DoomCAD

The state of the art Doom Level Creation System

DoomCAD is a powerful Drag-N-Drop style editor for id Software's Tour de Force game, DOOM. DoomCAD allows you to create new levels for DOOM, and preview them in **THREE DIMENSIONS!**

This software is in NO WAY connected to id Software. It is intended for the personal creation of Doom levels ONLY. You have been warned! id Software cannot and will not provide any support for this product, nor will they provide support to DOOM itself if it is using data files created by this product.

What you need to know first

Doom levels are a complicated set of structures. It would be nice if the DoomCAD user didn't need to know ANYTHING about the underlying structure of a Doom level, but this is unfortunately not possible. It is STRONGLY recommended that the user acquaint himself with the various structures before trying to create a new level.

There are many ways to do this, the best being Matt Fell's fabulous work, "The Unofficial DOOM WAD file specs". The dedication that Matt has put into this work is staggering, especially in light of the fact that he has NO plans for using the specs to create a level editor himself! This exhaustive treatise has become the Doom level editor programmers' bible, having a byte-for-byte description of each structure in the WAD file.

As the Doom level creator, you need not concern yourself with a byte-by-byte explanation of DOOM.WAD, but you WILL need to know what each structure is, and how they relate. How do I add a new monster? A new weapon? How do I create a teleport pad? A staircase? A crushing ceiling?
By acquainting yourself with the Wad file 'lingo', we can speak to each other using the same language. Questions you ask can be more easily answered by other WAD file creators, or the authors of the level editors themselves.

The MINIMUM structures needed to know what makes a Doom level tick, and how they interact, are in the following sections. More information will be given for each structure when talking about editing that structure.

Things

A Thing is ANY object in the game that can be interacted with, as well as various things used as 'scenery'. The Things can be broken into the following categories:

Monsters
Player and Deathmatch Start Positions
Weapons and Ammo
"Scenery" (torches, barrels, various bloody corpses)
Goodies (RadSuits, Armor, Light goggles)
KeyCards and Skull Keys

A thing can be defined to appear on only certain difficulty levels. In addition, monsters, player positions, and teleport detstinations can be set to facing one of 8 directions at the game's start. (All other objects face in 'all' directions at once)

Vertices

(or VERTEXES)

A Vertex is a point in space where walls meet. It has an x and a y coordinate, but not a z, or 'up' coordinate. Doom looks like it is a truly '3D' game, but if you study it carefully, the levels are still really a modified '2D'. Nothing in the game is ever 'above' or 'below' anything else. That is, by looking at a plan view of a level, ALL areas of the map are visible, nothing is ever obstructed by being under something else.

LineDefs

A LineDef connects two vertices together. This line can then be a wall in the game, although not all LineDefs are walls. LineDefs can also be 'transparent', or not visible to the player. A LineDef can have either one or two sides. It will only have one side if the player could never be on the second side (For example, a wall that defines a square column, the 'inside' portion of the wall need not be defined because the player can never be inside the column). A LineDef MUST have two sides if the player could be on either side of it, looking at that LineDef.

In addition, a LineDef has DIRECTION: it goes FROM one vertex TO the second vertex. Because the LineDef has direction, we can define which side of the LineDef is the 'right' side and which is the 'left' side (imagine standing on the 'from' vertex looking at the 'to' vertex, along the line, the right and left sides become apparent.)

This brings us to golden rule #1 of Doom Levels: if a LineDef has only one side, **IT MUST BE THE RIGHT SIDE.** If a LineDef has two sides, then the direction of the LineDef does not matter.

SideDefs

There is one SideDef defined for each visible side of each LineDef. That is, there are either one or two SideDefs defined for each LineDef. Again, if a LineDef has only one side (and therefore one SideDef), it must be the RIGHT side. We call the right side SideDef "SideDef 0" and the left side SideDef "SideDef 1".

The SideDefs define what textures are seen when standing on this side of the LineDef. There are three textures to be defined for each SideDef. The first is the texture of the Wall itself. If the LineDef is transparent, the texture is defined as "-". The second texture is called the "WallBelow" texture, and is the texture that will be seen "below" the level of the floor defined for the other side of the Linedef. For example, consider a stairway. As you go up the stairway, the front of each step is a LineDef. The stair on the other side of the LineDef is higher than the stair you're standing on now, so the "WallBelow" texture is the texture from the floor of the higher stair down to the floor of the stair you're standing on. In other words, it's the texture for the front of the step!

The third texture is the "WallAbove" texture, and is the texture that will be seen "above" the level of the ceiling on the other side of the LineDef. For example, if a column is jutting down from the ceiling, the column's "ceiling" is the bottom of the column, and the columns' "walls" will be the WallAbove texture for the LineDef defining each side of the column.

The last (and maybe most important) thing to understand about SideDefs is that **an enclosed set of SideDefs defines a sector.**

Sectors

This is the important most one to understand. **A sector is an enclosed area of space with a common floor and ceiling.** Therefore, a stairway is made of several sectors, one for each step. The outside courtyard on original Doom game 1, level 1, is two sectors, one for the actual courtyard, and one for the pool of green nukeage in the center. The pool must be another sector because it has a different floor (the nukeage).

Also, a sector's floor and ceiling can change heights, which is how we create lifts, doors, traps, etc. This is an important distinction: **Sectors move, NOT LineDefs.** A Door is a sector whose ceiling is defined at the same level as the floor. When we open the door, the ENTIRE SECTOR's ceiling raises up, and we walk through. When (if) the door closes, the ENTIRE SECTOR's ceiling drops back down to the level of the floor.

Now to tie it all together. To make a Doom Level, we first create some Vertices. Then, we attach the vertices together with LineDefs (creating SideDefs as we go). Then, we select a group of enclosed SideDefs to define a sector. We repeat the process until we have all the sectors done, then we save.

The clockwise rule will help to further demonstrate how the structures fit together. That is, if we were to define a set of Vertices in a rough circle, then travel from vertex to vertex in a CLOCKWISE fashion, ending on the same vertex we started from, the enclosed space we will have just created is a sector. The LineDefs between each vertex will all have one side, and the fact that we went clockwise means that the RIGHT side of each LineDef will be facing inward toward the new sector.

Using the Map

When you first select a level, the entire level is displayed in the window area. Resizing the windows will automatically resize the map.

You can zoom in and out of the map by pressing the + and - keys. Five zoom levels are supported in each direction.

To center on a new spot, Hold down the CTRL key and click on that spot with the left mouse button. In addition, you can pan in either of the four directions by using the four arrow keys. The amount panned is one half the distance currently visible in that direction.

NOTE: The four arrow keys are available for map panning in all modes except Multiple Vertex edit mode, where they are used to move the selected block of vertices 8 units in the four directions.

The shape of the level is determined by the minimum and maximum X and Y values in the map. This has the effect of making the map appear 'squished' in one of the directions, in some cases (original E3M2, the 'hand' level, for example)

You can 'stretch' the map's scale in the X direction by pressing X, or stretch the map's Y scale by pressing Y.

By pressing Ctrl-X or Ctrl-Y, you can 'squish' the map scale in the X and Y direction.

In general, the various objects are moved around (if they are moveable), simply by clicking on the object and dragging it around. The right mouse button is used to bring up extended information about the current object, for further editing.

Thing Editing

 To get into Thing Editing Mode, select the thing toolbox item

When in Thing Editing Mode, all the Things appear as colored dots using the following color codes:

Red: Monster
Dark Gray: Player Start Positions
Cyan: Weapons/Ammo
Light Gray: Scenery (torches, carnage, etc)
Green: Barrels
Magenta: Goodies (RadSuits, Armor, Light Goggles, etc)
Dark Yellow: KeyCards and Skull Keys

To move an existing thing, just click on it and drag it to the new location.
To add things, select INSERT THING from the menu. The new things will be an exact duplicate of the previous thing. Click on the map for each position for the new thing, adding as many new things as you wish. When you are done adding, right click with the mouse to turn insert mode off.
To delete a thing, select DELETE THING from the menu. The current thing will be deleted.

To edit a thing's attributes, right click on the map when a thing is selected. The Thing Editing window will appear. To change what the thing is, select first the Type of thing using the "KMAGS" buttons. Each button stands for the following:

K: Keys (and Player Start positions)
M: Monsters
A: Ammo and Weapons
G: Goodies (potion, armor, stuff you can pick up except weapons)
S: Scenery (including barrels)

Once a button is selected, choose the desired thing in the list box.

The angle boxes are relevant only to monsters and player start positions, as other objects always face the player. The skill bits check boxes are for selecting which levels the thing will appear on, and other attributes:

1/2: Appears in difficulty levels one and two.
3: Appears in difficulty levels three
4/5: Appears in difficulty levels four and five. (Difficulty level five available as of Doom version 1.2)

Deaf: Clicking on this check box will make a monster 'deaf': he will not respond to gunshots within earshot. He will attack a player only when he sees him.

Net: Makes the thing available only in network games.

 **Multiple Thing Editing.** By selecting this box, you may drag a box across the map to select multiple things. Then, by moving the box, all things within the box will move at once. In addition, if you right click on the box with multiple things selected, you may ALIGN the things in either the x or y direction by clicking on the appropriate check box.

Vertex Editing

 To get into Vertex Editing Mode, select the Vertex toolbox item

To move an existing vertex, just click on it and drag it to the new location. You may also right click on the vertex and type in the coordinates manually.
To add vertices, select INSERT VERTEX from the menu. Click on the map for each position for the new vertex, adding as many vertices as you wish. When you are done adding, right click with the mouse to turn insert mode off.
To delete a vertex, select DELETE VERTEX from the menu. The current VERTEX will be deleted. (You cannot delete a vertex that has one or more LineDefs attached to it).

 **Multiple Vertex Editing.** By selecting this box, you may drag a box across the map to select multiple vertices. Then, by moving the box, All vertices within the box will move at once. In addition, if you right click on the box with multiple vertices selected, you may ALIGN the vertices in either the x or y direction by clicking on the appropriate check box. This is good for creating straight walls.

Also, when you have multiple Vertices selected, you may use the arrow keys to move the selected block of vertices 8 units at a time in four directions. (In all other modes except multiple Vertex edit mode, the arrow keys pan the map).

Sector Editing

 I'm jumping right to Sector editing because you can create new Sectors, and that sector's defining LineDefs/SideDefs, all in one step!

To get into Sector Editing Mode, select the Sector toolbox item

Once a given sector's Vertices have been defined, it's VERY easy to create a new sector from these vertices. Just go into Sector editing mode and select INSERT SECTOR from the menu. Then, connect the new vertices in a CLOCKWISE fashion by dragging the displayed red line from Vertex to Vertex and clicking the mouse button.

Once the sector has been defined, simply right click with the mouse. You will have just done the following:

Created one LineDef between each pair of Vertices.
Created one SideDef for each LineDef, (on the right side, of course)
Create one new Sector. All the new SideDefs point to this sector.

That's it!. Now, just right click with the mouse to edit the sector's attributes. You may also edit the attributes of an existing sector by clicking on one of the LineDefs associated with that sector. Clicking on the LineDef a second time will switch to the OTHER SideDef for that LineDef, and it's associated sector.

To edit an existing sector, make that sector current, and then select "Edit Current Sectors LineDefs" from the menu. Then, select the new LineDefs to attach to that sector. You can pick a LineDef that's already attached to this sector while in this mode, and DoomCAD will not duplicate the definition.

To delete a sector, select DELETE SECTOR on the menu with that sector highlighted. All SideDefs pointing to that sector will be deleted.

Sector Edit Dialog
The following attributes can be edited for an existing sector:

Ceiling/Floor Height.
The upper and lower limits of floors/ceilings are unknown. I've seen ceilings as high as 512 or so.

Ceiling/Floor Texture:
To choose a texture, just find the texture you want in the list box and drag it to the Ceiling/Floor Texture box. Double clicking on the texture will display it in the Preview Window.

Special List: Choose one of these attributes for the sector. The -10/20% attributes are how the 'nukeage' areas are created. The secret areas are added up to get the secret

percentage at the end of the level. (To get credit, the player must step on the sector marked secret. This is easily accomplished by putting something good in the sector, like ammo or a medkit.)

Brightness: Can go from 0 (dark) to 255 (outside light).

Trigger: This number ties the sector to a LineDef somewhere on the level with the same trigger number. See the LineDef Editing section for a description of triggers.

 **Multiple Sector Editing.** By selecting this box, you may drag a box across the map to highlight several sectors. All Sectors defined by any LineDef within or crossing the box will be selected. Right clicking on the box will then bring up the Sector Edit screen. You may edit Ceiling and Floor Textures, Ceiling and Floor Heights, and Brightness for multiple sectors. Clicking ok will change the attributes of all selected sectors

LineDef Editing (also SideDef Editing)

 To get into LineDef/SideDef Editing Mode, select the LineDef toolbox item

To Select a LineDef, click on it near its center.
To add a new LineDef, select INSERT LINDEF from the menu. Click near the starting vertex of the new LineDef, then drag over to the ending vertex, and release the mouse button.

(**NOTE**: It is recommended that LineDefs are created using either ADD SECTOR, or EDIT CURRENT SECTOR'S LINEDEFS, rather than using INSERT LINEDEF. The reason is that when using the former two functions, SideDefs are inserted automatically and tied to the current sector. Using INSERT LINEDEF, no sector ties are created, they must be tied manually later using one of the two former function anyway!)

To delete a Lindef, select DELETE LINDEF from the menu.

To edit attributes for the LineDef, right click on it. The following attributes are editable for each LineDef:

Length: Changing the length will change the position of the LineDef's TO vertex. Note that if other LineDefs are attached to this vertex, their lengths will change as well! This is good for making sure a Wall is exactly 128/256/nn long, to match its texture size.

Action: The largest listbox defines what the LineDef will do when it is 'Activated'. The activation event is one of the following, denoted by the first characters in the listbox elements:

W: Linedefs is transparent and must be <W>alked over to trigger action to triggered sector.
D: Linedef is a <D>oor: SideDef1 of this LineDef points to the door's sector
S: Wall is a <S>witch, and must be pressed to trigger action to triggered sector.
1: Action happens once
R: Action happens repeatedly

The events are split into 3 sections. Select one of the three radio buttons to see the three lists.

Doors/ Msc. (All the "D" items, above and non-"S" or "W" items) These are doors and the few miscellaneous items (scrolling walls, the 'shoot me' trigger)

Switch: (All the "S" items, above). Requires the user to press a switch first. NOTE: You need to define this wall using a "SWITCH" graphic, or the user will have no visual cue that the wall needs to be pressed to activate the switch.

Walk: (All the "W" items, above) Walking across the LineDef activates the event.

Some actions happen only once at a time, but can be 'undone' to happen again. For example, if a sector is triggered to "Move Down to Neighbor Floor" by crossing LineDef A, and to "Floor Move up to Neighbor Ceiling" by pressing switch on LineDef B, that sector can move up and down by repeating the two triggers

Attribute Bits:
Bit 0 Impassable: Players, monsters, and projectiles will not cross this line.
Bit 1 Monsters cannot cross: They can't cross if Impassable is set anyway, so this is good for 'invisible' cages or the like.
Bit 2: Transparent. If you don't set this bit and select '-' for either of the SideDef textures, the player will see "hyper junk" (quoted from Matt Fell: I love that!) Also note that if this bit is set, projectiles WILL travel through this LineDef, even if there is nothing on the other side (probably resulting in a Doom CRASH).

NOTE: Every LineDef must either be Impassable or Transparent. Both of these attributes may be set, creating a Transparent area you can't walk through (a Window). An unpassable window may also be created by making its floor height 32 above the current floor height.

Bits 3/4 :Unpegged textures do NOT move with moving ceilings/floors, so the effect is that the texture "just sits there" as the ceiling/floor moves, looking more like the texture is being created to fill the space. The best way to describe this is to see it: mark a door as having an Unpegged Ceiling and watch it open/close. Door Track textures are marked as unpegged, in many cases.

Bit 5: Secret: This protects the drawing of what's behind the LineDef on the Automap. It does NOT have anything directly to do with finding secret areas that count toward the percentage at the end of the level. Those secret areas are sectors with a secret attribute set.
Bit 6: Blocks Sound. Sound does NOT travel through Impassable walls, anyway. Use this bit to block sound going through an open door, so that the monsters you have waiting to ambush in the next room don't respond to the fighting in the current room.
Bit 7: Not shown on automap. Good for outer level boundary walls.
Bit 8: Already on map: This Line is seen right from the start of the level, even if the player hasn't been there yet.

Trigger: Trigger numbers tie the LineDef to a Sector with the same Trigger number. Type the trigger number you wish in the box provided. If you wish to create a new trigger, select the 'new' button above the trigger box, and a previously unused trigger number will be assigned. Once a trigger has been assigned, clicking on the 'Sector...' button will bring up the Sector Editing box for the Sector having the same trigger number.

NOTE: A LineDef can be tied to more than one Sector, and vice-versa: If this is the

case, clicking on the 'Sector...' button will bring up only ONE of the Sectors tied to this LineDef. The other Sector can be edited in Sector Editing Mode, of course.

SideDef Data:
X,Y: The Coordinates that textures will be 'pasted' onto the Linedef.
WallAbove: The texture that is visible on any part of this LineDef above the corresponding sector's ceiling. For example, it this SideDef's Sector has a lower ceiling than its neighbor sectors, the side of the wall will be visible. This texture defines that part of the wall. (Note: Doors are initially defined as a sector whose Ceiling is at the same height as the floor, so the door graphic is actually the WallAbove texture. The Wall texture for a door sector should be left "-", so the door can be passed through when it is opened.

Wall: The texture that makes up the area between the ceiling and floor of the corresponding sector. This texture is usually "-" for transparent LineDefs, although SOME of the transparent LineDefs have textures (cages). This must be defined for Impassable LineDefs (on both sides, if they exist), or you get 'hyper-junk' on the wall.

WallBelow: The texture that is visible on any part of this LineDef below the corresponding sector's floor (like the front of a stair)

To edit the above textures, drag the desired texture from the scrolling listbox and drop it in the desired box. (The scrolling listbox shows the texture name, height, and width). To clear a texture, click the small button marked 'c' next to the desired box. To preview a texture, double click on the texture, either in the listbox or the Wall/WallAbove/WallBelow box (the palette for the picture may not be exact, Windows won't let me have all 256 colors!)

 **Multiple LineDef Editing:** Using this tool, you can Select MULTIPLE LineDefs by creating a box on the map. Any LineDef partially enclosed by the box will be selected. When you right click on the map after selecting the LineDefs, the LineDef edit dialog box will be displayed. The following information can be edited across multiple LineDefs:

Attributes (Impassable, unpegged textures, etc)
SideDef 0/1 textures (Wall, WallAbove, WallBelow)

Keep in mind the following when editing multiple LineDefs:

The SideDef 1 (left side) editing box will be available if ANY ONE of the selected LineDefs has two sides. Editing the SideDef 1 texture data, however, applies ONLY to those LineDefs with a second side. LineDefs with only one side will not be affected by editing SideDef 1 data.

Three Dimensional Preview

 3D preview allows you to look at the current Doom level in a three dimensional black and white preview, with hidden lines removed. Clicking on the camera icon will bring up a Blue Line segment, representing a camera. The lower left side of the line is the camera location (where the viewer is standing). The other line segment is the camera viewpoint (where the camera is looking). You can move either one of these viewpoints around. When you get to a spot you'd like to see in 3D, press the right mouse button.

You can change the height of the viewpoint and the height of the camera location with the two slider bars. After changing heights, press <redraw> to redraw the scene.

You can also Zoom in and out of the 3D view by pressing + and -. Make sure the 3D Window (and not the ToolBar Window) is the active one when pressing + and -.

Software Creations

"Home of the Authors"

Don't forget where id Software's home is! The one and only Software Creations BBS.

1200/2400 V.42/MNP Lines : (508) 365-2359
2400-14.4k HST US Robotics Lines : (508) 368-7036
2400-16.8k V.32/V.42bis US Robotics lines : (508) 368-7139
14.4-28.8k V.32/V.42bis/V.fc Hayes Optima lines: (508) 365-9352
14.4-28.8k V.32/V.42bis/V.32terbo/V.fc US Robotics lines: (508) 368-3424

This is by far the best BBS on earth! Call today!

Note: Software Creations is NOT responsible for distribution of DoomCAD, it is being distributed solely by the author. Registration will be handled solely by me!

PreFab Mode

 Prefab Mode lets you create Prefabricated structures. Specifically, you can create stairways, doors, or rooms in a few quick steps.

To add a Prefab object, select Prefab mode then Insert Prefab <door/ room/ stairway> on the menu. A dialog box will come up, asking questions about the structure. When the structure definition is complete, you will locate the spot on the map to place the structure and click. The structure will then build itself.

Prefab Mode is particularly easy for building stairs. If your Stairway is going to be inside another sector (a room, for example), be sure to build that sector first. Then, while in PreFab mode, select that "parent" sector by clicking on it, as you would do in Sector Editing mode.

Now select BUILD PREFAB STAIRS from the menu. Note that one of the editable fields on the Prefab Stairs Dialog box is "outer sector", and its number is the same as the sector you just created. (You can also set this number to -1, if your stairs won't be "inside" another sector, like if they're going to be in a hallway).

Select the remaining stair options as you wish, number of stairs, stair height, etc. When you click the ok button, place the stairway inside the highlighted sector. All SideDef/Sector pointers will take care of themselves, and you've just put a stairway inside a room!

To build a Prefab teleporter, first select a 'parent' sector (the room/hall that the teleport pad will be in), just as you would with Prefab stairs. Then, select the BUILD PREFAB TELEPORTER menu option. Then, place the teleporter inside the parent sector. DoomCAD then reminds you to complete the teleporter by: (Note that while placing PreFab teleporter, the Grid Snap is temporarily set to 64 in each direction. This is to ensure the teleporter's ceiling and floor textures are drawn correctly).

1). Going into sector mode, highlighting the 'destination' sector of the teleporter, and giving it the proper trigger number.

2) Creating a teleporter destination object in that sector (Note that by changing this object's facing direction, you change which way the player will be facing when he arrives.)

To create a two-way teleporter, repeat the process as above. Make sure that you assign the sector of Teleport B the trigger numbers of Linedefs in Teleport A, and vice-versa. Also, make sure there is one (and only one!) Teleport Destination Thing
in each of the sectors.

Note that placing prefab rooms or doors has no automatic option for placing them "inside" an existing sector, so if you do place a Prefab room inside an existing sector, be sure to go into Sector Edit mode, highlight this outer sector, and Select "EDIT CURRENT SECTOR'S LINEDEFS" from the menu. Then trace around the outline of the Prefab room to attach its outer walls to the parent sector.

In the case of doors, both outer walls (SideDef 0 is the side of the walls with the actual door graphic) will need to point to new sectors, those being the room (or hall) the Doom player is

standing in when he's looking at the door. This is done in the same way. Select that sector in Sector Editing mode, choose "EDIT CURRENT SECTOR'S LINEDEFS" from the menu, then trace around the sector, including the LineDef containing the door.

About The Authors

matt tagliaferri: just your average Doomaniac with way too much time on his hands. I quickly played all 27 levels of the game, and was left craving more. I have two motivations for writing DoomCAD:

Creating some quality levels for Doom play.
Having hundred of users create levels for my use! (ha)

other contributors:

Matt Fell*:* Author of *"The Unofficial Doom WAD File Specs",* my personal bible for this project. His tireless quest for knowledge made DoomCAD possible, because I probably wouldn't have had the patience to hack a 10 meg file in its entirety!

Raphael Quinet: Author of Deu 5.0, the FIRST level editor with levels from scratch support. He was the first editor author to delve into the complicated Binary Space Partition (BSP Tree) algorithm needed to correctly save a Doom level. His generous inclusion of source code for this algorithm was the final piece of the puzzle for DoomCAD's ability to create levels from scratch.

Everett Pence: another Doomaniac who's sick of playing the original 27 levels. He wrote the initial code for the graphic preview, and had many good suggestions for interface improvement.

Carlton Guc: Beta tester extraordinaire, and future hall-of-fame level author. His bent imagination is perfect for creating evil Doom levels. (Example: When DoomCAD was just a thing editor, he put a Cyberdemon in the outside "secret" area of E1M2, and he moved the red key out there! We were playing 3 player cooperative mode, and didn't have any weapons yet, either....)

Brad Rembielak: Beta tester extraordinaire number II, and NetPlay partner. Fastest use of a mouse that I've ever seen.

Micro System Options: Creator of 3D Graphic Tools for Windows, the library used for 3D preview. Great product for 3D graphic gearheads like myself. Believe it or not, the 3D stuff took me only 6 hours to implement using this great product.

id Software: the ones who started this mess. They didn't do much <g>, except provide us with the most immersing 3D game to date, leave the file formats unencrypted and hack-able, then give us permission to hack it and charge money for it! Special thanks and applause to this stand-out organization.

Registration

To register DoomCAD, please send a paltry $15 to:

matt tagliaferri
4416 Brooklyn Avenue
Cleveland, OH 44109 (go **TRIBE!**)

id Software is graciously allowing Doom Editor authors to charge money for our work, please support us through registration.

As of version 4.2, DoomCAD is longer crippled in any way, due to strong customer demand (and a non-trivial number of flames directed my way).

Source code for various sections is available. I cannot provide source for the 3D stuff (not in my license), but if you'd like my node/blockmap algorithms, let me know. DoomCAD is written in Visual Basic 3.0, professional edition.

Nodes

(also BlockMap, SSectors, Segs, and Reject)

These are additional structures that you as Doom Level author need not concern yourself with, but Doom Level *Editor* authors (like me) pulled their collective hair out trying to figure out! These structures are created from the other structures (Vertices, Sectors, etc). The automatic generation of these structures is quite complicated, and if it fails, your Doom level will not display properly. If your level fails during Nodes/Blockmap etc generation, please EMail me or send the WAD to me and I'll try and figure out what the problem is.

Other Features

These are some additional features of DoomCAD:

Save as BMP: Saves the current Window as a BMP, suitable for printing.

Grid: The standard grid feature found in many drawing packages, allows your movement to be restricted to user specified intervals. Grid snap settings are saved from session to session. Note that if not grid is active, a minimal Grid setting of 8 units is always active (all Doom levels from id snap to an 8x8 grid, it is unknown if a level will display correctly which does not conform to this grid snap)

Direction Marks: Shows the right hand side of each LineDef with a perpendicular indicator mark. If your LineDef is facing the wrong way, select that LineDef and choose FLIP LINEDEF from the menu.

Flip LineDef will swap the from and to vertices, as well as the SideDef pointers (You may only FLIP a LineDef with two SideDefs defined, to Flip a one sided LineDef, delete the LineDef and redraw it). You will usually need to FLIP the long LineDefs of Doors you create (if you don't use Prefab mode), because the SideDef0 of a Door points to the ROOM the Door is facing and SideDef1 points to the Door itself.

Split LineDef will split the current LineDef in half. ALL attributes of the original LineDef will be present in the two halves, including sector definitions,
textures, etc.

NODEBLD.EXE: This program will rebuild the Nodes, Segs, SSectors, Reject, and Blockmap for the .WAD file given as a command line parameter. It uses the same algorithm that DoomCAD uses, but is much faster because it is a compiled DOS executable. If you wish, you may save DoomCAD levels with the None (quicksave) option for Node Building, and then use NodeBld.EXE on the WAD file from DOS just before you test the level in Doom.

AutoBackup: Before any WAD file is overwritten, it is saved under the name OLD.WAD in the same directory. This way, if DoomCAD happens to crash during the Node Building process, you have a backup of your previous WAD file.

Saving Your Work

When it's time you save your WAD file, you're presented with several choices.

Game. You may choose Game 2 or Game 3 to save this level. Game one maps cannot be created, at id's request.

Level. Choose which level (one - nine) to save this game as.

Integrity Check. This is the same as the Integrity check option from the menu.

Node Building:

You have three options for the saving of the NODE, SSECTORS, SEGS, etc structures.

Full Build (default). This option will build the Nodes structures from the current level. Be patient, this can take several minutes.

Use previous. If you are saving an old level, you may use the Node structures from your last save of the level. Note that this option is available ONLY if you have made NO changes to structure data (Vertices, LineDefs, Sectors) since your last save. This option is useful when a level's structure is complete, and you are now just adding THINGS, and you don't need to rebuild the NODE data between each save.

None (QuickSave). This option doesn't bother creating or saving NODE data at all. **IMPORTANT: You cannot play a level in DOOM if you do not build the NODE data.** This option is useful in the following circumstances:

1). You're done working for the day, you don't wish to test the level, you just want to save until next time.

2). The Node algorithm doesn't work on the current level. You can use the quicksave option so you don't lose your level. If you wish, you can send the WAD to me to aid me (and Raphael Quinet) in debugging the Node algorithm (Don't get me wrong: I HAVE found it to be pretty stable).

3) You'd rather use the DOS version of the Node generator included in DoomCAD (NODEBLD.EXE), or another third party NODE generator (like BSP).

Integrity Check

This feature looks for anomalies in your levels, such as:

Impassable walls with no texture defined.
Transparent walls WITH a texture defined.
Walls not defined as either Transparent or Impassable.
Sectors with a ceiling lower than its floor.
LineDefs/SideDefs that don't point to any sector
Missing WallAbove/WallBelow textures
Orphaned sectors (defined sectors with no SideDefs pointing to them)
Orphaned SideDefs (defined SideDefs with no LineDefs pointing to them)
Unclosed sectors

Many of these anomalies are handled automatically, some are just displayed as an error or warning for you to correct.

Note that this function is intended to be used *only* when you think your level is done and ready to be played. Do not use it after every LineDef/Vertex/Sector insert or delete. For example, if you use the function INSERT LINEDEF (as opposed to creating LineDefs automatically through INSERT SECTOR). create a LineDef, then immediately perfrom an Integrity Check, that new LineDef will be deleted! Why? Because the LineDef doesn't point to any sectors yet. LineDefs that don't point to any sectors are automatically deleted.

Level Troubleshooting

If the Node algorithm fails on your new level: look for the following things:

Unclosed sectors. Sometimes, you will have a sector and want to delete it. You delete all the LineDefs for that sector except maybe one or two, which you decide to use as the start of a new sector. Those one or two LineDefs are actually still pointing to that old sector (You never explicitly said "Delete sector"). And that sector is hard to find now (those one or LineDefs are hard to locate out of the 100s you have defined.)

When this happens, go into Sector mode and click on each LineDef to see which sector it points to. Click on it a second time to see if it has a second sector pointer. If the description <no sector> is displayed for that LineDef, you have a SideDef with a null sector pointer. Integrity Check should delete this SideDef.

As of version 4.2, Integrity check will find unclosed SINGLE PART sectors. Multipart sectors (where sectors are entirely enclosed by other sectors, like the nukeage pit in original E1M1), may still go undetected.

Wrong way LineDefs: Make sure all one-sided LineDefs point INWARD to the sector they enclose. You can make sure of this by turning on DIRECTION MARKS in LineDef editing mode. Two sided LineDefs can point in either direction.

If your level saves, but you get the 'hall of mirrors' effect on some walls, look for the following:

Transparent Walls with a texture defined.
Impassable Walls with NO texture defined. (and no sector on the other side)
Missing WallAbove or WallBelow definitions.

Integrity check should notify you of ALL of these instances, with the exceptions of SideDef pieces made visible by the movement of a sector (ie: a lift). Integrity check currently works only on the starting state of the level, before movement of any sectors.

Template Mode

 Template Mode is a powerful cut and paste facility. It allows for the complete saving of any area of any current Doom level in a special 'TWAD' file. This file conforms exactly to the same standard that standard Doom .WAD files use, so other Doom editors should be able to load them just like minature levels.

To go into Template mode, click on the Template button. The following options are available in Template Editing Mode.

Cut/Copy: Cuts/Copies the currently selected area from the map, and stores it in the Cut Buffer. The following data is cut:

Things
Vertexes
LineDefs/SideDefs (only those that have BOTH vertices within the selected box, those with only one Vertex inside are not cut)
Sectors

Paste: Allows the current contents of the Cut Buffer to be placed on the map, Place the moving rectangle where you would like the new area and click on the map. All data in the cut buffer will be pasted into the new area.

Save Cut Buffer: Saves the current data in the cut buffer into a 'TWAD' file. This data may be loaded at any time and pasted into any level.

Load Cut Buffer: This allows for the loading of a valid TWAD file into the Cut Buffer. Double clicking on the filenames, or clicking on the 'preview' button, will display that TWAD's contents in the Preview Window. **NOTE:** The current contents of the cut buffer is cleared when you click on the 'Preview' button, so that the TWAD's contents may be loaded and displayed. Don't use PREVIEW if you have information in the cut buffer that you haven't saved!

Pay close attention to the following when Pasting from the Cut Buffer:

New sectors are created, as opposed to using the same sector ids as the original data. This is done so that a Cut buffer can be saved into a TWAD file and loaded into ANY map as new data. The down side of this is that if you wanted to copy a teleporter, for example, and paste a duplicate teleporter in the same room as the original, you will have to remap the new teleporter's 'parent' sector to the parent sector of the original teleporter, and delete the new 'parent' sector that's created by the paste.

I realize that this is a bit of a pain, but the alternative is worse: that is, NOT creating new sectors and mapping all the new LineDefs/SideDefs to the old sectors. If you did indeed want to create a whole new area, you would have to remove all the sector

references of all the new lines and create new sectors anyway! (Not much of a cut/paste). Or worse yet, if you're pasting a cut buffer from a totally different Doom Level (say one of the original levels), the sector pointers would have NO correspondence with the sectors in your current map.

I was thinking of allowing the user a choice between either remapping or retaining original sectors, but this choice would have to be for EVERY sector in the cut buffer. Going back to our duplicate teleporter example, we need one new sector (the new teleporter), and we need to retain one old sector (the original teleporter's parent). This would be a hassle for the user, to specify "map/don't map" for all the sectors in the cut buffer.

The one id that IS retained in the cut buffer are Trigger pointers. That is, if you cut/copy an area with trigger pointers, these ids are retained in the paste, which may not be desirable on your new level. Creating new triggers is easy, however: just go into the LineDef editor for the trigger you want to change and click the 'new' button.

 **Note:** In Template mode, only the Multi Edit Tool is available.