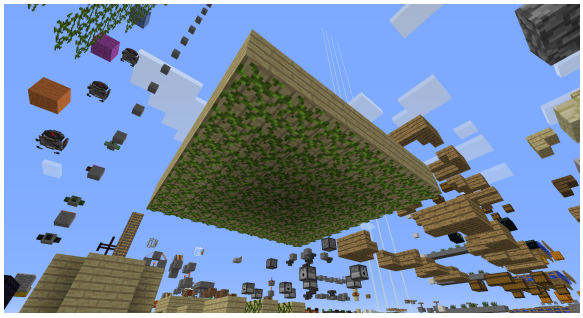
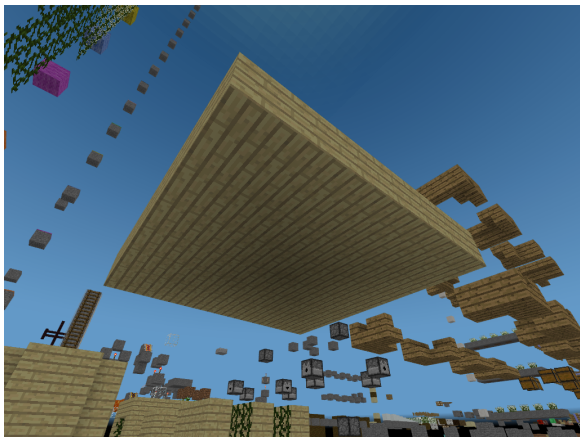
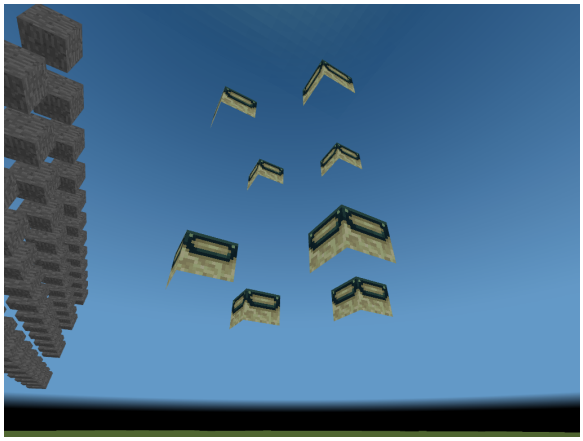
Possible eihort v0.4 Features

A few ideas about how eihort could work in the future. In this text new and *old* (already posted) ideas are mixed. The ideas are not in a specific order.

Bugs

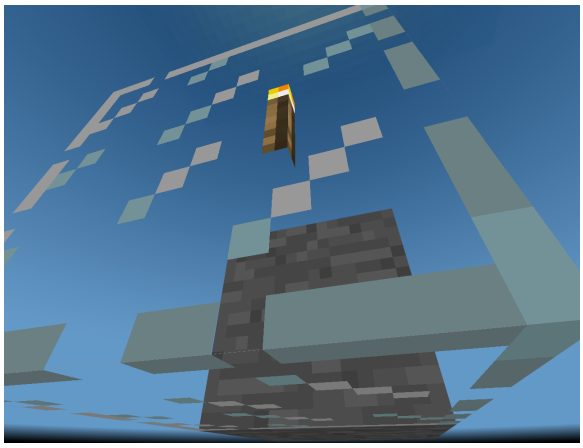
Slabs cut each others faces This problem existed before with stairs and was fixed some time ago. Issue #64: https://bitbucket.org/lloigor/eihort/issue/64/slabs-cut-each-others-faces	
Vines at the bottom of blocks are not displayed The sandwiched vines are fixed but the vines at the bottom of block still do not get displayed Issue #66: https://bitbucket.org/lloigor/eihort/issue/66/vines-as-bottom-cover-below-a-block	Minecraft:  eihort: 

CompactedBlocks have no bottom cover Sometimes the bottom block of Compacted Blocks are not displayed. Issue #59: https://bitbucket.org/lloigor/eihort/issue/59/pressure-plates-need-a-bottom-cover	
Torches do not have a bottom cover torches somewhen got a bottom cover which was so far no included in eihort. The colours: • green: north-west corner • blue: north-east corner • red: sout-west corner • yellow: south-east corner	Minecraft: left side is north face:  left side is south face: 

used torch texture:

bottom (stone is north):

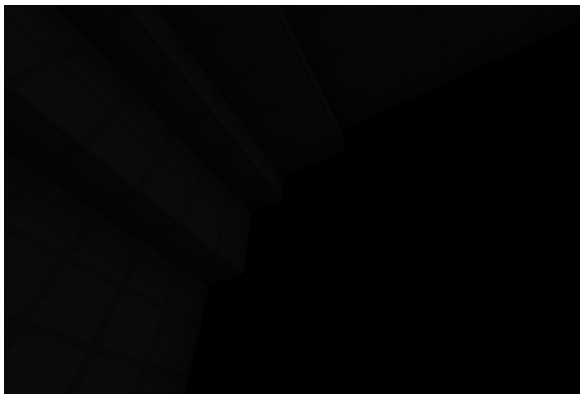
eihort:

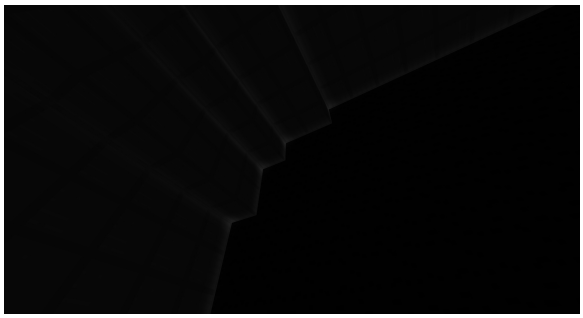


Wrong biome colours:
eihort does change the grass colour depending on the biome, but minecraft uses other colours. Most of the time they look similar but when you compare them side by side you can see the difference.
On the right side is the best example (swampland).

Issue #93:
https://bitbucket.org/lloigor/eihort/issue/93/biome-colours

Minecraft:

	Eihort: 
<u>Spline funciton *charges* itself</u> The spline options works fine for the most part. Exept for one big thing: Before the spline goes down a long fast line (because the next point is pretty far away) it passes the point goes a little bit back (that's what I mean with “charges itself”) and flys extremly fast into direction you want to have it. Not only that the *charging* thing looks stupid, you sometimes end up in some wall or another thing you don't want to be in.	
<u>Facing direction is not saved</u> When the “Move player here” option is used the facing direction is never saved. After the player starts minecraft and loads the world he is always facing south-west	
<u>A light engine bugs</u> The light engine needs to be corrected a little bit. The top image set shows a dark corner in The Survival Games 2 map. As in The End Dimension the light shines through the edge of the block. Another thing is that some block are displayed too dark. The carrot is just one example for many other blocks like vines and ladders.	Corner – Minecraft: 

	Corner – Eihort:  Carrot – Minecraft:  Carrot – Eihort: 
Extra options menu can't be snapshoted Maybe not a bug but a feature. If it's planned that this happens there should be an option to turn it off. The use of this would be that you can locate the a place on the map very easily (eg. For bugtracking) or that some people way what to have the additional information in there.	

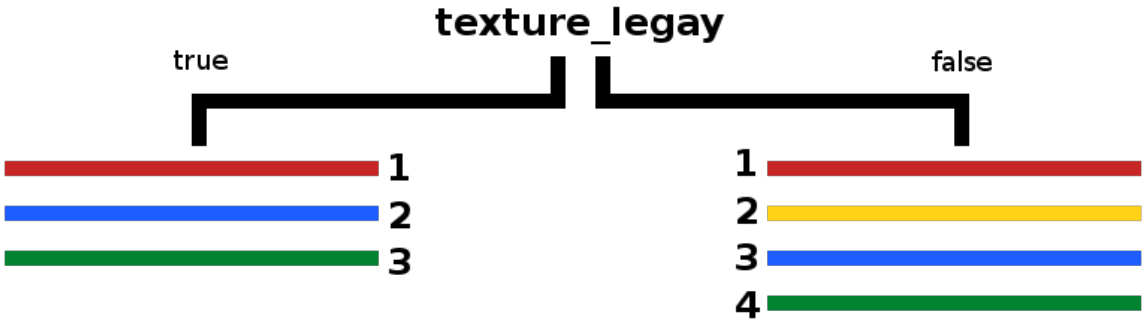
Suggestions

New options in eihort.config:

- Toggle sun & moon
  A new part in the eihort.config file which allows the user to toggle the sun&moon if prefered (eg “display_luminary=true/false”)

- Use hidden features
  This option could be used to include new features which do not fully work but which should be available for testing. To prevent the ordinary user from making eihort unusable it could be implemented as a non existing line in the eihort.config file. You would have to add eg. “use_hidden_features=true” to make them usable.

- Add texture legacy support

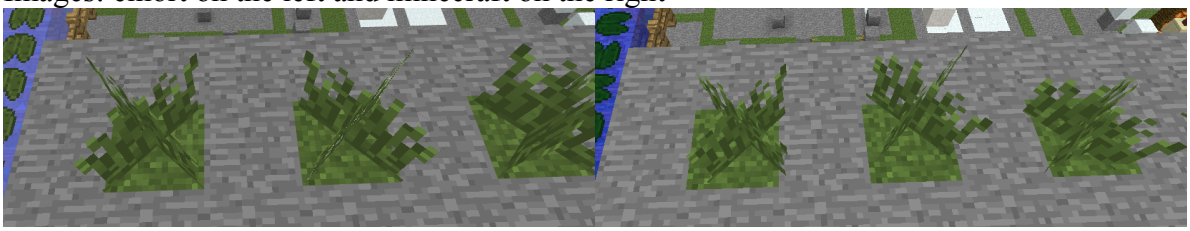


So far eihort can display anvil as well as MCRegion maps. It is good that eihort did not drop the ability to display MCRegion maps since there are still users out there who play the older versions of minecraft. However even eihort does display MCRegion maps, it can not load textures from the terrain.png file any more. So either the MCRegion map support should be dropped or (what I would prefer) the part to load a texture get's legacy support. If the option “texture_legacy” is set to true, eihort expects the terrain.png file somewhere. If it's set to false the formate of 1.5+ will be used.
“true” will check if there is a terrain.png file in eihort's location (1), if not check if there is a minecraft.jar file in eihort's location (2) and if not check the regular minecraft path in the config file (3). If none of these three steps exists it drops the usual 'file does not exist' warning.
“false” would check if the texture file (eg stone.png) does exist in eihort's folder (1), if not check if there is a path in eihort's folder named “textures\blocks\” in which stone.png is located (2), if not check if there is a minecraft.jar file in eihort's folder out of which the texture files can be loaded (3) and if not check the regular minecraft path from the eihort.config file. The additional step is because if you want to test more than one texture you would have drop all the files in eihort's folder which would be come pretty messy.

New features:

- Block orientation for lily pads
  Lily pads face different directions depending on their location. This means that the pads always face the same direction no matter how often you place them. It doesn't matter on which map you are the lily pad on coord x/y/z will also face the same direction. Since there is no documentation how the facing is calculated eihort could include it's own calculation which should deliver the same results over and over again. The left image shows lily pads in minecraft the right one in eihort.

- Block orientation of Tall Grass
  Tall grass needs a correction as well. Lily pads do only face a certain direction. Tall grass can have 9 different places and most of them are *outside* of the block. That means that it can visually appear in the part of another block. So far eihort displays tall grass in the middle of the block. The same idea like for lily pads: Either find out how the orientation of tall grass is calculated or create a self-made way to get an orientation.
  Images: eihort on the left and minecraft on the right



- Drop single dimensions on eihort
  Since there are a ton of mods which add new words eihort will never be able to cover all of them. Since they're usually still saved as anvil or MCRegion file eihort should be able to display them. You could drop any dimension on eihort and it would display it (so you could enter eg. The nether without the need of loading the overworld map and then going into the nether. This would also be the same with worlds like the moon, twilight forest and more world that mods would add)

- Adding a command line to eihort
  The command line would work much like in minecraft and you would be able to config eihort on the fly without the need of restarting eihort every time when you change something (eg. Change key layout on the fly, set values to true/false, config spline paths, restart eihort with a single command...). The command line could also 'talk' to you and tell you everything that you change in eihort.

- Include the christmas chest
  The chest does not fully work but one day it hopefully will do. At that point the christmas chest could be added since the texture is auto used during christmas.
  

- Display only the raw grid
  eihort loads minecraft textures and puts them on the correct face. The idea is to skip the texturing part and make eihort to only display the raw grid without any textures. This feature could be implemented as hidden feature.

- New access to Item and Entity IDs
  Since some information which is essential to display blocks correctly is stored in the Tile Entity Format, there should be a way to access these as easy as with “DataAdapter”. The head block for example can not be displayed accurately without additional information which can not be accessed yet.

- Eihort loads minecraft.jar file from eihort's folder
  To get the terrain.png file (or the 1.5+ version of it) you need to extract the files from the .jar file which some people aren't able to. So eihort could load the texture files from an .jar file which is dropped in eihort's location (pretty much the same way it does with terrain.png). The problem will be the new way of sorting the versions of minecraft since they are no longer named “minecraft.jar”

- Separate eihort to modules
  The main idea is that eihort has two modules for the texture part. One module loads the texture and puts into an image-array, the other one puts the textures onto the correct block. First eihort checks if the legacy support for textures if set to true. If it is, eihort will expect to find the terrain.png file somewhere. If it's set to false eihort auto checks if some folders (which contain the texture files) do exist. The location of 1.5 textures is “minecraft.jar\textures\blocks\xyz.png”. The location of 1.6 files is “1.6.jar\assets\minecraft\textures\blocks\” due to the “resource packs”. The module would get a list of snapshots (and *main versions* of course) and in which snapshot a block appeared and where it is saved. It would check the latest snapshot or version and will then go down to the oldest one and load the texture file out of the latest .jar file. In the layout for the versions should be included where texture files can be found (eg: if .minecraft\bin\minecraft.jar\textures(\blocks\stone.png) exists it's very likely that it is 1.5. If there is a folder called “versions” and the folder with the highest version number is 1.6 eihort knows to go into \assets\minecraft\textures\blocks to find stone.png). The locations of the current versions/snapshot would need to be updated every week or with every new minecraft version but I guess that's an easy way of making eihort compatible with pre 1.5, 1.5 and 1.6 texture stuff.
  If the texture files are located and eihort knows which files can be loaded the first module loads all available textures into an array of images (I have no idea if something like this really exists and/or is even possible – I've taken the idea from programing in C: eg. Char[10][6] which will give you an array with 10 lines with 6 characters each). It would put the

texture file into the correct part of the array (depending on their blockID eg. stone into [1][0] to [1][5]). The first element could be the top, then bottom, north, east, south and west. If there is no separate top, bottom,... texture a transparent file could be included.
The second module (located in blockids.lua) would load the textures from the image array. That way the second module does not need to know the names of the texture files and the location of the files if the first module puts the textures into the right blockIDs. If the textures are lined up correctly, there would be now need for an extra line up which texture goes on which face since it would be always top, bottom, north, east, south and west.
So even there will may be changes in the future which does require recoding in the first module, the second module can stay the same since every part of the first module has the image array as end goal.

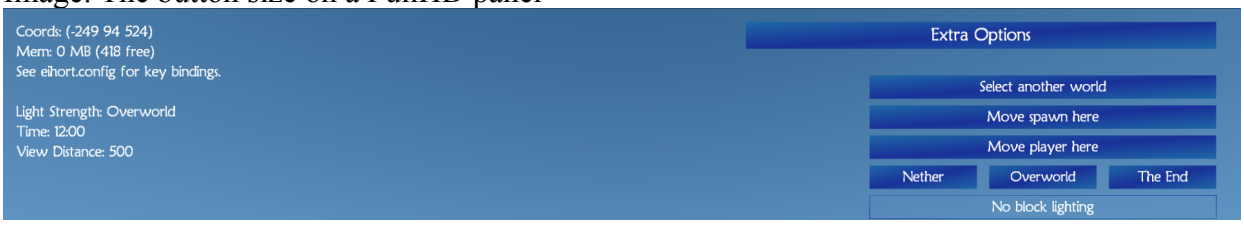
- New way to include further textures
  Nearly all mods do include new blocks. If eihort could display them it would be very good since it would increase the possibilities to use eihort. This option could be included with another lua file (eg “textures.lua”) or could be handed over to the first module.

- Option to cut out and rebuilt textures
  Since some blocks have multiple faces in just one texture file (the cocoa plat for example). Eihort will need a tool which can cut out the needed faces and putting it onto a block. The blocks texture should be able to be stretched or made smaller if the texture's width is 7 and the block's width is $^{8}/_{16}$ (the large cocoa plat for example).

- Rotate option for block faces
  Since some blocks use the same cover but only with a rotated face (eg. pistons) you need an option to turn the texture files. This could be done in the second module which puts the textures on the blocks since there you can specify that block xyz gets face A when he 0x8 bit is set, if not it get's rotate(180,A)

- Vertical and horizontal flip option
  Some blocks do need an option to flip the textures to be displayed correctly (eg. beds). This option would much work like the rotate option and could be included in the second texture module. Since some blocks require the rotate and the flip option to be displayed correctly rotate(180, flip(v, texture.png)) (or something like that) would also be useful.

- Add programmable blocks
  The idea is very simple but would change how eihort could display blocks. Example: A large chest actually contains two small chests. The cobblestone wall does behave different depending on the blocks around it (it does or does connect or builds a pillar). The idea is to combine all such 'special' blocks in one location which works somewhat like a programmable block. For water it would be: if (blocktype(block-y-coord++) != blocktype(block-y-coord)) then use small block (one with a lower height).
  Image: How eihort (left) and minecraft (right) display corner stairs. The stair block could be *programmed* so that it's behaves differently depending on other stairs around

- Limit max button size
  The buttons resize depending on the windows size. This is good. However, if you have a FullHD panel the extra options button will become incredibly oversized (it takes up like a third of the top screen). If there would be a size limitation for that button eihort would look better on large screens. This could also go the other way as well and include a minimum size for all buttons. I think you can expect that a display has at least a width of 1024 pixel. The maximum size of the buttons could be reached with the *small* HD resolution (1280 x 720).
  Image: The button size on a FullHD panel
  

- Close window with Alt+F4
  You do not really notice it when you only use eihort once in a while but when you include new blocks to eihort it does become a little bit boring to always grab the mouse to close eihort's window. Since the Alt+F4 combo works on many programs on Linux as well it wouldn't be only a windows feature.

- Show animated block faces
  Some blocks do already have animated faces (eg. the portal, water,...) and some texture packs do add even more animated faced (for the enchantment table for example). If disable_cpu_saver is set to true the images get rendered no matter if something changes or not. If the images are still rendered no matter what you could include the animated faces.

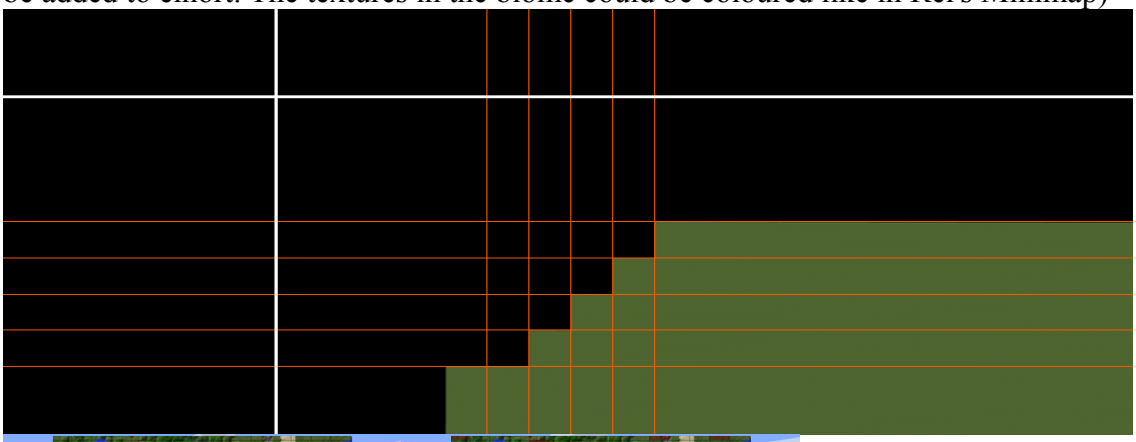
- Save spline path
  Since eihort can show massive builds due to it's huge render distance you could ducument the steps of an ingame building in a video. You will probably have a hard time to get the exact same spline path, points and view angles. So it would be a usability increase for such a project if you could save the spline path somewhere (like you can do with screenshots). Of course you would need to be able to load and change it after it is finished. So a plane collection of coordination values (including view angle and so on) in an xml file is probably the easiest way.

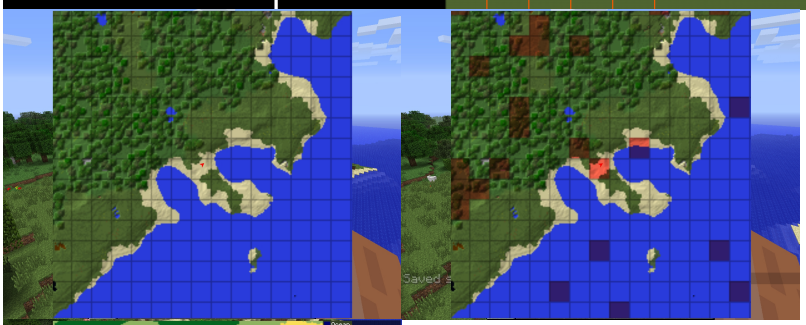
- Full spline editor
  If you are able to save and load spline paths the next step would be to create a spline path editor. This editor could be accessed via the extra options menu (maybe a small button at the bottom) which allows you to swap between the 'real' eihort (with all it's highlight and other features) and the spline path editor. The spline path editor would show you a thin line which connects all points in the order you would pass them. Every point of the spline path shows a pyramid. The tip is located in the point and the base is the part of your world you would see. You would be able to edit the speed of all ways and the view angle of all points even the spline path is already saved. Another interesting option would be an extra window which shows you what you see in eihort when you follow that path. Your 'real' window however stays at the position you are at and is not moved unless you do it. To show where the window exactly is, a small light ball could move along the spline path line. If you have changed anything you should be of course be able to save it. The save and load options could be done like the already existing buttons for the highlight, move spawn,... options.

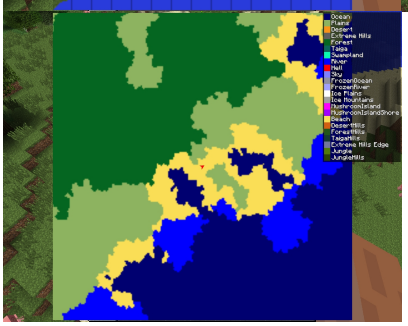
- Button to re-add last deleted spline point
  So far you can delete the last spline point in the row with “E”. Sometimes it is useful to have a button which re-adds the point after you realized that you deleted one point to much. That button could be added somewhere on the Keyboard (since I guess it wouldn't be used that often). I would suggest “C” since it's near the main control keys and at the opposite end of “E” which deletes the last spline point in the row.

- View distance and time slider
  It usually takes a little bit longer to get from view distance 500 to eg. 5000. So a time slider would be a good option to adjust both values in a fast and easy way. However, the option to change the values with the buttons should be kept because it would be hard to change the view distance to an exact value if you could only use your mouse.

- Show size of region and junk files
  So far you can see in which region file you are located. The idea is that every region file get's a small border if it's set to be that way in the config file. It could look like a 10 block tall, 50% transparent, white wall. The chunk files could be 5 blocks tall and have a 50% transparent green colour. The colours could be adjusted like in the different places to load textures.
  Image: A view from above the map. The orange lines would be the chunk borders and the white line would be the region file border.
  The bottom three images are from the mod “Rei's Minimap” (Tekkit). It shows the chunk grid and slime chuks. An additional feature is to show biomes (also something which could be added to eihort. The textures in the biome could be coloured like in Rei's Minimap)



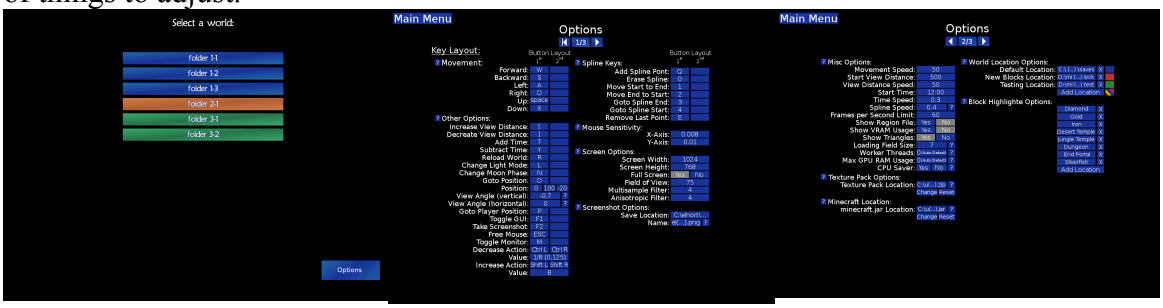




- Add goto button in the extra options menu
  Sometimes it takes some time to get from one edge of the map to the other. Yes, there is P and O for the players position and another position. What's still missing are three fields where you can enter the new coords you want to be at and then type enter and eihort 'teleports' the view to the new position.

- Reload everyhing button
  If you change something like blockid.lua entries to include new blocks it takes a lot of time you spend when you have to restart eihort because the values were not entered correctly. The idea is to include a (maybe hidden or turned off by default) button which works like you would restart eihort and reload the map. Eihort checks all the files it needs to start and changes the existing values if necessary.

- User Interface for config file
  Something which I really hoped for a long time is an easy usable graphical interface to edit the config file. It would be accessed via the main start screen and contains every thing you can change with eihort. The reason is not only that it looks great but rather also that some people are afraid to ruin something when they touch the numbers in the config file. Such a user interface would make it easier for them to enjoy all of eihort's possibilities.
  Images: left to right: Enter the options GUI via the main map page. Different sides with lot's of things to adjust.

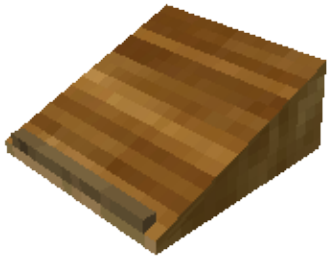


- .eihort file in map folder
  Since different maps require different values for all kinds of options it would be good to have an option with which you can bind the settings to a specific map. Eg: A map with dungeons, rather small houses and small areas require different values of speed as a map with wide areas and something like landart or huge houses. The position you want go to with “O” will also be different for all maps. If set to true in the config file eihort could save the current options in a separate file in the map's folder. It could be something like “settings.eihort” or “config.eht”

- Adaptive highlight menu
  Since eihort could be extended to more than the regular vanilla minecraft, the highlight menu should be easy adjustable and more flexible. If a mod adds for example new ores you may want to search for it. If you have a mod pack like tekkit with like 10 new ores you will have a space problem on smaller screens because you can't make a list for 30 different ores and/or blocks you want to highlight. If you group the highlight buttons and include buttons to change them it would be a lot better. It would be even better if you would be able to adjust that highlight menu with a UI since many user do not want to go into detail how to change the highlight list (they just say something like “eihort can't do that it's crap”)
  Images:
  left & middle: Some blocks will be used over and over again so there is no need to make them easy adjustable. The adjustable part could be better controlled if it would again be accessed via a GUI.
  right: since mods usually add some ores it can be pretty messy if you are using something like eg. tekkit. If blocks highlighted blocks are put together in a category there can be a small part for categories and another small part with blocks which are may a little bit bigger than the layout right now, but is in sum smaller because you have much more blocks to highlight.

- Adaptive extra options menu
  Not only the highlight menu needs to be adaptive but also the extra options menu. Some people may want to deliver a specialized version of eihort especially for their mod, world , texture- or map-pack. Such a usage could be that the all ores are removed from the highlight list, or that there is only the nether and overworld because the end is not part of that adventure map. Or maybe somebody wants to add something like a *mark area* button which works hand in hand with the minecraft mod which then highlights an area with a special tool. These are only a few thoughts of countless other possible options.

- Specialized blocktype for blocks
  Blocks like the head block or the sign post can not be displayed 100% correctly since they need options that not blocktype can deliver so far. Since there are even more blocks which require even more different options there may could be a single blocktype for all for which you can add features. The blocktype would be able to display anything that you can think of. You create something like a subblocktype by booking abilities of the block. So you could book float instead of integer values, or book an option which is capable of displaying a level and some glowing options. Every not included feature which is not used will be ignored. So if the highlight option is not used there will be no highlight option. The use of such a rater complicated blocktype would be of course to have a powerful tool which is able to display even the most complicated block. If eihort can not display them with the same quantity as OpaqueBlocks it's fine I guess since such blocks are usually very rare.
  Images:
  left: A “Lectern” which does only need an option to display a tilted surface
  reight: A “bookstand” which would require some more detailed coding




- Add items to highlight list
  Since some blocks (which aren't blocks but items) are valuable there should be an option to show them as well. An example would be the Minecart with Chest which can be found in abandoned mineshafts since 1.5. Regular chests can be highlighted with eihort. But not the minecart with chest. You can highlight the rail below it but you can't see where such a minecart is located at.

- DataAdapter can get values from blocks with other coords
  The main use of this would be for blocks like the doors which do store some information in the bottom part and some in the top part. So far there is no easy way to get the information out of a different block. A single part where you could may include x++ or y-- to get the next higher x-value or next lower y-value block.

- Extra int-value to move blocks
  Farmland is missing 1/16 of a regular block. So wheat, carrots and so on are too high. So *Cactus* or ' *SharpShapeBlock*' (see below) needs an extra value so that it can be lowered. Since this can may happen to many other block types I would suggest something *outside* the block itself. Like DataAdapter it would not be included in the Opaque-, Compacted- or whatever block, but rather outside of it.

<u>Other changes:</u>

- Rename Block types
  The “MultiBlockInBlock” is not really what I thought it should be since it's the actual “MultiCompactedBlock”. While the “MultiCompactedBlock” is rather a MultiCompacted**Connected**Block. This applies also to other blocks such as the “Cactus” and the “Flower” which I would rename into SharpShapeBlock and XShapeBlock since the blocks after which the blocktype is named are not the only ones which use them.

- Eihort should not require block textures
  So far eihort does require the texture files of all programmed blocks. This will become a problem with newer versions of minecraft since the Hay Bale did not exist in 1.5. If you load a later version of eihort which does include that block you can not use it with an older version of minecraft which does not deliver that texture. If the texture can not be loaded from it's file eihort could drop a message if the command line talk option is set to 'true' (tells you something like “file xyz.png does not exist”). I wouldn't include a window which tells you that since you would have to close it every time when you are using eihort.
  Eihort should not only not require textures, there should be also an option to have some kind of a fallback texture. This could be used to if there was one face for many blocks and now there is a different texture for every face. I'm talking about the cut faces of wood. In snapshot 13w24a (1.6.1) every wood log got it's own cut face. If you are using eihort for pre 1.6 maps you will need an older version of eihort which does may not deliver some new features that you want to use. I guess the whole thing can be included pretty easy with something like if (file_exist (stone.png)); This would be integrted in the first module and if there is texture b.png use it, if not use texture a.png. Another example would be if the nether get's a new type of block and it's called “hellrock”. Netherrack is so far stored in “hellrock.png”. So there would be again a break which version of eihort can be used with which version of minecraft. The solution in such a case could look like this: if (file_exist (netherrack.png)) use netherrack.png; else use hellrock.png

- More blockvalues
  In 0.3.14 the block value was increased. This is a good idea and would have been suggested by me as well. We need to wait for the mod API but I think there is no more need for a block count limitation since nobody knows which IDs maybe vanilla will use in the future.