yesyesyesnoyesnonoyesyesyesDoomEd 2.60 Helpdoomed

Table of Contents

Help file produced by **HELLLP!**, a product of Guy Software, on 5/21/94.

Doom is an incredible 3D game from Id software. In it, you walk in real time through an entire world, where everyone is against you. Using weapons or fists, you create a gore-fest fit to throw weak stomached people in fits.
Doom is, of course, much more than that. It is a beautiful masterpiece of real-time programming. You can design your next house with it, and walk through before actually building. You can draw existing buildings, and use it for practicing finding your way around the real building... The uses are many and varied. Experiment!

Acknowledgements

This is a prominent location, so please scan these acknowledgements:

- Most doom data structures decoded by me, Geoff Allan, contact via `pringler@cuug.ab.ca` Take note that pringler is not my account, but a friends. Please dont leave him rude or obnoxious messages.
- Specific actions of line triggers and sector flash parameters by Matt Fell, along with a few of the **things**, contact via `matt.burnet@acebbs.com`
- Nodes, Ssectors, and Segs rebuilding from ID Software. Thank you very much, ID, for making this available to us third party developers! (DoomBSP.Zip - April 6, 1994) DEU 5.0 nodes builder is in this BETA version... thanks to Raphael Quinet
- Beta testers (and registered users...) Dave Nixon, Dan Mitenko, everyone on Alt.Games.Doom, who tested and gave me some great ideas.
- ID Software, for designing and releasing Doom in the first place.

Editing Things

To **edit things**, for example, to turn that stupid blue bottle into a pink demon, do this:
Get a map displayed. Check that the far right word on the control bar is Things. If not, click on the Thg button. Click on the little squares, they are the things. The currently selected thing will change color to indicate that it is selected. Modify the information in the dialog box. If you select any of the other buttons on the control bar, you will need to click Thg to return to editing things.

You can drag the location of any thing to a new location. Note that it will snap to position on a grid. This grid can be changed with the Set Grid... selection under Maps.

To **add things**, click the right mouse button. You can adjust the position of the new thing before you release the mouse button. The new thing will be a copy of whatever the last thing was that you viewed or edited. If you havent yet viewed or edited anything, then it will be a barrel, visible on all levels.

To **delete things**, select a thing and click on the Delete button in the things dialog.

Editing Walls

To **move walls**, click the Vtx button on the control bar. Vertexes will be shown at all line intersections. Using the mouse, move a vertex around to its new position.

To **split a wall** in two, which allows stretching it out, click the right mouse button on the line. It will be magically split in two. You can move the position of the new vertex before releasing the mouse button. See Building and Editing a Map

To **delete a line**, select one of the ends, and move it on top of the other end. The line will be deleted.

To **change walls**, for example, to change the startup room into a nice green marble:
Choose a map to work with. Click on the Lin button. This puts you in LineDef editing mode. Select any line on the map by clicking with the mouse. The Line Definitions dialog box will appear.

Changing the textures involves expanding the dialog by clicking on the **SideDefs >>** button. While the dialog is expanded, you can shrink it again by clicking the **SideDefs <<** button. The textures and their placement are shown here. For fine tuning the location of a texture on the wall, use the texture offset values. Note that a positive Y value raises the texture, a positive X value moves it to the left.

If you want to preview the wall types that you will be using, select RESOURCES, VIEWER, click on TEXTURES, and click on a texture name. These take a second or two to come up. (Dont blame me, Doom always does this at startup and you know how long THAT takes...). This can be done while the texture dialog is showing, since the texture dialog is modeless.

Editing Sectors

To **change sectors**, which includes height of floor and ceiling, along with the floor and ceiling textures: Choose a map to display, Click the Sec button. Now click inside any sector to have its information appear. You may edit this information at will.

To **delete a sector**, and all of the associated data: click the Delete button on the sectors dialog box. DoomEd will correctly detach all connected lines and keep everything else stable.

To **insert a sector**, or group of sectors, click the Sec button. Place the mouse cursor in the center location where you wish the new sector to be. Click the right mouse button, and a dialog box will pop up for adding a sector. If the mouse is currently inside another sector, the dialog settings will reflect this. You will then be allowed to select whether the floor and ceiling are higher, lower, or flush with what is currently there. To change the floor and ceiling heights of the sector **before** you place it, click the Style... button. This will also allow you to choose either a predefined sector style, or select the new floor, ceiling, and wall textures.

Viewing Graphics

To **View Graphics**, select Viewer from the Resource menu.
Bitmaps (also called Sprites, Panels, Tiles, and Textures) usually have a set of X,Y co-ordinates that specify the lock point. You can toggle the bitmap's use of this lock point with the Lock button. Some graphics, such as the weapons, are way off of the display if lock is on.

The four types of bitmaps are:

Sprites: objects, things, people, plants, etc.
Panels: wall panels, doors, stuff like that.
Tiles: floor and ceiling tiles.
Textures are panels connected together and attached to walls.

As an example of textures, the POISON sign is a panel, and the wall behind it is two panels. The texture is the connection of these three pieces into a single wall mapping bitmap. The computer displays are actually 6 different panels, pasted onto a background. This texture appears in the second major room in E1M1 (look up).

Play with the Viewer to see all of the graphics available.

Graphics editing abilities will be in version 2.8 of this program.

Editing Sounds

To work with **Sounds**, select Sounds from the Resource menu. This will allow you to play sounds, extract the sounds to a .WAV file, or put your own sounds into Doom. There is no limit to the replacement, DoomEd will rebuild the Wad file to allow larger .WAV files to be inserted. Any sounds that you REPLACE with must be **11025Hz, 8bit, mono**. Anything else will just give you silence.

You can also extract WAV files from DOOM and use them as your Windows sounds. From control panel, select Sounds. The rest should be obvious.

There is currently no way of extracting or playing the Music (songs), since I still have not found the format for the MUS data. As soon as I have this information, I will include this ability. However, there is a file available called MIDI2MUS.EXE, which will convert most midi files into the MUS format which Doom uses. After processing a Midi file this way, it will then be possible to import it into a miniwad file. Automating this is planned for the release version of 2.60

Registration

1) **Why Register?** Simple, really. This program has consumed the better part of several months, during which time I have had little or no income. If I get enough money from users, then I can afford to spend even more time improving and adding functions. If not, then I might be forced to stop working on this project.

2) **What do I get?** If you register, you will get the next release version when it is done. Optionally, you can order the source code to make your own modifications and additions (for your own use), as well as understanding the data structure used in Doom. You also get a warm feeling inside.

3) **I never register Shareware.** Well, you should. Personally, I rarely ever did until I got into this project. Now that I understand just how much work goes into a large project like this, I am running around registering the shareware that I use. Really!

Register this version AND subsequent versions with a ONE TIME fee of **$15**, or **$25** for source code. In the UK, send **£8**, or **£15** for source (the mails are expensive). I sincerely hope that you find the price that I am asking to be reasonable. (This is a blatant **plea** for money). Just mail a check or international money order to:

> Geoff Allan
> 7232 Kananaskis Drive SW
> Calgary, Alberta, CANADA
> T2V 2N2

Be sure you include your full name and mailing address, and any comments would be welcome. If you are on the Internet, I would prefer to e-mail you the data, so make sure you include your e-mail address. And... thank you in advance.

DoomEd -
The Real Thing

Map Builder and Editor

Version 2.60 beta 4

Welcome to DoomEd - The Real Thing. This help file will take you on a tour of the basic steps needed to create a great Doom map level. Several new features make their debut in this version.

Completely re-written since last beta version:

- Selection of Things, Vertexes, and Lines
- All select/deselect code
- All mouse handling routines
- Dialog update routines
- Platform editing

New in this beta version:

- CTL3DV2 for 3d Dialog boxes
- Setup Program (very simple)
- Pictures of Things in Things dialog
- Ability to DELETE objects
- More solidity, better code
- User Interface improvements
- Keyboard Zoom support (using Alt-Z, Alt-X)
- ID Software Reject builder (still problems)

Code written but not in this beta version:
(ie. Will be in release version Real Soon Now)

- Sector texture changing and stitching
- Ability to drag sectors and lines with mouse
- Ability to design your own sector Styles
- ID Software nodes builder (not DEUs)

This editor was written by Geoff Allan. I will not list all of the users who have sent suggestions and comments, suffice to say that I thank you all. Almost all of your suggestions have been incorporated in this version. Some of the more difficult things are left for the next version. I truly desire that this editor be the BEST Doom editor / map builder that is available.

Geoff Allan is in no way associated with ID Software. You will receive NO support from ID for this program or its associated data files (you can get support from Geoff Allan - see Registration). If you create a Doom map using this editor, you MUST NOT create a map which can be used with the Shareware version of Doom. You may distribute maps created from this editor, but you may not charge money for them without consulting ID Software. Please read the license.txt document which ships with Doom for more information about your rights and restrictions. The name Doom, and the Doom logo, are the property of ID Software.

Jump to Contents

3-d Wireframe Display

The 3-d wireframe display is an extremely useful tool for previewing your map. With it, you can zoom and pan in real-time, showing the level in full perspective view. Although there is no hidden-line removal or texture mapping, after a little experimenting I am sure you will agree that it is very helpful as it is.

To get true real-time display, you will probably need a 486-DX/2 @ 66mhz, or better, and that is the development machine which I wrote it on. But realistically, if you have a machine capable of running Doom at a good speed, then you are likely OK...

When you select 3-D Display from the Map menu, you will be presented with a dialog box for selecting the viewer location and the look-toward point. By default, the look-toward is the exact center of the map, and you can likely just ignore this dialog.

While displaying the map, use the mouse to change the view as follows:

1) Hold **both** mouse buttons down. Then, move left to zoom out, move right to zoom in.

2) Hold the **left** mouse button down. Move left and right to rotate around the look-toward point, or up and down to change your altitude.

3) Hold the **right** mouse button down. Move left and right to rotate the look-toward point around you, move up and down to look higher or lower.

You can maximize the display to full screen, although smaller windows will display faster.

How Doom Maps Work

At least an elementary understanding of how Doom stores map information will help you get the most out of any editor. The full map structure is quite complex, but here is a summary:

1) **Vertexes** (or vertices). These are x,y coordinate points on a 2-dimensional plane. Remember your high-school math courses? Well, welcome back to that nightmare... Vertexes (yes, I know they are called Vertices, but the map stores them as Vertexes, so that is what I call them) are at each end of every line.

2) **LineDefs**. These are the line definitions. Each linedef has two numbers representing the vertex numbers for the start and end points. Linedefs also have other information in them: Does crossing this line trigger some action? If it is a wall, does operating the line flip a switch or cause some other action to occur? Each linedef includes information on whether YOU can cross it, whether MONSTERS can cross, whether SOUND can cross, etc. To see all of these choices, select a line using the Lin tool and look at the dialog box.

3) **SideDefs**. These are connected to Linedefs, and tell Doom which textures to paint on the wall. Textures, of course, are the pictures or bitmaps. Sidedefs also are what define the sector number that a line is part of.

4) **Sectors**. Every enclosed area is a sector. That means each step in a staircase is one sector, each giant room, each window frame, door, little room, etc. Sector information includes the brightness, the floor and ceiling heights, the floor and ceiling textures, and a platform number. By giving a sector a platform number (say, 4), you can cause it to perform some action (like moving up and down, crushing you, becoming totally dark, or opening like a door) when a linedef is crossed or triggered. Load in a map where you know these actions occur, and select the linedef that triggers some such action. Check the platform number, then look at the affected sector. This is the key to the dynamic nature of the Doom world.

5) **Things**. Every item, bad guy, tree, blue bottle, ammo pack, weapon, start position, and teleport destination is a thing. The things data includes the x,y coordinates, the type of thing, what levels it appears on, what direction it is facing (if alive), and whether or not the guy has to actually see you before attacking (deaf).

6) **Nodes, SSectors, Segs**. You have probably heard a lot about these things, but have no idea what they do... well, until recently that was a mystery. The Doom display engine needs this information to work fast. Basically, it is the pre-calculation of a lot of the 3-d display stuff, and since the editor creates it from your work, you shouldnt need to understand it.

7) **Blockmap**. Doom divides the map into small blocks, and then for each block there is a blockmap entry defining which walls are near. This is to speed up the wall-collision detection, since Doom doesnt have to check every wall in the game, just the ones you are near.

8) **Reject**. This is a table showing Doom the connections between sectors. Using reject information allows Doom to speed up the monster activation and animation routines by excluding the sectors which cannot be accessed anyhow.

What Next?

When I started this project, I was a professional Visual Basic programmer. This looked like a good incentive to learn C programming to the Windows API, and at first, doing even simple tasks in C required several hours with the Windows programming manuals. Now, I can add new features quicker and with fewer problems (3 months intensive training will do that...). The 3-d wireframe display took about an hour, plus another hour for debugging. So, I guess that makes me an (unemployed) professional Visual C++ programmer.

There are a few enhancements which I am currently working on. These will be released in subsequent versions, and will significantly simplify the map-building process:

For 2.60 release (or 2.61):

Drop DEU nodes builder, use Ids
Move sectors, group vertexes and move them
Group changes of sectors and linedefs
Automatic change of textures for an entire sector
User defined sector styles

For 2.70 release:

Graphic selection of textures in dialogs
Predefined room shapes and quick placement
Grow-Shrink of sectors or entire map
more...

For 2.80 release:

Editing of graphics images
Export / Import of graphics
Copy, paste, Undo
Recording of new sounds from DoomEd
Internal Midi import / export
more...

For 3.00 release:

Complete re-write using MFC in C++
OLE 2.0 support (stick a map in a WinWord document!)
Floating toolbar, Corel-style interface
Total stability and crash immune
Win32 model for Windows NT and Chicago (Windows 4.0?)
More features, faster, more flexible...

The Grid

The Grid as displayed is not the grid that the mouse snaps to. Instead, it is the 64x64 grid where the floor and ceiling tiles are joined. This will help, for example, when creating teleport pads, where you want it to be exactly one tile square.