

```

;
;      HELP.TXT for EdMap v1.10x+
;
;lines starting with ";" are ignored
;
;help entry format:
;      =====
;      <keyword>
;      =====
;      <text>
;["see_also"
;<keywords>]
;
;the "====" line must contain nothing but "=", of any length>0
;
;<keyword> is the name EdMap will search for to find any entry
;      from the menus, the function name is the keyword (ex."New Map")
;
;<text> is the body of the help message.  there cannot be
;      and <CR>s in the text unless it is used to start a new
;      paragraph, which should be indented with a tab (#8).
;
;if there are reference keywords to other help entrys,
;      "see_also" must mark the end of the text and the start of the
;      keyword list.  keywords are one per line, no characters other
;      than the actual letters (numbers, etc: #32..#122?).
; not case sensitive; lowercase is made uppercase
;
;
;of course, after editing this file, MAKEHELP.EXE must be run
;      to create the index file used to read this file.
;
=====
EdMap basics
=====
      Editing in EdMap is done in one of four modes: vertices, linedefs
(walls), sectors (rooms), and things.  Each mode controls the objects of
that type.  While in any mode, objects may be highlighted or selected.
Normally the nearest object to the mouse at that moment is highlighted,
this is the Current````object.  If enter is pressed, the current````object
becomes the Selected````object and remains selected until enter is
pressed again or the mode is changed.  Almost all editing is done on the
current object, but some operations require a selected object.
see_also
creating sectors
merging sectors
object basics
add/split
delete
editing objects
menu commands
display arrangement
help

=====
Menu commands
=====

```

The menu in the upper left (in standard view) is controlled by pressing the first key of the desired selection or by clicking on the selection with the mouse. For example, to check the map for any errors, press "C" for the "Check" menu, the press "E" to check everything.

see_also
mouse control

=====
Creating sectors
=====

To create a new area, like a room or a different part of a room, insert a sector. While in sectors mode, press insert (or the mouse command) and a new 64x64 sector will appear under the mouse pointer. Note that this is usually done outside any other sector, when the object information panel shows no sector.

see_also
merging sectors
sectors

=====
Merging sectors
=====

Sectors are merged when they share a LineDef. To make two sectors merge, a line from one must be moved ontop of a line from the other so that the vertices connect. When both pairs of vertices fuse together, the two lines will merge into one two-sided linedef marking the boundary of both sectors.

see_also
sectors

=====
Display arrangement
=====

While in normal view, EdMap displays information to the left while the rest of the screen shows the map. The top half of this is the main menu. Below the menu is a small square for general information (filename, memory, position, etc) and a mode bar highlighting the current mode. The bottom half is the object information bar. This provides data about the currently marked object.

If the screen is set to full-view, all information to the left is removed, leaving the entire screen to display the map. The menu can still be used in this mode with the keyboard.

To switch between normal and full screen, either use the hotkey: Ctrl-S, the menus: press "D" (Display) and "F" (Full screen), or the mouse pad.

see_also
Using the menus
Mouse pad

=====
Using the menus
=====

To choose a selection from a menu, either click on the menu with the mouse or press the key of the first letter of that selection. For example, to bring up "Map Utilites" press "M" and the Map Utilities menu will appear. From here, pressing "S" would choose "Shift Map" instead of "Sectors" from the main menu. To return to the main menu, Press

backspace or escape.

In full-screen the menus still appear when brought up (as by pressing "M") but are not displayed when not in use.

=====
Mouse pad
=====

The mouse pad in EdMap is a tiny menu that appears under the mouse pointer which can be used to quickly change modes or screen size without using the keyboard. While holding the first mouse button, move the mouse slightly to bring up the pad. Release the button to select. To use LineDef mode, for example, hold the first button and slide the mouse to left and release.

see_also
Mouse control

=====
Object basics
=====

Objects are the basic components of maps; vertices, linedefs (walls), sectors and things are all objects. Vertices are the points on the map that all other structures are built from. Each LineDef connects two vertices together. Sectors are useable areas on the map, bordered by LineDefs. Everything happens within sectors; no players or monsters should exist outside sectors in the void. Things are actual objects in the game; monsters, weapons and player starts are all things.

see_also
Vertexes
LineDefs
Sectors
Things
editing objects
EdMap basics

=====
Info
=====

This provides general information about the map and the status of EdMap itself. Also the Preferences panel can be used to adjust some of the defaults in the config file that are not accessible from EDMAPCFG.

see_also
Calculator
Map Information
System Information
About EdMap
Preferences

=
help
=

This help panel provides assistance for the current feature. While in almost every panel, F1 may be pressed to activate the help panel. Clicking on the "Ok" button, pressing Enter or ESC will leave help and return to the previous function. If memory is low a disk file will be created to store memory, unless there is too little memory to do even this.

Whenever in any panel, buttons on the panel are used to input information. Use these buttons by moving the mouse arrow over them and

click the left button (unless the buttons are switched). ESC may be pressed from any panel to abort, and Enter may be pressed to accept.

If a panel is active and an error occurs, but the active panel must stay in the foreground, the message "error" will appear. (Messages are in a red box in below the menu in normal view). When the panel is done the error will appear in a normal panel.

see_also

EdMap basics

=

Map Information

=

The Map Information panel displays statistics about the map.

see_also

Info

=

System Information

=

System Resources provides information concerning the state of the operating system, including memory usage.

see_also

Info

=

About EdMap

=

The About panel displays version information about EdMap.

see_also

Info

=

Preferences

=

The Preferences panel can be used to customize the editor. The Play`map skill level, mouse x/y sensitivity and double-speed threshold, error checking and some verify options are set from Preferences. Changes are saved to the config file.

see_also

Info

=

File (map)

=

The File/Map menu handles the currently loaded map and control out of EdMap. This menu controls only the current PWAD; see WAD``list for more on managing the active PWADs.

see_also

New map

Open map file

Load PWAD map

Rename map

Save map data

Hard copy (print)

Build & save map

Play map

Quit to DOS

WAD list

=

New Map

=

New Map will clear the memory of the current map and start fresh

with a new map. The new map is a 64x64 room (using the current styles) with the Start-1 object at the origin.

see_also
File (map)

=

Open map file

=

This displays all the *.WAD files in the pwad directory (as specified in EDMAPCFG), so a PWAD map file may be chosen and loaded. The files may be sorted by name or timestamp (last save).

Files may also be deleted from this panel. Highlight the map to be deleted with the mouse and press delete.

see_also
File (map)

=

Load PWAD map

=

This displays all the maps (E1M1 to E3M9) and loads one according to the PWAD list.

see_also
File (map)

=

Rename map (ExMy)

=

Renames the map to a different episode/mission. Generally single maps are E1M1; the first mission. If separate maps are concatenated, however, each must be under a separate mission.

see_also
File (map)

=

Save map data

=

Saves the current map to the PWAD directory. If the map has no filename a panel will ask for one. The main DOOM.WAD file will not be written to. (DOOM.WAD should be read-only to prevent anything from accidentally changing it.)

Saving the map does not build the BSP data necessary to play the map. See Build & save map about creating the BSP.

see_also
Build & save map
File (map)

=

Hard copy (print)

=

(may not be available yet)

Prints the current map to the printer on the port specified in EDMAPCFG.

see_also
File (map)

=

Build & save map

=

Saves the current map and builds the BSP data necessary to play it. If the build fails to complete for any reason, the map will be automatically reload as it was prior to the build.

see_also
File (map)

```

=
Play map
=
    Saves, builds, and plays the current map. Similar to
Build`&`Save`map, but instead of returning directly to the editor,
DOOM.EXE is loaded with the current map. DOOM will start in the map,
skipping the demos. The skill level chosen is specified in the
Preferences panel. If saving and/or building is unnecessary, it will be
skipped. Additional parameters for DOOM may be added using EDMAPCFG.
    Note that id software is not responsible for any maps created or
modified using this editor.
see_also
Preferences
EDMAPCFG
File (map)
=
Quit to DOS
=
    Exits EdMap. If the map has unsaved changes, a panel will ask what
to do.
see_also
File (map)
=
WAD list
=
    This menu handles the PWAD files. Loading, creating and listing
PWADs is done using WAD`list. While the File`(map) menu controls the
current map/single PWAD, WAD`list handles all of the active PWAD files.
see_also
List WADs
Remove PWAD
Add PWAD file
Save as PWAD...
PWADs
File (map)
=
List WADs
=
    Lists the active PWAD files and the maps they contain (that are
not outdated by other PWADs).
see_also
PWADs
WAD list
=
Remove PWAD
=
    (may not be implemented yet)
    Removes a PWAD from the PWAD list. This does not delete any
files.
see_also
PWADs
WAD list
=
Add PWAD file
=
    Adds a file to the PWAD list as if it were included on the command
line as a parameter. Open`map`file preforms the same function (and is

```

easier to use).

see_also

Open map file

WAD list

=

Save as PWAD...

=

Renames the map file and saves it. After this function, all saves will be to the new file.

see_also

WAD list

=

Edit

=

The Edit menu provides control for manipulating objects on the map.

see_also

Add/Split

Delete

Merge

Find

Next object

Previous object

Tag line to sector

=

Add/Split

=

This function serves different purposes for various modes.

In Vertex mode this is a vertex-break. All sectors attached to the current vertex break and recede from the that vertex. This is extremely useful for correcting mistakes.

In LineDef mode this will split the current linedef into two halves.

In Sectors mode a new sector is created on insert. Normally sectors are created in the void and then added to the map, but if there is a sector currently highlighted during the insert, then the new sector will be made within that sector.

In Things mode this will duplicate the current Thing object. If there is no current Thing, a new object is created.

see_also

Vertexes

LineDefs

Sectors

Things

Edit

=

Delete

=

Like Add/Split, the result of Delete depends on the mode.

This does nothing in Vertex and LineDef mode.

In Sectors mode, deleting removes the selected sector. If the sector is within another sector, void is left where the sector used to be, surrounded by 1-sided linedefs. If the sector was in void itself, nothing is left.

In Things mode, the current Thing is deleted.

see_also

Vertexes

LineDefs
Sectors
Things
Edit
=
Merge
=
 Merges stuff.
see_also
Edit
=
Find
=
 Locates the given object by number.
see_also
next object
previous object
Edit
=
Next object
=
 Locate the next object (by number) if it exists.
see_also
previous object
find
Edit
=
Previous object
=
 Locate the previous object (by number) if it exists.
see_also
next object
find
Edit
=
Tag line to Sector
=
 Associates a LineDef to a Sector. This is needed for most LineDef
actions.
 The LineDef that holds a switch, for example, could be tagged to a
sector that lowers its floor.
see_also
Edit
=
Map Utilities
=
 Allows map-wide editing of the map.
 Shifting the map (or origin), adjusting the map size and changing
the light levels can be done using Map``utilities.
see_also
Shift map (X/Y/Z)
Expand/reduce map
Light adjustment
=
Shift Map (X/Y/Z)
=
 Shift map allows each Vertex and Thing object to be shifted on

their X and Y coordinates. Sector floors and ceilings may also be shifted up or down (Z). Positive values shift right/east for X, north for Y, and up for Z.

Centering the map will enter values to move the origin to the midpoint of the extreme vertices (not the mean average). After pressing the center button, the values may be changed before shifting.

see_also

Map Utilities

=

Expand/reduce map

=

Expanding (and reducing) the map is done by changing the X,Y, and Z percent values. Values above 100% expand the map on that axis, below 100% reduce, and a value of 100% will remain unchanged.

see_also

Map Utilities

=

Light adjustment

=

The overall light intensity for the entire map may be changed using the Map light adjustment panel. Two variables determine what each new sector light value will be. A (amplify) is a percent-value which is multiplied by the old light value, and B (brighten) is added to this product. The formula is:

$$\text{New``Light}``=`Old``Light` x A/100 + B.$$

If A is 100% and B is zero, no light is changed.

see_also

Map Utilities

=

Sectors

=

Sectors are homogeneous areas on the map where DOOM is played. Each sector may have only one floor and one ceiling. Any space not in a sector is void and unaccessable. A single sector may be a simple room, but generally rooms are more than one. Sectors may also be inside other sectors.

The sectors menu provides control for creating and editing sectors, it also controls sector styles.

see_also

Object basics

Edit styles

Texture style

Align textures

Polygon

Rotate

Size

=

Edit styles

=

The sector styles list may be edited in two ways: either adding/deleting records, or by rearranging existing records.

Adding or deleting style records is done after selecting a one-sided LineDef. The Edit sector styles panel then displays the style associated with that LineDef so it may be added to the list. Existing records may also be deleted from this panel, but a LineDef must still be selected to bring up the panel.

Rearranging style records must be done in sectors mode. A list of all existing styles will appear, and the chosen style will be moved from its current place. Again the list will be displayed, and the chosen slot will be the destination for the first record. The two styles will swap places in the styles list. The first record will always be the default style when EdMap starts up.

If there are no records in the styles file (or the file does not exist), a new file will be created when a style is added.

see_also

Sectors

=

Texture style

=

This function retextures the ceiling, floor, and all walls of the current sector with the sector style.

see_also

Align textures

Sectors

=

align textures

=

Align texture automatically adjusts the x-offset of adjacent sidedefs whose textures are the same to make the wall texture appear as one. Use in either linedef or sectors mode.

If done in sectors mode this will adjust all wall textures facing the sector. However, since every texture may not be correctable (as in a round room for example), this may leave "seams" where the last texture has already been aligned.

If used in linedef mode, "seams" can be placed. In linedef mode the current (one-sided) line texture is used as the starting point, whose x-offset is 0. The next texture to the right is adjust accordingly, and so on, until the first texture is reached again or a new texture is used. This way the "seam" (if any) is always on the left of the initial texture.

Note that in addition to testing the main texture, both the upper/above and lower/below textures will be checked for a texture match in order to continue aligning.

see_also

Sectors

=

polygon

=

After selecting polygon, the center of the polygon must be placed and the polygon panel will appear. The number of sides and the radius may then be entered. The number of sides can be no less than 3, and no more than 40. The number of sides should be as few as possible; more sides take resources: (size of the map is larger, DOOM may run slower, etc). The radius of the polygon is measured from the center to each vertex, not to the LineDefs.

see_also

Sectors

=

rotate

=

Sector rotation and resizing are handled similar to a normal drag. Once sector rotation is enabled, "rotate" will appear in the message window. A sector may be picked to rotate just as it would be picked up

for dragging. The mouse will then snap to the lower right corner of the sector. The sector will pivot on its midpoint, following the mouse until it is dropped, just as dragging. A panel will verify that the change is to be kept. If rotation is enabled accidentally, selecting sector rotation again will return to normal drag. Some Thing objects may also be rotated.

see_also
size
Sectors
=
size
=

Sector resizing and rotating are both handled similar to a normal drag. "Resize" appears in the message window when sector resizing is enabled. Just as dragging, the sector to be changed must be picked up, edited (by moving the mouse), and dropped again. When a sector is picked up for resizing, the mouse will snap to the lower right corner (as in rotating), and the sector will fit in the imaginary box whose lower right corner follows the mouse. Click to drop, and verify that the change is correct. The sector will not flip (to be mirror-image or upside-down), if the mouse passes the center of the sector, instead another corner of the box will follow the mouse. If resizing is enabled accidentally, selecting sector resizing again will return to normal drag.

see_also
rotate
Sectors
=
Automatic
=

The Automatic menu can be used to convert or create complex structures on the map. All changes made in any of the panels are saved to the config file.

see_also
door
stairs
lift
=
Door
=

The Door feature converts the current sector into a door using specifications in the Door panel. All textures and other options in the Door panel are saved.

Door texture is what the door will look like when closed. Sill texture is what the wall that the door slides up on will look like. The bottom texture is the texture on the bottom of the door when it is open. Key specifies what key, if any, is needed to open the door. Doors will 6 seconds after it opening unless it is set to stay open. A moving sill will rise with the door when it opens.

see_also
automatic
=
stairs
=

The Stairs option creates a new set of sectors that may either be a staircase or a floor that when triggered rises into a staircase. After accepting the Stairs panel, place the first step of the staircase.

If created within an existing sector, only the floor will be changed; the ceiling will appear to be the same. If made in the void, the current sector style will be used for data not specified in the Stairs panel.

The panel specifies textures, sizes, and other attributes needed to construct the staircase. Step top and step side textures modify only the steps; walls are taken from the style. Rise, depth and width determine size of each step. Ceiling height (if created in void) is this value above the floor of the highest step. The staircase may go up, right, down or east.

There are special considerations for staircases that rise out of the floor. First: each step must rise to 8 high. Second: Modifying the shape of the steps can disable its operation. Experiment; sometimes it will work, but it cannot be guaranteed.

see_also
automatic

=
lift
=

Not implemented yet.

What? Can't you make a lift youreself?? *hehe..*

see_also
automatic

=
display
=

The display menu offers control over the normal viewing screen. Additional information may be displayed or changed from this menu.

see_also
Enhance map
Full screen
Snap/grid
Grid on/off
Origin on/off
Center map

=
Enhance map
=

While in enhanced map mode, the normal map drawing routines will shift sectors according to altitude, and draw some lines for walls, for a 3-dimensional effect. Many operations are simplified in this view; breaks in one-sided walls are drawn easy to see, and merging sectors is easier since the wall disappears when the sectors are connected. Redrawing the map in enhanced view is not as fast as normal view because of the extra lines and calculations; this is may be noticeable when panning the map. The shifting is not proportional. It is always fixed so when the map is zoomed in, the walls do not extend off the screen, and so it is easier to determine between walls and floors.

see_also
Display

=
Full screen
=

Full screen display removes the menu and the information bar, leaving the entire screen to displaying the map. The menu may still be used with the keyboard. The object information bar will either still edit objects or it will be locked to prevent accidental changes.

Whether or not it is locked is set using EDMAPCFG. Switching between full and normal views can be done using the mouse pad, the menus: "D" (Display) "F" (Full screen), or the hot-key: Ctrl-S.

see_also
mouse pad
Display

=

Snap/grid

=

Using the Grid/snap sizes panel, the blue dot grid and the drag snap grids may be adjusted.

The grid is a grid of blue dots draw on the map to help align objects. If the map is zoomed out too far so the grid dots become too dense, the grid will not be drawn. (This also means that a grid of 2 points may never be drawn.) The grid may also be turned off using Grid``on/off.

The snap grid is used when dragging any object. All vertices of the current object (or Thing) will snap to the closest point on the snap grid while moving. This helps with merging since dragging need only be as accurate as the size of the snap grid. Although the minimum size is 2 points, it is generally best to keep the snap grid between 4 (or 8) and 32 points. Too low and merging vertices becomes difficult, too high and new sectors snap into oblivion. The snap grid is unrelated to the normal dot grid.

see_also
Grid on/off
Display

=

Grid on/off

=

The normal dot grid may be toggled on or off with this function. The screen refresh may be slightly faster and a small amount of memory is freed when the grid is off.

see_also
Snap/grid
Display
Origin on/off

=

Toggles drawing the origin (0,0) on the map. Will not noticeably affect performance or memory.

see_also
Snap/grid
Grid on/off
Display

=

Center map

=

Centers the map on the mean average of all vertices.

see_also
Display
Check

=

EdMap can test for several errors that may make the map less playable.

Transparent textures, multi-patch textures, hanging textures,

short textures, short lines, long lines, missing tags, missing things, thing heights, and map exits can all be checked.

see_also

Error list

Quick check

Check all

Textures

Associations

Heights/widths

LineDefs

Begin & end

=

Error list

=

Compiles a list of all errors found on the map by calling the individual checks. Warnings may be disabled using the Preferences panel.

see_also

Preferences

Check

=

Quick check

=

Quick check browses over the map, but does not check for every possible error.

see_also

Check

=

Check all

=

Checks the entire map for errors.

After each error, the check may be stopped or resumed to find the next error, if any.

see_also

Check

=

Textures

=

Checks textures:

Hall of Mirrors/Missing textures, if a texture is used, the texture name will be checked to see that it exists in DOOM.

Medusa effect: a multi-patch texture on a 2S (2-sided) main wall.

Tutti-Fruitti effect: (1) short texture on main, cannot be tiled evenly, (2) transparent textures on above/below surfaces.

see_also

Check

=

Associations

=

Tests that all LineDefs that activate sectors and need a trigger are tagged to at least one sector. Also tests for appropriate teleporter destinations.

see_also

Check

=

Heights/widths

=

Tests Thing objects (mosters, players, teleporter destinations, etc.) that must be in a sector at least as tall as they are.

Widths are not tested.

see_also

Check

=

LineDefs

=

LineDefs are walls that bound sectors and are bound by vertices. Each LineDef has one or two SideDefs, one for each side. Two-sided LineDefs (2S) separate two sectors, while one-sided (1S) lines border void and one sector.

The LineDef check tests LineDefs:

Long Wall Error (LWE): LineDefs that are very long (1000 units) can cause problems in the DOOM engine. Split them into shorter lines to prevent any problem. This is a warning, not an error.

Short/zero-length lines: walls too close together can cause problems. zero-length is obviously an error and should never occur, however since the snap grid pulls lines around on its own, sometimes two vertices can occupy the same point (but this is rare).

=

Level begin & end

=

Tests for all start positions on the map. Start 1-4 and at least 4 deathmatch starts should be included. Also tests for at least one exit.

see_also

Check

=

Object Info bar

=

The object information bar displays data for the current object and allows editing this data.

To change a field, press the key associated with that line. For example, press "7" to change the direction of the current thing object. Not all fields are editable; the "two-sided" bit for linedefs is handled automatically and cannot be edited.

see_also

editing objects

=

Editing objects

=

There are two parts to editing most objects: the location of the object and the data about the object.

Changing the location of an object can be done simply by dragging the object: click the first mouse button when the desired object is highlighted and the object will be "picked up" for the mouse to move. Click again to drop the object. Sectors also may be rotated and resized.

To edit the object data use the object info bar. Press the key beside the line of the field to be changed.

see_also

object info bar

=

mouse control

=

Many functions are built into the mouse.

First-button commands: (object manipulation)

Tap the first button to pick-up and object for moving. Tap again to drop the object where it is.

Hold the first button and move the mouse to use the mouse-pad.

Hold the first button and tap the second to insert an object.

Different modes use insert differently.

Second-button commands: (display control)

Hold the second button and move the mouse to pan across the map.

Panning control is not proportional: at far zooms, the control is less sensitive.

While holding the second button press and hold the first button.

This enables zooming adjustment. Push the mouse forward to zoom in and pull back to zoom out. Be sure to hold the second button first since pressing the first button, then the second is add/split.

see_also

mouse pad

add/split

=

choose an angle

=

Select the direction the Thing is to face by moving the mouse cursor relative to the center of this panel. Click to accept, ESC aborts.

=

choose a thing

=

All of the Thing objects in DOOM are displayed on this panel.

Choose by clicking the mouse on the desired type or press ESC to abort.

=

choose sector style

=

Any of the styles in the sector styles file may be chosen from this panel.

see_also

edit sector styles

=

choose sector type

=

Each sector may have one of several special properties, as displayed on this panel. Flickering lights and damaging floors are examples of sector types.

=

choose an action

=

LineDefs can trigger sectors to perform actions, such as raising a ceiling (door), lowering a floor (lift), or even teleporting.

Actions that are repeatable (marked by an "r") may be triggered more than once, while others may only work a single time.

Switch-operated actions (marked by a "s") can only be activated by walking in front of the LineDef and pressing the USE key (as when opening doors). Most other actions are triggered by walking over the LineDef (and a couple must be shot).

Almost all actions must be tagged to a Sector (mark the line, press F7, click on the sector). Some actions, however, do not need a trigger number. "MDoor"/manual doors (except for the shoot-to-open MDoor #46), exits and scrolling walls do not need to be tagged.

=

choose wall texture

=

All of the wall textures in DOOM are displayed and may be chosen from this panel.

The desired texture may be chosen by either clicking texture name on the panel or by using the viewer.

To activate the viewer press F10. There should be 90k free memory for the viewer to load. Select the displayed wall texture by pressing Enter.

Pressing ESC from either screen will abort.

see_also

viewer

=

choose floor/ceiling texture

=

All of the floor (ceiling) textures in DOOM are displayed and may be chosen from this panel.

The desired texture may be chosen by either clicking texture name on the panel or by using the viewer.

To activate the viewer press F10. There should be 90k free memory for the viewer to load. Select the displayed floor texture by pressing Enter.

Pressing ESC from either screen will abort.

see_also

viewer

=

viewer

=

The viewer displays wall, floor (ceiling), sprite, and patch graphics exactly as they would appear in DOOM.

Wall graphics are the textures used on every wall in DOOM. Some textures are partially transparent and some are animated. Transparent graphics may only be used on the main texture for 2-sided linedefs, where a texture is unnecessary.

Every sector must have two floor graphics: one on the floor and one on the ceiling. Some floor textures are also animated and F_SKY appears transparent by showing scenery.

Sprites are any graphic images of Things that are displayed within a room in DOOM. Monsters, weapons, obstacles, etc. are all sprite graphics.

Patches are used only to assemble wall graphics; Each wall texture is an arrangement patches. Multi-patched walls cannot be used as the main texture for a 2-sided linedef.

Switch between display modes by pressing left and right, choose textures by pressing up/down/PgUp/PgDn or by pressing the first letter of the texture name. Press Enter to select, ESC to abort.

=

Sidedef panel

=

The SideDef panel displays information about the selected SideDef. This panel can be used to adjust the X and Y offsets of the textures it uses. In Addition to the numbers of the associated objects (supporting LineDef, SideDef and facing Sector numbers), this panel also indicates the size of each texture surface.

=

Things

=

Thing objects are monsters, players, weapons, ammunition, corpses, pillars, lamps, spikes, etc; anything that is not part of the walls or floors in DOOM is a Thing object.

see_also

Vertexes

LineDefs

Sectors

=

Vertexes

=

Vertices mark the endpoint of every line on the map. Move them to alter the shape of the room.

see_also

LineDefs

Sectors

Things

=

PWADs

=

PWADs, or "Patch-WADs" are the files DOOM and EdMap use to store map data. Each PWAD (*.WAD) file may contain several maps and other resources like sound effects and graphics. PWADs supplement the IWAD ("Internal-WAD"), DOOM.WAD, which contains all the resources necessary for normal game play. When a PWAD is added, the resources contained in the PWAD update those in the IWAD. Each time a PWAD is added to the list, some number of resources from the existing list are outdated, sometimes outdating an entire PWAD file.

see_also

List wads

=

EDMAPCFG

=

EDMAPCFG <config file>

EDMAPCFG.EXE is the configuration program used to create a new config file. If no filename is given, EDMAPCFG will use the default config file. If the current file does not exist, a new file will be created.

=

Sector in sector

=

If a sector is created within another sector, note that that sector cannot be moved out of its surrounding sector. Nor can a sector created in the void be moved inside an existing sector.

Of course, if a sector is created within another, the existing sector must be large enough to surround the new 64x64 sector.

=

Delete sector

=

Deleting a sector will erase that sector and all objects dependant on it (except Things, which are not associated with sectors).

=

Overwrite on save

=

Overwriting a file will erase all data the file currently holds and replace it with the data to be saved, no matter what the original file contains.

=

xxx/this should be at the end of the file