L3DEdit Manual

Written for L3DEdit V0.48

Welcome to L3DEdit!

L3DEdit is a level and graphics editor for Lemmings 3D. You can use it both for the modification of the official levels, and for the creation of entirely new levels. You can also extract and import many Lemmings 3D graphic files with it.

If you’ve worked with tilemap-based editors for 2D games before, you already know some of the basic concepts behind how L3DEdit works – the biggest difference is that our tilemap is 3D.

There are three versions of L3DEdit available. I recommend using the Windows Delphi version of L3DEdit (under WINE, if you’re using Mac or Linux); as the Lazarus version (on any OS) is laggy when rendering, and the Linux build is also mostly untested.

Please note that L3DEdit is only guaranteed to work with the full PC version of Lemmings 3D (either the CD or non-CD version).

L3DEdit is open-source software, under the MIT License.
Source code: https://bitbucket.org/namida42/l3dedit/src/master/

Warning: Images in this document may at times, be from older versions of L3DEdit. If something appears to be missing in the images, or something is present that you can’t find in your copy of the editor, this might be the reason. Please let me know if you discover any situations where the use of old images makes it difficult to understand this document. There are also minor visual differences between the three versions; this document’s screenshots come from the Delphi version.

Getting Started

You will need a copy of the full PC version of Lemmings 3D in order to use L3DEdit. You'll also, of course, need to be able to run Lemmings 3D – which can itself be tricky on modern systems. Getting Lemmings 3D running in DOSBox (the easiest way to do so) is beyond the scope of this guide, but there is a topic on Lemmings Forums with pointers:

https://www.lemmingsforums.net/index.php?topic=4289.0

You will need to place L3DEdit in the same folder as Lemmings 3D. If you're using the CD version of Lemmings 3D (whether with a real disc or a ripped ISO image), you will need to copy all the files to a location on your local hard drive.

It is possible to use the Lemmings 3D Winterland graphics with the full version of Lemmings 3D (and thus, with L3DEdit) in custom levels. If you wish to do so, copy the following files from Winterland's GFX folder to L3D's GFX folder:

- ANIMOBJ.031
- LAND.031
- OBJ.031
- OBJ.056
- SEA.056
- SIGNS.031
- SKY.031
- TEXTURE.014
- TEXTURE.024
- TEXTURE.031
- TEXTURE.056

You will notice that other files of these types exist, but those are identical between Lemmings 3D and Lemmings 3D Winterland. The only graphic file **with a number as the extension** that exists in both, but is not identical, is BGRD.000 (the main menu background).

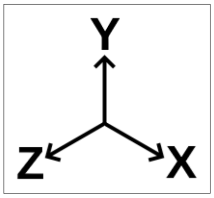
That's all – you should be ready to go. The first time you run L3DEdit, you will be prompted to use another tool called L3DUtils to apply a music patch (unless L3DEdit detects you've already done so). This fixes inconsistencies Lemmings 3D has between the music selection with CD audio, versus the music selection with other types of audio. **L3DEdit's music-related options when creating levels assume you have applied this patch!**

In terms of system requirements, L3DEdit should run on just about any PC with a suitable operating system (or a suitable ~~emulator~~ compatibility layer), although weaker systems will need to use lower rendering resolutions. You will also need to use lower resolutions on otherwise-equal setups to get the same performance when using the non-Delphi Windows build or the Linux build, compared to the Delphi Windows build. You'll also need a screen resolution of at least 1280x720 (16:9) / 1024x768 (4:3) – and higher is better.

Structure of a Lemmings 3D Level

In order to properly learn how to use L3DEdit, you must have a basic understanding of how a Lemmings 3D level is structured. This doesn't mean you need to know what each byte in the level file means (or even what a "byte" is), just that you need to know in general what components a level's layout is made up of.

A Lemmings 3D level is mostly made out of blocks, in a similar fashion to a tilemap – except it's 3D. The grid is 32 blocks wide, 32 blocks deep, and 16 blocks tall. The graphic next to this paragraph shows the arrangement of the axes, with the point at which they meet being the origin (0, 0, 0).



Each space may contain one block and/or one object. It is important to note that Lemmings 3D's definition of an "object" does not line up completely with the definition in many of the 2D Lemmings games.

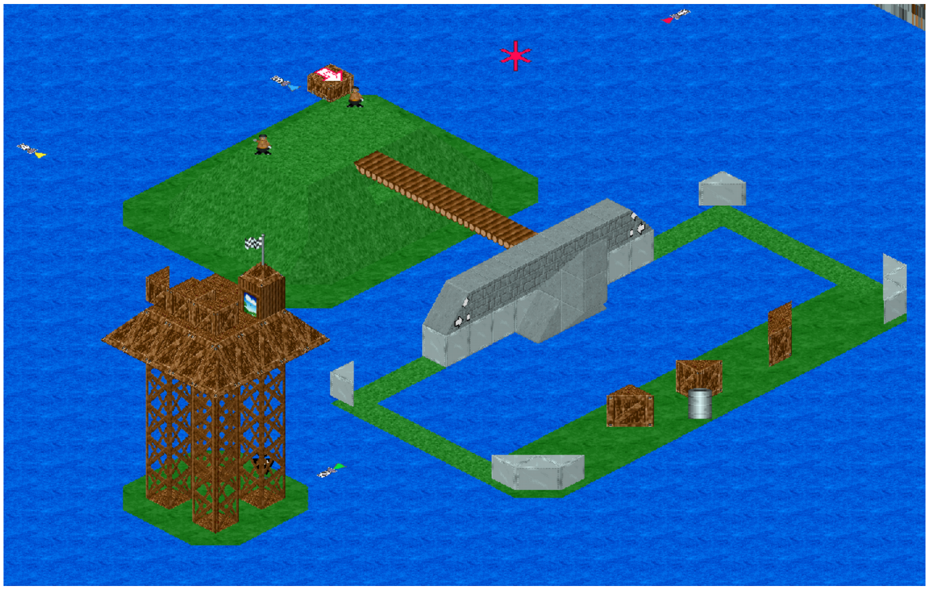
In addition to these, a Lemmings 3D level also contains land areas. These are large areas at the bottom of the level, which act as solid, indestructible ground.

Levels also have four cameras, and a "pivot point" for the preview screen. The level can also specify graphics for the water (if any) at the bottom of the level, and for the background (sky) in the distance.

Coming back to blocks, as well as the actual layout of blocks, levels also contain "metablock" data. If the layout is a tilemap, you can think of the metablock data as the tileset being used. Lemmings 3D does not have predefined global tilesets, each level specifies a graphic file (which contains up to 100 possible graphics for block faces) and creates its own tileset using graphics from that file. This is in fact why there are two files for each level – the “BLK.xxx” file contains the metablock data, and the “LEVEL.xxx” file contains everything else.

L3DEdit allows you to either create your own metablock info for your levels, or to import the info from an existing level. Importing from an existing level is generally much faster and easier, but as you become more experienced, you may prefer to create your own in most cases. It’s also completely possible to import another level’s metablock info as a starting point, then tweak it yourself.

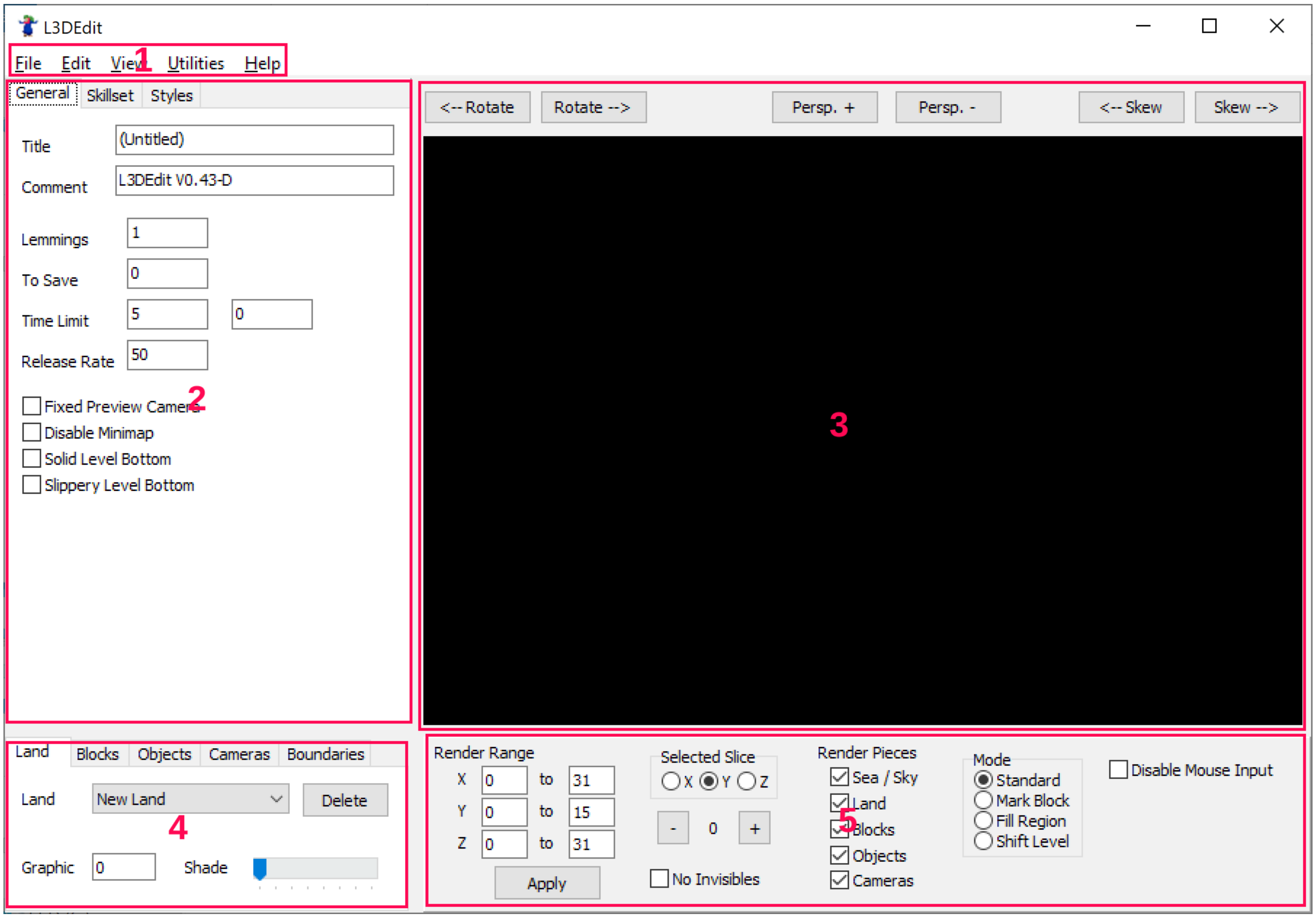
It is important to note that L3DEdit does not, however, edit the two files separately. The “LEVEL.xxx” and “BLK.xxx” files are loaded / saved as a pair, not as two individually-editable files.



In this image, we can see many components of a Lemmings 3D Level. The grass regions at sea-level are examples of land areas. The army-gear lemmings near the entrance, the barrel near the bottom-right, and the flag on the exit, are objects. Most other components are blocks, including the entrance, exit, and one-way walls.

The Basics

Upon starting L3DEdit, you’ll see the following screen.



Title (Untitled)
Comment L3DEdit V0.43-D
Lemmings 1
To Save 0
Time Limit 5 0
Release Rate 50
□ Fixed Preview Camera
□ Disable Minimap
□ Solid Level Bottom
□ Slippery Level Bottom

Land New Land
Graphic 0
Shade

Render Range
X 0 to 31
Y 0 to 15
Z 0 to 31

Selected Slice ○ X ● Y ○ Z
0
□ No Invisibles

Render Pieces
☑ Sea / Sky
☑ Land
☑ Blocks
☑ Objects
☑ Cameras

Mode
● Standard
○ Mark Block
○ Fill Region
○ Shift Level

□ Disable Mouse Input

1 – Menu Bar
2 – Level Metadata Panel
3 – Level View
4 – Level Component Panel
5 – Controls / View Options Panel

You can resize this screen. This will resize the Level View region, while all other segments will remain the same size (but may move).

Menu Bar

I’ll assume you already know how a menu bar works in general, and won’t explain that. However, here’s what each item in the L3DEdit menu does:

[File → New] [File → Open] [File → Save] [File → Save As]

These are all self-explanatory. One thing to note is that when saving, only the file extension matters. Whether you type “BLK.000”, “LEVEL.000” or “HELLO.000” - or even just “.000” - your level will get saved to the same files; the block data in BLK.000 and the level data in LEVEL.000.

[File → Import Block Data]

This option lets you select another level to import the metablock data from. For sanity reasons, this also copies the Texture Set (but not any other style-related settings!) from the source level.

[File → Quit]

Self-explanatory.

[Edit → Undo] [Edit → Redo]

More common, self-explanatory functions. Do note however, that L3DEdit does not yet disable these options when there’s nothing to undo / nothing to redo – it will simply do nothing when you try to use them.

[View → Zoom In] [View → Zoom Out]

Self-explanatory.

[View → Grid Style]

Allows you to change the style used to display the grid in the Level View area. Options are:

Standard – Displays an outline of the top face.
Corners – Displays dots at the corners of the top face.
None – Does not display any grid.

[View → Resolution]

Allows you to configure the resolution of the level render. Turning this down will improve rendering speed, but at the cost of graphical quality.

[Utilities → RNC Compress] [Utilities → RNC Decompress]

This allows you to compress or decompress RNC compressed files, which is commonly used by Lemmings 3D. L3DEdit will generally decompress files as needed, but this can result in slowdowns, so it's better to decompress the files up-front. In particular, note that Lemmings 3D does not care whether files are compressed or not, it will accept them either way. You can either compress / decompress a single file, or one-click compress / decompress all known compressed files.

[Utilities → Graphic Files]

There are options here to extract and import graphics to / from Lemmings 3D's graphic files. See the notes later in this document regarding how to use these options.

[Utilities → Reorder Levels]

This lets you quickly reorder or delete levels (or more accurately – replace levels with empty placeholders, as Lemmings 3D doesn’t like it if a level file is outright missing). You can also use this simply to see a full list of the levels.

Use of this menu should be all self-explanatory. Remember that you can hold Ctrl or Shift while clicking on an item in the list, to select multiple items.

The Style Count button will count all levels if none are selected, otherwise it will count the selected levels only. The numbers next to each style indicate the music usage – **S**tandard, **A**lternate and **R**emix.

WARNING: If you have a level open in L3DEdit, then change its position in the level order using this menu, the filename used when clicking “Save” is **NOT** automatically updated – so make sure to use Save As and save it to the correct new filename afterwards, otherwise you might overwrite the wrong level!

[Help → About]

You can use this to check which version of L3DEdit you’re using. If the version number has “-D” at the end of it, this indicates it’s the Delphi build of the corresponding version.

Level Metadata Panel

The level metadata panel lets you customize many of the global details of your level. This includes things like the title, the skillset, the save requirement, which graphic files it uses, etc.

Many of the options on it are very obvious, so I will only explain the ones that aren't, or where there are "gotchas" to be aware of.

General Tab

Level Title – The title can be up to 31 characters long. Note that certain characters might not work correctly in-game.

Comment – The comment can also be up to 31 characters long. It is not displayed anywhere in game. In the official levels, it always contains the level creator's name.

Time Limit – The maximum allowed is 99 minutes and 99 seconds. (Yes, 99 seconds, not 59 – Lemmings 3D has no problem with >59 seconds, as long as it's still two digits.)

Fixed Preview Camera – If this option is enabled, then previewing the level will simply show Camera 1's initial view, instead of a rotating view around the level. (This is used on indoor levels like Alilemms, Lemlab, etc.)

Disable Quick Jump – This disables the ability to quickly move the camera anywhere by click-dragging on the minimap (or in the area where the minimap would be, if the level disables it).

Solid / Slippery Level Bottom – These turn the bottom of the level (that would otherwise be sea) into solid land or slippery land, respectively. Note that these have no effect on the graphic; you must choose a sea graphic that looks like land / ice too. This only applies to sea area; anywhere covered by land area is not affected.
‘

Skillset Tab

Everything on this tab is self-explanatory.

Styles Tab

Note: Some of the options here correspond to graphic files, which are identified by an index number. The dropdown box next to these fields, which contains a list of names, is generally a more convenient way to make your selections. If you wish to modify these presets, for example when using custom graphics, modify the L3DEditPresets.ini file (created the first time you run L3DEdit). The default presets file includes the Winterland graphics, although they will only show up if you have a copy of them present.

Theme – This determines the preview wallpaper that comes up before a level, as well as which musics can be selected from for the level.

Music – Note that the options here assume that either (a) the music patch has applied, and/or (b) CD music is being used. If you’re using soundcard music without the music patch, go apply the damn music patch. If you really don’t want to, then keep in mind that under soundcard music, no-music-patch conditions, “Alternate” and “Standard (Remix)” will be swapped (and you’ll also get wrong

music on most levels when you try to play anyone else’s content). Also note that CD music doesn’t *have* a “Standard (Remix)”, and will just play Standard on levels that specify this.
Texture Set – This specifies the block face graphics that can be used in the level to construct the metablocks.

Land – This specifies the graphic file used for land areas. Most of these files contain a single land texture, but some contain multiple textures.

Objects – This specifices the graphic file used for Static Object type objects. Each file contains up to 20 static object graphics.

Signs – This specifies the graphic file used for Sign type objects. Each file contains up to 8 sign object graphics, which have both front and back versions.

Sea – This specifies the graphic file used for the water at the bottom of levels. There are two different sizes, and those of the smaller size can be animated.

Animation – This specifies the graphic file used for the Animated Object type objects.

Trap – This specifies the type of interactive object (not always a trap, per se) used by the level – ie: flamethrower trap, squasher trap, bear trap, laser trap, “weird trap” (I couldn’t think of a good name for it), trampolines, springs, teleporters and rope slides.

Sky – This specifies the sky graphic used in the level’s distant background. There are two different sizes; on levels that use a full-height sky graphic, the sea will not be drawn.

Walls – This specifies the graphic file used for the Wall type objects. Each file contains up to 16 wall object graphics, although no official data file contains more than 8.

Non-Animating Sea – If you’re using a smaller-sized sea graphic that can animate, this option will override that and make it not animate. It has no effect when using a large-sized sea graphic.

Large Sea Texture – Specifies that the sea graphic being used is a large-size one. If this option is set incorrectly, graphical glitches will likely result. When selecting a sea graphic via the preset dropdown, this option will automatically be set correctly, but it is up to you to set it correctly when using custom graphics.

Single-Graphic Land Texture – Specifies that the land graphic being used contains only a single texture. Like with the Large Sea Texture option, setting this incorrectly can result in graphical glitches, and selecting via the presets will automatically set this option correctly.

Full-Height Sky Texture – Specifies that the sky graphic being used should take up the full screen height (and thus sea should not be drawn). This can only be used with sky graphics specifically designed for it; using this option on other graphics (or not using it on ones that *were* made for it) will, again, result in graphical glitches. And yep, again, using the preset dropdowns will set this to the correct setting automatically.

> **Note:** This option only stops sea from being *drawn*. Lemmings will still drown if they touch the level’s bottom anywhere that there isn’t a block or a land area for them to stand on (unless the Solid Level Bottom option is enabled, of course).

Zapping Replaces Drowning – If this is enabled, lemmings will never drown on the level – any contact with sea, or with blocks that have the “Liquid” flag enabled, will cause them to be zapped instead of drowning. This still kills them, just with different visuals / sound.

Layout Editing – Introduction

The remaining three sections of the editor view – the Level View, the Level Components Panel and the Controls / View Options Panel – all work together, and cannot really be explained independently of each other.

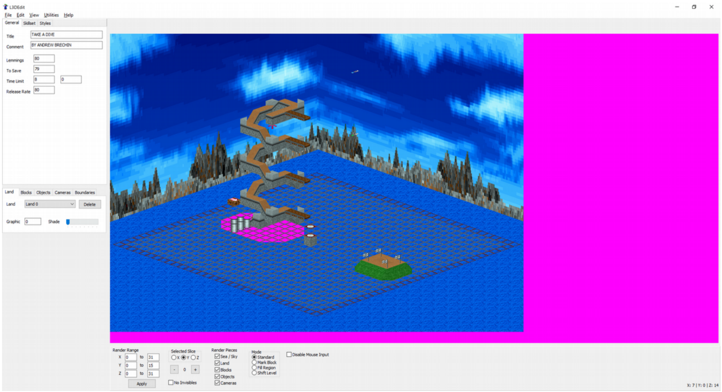
As such, I will instead divide this into several parts, starting with some of the really basic features and moving on to the more complex features.

For some of the early parts, you will not yet have learnt enough to make your own levels, so you will instead be modifying / examining levels from the official game. In these cases, I will mention certain levels that I feel are well-suited. However, I completely encourage you to experiment with other levels, beyond what's explained in these sections – the best way to learn is by doing!

Layout Editing – View Control

To start off with, let’s get used to how to control what parts of the level we can see. I mean “parts” in several senses here – we can scroll the section of level that’s displayed on-screen, we can select to only render certain parts of a the level (perhaps we only want to render the bottom half of the level), and we can select to only render certain components (maybe we want to hide everything except for blocks and land).

It won’t be very interesting to scroll or slice a black screen, so let’s load a level. Use the menu bar to open “Take A Dive” - this is the first file, LEVEL.000 or BLK.000 (you can open either file and it will work; L3DEdit will find the other one by itself as long as it’s in the same folder – which it should be if you’re using actual L3D data files).



Title: TAKE A DIVE
Comment: BY ANDREW BRECHIN
Lemmings: 80
To Save: 79
Time Limit: 8 0
Release Rate: 80

Land: Land 0
Graphic: 0
Shade

Render Range
X 0 to 31
Y 0 to 15
Z 0 to 31

Selected Slice: ○ X ● Y ○ Z
- 0 +
□ No Invisibles

Render Pieces
☑ Sea / Sky
☑ Land
☑ Blocks
☑ Objects
☑ Cameras

Mode
● Standard
○ Mark Block
○ Fill Region
○ Shift Level

□ Disable Mouse Input

X: 7 | Y: 0 | Z: 14

Tip: Unlike some of the older games, Lemmings 3D’s levels are stored almost exactly in the in-game order. There are two exceptions – firstly, the Practice levels are stored at the end (LEVEL.080 to LEVEL.099), and secondly, the Practice levels for Virtual Lemming and Highlight Lemming are swapped in the data files compared to their positions in the Practice menu. However, do also note that the level data files start at 0, whereas the in-game numbering starts at 1.

We’ll start by zooming in and out. There are three ways to zoom – you can use the options under View in the menu bar; the shortcut keys for those options; or the scroll wheel on your mouse. Note that L3DEdit does not have a particularly smart zoom – it won’t eg. center on the mouse position, so you will likely need to scroll after zooming in or out.

Tip: If zooming with the mouse wheel isn’t working, it may be because “focus” is somewhere other than the level view (perhaps because you’ve clicked on one of the text input boxes, so that input box has focus instead). To put focus on the level view area, **without** any chance of changing anything about the level, hold Shift and click on the level view image.

So how can we scroll? To scroll the view area, hold the middle mouse button and move the mouse. Most external mice have a middle button, which is used by pressing down on the scroll wheel. Some laptop touchpads can be configured to have a middle button, while others will treat pressing both buttons at once as a middle-click.

However, if you don’t have any means of clicking the middle mouse button, there is another way. In the Control / View Options Panel, there is a checkbox labelled “Disable Mouse Input”. When this box is checked, the normal functions of the left and right mouse buttons on the level view area are disabled, and they instead work the same as the middle button – ie: click and hold with them to scroll.

We can also rotate the level view. We don’t have the full 360 degrees of freedom like we have in-game in Lemmings 3D, but we can rotate it in 90 degree increments by clicking the “Rotate” buttons above the level image. How about “Skew” and “Perspective”? Try them out!

Next, look at the “Render Pieces” checkboxes in the Control / View Options Panel. These allow you to select which types of level components do or don’t get rendered. These aren’t useful that often, but it’s still nice to know where they are if you need them.

More important are the “Render Range” options. These let you select the range of coordinates that will be rendered. Note that these work relative to your current point of view, not relative to the raw level data. Try setting the maximum X to 15 (you need to click “Apply” afterwards for it to take effect), then rotating the level view, and observe what happens.

Wait a sec… You’ve probably noticed by this point, when viewed in the level editor, Take A Dive has a second trampoline floating in the air above the first one! This is not a bug in the editor, but rather an oddity in the level data itself – and there are many cases like this in the official Lemmings 3D levels. I will explain later why this happens.

The last option we’ll look at in this part is the “No Invisibles” option. Enabling this option reveals invisible block faces (which can sometimes be every face of a block, leading to an entirely invisible block) in the level. When enabled on Take A Dive, you’ll notice that invisible blocks appear overlapping the flags near the exit, as well as the barrels and the army guys near the entrance.

This is because these are “objects”, which aren’t inherently solid in Lemmings 3D. They’re solid because of the invisible block placed at the same location, nothing else – without that block, lemmings walk right through them. While yes, just placing invisible blocks with no indicators is extremely unfair and bad-quality level design, Lemmings 3D has several legitimate use cases for them, and this is the most common one.

Once you’re done experimenting with the different views on Take a Dive, open up a few other levels and take a look at those. You might see some things that you find particularly interesting when experimenting with the various view options, on “Alpine Assault Course” (LEVEL.011) and “Ski Jump” (LEVEL.063), as well as the indoor levels such as “Alilemms” (LEVEL.019) or “Lemlab” (LEVEL.035). You can probably figure out for yourself what’s going on in the case of the ice levels; I will explain later why the indoor levels are the way they are.

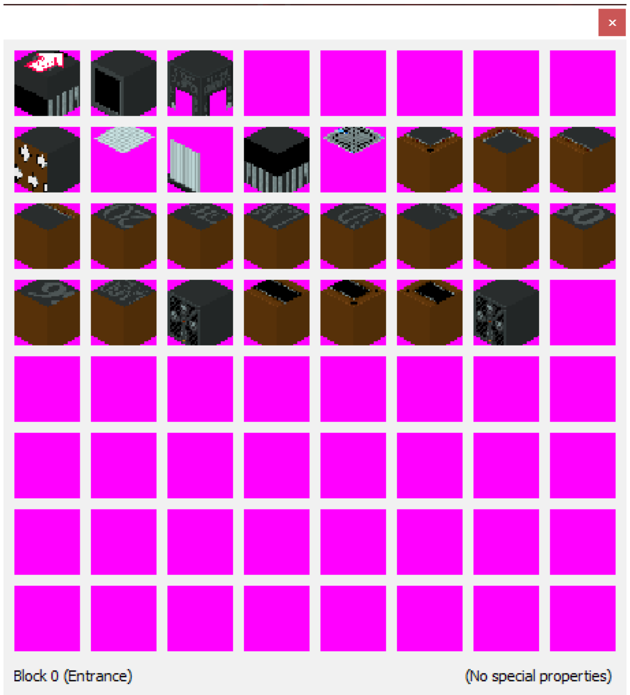
Tip: You probably noticed that one land area in each level (except levels that have none) is rendering solid magenta, instead of with its proper graphic. In the Level Components Panel, change to a different tab (other than land) and all lands will render normally. The magenta render is used to indicate the currently-being-edited land, and since Land is the default tab...

Layout Editing – Blocks

Generally speaking, the most important component of a level is the blocks. It is possible to create a fully-functional (and even quite complex) level completely out of blocks, without using any land areas or objects. This is possible because in Lemmings 3D, several things that are considered objects in other Lemmings games, are considered special types of blocks here.

As mentioned earlier, blocks are placed in a 32x16x32 grid, and each level defines its own “tileset” of blocks, with external files providing only the graphics, not the actual arrangements or physics properties.

Let’s start by taking a look at the “tileset” of an existing level. A good level for this is “Across The Network” (LEVEL.070), as it contains several types of special blocks.



Block 0 (Entrance)

(No special properties)

In the Level Components Panel, switch to the “Blocks” tab, then click the “...” next to the Metablock input field.

We can see the various blocks that are used in this level. By left-clicking a block here, we can select that as the block to draw with. Mousing over a block will also reveal details about any special properties that block has. There are two kinds of special properties – those that are specifically defined for each block, and those that are hardcoded to specific block indexes. Blocks can have the following special properties:

Hardcoded Properties

- Entrance: Block 0 is always an entrance
- Exit: Block 1 is always an exit
- Splitter: Block 2 is always a splitter
- No-Render: Block 3 is the “no render” block, I’ll explain it later
- Invisible: Block 4 is always invisible even if given face graphics
- One-Way Wall: Blocks 5 to 8 are always one-way walls, towards the Z=31, X=31, Z=0 and X=0 sides respectively

Configurable Properties

- Double-Sided: A block’s face graphics will render double-sided
- Steel: The block is indestructible
- Liquid / Electrified: Lemmings landing on top of the block will drown or get zapped (as per “Zapping Replaces Drowning” option)
- Anti-Splat: Lemmings falling from any height onto the block will not splat or get stunned
- Slippery: The top surface of the block is slippery
- Non-Solid: The block is visible but acts as non-solid. This can be either for lemmings only, for the camera only, or for both.

You’ll notice that Across The Network only has one type of one-way arrow block. Some levels only use one or two types, while some use multiple. Some levels also have metablocks for one-way arrows despite not actually containing any one-way arrow blocks, an example of this is “3D – A Lemming Odyssey” (LEVEL.044).

While Across The Network is a decent level for getting a look at various block types, it’s not particularly great for experimenting with editing blocks alone, so let’s load up a different level.

I suggest we load “Birthday Cake” (LEVEL.054). Once you’ve loaded it, select a block from the menu, and try placing it in the level. You’ll notice that all your blocks get placed at the very bottom of the level – and that only the top face is visible.

The “top face only” thing is simply what happens for blocks placed on the very bottom layer of the level. This happens even when using full-height sky graphics (and thus, there’s no sea to hide the other faces of the blocks).

If you look on the Control / View Options Panel, you’ll notice near the No Invisibles option, there’s some controls we didn’t look at yet – selections between the X, Y and Z axis, as well as + and – buttons with a zero inbetween them. Using the axis buttons, we select one axis that’s “locked” - any block we select with the mouse will always be on that axis, with the mouse position determining the coordinates on the other two axes. The + and – buttons can be used to control the coordinate on the locked axis.

In my experience, I don’t often find myself changing the locked axis from Y to either of the others, but it goes without saying that changing the coordinate is a very regular thing. Experiment with it and try to place some blocks.

Note that you can also select an existing block from the level by right-clicking it.

Before we move on to different block shapes / etc, let’s learn some keyboard shortcuts. I don’t intend to fully explain the keyboard shortcuts throughout the basic instructions, but these ones tend to be so frequently used and so useful that it’s good to know them even if you aren’t interested in keyboard shortcuts in general. Note that the keyboard shortcuts among these that **aren’t** on the numpad, will only work when the level image has focus. As a reminder, you can give focus to the level image area (without making any changes to the level) by holding Shift and clicking on the level image. Also note – these controls work *even when the “Disable Mouse Input” option is active,* and some of them are in fact specifically part of the alternative keyboard-based control scheme (which I will explain in full later).

Shift + Z – Reduce the coordinate of the locked axis
Shift + X – Change the locked axis (X → Y → Z → X)
Shift + C – Increase the coordinate of the locked axis
Numpad 4 – Decrease the selected block coordinate on the X axis
Numpad 3 – Increase the selected block coordinate on the X axis
Numpad 6 – Decrease the selected block coordinate on the Z axis
Numpad 1 – Increase the selected block coordinate on the Z axis
Numpad 2 – Decrease the selected block coordinate on the Y axis
Numpad 6 – Increase the selected block coordinate on the Y axis

In short – the Shift + ??? shortcuts work on the basis of “same input, always affects current locked axis”, while the numpad ones work on the basis of “the same button affects the same axis always, regardless of locked axis setting”.

As I mentioned before – and as you probably know from playing Lemmings 3D – there can also be different shapes of blocks, such as slopes, deflector blocks, even pyramid shapes. In the Level Components panel, on the Blocks tab, there is a dropdown from which you can select shapes, as well as buttons to rotate the block. There are also checkboxes that allow you to vertically slice the block. Note that all non-absent slices must be consecutive; ie: you could have just the top slice, or just the two middle slices; but you can't have the top and bottom slices present with the two middle ones missing.

For most block shapes, the vertical slices are a simple matter of that each slice is either present or absent; the block can start and end at any slice. However, there are a few shapes that have special rules:

Pyramid A Up: The top two slices must both be absent for the block to work. The two bottom slices work as normal.

Pyramid A Down: The bottom two slices must both be absent for the block to work. The two top slices work as normal.

Pyramid B (both directions): These essentially work as a two-slice block: the top two slices are one large slice, and the bottom two slices are one large slice.

22.5° Slope (both directions): The two highest / lowest (as applicable for the direction in use) slices that are present will form the slope. If only one slice is present, it alone forms a 22.5° slope that goes half way across the block horizontally. (As an interesting side note, try seeing what happens in-game if you use a destructive skill to destroy part of the slope.)

Special effects of blocks may or may not work with various block shapes and/or slicing. In general, it’s best to avoid different block shapes or slices with the hardcoded special effects, but it’s usually fine to use them with the configurable ones.

Two notable exceptions to this are firstly, entrances. These must be standard shape, but they must have **only** the top two slices present, in order to work. And secondly, one-way blocks, which work perfectly fine with whatever shapes or slices you want to use with them (although do double-check for visual oddities; they’re usually fine but some cases look a bit weird).

In terms of block editing, the most important thing is practice – and getting used to the hotkeys, to save yourself heaps of time. Experiment with modifying a few of the official levels and see what you can do!

One last thing – I mentioned earlier that Block 3 is the “no-render block”. Similar to Block 4, Block 3 is always invisible even if you give it graphics. However, the difference is that whereas Block 4 is just invisible itself, Block 3’s invisibility is “contagious” - any faces touching Block 3 don’t get rendered either.

This might seem like a pointless or even stupid-gimmick-y feature, but there is a very good reason for this to exist – there’s a limit on how many block faces Lemmings 3D can handle in a single level. However, any faces that aren’t rendered due to block 3’s effects, do not count towards this limit. By far the most common use case for this is indoor levels – block 3’s are generally placed around the outside of the blocks that make up the level walls / ceilings, so the outside faces don’t get rendered and thus don’t use up face limit capacity. Block 4’s own faces also don’t count towards this, whereas the same is not true for a standard block with some / all faces invisible; so Block 4 should be used for invisibles where possible.

(The exact mechanics of this face limit are not known, which is the biggest reason why L3DEdit has no capability to let the user know how close they are to the limit.)

If you’re trying to create your own level from scratch, for now, I suggest you import the metablock info from an existing level, via the “Import Block Data” option in the File menu. Editing the metablocks is a more complex task, and will be covered later.

Layout Editing – Land Areas

Aside from blocks, the other way you can add solid parts to your level is with land areas.

The graphic of the land areas is determined by the Land graphic file (in the Styles tab under the Level Metadata Panel). As was briefly touched on earlier, there are two types of land graphic files – those which contain a single texture, and those which contain multiple textures. Multiple-graphic files contain much smaller land textures, as a tradeoff for having more than one of them. All official multi-graphic files only contain two (or in one case, one!) land graphic, but it is possible for such a file to contain up to 8 graphics.

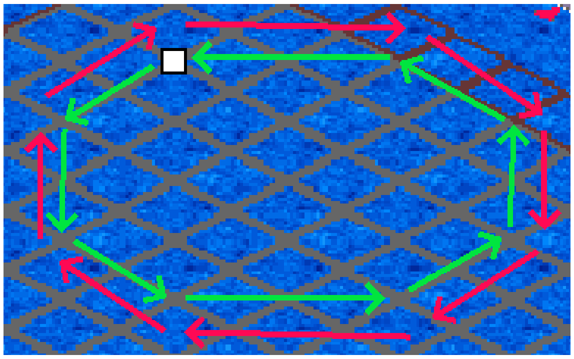
Aside from the graphic file itself and, where applicable, the selection from several texture options; land can also have a shade, which will darken the texture. This can be used even with single-graphic land files, to give different visuals – an example of this can be seen on “The Prisoner” (LEVEL.064). Try loading it, and in the “Land” panel, selecting different land areas from the dropdown box and experimenting with the Shade slider. For different textures, try “Lemlab” (LEVEL.035) or “The Hanger” (LEVEL.069) – you’ll need to use the Render Range options to be able to see the land areas in those ones! I am not aware of any examples where official levels use different textures from such files in the same level (they’ll use such files, but all lands in a single level will be the same texture from it), but it is completely possible to mix them together in custom levels.

When you're done experimenting with the graphical options, let's look at modifying the actual shape of the land areas. Land areas are defined in terms of between 3 and 8 vertices (corner points) each. Importantly, they do not have to consist only of 90° angles – or even of 45°, or even 22.5°, angles. They can be completely arbitrary.

However, there is one very important rule – the overall shape must be convex. Any concave land shapes, or land areas with holes in the middle, must be created by using multiple convex land areas. "Castle Lemmalot" (LEVEL.009) is an example with a hole in the middle, while "Ricochet" (LEVEL.045) is an example with a convex shape.

To add to an existing land area, select it from the list. If you want to create an entirely new one, select the "New Land" option. Note that a level can have at most 8 land areas, regardless of how many or few vertices each one has. However – keep in mind you can also place blocks at the bottom of the level, and use their top faces in much the same way as land. For an example of this, look at the bridge maze section of "Driving Range" (LEVEL.066) – most, but not all, of the "land" is actually blocks.

To select points for the land corners, simply click them in the level view area. You can start at any corner you like, but should add the points in the direction of the green arrows, not the red:



You can also delete the most recently-added point by right-clicking. Currently, there is no way to delete other points besides the most recently-added one, or to insert points between two earlier ones.

Some things to note:

- If there is the top face of a block, and a land area, on the same block space, the block has priority, both in visuals and physics.

- When land’s edge is at a non-90° angle, lemmings drowning (or getting zapped, if applicable) when walking off it is “pixel”-perfect; but skills will still only trigger at block edges / half-block points as usual.

- Overlapping lands will automatically tile correctly.

Layout Editing – Cameras / Preview Pivot

You’re very likely already familiar with how cameras work in Lemmings 3D – there’s four of them in a level, each with a set initial position.

The preview pivot is quite similar, but this defines the “center” point that the level preview rotates around. Note that this is irrelevant if the level has the “Fixed Preview Camera” option enabled – in this case, the preview just displays Camera 1’s initial view instead.

In L3DEdit, we use the Cameras tab on the Level Components Panel to place cameras, as well as the preview pivot. We use the dropdown to select which camera we’re placing, and then place the camera in exactly the same way as we’d place blocks.

Tip: When you’re done editing cameras, don’t forget to switch back to the Blocks tab (or whichever other tab you want to work on).

We can also use the Rotate buttons to change which way the camera initially faces. For the preview pivot point, the rotate buttons do not do anything.

It generally takes some trial-and-error to get camera positions right. One word of caution: When Lemmings 3D is in replay mode, camera boundaries are not enforced. While exceeding the boundaries on the X or Z axis, or going above the top of the level, has no ill effects, the game will crash if the camera goes below the bottom of the level, which can happen if a camera has been moved downwards since the replay was recorded.

Layout Editing – Level Boundaries

You’ve probably noticed how on many levels, if you stray too far from the level area, your lemmings get zapped. The level boundaries at which this occurs can be edited via the “Boundaries” tab in the Level Components Panel.

This is extremely simple – choose which boundary you want to edit from the dropdown, then click any block space along that boundary. Note that it is not possible to edit the minimum Y boundary – this is always at the bottom of the level.

There is no in-game visual indicator of the boundary positions, so if you’re using this at all, take care not to use it in ways that are intentionally misleading.

Layout Editing – Objects (Part 1)

There are several things that fall under “Objects” in Lemmings 3D:
- Static Objects
- Sign Graphics
- Animated Objects
- Interactive Objects
- Wall Graphics

Note that the names “Sign” and “Wall” here are quite misleading; and are used simply because (a) I can’t really think of better names for them and (b) that’s what the actual data files call them.

In this part, we will cover the general concepts of object placement, as well as all the specific types other than interactive objects (which will be covered in the next part).

You probably recall that blocks were placed into spaces on a 32x16x32 grid. Objects work exactly the same way. It is possible for a single space to contain both a block and an object (but it can’t contain two blocks, or two objects). Note that the results are often strange if objects are placed at the bottom of the level (ie: Y coordinate of 0), although there’s a few cases it works alright in.

It’s important to note that the majority of objects have a “vertical shift” effect. Specifically, if there’s a block on the same space as them, and the block’s top slice is NOT present, the object will be moved up by the height of the highest slice that is present. For example, if the second slice from the top is present (regardless of the presence or absence of any below that), the object will be moved upwards by three slice heights. This applies to all objects except for some types of interactive objects. There are also some types of interactive objects which apply this rule even when the top slice is present; this will be elaborated on in the next part. Please also note that L3DEdit does not always perfectly replicate these vertical shifts, so double-check in game if you're placing objects and blocks on the same space.

Static Objects and Animated Objects are quite similar. Place these in a block space, and the relevant graphic appears in that block space. That's all.

The graphic file for the Static Object can contain up to 20 graphics, though the majority of graphic files have several unused slots. The graphic can be selected via the Graphic input box (in the future, I intend to add a proper selection menu similar to the block selection). There are several different size graphics in the file.

The graphic file for the Animated Object works a bit differently. When the graphic index is set to zero, the full size image is used. When it's set to 2, only the left half is used, and 3, only the right half. In any of these cases, it's a four-frame animation.

When the graphic index is set to 1, it doesn't use the Animated Object graphic file at all, but rather uses the four largest-size graphics from the Static Object graphic as the animation frames.

For Animated Objects, there are also "rotate" buttons. "Rotate" is perhaps misleading here – this changes which corner of the block the animation appears on. For graphic 0, there are three options – the center, or two of the corners. For graphic 1, there's only the one option; in the center. For graphics 2 and 3, it can be any of the four corners (there’s no “center” option).

Sign graphics and Wall graphics are decorative graphics that are applied to a block face – or to the space where one would be, if no block is present. There are slight differences between them – sign graphics have proper double-sided graphics, while walls look the same on both sides. On the flipside, sign graphics cannot be applied to the top or bottom faces of a block space, whereas wall graphics can.

Both of these ultimately work the same way – select a graphic, rotate to the desired face, place object in space.

The one thing to watch out for – you might expect otherwise, but Sign and Wall graphics absolutely do exhibit the vertical shifting behaviour described above!

As a reminder, objects are never inherently solid. You must place an invisible block in the same space if you want to make them solid. To minimize the risk of hitting the face render limit, wherever possible Block 4 should be used for this.

Tip: Just like with blocks, you can right-click for picker-style selection of objects.

Layout Editing – Objects (Part 2)

At the most basic level, interactive objects are placed (and picker-ed) the same way as other objects. However, they often have special rules regarding their placement, and some also have pairing to consider.

A single level can only contain a single type of interactive object, which is selected under the Styles tab. Different objects have different considerations when placing them. We'll cover them in order from least complicated to most complicated.

Squasher, Bear Trap, Weird Trap

These all work the same way and are very straightforward – place them, and they simply work. The only catch to be aware of is in regards to vertical shifting (see previous part if you aren't sure what that's about) – these will vertically shift upwards even when the top slice of the block is present.

The "weird trap" is the one seen in "3D – A Lemming Odyssey" (LEVEL.044) and "The Hanger" (LEVEL.069), with something pulling the lemmings down into a black hole. I have no idea what to actually call it.

Flame Blower Trap

This is also a "place it and it just works" trap, however, this one needs to be rotated. If there's a block face on the side the flame originates from, a graphic gets placed on the wall – but placing a wall graphic on the same face can override that. Note that the flame blower trap does NOT shift vertically.

Laser Trap

Placement of this is the same as the Flame Blower Trap, right down to that it doesn’t have the vertical shift. However, I’ve mentioned it separately due to a unique and easy-to-overlook attribute of the trap itself – unlike other traps, where each instance is separate, every laser trap in the level operates together. If one of them is activated, all of them in the level are considered “busy” (and thus lemmings can safely walk past) until it’s finished, and all of them will animate.

Trampoline

Trampolines must have a block on the same space as them, and the trampoline itself will be placed on the top face of that block, whatever height it may be at (including on top of the top slice). The graphic index is used to determine if it’s a red trampoline or a blue trampoline. Beyond this, they’re place-and-they-work.

L3DEdit will display teleporters even when they don’t have a block in the same space, whereas Lemmings 3D itself won’t. This is the reason for the mid-air trampoline you noticed earlier on “Take A Dive” - there’s a trampoline in the object data, but no corresponding block for it to appear on, at that position.

Note that land does not count; it must specifically be a block. If you want to place a trampoline on the bottom of the level, place a block (with the top slice present) in the same spot.

Teleporter

Teleporters follow the same placement rules as trampolines. The “Pairing Index” is used to determine which teleporters pair with each other. If there is only one teleporter with a given index, it won’t work. If there are three or more, only two of them will work.

The range of valid values for teleporter pairing indexes is 0 to 7. This allows for up to 16 functional teleporters in a level, and at least one level reaches this (in addition to also having a couple more that, due to no corresponding block in the same position, don’t appear in-game) - “Motherboard” (LEVEL.052).

Spring

Springs consist of pairs of senders (the spring) and receivers (the landing pad). They use a pairing index like teleporters, but are one-way – a spring with a pairing index, sends lemmings to the landing pad with the same pairing index. The “Receiver” checkbox is used to switch between senders and receivers.

Aside from this, they follow the same rules as teleporters (including placement requirements). The only difference is that valid pairing indexes are only from 0 to 3, instead of 0 to 7.

Rope Slide

Rope slides are by far the trickiest interactive object to use correctly. Pairing works the same way as springs – there’s a sender and a receiver, with pairing indexes from 0 to 3. However, no visible graphic is produced by placing the start or end points. Instead, the graphics of the rope slide are produced using blocks, with the “Non-Solid” flag enabled.

In terms of placement, the sender follows the same rules as trampolines, teleporters and springs. The receiver however does not – it can be arbitrarily placed anywhere, with or without a block. The one thing to be aware of, is it should be placed in the block space the lemming will stand **on** when it finishes the rope slide, not the space the lemming itself will be **inside**.

Note that among the official texture sets, only the four Army texture sets (TEXTURE.0?6), plus one of the Maze texture sets (TEXTURE.095, or in the L3DEdit presets, “Maze 3”), contain rope slide graphics, so they’re pretty much restricted to Army or Maze levels unless you’re using custom graphic files.

Layout Editing – Filling and Shifting

There’s one last thing you might have noticed and wondered about – the “Mark Block”, “Fill Region” and “Shift Level” options. Fill Region allows you to quickly fill a region of the level with a single block or object (depending on the current selected tab in the Level Components Panel), while Shift Level allows you to move the entire level.

In order to use either of these, first, a block must be marked. To mark a block, select the “Mark Block” option, then click the block you want to mark. Now, a green outline will appear around that block (you might need to move the mouse away from it first, otherwise the usual white outline of the mouse-overed block will override it). When you click the block, the “Mark Block” option will automatically deactivate, and you’ll go back to normal mode.

When you activate the “Fill Region” option, and click a block (the “selected block”), every block space between the marked block and the selected block will be filled with the currently selected block or object (depending which tab you’re on in the Level Components Panel).

When you select the “Shift Level” option, and click a block, the entire level will be shifted by the difference in position between the marked block and the selected block. This includes camera positions, boundaries and land, although of course land will only be moved along the X and Z axes, not up or down.

Editing Metablocks

By this point, you definitely know enough to create your own levels from scratch. Maybe you *have* already created a few by now. However, the one catch is that so far – you've been stuck importing the metablock data from other levels.

You've got three options regarding metablocks. You can import them from another level, and use them as-is; as you would have been doing so far.

Or, you can import them from another level, then make your own modifications to them as you see fit. I recommend doing this while you're still getting used to metablock editing.

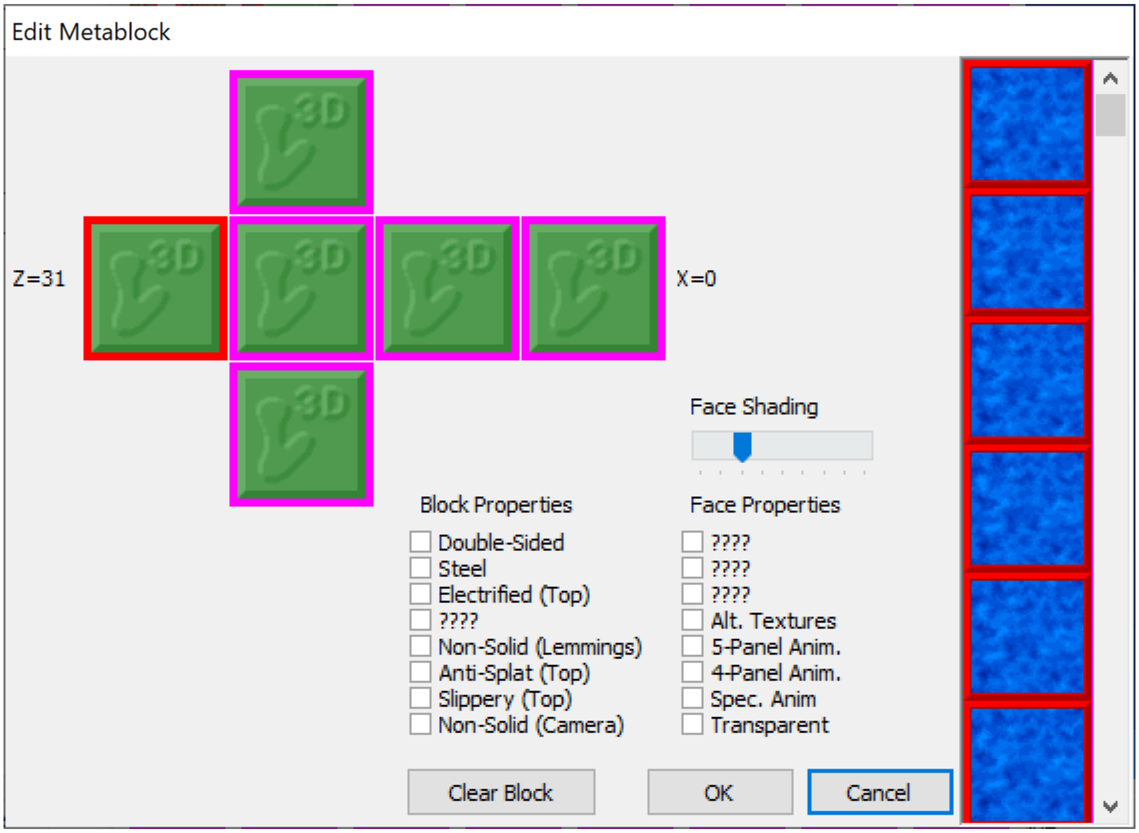
The third option is to completely create your own metablock data from scratch. As you get more familiar with Lemmings 3D editing, you very possibly will find yourself preferring this approach – or at least, to create levels with your own pre-set collections of metablocks that you import from, instead of using official levels.

A fourth option could be to open an existing level, erase the layout, and create a new level using its metablock data. This is essentially just a more-complicated variation of option 1.

Let's keep it simple for now – let's change some stuff on existing levels and see what happens. Open "Attack Of The 50ft Lemmings" (LEVEL.018); this level is selected because it has a small number of metablocks, all of which are visually distinct.

To edit a metablock, first open the block selection window – as a reminder, you can do this by clicking the "..." button in the Blocks tab. Then, right click on the metablock you want to edit.

Let's start by changing the graphic. Right click on one of the plain colored Lemgo blocks (since they're easiest to spot in the level), which will open the metablock editing window.



Notice how the face near the "Z=31" label has a red outline, instead of magenta. That marks it as the currently-selected face. The "Z=31" and "X=0" themselves, indicate which way the face next to them would be facing, when neither the block nor the level view have been rotated. This is important, as the Z=31 face, and sometimes one adjacent to it, is usually the one that participates in deformations for different block shapes, as well as usually the one that has the functional effect in hard-coded special blocks.

On the right are all the face textures in the level’s texture set file. If you click a texture in this list, it will replace the graphic of the currently-selected face. You can also darken the face graphic with the Face Shading slider.

You’ll also have noticed the checkboxes. The eight on the left relate to the entire block, while the eight on the right relate to the selected face only.

The block properties are fairly self-explanatory; the “????” is a flag that exists in the data files but I don’t know what it does (it was easier to code, to have a checkbox than to not have one for it).

“Double-Sided” is generally used when some faces are missing or have transparent sections, so that the opposite faces get rendered on the inside as well.

“Liquid (Top)” is replaced with “Electrified (Top)” in levels that have the “Zapping Replaces Drowning” setting; the difference is purely cosmetic.

“Anti-Splat” means that lemmings will not splat, or even get stunned, when falling on to the block from any height.

The face properties are a bit less straightforward. The unknowns are the same deal as the unknown under block properties.

“Alt Textures” can only be used with certain texture indexes, and does nothing if the Double-Sided block flag is not enabled. If it is, then the reverse side of the face will have the next or previous texture as its graphic, instead of the same one. Textures are essentially paired for this one – eg. texture 43 will have 44 on its reverse side, while texture 44 will have 43 on its reverse side. All textures that this flag can be used with are consecutive in the texture set file; specifically #43 to #54.

“5-Panel Animation” and “4-Panel Animation” make the face graphic animate between the selected texture, and the next three or four (as applicable). The animation is in an A, B, C, D, C, B, A etc order (or equivalent for 5-panel).

“Special Animation”, like Alt Textures, can only be used with certain texture indexes, and works as a four-panel animation with an A, B, C, D, A, B, C, D frame order. It can be used with texture 0, 12, 16, 28, 32 or 36 (or a texture that would occur during an animation starting from one of those). As a special case, when used with texture 0 (or part of its animation), it’s an eight-panel animation rather than four-panel.

“Transparent” is required for part (or all) of the face graphic to be transparent; it will be solid black instead if Transparent is not enabled.

Don’t forget that blocks 0 through 8 have their own inherent special effects. To properly use these blocks:

Block 0 (Entrance)

The Z=0 and Z=31 faces should have the half-height sides (no flaps), while the X=0 and X=31 faces should have the whole-height sides (with the flaps). And don’t forget – when placing entrances, they should have only the top two slices present! Entrances should be given the Steel property.

Block 1 (Exit)

The Z=31 face is the exit. If the 9th texture in the set is used, the 10th to 12th will be used to animate the exit door opening and closing. If any other texture is used, it works as a normal face graphic. Exits should be given the Steel property.

Block 2 (Splitter)

Face textures work completely normally for this one, but it should have the “Steel” and “Non-Solid (Lemmings)” properties. More often than not you’ll probably want the “Double-Sided” property too.

Block 3 (No-Render), Block 4 (Invisible)

Any face graphics given to these will show up in the editor only; these blocks are completely invisible in-game no matter what. Their non-visual properties (Steel, Slippery, etc) still work as normal, though.

Block 5 to 8 (One-Way Walls)

Face graphics work completely normally for these. The arrows should point towards the Z=31 face for Block 5; the X=31 face for Block 6; the Z=0 face for Block 7; and the X=0 face for Block 8.

Tip: There are a lot of graphics in the texture files that aren’t used in any official Lemmings 3D levels, for example, a Pyramid splitter block. In most cases, these aren’t even used in unused metablocks, so you’ll have to edit metablocks yourself (or find custom levels to import from) to use them.

Hotkeys / Keyboard Controls

Many of L3DEdit's functions can be carried out using the keyboard. It's possible to almost completely use the keyboard when creating a level, though in my experience, the fastest approach is a hybrid usage of keyboard and mouse controls. I suggest experimenting and seeing what works for you – mouse control is often faster for block selection (except when changing coordinates on the locked slice), while keyboard control is faster for moving the locked slice, and is also easier to achieve precise positioning with.

Keyboard controls can be divided into three sets – menu shortcuts, the numpad controls, and "everything else".

Menu controls usually involve holding Ctrl and pressing a letter. These always correspond to an item in the menu bar, and the shortcut keys will be displayed next to the menu items.

Ctrl + N – Create new level
Ctrl + O – Open a level
Ctrl + S – Save the current level
Ctrl + Shift + S - "Save As" the current level
Ctrl + I – Import metablock data from another level
Alt + F4 – Exactly what you expect it to do
Ctrl + Z – Undo
Ctrl + Shift + Z – Redo (note: Ctrl + Y for Redo is not supported)
Ctrl + + - Zoom in
Ctrl + - - Zoom out

Numpad controls are an alternative to mouse control in the level view, although they can also be used to complement rather than replace mouse control. Num Lock must be enabled for the numpad controls to work.

Numpad 4, 5, 6, 1, 2, 3 – Move the selected block
Numpad 7, 9 – Rotate the level view
Numpad 8 – Place block / object (equivalent to left-click)
Numpad 0 – Pick block / object (equivalent to right-click)
Numpad . - Mark block
Numpad / - Fill region
Numpad * - Shift level

Note that Mark / Fill / Shift here do not expect you to first press them, then select a block and press 8 (or click) – the currently-selected block is immediately marked / filled to / shifted to upon pressing the hotkey.

The remaining hotkeys only work when focus is on the level view image. As a reminder, you can put focus on the level view image by holding Shift and clicking on it. The level view image also gets focus any time you click on it (with or without Shift; holding Shift just makes sure nothing *else* happens from the click, like drawing the current block), or any time you use a numpad hotkey.

All of these hotkeys involve holding Shift and pressing a letter or number. Some can be used at any time; others can only be used when certain tabs are active in the Level Component Panel. Any that involve a number, the non-numpad number key must be used.

Any Time

Shift + Q – Switch to editing land areas
Shift + W – Switch to editing blocks
Shift + E – Switch to editing objects
Shift + R – Switch to editing cameras
Shift + T – Switch to editing boundaries
Shift + Z – Decrease the current coordinate on the locked axis
Shift + X – Change the locked axis (X → Y → Z → X)
Shift + C – Increase the current coordinate on the locked axis
Shift + F – Mark the currently selected block
Shift + B – Rotate the level view
Shift + O – Display the number of blocks in the current level

Blocks or Objects Tabs Only

Shift + V – Fill region between marked block and selected block
Shift + H – Shift level between marked block and selected block

Land Tab Only

Shift + A – Switch to the next land area
Shift + S – Switch to the next land graphic
Shift + D – Darken the land graphic

Blocks Tab Only

Shift+1, Shift+2, Shift+3, Shift+4 – Toggle the respective slices
Shift+A – Open the block selection window
Shift+S – Change the block shape
Shift+D – Rotate the block

Objects Tab Only

Shift+A – Change the object type
Shift+S – Change the graphic or pairing index
Shift+D – Rotate the object / switch between sender and receiver

Cameras Tab Only

Shift+1, Shift+2, Shift+3, Shift+4 – Switch to respective camera
Shift+A – Switch to preview pivot
Shift+D – Rotate the currently selected camera

Boundaries Tab Only

Shift+D – Switch boundary