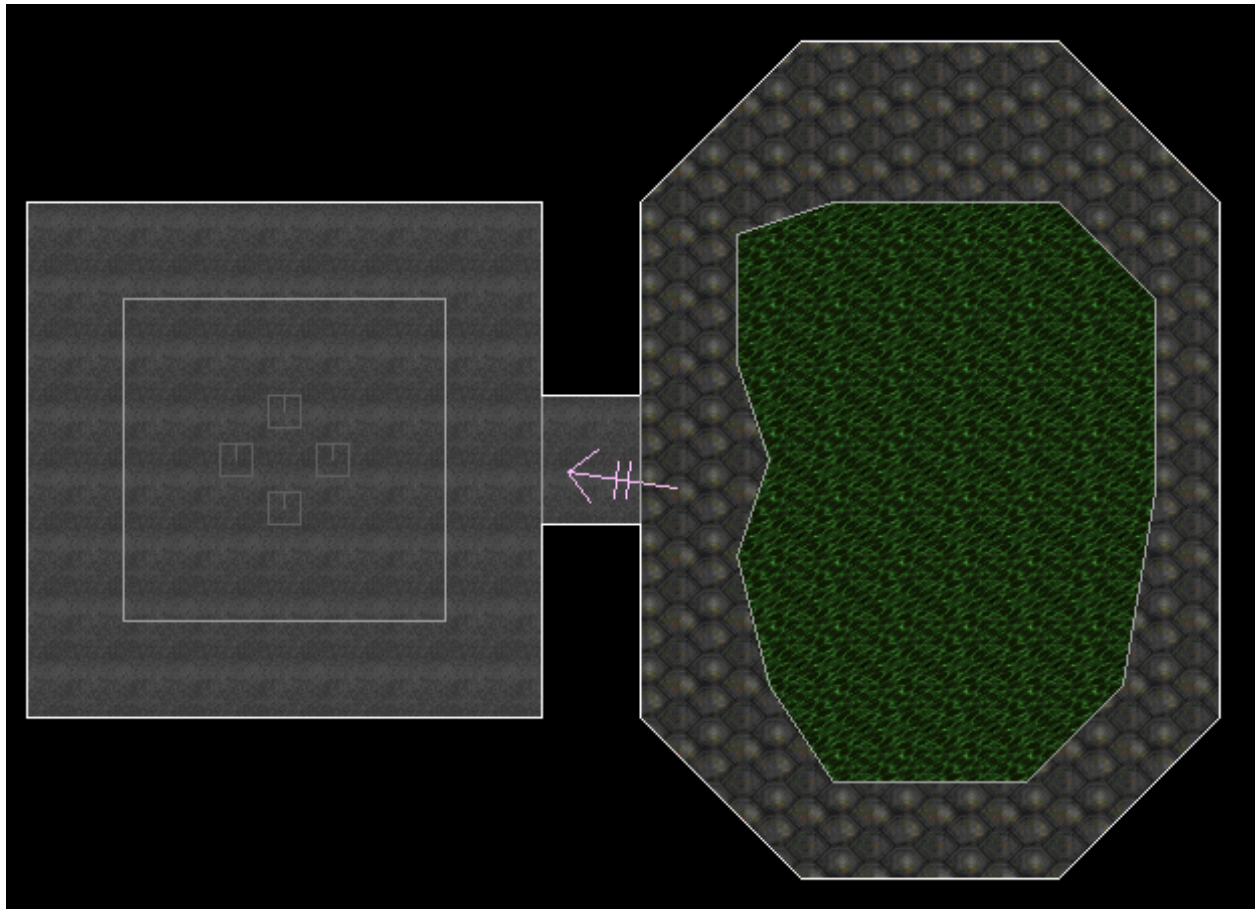
## 3.4 Rendering Mode

The *View / Sector Rendering* menu toggles how the 2D view draws sectors.



### 3.4.1 Floors

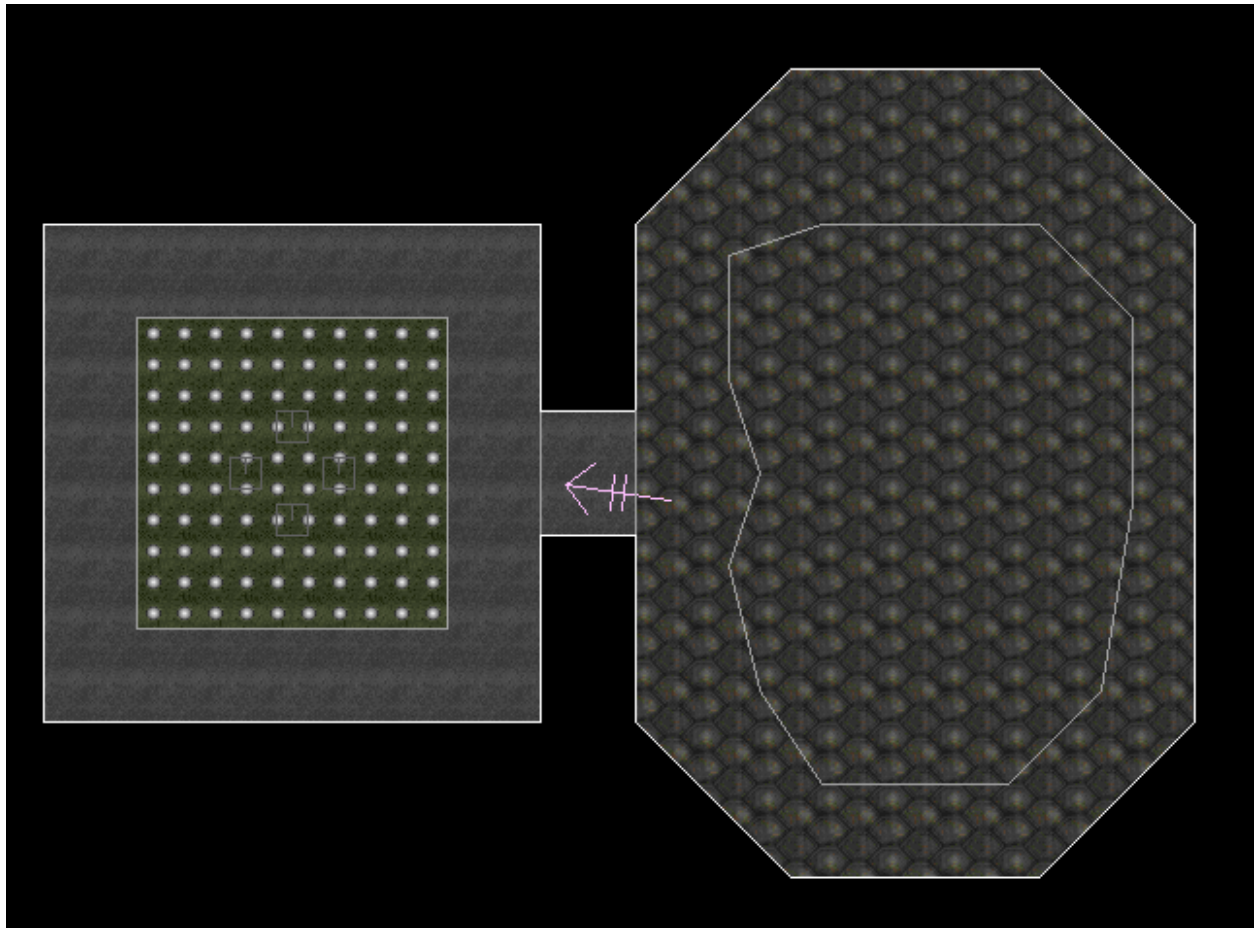
This mode draws the floor textures of sectors.



### 3.4.2 Ceilings

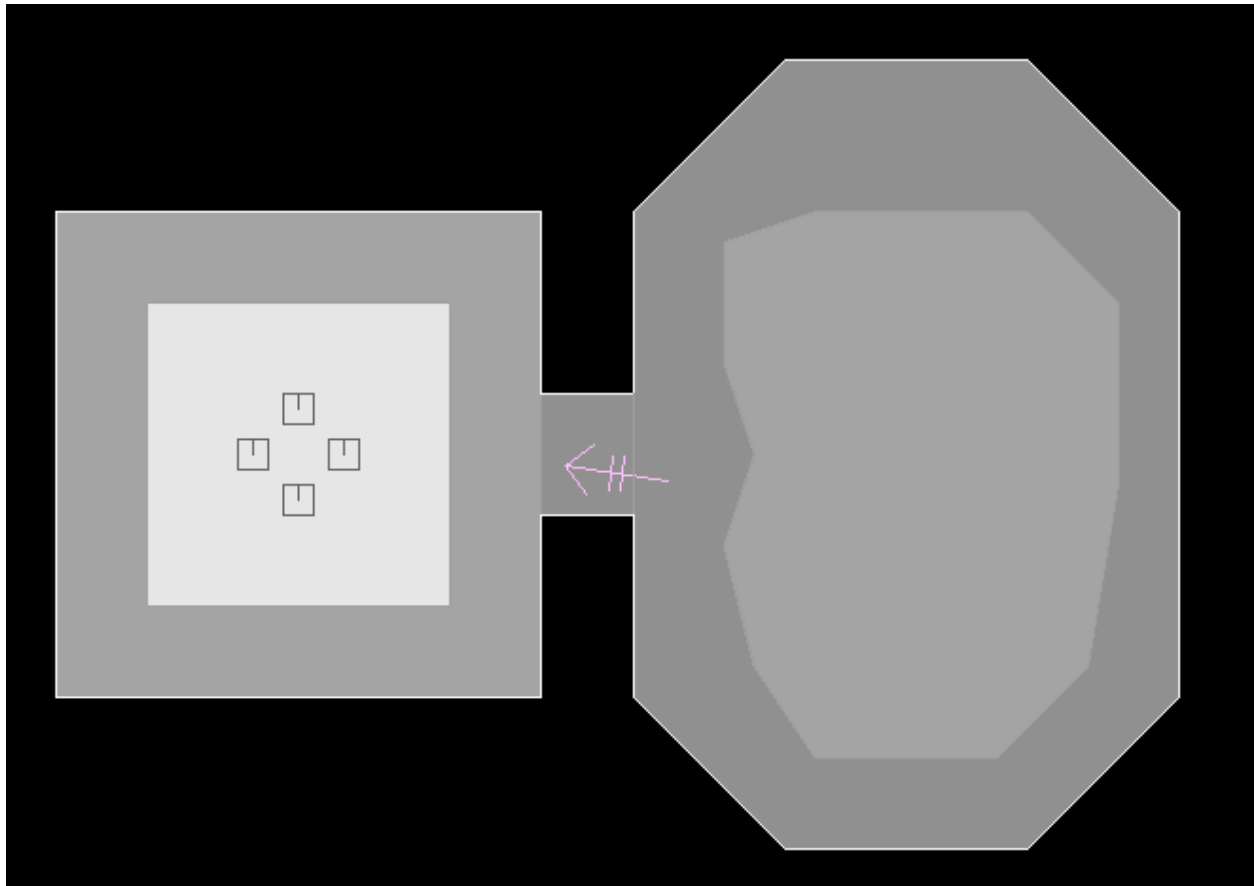
This mode draws the ceiling textures of sectors.





### 3.4.3 Lighting

The light render mode draws shades of sector light levels.

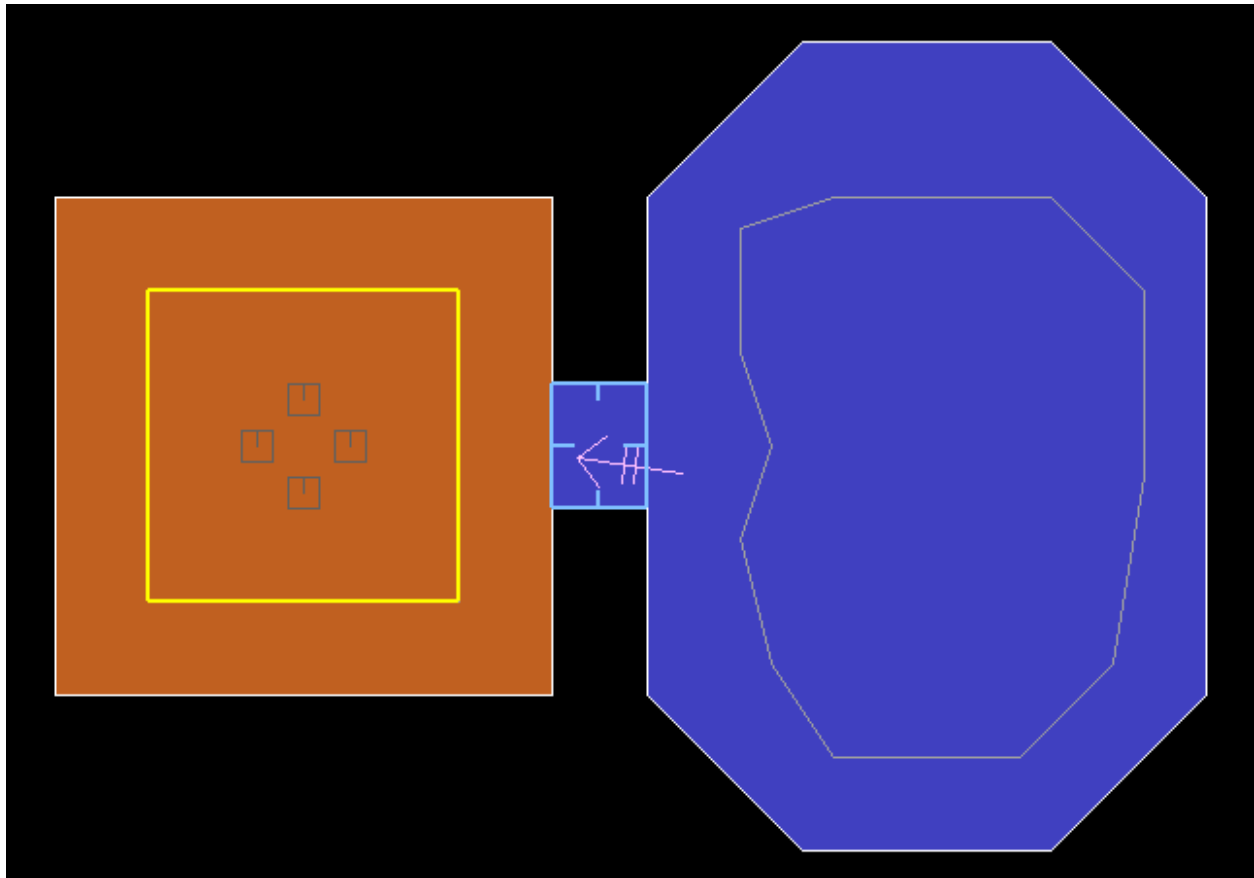


### 3.4.4 Sound

The sound render mode highlights sectors based on how sound travels. You have to be in sector edit mode for this mode to work (press `s`), hover your mouse cursor over a sector to see how sound will propagate.

- Orange sectors indicate where sound will reach at volume 2
- Blue sectors indicate connected sectors where sound does not reach
- Red sectors indicate where sound will reach at volume 1

By setting the *sound block* flag on linedefs, you can lower the volume of traveling sounds. Sounds do not travel across two sound-blocking lines.



## 3.5 Find and Replace

Open the find panel with the *View / Find* menu or press `control-f`.

You can search for Things, line textures, sector flats, lines by type (specials) or sectors by type.

X Find and Replace

Lines by Type

Match: 1

Desc: DR Open door

Choose

Next

Select All

New:

Desc:

Choose

Replace

Replace All

## CHAPTER 4

---

### Project Management

---

Your WAD can contain multiple maps, and their numbering is determined by the source IWAD you choose to use.

For Doom1, or Freedoom1, you get 4 episodes of 9 maps each: E1M1-9, E2M1-9, E3M1-9 and E4M1-9.

For Doom2, or Freedoom2, you get 32 maps: MAP01 - MAP32.

This manual will focus on the Freedoom1 IWAD.

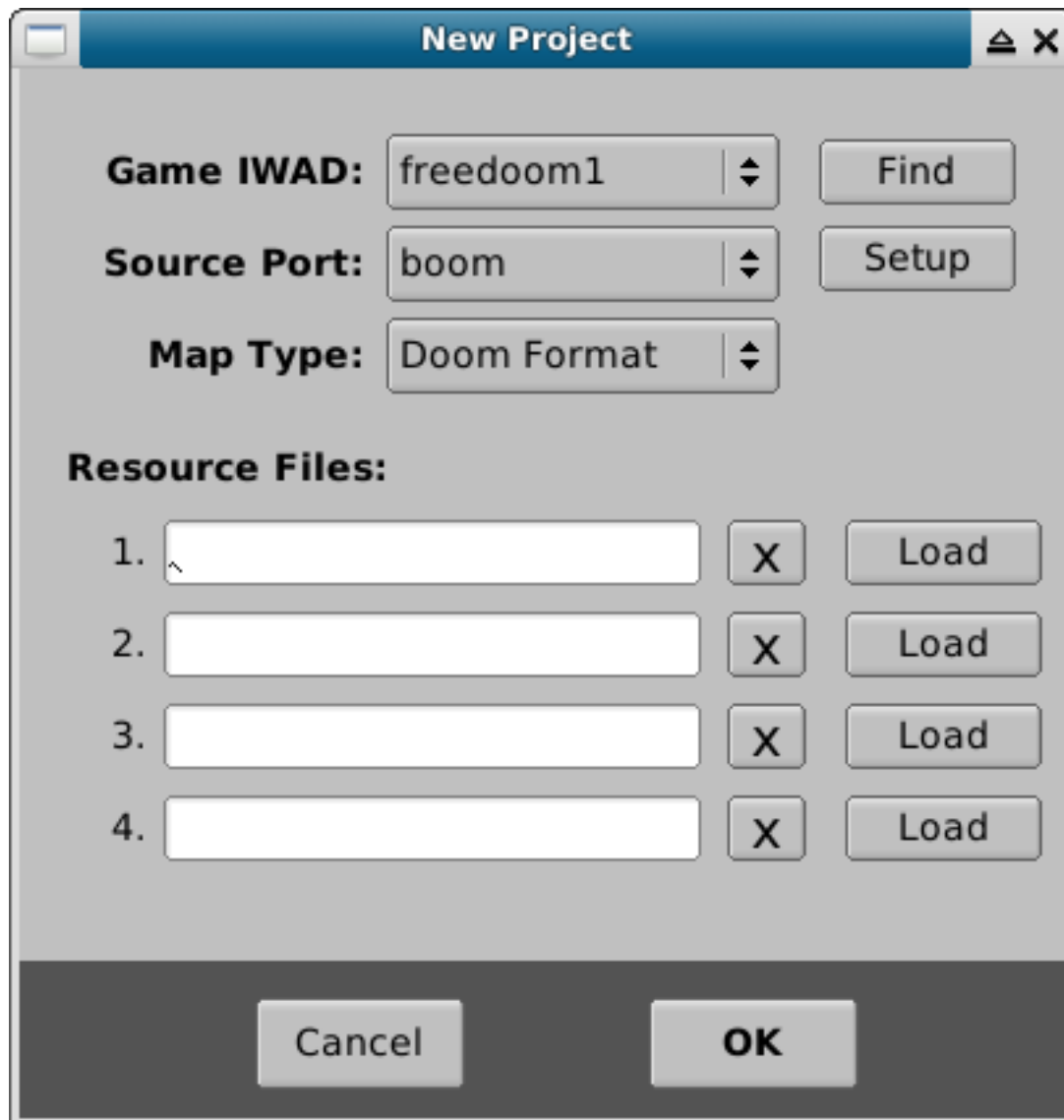
### 4.1 Create a new WAD

Use the **File -> New Project** menu to create a new WAD, choose the location and file name, click OK.

Next select the settings for you new project:

- IWAD: pick freedoom1 from the list, use the **Find** button if you need to browse for it
- Source Port: pick boom (or your nearest source port) from the list, use the **Setup** button to browse for the executable
- If you are a GNU / Linux user, you can find the location of the executable with the bash command `which prboom`.

Click **OK** when done.



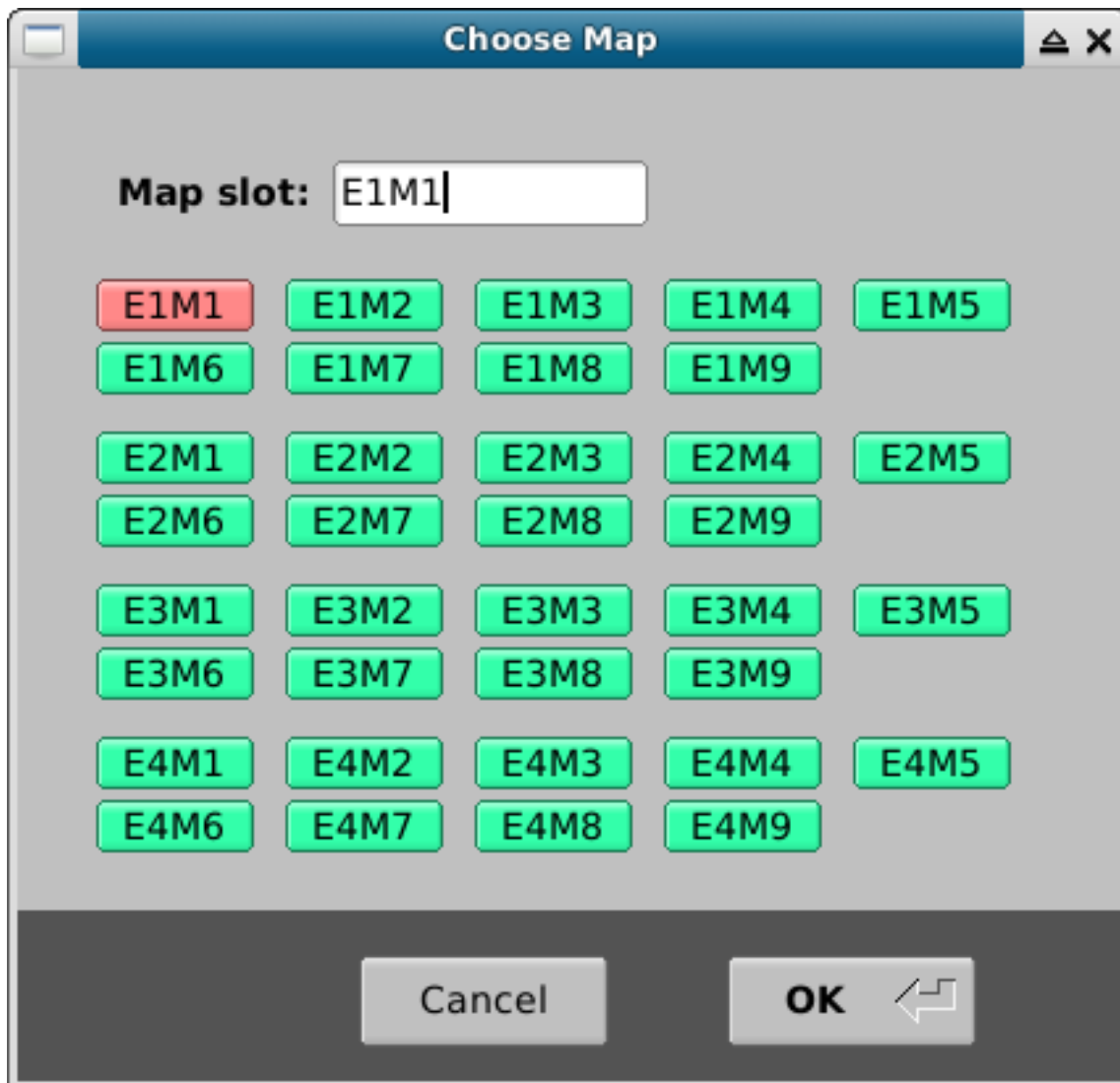
## 4.2 Open a map

Use the `File -> Open Map` menu or press `control-o`. Make sure you pick **Find map in the PWAD above**, otherwise you will be modifying the maps from the `freedom1` IWAD.



### 4.3 Add a new map

Use the `File -> Fresh Map` menu, you are prompted to pick the slot where the new map will be placed, green slots are empty and red slots already contain maps.

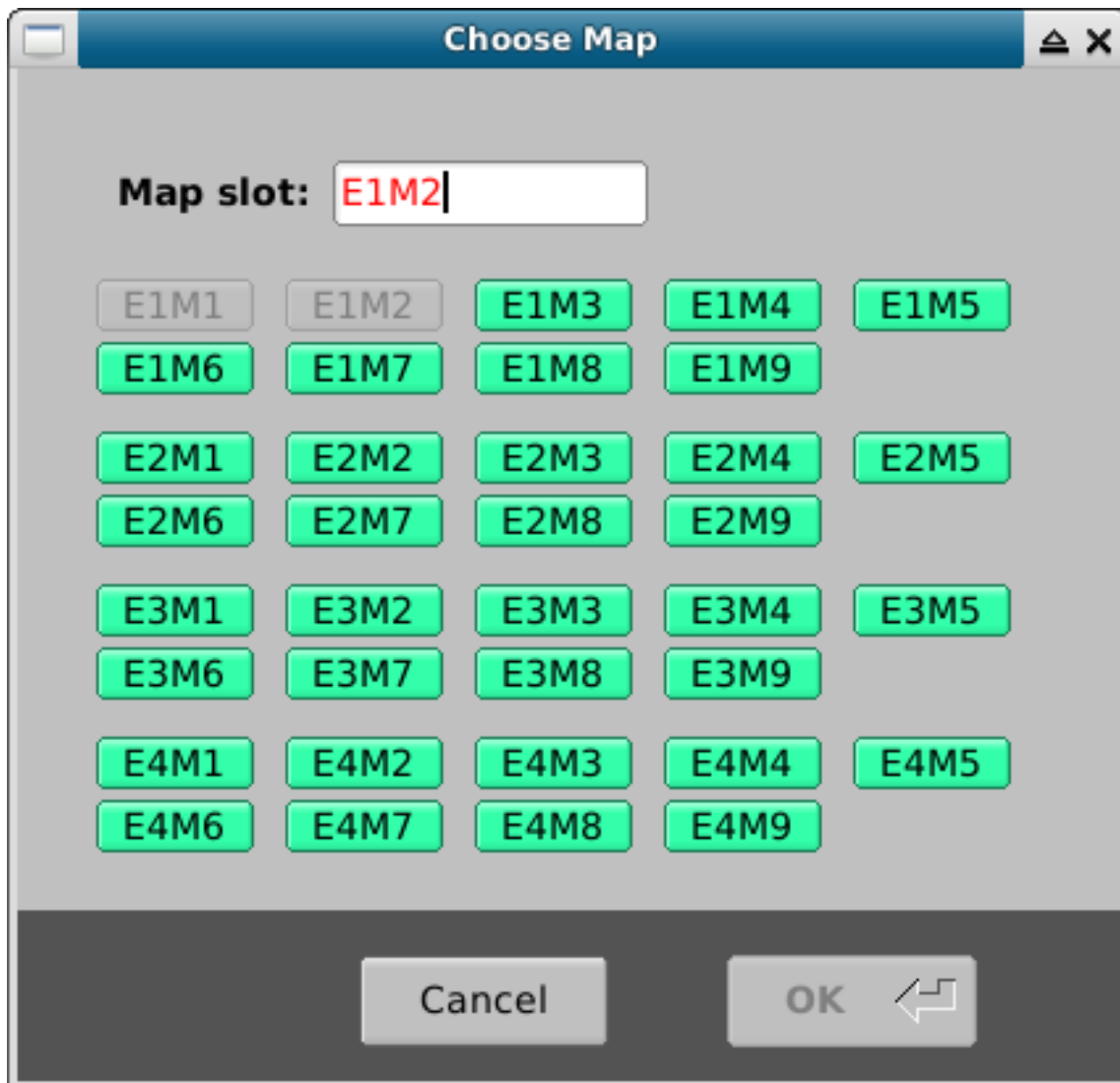


## 4.4 Rename a map

Use the `File -> Rename Map` menu to change the play order of a map, a dialog prompts you to pick the new slot for the map.

Slots with existing maps cannot be selected.





## 4.5 Copy a map

Use the `File -> Copy Map` menu to copy the current map to another slot.

## 4.6 Delete a map

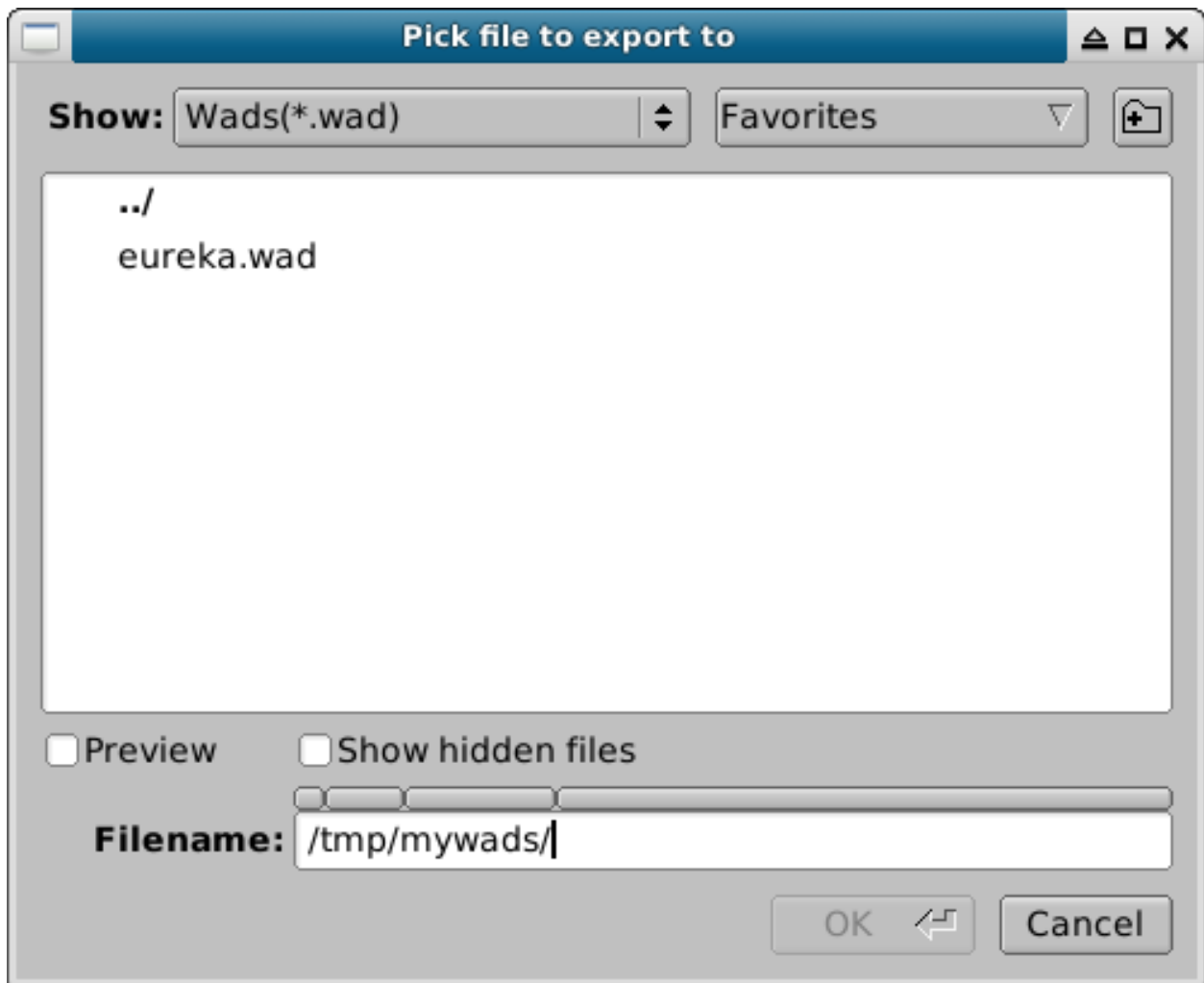
Use the `File -> Delete Map` menu to remove the current map from the WAD. You will be prompted to confirm the delete.

## 4.7 Export a map

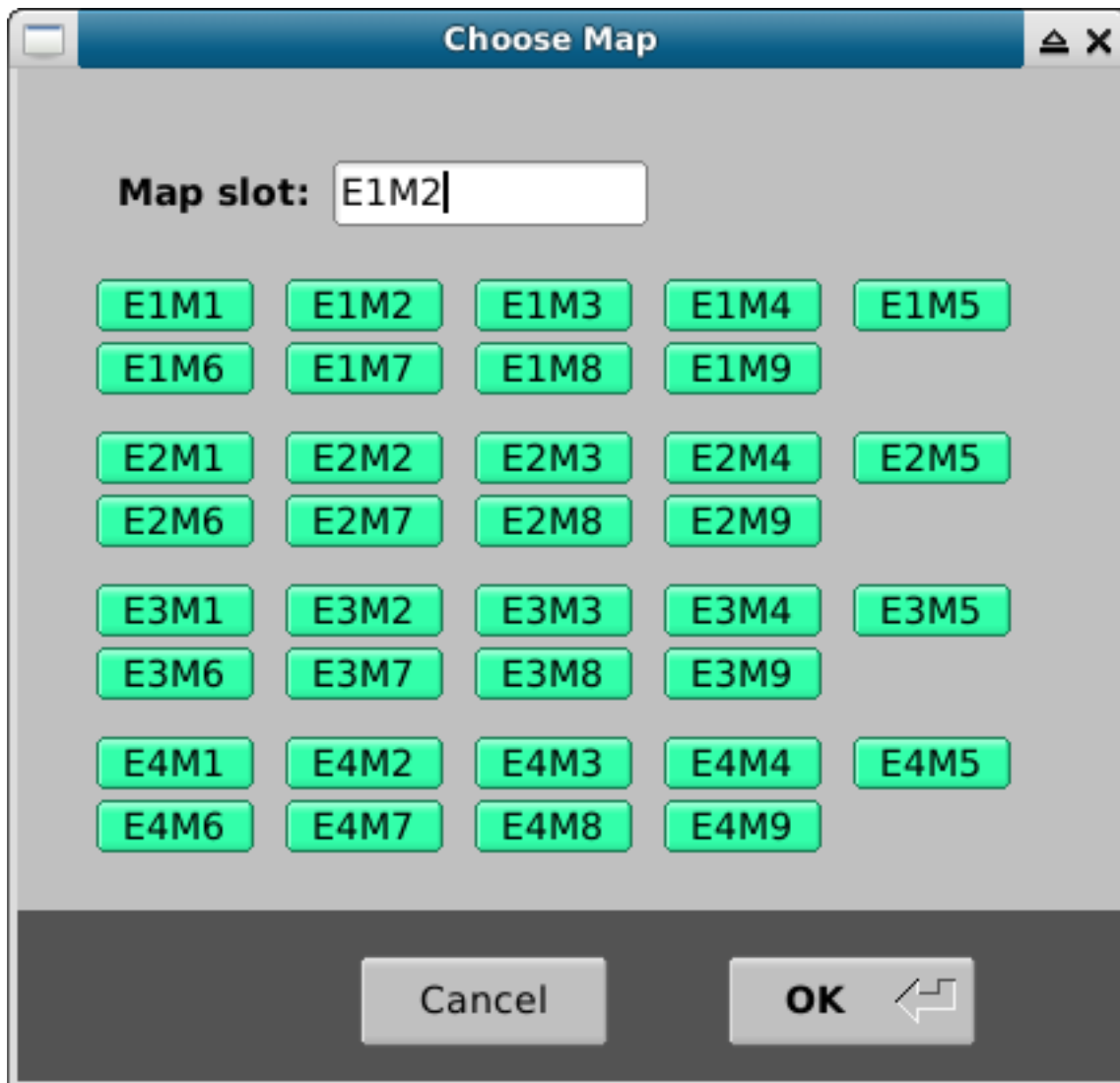
Use the `File -> Export Map` menu to export the current map to:

- a new WAD

- an existing WAD



*Pick the file to export to*



*Select the level of the exported map*



---

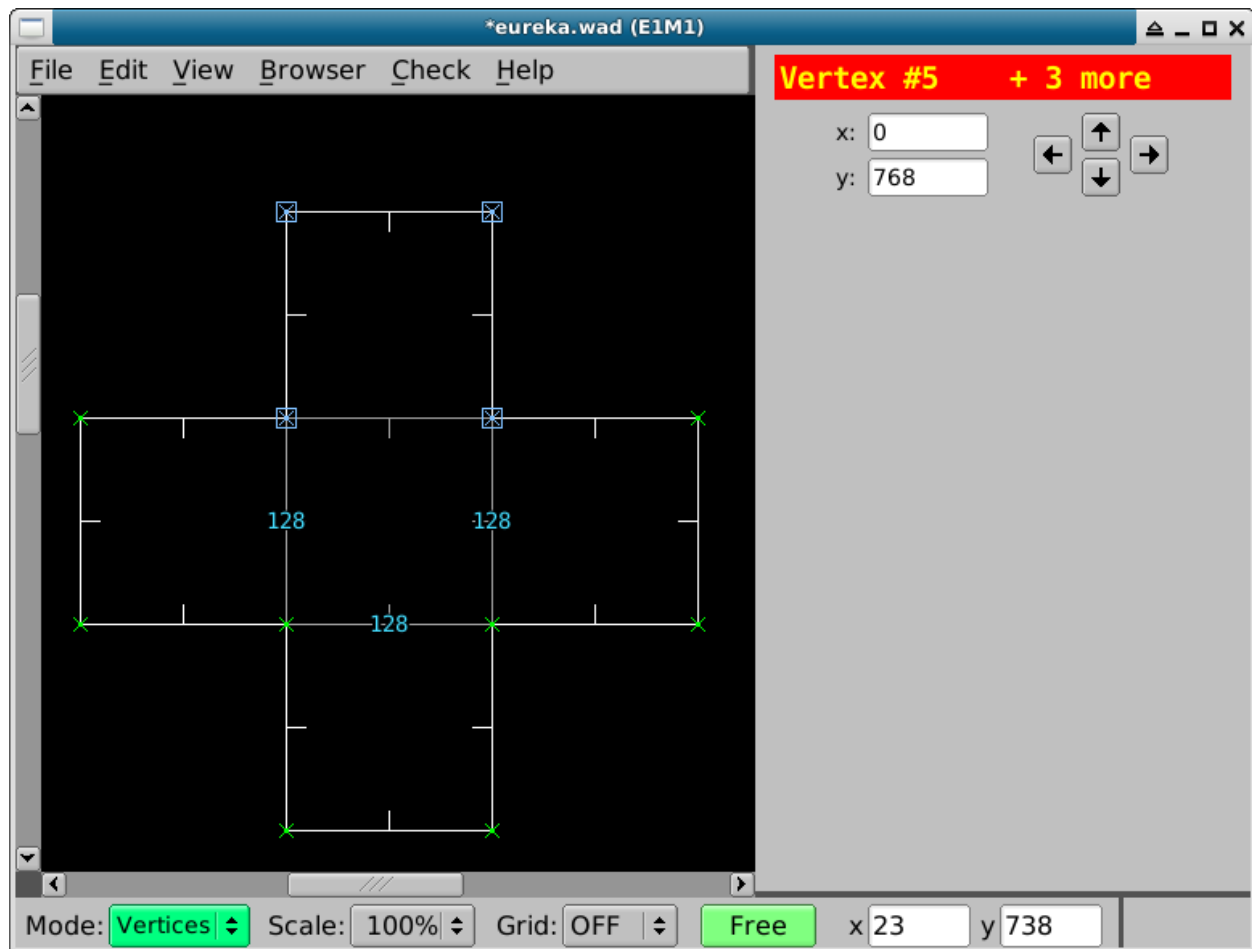
**Note:** LMB: left mouse button click, RMB: right mouse button click

---

### 5.1 Vertices

Vertices are the joins between Lines, and the corners between walls.

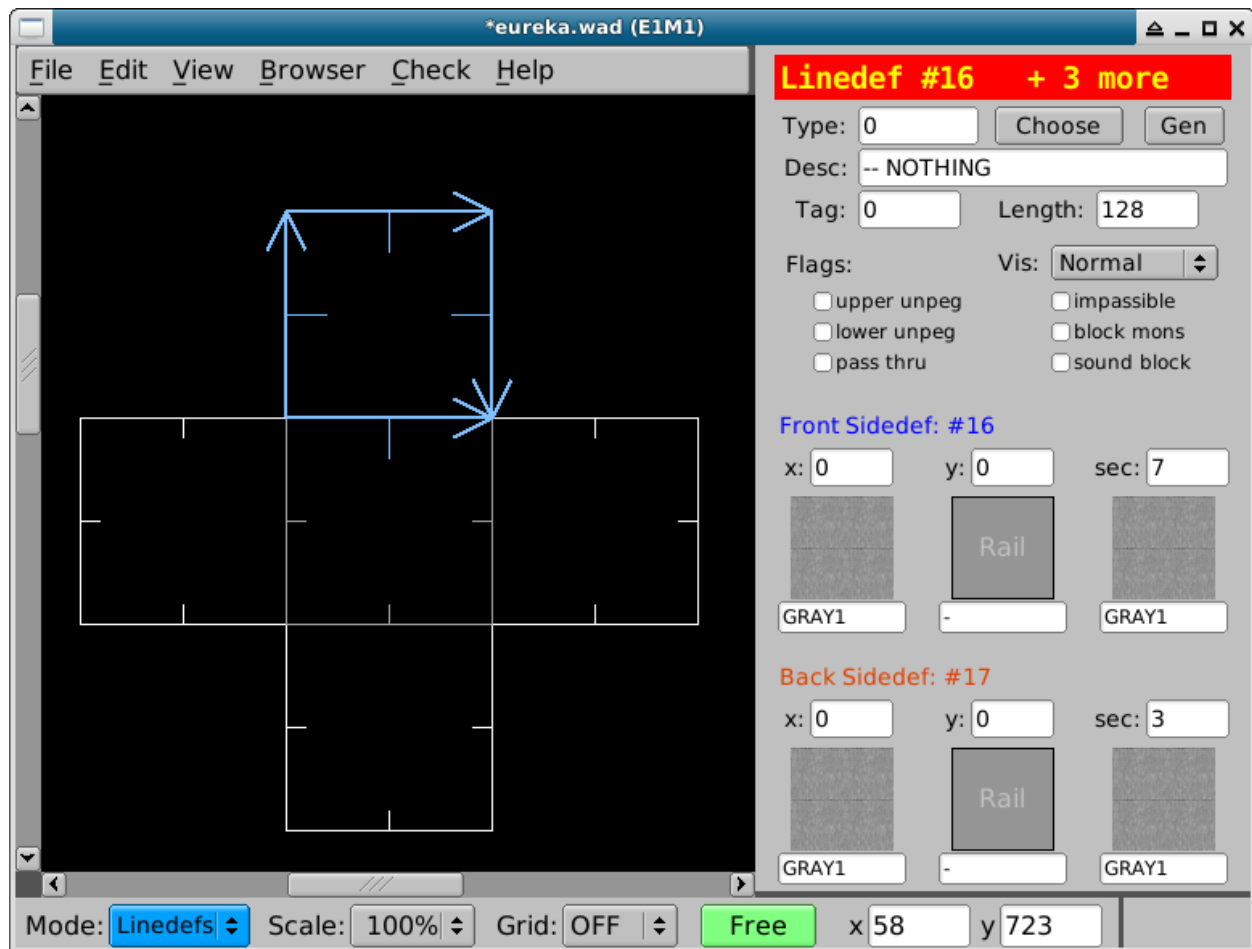
- Press `v` in the 2D view to enter Vertices edit mode
- LMB while hovering over a Linedef inserts a single vertex at the cursor position
- RMB inserts a vertex in **line drawing** mode, keep adding vertices with RMB until you close the polygon



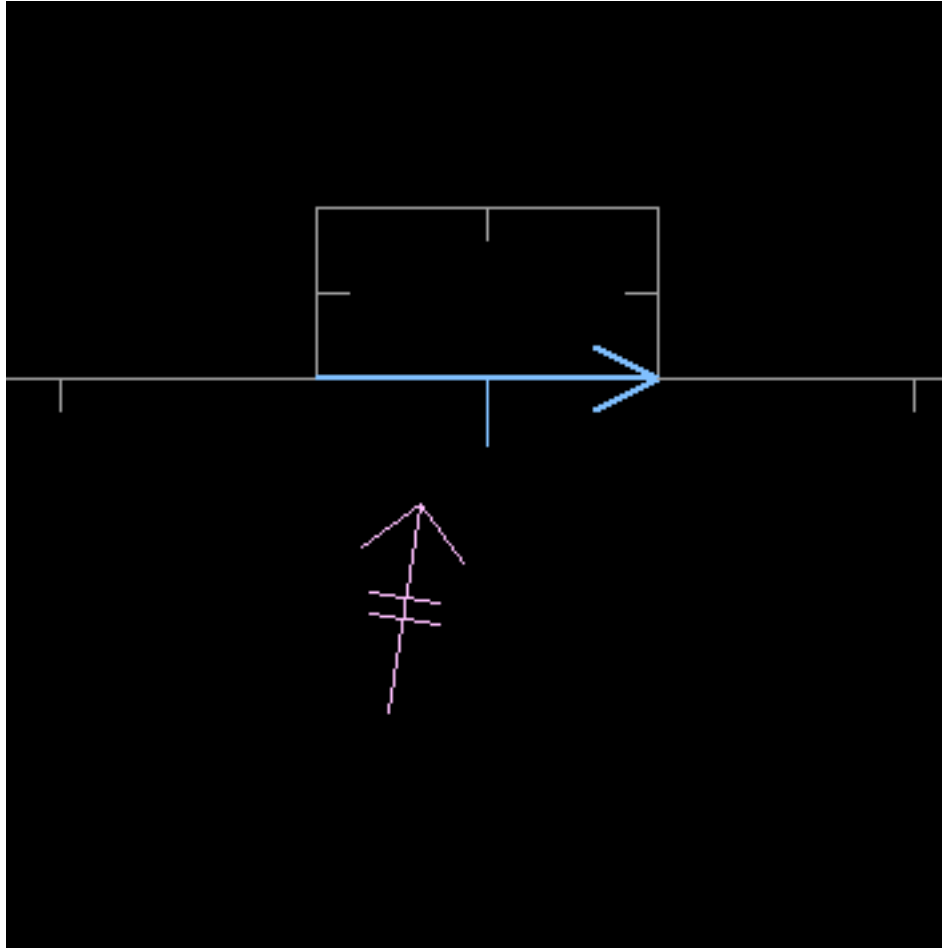
## 5.2 Linedefs

Linedefs divide the map into sectors, and they are also used to trigger actions.

Press 1 in the 2D view to enter Linedef edit mode.



A linedef also has a front and a back. You can tell the front of a Linedef by the direction of the protruding pin in the middle of the Linedef.



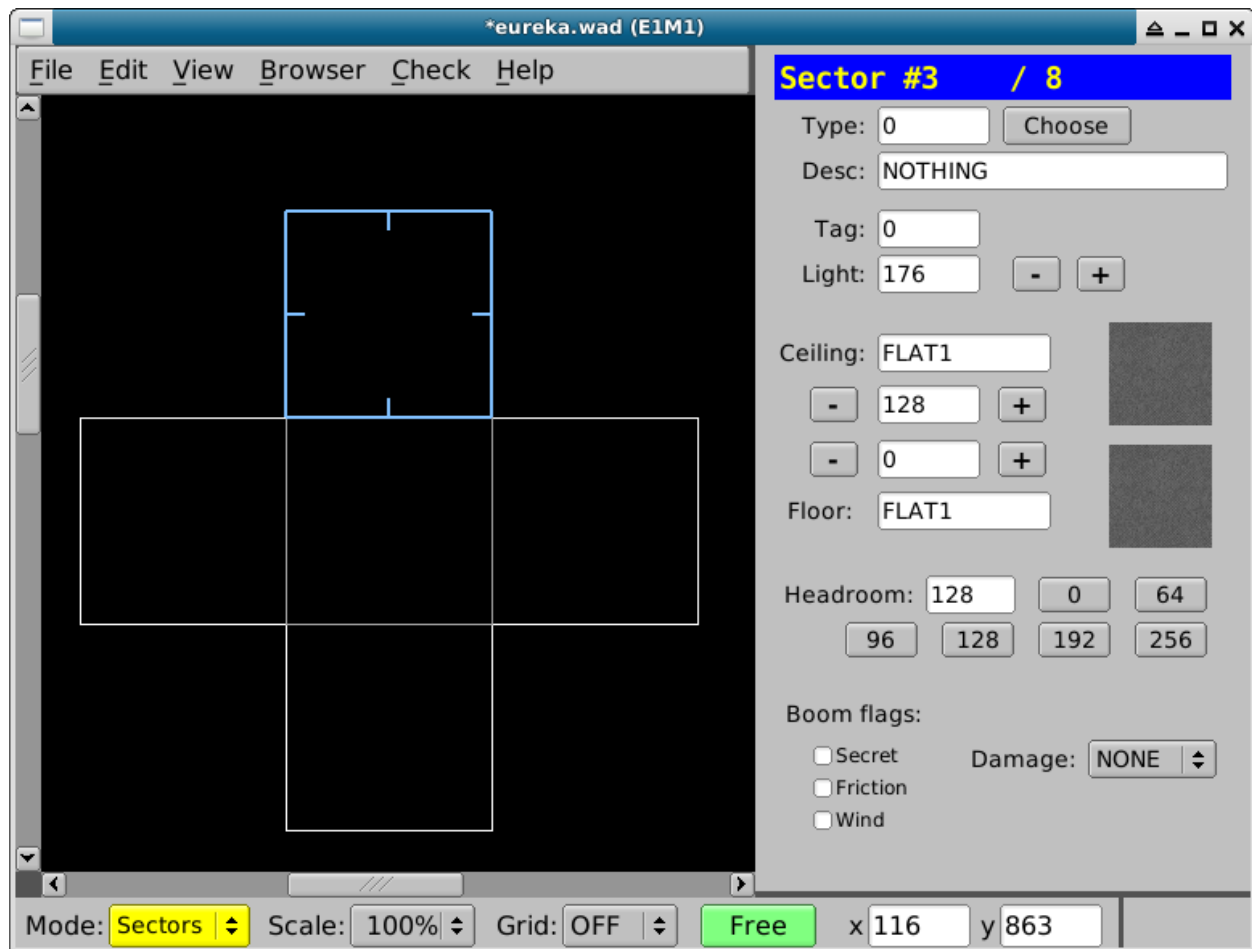
*this linedef front is facing down*

## 5.3 Sectors

A sector is an area defined by multiple Linedefs.

Press `s` in the 2D view to enter Sector edit mode.

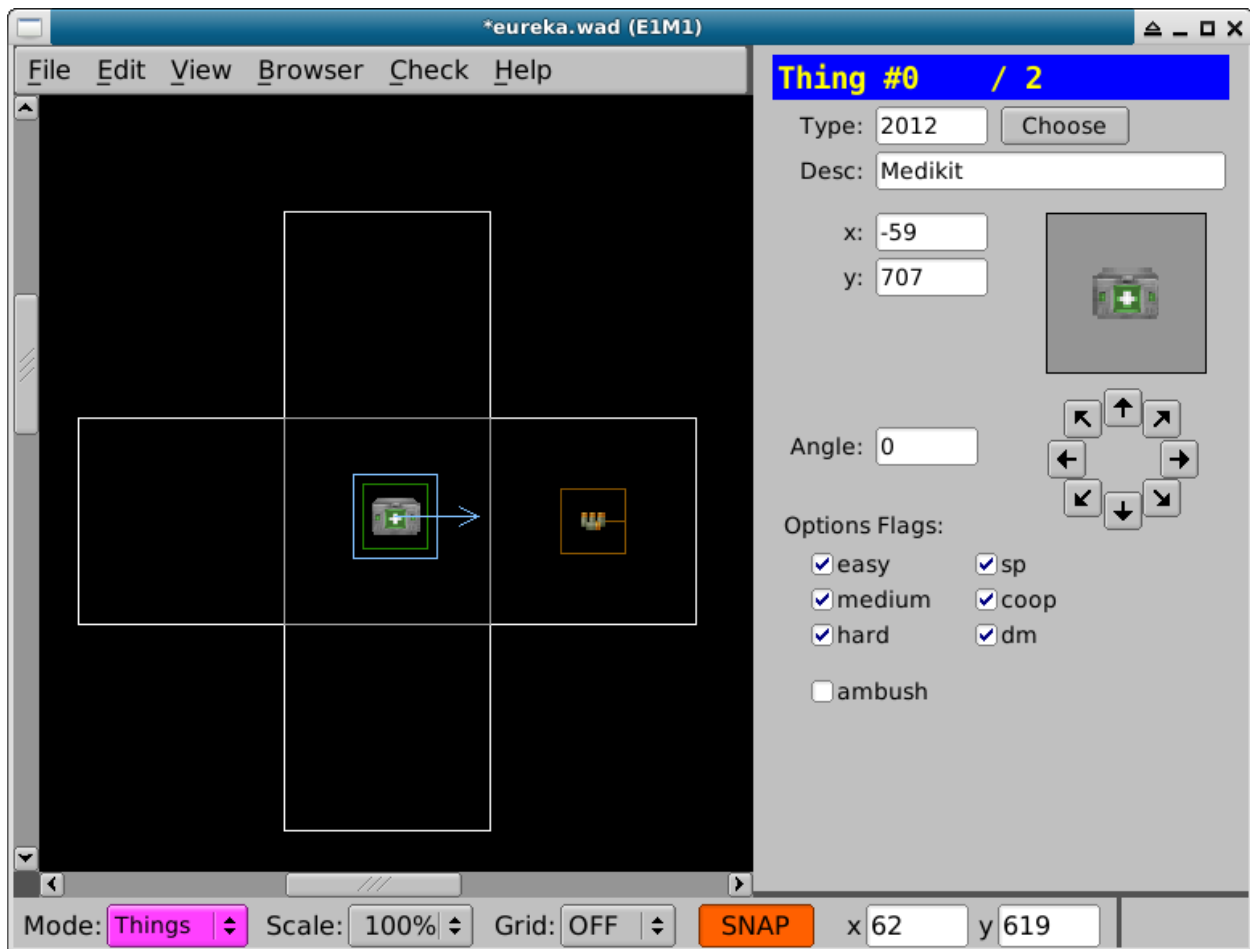




## 5.4 Things

Things represent players, monsters, pick-ups, obstacles, decorations, player start positions and teleport landing sites.

Press `t` in the 2D view to enter Things edit mode. `space` or `ins` inserts a new thing at the cursor position.



## 5.5 Multiple selections

While in the 2D view, you can select Vertices, Linedefs, Sectors and Things by:

- Clicking an item to select/deselect it
- Drag-select multiple items to include/exclude them in the selection
- Hold `shift` to prevent moving things around when making a selection

Switching between Vertices, Linedefs and Sectors will preserve any selections as much as possible.

## 5.6 Deselecting

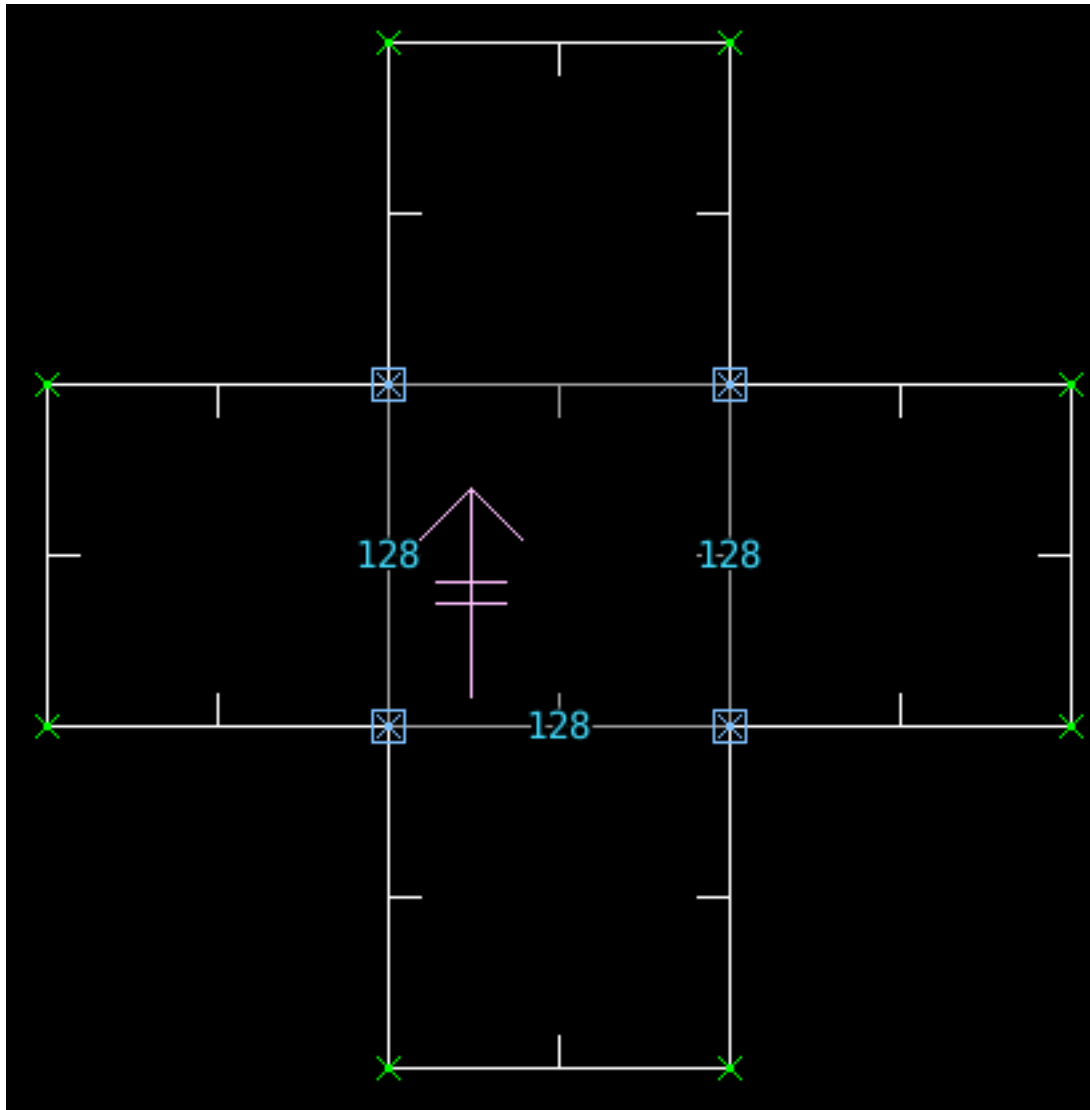
Press ``` (back quote) in the 2D view to deselect everything.

## 5.7 Transformations

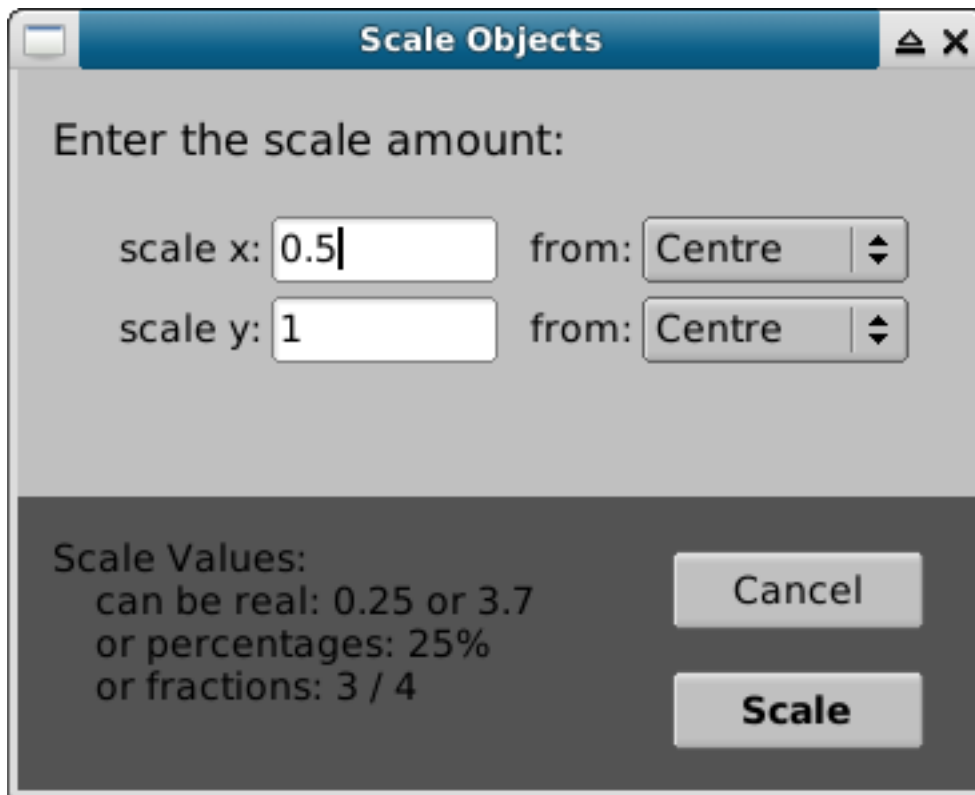
You can scale and rotate selected Sectors, Linedefs and Vertices.

### 5.7.1 Scaling

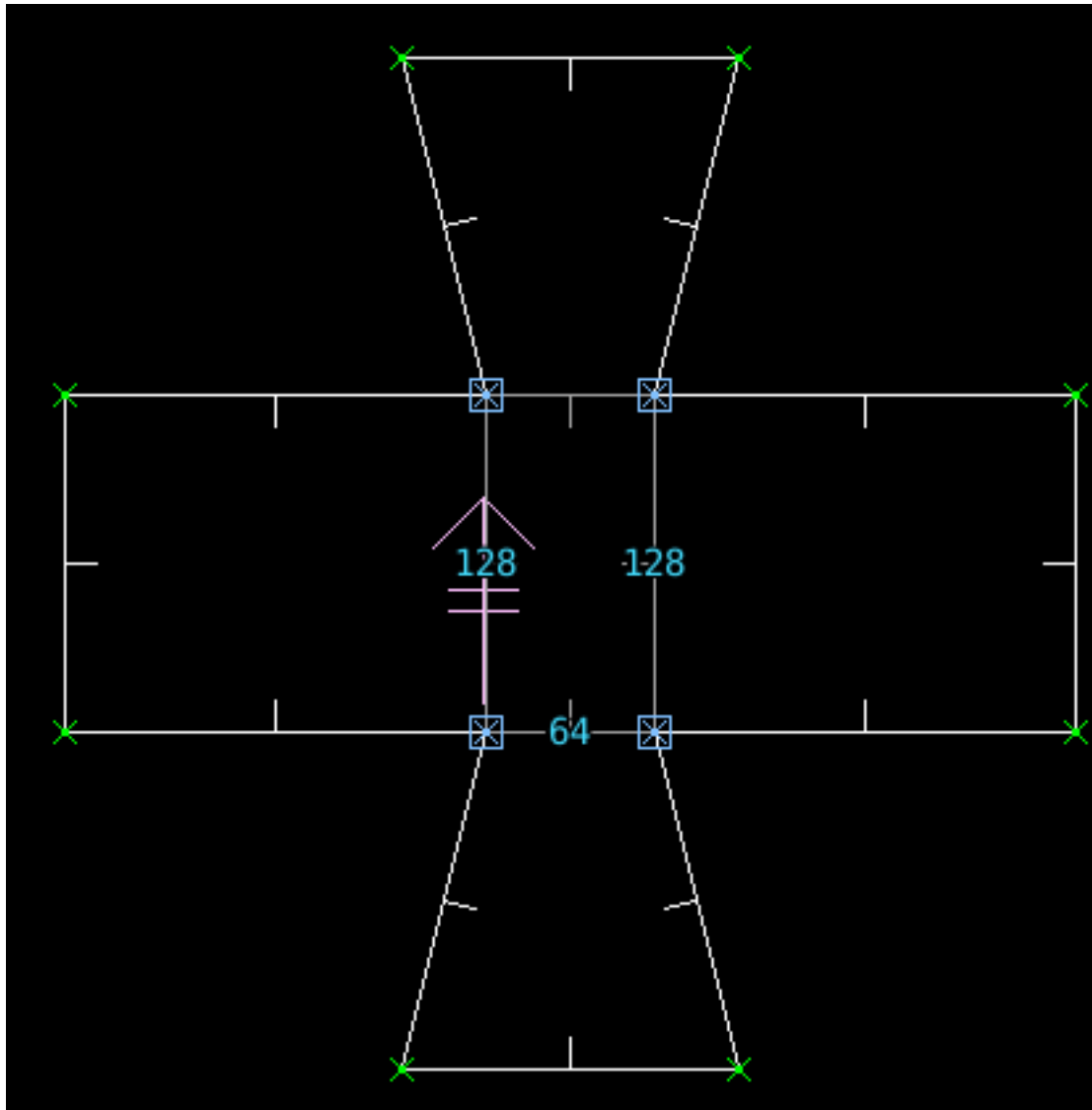
Click on Edit -> Scale Objects or press F3 to show the scale dialog.



*Four inner most vertices are selected*

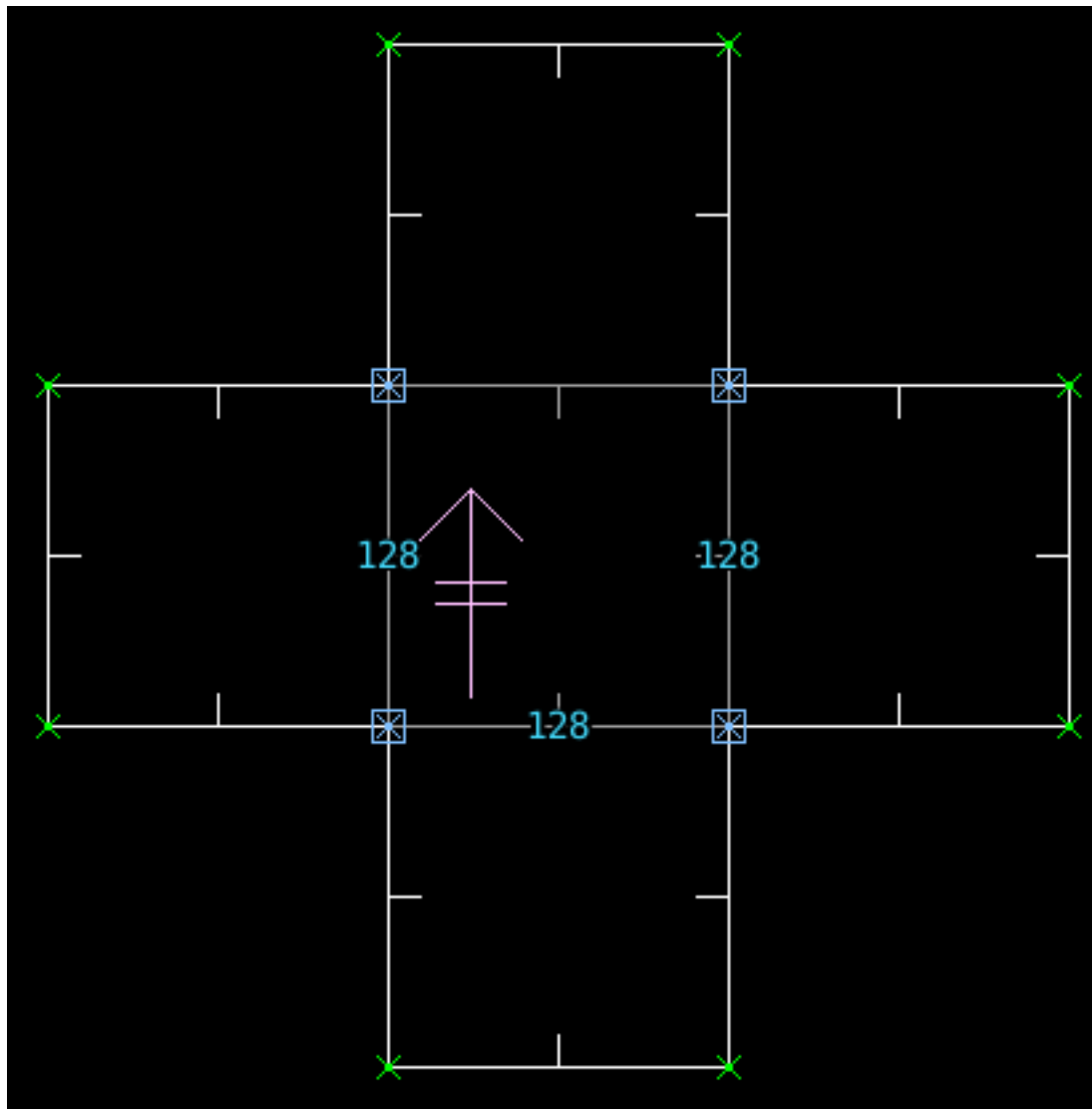


*Scaling the vertices by 50% on the x-axis*

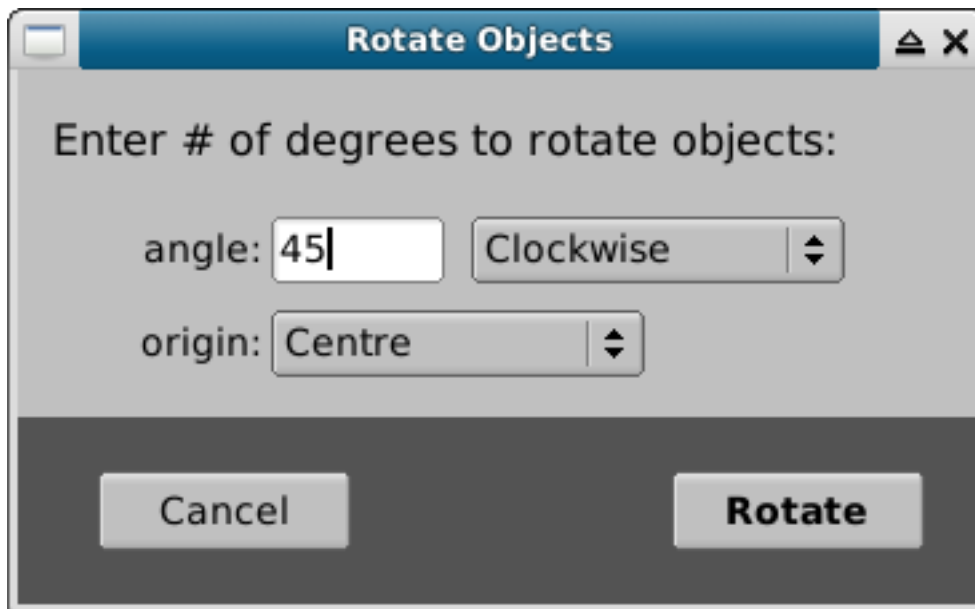


### 5.7.2 Rotating

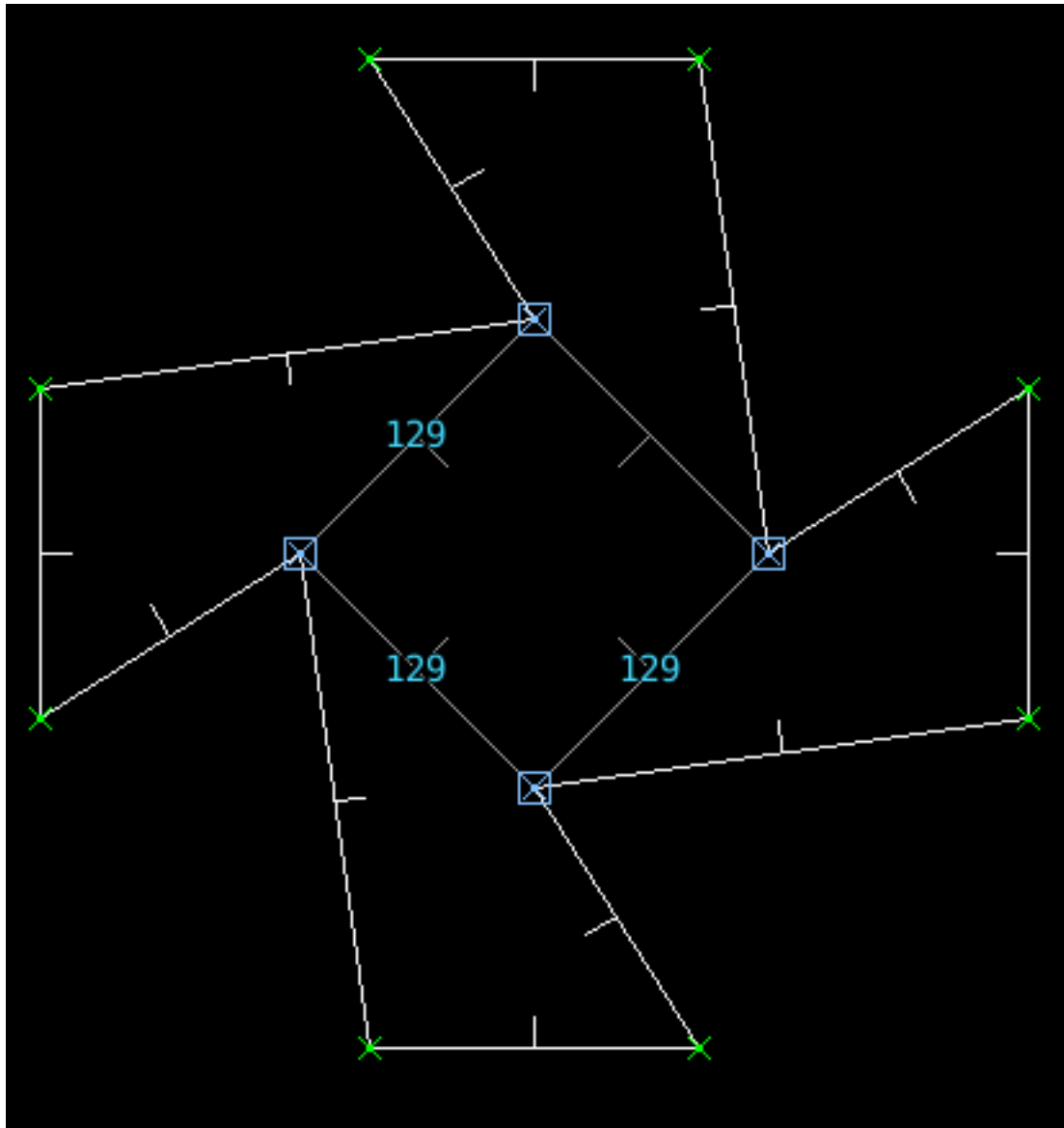
Click on Edit -> Rotate Object or press F4 to show the rotate dialog.



*Four inner most vertices are selected*



*Rotating the vertices by 45 degrees*



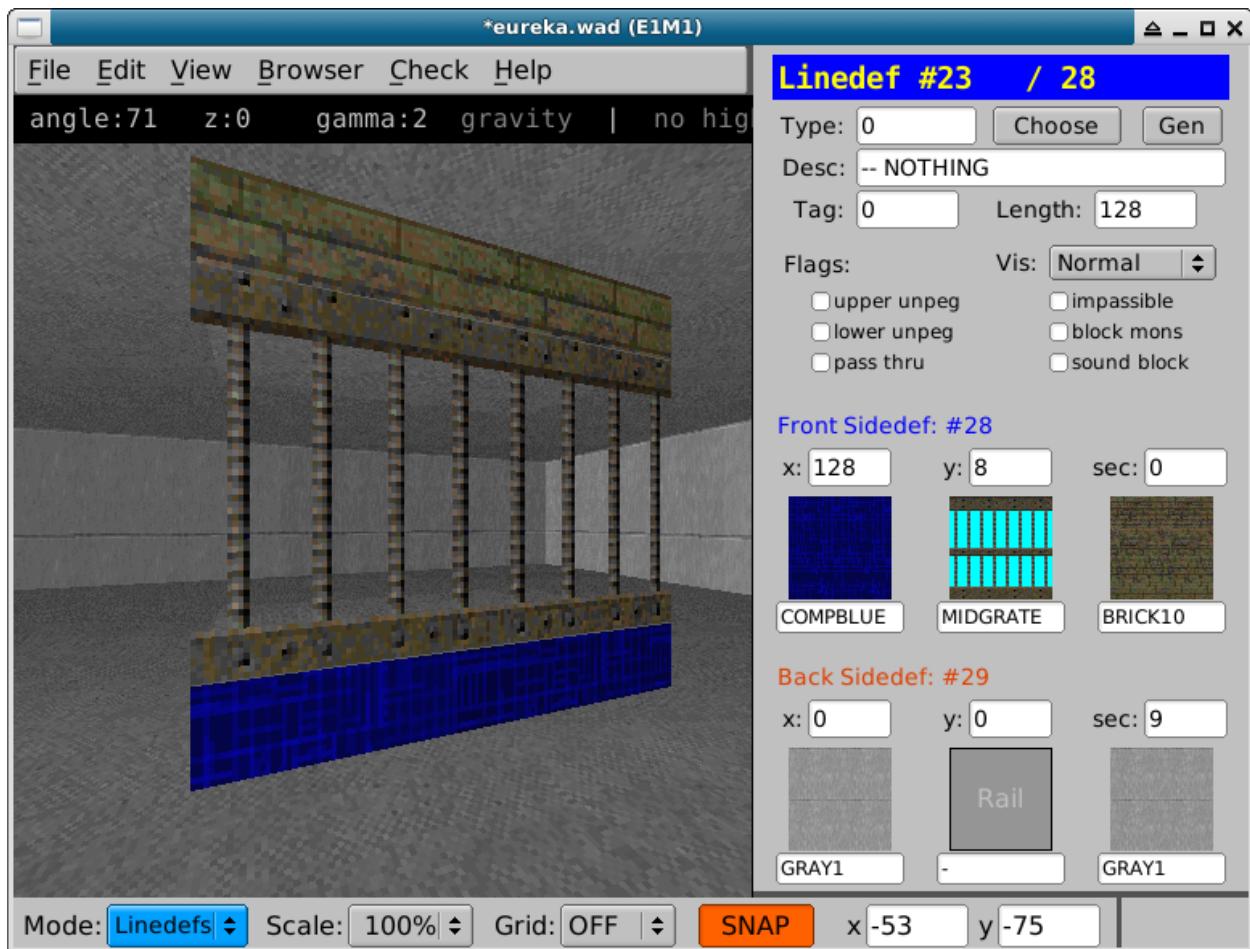
## 5.8 Textures

### 5.8.1 Sidedefs

A Sidedef refers to the texture data for a Linedef, it can have **lower**, **middle** and **upper** textures.

The relation of the sidedefs are best illustrated with these poorly-matched textures:

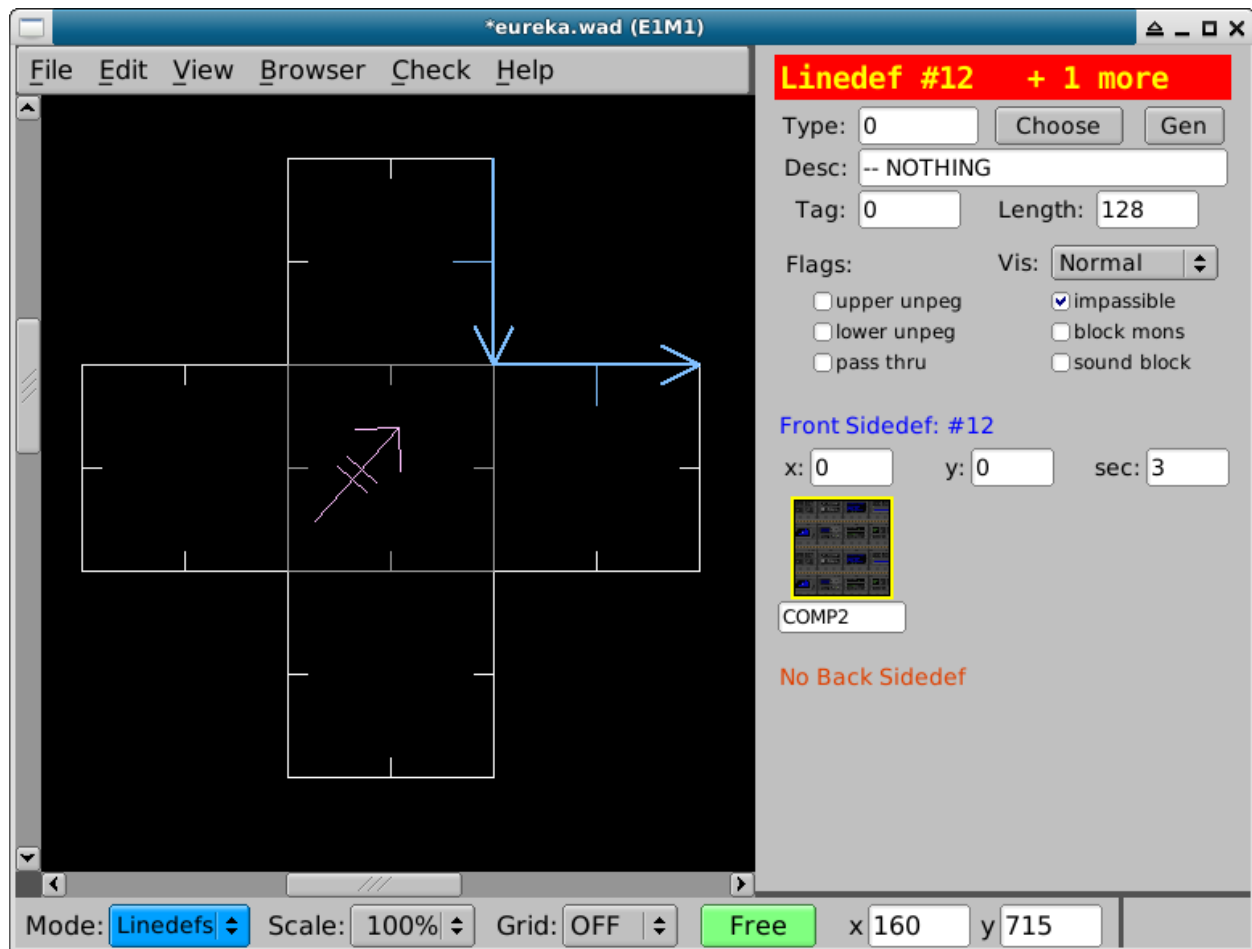




*Lower texture is COMPBLUE, middle is MIDGRATE, and upper texture is BRICK10*

#### To change the sidedefs

- Switch to 2D mode
- Enter Linedef mode (1)
- Select one (or more) linedefs

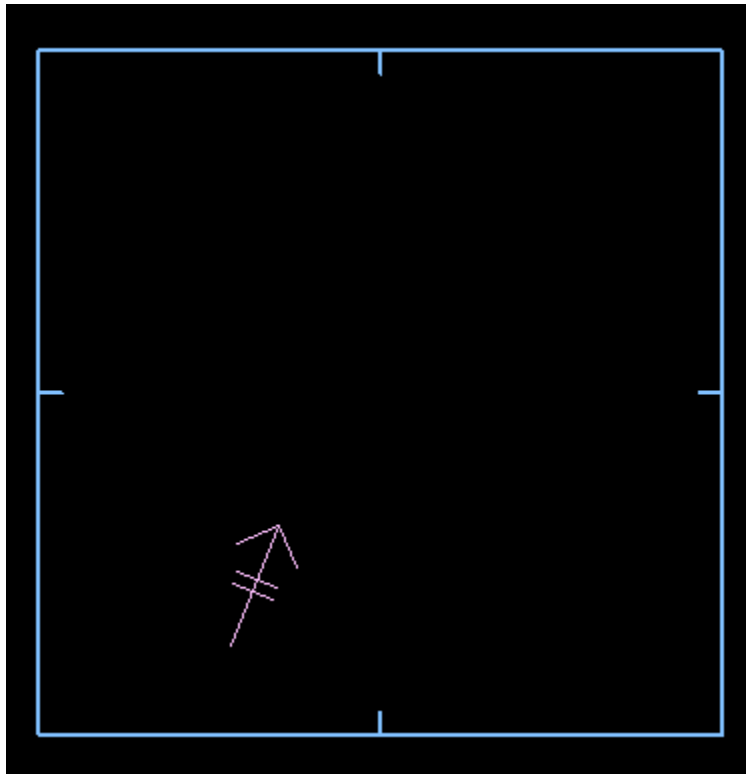


- Click the “Front sidedef” texture button to open the texture browser
- (optionally) Press `tab` to switch to 3D view for a live preview
- Select a texture in the browser window

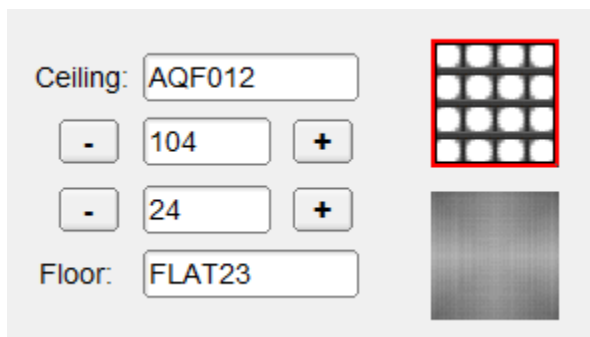


### 5.8.2 Floor / ceiling textures

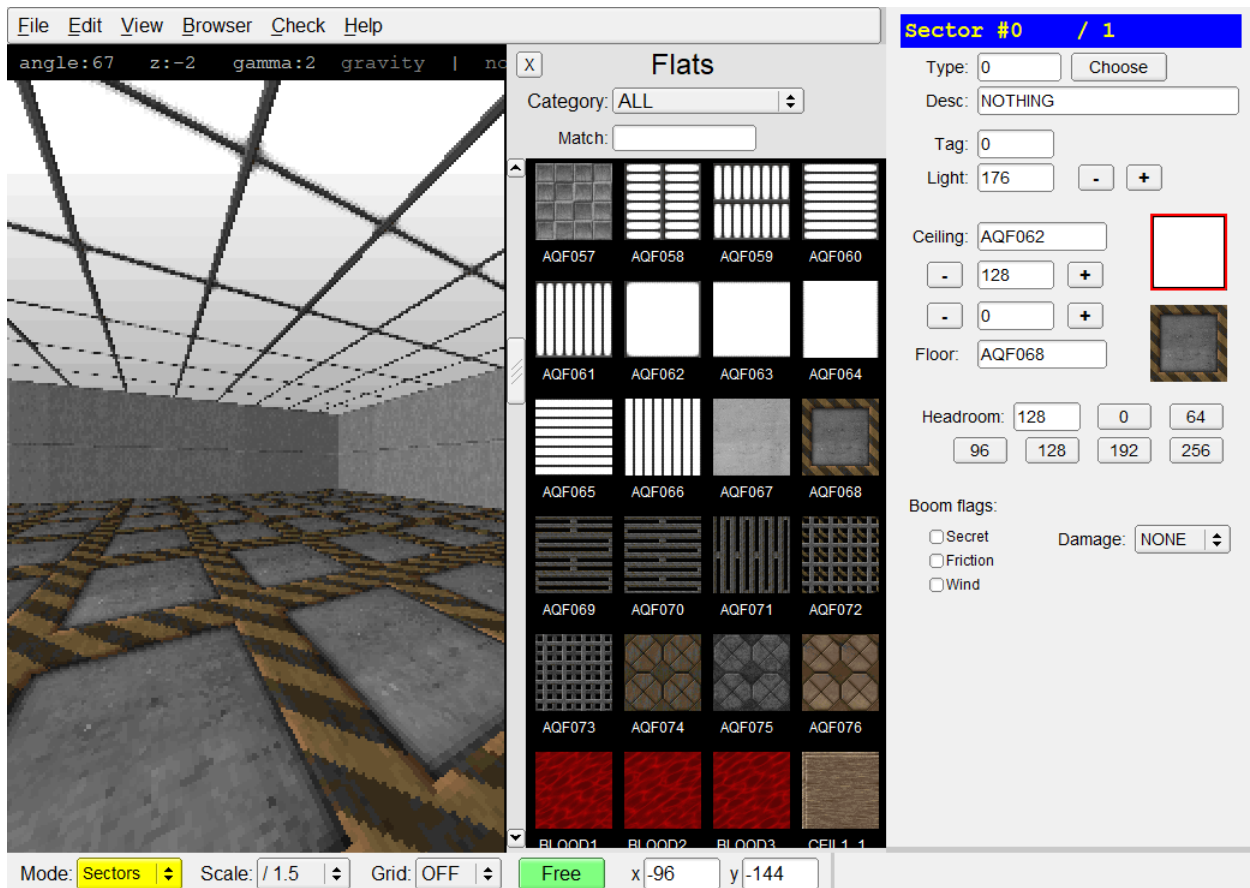
Enter Sector editing mode and select the sector to change:



Click the texture buttons to open the texture browser:



You can enter the 3D view to see a preview of the texture changes:





---

**Note:** LMB: left mouse button click, RMB: right mouse button click

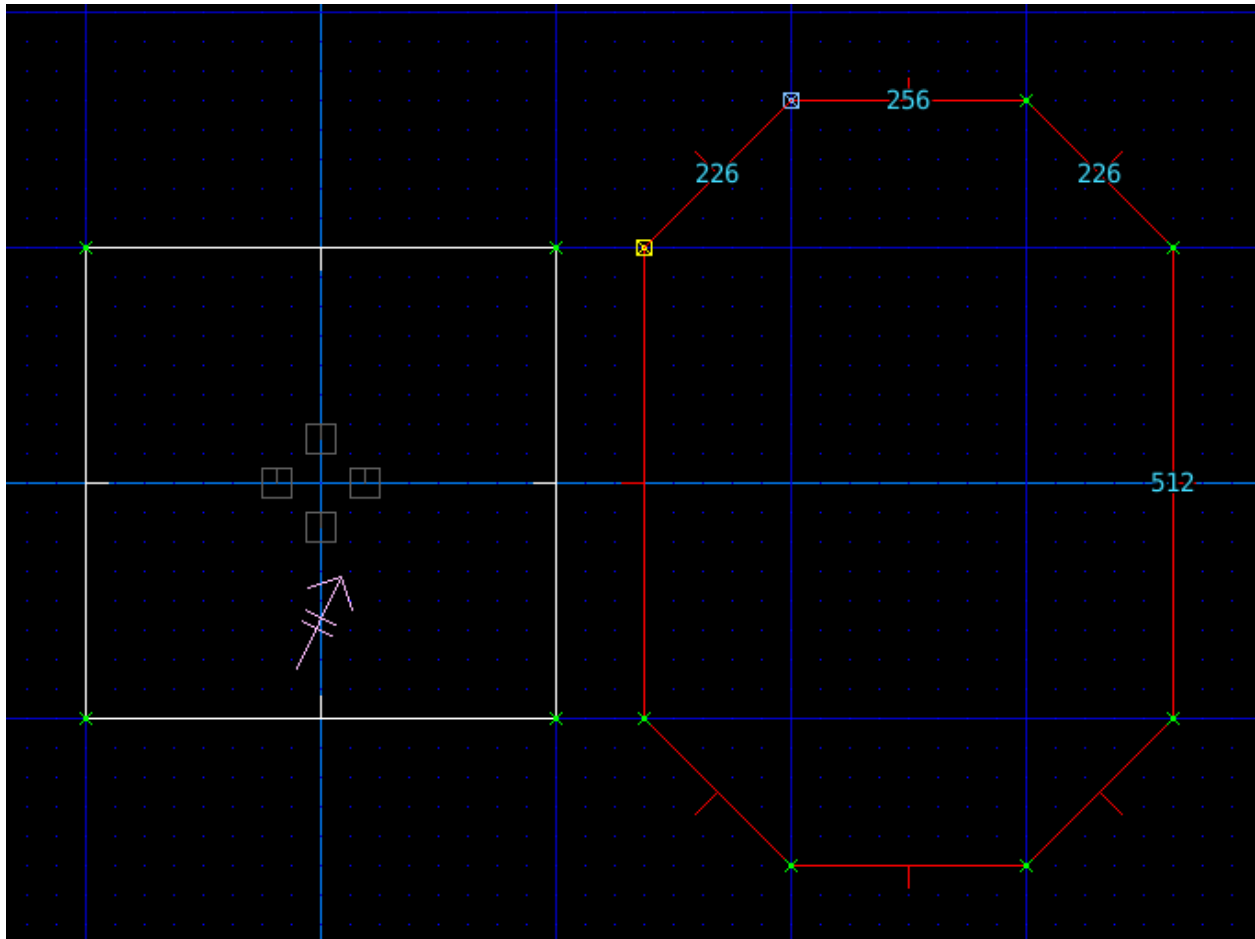
---

To follow these building basics, create a new project in Eureka.

### 6.1 Adding a room

To help with drawing, press `f` to toggle grid snapping, press `g` to enable grid display, and press `5` to set the grid size to 32 units.

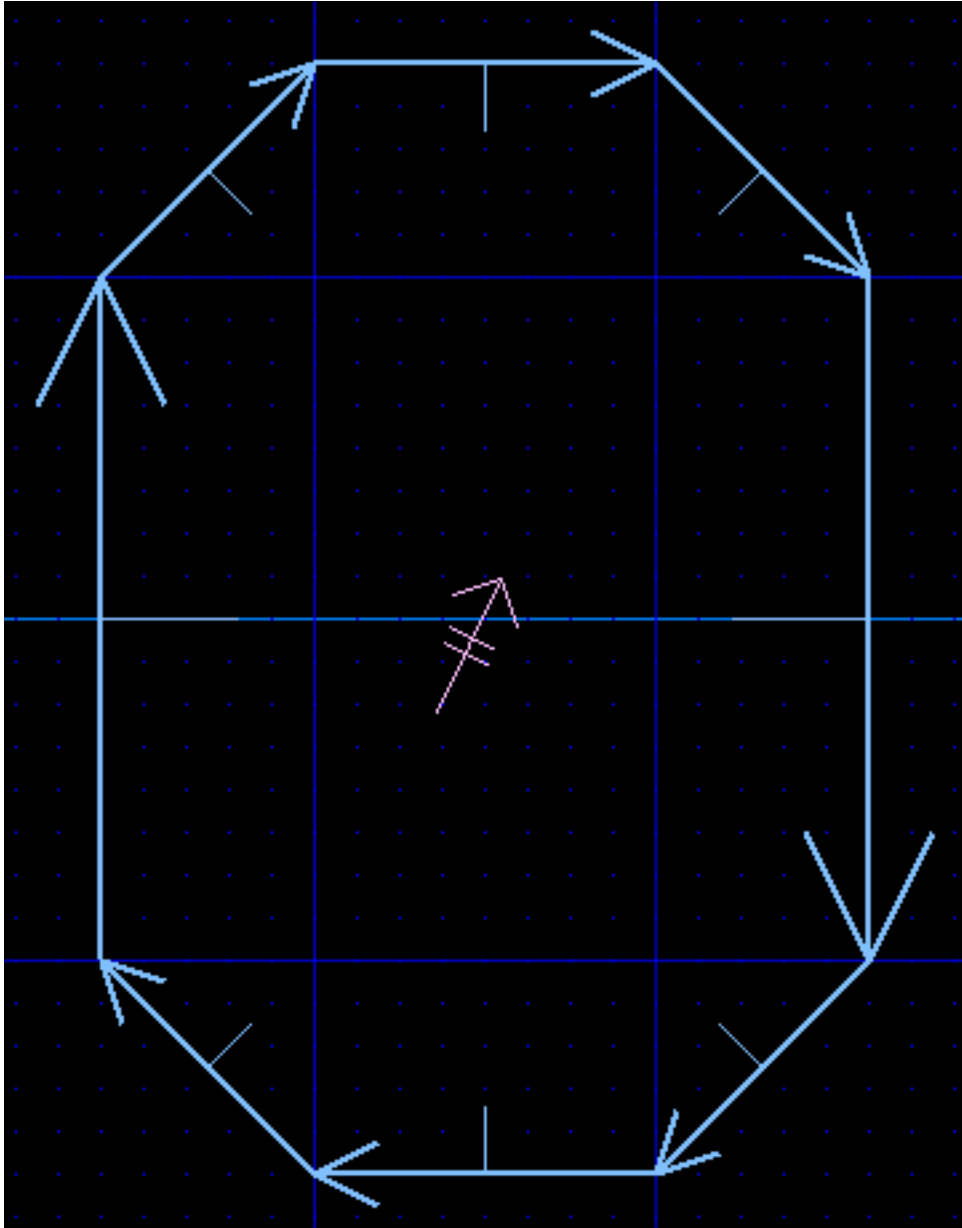
- Press `tab` to switch to the 2D view
- Press `v` to enter vertices edit mode
- Use the RMB to draw a new sector. Make sure to keep adding vertices with the right mouse button, and to close the sector.



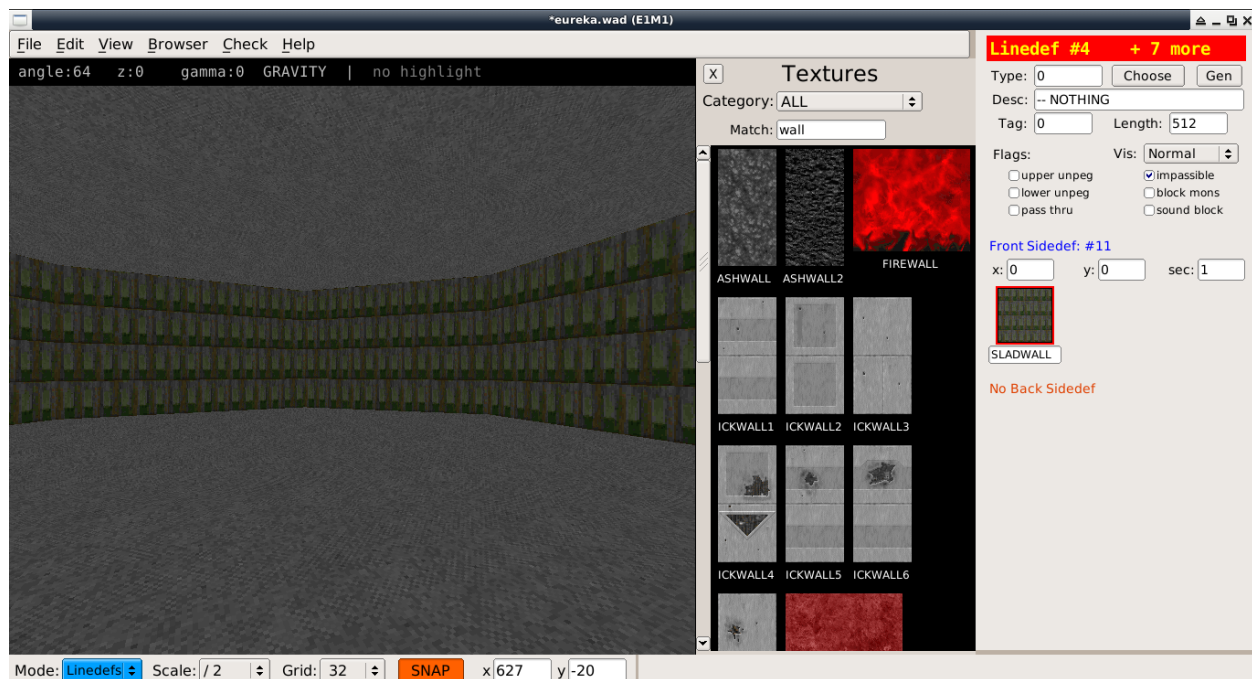
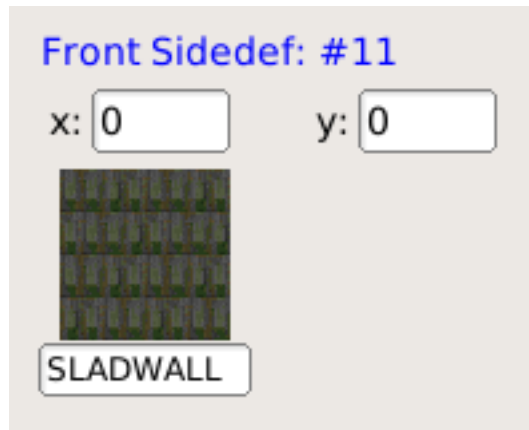
### Texturing the new walls

Press **1** to enter linedef edit mode. The new linedefs should be selected, if not click with the LMB and drag a box around them to select.



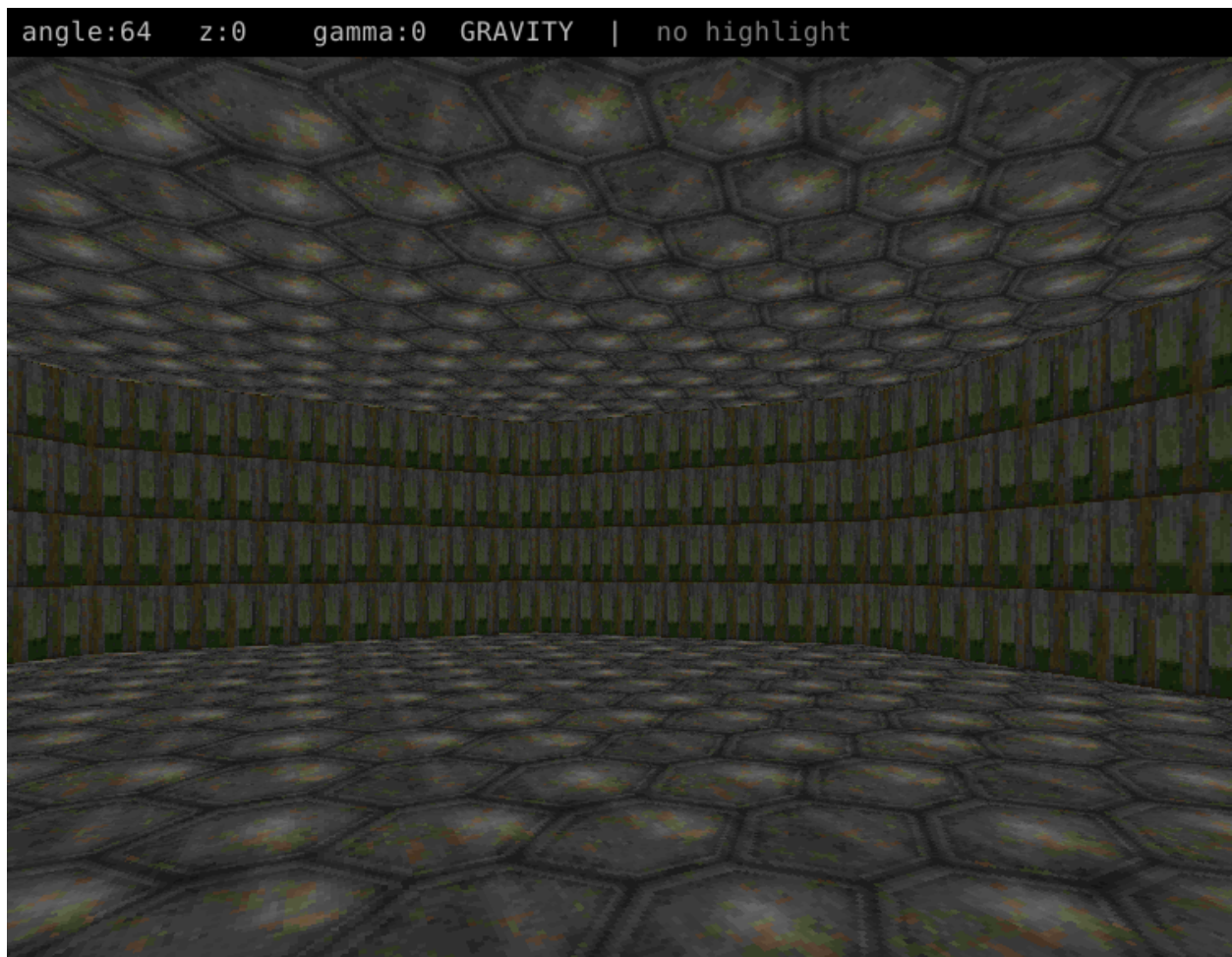


- Position the mouse cursor inside your new sector and press ' ' to place the camera inside the room.
- Press `tab` to enter 3D view
- Click the front sidedef texture button and pick the `SLADWALL` texture.



### Texturing the new floor and ceiling

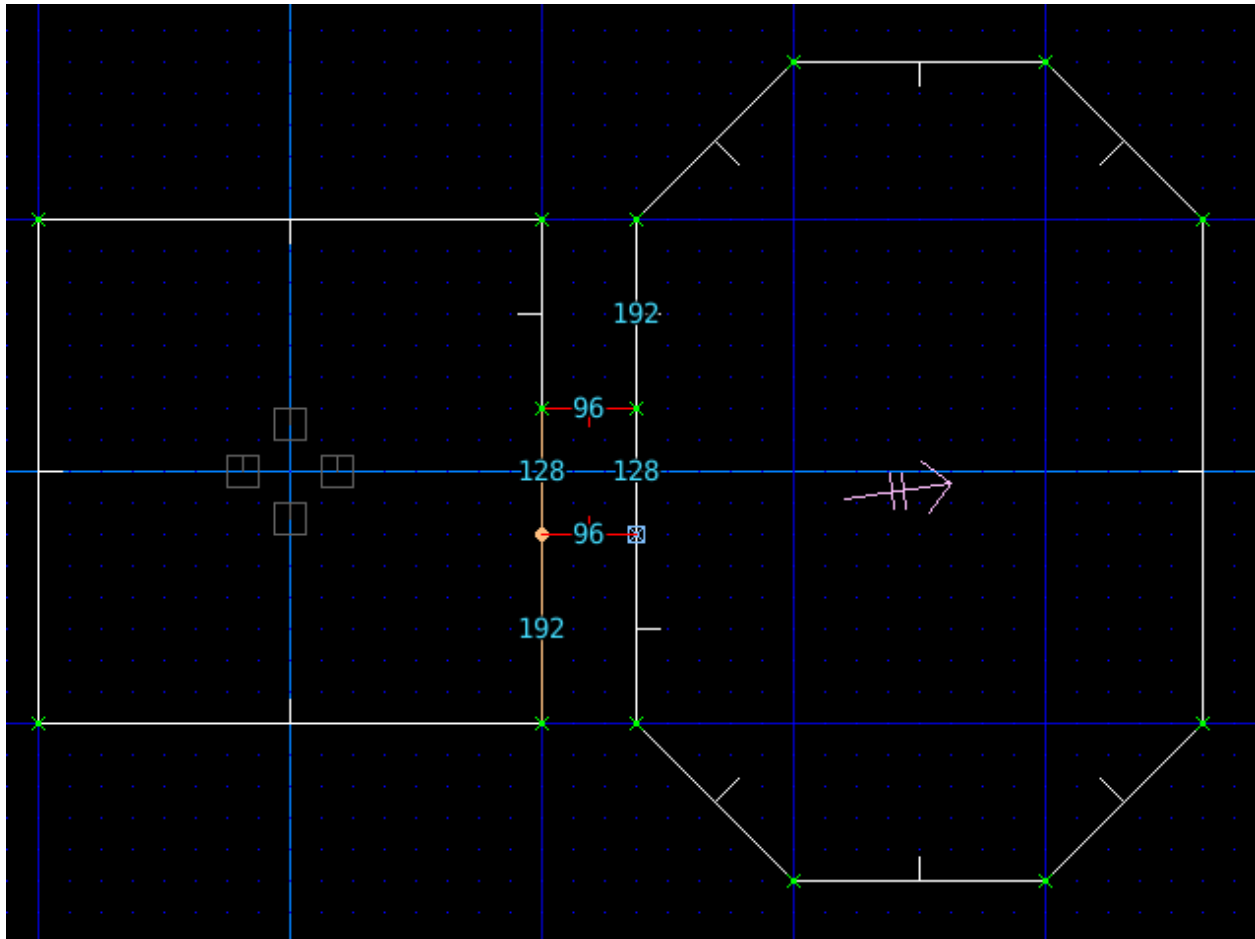
- Press `tab` to return to 2D view
- Press `s` to enter sector edit mode. The new sector should remain selected, if not click the LMB to select it.
- Press `tab` to return to the 3D view
- Click on both the **Floor** and **Ceiling** texture buttons, and select the FLOOR5\_1 texture.

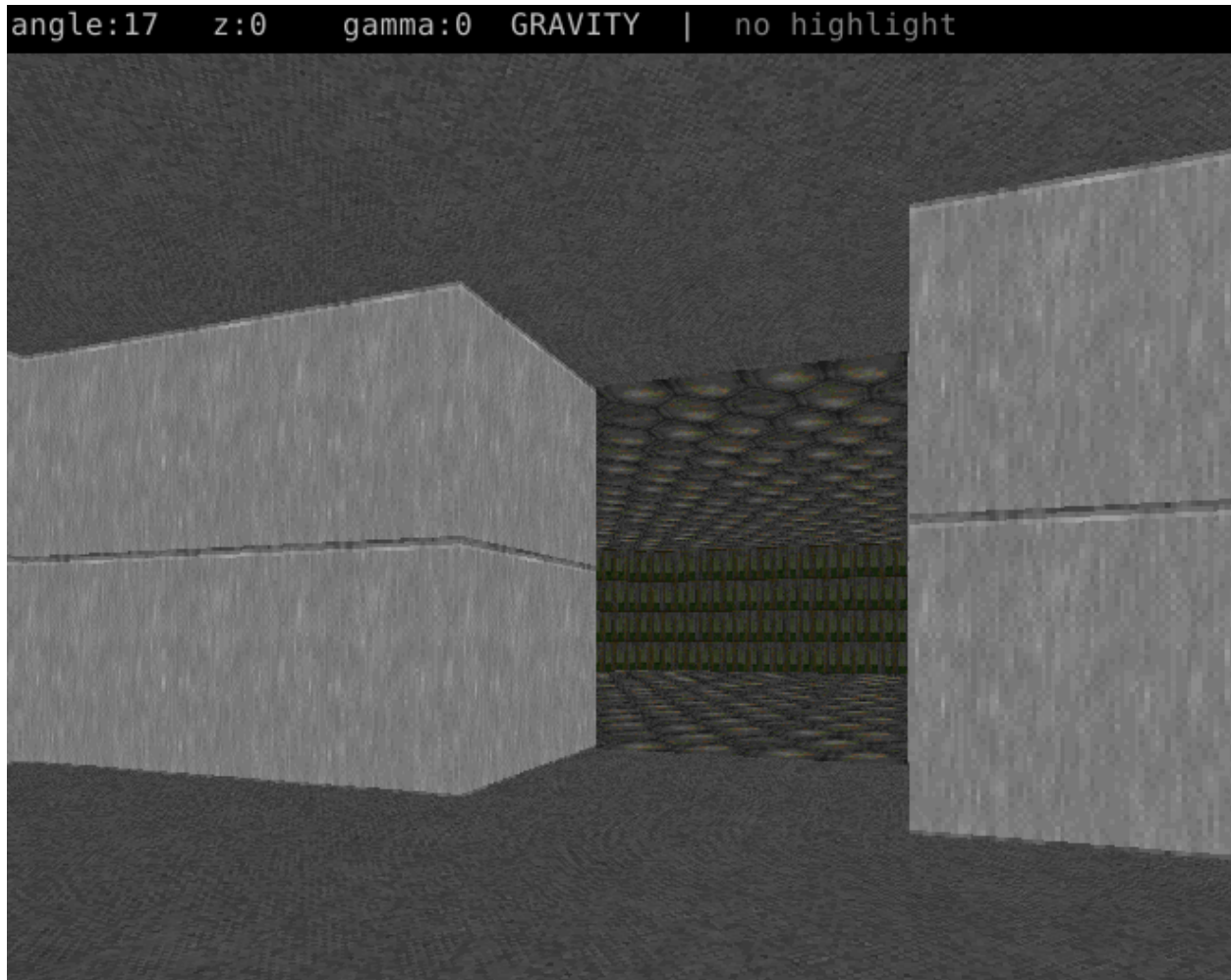


## 6.2 Joining rooms

- Press `tab` to enter 2D view
- Press `v` for vertices edit mode
- Click with the `RMB` to insert vertices that bridge the neighbouring sectors

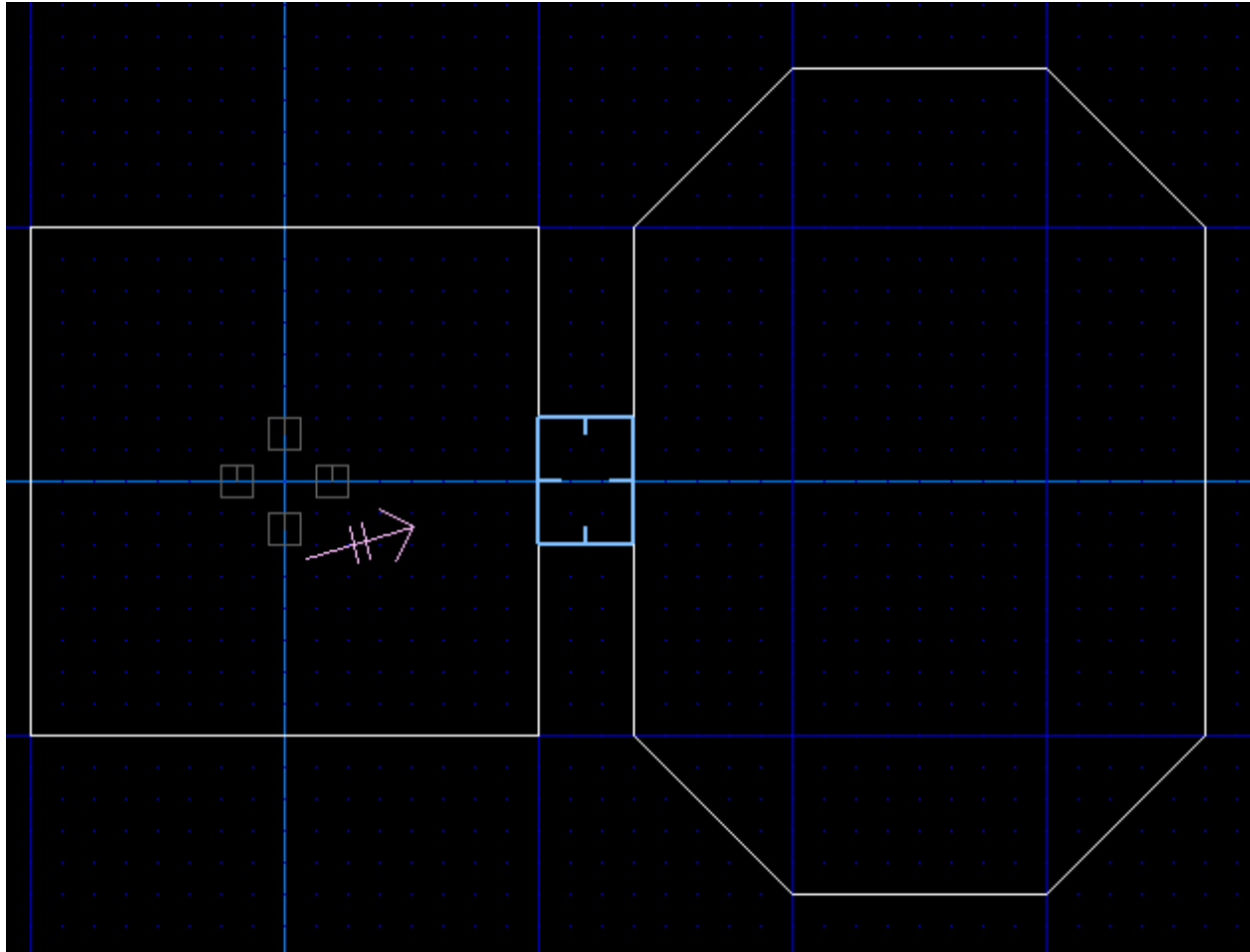
- ## Chapter 6. Building Basics



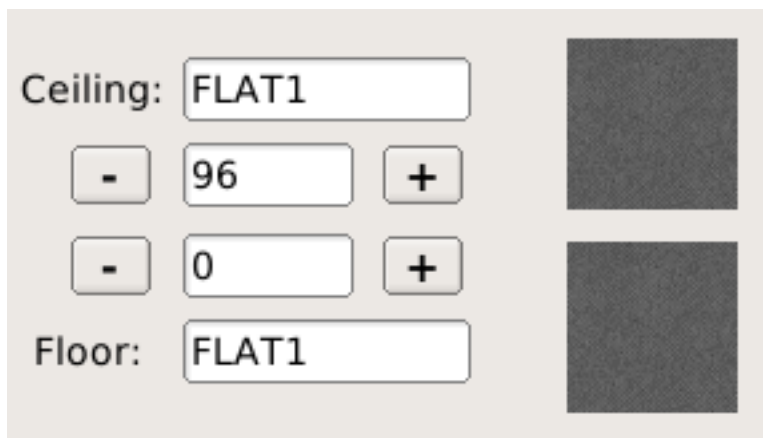


## 6.3 Adjusting ceiling height

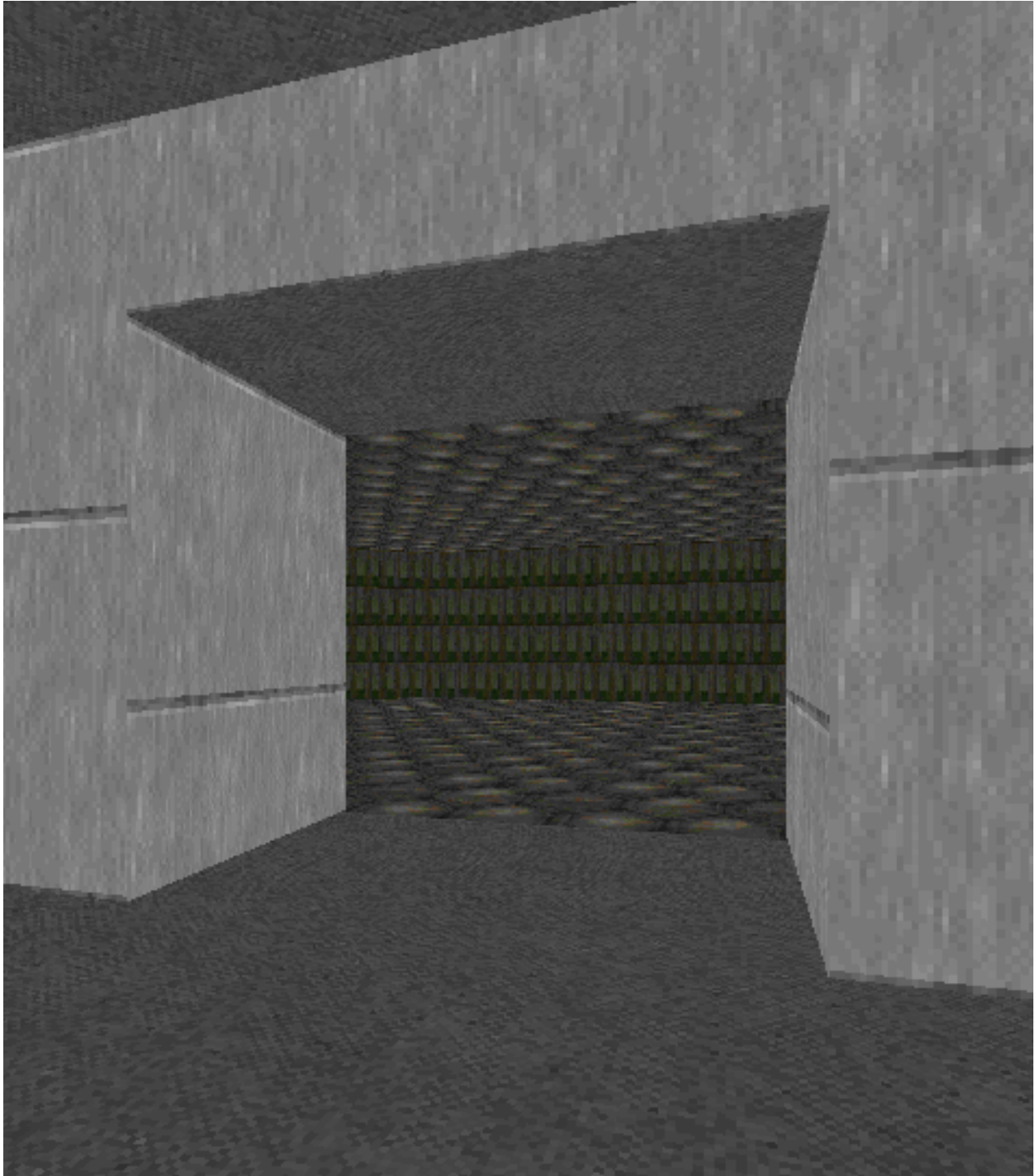
- Press `tab` to enter 2D view
- Press `s` for sector edit mode
- Select the sector you want to adjust



- Press `tab` to return to 3D view
- Click the ceiling `+-` buttons to adjust the ceiling height



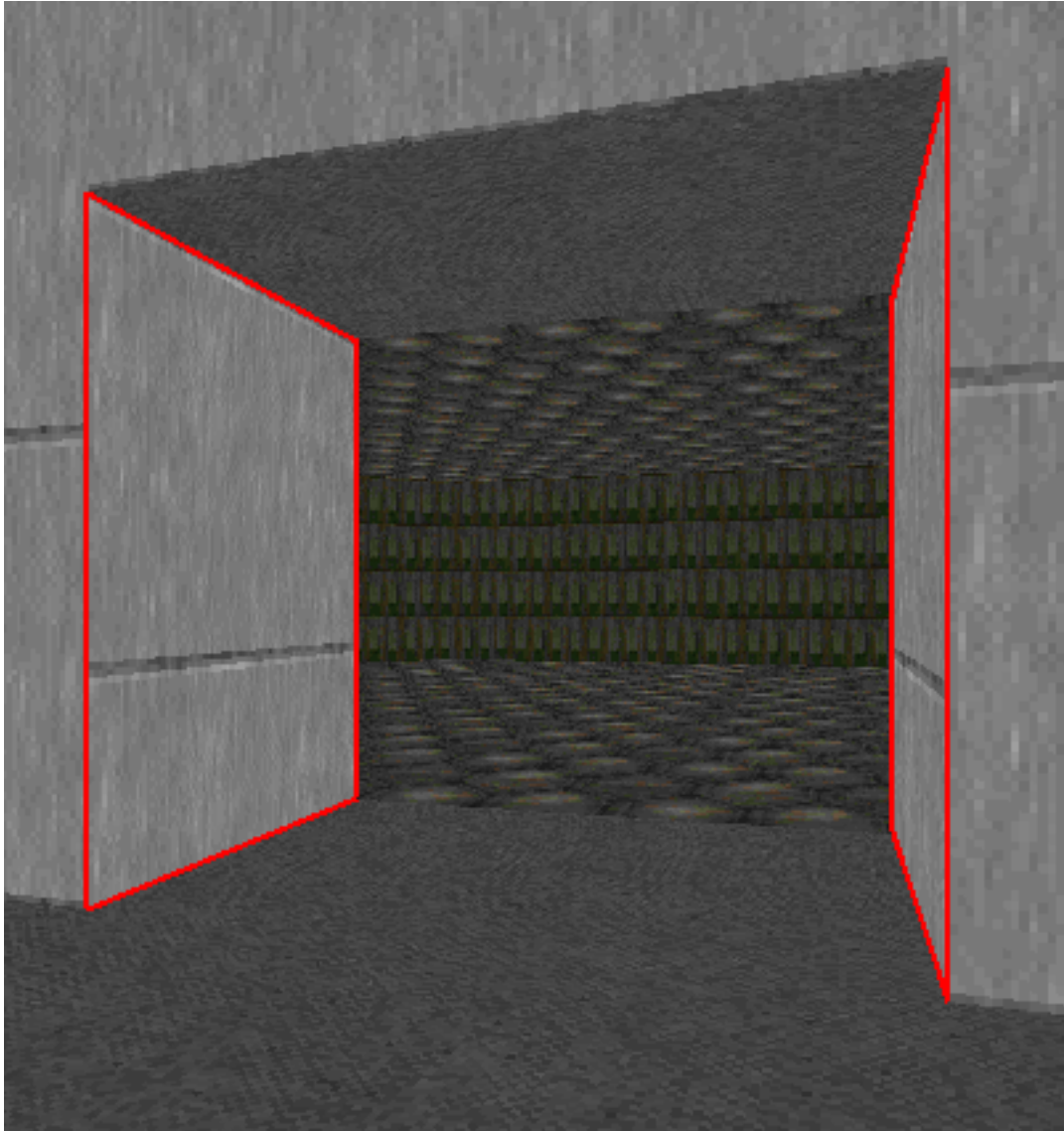
- Alternatively press `[]` to adjust ceiling height via keyboard shortcuts



## 6.4 Texture alignment

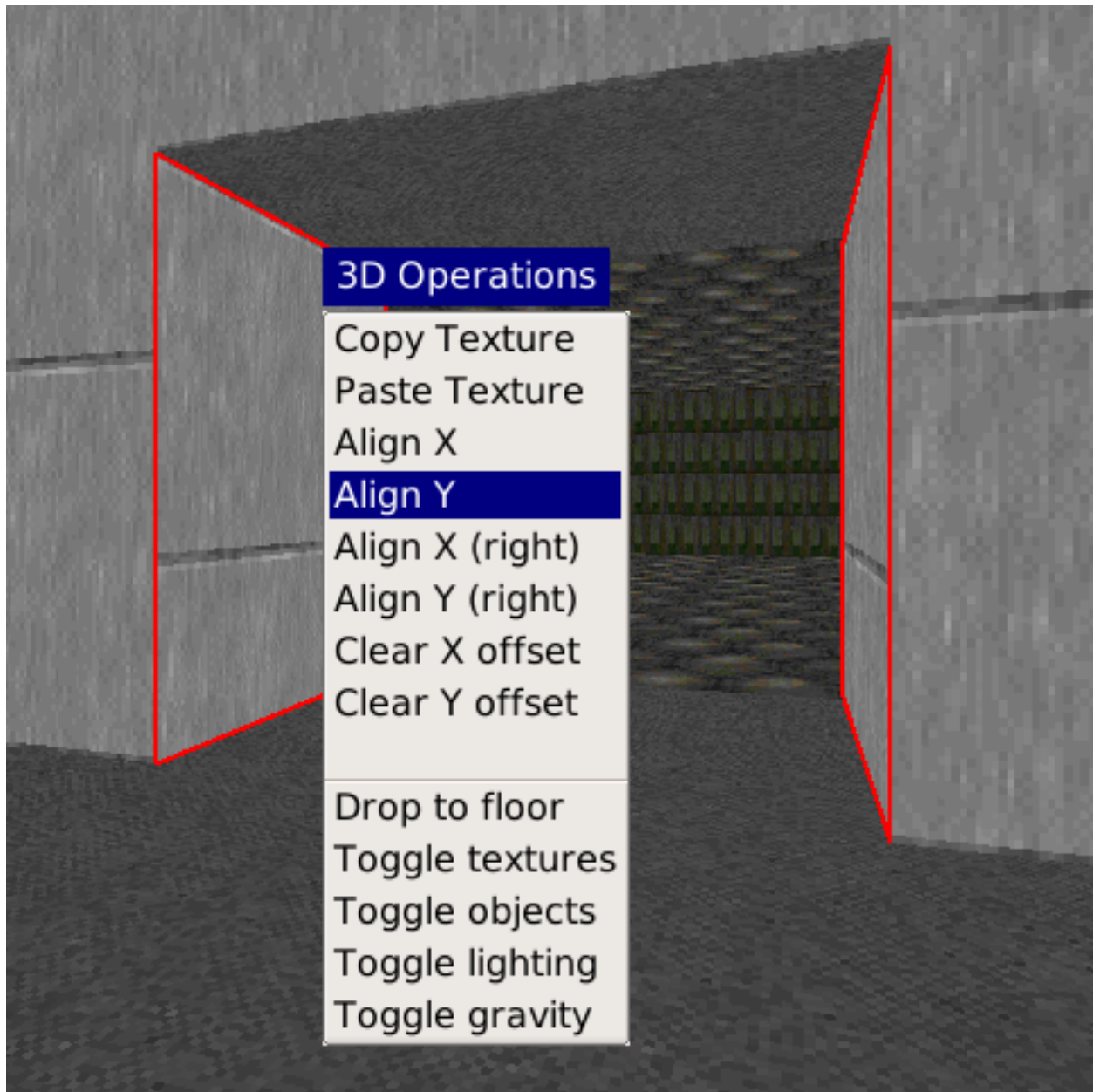
- Press `tab` to enter 3D view
- Click with the `LMB` to select walls with misaligned textures





- **Press these keyboard shortcuts to align:**
  - `x`: align X offset with wall to the left
  - `y`: align Y offset with wall to the left
  - `z`: align both X and Y offsets with wall to the left
  - `shift-x`: align X offset with wall to the right
  - `shift-y`: align Y offset with wall to the right
  - `shift-z`: align both X and Y offsets with wall to the right
- Alternatively press `F1` to open the operations menu and select one of the align options.





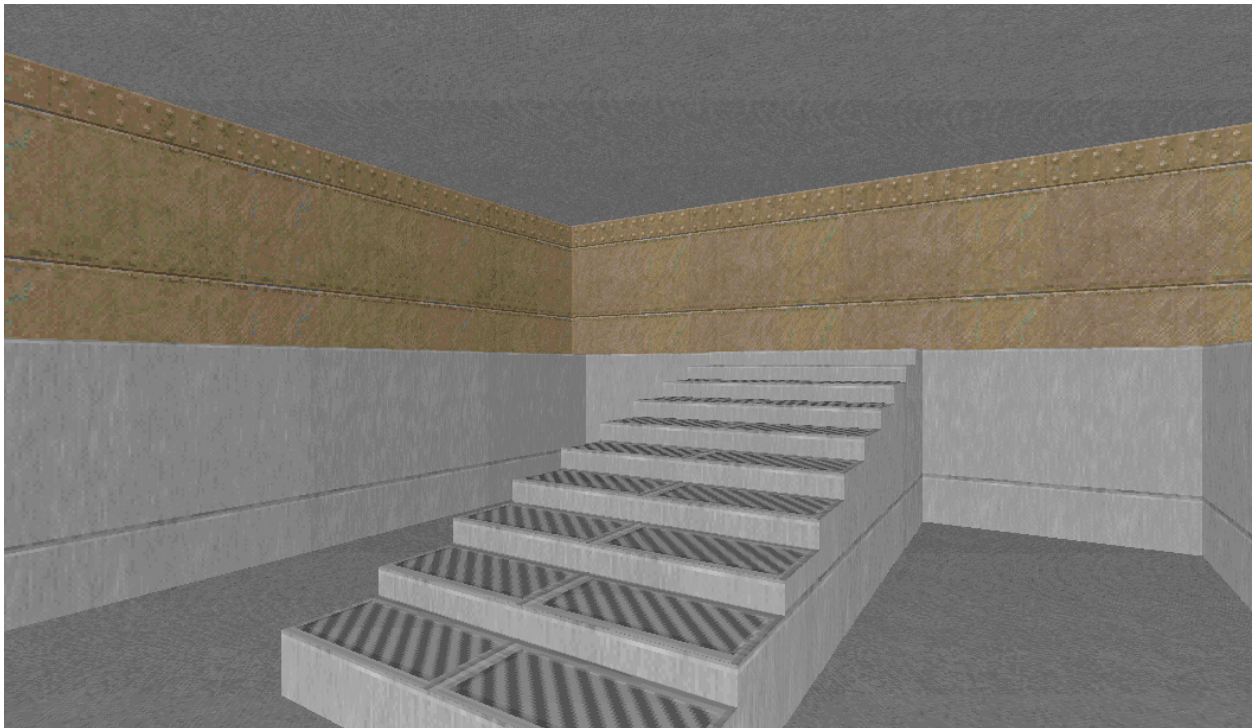
## 6.5 Auto alignment

If you need to align multiple surfaces, you can use the auto-align feature:

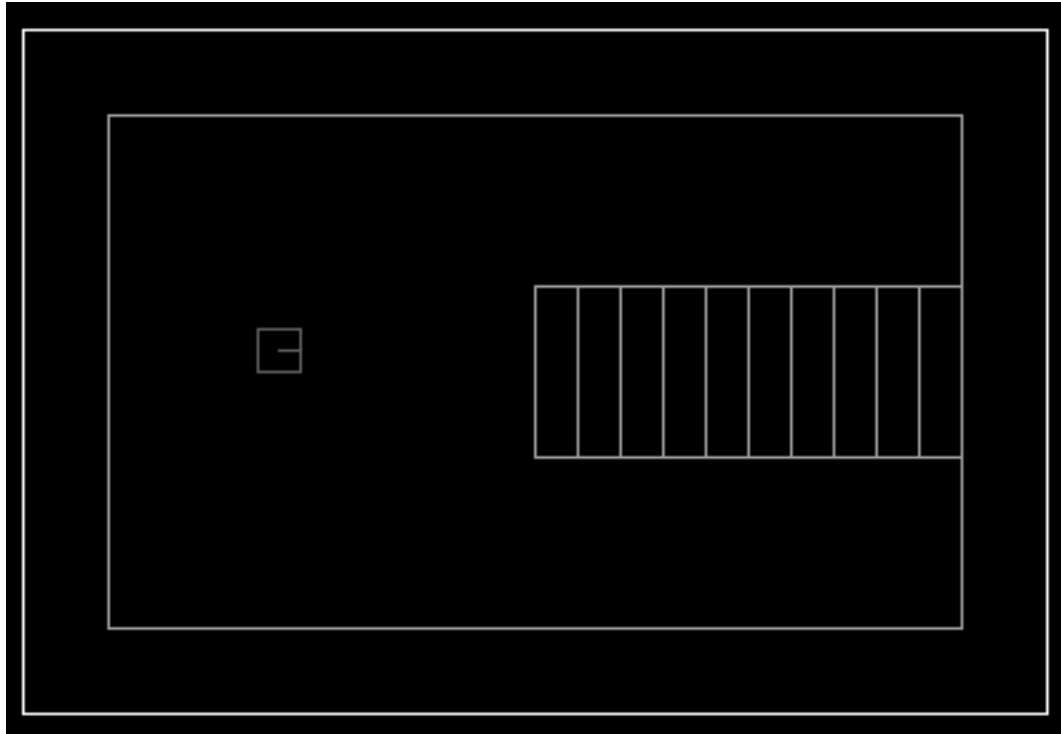
- In the 2D view, enter linedef edit mode `1`
- Highlight all lines to align
- Press `shift-A` to auto-align offsets on all selected linedefs
- This shortcut works in both 2D and 3D views



### 7.1 Stairs

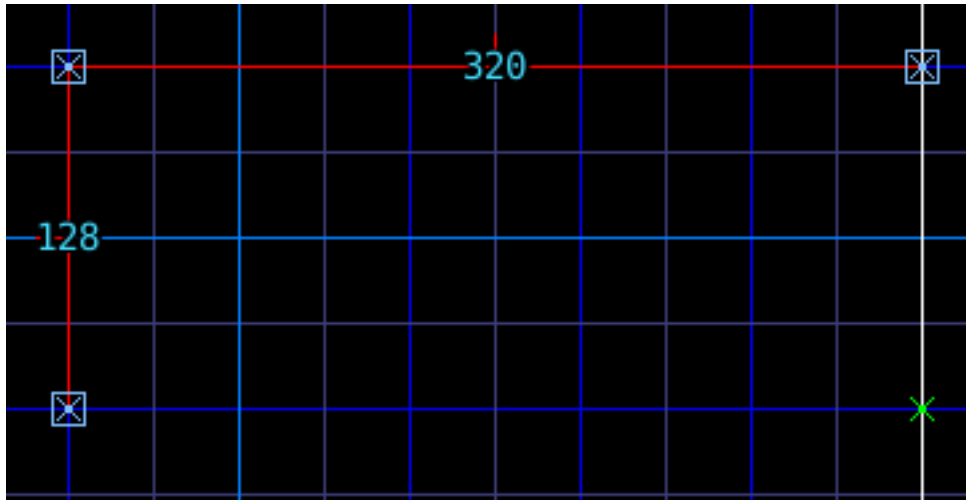


### 7.1.1 Layout

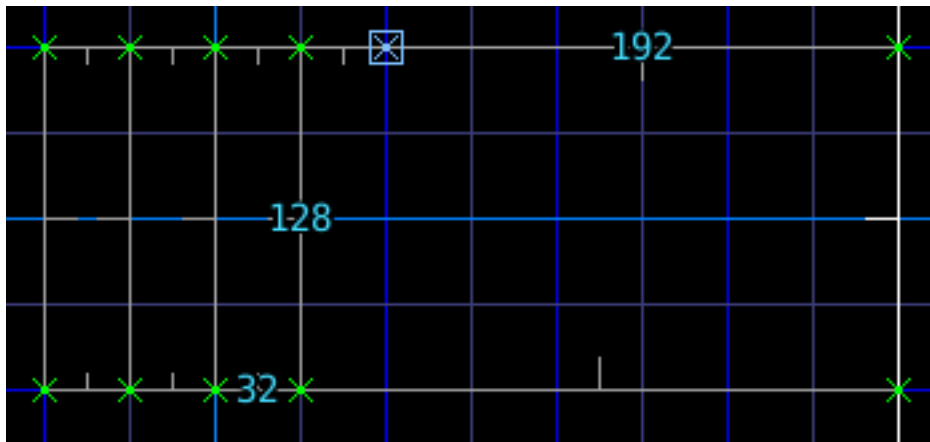


### 7.1.2 Method

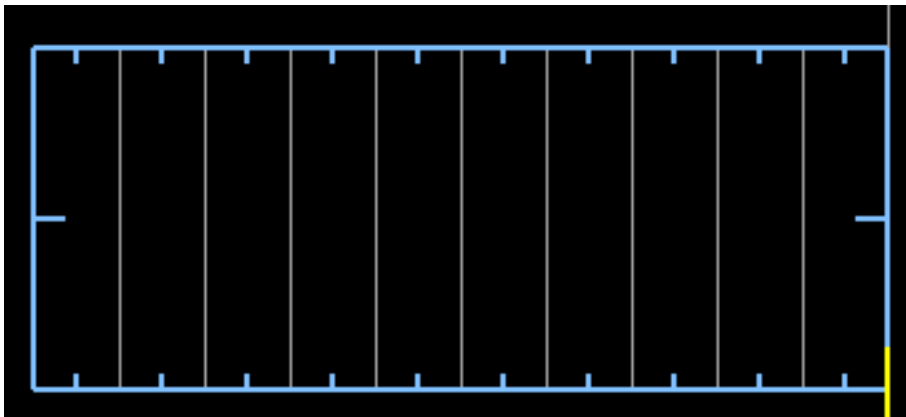
- In vertex edit mode, use the RMB to create an outline:



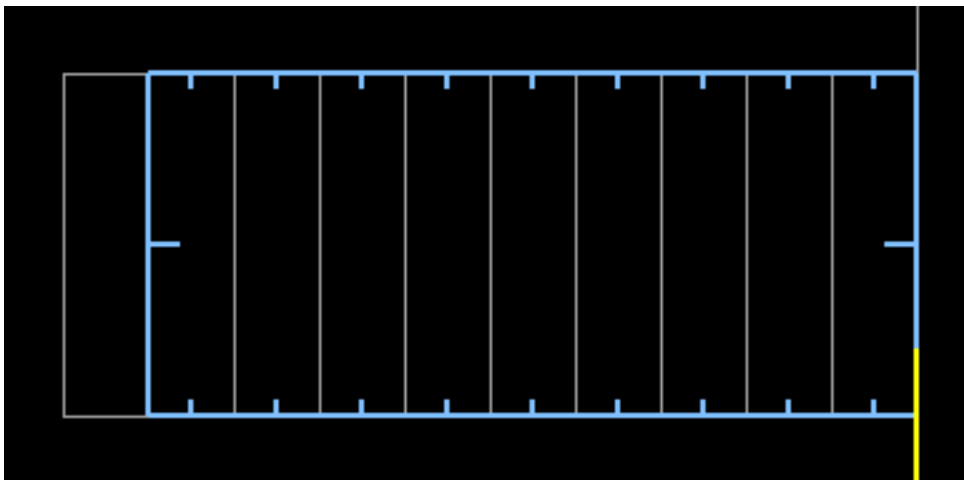
- Now add vertices for the steps in between:



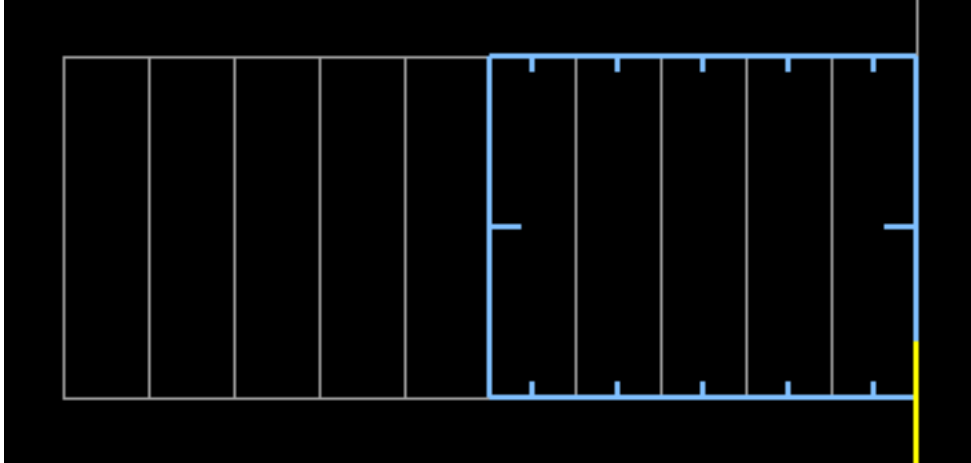
- In sector edit mode, select all the steps. Hold `shift` to make the selection without moving the room sector:



- Press the raise floor shortcut key `.` twice
- Using the LMB, deselect the first step:



- Repeat raising the floor and deselecting the next step:



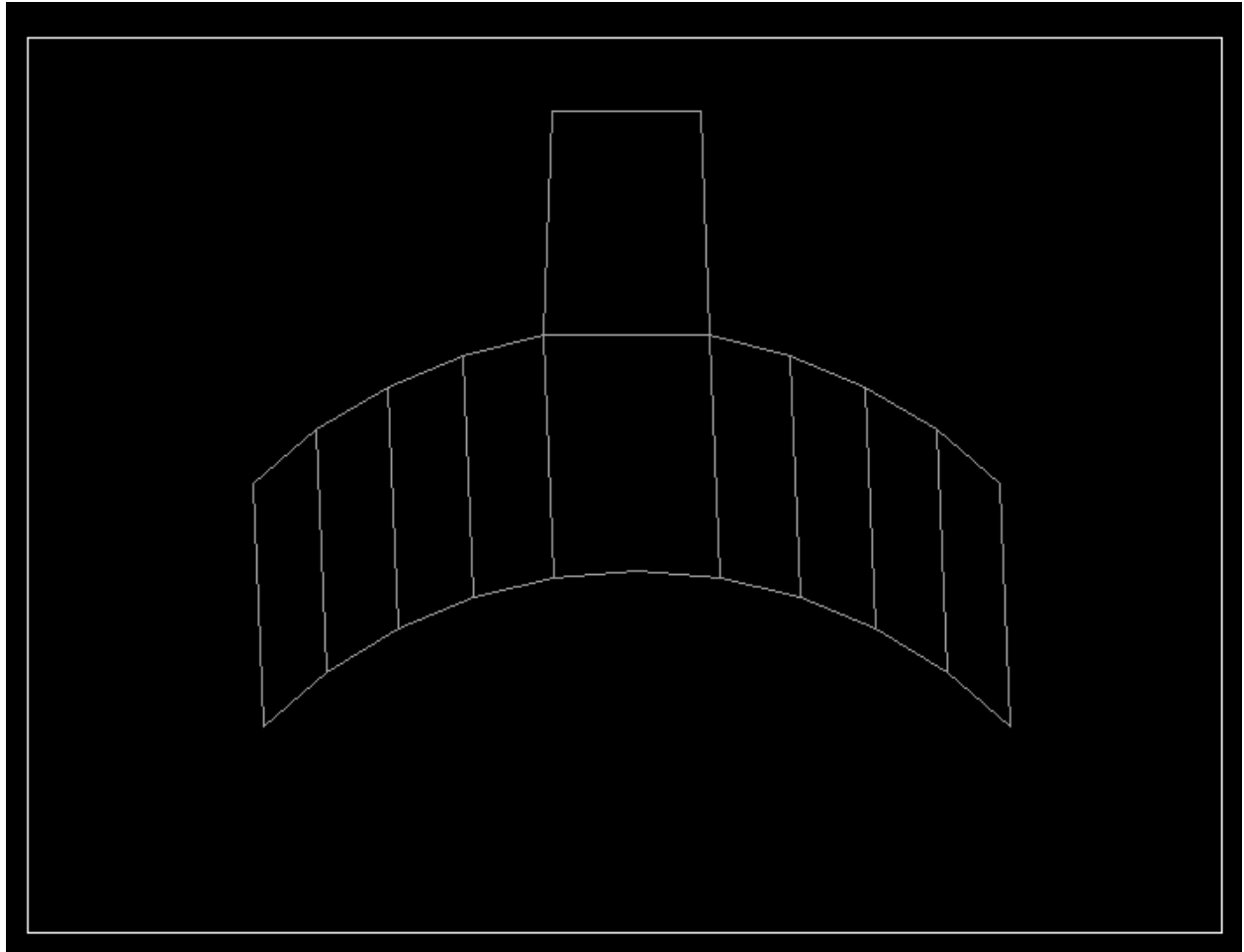
### 7.1.3 Downloads

`stairs.wad`

## 7.2 Curved Stairs

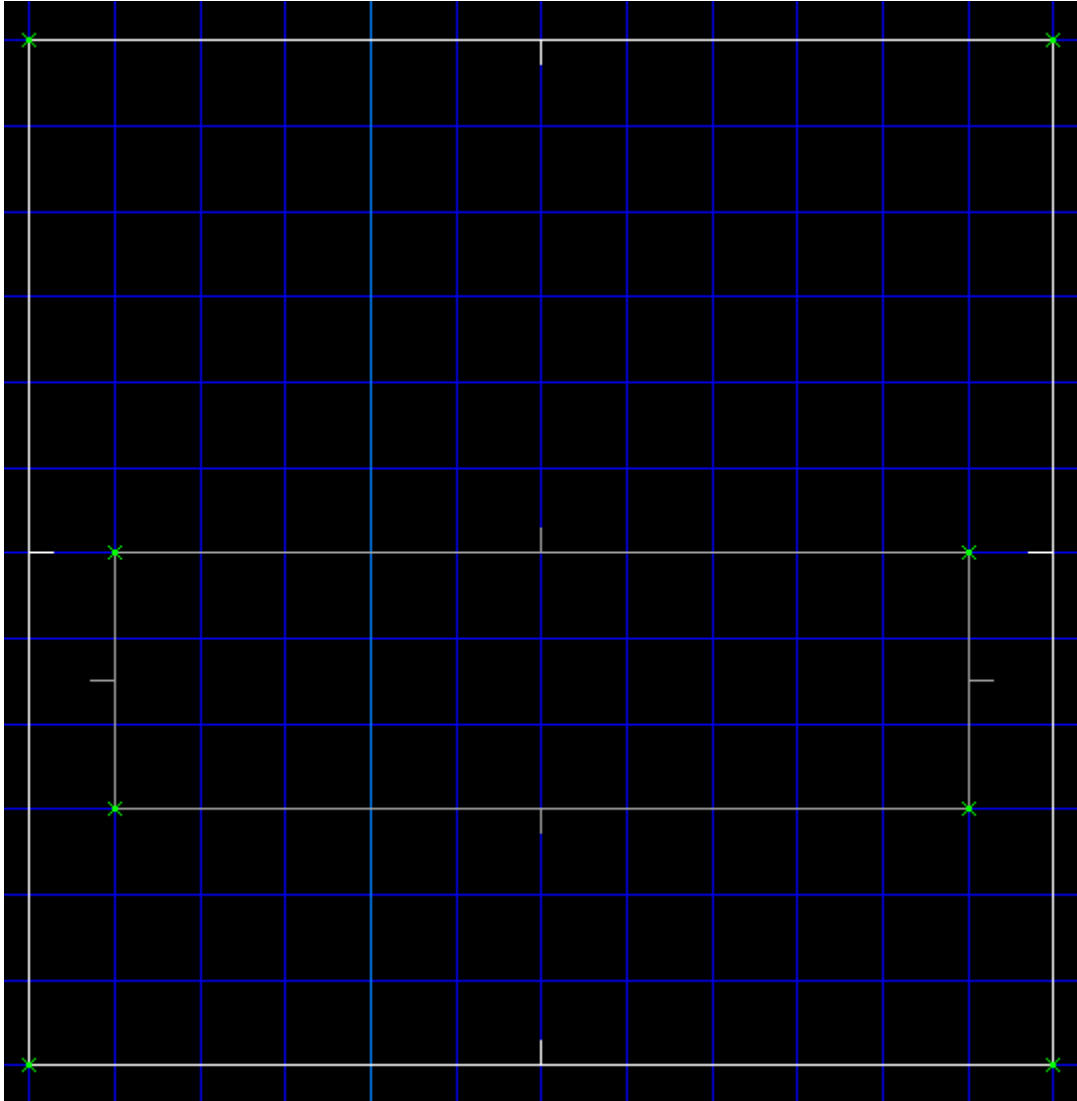


### 7.2.1 Layout



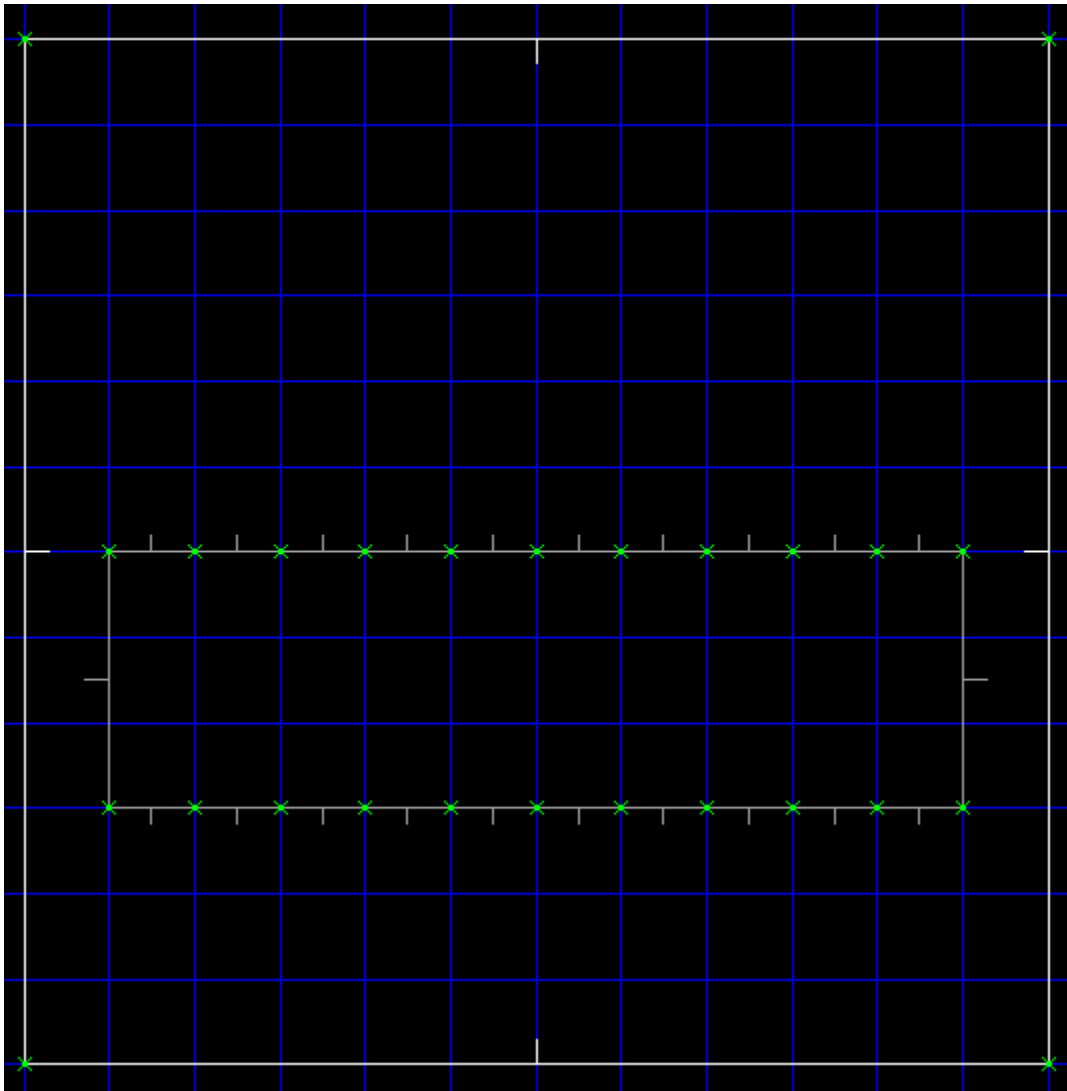
### 7.2.2 Method

Draw the outline of the stairs in vertex edit mode:

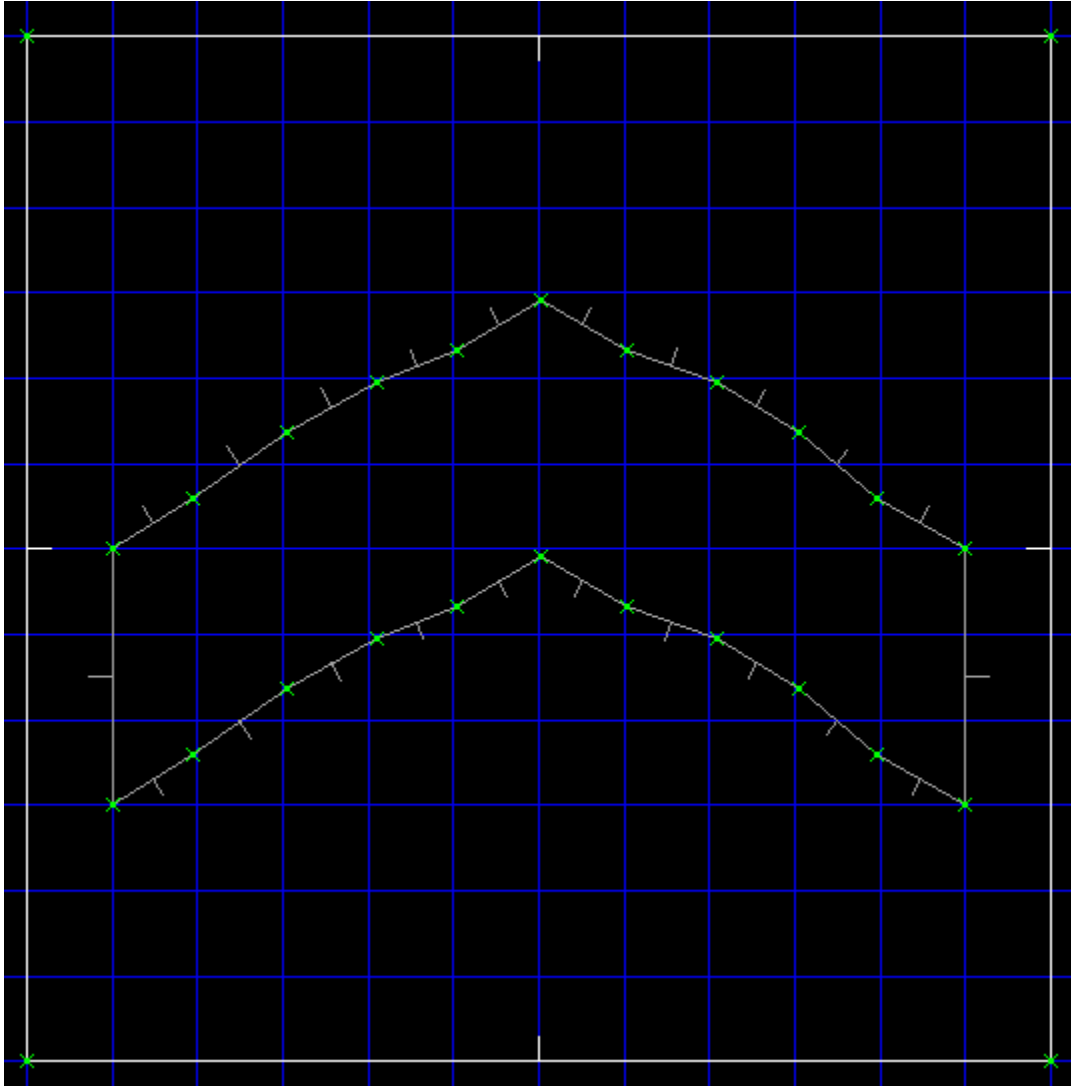


Use the **LMB** to insert vertices along the edges, to define where the steps will be:

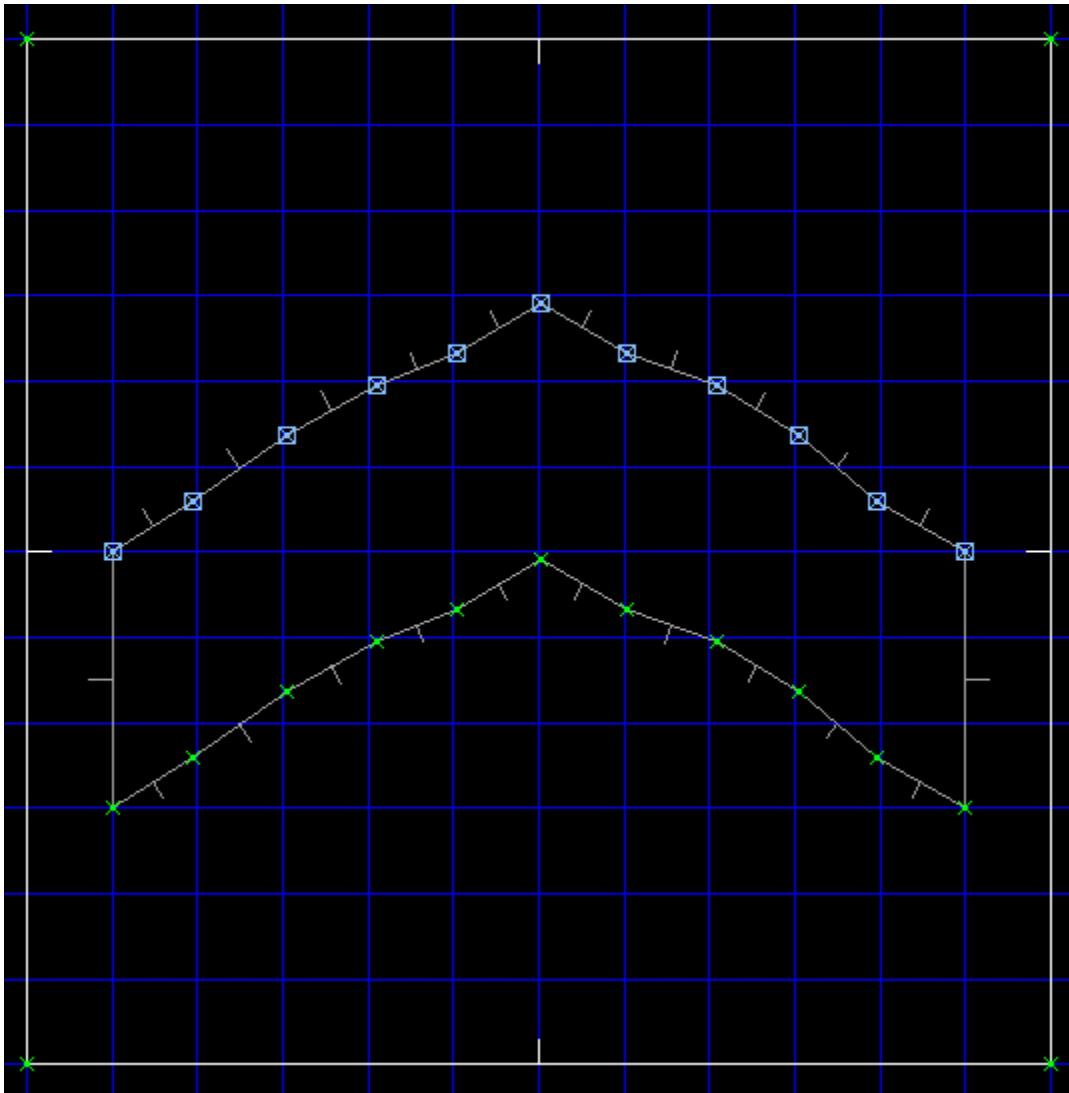




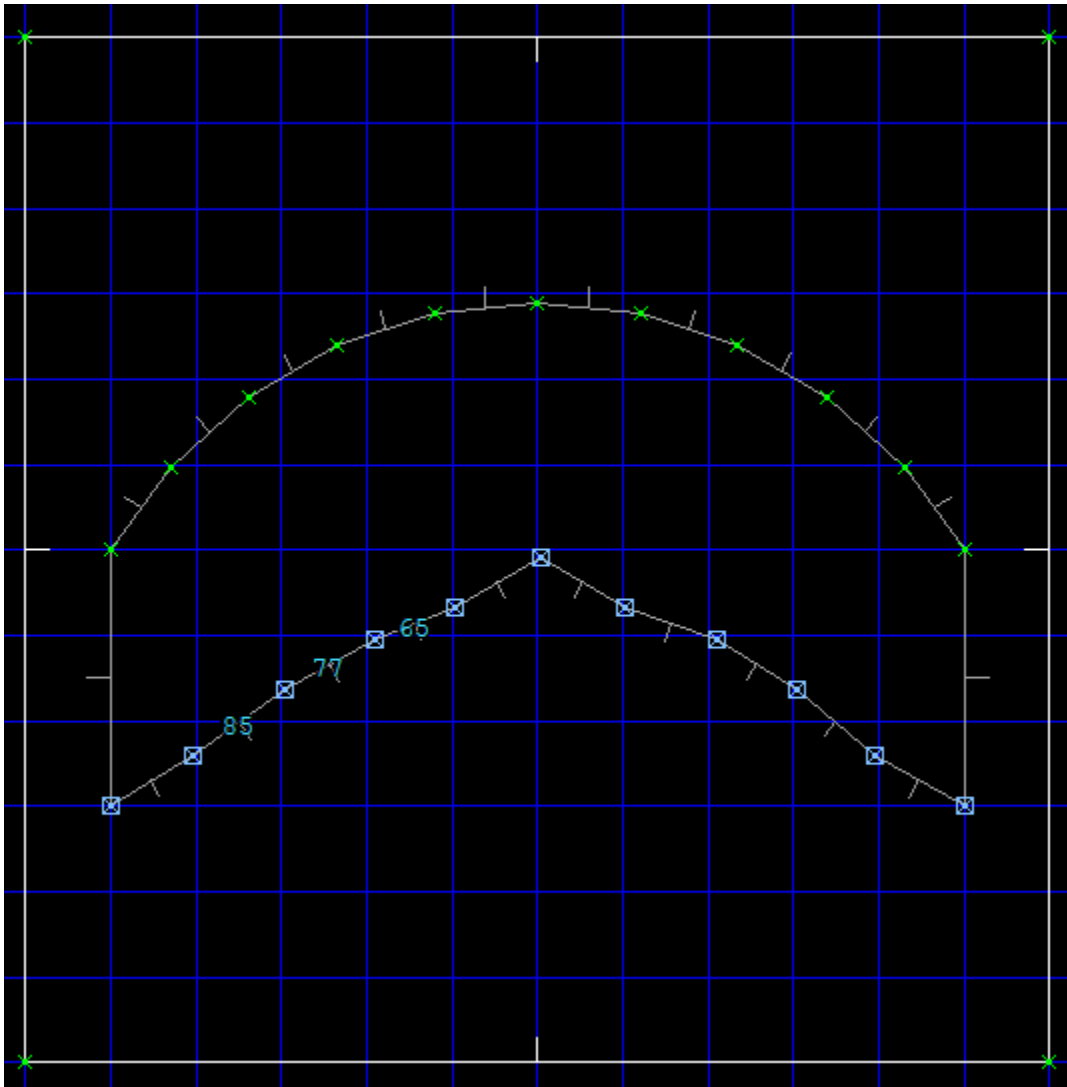
Press  $\text{f}$  to toggle free mode (no grid snapping) and drag the step vertices *roughly* into the arc shape.



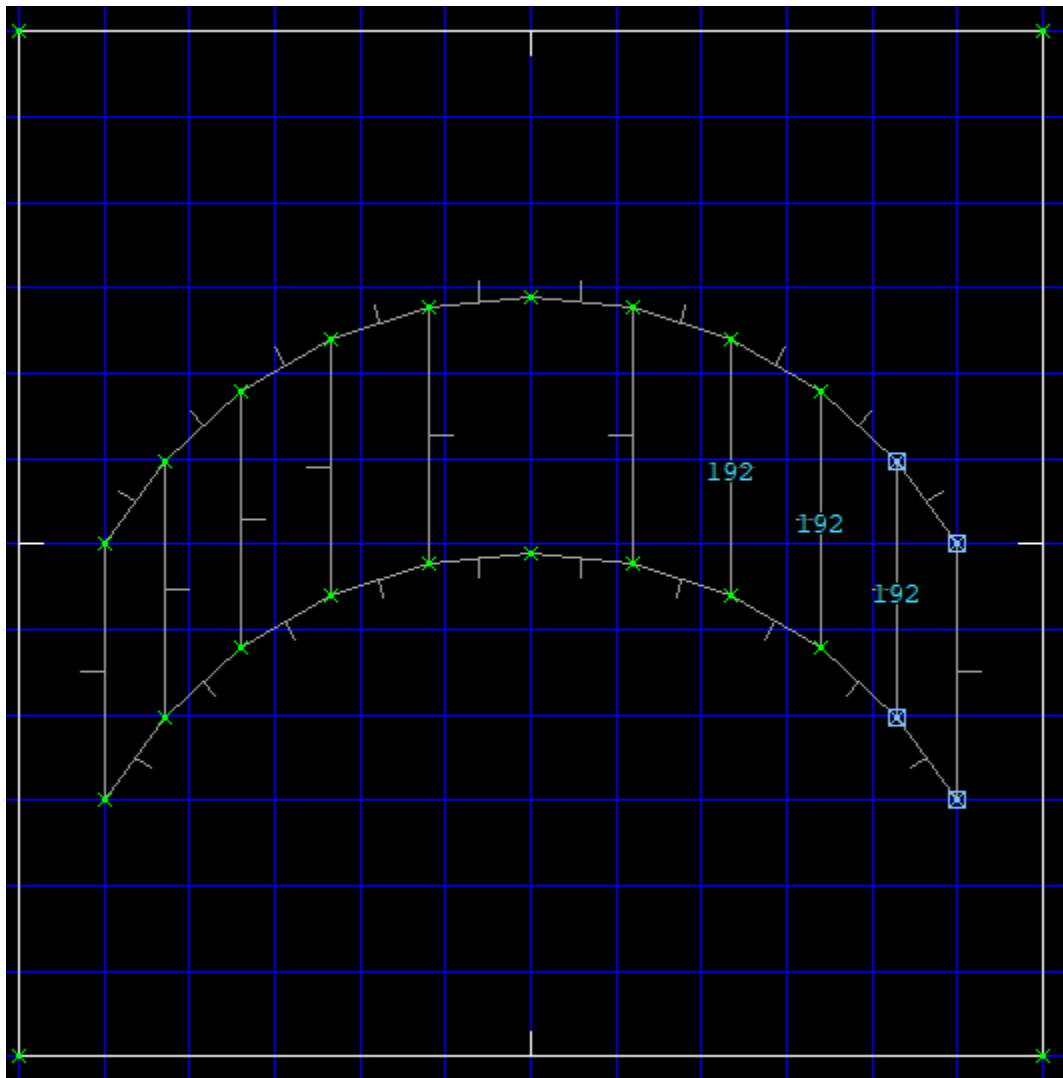
Deselect all vertices ( ` ) and only select those on one side of the stairs:



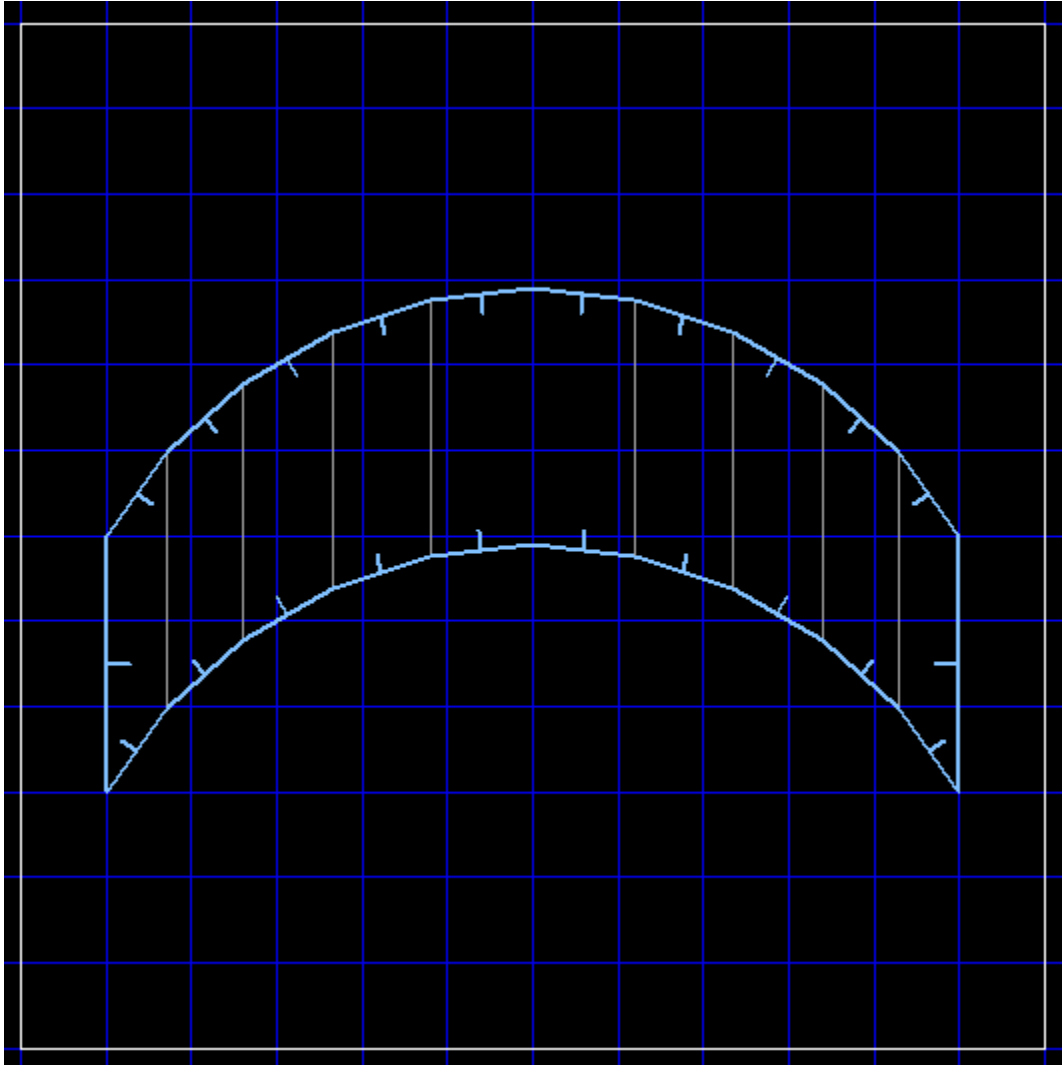
Press `shift-c` (shape arc to 120 degrees):



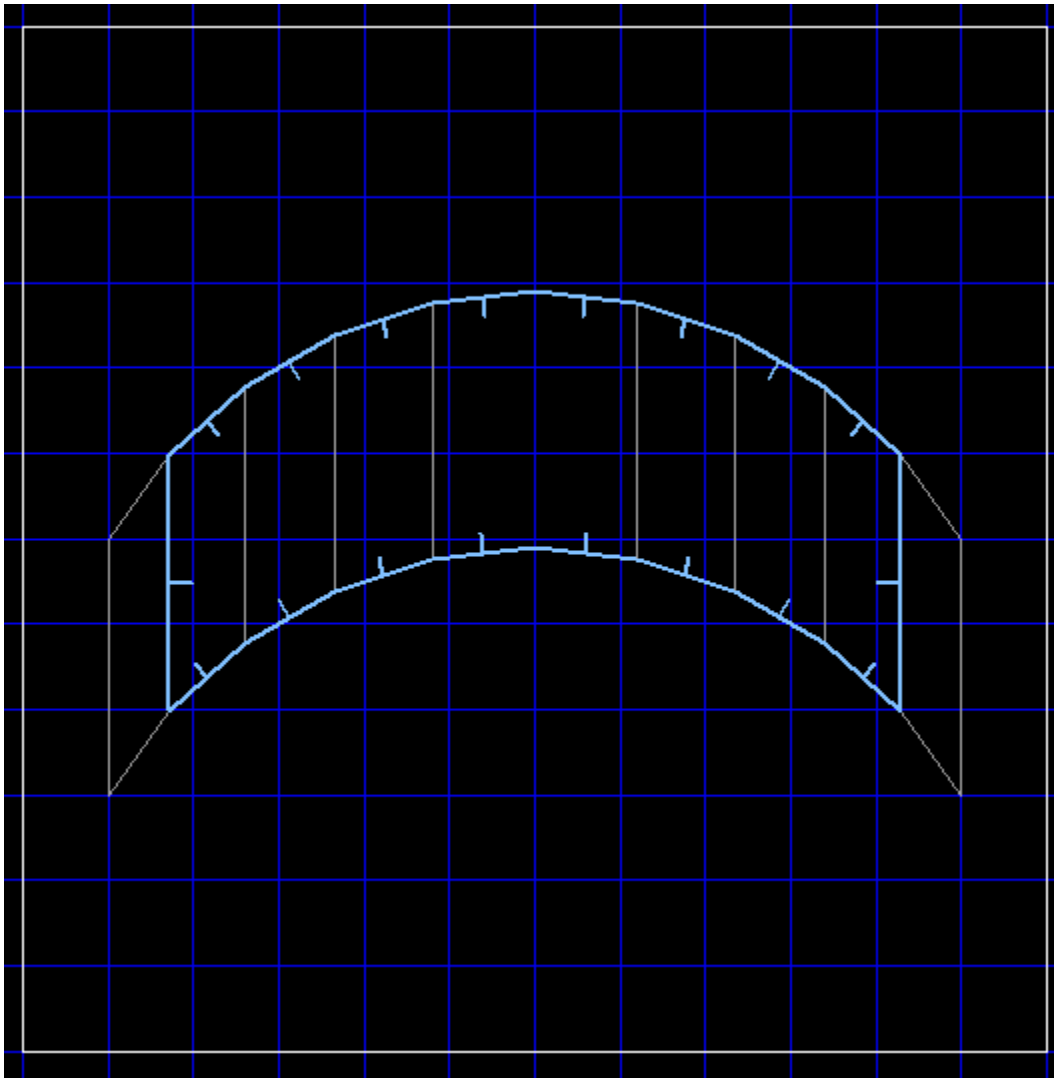
Now deselect all vertices, select only the other side and apply the shape arc operation again. Next use the RMB to join the step vertices:



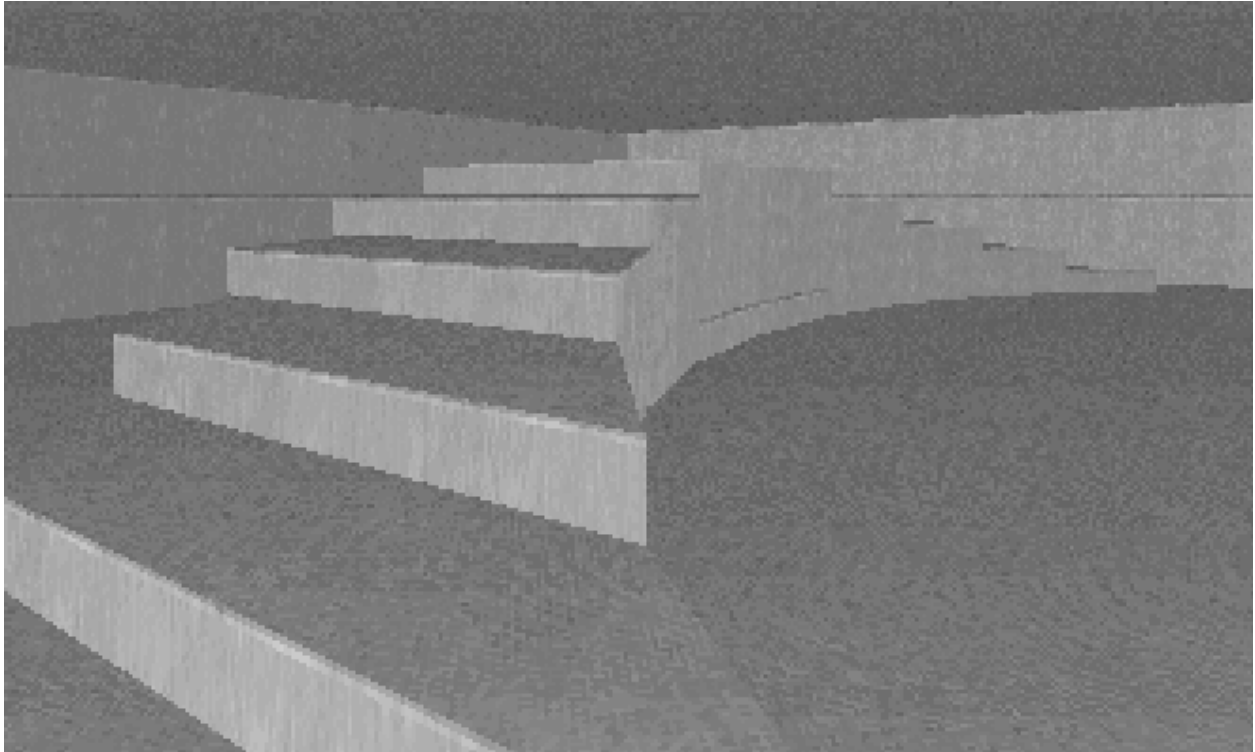
Switch to sector edit mode, hold `shift` and drag-select the stair sectors:



Press the raise floor shortcut (.) twice. Using the LMB, deselect the left-most step, and deselect the right-most step:

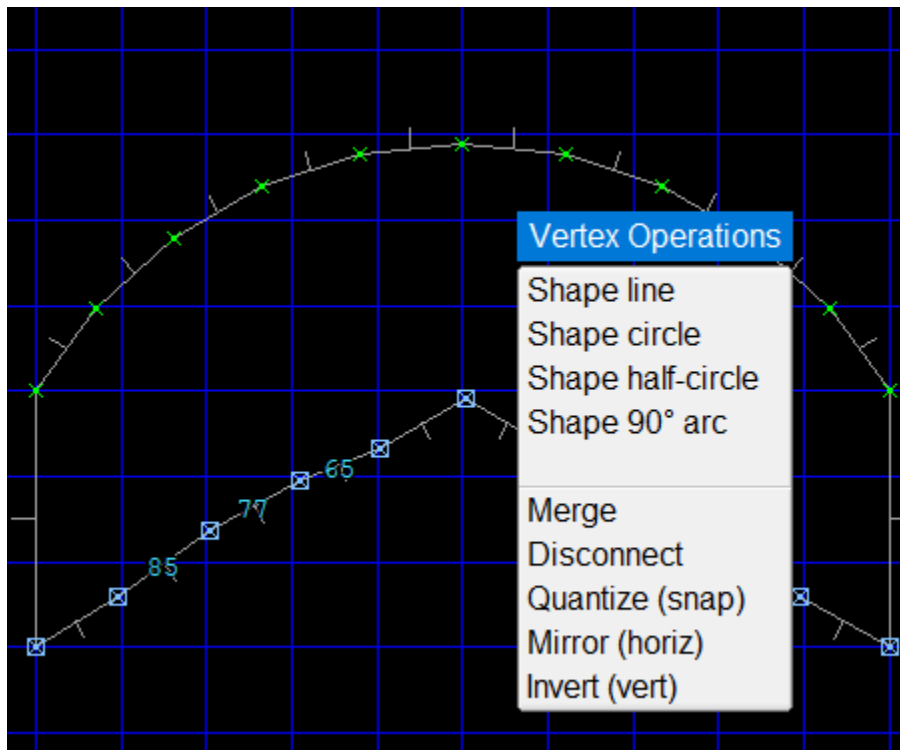


- Repeat raising the floor and deselecting a step from either side, until you reach the center step. Your stairs are now done:



### 7.2.3 Notes

The shape arc operation is also available by pressing `f1` to bring up the operations menu. However this menu does not have the 120 degree arc, so we used the `shift-c` keyboard shortcut.





## 7.2.4 Downloads

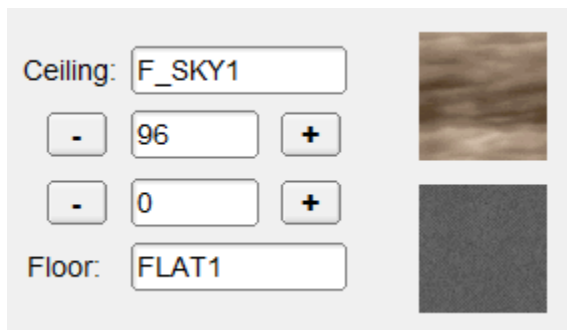
`curved-stairs.wad`

## 7.3 Sky

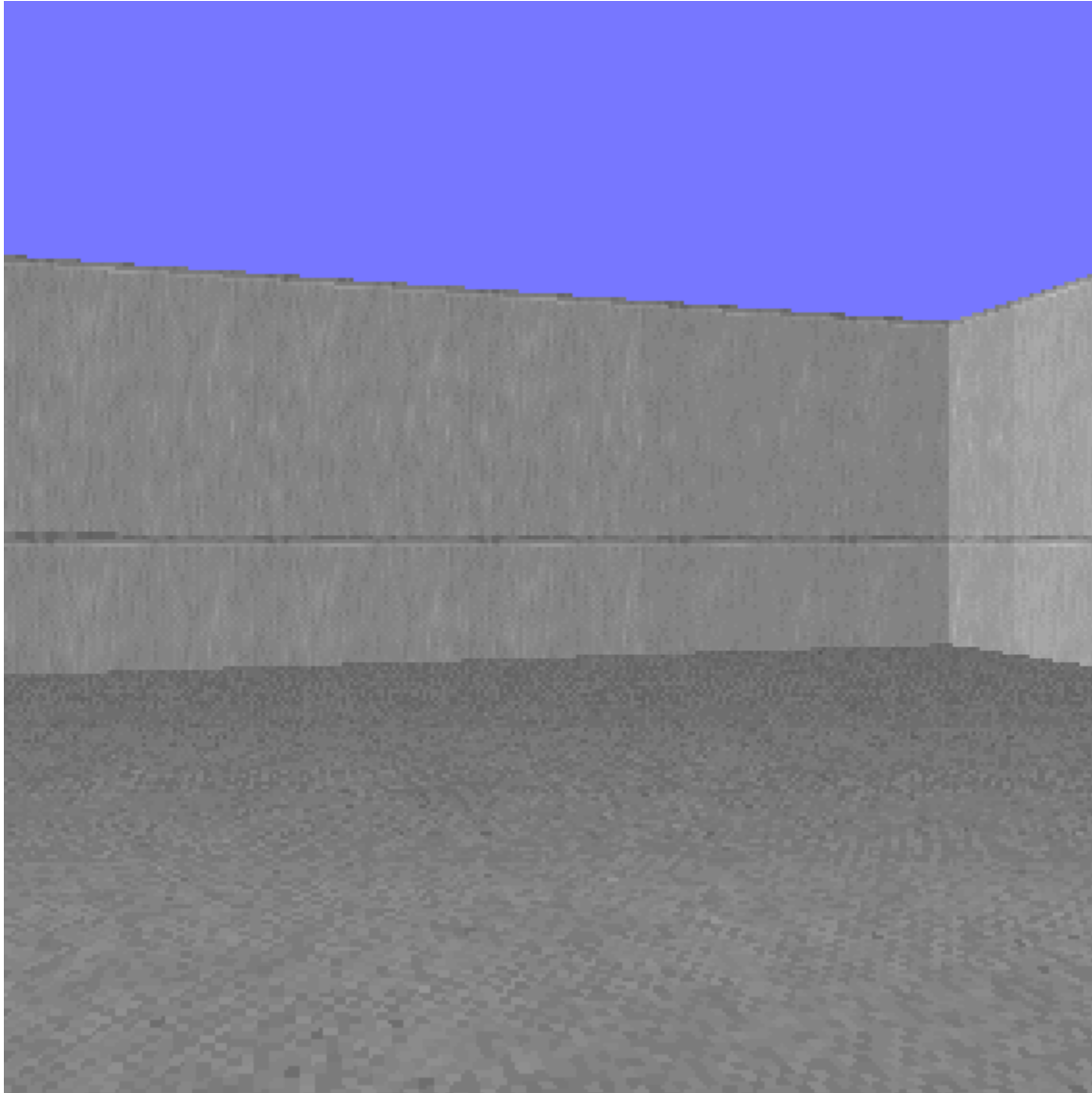


### 7.3.1 Method

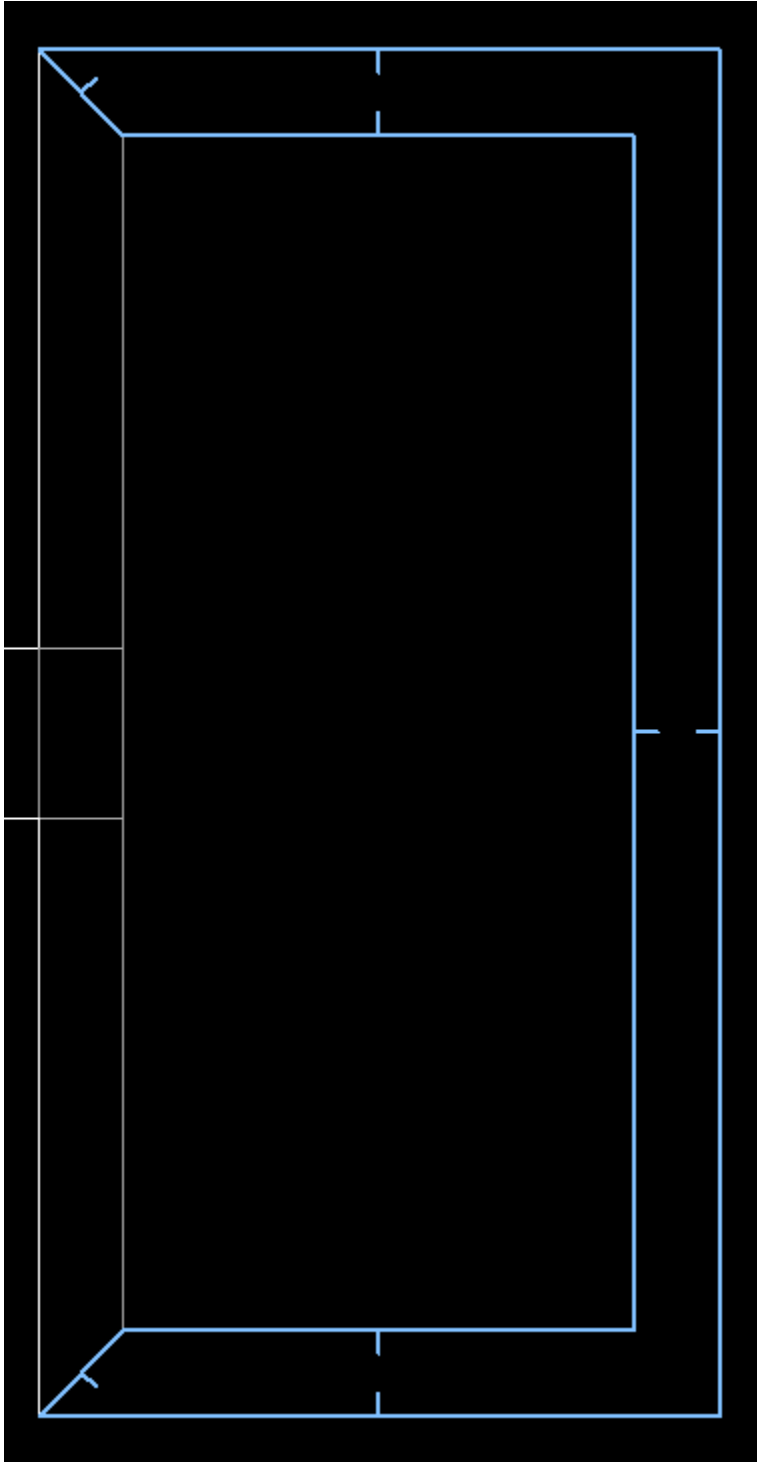
Setting the sky is a simple matter of picking the right ceiling texture, *F\_SKY1*.



However the walls can get in the way of that nice view:



To lower the walls, add a border sector near the edge (highlighted below):



Lower the ceiling of this border sector ( `[]` ) so that you can see more of the sky.

### 7.3.2 Downloads

`sky.wad`

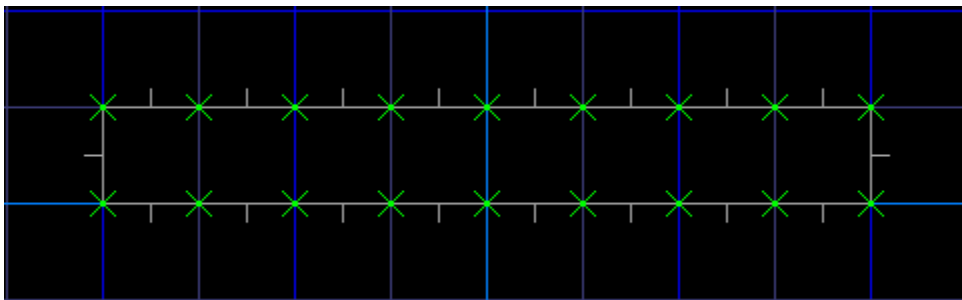
## 7.4 Sacrificial Altar



Using concentric circles we construct a sumptuous sacrificial altar.

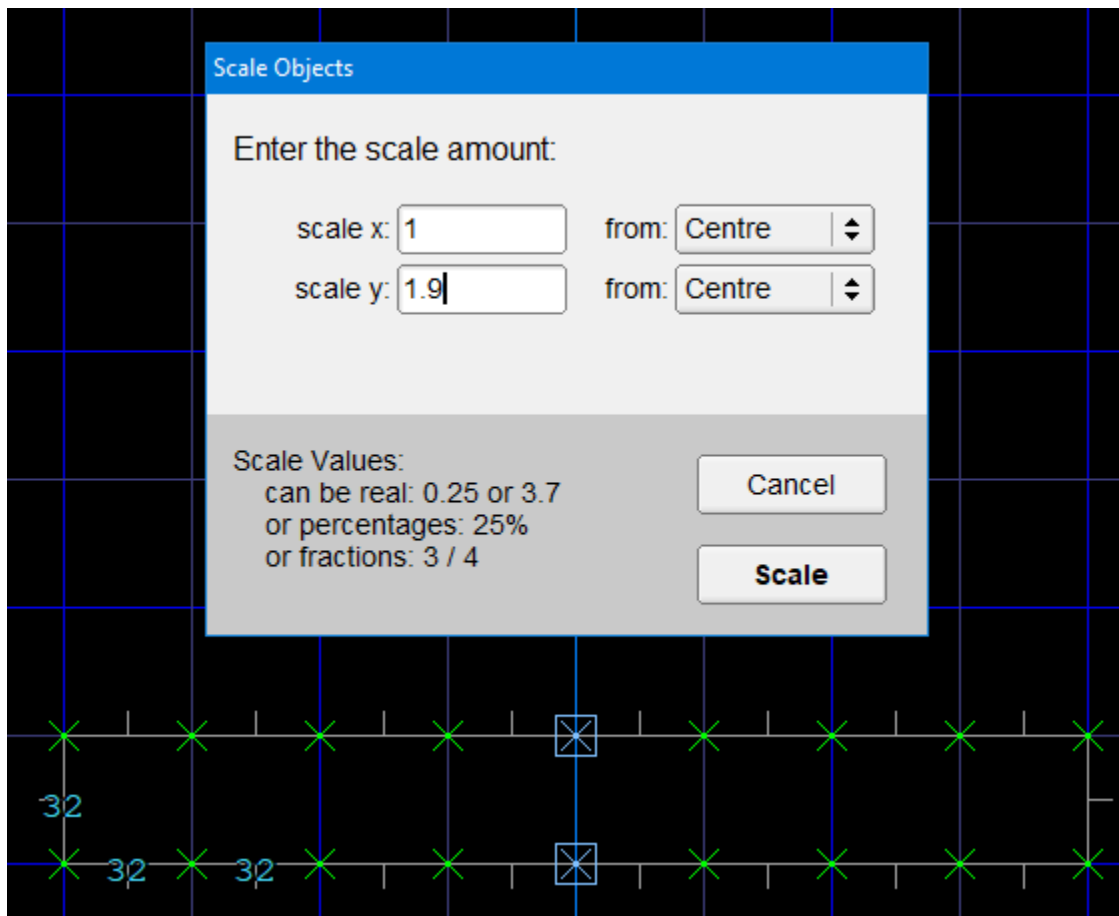
### 7.4.1 Method

In vertex edit mode, create a rectangle of points:

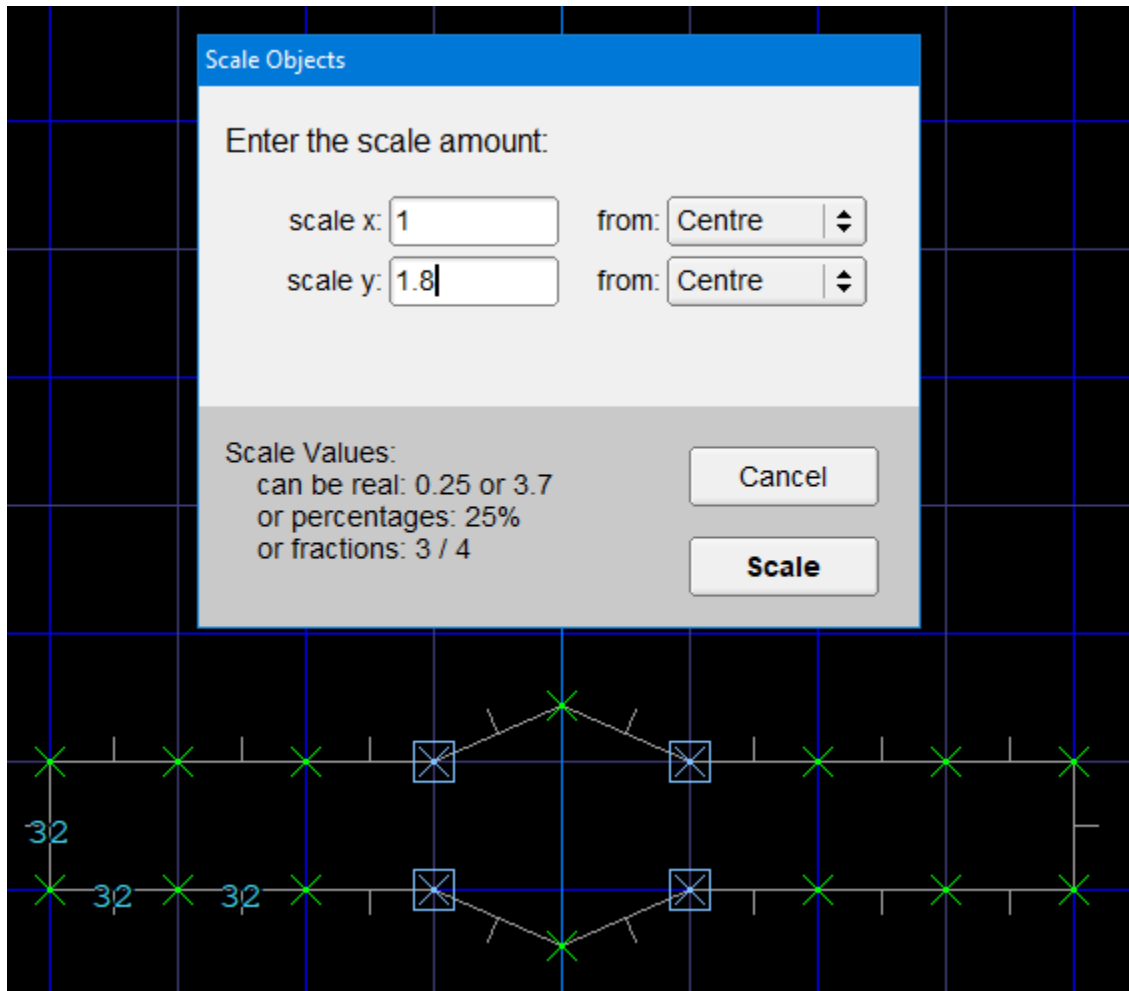


This rectangle will be transformed into a circle using the arc shaping tools.

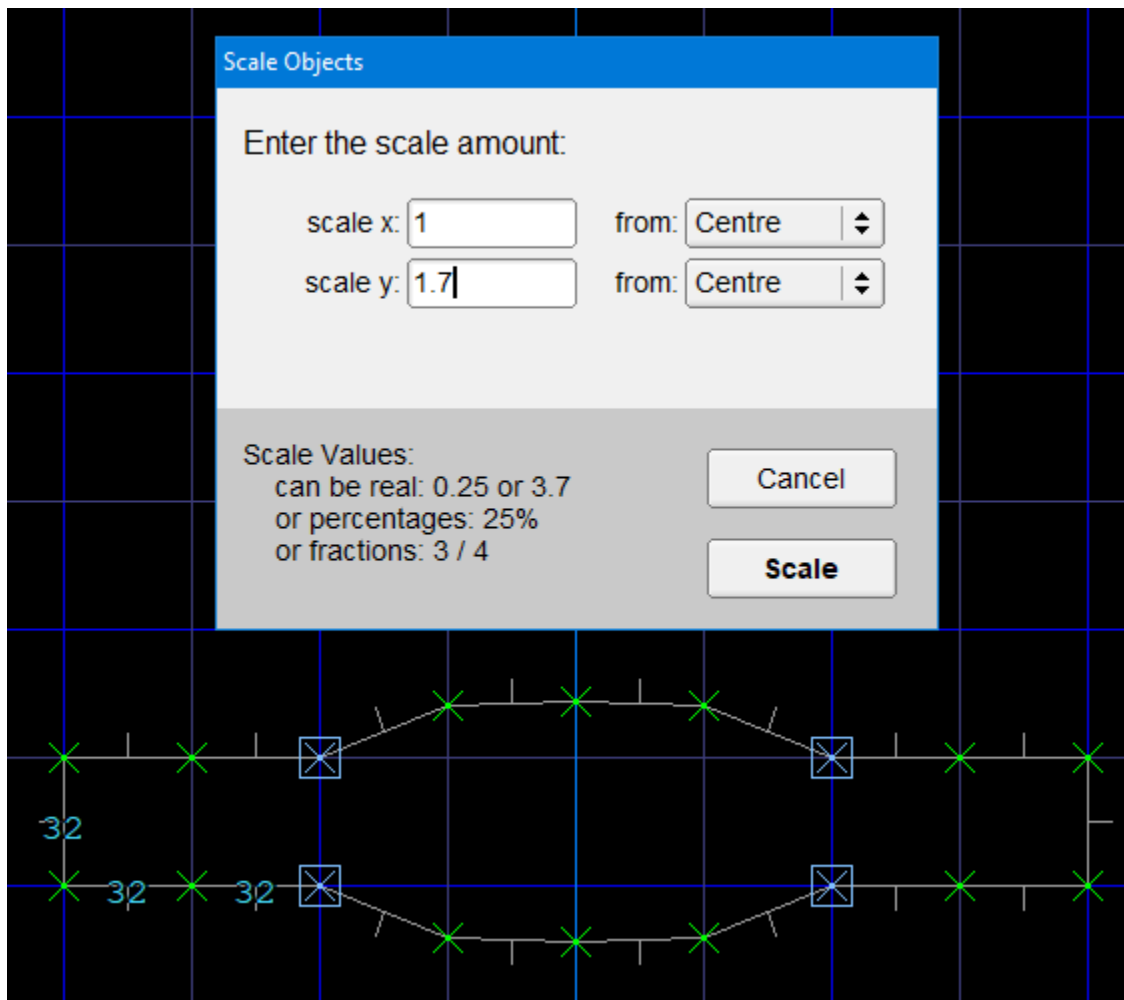
Select the centre pair of points and scale by *1.9* on the *y* plane:



Select the next pair of points on either side and scale by  $1.8$  on the  $y$  plane:



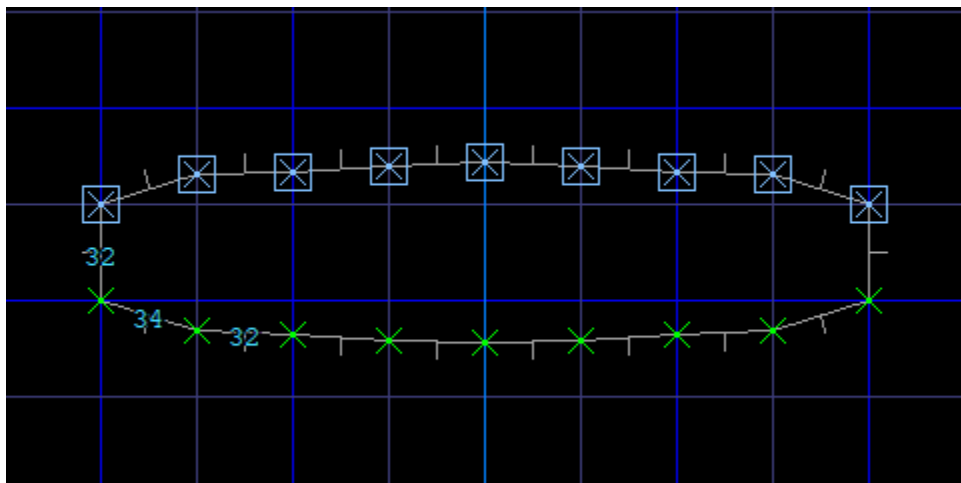
Select the next pair of points, scale by 1.7.



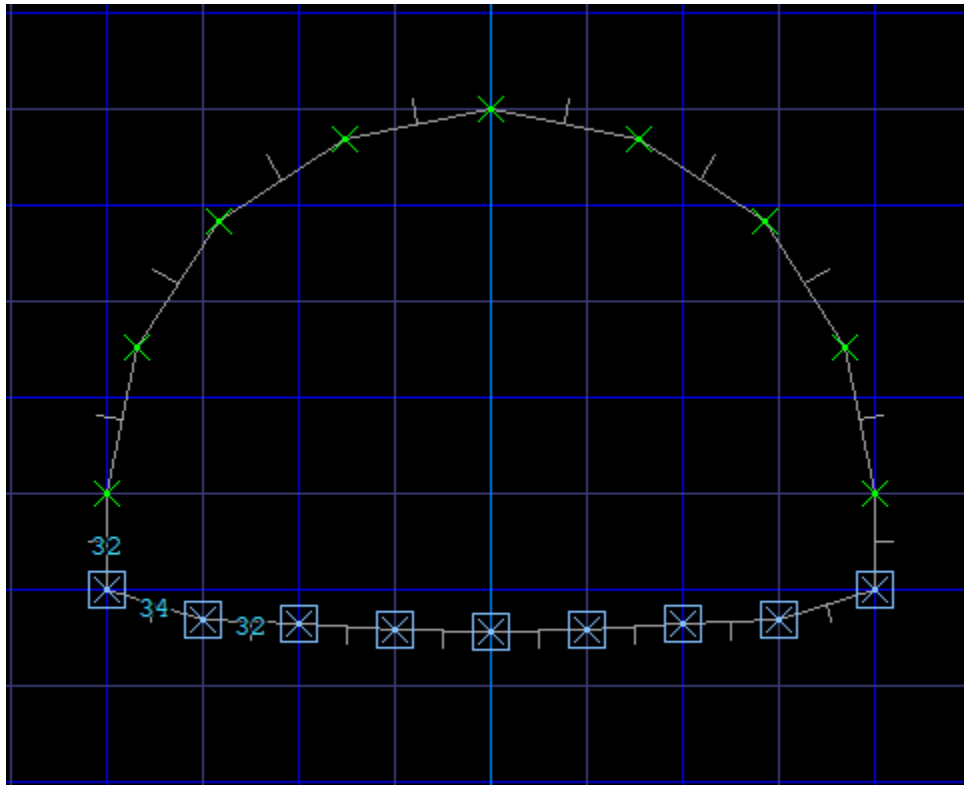
You can stop when you reach the edge vertices.

For each pair of points we scale by a fraction lower, this is to give the arc shape tool a hint as to the direction of the arc. If we tried to shape the points without this hinting we would instead see the “strange shape” status message.

Select all the top vertices and apply the arc shape 180 tool with `shift-d`:

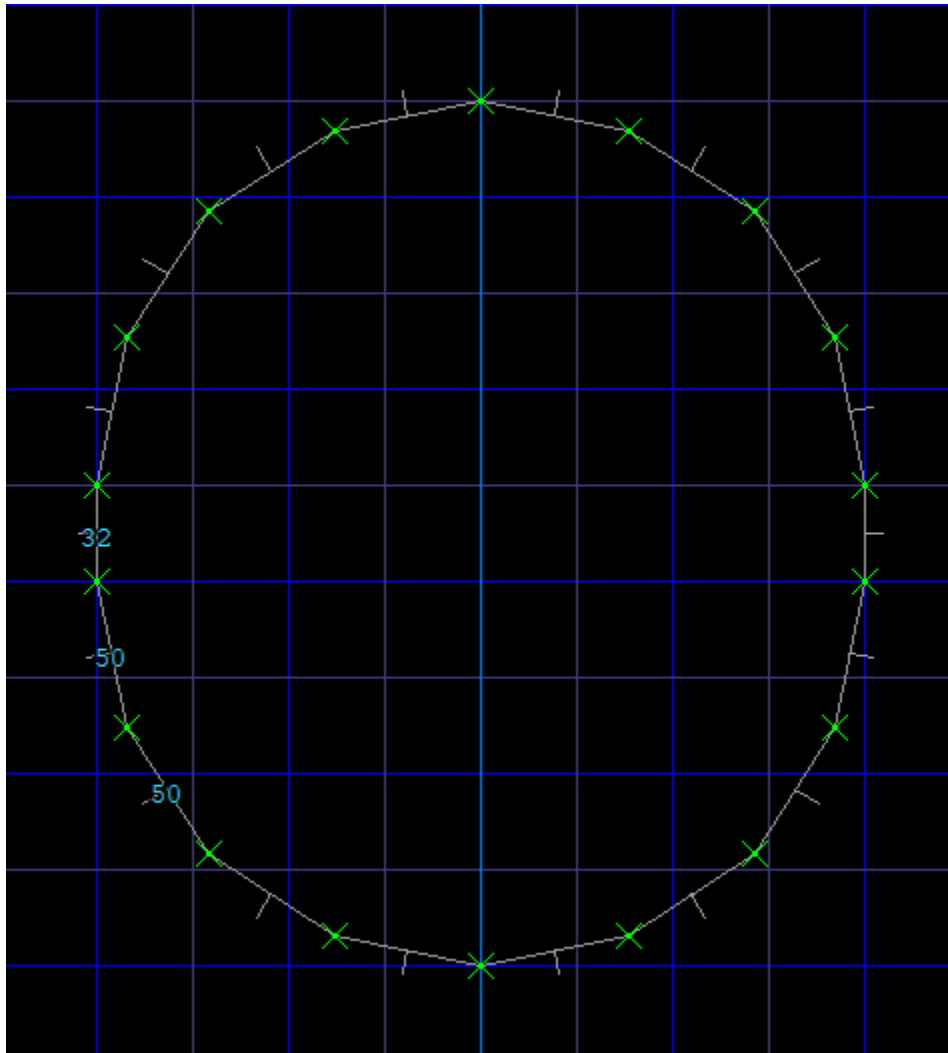


Select the bottom vertices and `shift-d` again:



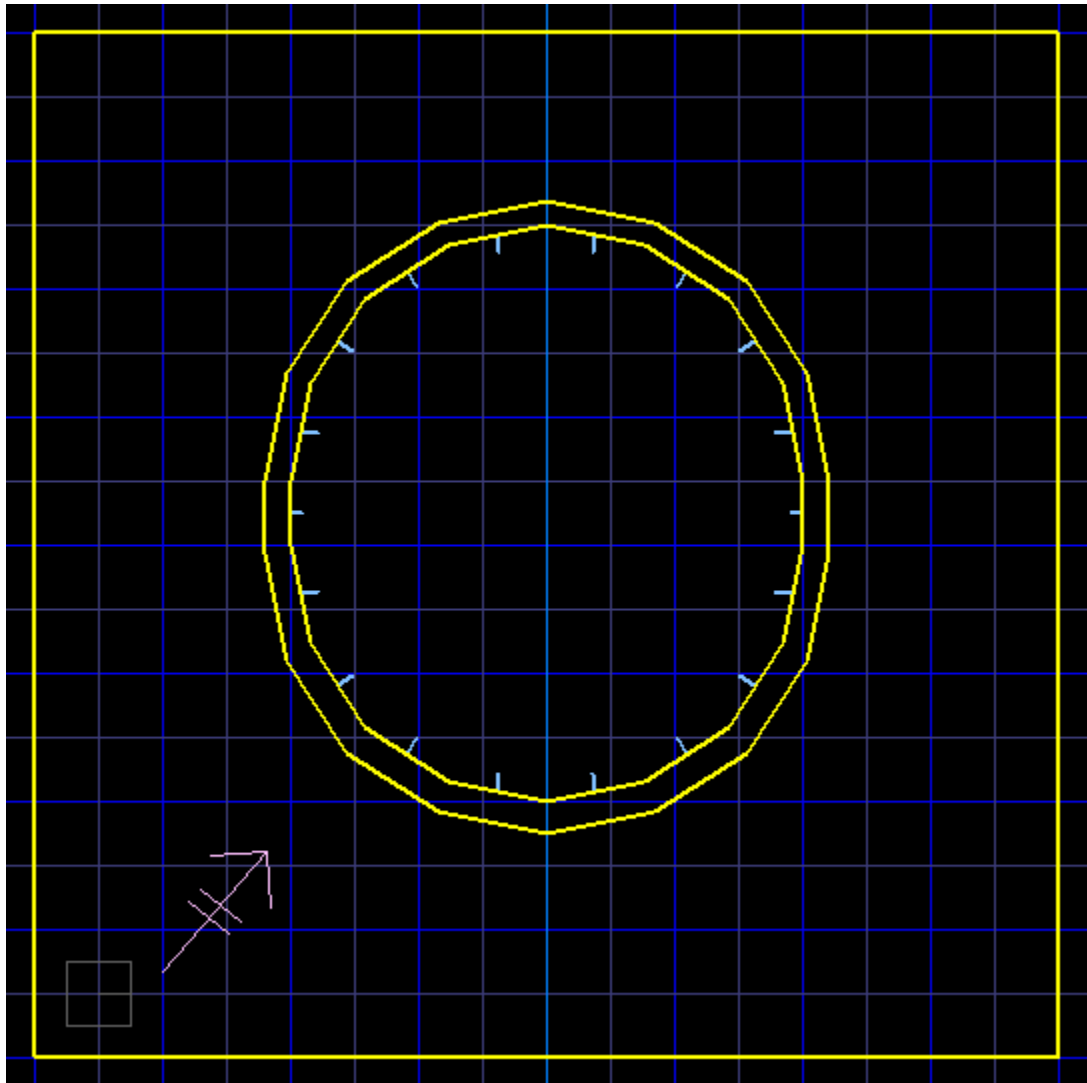
We now have the base for our altar:



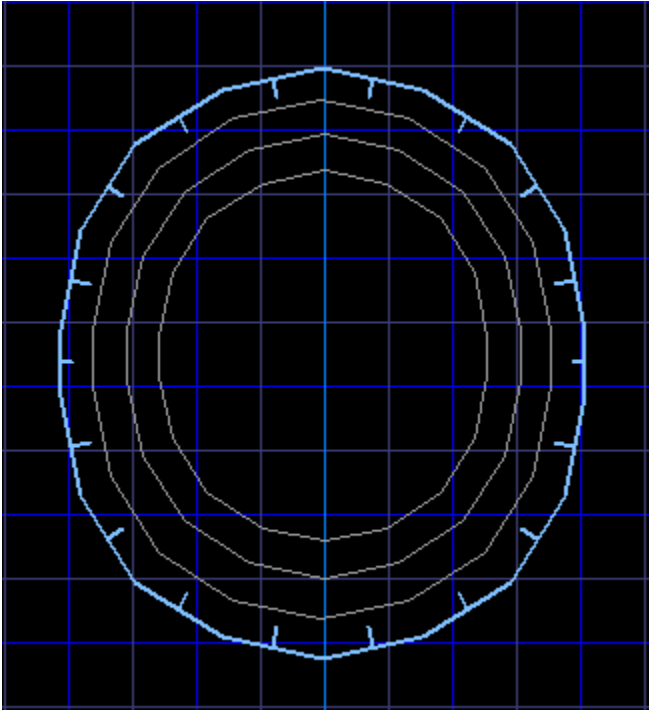


Switch to sector edit mode and select the circle sector. `control-c` and `control-v` to make a copy. Scale the copy to 1.2 on both *x* and *y* planes.

Switching to 3D view at this point will show the copied sector rendering with artefacts, this is because the copied sector needs merging into the room. Hover with the cursor inside the copied sector, the surrounding room highlights up to indicate this, and press `spacebar` to solidify the sector:



Repeat the above step two more times, scaling by  $1.4$  and  $1.6$  on both  $xy$  planes. You will end up with four concentric circles:

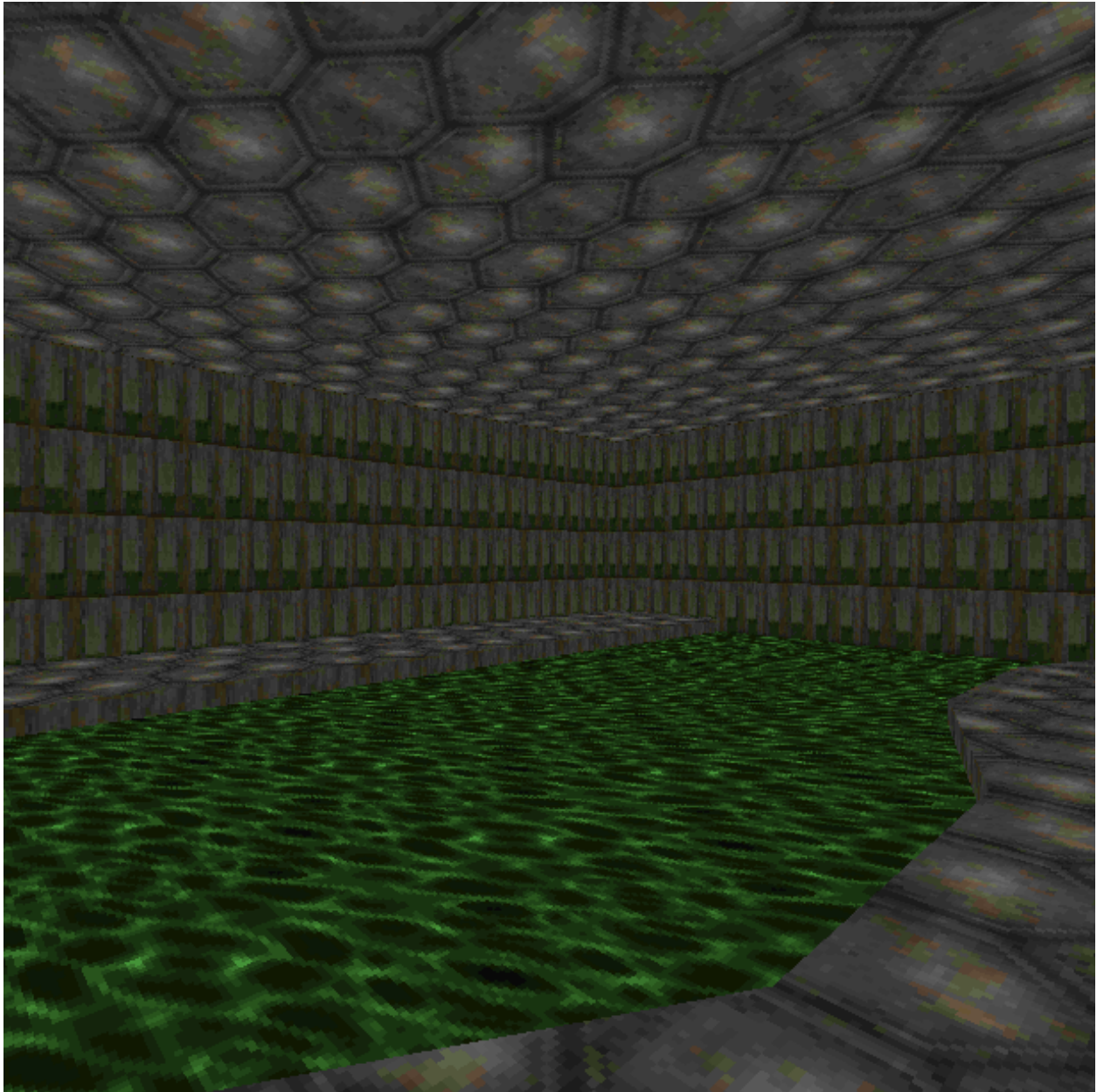


Select each circle in turn, lowering it's floor and adjusting the Sidedef textures to your liking.

## 7.4.2 Downloads

altar.wad

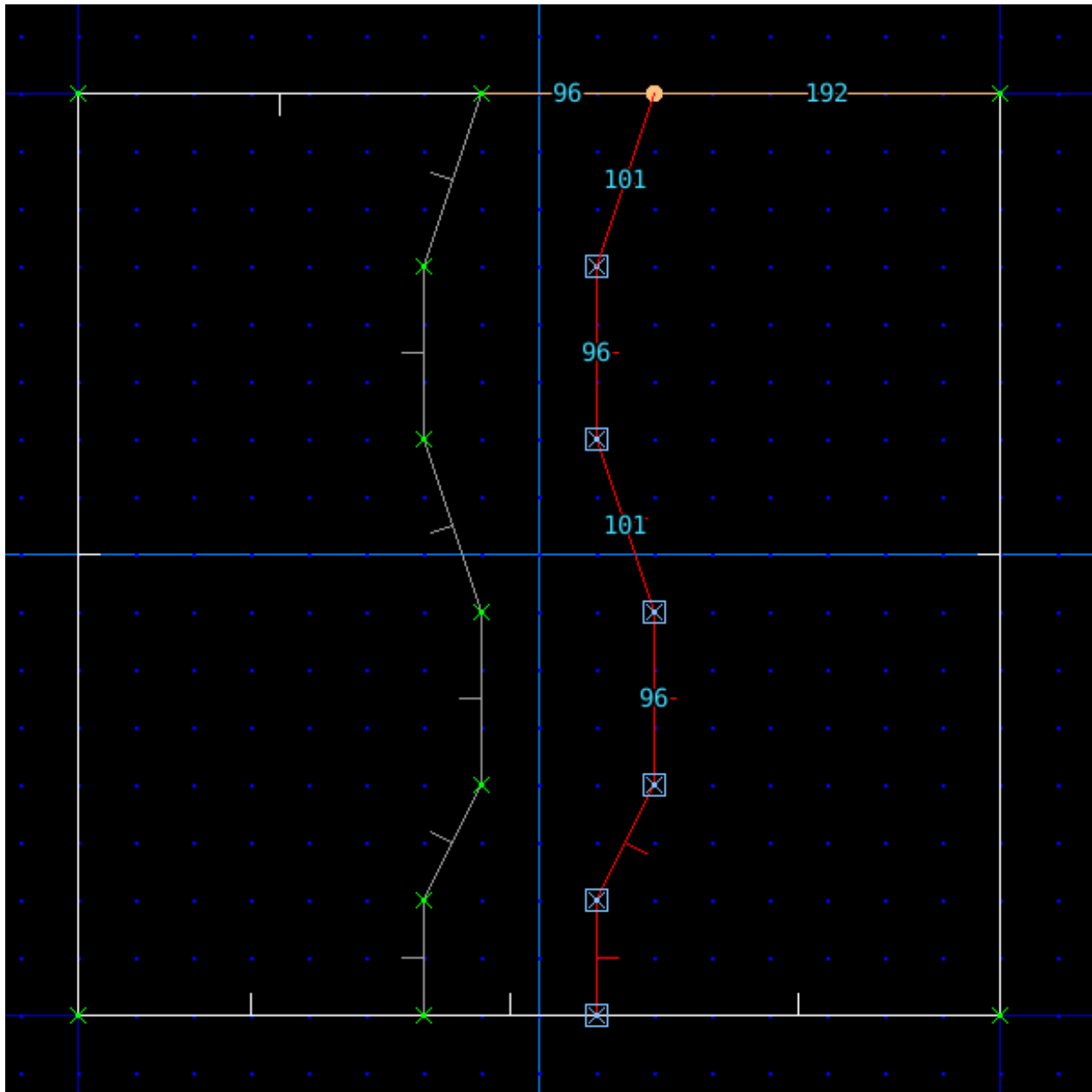
## 7.5 Toxic Pool



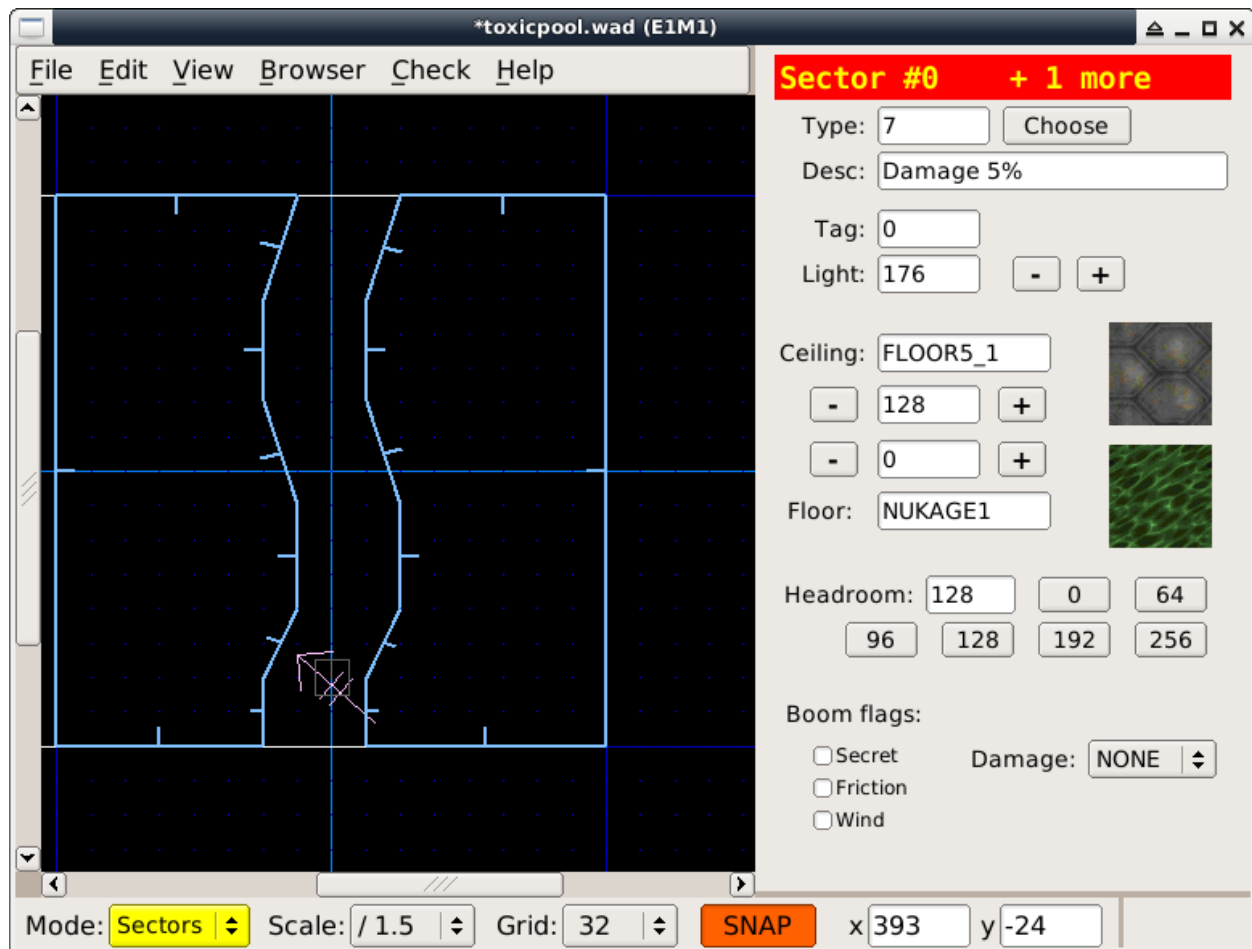
Toxic waste causes damage when walked on.

### 7.5.1 Method

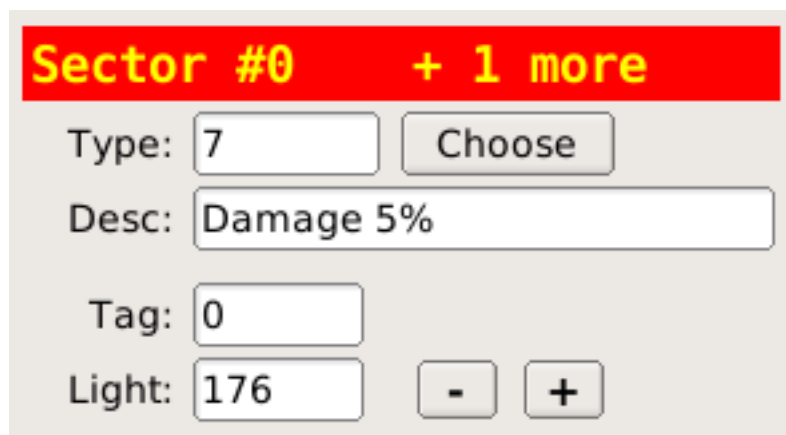
- Press `v` for vertices edit mode
- Click the `RMB` to draw a path



- Press `s` for sector edit mode
- Press ``` to deselect everything
- Select all the sectors that will become the toxic pools
- Click the Floor texture button and pick the *NUKAGE1* texture



- Choose the sector type using the **Choose** button



- Pick the **Damage 5%** type

X

## Sector Types

Category: ALL ⬆ ⬆

Match:

☐ Alpha

⬆

0/ NOTHING

1/ Light Blinks Randomly

2/ Light Flashes 2 Hz

3/ Light Flashes 1 Hz

4/ Flashes / 20% Damage

5/ Damage 10%

7/ Damage 5%

8/ Light Oscillates

9/ Secret Area

10/ Close after 30 sec

11/ End level / damage

12/ Flashes 1 Hz sync

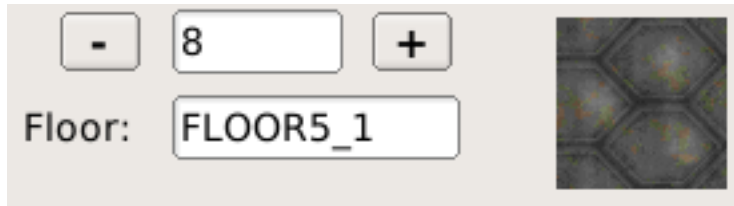
13/ Flashes 2 Hz sync

14/ Open after 5 minute

16/ Damage 20%

17/ Light Flickering

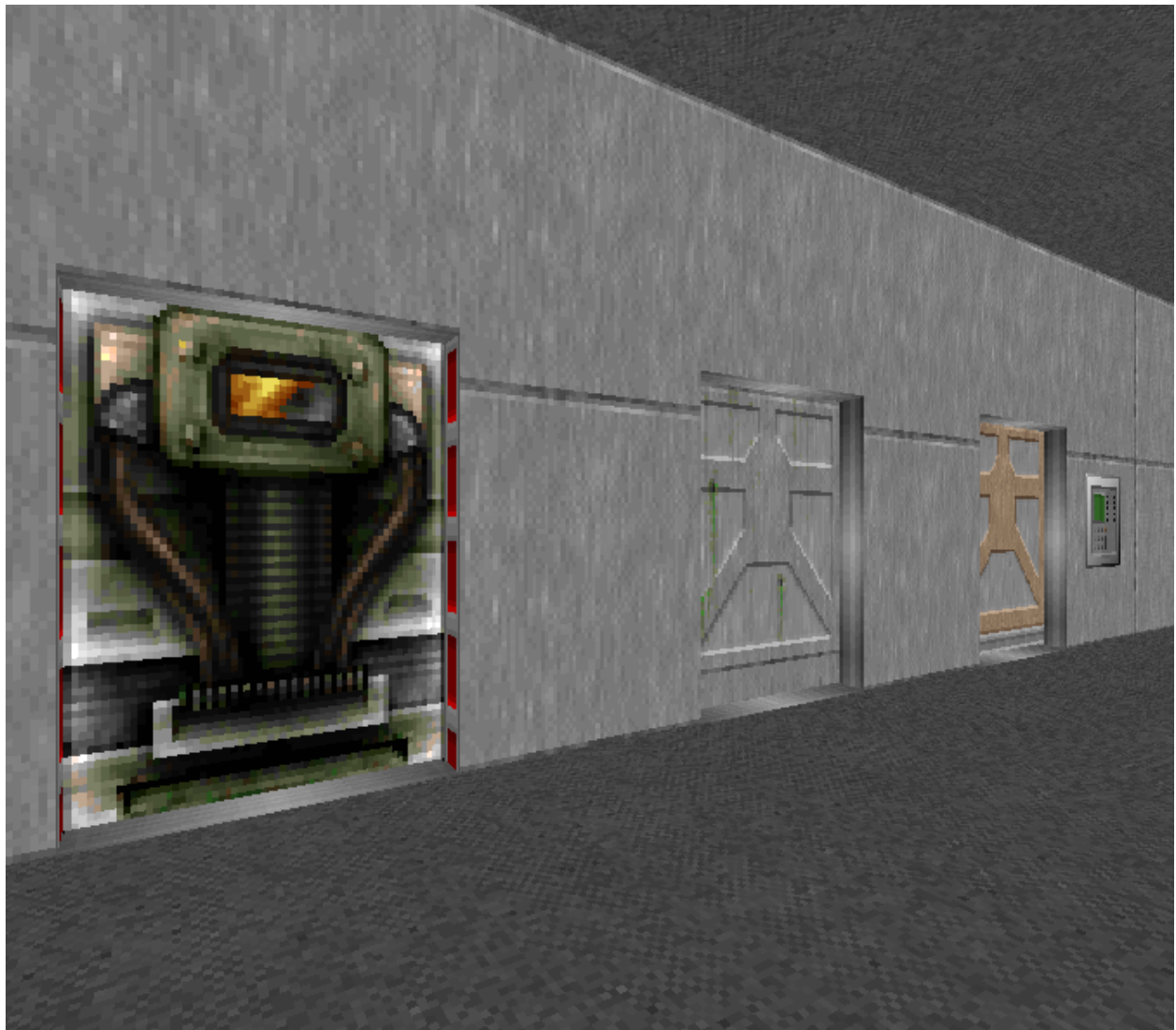
- Press ` to deselect everything
- Select the sector that will become the walkable path
- Raise the Floor height by clicking the floor + button
- Alternatively press the , . keys to adjust floor height via keyboard



## 7.5.2 Downloads

`toxicpool.wad`

## 7.6 Doors



Manual, remote and locked doors.



---

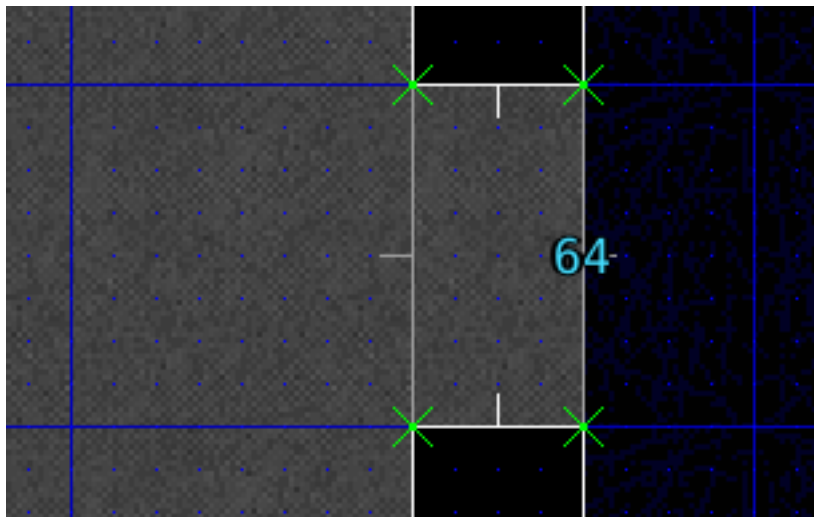
**Note:** Door textures come in two sizes: 64 and 128. When making your doorway, opt for a passage width to match.

---

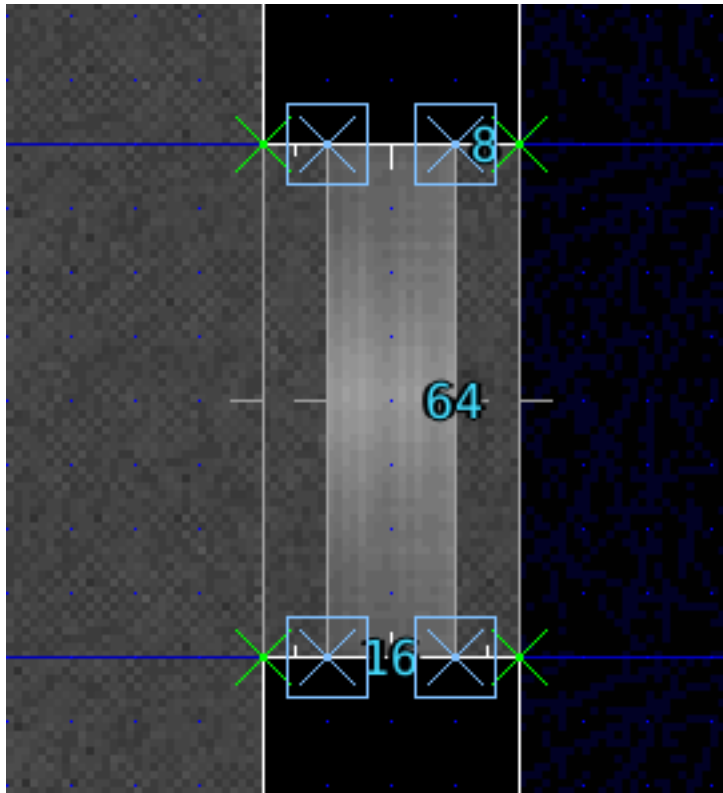
### 7.6.1 Building a door

Use this method to make a basic door. It can be used as a base for a manual or a remote door.

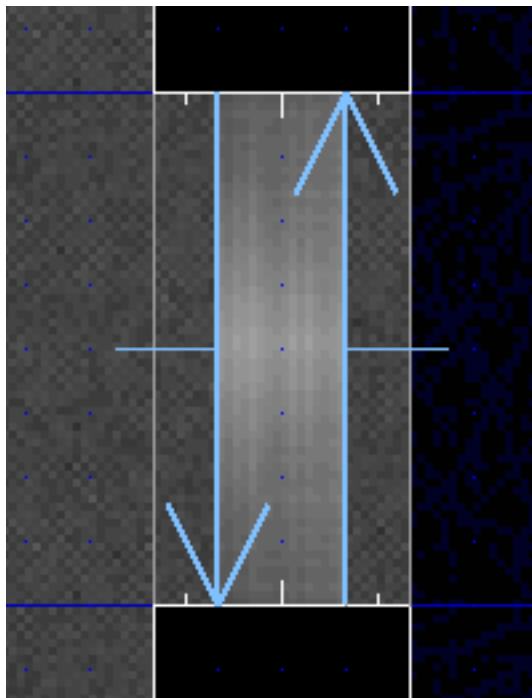
- Join two sectors together with a passage



- Increase the grid detail as needed by pressing 3–5
- Enter vertices edit mode (v)
- Add vertices for the door inside the passage



- Enter linedef edit mode (1) and select the front/rear sides of the door



- The door raises into the ceiling, set the front upper sidedef texture to the *ICKDOOR1* texture.

**Linedef #44 + 1 more**

Type:

Desc:

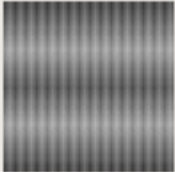

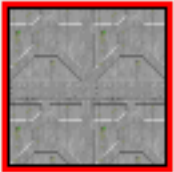
Tag:  Length:

Flags: ☐ upper unpeg ☐ impassible  
☐ lower unpeg ☐ block mons  
☐ pass thru ☐ sound block

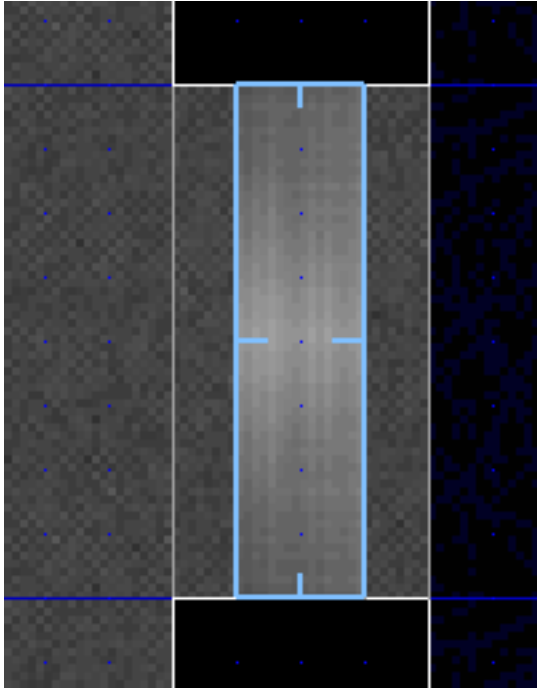
Vis:

**Front Sidedef: #52**

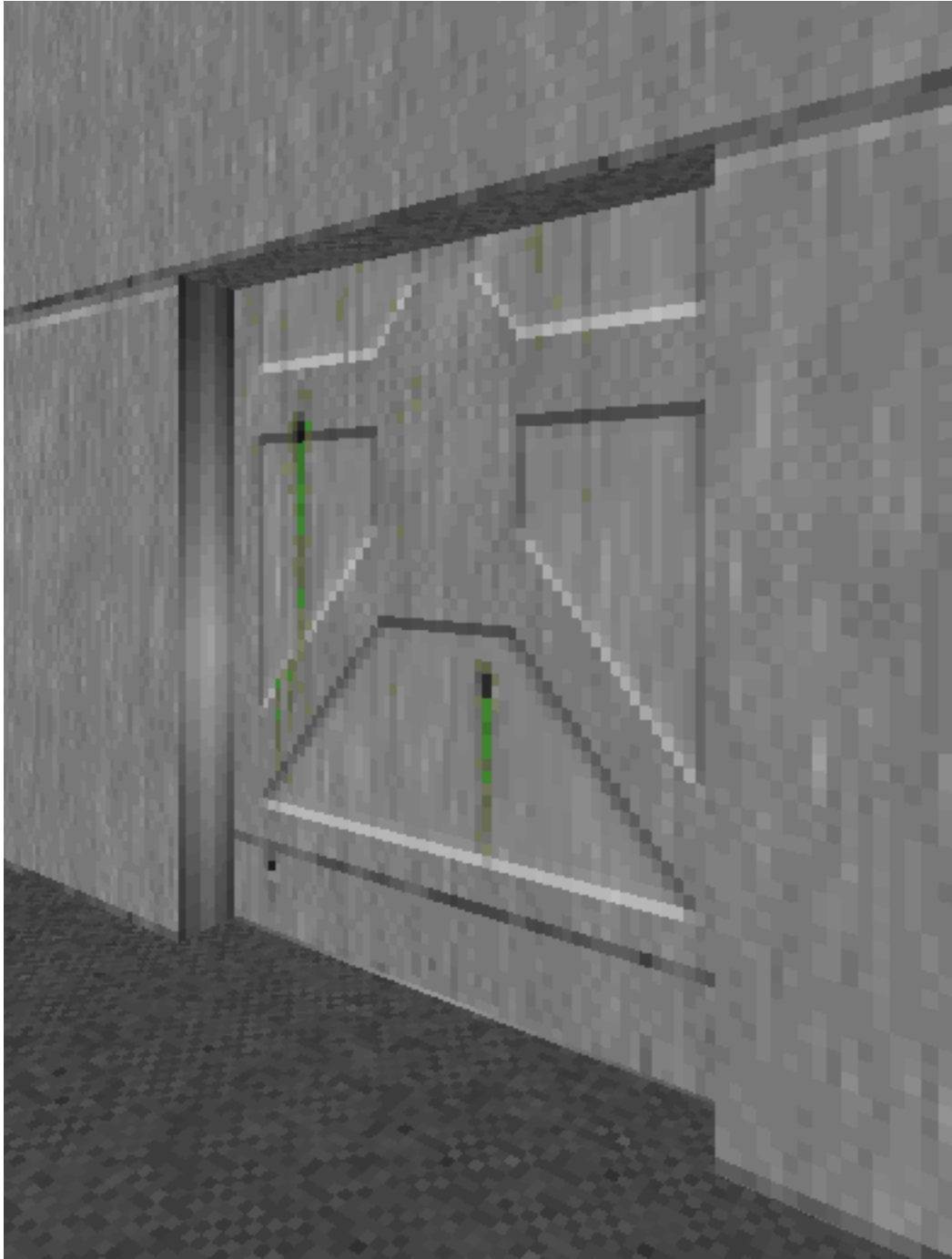
x:  y:  sec:

		
<input type="text" value="DOORSTOP"/>	<input type="text" value="-"/>	<input type="text" value="ICKDOOR1"/>

- Enter sector edit mode (s) and select the door sector



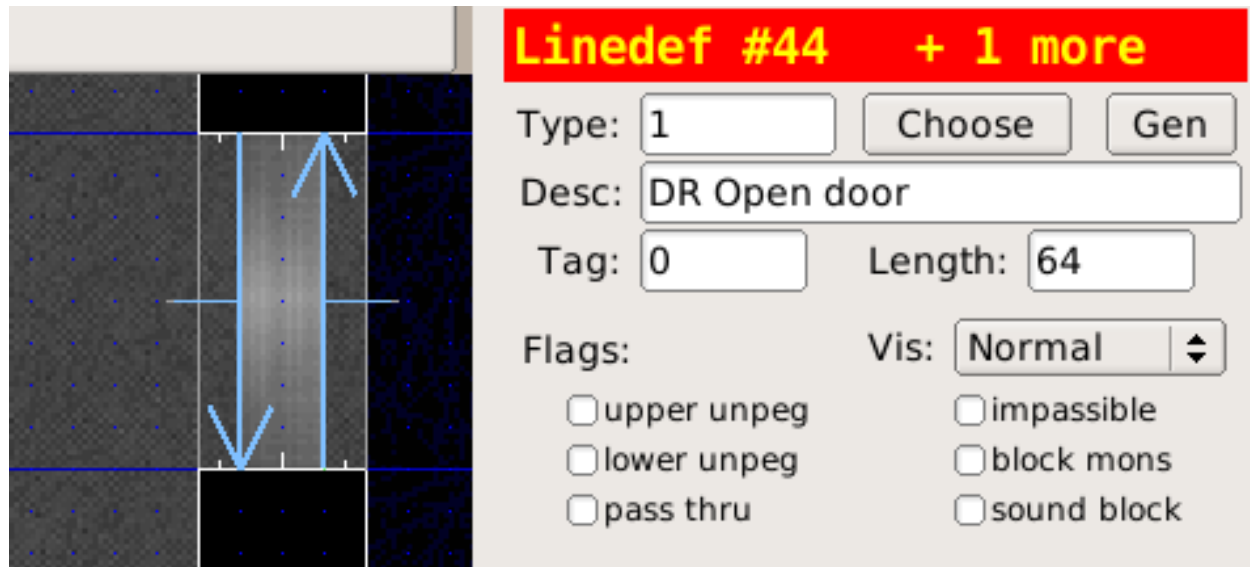
- Lower the ceiling all the way to the floor so that the door is closed. Use the **Ceiling** +/- buttons or the [ ] keys.



### 7.6.2 Manual Doors

Manual doors open when the player performs the USE action on the door.

- Enter linedef edit mode (1)
- Select both linedefs of the door



- Choose the **Type** of the linedefs as *1 DR Open Door*

To make the door open on a fixed track (the sides stay still while opening and closing):

- Select the track linedefs
- Check the **upper unpeg** and **lower unpeg** options

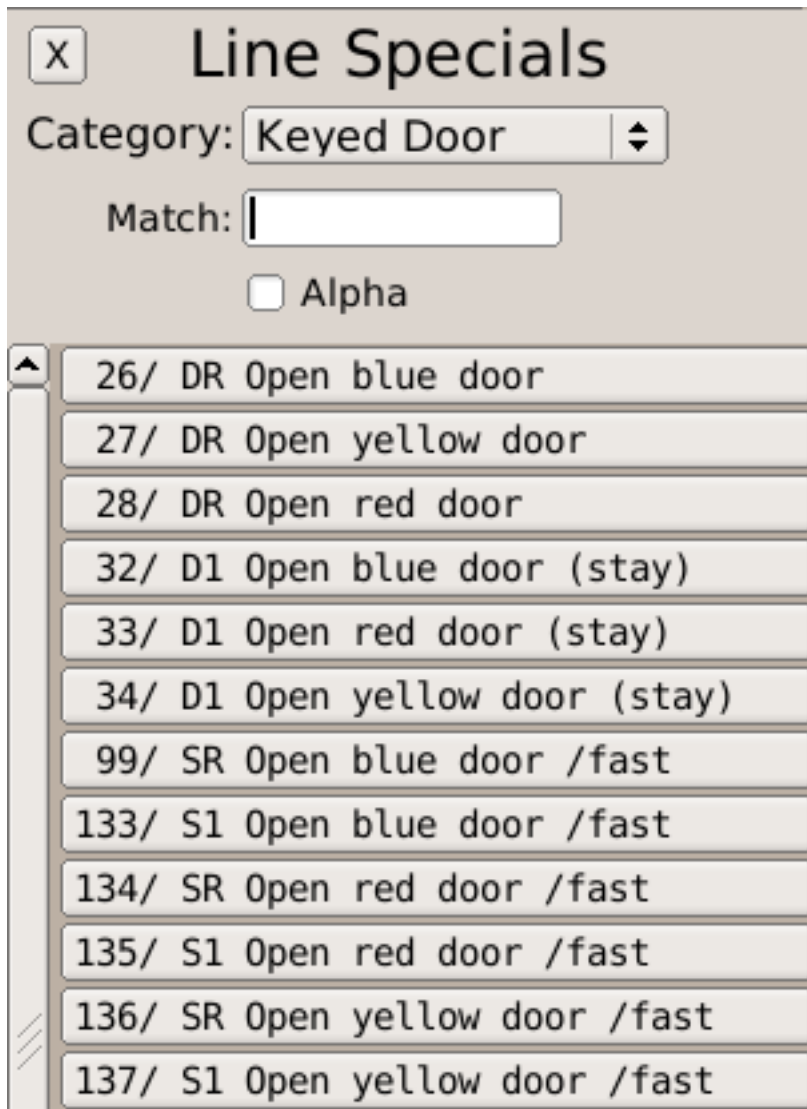


**Note:** The door line specials indicates that the sector facing the back of the linedef is a door, this special does not need a tag either.

The DR special can be opened repeatedly, while D1 can only be opened once.

### 7.6.3 Locked Doors

Doors that require a blue, yellow or red key to open are created similarly to manual doors. When choosing the door linedef type, pick one of the specials that target the keys:



**Line Specials**

Category: **Keyed Door**

Match:

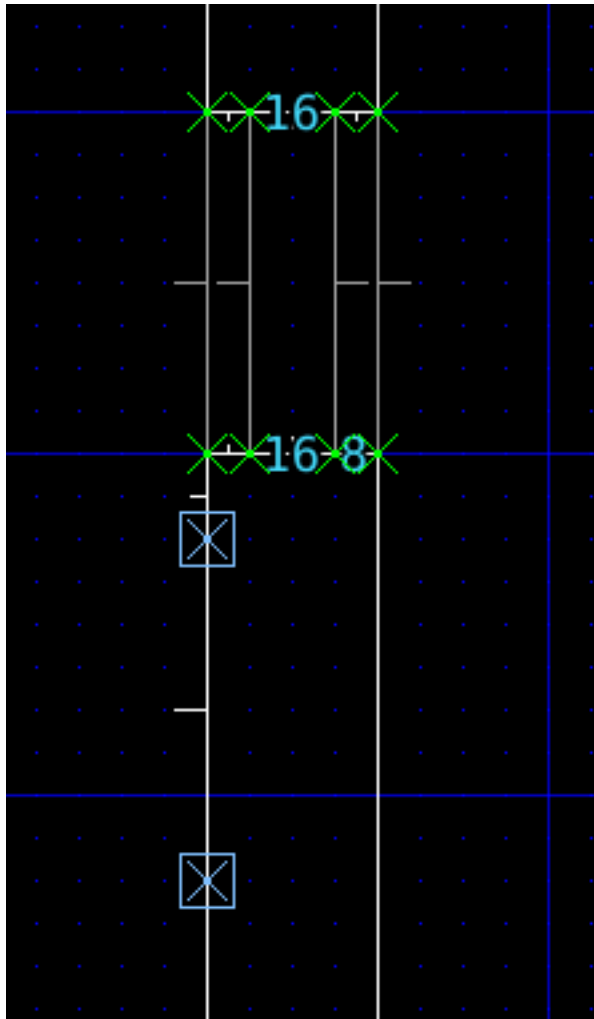
☐ Alpha

- 26/ DR Open blue door
- 27/ DR Open yellow door
- 28/ DR Open red door
- 32/ D1 Open blue door (stay)
- 33/ D1 Open red door (stay)
- 34/ D1 Open yellow door (stay)
- 99/ SR Open blue door /fast
- 133/ S1 Open blue door /fast
- 134/ SR Open red door /fast
- 135/ S1 Open red door /fast
- 136/ SR Open yellow door /fast
- 137/ S1 Open yellow door /fast

### 7.6.4 Remote Doors

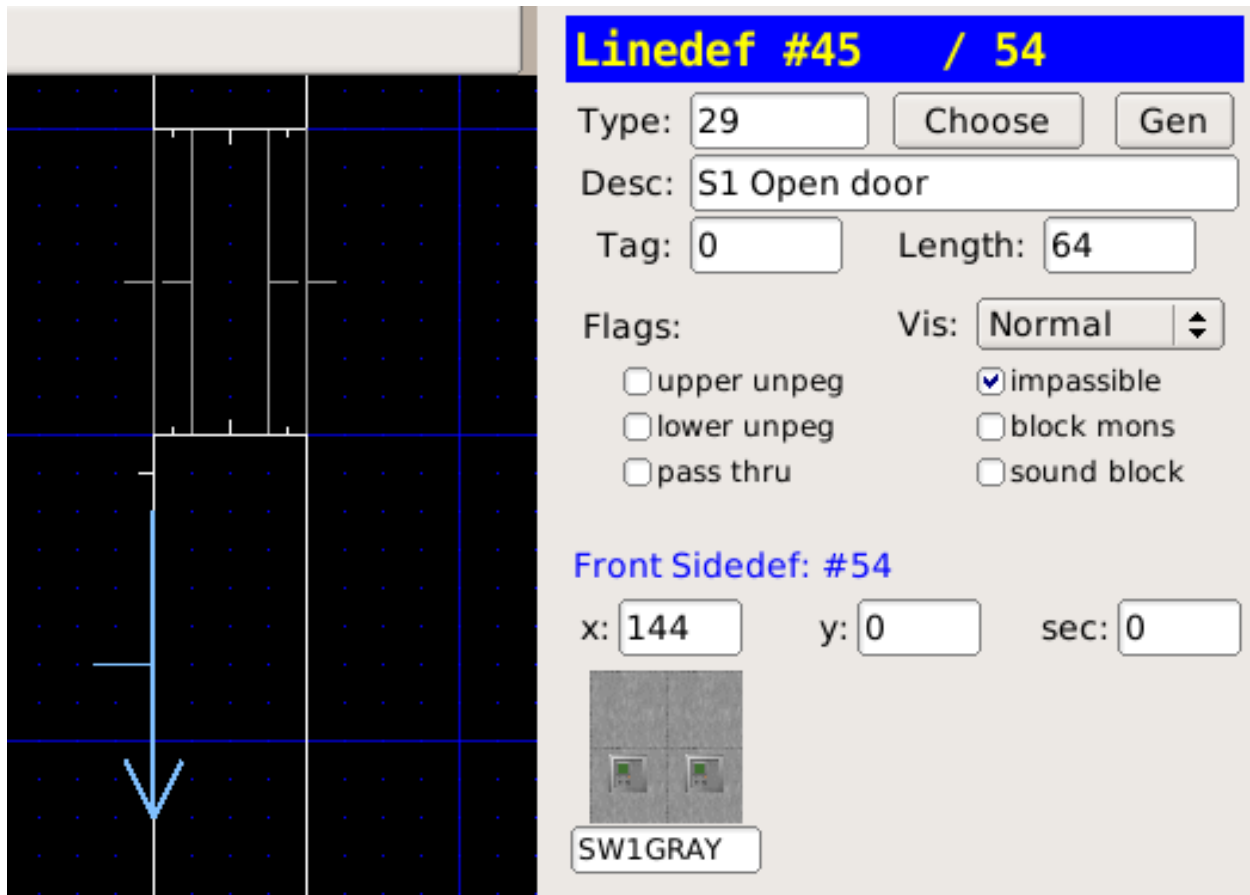
Remote doors are opened through a switch.

- Enter vertex edit mode (v)
- Use the LMB to insert vertices along the wall, make the linedef 64 units long

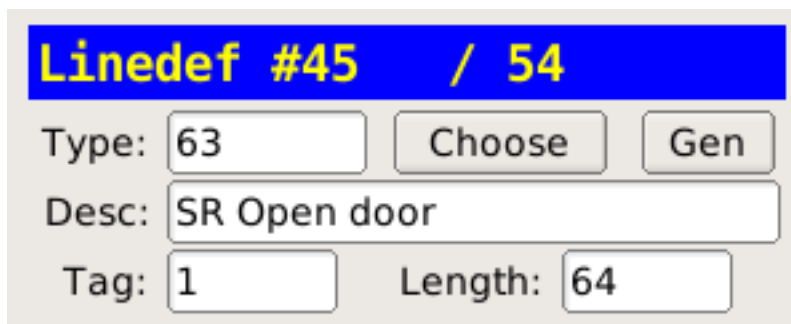


- Enter linedef edit mode (1), select the new linedef
- Choose the *SWIGRAY* switch texture

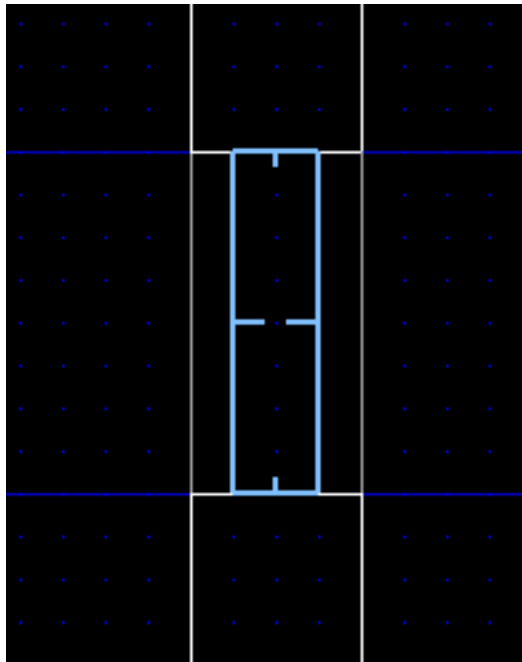




- Choose the linedef **Type** as *63 SR Door Open*
- Move the mouse cursor over the grid to ensure focus is not stolen by the **Line Specials** panel
- Press ; then f to apply a fresh tag to the linedef



- Enter sector edit mode (s), select the door sector



- Press `;` then `1` to apply the last tag to the door sector

**Sector #11 / 12**

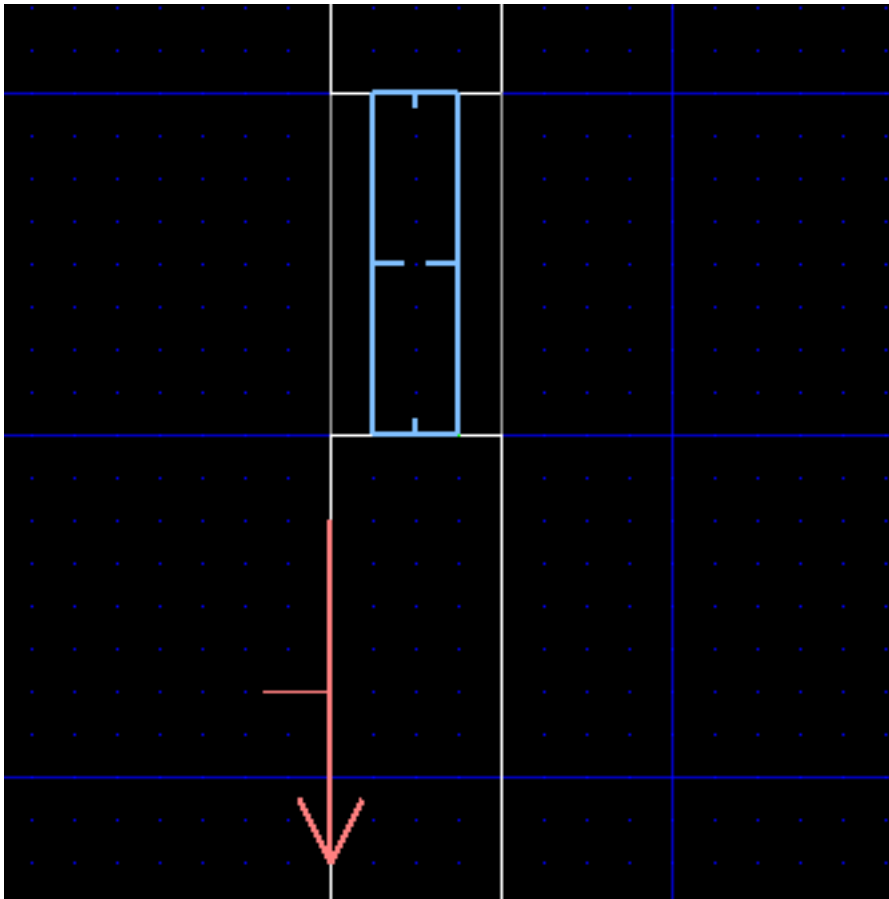
Type:

Desc:

Tag:

Light:

- Eureka highlights both the sector and the linedef that share the same tag. This shows us the two are linked:

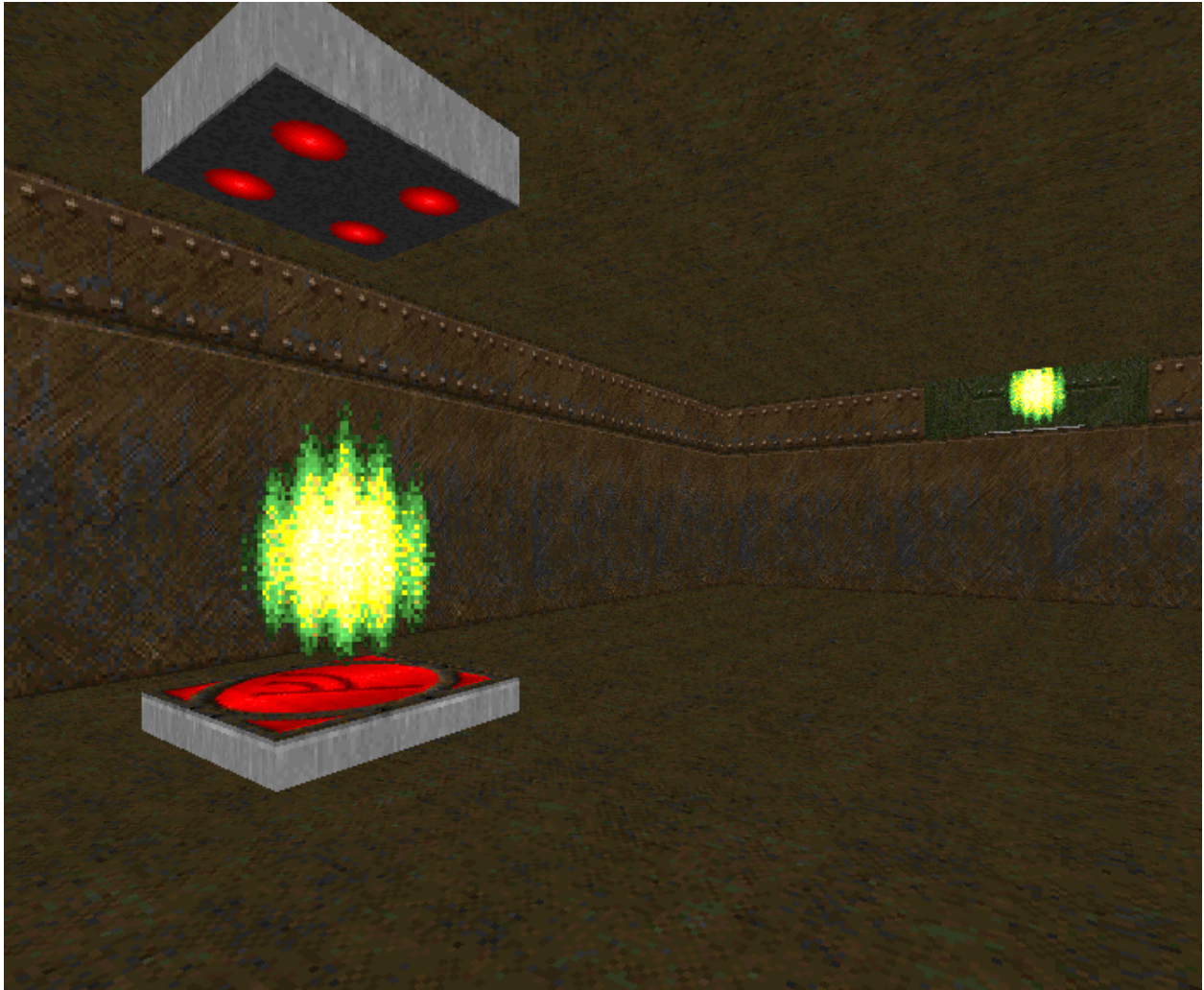


**Note:** The SR line special indicates a switch that can be toggled repeatedly, while S1 is a switch that can only be toggled once.

## 7.6.5 Downloads

`doors.wad`

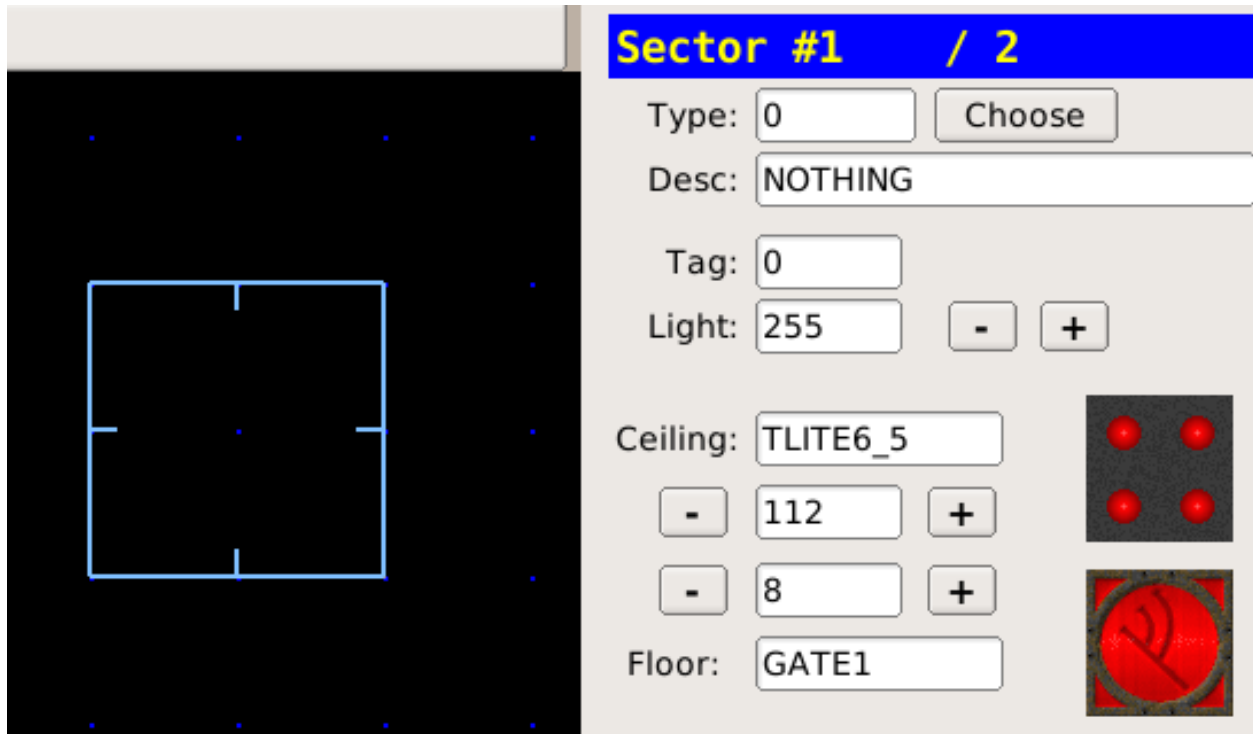
## 7.7 Teleporters



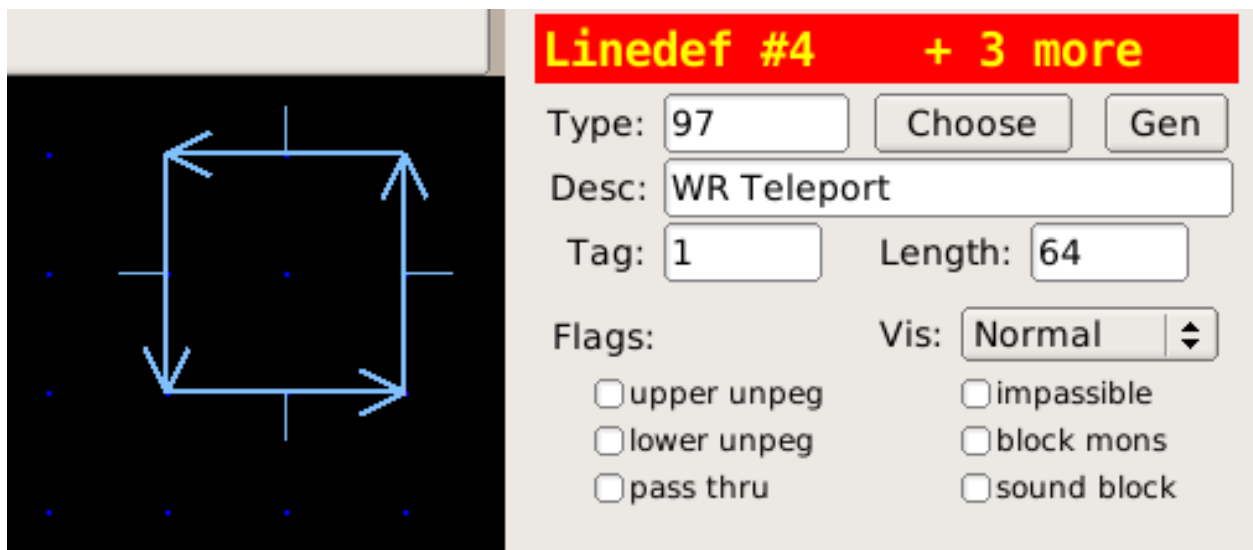
A teleport is triggered by walking over a linedef with the teleport special. The tag of the linedef points to the sector where the teleport lands.

### 7.7.1 Teleport Platform

- Create a 64x64 sector for the teleport platform
- Set the floor texture as *GATE1*, the ceiling as *TLITE6\_5*



- Enter linedef edit mode (1)
- Press ; then f to apply a fresh tag to all four linedefs. This will point to the teleport landing sector.

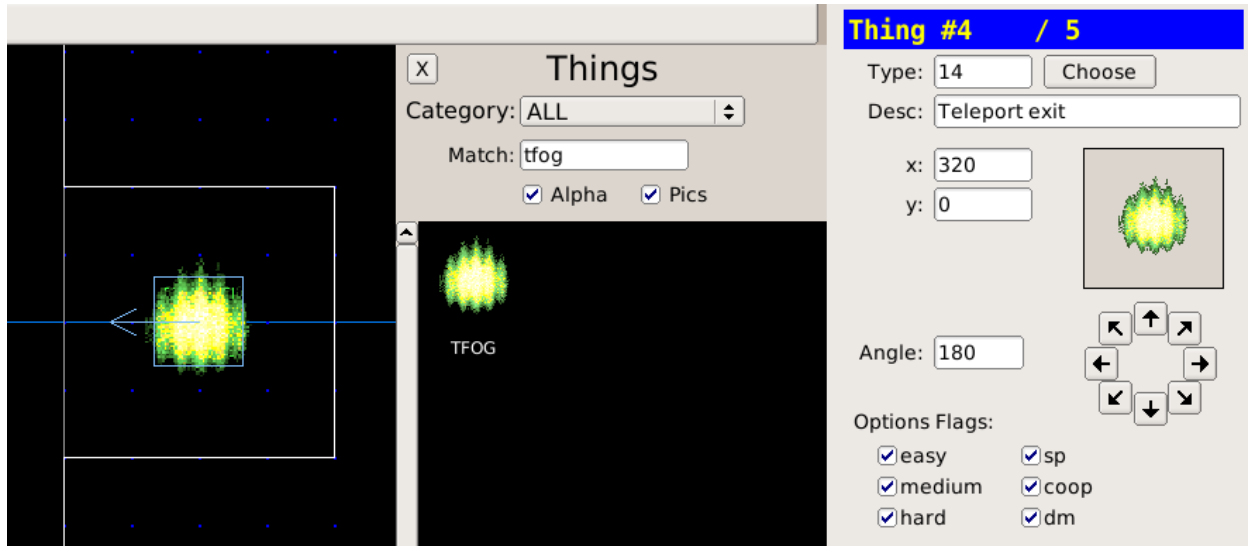


- Choose the linedef **Type** as 97 *WR Teleport*

**Note:** A teleport is only triggered when walking from the **Front** to the **Back** of a linedef. This is intentional as it allows the player to walk off the landing platform without triggering another unintended teleport.

## 7.7.2 Landing Site

- Tag the landing sector the same as the teleporter linedefs. Press `;` then `1` to apply the last used tag to the landing sector.
- Enter Thing edit mode (`⌘`), position the mouse cursor inside the landing sector and press `ins`. Choose the **Type** as *14 Teleport Exit* (labelled as *TFOG* in the thing browser)

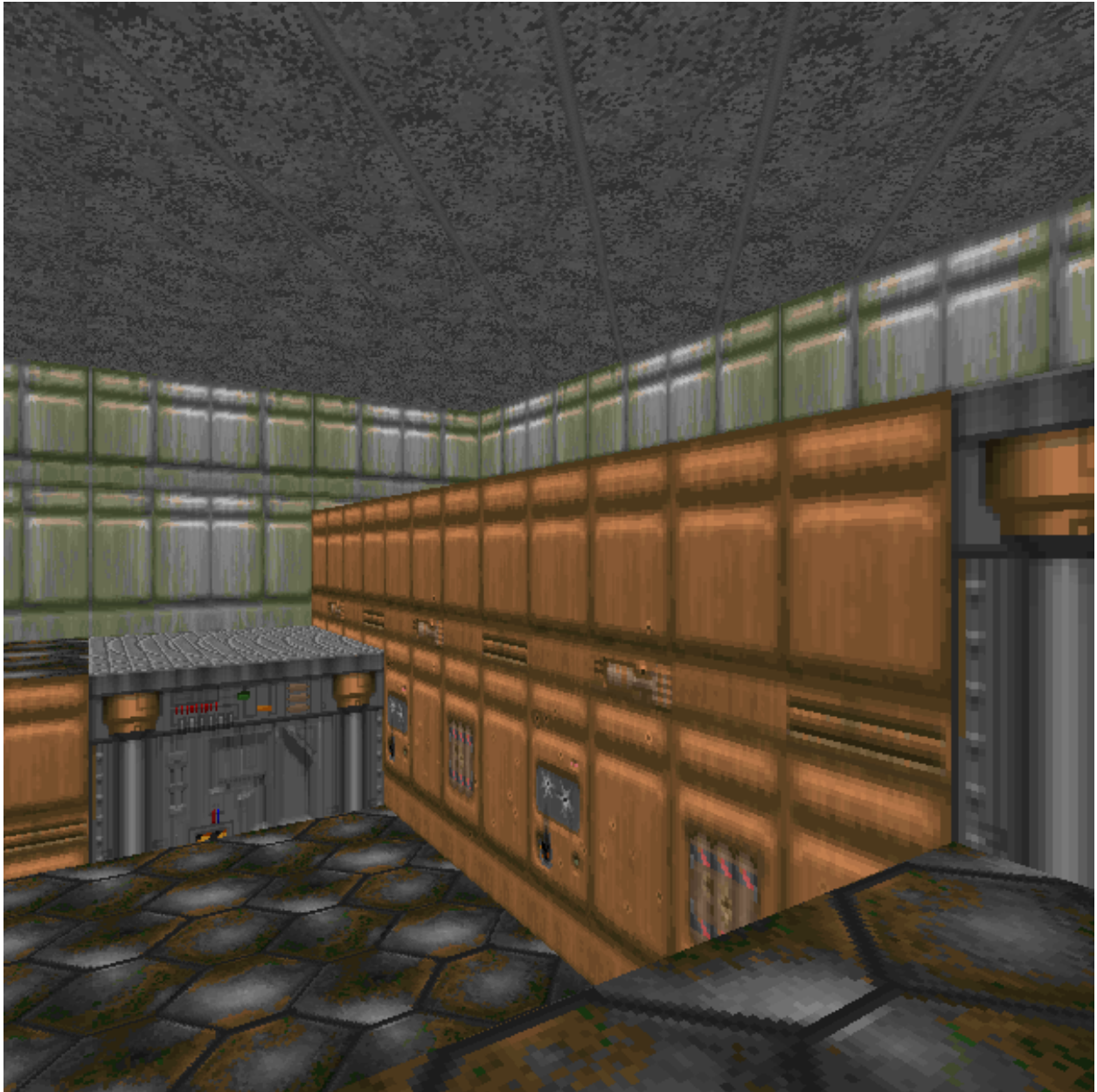


- Click the directional arrows to set the angle of the *TFOG* thing. This is the angle faced after teleporting.

## 7.7.3 Downloads

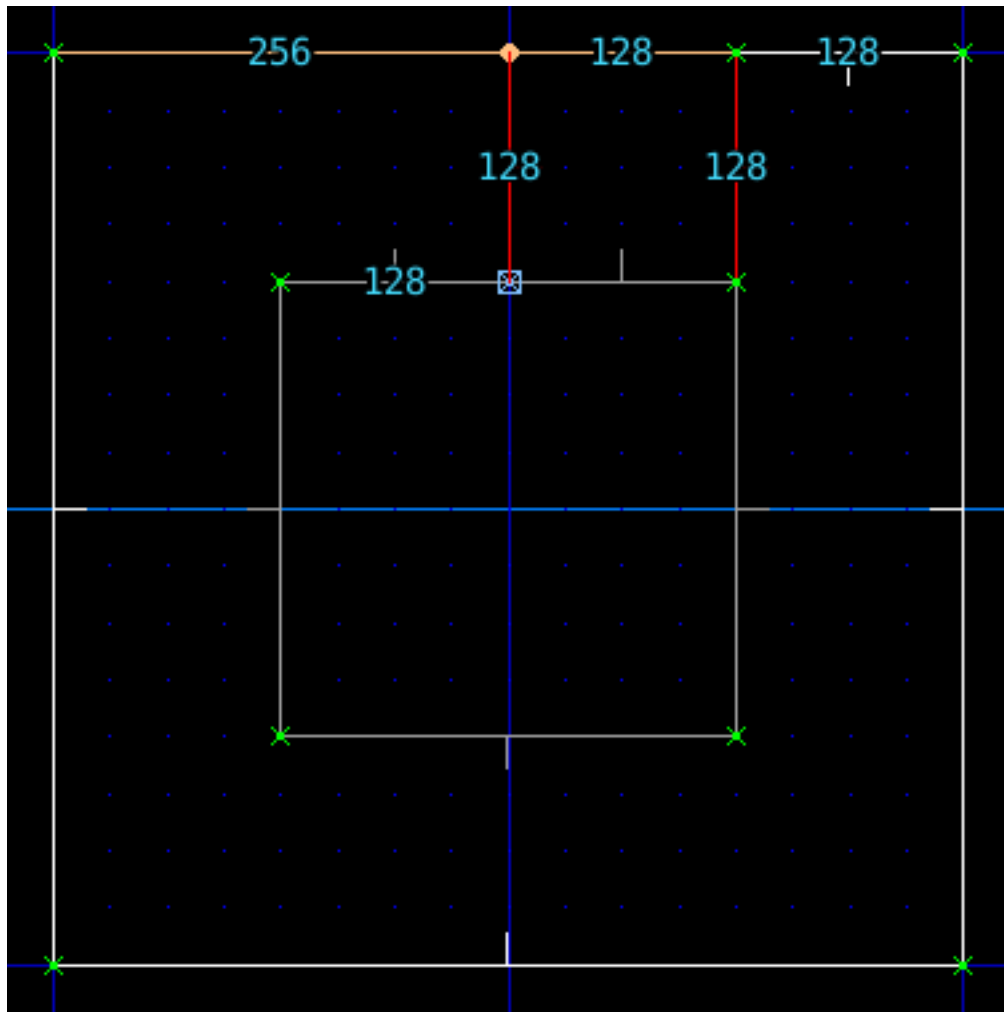
teleporters.wad

## 7.8 Lifts



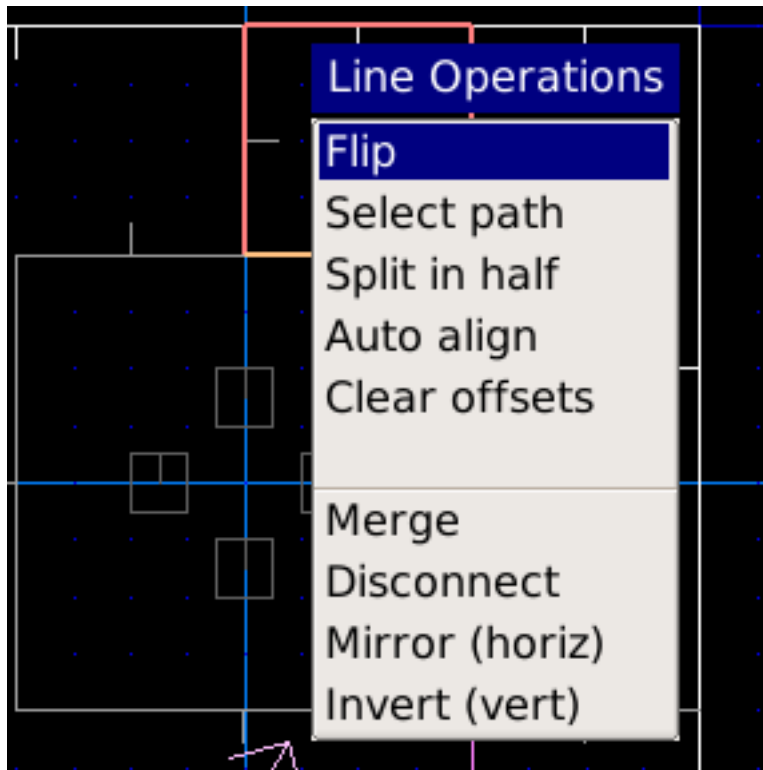
### 7.8.1 Method

- Enter vertex edit mode and create the lift platform. This can be a simple split of a raised sector:

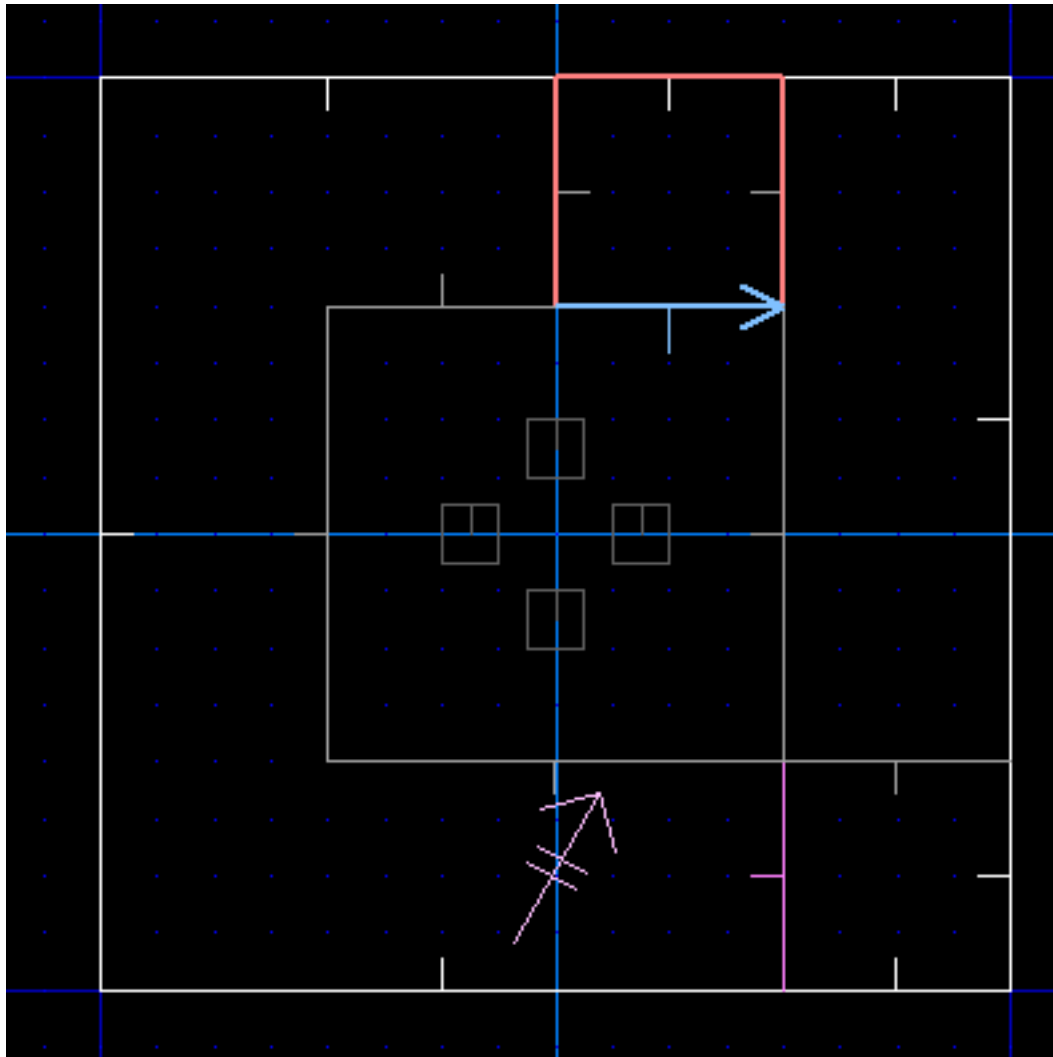


- Ensure the linedef on the front of the lift is facing outward. A lift is triggered from the **Front** linedef.
- If the linedef is facing the wrong way, you can select it and press F1 for the operations menu, and select *Flip*:





- Enter linedef edit mode, select the front of the lift



- Choose the **Type** as *62 SR Lower Lift*
- Press ; then £ to apply a fresh tag to the linedef

**Linedef #4 / 17**

Type:

Desc:

Tag:

Length:

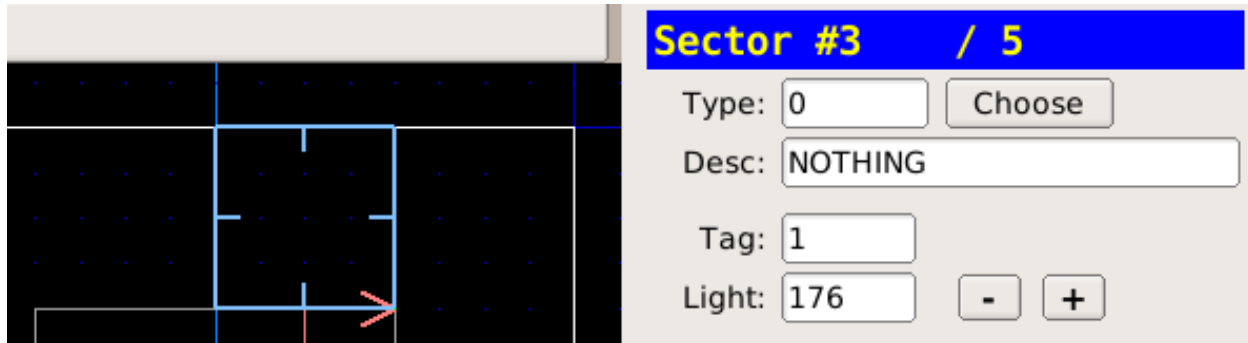
Flags:

☐ upper unpeg  
☐ lower unpeg  
☐ pass thru

☐ impassible  
☐ block mons  
☐ sound block

Vis:

- Enter sector edit mode (s), select the lift sector
- Press ; then 1 to apply the last tag to the sector



**Note:** The SR line special indicates Switch Repeating, so the lift can be used over and over.

## 7.8.2 Downloads

lifts.wad

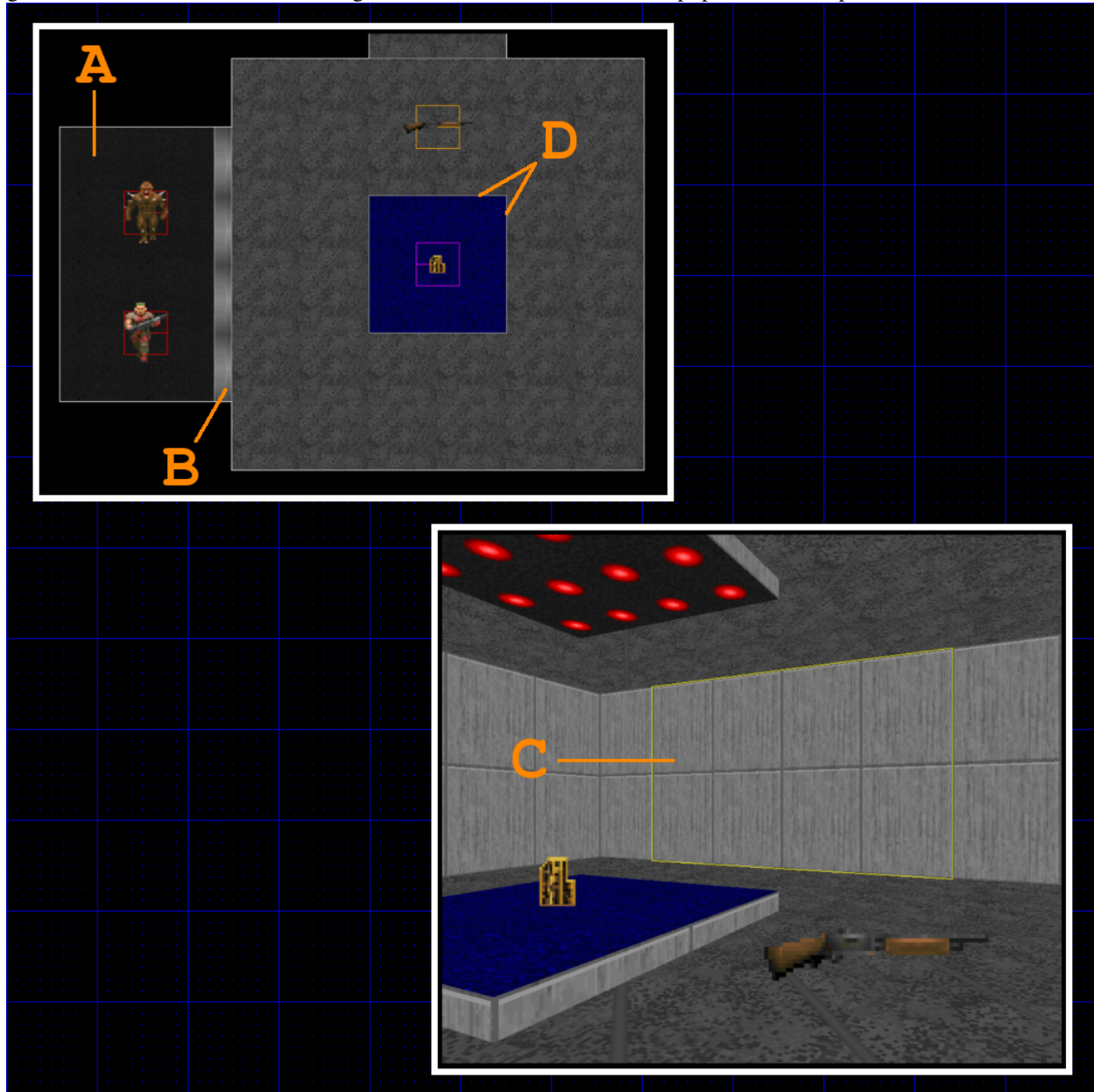
## 7.9 Traps

### 7.9.1 Examples

All trap examples can be played in traps.wad

## 7.9.2 Monster Closet

Monsters are staged in a hidden room with a door disguised as a wall or other nondescript panel, which opens to sur-

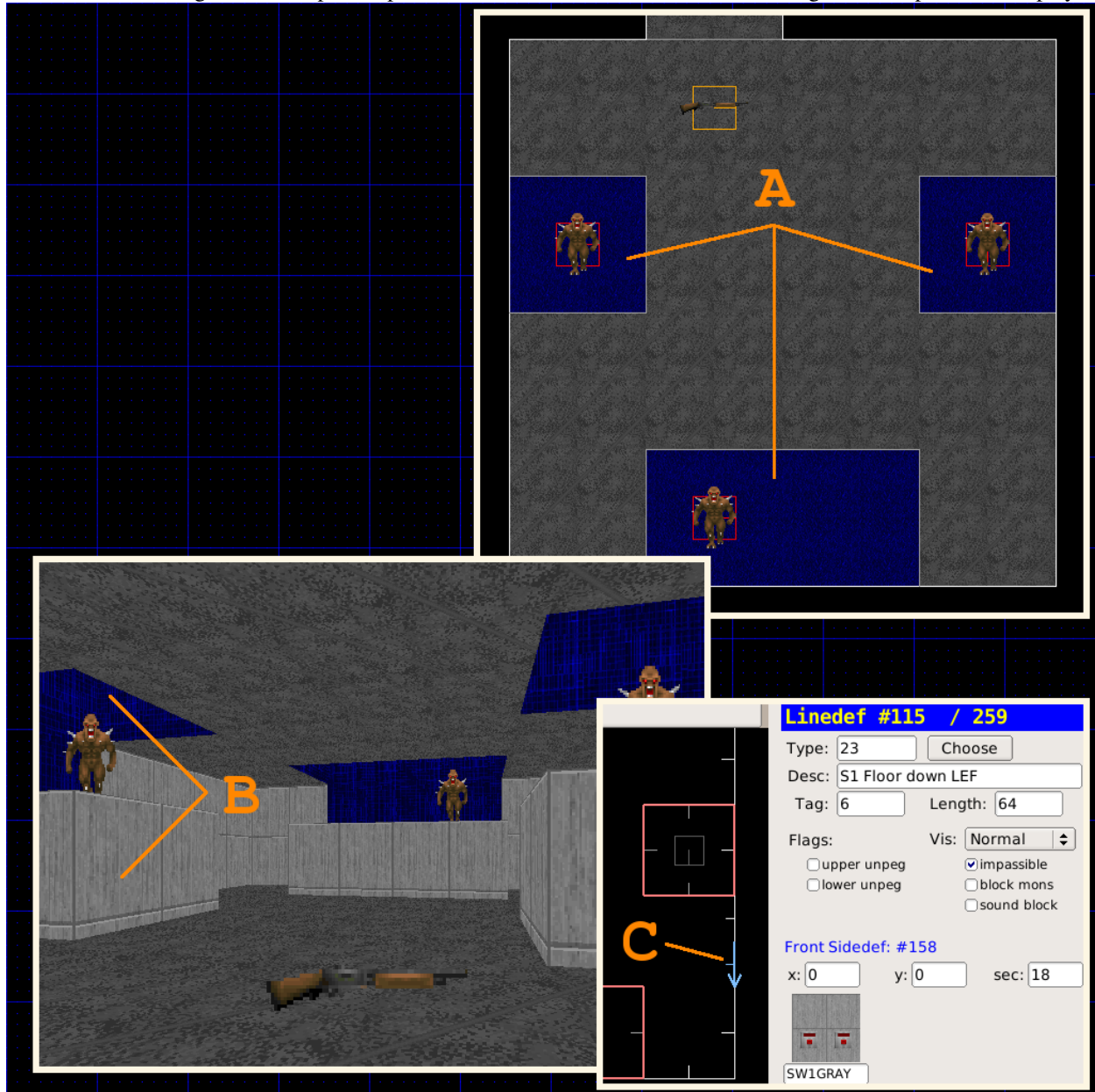


prise the player.

- Create a hidden closet sector (A), add some monsters inside it.
- Create a closet door sector (B).
- Close the closet door so that it looks like a normal wall (C).
- Apply a fresh tag to the closet door sector (B).
- Add the trigger lines that will open the closet, and apply the same tag on them (D).
- Assign the linedef special **Type 109 W1 Open and stay fast** to the trigger lines (D).

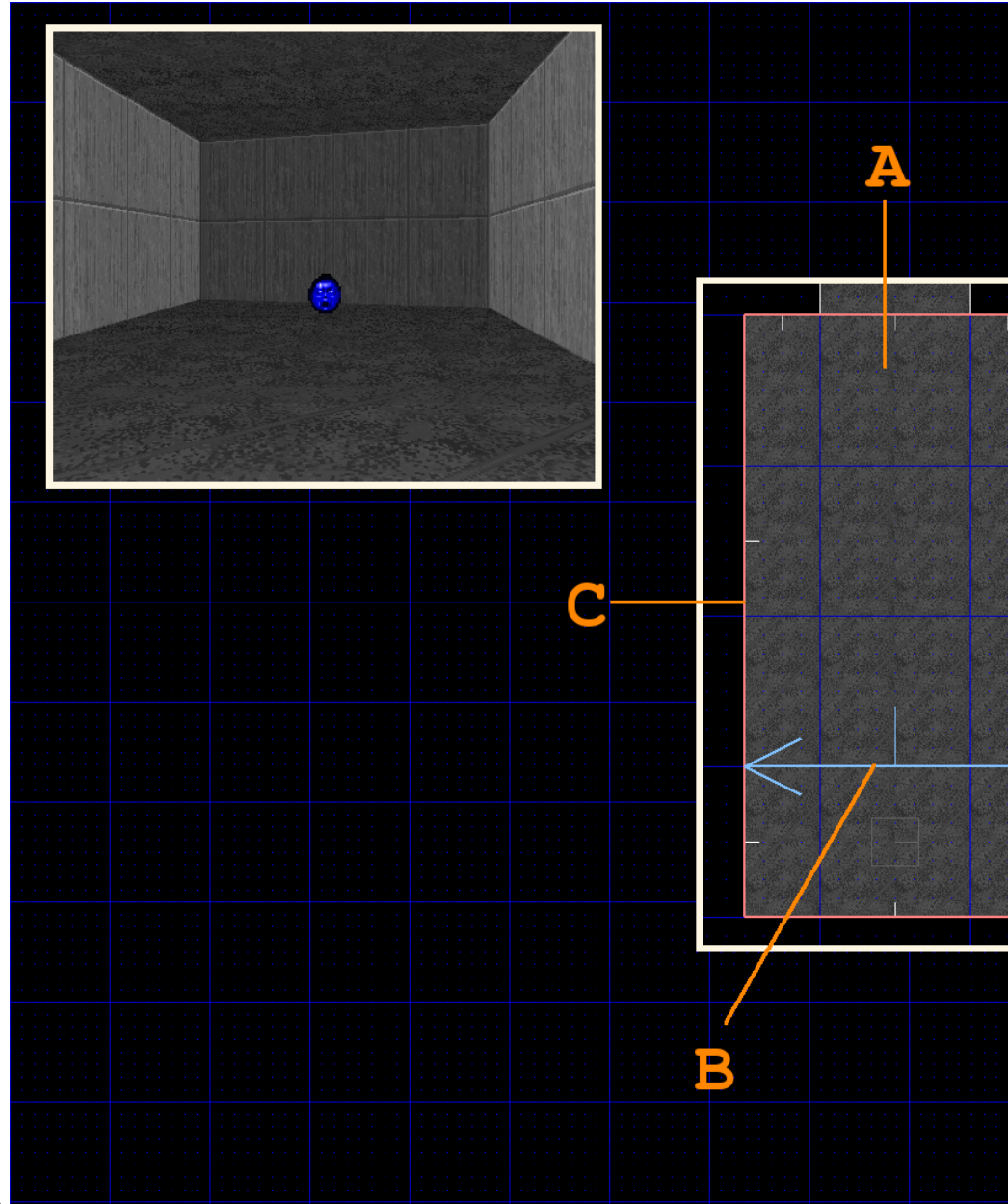
### 7.9.3 Monster Platform

Monsters are staged on top a platform that lowers from the ceiling to surprise the player.



- Add the platform sectors with some monsters in them (A).
- Raise the platform floors and ceilings (B) until flush with the surrounding ceiling.
- Select the platform sectors and apply a fresh tag (A).
- Insert vertices along a wall to make a 64 unit long line, for the switch (C).
- Set a switch texture (C).
- Apply the platform tags to the switch linedef (C).
- Set the switch linedef (C) type to **23 S1 Floor down LEF**.

## 7.9.4 Crushers



The ceiling lowers to crush the player.

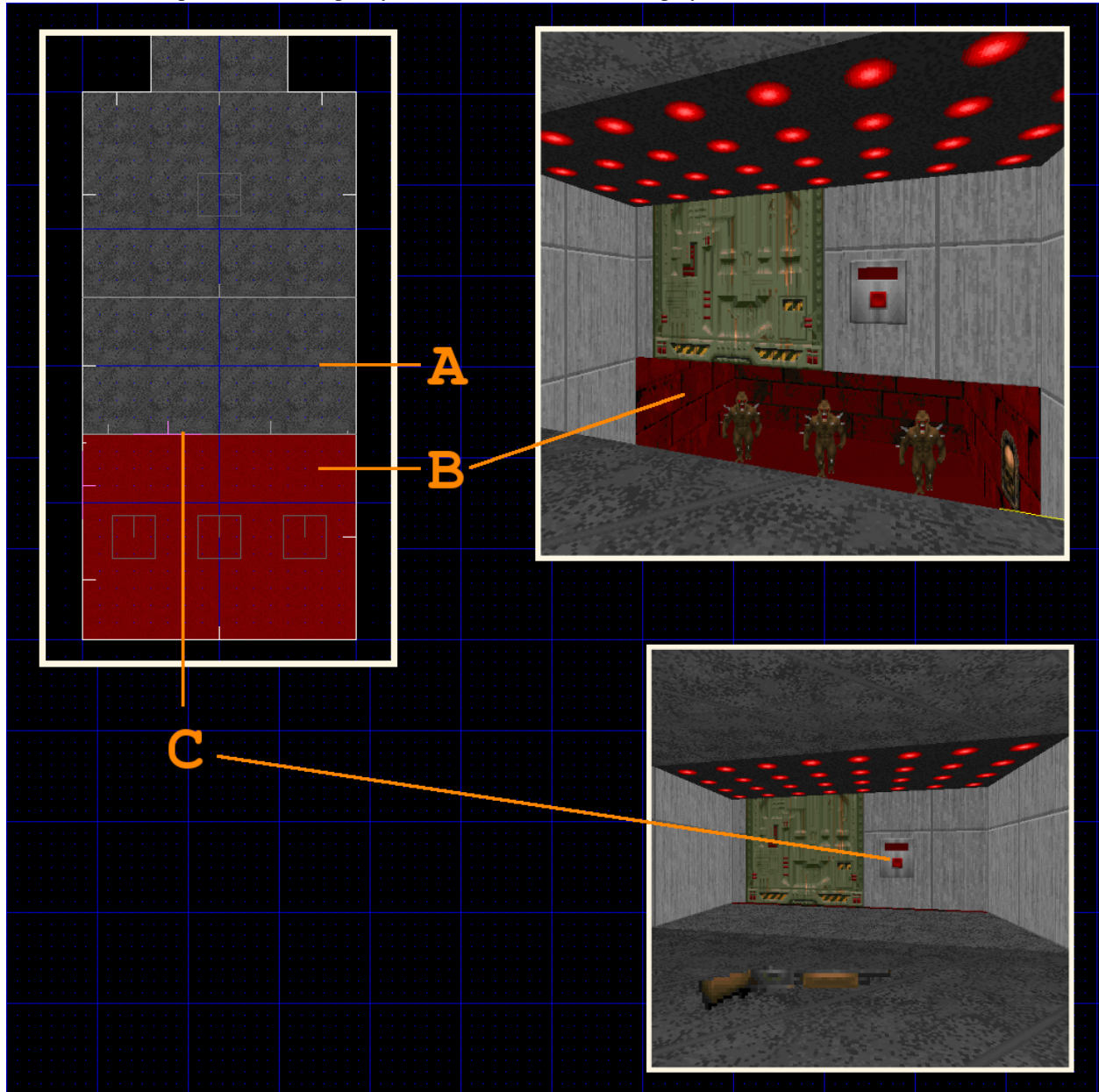
- Assign a fresh tag to the sector that will do the crushing (A).
- Create a linedef that will act as the walk-over trigger (B).
- Assign the same tag to the trigger linedef (B).
- Set the trigger linedef (B) **Type 6 W1 Crusher /fast**

**Note:** You will need to **lower unpeg** the room sector walls (C) if you do not want the walls to move up and down with the crushing ceiling.

**Warning:** Fast crushers do less damage and the player may even survive one round of crushing. Slow crushers take their time, dealing a lot of damage and ensures player death.

### 7.9.5 Drop Trap

The floor drops out unexpectedly from beneath the player into a room with monsters.



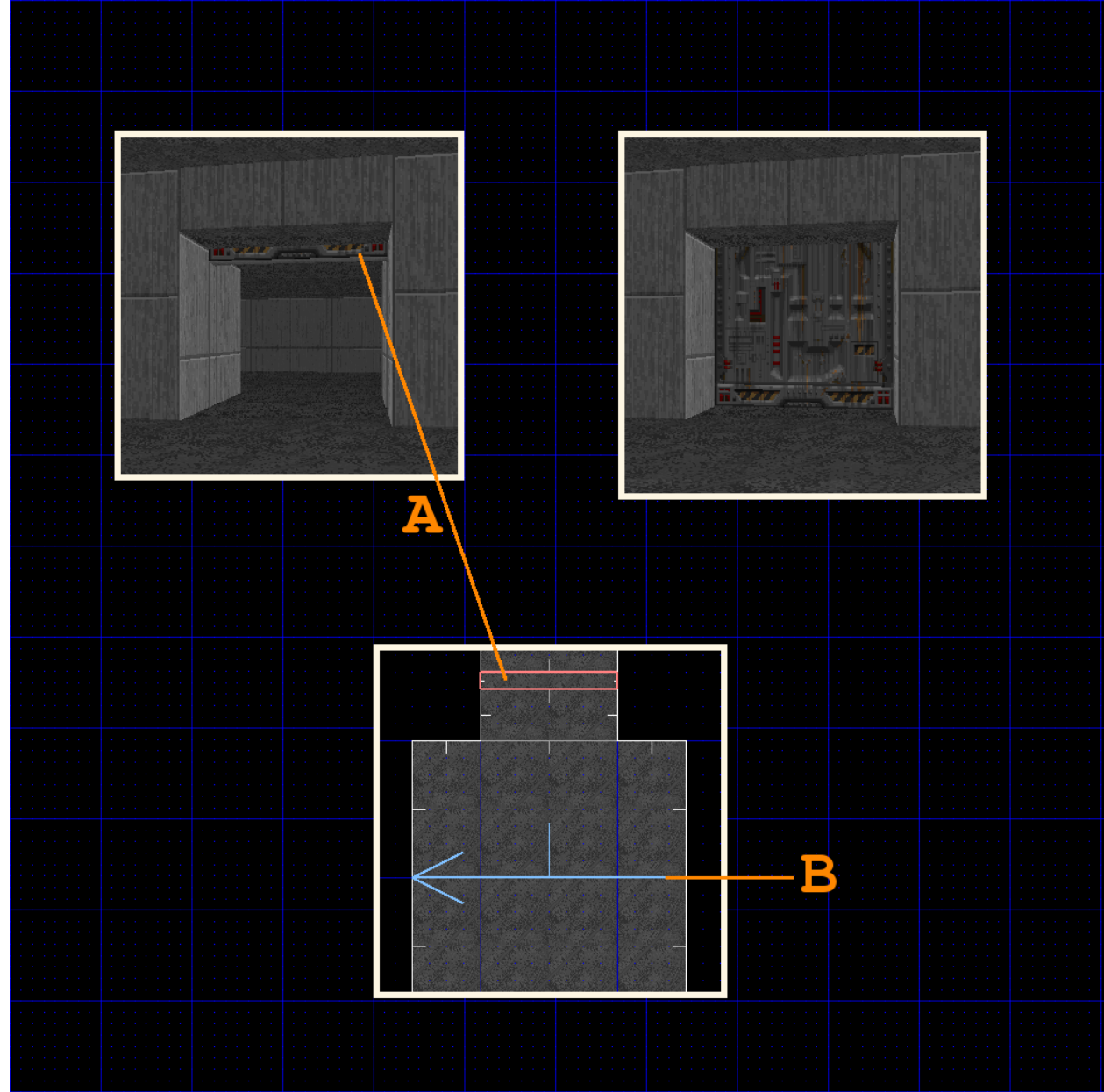
- Add the sector that will drop out beneath the player (A).
- Tag the sector (A).
- Add the hidden room the player will drop into, lower it's floor (B).
- Insert vertices on a wall to create a switch (C).



- Set the tag of the switch linedef (C) to match the drop sector (A).
- Set the switch linedef **Type 123 SR Lift Lower /fast** (C).

### 7.9.6 Lock-In Trap

Lock all the exits in a room, forcing the player into close combat for a short time.



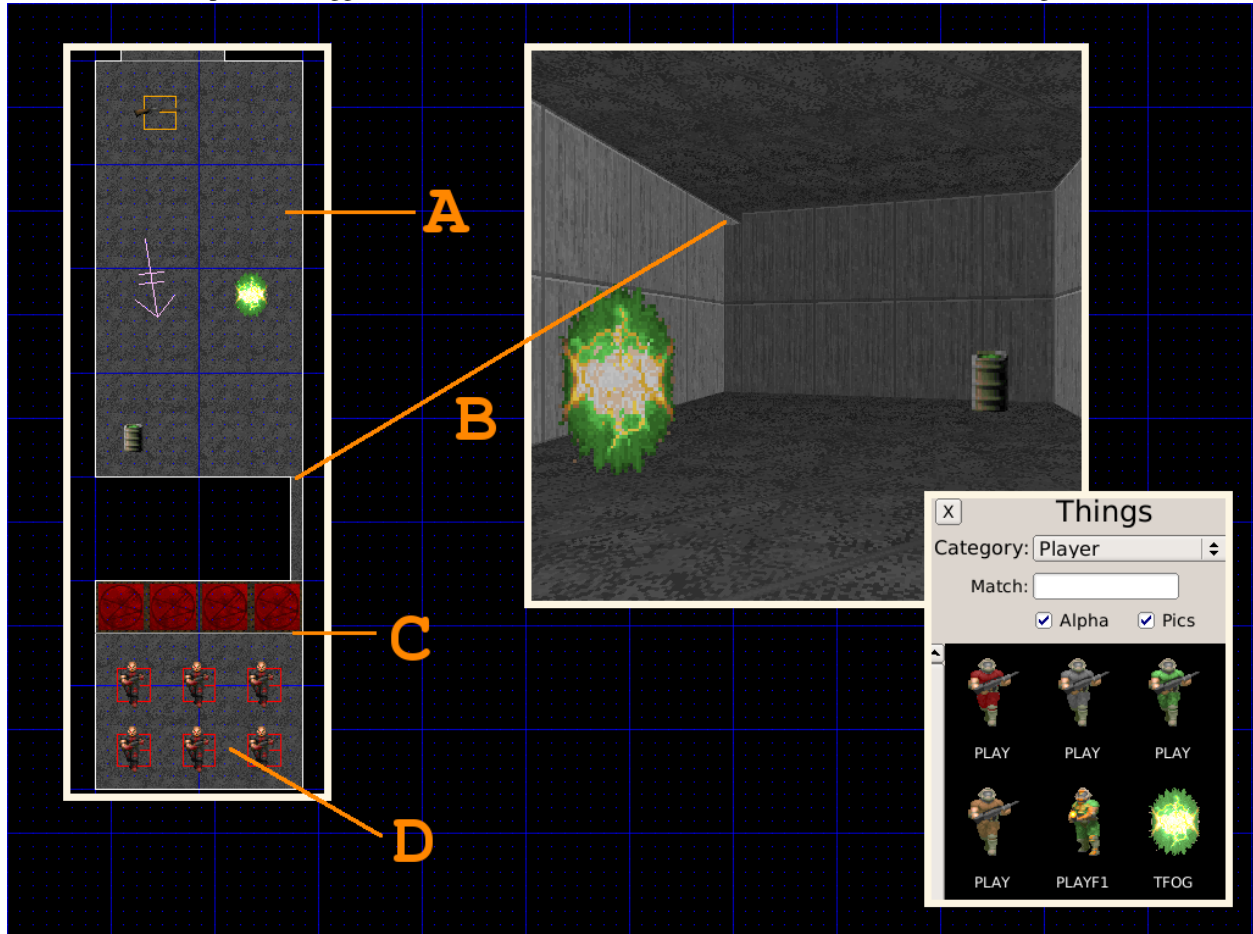
- Create the open door sector (A) and assign a fresh tag to it.
- Add the linedef that will trigger the door to close (B).
- Set the linedef **Type 16 W1 Close for 30s**



**Note:** If you want to close multiple doors in the room, give them all the same tag.

### 7.9.7 Teleport Ambush

Monsters stationed in a hidden room wake up to the sound of gunfire, walk over a teleport trigger and summon themselves to the designated location.

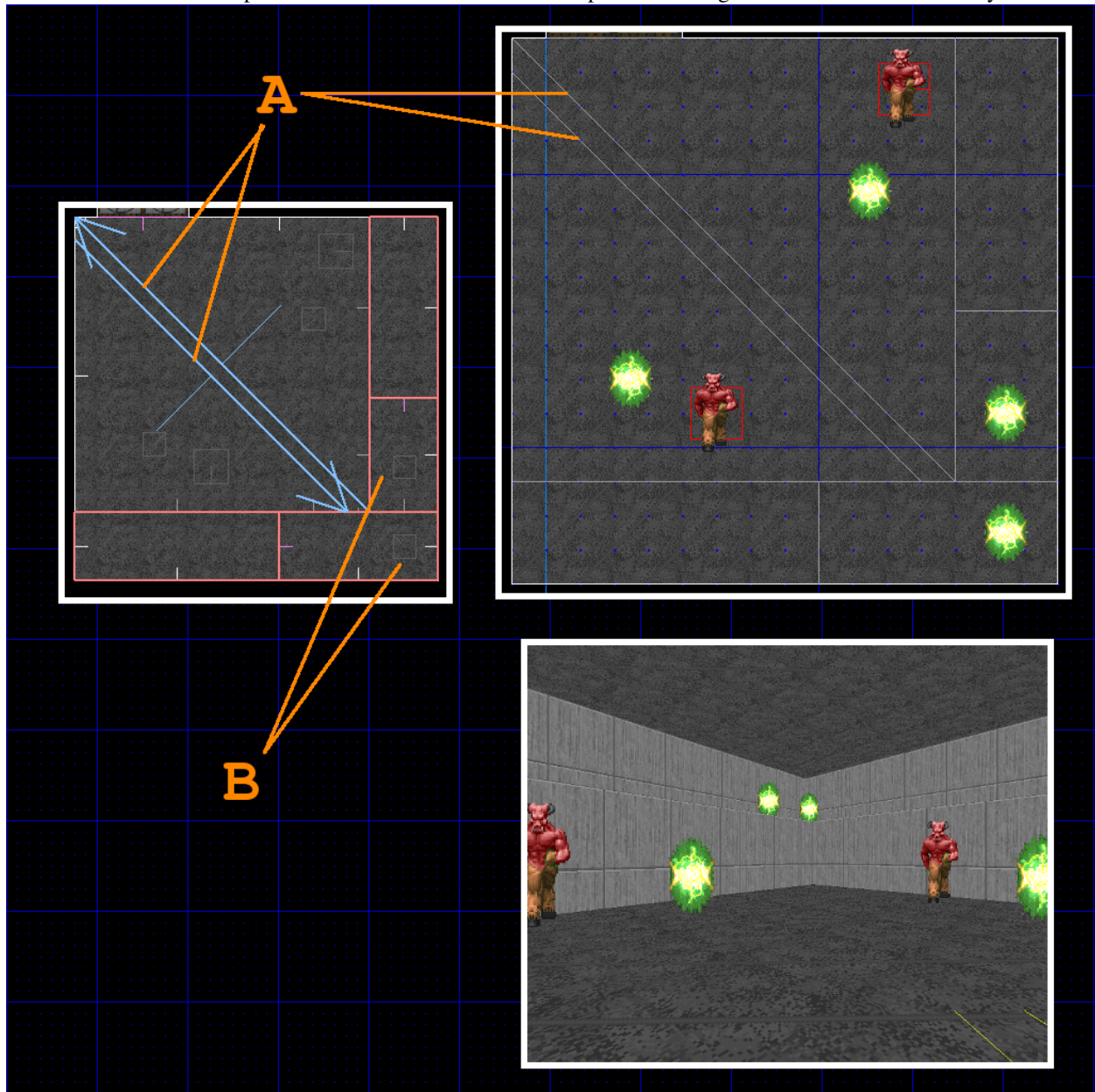


- Apply a fresh tag to the sector where the monsters will teleport into (A).
- Insert the **Teleport exit** thing (type 14, under the “Player” category called *TFOG*) in the sector (A).
- Create a hidden room where the monsters are stationed (D).
- Connect the main sector (A) with the hidden room (D) via a sound pipe (B). This allows the sound of gunfire to reach the hidden room, alerting the monsters.
- Insert a teleport trigger linedef in the hidden room (C), assign the same tag as the exit sector (A), and set the **Type 97 WR Teleport**.

**Note:** Use the **Sound Sector Rendering** mode (under the View menu), and enter **sector edit mode** in the 2D view to see how sound travels in your map.

### 7.9.8 Combat Teleporting

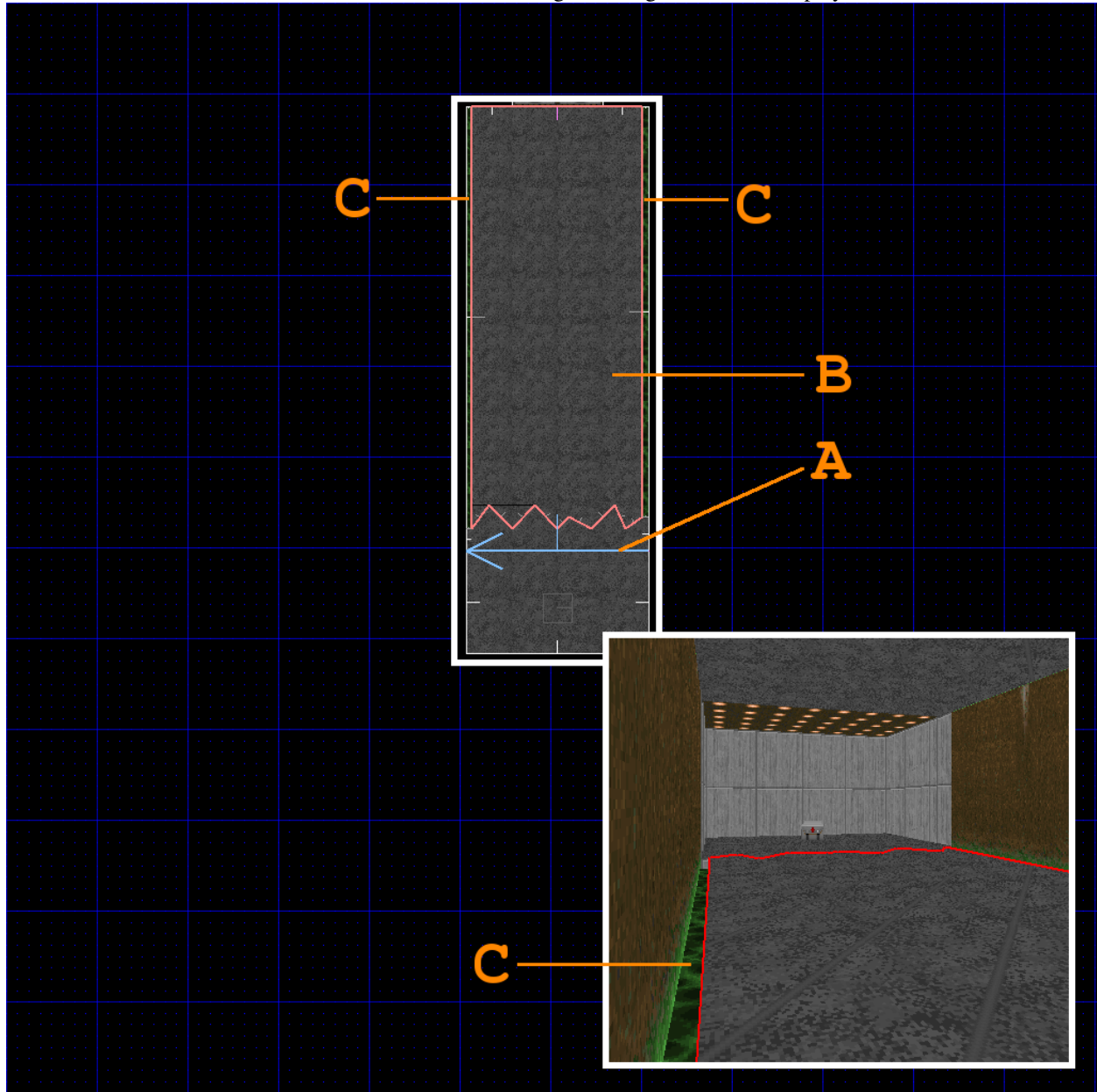
Monsters teleport around the player during combat, making engagements less predictable. This trap is used in Doom II map 10 to great effect with a Cyberdemon.



- Create teleport trigger linedefs of **Type 126 WR Teleport /mon** (A).
- Apply a fresh tag to the trigger line (A).
- Apply the same tag to a destination sector (B)
- Place a **Teleport exit** in the destination sector (B).
- Repeat as needed, more teleport triggers with different destinations will make for varied encounters.

### 7.9.9 Nukage Surprise

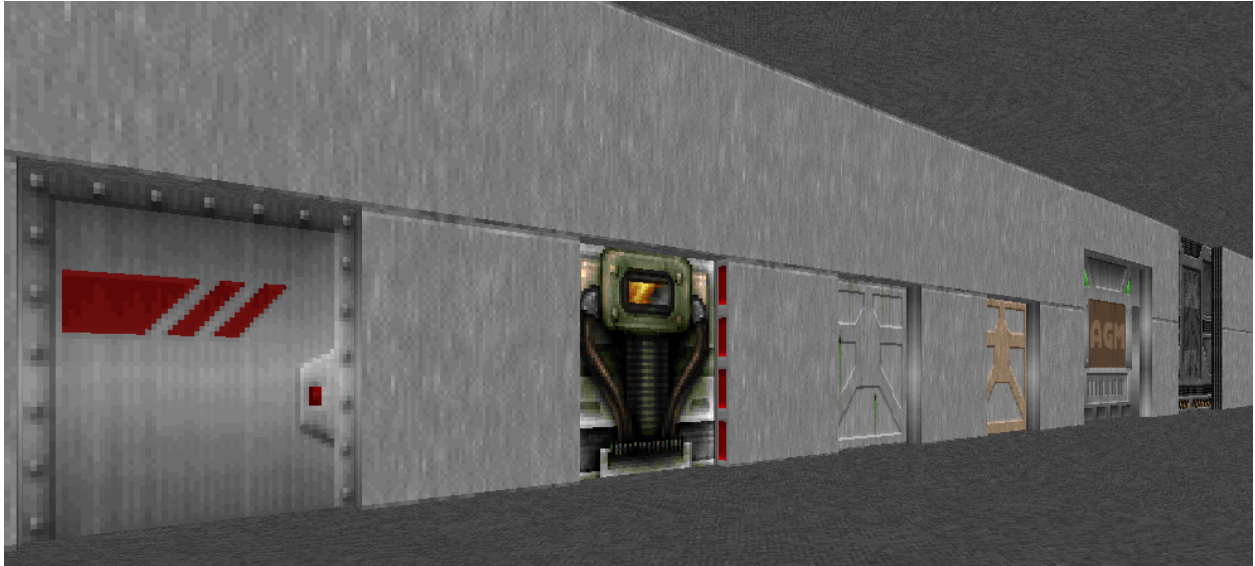
A floor turns into radioactive waste, dealing damage to the player when walked on.



- Add a trigger line of **Type 37 W1 Floor down LEF /NXP** (A).
- Apply a fresh tag to the trigger line (A).
- Tag the target sector (B) the same as the trigger tag.
- The target sector (B) must be surrounded by damage sectors (C).
- The damage sectors (C) are lower than the target sector (B).
- The damage sectors (C) have **Type 7 Damage 5%** and the **NUKAGE1** texture. On trigger the target sector (B) will lower to the damage sectors (C) and the damage and texture is transferred.

## 7.10 Prefabs

Prefabs are useful to quickly throw maps together by copy-pasting pre-designed structures.



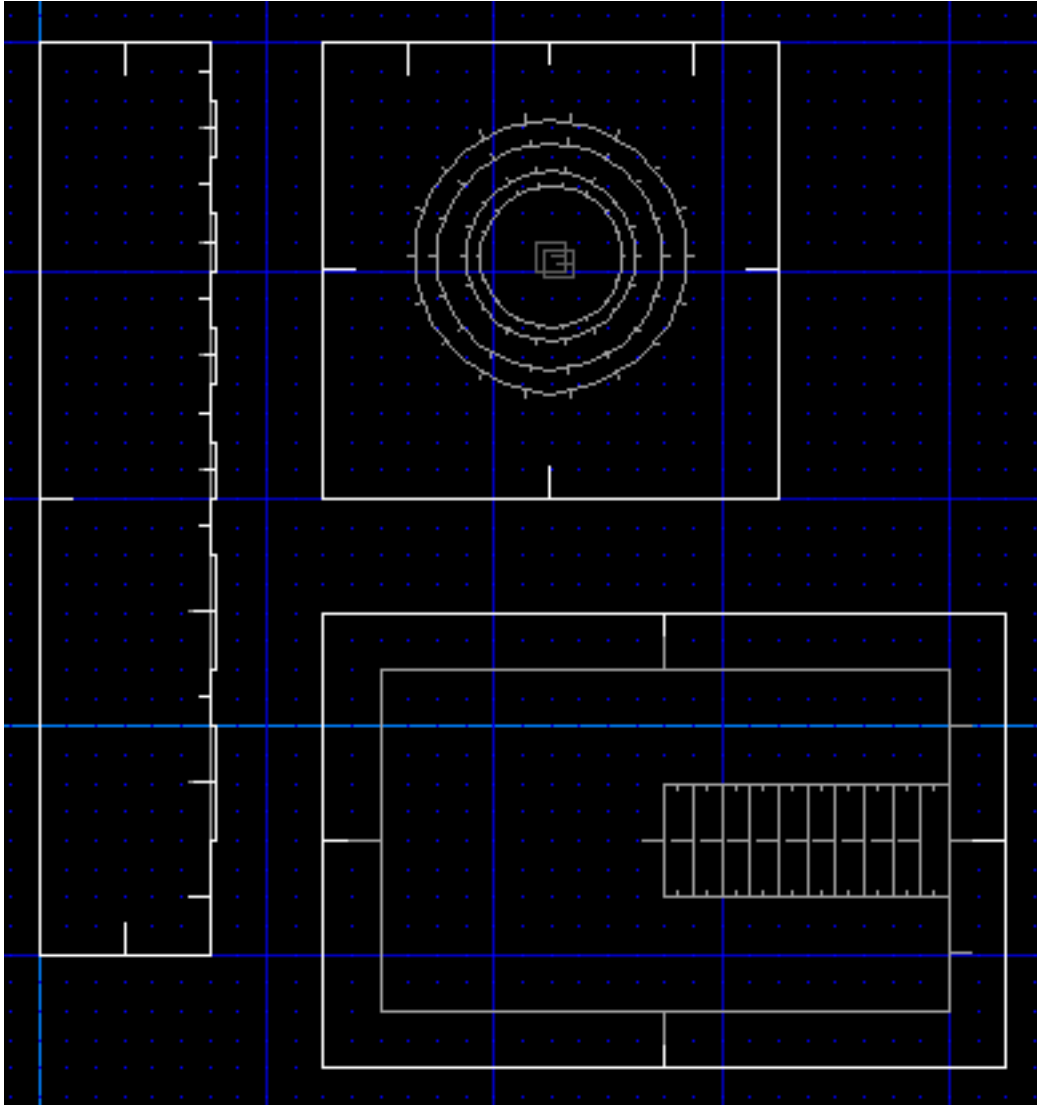
Things to keep in mind when using prefabs:

1. You can only copy-paste in the same Eureka instance.
2. You have to merge common vertices and sectors after pasting.

Exporting your prefab map to an unused slot in your game wad allows quicker switching.

### 7.10.1 Downloads

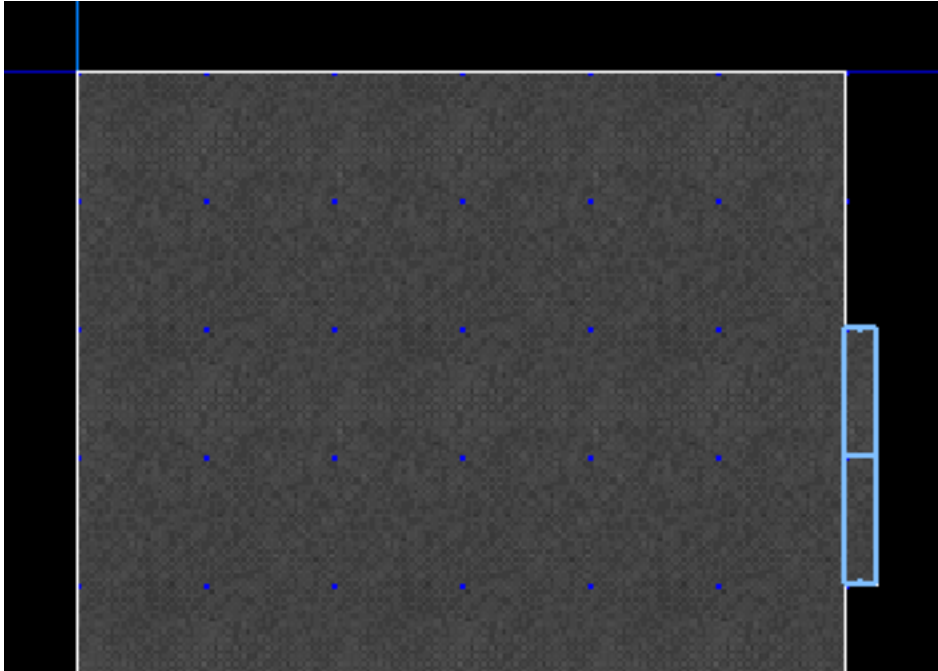
Use `prefabs.wad` as a base for adding your own designs.



### 7.10.2 Merging Vertices

Copying a prefab into void space, like a door prefab, needs vertex merging.

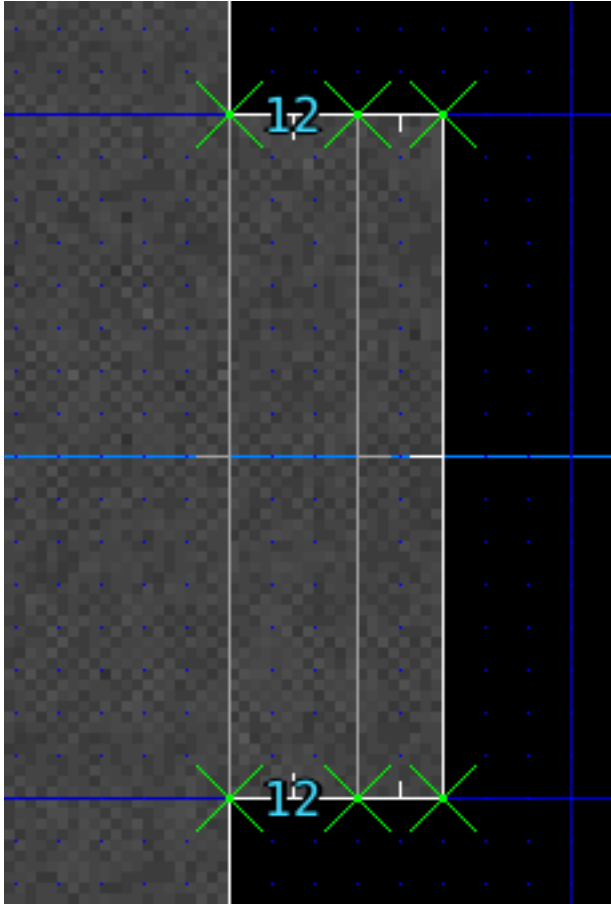
Select the door prefab in sector mode, press `control-c`:



Open the target map, position the mouse cursor where you want the prefab and paste with `control-v`:

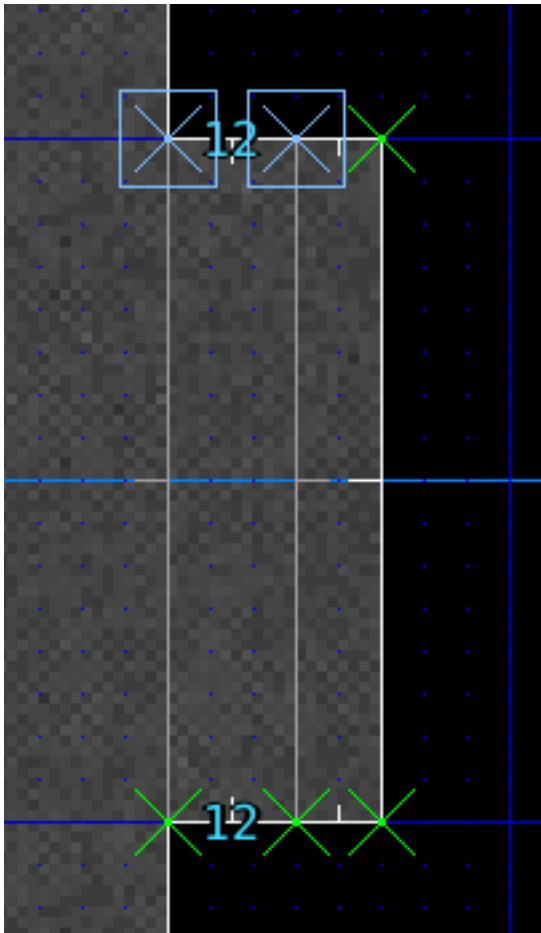


Enter vertex edit mode, use the RMB to link the room with the door vertices:

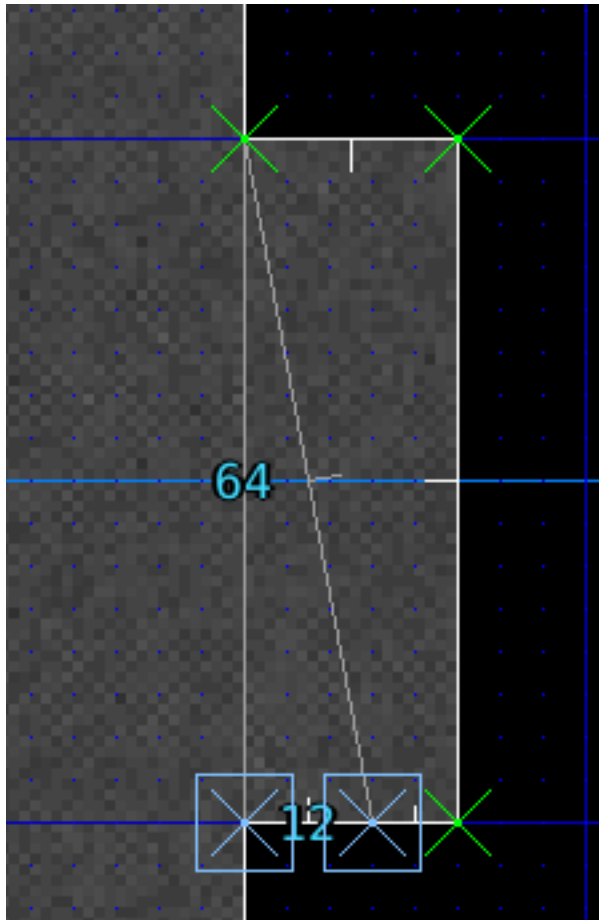


If you optionally want to get rid of the extra sector in between, select the one room vertex, then the matching door vertex (this order matters) and merge them:

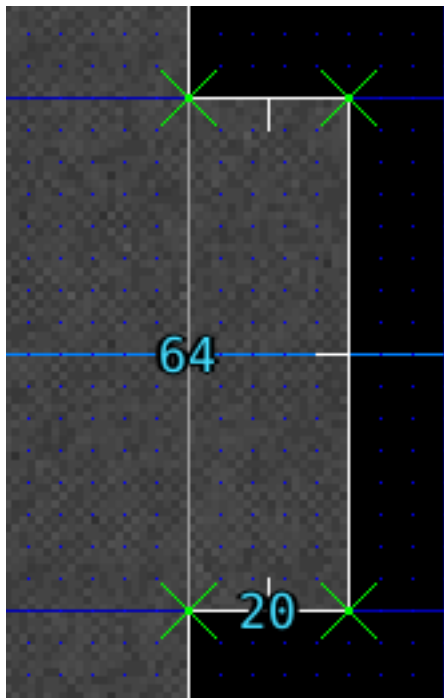




Do the same for the other vertices (selecting the room vertex first):



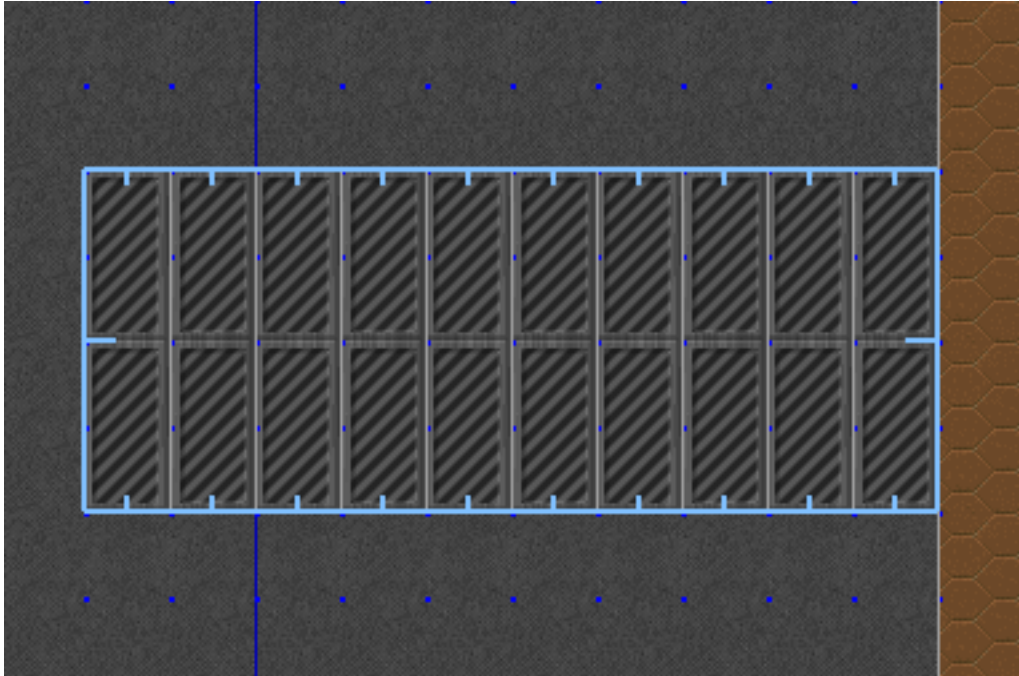
This completes your door:



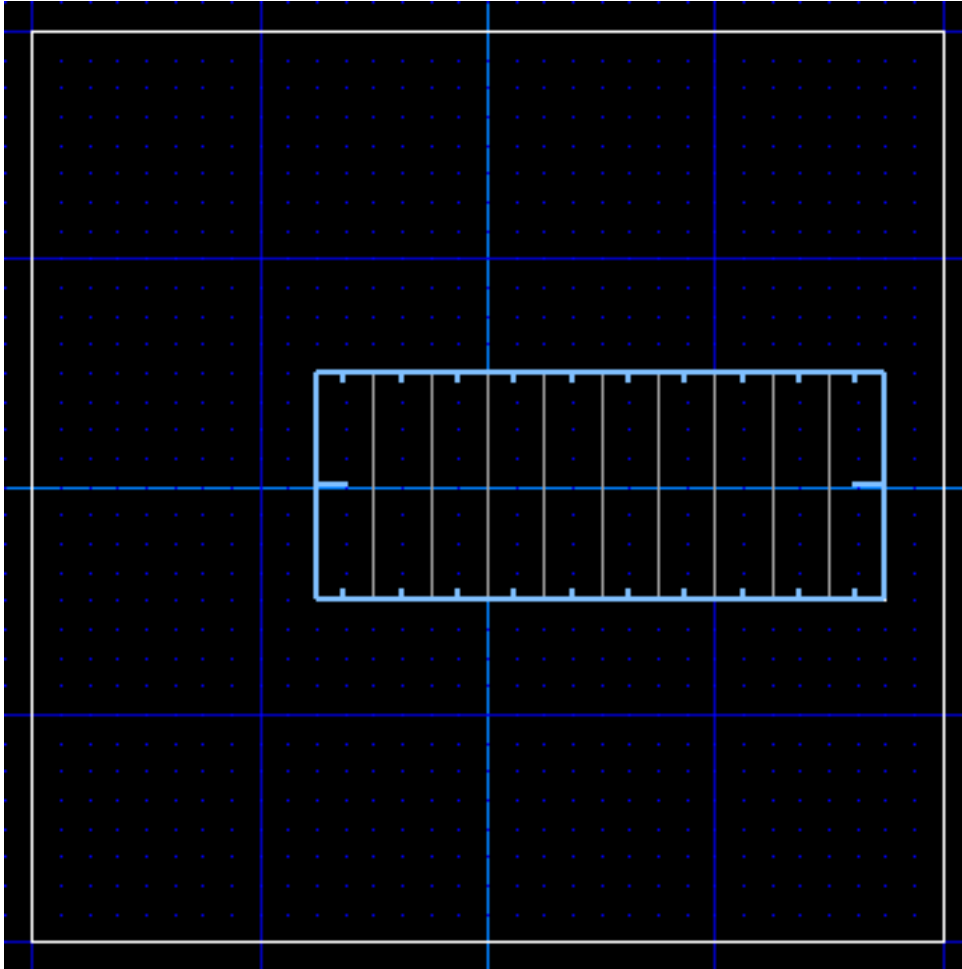
### 7.10.3 Merging Sectors

The process of pasting sectors into sectors requires another kind of merging. The stairs prefab is one such example.

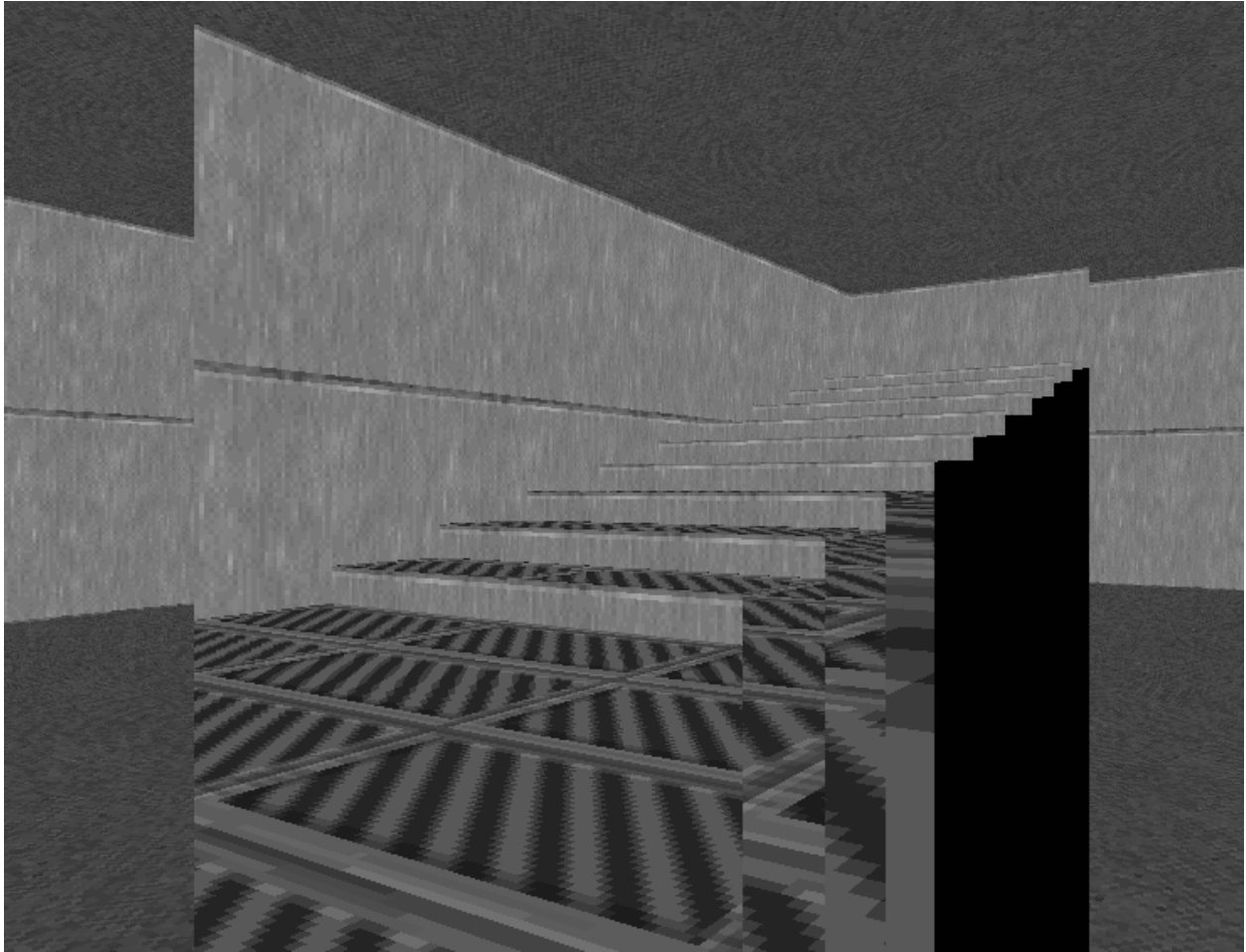
Select the prefab sectors. You might need to use `shift` `LMB` to avoid dragging sectors during selection:



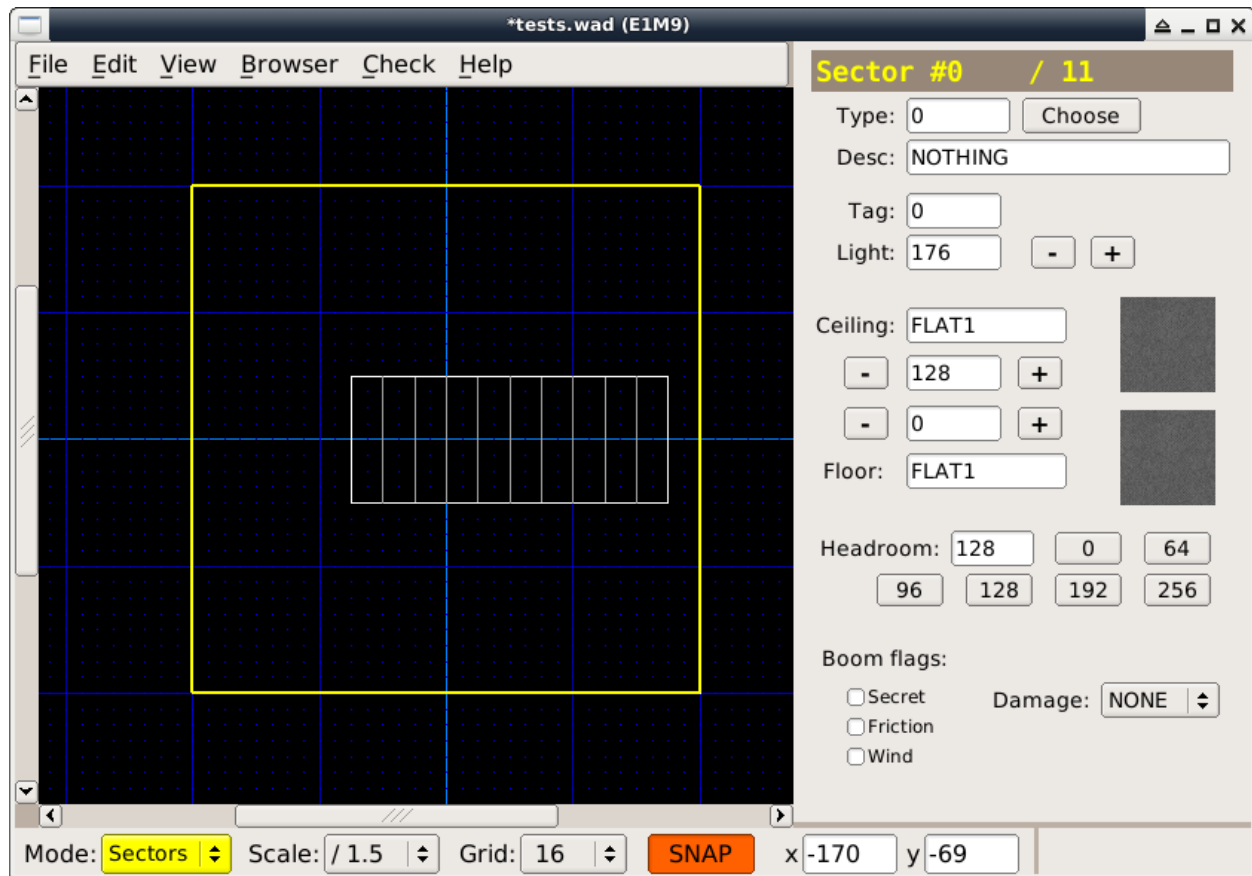
Copy, open your map, and paste. Ensure your mouse cursor is positioned where you want the prefab pasted:



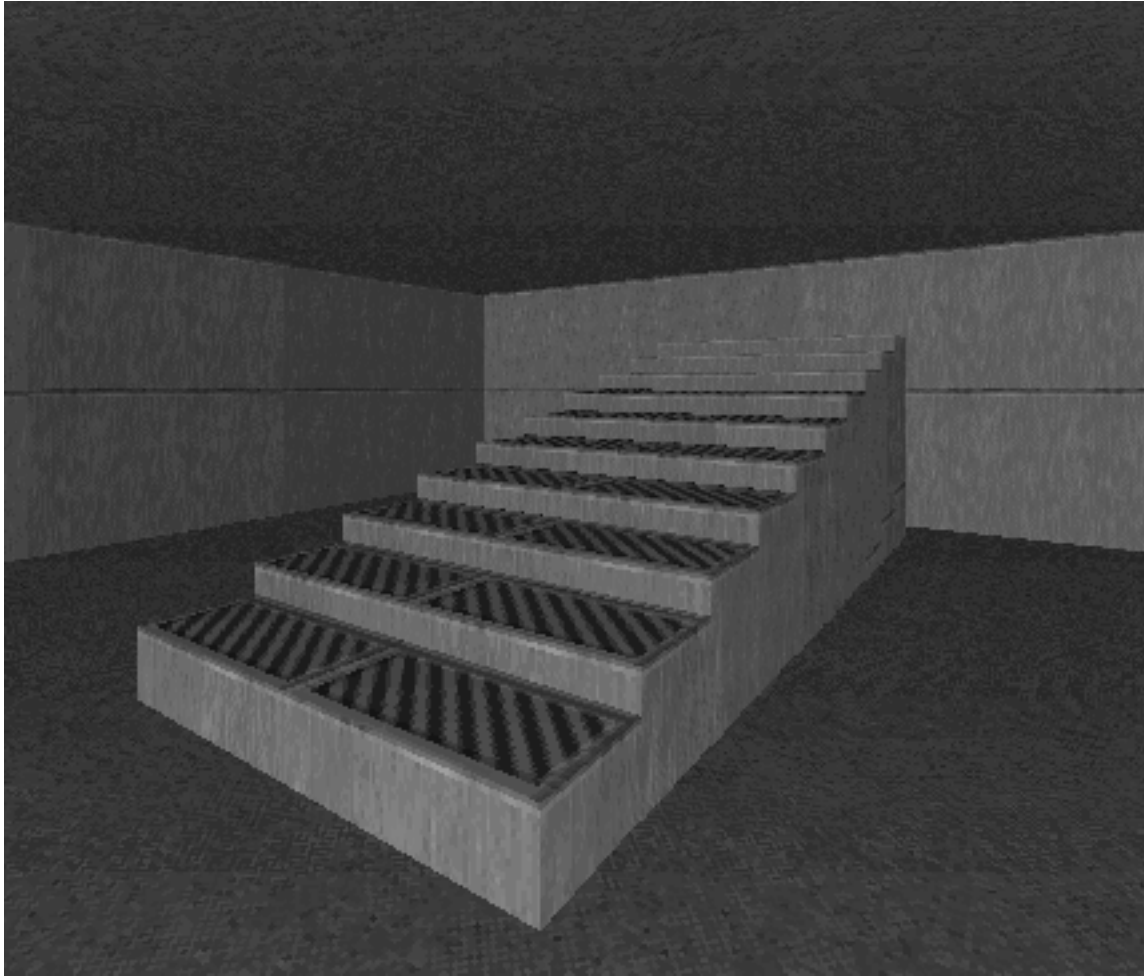
Switching to 3D view shows us artefacts, the prefab needs merging first:



To merge, deselect everything (`), hover to highlight the room sector and press `space`:



The stairs prefab is now merged with your room sector:







## CHAPTER 8

---

### Cheatsheet

---

The cheatsheet is a printable page of keyboard and mouse shortcuts for using Eureka.

- `cheatsheet.pdf`
- `cheatsheet.ods`