GDR Doom Editor V1.0

Written by Ivan Hawkes and Scott Little

Table of Contents

Disclaimer

Golden Dawn Research Pty Ltd. takes no responsibility for the use of this software. We do not guarentee it will perform any task, stated or otherwise. We take no responsibility for any damages which use of this program may cause. Any resemblance to real people either alive in or virtual reality is purely of no interest.

Copyright

Do not alter the program title and about box and then release it as a new program. We are being VERY fair with you by releasing the source code, please show some respect. We would appreciate if any alterations, improvements or additions to the code are sent back to us at slittle@enterprise.powerup.com.au or you can mail me on compuserve as user 100240,1765. I don't check my CompuServe account too often,so don't be upset it it takes a little while to get a reply.

Credits

We would like to acknowledge and thank the following people for their contribution to this product. Without their efforts and the desire to share their hard won knowledge with us there would be no DOOM editors. Imagine that!

Matt Fell — Author of the Unofficial DOOM Specs .
Matt has paved the way for all by publishing the specs for the DOOM.WAD file. His file was absolutely vital in creating this editor. It's also quite handy for those of you out to create a new level. The file is called DOOMSPEC.TXT, you should be able to find it on any good BBS.

Colin Reed — Author of BSP.
You will need BSP to build the nodes, ssectors and other obscure elements of a DOOM level. Colin has kindly published the source code for BSP. We have included a compiled version of the program to get you started. Those interested in the full version should

download BSP11X.ZIP from a BBS.

Raphael Quinët
Brendon Wyber Authors of DEU.
For being an inspiration to all and providing extra documentation for creating the editor. DEU was proof to us that there wad enough info out there to build an editor of our own.

Matt Tagliaferri Author of DoomCad.
You did a good job on the editor but unfortunately were let down by the speed of Visual Basic.If it weren't for that we may still be using yours today. Lot's of good design work went in to this product.

Introduction

Doom, great as it is, cannot provide the ardent DOOM'er with the challenge needed to keep playing. You are either lucky and own a network (we do, yea!) so you can slug it out with your friends (who really are the worst monsters on the level!), give up and say it's boring (heretics!), start to download levels from a BBS or create your own.

As fun as it is to play other peoples levels there's no substitute for playing a level you have designed. You know exactly what you want from a level so can provide it in abundance.

A while ago we started looking for a DOOM editor. Sadly the first editors were kludgy, error ridden or bogged down with arcahic interfaces. One of the biggest obstacles of all to face however is HOW to build a DOOM level and how the hell everything works. Many of the editors took the approach of "look at the DOOM.WAD file and you will be an instant master", so it wasn't until we got hold of DOOMSPEC.TXT that everything fell into place.

This editor has been built to create levels from scratch or to modify current levels. It can handle reading levels from the main DOOM.WAD (but for gods sake don't make the mistake of trying to save to this wad file!) as well as PWAD files you find on BBS's. The changes made can then be saved to a PWAD file for play.

We haven't written the code needed to rebuild all the nodes, blockmaps, etc as yet so after building a level you will have to run BSP on the file to make a level you can play. This stage can be included in a batch file anyway so it shouldn't provide you with any dramas.

Installation

Couldn't get any easier. Just copy the executable files into a directory on your hard drive and run from there. The program uses BWCC to customise the dialog boxes. You may want to place the BWCC.DLL into you WINDOWS\SYSTEM directory to avoid having several copies of it.

There is an INI file which the program uses for settings. Keep this in the same directory as the executable! You may need to change the location of your DOOM directory if it is not in the usual place. You can set the gridsnap size here too. We will place this in a dialog box when we get a chance.

Building a Simple Doom Level

A doom level is made of several elements. The levels are basically laid out in two dimensions although they give the appearance of three dimensions when viewed from the game itself. So, although DOOM is a three dimensional game the layout of a map will be very much like the design of a floor plan for an office or house. Let's create a very simple level.

STEP 1:
Start up the program.

STEP 2:
Click the right mouse button in the middle of the screen. This will create a new object on the map. By default the object will be a player start position. This is where the first player in a multi-player co-operative game will appear. It is also the start position for single player games.

STEP 3:
Using the menu switch to the vertex view. Now click the right mouse button four times and form a square shape around the spot where you placed the player 1 start position. These are the joining points for lines.

STEP 4:
Change to the linedef view. Click the little 'plus' button on the button bar to begin adding a new set of linedefs. Now click the left mouse button on or near a vertex. Click the left mouse button four more times, moving around the vertexes you created earlier in a clockwise fashion. As you click red lines should appear joining the vertexes you selected. Once you have joined all four vertexes press the right mouse button to stop adding linedefs. NOTE: The little right angle tags on each line should face towards the centre of the square. If they don't you have added them incorrectly and should start again.

STEP 5:
Switch to the sector view and press the 'plus' button on the button bar to add a sector. Click on each of the vertexes in a clockwise direction to add the linedefs you just created to the sector.

STEP 6:
Save your level by choosing the 'save' command from the 'file' menu. Your level will be saved under the name "NEW.WAD" in the same directory as you have placed the software.

STEP 7:
Exit to DOS and run BSP on the WAD file. Type in something like BSP NEW.WAD TEMP.WAD.

STEP 8:

When BSP is finished processing the file you can run it by typing something like:
DOOM -DEVPARM -FILE TEMP.WAD

You should be in a very dull room with four walls. Not particularly exciting but a good start. Give yourself the plasma gun by typing IDKFA and reward yourself by going berserk with it in a small room. It's what we always do!

The Elements of A Doom Level

A doom level is made up of five main elements:

Things	**The guns, monsters and decorations on a level.**
Vertexes	**The points where lines are going to join.**
Linedefs	**The lines on the map and their properties.**
Sidedefs	**The textures mapped onto the lines.**
Sectors	**Areas on a map where you may travel.**

The other parts of a .WAD file are automatically generated for you by BSP so you need not be concerned about them.

Things:

These are the objects which you find in the game. They encompass monsters, guns, ammunition, decorative skulls, etc. You may add new things to the map by clicking the right mouse button on the screen at a spot clear of things. The new thing added will be an exact copy of the last thing you were editing. If you right mouse click on an existing thing you will delete it.

Things may be moved by clicking on them with the left mouse button and dragging it across the map to a new location then releasing the mouse button.

For each thing in the system you can set six different values. The *level controls* determine which skill levels the object will appear on. If the box is checked the object will appear on that skill level. These correspond to "Hurt me Plenty", "Ultra Violence", "Nightmare", etc.

Deaf Guard only applies to monsters. If this box is checked then the monster will stand there stupidly while you sneak up behind them and blow them away. They will only attack when they see you.

If *Network Game* is checked then the object will only appear in a network version of the game. It will apear whether you play co-op or deathmatch.

The direction of an object may be set using the *direction control* buttons. The direction is only useful for monsters and players. All other objects face you no matter which angle you look at them from.

Vertexes

These are used to join linedefs together. You can add new vertexes by clicking with the

right mouse button on a spot free of vertexes. They can be deleted by right mouse clicking on a vertex. Apart from that you can only move them around the playfield. Movement is the same as it is for things.

The use of vertexes ensures that lines always join up with each other. This stops the posibility of gaps between the polygons.

When you delete a vertex you automatically delete all the linedefs and sidedefs using this vertex. Don't forget to update your sectors when you delete linedefs that form part of a sector.

Linedefs

Linedefs are the walls you see (and the ones you don't too) in the game. Each linedef has at least one sidedef attached to it, with two sided linedefs having two sidedefs. The linedef has nine switches which control its behaviour:

Lines have a 'from' vertex and a 'to' vertex. This gives the lines direction which in turn gives the line a right and a left side. The right hand side of a line is highlighted with a little 'tag' to remind you. The tag points to the right normal for that line.

For a line to have two sidedefs you must add two sectors to the map which use the linedef. One sector will use the right hand side of the line while the other will use the left hand side. Since each linedef must have at least one sidedef we create a default sidedef when making the linedef. This sidedef is ALWAYS the right side of the line. When you add a sector which uses the left side of the line a left sidedef will be created. You should create all linedefs for a room, add the sectors you want then play with the sidedefs.

IMPORTANT: A two sided linedef must front onto two sectors, even if they are the same sector! If you don't do this you will get the HOM (Hall of Mirrors) effect or DOOM will just refuse to run.

Wall above is unpegged:
When the sector is moved up or down this will keep the texture mapped on the sector stationary.

Wall below is unpegged:
Use this on the linedefs you map in the side of doorways to stop the door track from rolling up and down when you open the door. Check out the effect to see what I mean but removing this checkmark from an existing doorway.

Monsters cannot cross:
Monsters will not be able to cross over this line. These can be used to trap monsters in rooms, keep them from scrambling all over the sludge areas or toppling into pits and crevices. It will almost surely be used to *chain* a battalion of cyberlords to the centre of

big open areas for the fun of watching people sprint past them. Ha ha ha.Hardly origninal, but still a lot of fun!

Mapped at start of game:
This line will show up on the map at the beginning of the game, even if a player hasn't seen it yet.

Soundproof Wall:
Use this to stop noise from adjacent sectors crossing this linedef. Sound will not pass through an impassable wall in any case so you only need to use this on two sided linedefs.

Impassable:
Neither the players, monsters, bullets or sound is able to cross a linedef marked as impassable. This rule about bullets doesn't apply to *two sided* and *impassable* walls. You can set both values on (despite what has been said about this in several DOOM files) to create cages like on level E1M9 or our demo level.

Two Sided:
If a line can be crossed then you MUST select two sided. When you are using two sided lines ensure there is a sector on both of the sidedefs, and set the sidedef textures to transparent where needed. Improper settings on these lines is the most common cause of HOM.

Be sure to think carefully about what parts of a sidedef are going to be visible. Set textures only on the visible parts and clear the other textures to avoid (HOM) and overprocessing.

Secret Wall:
These are just walls that don't appear on the automap like other walls do. You do not get secret credit when you find these walls, that is attached to a sector. You need to entice the player into a sector with secret credit to get the benefit of that in the final tally.

Unmappable:
These are walls that just will not ever appear on the map. This is great when you are creating light sourcing or using the walls as triggers. Also good if you just want to change sector values and don't want to clutter up the map.

LineType:
This defines what type of line you are using. Lines may perform special actions to their own sectors or to the sector which has the same trigger number. Each linetype has a two character preface. These charactes have the following meaning:

W: You have to walk over this linedef to activate it.
S: You must press this linedef to activate it.
D: This linedef is part of a door.

1: The action may be performed once only.
R: The action may be repeated.
-: Not defined.

For instance, to create a door, place a sector on the map and give two of it's sides the "(DR) Open, 5 seconds" line type. Make sure the side linedefs are unpegged. Set the ceiling height of the sector to the floor height and you're pretty well done.

Some linetypes need a trigger value to be supplied. You give the same trigger number to the lindef in question and all the sectors which it will affect. This ties the two elements together.

Editing Linedefs:

To select a linedef click close to the centre of the linedef. The program will select the linedef with the centre closest to your mouse click. You can edit the linedef by double clicking on it. This will bring up the linedef edit dialog box. This box is modal and will allow you to select different lines without closing the box. In this way you can change several linedefs then press the 'Ok' button to get rid of the dialog box. You can pull up the sidedef editor by pressing the "Sides" button on the lifedef dialog.

You can add new linedefs by pressing the "plus" button on the button bar. This places you into *add mode*, click on the vertexes you wish to join together into linedefs. Keep adding linedefs until you have them all placed then click the right mouse button to switch *add mode* off. All the linedefs created will have a default right sidedef created for them.

The program will detect when you try to join two vertexes together that already have a linedef and will ignore the attempt. Linedef creation will continue on from the last vertex clicked on.

You can delete linedefs by pressing the "minus" button on the button bar. This will automatically delete all the sidedefs it uses. Don't forget to check your sectors if you are deleting linedefs that belong to a sector.

Sidedefs

Each linedef has at least one sidedef. A line that needs to be crossed must have two sidedefs. Sidedefs are created and deleted automatically by the program when you add and delete sectors.

The sidedefs defines the textures used to draw the wall section and the front of the area above and below the sector. A sidedef also tells DOOM which sector the left/right side of this linedef is a part of.

You can think of sidedefs in this way: each line has two sides to it but to save space/time

we only want to consider the lines that a player can see. If there is no possibility of the user seeing the back side of a wall why paint it? DOOM uses the fact that many backs of walls will never be seen to help optimisation and it is for this reason that you may have only one sidedef for a line.

Remember, if a line has one one side to it (usually only solid walls are like this) then it is the RIGHT hand side.

For each sidedef there are three textures which can be defined. Above is the texture which is painted if you can see above the ceiling of the sector the sidedef belongs to. *Below* is the texture to use if you can see the front of the area below the sectors level e.g the front portion of a step (sides also) would be painted using the below texture. The wall texture is used to paint the area from the ceiling to the floor for that linedef.

For each texture you have two buttons, marked "C" and "S". Use these to *clear* and *set* the textures for each of the areas. If a texure is *cleared* it will be totally transparent (represented in the editor by having a "-" as the texture name).

Sectors

The final part is sectors. A sector is an area on the map which is totally enclosed and the player is able to see/enter. If an area is not defined as a sector then it cannot be used. You define a series of sectors to form corridors, rooms, teleporters, staircases and the like.

A sector defines the height of the ceiling in an area, it's floor height, light levels, floor and ceiling textures and any special actions which are to occur when you are in this sector. You can make the light in a sector blink by selecting one of the many blink options as a special action or you can bleed them dry of health.

Floor and ceiling height don't seem to have any limits. Room brightness has a range of 0 to 255, with 0 being pitch black, 128 being pretty dingy and 255 being bright daylight.

Sectors can be raised or lowered when triggered by the use of a linedef. Set the trigger value in the dialog for the sector to the same value as you used for the linedef trigger and when they activate the linedef the special action defined for the sector will occur. These actions are ofter used to lower pedastals to place *goodies* within reach of worthy players, operate a sector as an elevator or even drop the light levels.

Through cunning use of sectors and linedef triggers you can make some very interesting levels.

Putting it all Together

So, how does it all work? The best idea is to look at the levels ID provided as well as some off BBS's and see what they have done to make the levels. Study a level first then enter DOOM and see what the level looks like when played. Whenever you see something neat, take a note of it and use the editor to find out what they have done. After you have gleaned an idea of what to do, have a go a making one from scratch.

To start off a level you will need to add some vertexes. Place the vertexes around your map at the spots where the corners, doors, lifts etc will go. If you are planning on changing the height inside a room or using different floor textures then place vertexes for that down.

Now join your vertexes with linedefs. You should lay down a linedef for each wall in the game. Where you want a solid wall to go, put a linedef. Because the player can't move through it leave the linedef set as *impassable*. The default linedef will be a solid wall with a light brown texture as a sidedef.

Place sidedefs across the doorway for all the rooms. To enclose the room and form a sector you need lindefs across the doorways/entrances. If you can't make a complete enclosed area with linedefs then you can't set up a sector there. Your linedefs across doors should be made *two sided* and NOT *impassable*.

Lay down your sectors. Click around the sector clockwise to set the right sidedef's to this sector. Do it anti-clockwise to make the left sidedefs part of this sector. Remember, if a sidedef doesn't exist it will be automatically created when you add a sector that uses that sidedef.

If you want to raise an area up in the middle of the room what you have to do is place two sided linedefs around the platform area you wish to raise. Switch to sector mode and press the "plus" button to add a new sector. Now click your way around the outer wall linedefs (they make the solid walls of the room) in a clockwise fashion. When they are all highlighted click on a vertex of the inner area. Now make your way anti-clockwise around the inner area's vertexes. Moving anti-clockwise means you select their left sidedefs to be part of the big outer sector (I assume you have made the the inner sector linedefs with their right sides facing inwards). Click the right mouse button to stop adding sidedefs to the sector. The room sector is now defined as the area between the walls and the outside part of your inner sector.

Now you need to create a sector for the inner area. Just press the "plus" button and moving clockwise around the inner area click on each of it's vertexes. Right mouse click when they are all highlighted.

Now select the sector for the outer area and set its height, floor etc. Change to the inner sector and make it's brightness just a little higher, give it a slightly higher value for its floor height and a little lower for its ceiling height.

Back to linedef mode. In turn, make your way around the inner sector and set the wall textures to "-". Make the texures for all the right sidedef areas "-" because you will never see the right side of the lines. Give a texture to the *above* and *below* areas for the left sidedef. You've just made a pedastal. The maximum height a player can step up is 24 units so if you make the inner sectors floor 25 units above the floor of the outer sector they will not be able to get up there.

Go to thing edit mode and add a player start spot for each player (there are 4) and at least four deathmatch starting positions to ensure deathmatch play is feasible. It's better to add quite a few deathmatch positions. Add any monsters and objects you want to your level and save it.

Back at DOS you will HAVE to run BSP on your .WAD file to make it useable for DOOM. If you fail to do so you will have major problems.

Tips

Be warned, making a DOOM level from scratch is **hard** work. Most fully fledged levels take from 40 hours to 100 or more, depending on the skill of the designer, complexity, quality of the tools, etc. If you're still keen to make one here are a few tips to get you started.

Target the level
Work out what the target audience will be. Is it for deathmatch tournaments, team play or for single users? The game play for each of these modes is quite different. What makes for a horde of fun in deathmatch is dead dull for single user and great single user levels are ofter just too bulky for deathmatch. It's no fun spending 10 minutes after each frag trying to find the other guy again.

Plan your level
What elements will you use in the level. How many rooms, and roughly what will they be like. Are you going for a theme or will it just be a huge room filled with monsters (again)? Good planning and sketches at the start can save hours or vertex moving later.

Set the difficulty levels on things
Give the learners and less dexterous a fair go. What is easy you (a seasoned DOOMer) may be absolute hell on others.

Avoid the room from hell trick
We've all done it and most of us can see it coming a mile away. Rooms crammed to the top with monsters are fun a few times but don't base your level on it. Rooms lined with caged monsters that are released as you cross it are equally obvious.

Take the time to finish your level
Match up the wall textures. Add lighting effects. Place torches around. It's the little touches that make a level shine. (Yes, I know our demo is guilty of half completeness but it's not meant to be played. It's only a sample file!)

Clean up all HOM's
These really make a level look half done.

Spice it up
Don't settle for a flat terrain with just a series of open rooms. Add dead bodies, put in cages, come up with fiendish traps. Place extra graphics or sound in (coming soon). Try to mimick temples (TEMPLE11.WAD), castles (THE-KEEP.WAD) or keeps (OUTLAND.WAD).

Future Releases.

This is only a first release of our DOOM editor and it's not particulary full featured as yet. We will be adding new features when we get the time but if anyone out there wants to program additions feel free to. Please send us a copy of the source code added to the program (keep it seperate so we can integrate it into our copy).

We have plans and ideas we would like to implement, among these are:

Selection of multiple objects. Changes to be made to all objects.
Viewing of the bitmaps within doom making texture selection easier.
Adding of new graphics, sounds, etc to the PWAD.
Concatenation of multiple levels into one PWAD.
Level ID changing (it's only my slackness stopping it from being in this release).
Import of resources from other WADs.
Texture join helper to line those textures up.
Draw some real arrows for the button bar :)
Hypertext help system with full info on all linedef types, sector info etc. Feel free to send in any info you can on this subject.

We hope you enjoy using our editor. Feel free to send suggestions, gripes, hatemail or source code. Don't bother to send money, just DOOM levels and a thank you for my girlfriend Vicki who's been neglected for the last week. Bye for now and happy DOOMing !

Contact:

Ivan Hawkes
PO Box 801
Indooroopilly
4068
Australia

Compuserve Account Number:100240,1765

Scott Little
PO Box 801
Indooroopilly
4068
Australia

Internet slittle@enterprise.powerup.com.au

Our home BBS is PowerUp in Brisbane. Leave mail for user Scott Little.

Power Up Information Exchange
Phone 61-7-399-1322
Fidonet 3:640/215
Internet enterprise.powerup.com.au