



This repository Search

Pull requests Issues Gist



openstreetmap / iD

Watch 111 Unstar 1,099 Fork 460

Code Issues 191 Pull requests 15 Wiki Pulse Graphs

# Indoor mode #3097

Edit

Open zbycz wants to merge 39 commits into openstreetmap:master from zbycz:indoor-mode

Conversation 11 Commits 39 Files changed 39

+756 -27



zbycz commented 9 days ago • edited



Hi,

I am submitting a little larger pull request. I am not sure if its possible to be merged as is, but I did my best and will be happy to cooperate further to make it so.

I've been engaged in OpenStreetMap since 2008 and for the time being I've led many [workshops](#), written some school works and worked on getting the Czech community together on [openstreetmap.cz](#). Last summer I had to come up with a topic for my final engineer thesis, so I decided to use the effort already invested in indoor mapping and chose to update the iD editor accordingly. Let me elaborate...

Current work-in-progress proposal is called [Simple Indoor Tagging](#). I didn't want to rely on WIP, so I realized there is already an accepted solution – the `level=*` tag.

There are two challenges for indoor:

1. making underlying background map (this is addressed by Simple Indoor's `indoor=*` area features)
2. display ways – the accessibility of places – and this kind of was the nature of OSM from the beginning. Until we had all the `landuse=*` areas, OSM started with highways + pois.

Drawing `highways=*` for indoor feels very natural – corridors usually repeats every floor even in multi-storey buildings, it displays clear accessibility information, can be already used for routing and also works with parking highways. The logic is also very simple - when a `level=*` feature is in viewport - show “Indoor” button, and let the user filter on these.

**Several examples** (my build):

- [Shopping mall in Prague](#) - underground parking, main foot passages and few shops POIs
- [University building in Aachen](#) - complete room plans with Simple Indoor (older proposal version with optional `level=*` xor `repeat_on=*`). Another building [here](#).
- [Berlin main railway station](#) - several underground floors + `indoor=*` features
- [University campus Prague](#) - mapping in progress ;-)

Well, i think I made my point - let's discuss the technical background :-)

## Some thoughts / questions:

1. Is it ok, to call it indoor *mode*? It doesn't fit with current “modes” terminology, where modes are exclusive.
2. I think the philosophy of iD is to show disabled buttons for “available” actions. I chose to hide the button completely, where there isn't any `level=*` features on screen. Would you agree?
3. Current `level=*` allows only separate values by semicolon, I borrowed three useful tagging from SimpleIndoor. Is it ok to use these? Should i maybe make new official proposition?
  - ranges – `level=0-12` or `-3--1` for features spanning several floors.
  - `repeat_on` – for repeating the same (possibly multilevel) feature on other floors - ie. doors,toilets
  - `min/max_level` – when set on building, the building spans through these levels. It helps as a visual cue, and also for the renderers.
4. I had to unclip the building-outline shape, because a building completely covered with `indoor=*` areas, couldn't be clicked anyway. Kind of addresses the issue [#2225](#).
5. I submitted my original development commits, rebased to current master. Is it appropriate to squash the commits into one or few?
6. Although I tried to abide the current code style and wrote some test, tell me if i should change something globally. Easiest would be to comment the code in the specific commits..

## Something to do in future

- if agreed, I would like to add level indicator as shown here <https://github.com/zbycz/iD/wiki/devnotes-zbycz> – but rather use another svg layer instead of filters.


- once proposition is accepted – enable switching over floating point level values (ie. `level=1.5` ). But even now - any string value could be written in the level chooser, so it “works”.
- add icon for ‘highway=elevator and indoor=corridor+stairs=yes` as these had to be distinguished easily.
- Unfortunately I found later, I developed very similar changes as Panier’s [id-indoor fork](#). Although he added many more indoor presets, which could be eventually merged here.
- Create a level switcher in osmbuildings.org 3D viewer ;-) .. well, distant future indeed.

















































Changelog until now:

- added `iD.ui.IndoorMode` control ui
- added filtering using `features.indoor_different_levels`
- added `.indoor-mode` class for svg surface
- added general preset for `indoor=*` when indoor-mode enabled
- added several map styles for enabled `.indoor-mode`
- added `indoor_levels` as general field for building, using `iD.ui.preset.range`

btw, it was a pleasure to work with iD codebase, also d3 is so very amazing :-)

1	
---	--

 **zbycz** added some commits on Mar 26

	 indoor: ui-control attempts	aa11306
	 indoor: ui-control works, needs refactor	24cfd5e
	 indoor: ui-control refactoring	57bead6
	 indoor: basic rendering	26c8a54
	 indoor: choose level of selected entitry	d67fafa
	 indoor: refactor event name	c9bf632
	 indoor: hide building label-icons (added classes)	5fde1b6
	 indoor: fix build	757306a
	 indoor: new objects with correct level	86a25e7
	 indoor: moved filter to features	493147d
	 indoor: fix hiding building label	c2a67fa
	 indoor: better indoor-mode control with +/-	c1be9e2
	 indoor: refactored filtering using features.indoor	ca7ed19
	 indoor: fix showing relevant points	439753c
	 indoor: fix default tags on new entities	7f53f2b
	 indoor: level=* without whitespace	bd8d92f
	 indoor: hidden lower buildings then "current level"	5f96d62
	 indoor: hide surroundings for level < 0	3cd9ef3
	 indoor: supress losing selection when exiting indoor-mode	04f19a1
	 indoor: level parameter in url	e4d5f58
	 indoor: hide "building" icon for amenity	1fbad7f
	 indoor: fix – correct hiding of vertexes	aced1a4
	 indoor: mandatory level, optional repeat_on ...	f2c9118
	 indoor: hide other buildings underground	dbee347
	 indoor: translations	1719fc2
	 indoor: styles for underground view	ddb92a
	 indoor: inspector – indoor_levels field + ui.preset.range	1309b18
	 indoor: refactoring context.indoor() to separate file	507878d
	 indoor: show button when level=* feature in viewport	50d4d8d
	 indoor: map redraw on event	b5309d3
	 indoor: eslint	ce1288f
	 indoor: refactoring + codestyle	6a8a3ad
	 indoor: style changes from user testing	4bacbdf
	 indoor: add tests	✓ ce6de5c

  **zbycz** referenced this pull request

haveing trouble stacking doors at a lift well (for indoor mapping) #2746

[Open](#)



bhouse1 commented 5 days ago

OpenStreetMap on GitHub member



Wow, impressive work @zbycz! I really like this and would like to see this merged in some form.

I'll need some time to think about how to present the UI in a way that doesn't clutter things up too much.

I'll try to answer your questions now:

Is it ok, to call it indoor mode? It doesn't fit with current "modes" terminology, where modes are exclusive.

I think it's ok to call it indoor mode. (People will call it that anyway). You're right it's not technically a "mode" like the other exclusive drawing modes.

I think the philosophy of iD is to show disabled buttons for "available" actions. I chose to hide the button completely, where there isn't any level=\* features on screen. Would you agree?

I'm leaning towards having the button smaller but available, maybe just an icon with tooltip, and it can transition open to reveal the indoor drawing controls when active. (I'm struggling with this same problem on #2699 - it's cool and useful, but where to put it?)

Current level=\* allows only separate values by semicolon, I borrowed three useful tagging from SimpleIndoor. Is it ok to use these? Should i maybe make new official proposition?

ranges - level=0-12 or -3--1 for features spanning several floors.

repeat\_on - for repeating the same (possibly multilevel) feature on other floors - ie. doors, toilets

min/max\_level - when set on building, the building spans through these levels. It helps as a visual cue, and also for the renderers.

The right thing to do is to make a proposal and loop in the OSM tagging list. At this point, indoor editing is not solidly established, so I think you have a good chance of pushing the community in a sane direction.

I had to unclip the building-outline shape, because a building completely covered with indoor=\* areas, couldn't be clicked anyway. Kind of addresses the issue #2225.

Makes sense.. Yes I agree that in "indoor mode" the building outline should not render as an area, as this covers up the indoor features.

I submitted my original development commits, rebased to current master. Is it appropriate to squash the commits into one or few?

It would be better to squash them into fewer commits, but I'm not super picky about this.

Although I tried to abide the current code style and wrote some test, tell me if i should change something globally. Easiest would be to comment the code in the specific commits..

I'll try to review it in more detail, but what I saw at a quick glance looks great! And the tests are much appreciated...



zbycz commented 4 days ago



Thanks, @bhouse1 !

I will make the proposal next week, in the meantime I am looking forward to your code comments      Once we're done, I will squash the commits in fewer.

**ad button positioning**) the more buttons, the more the user think how complex the software is (aka Norman: The design of everyday things) - and I think iD should stay "simple". But also its a bad practise for controls to disappear unexpectedly.

I completely agree with a smaller icon expanding to full control. What about placing Indoor-mode button in a dropdown menu "Tools" on the right of the top toolbar. It could be left closed forever, or left toggled opened. It would be a nice drawer for more useful tools in the future (including the #2699).



bhouse1 commented 3 days ago

OpenStreetMap on GitHub member



What about placing Indoor-mode button in a dropdown menu "Tools" on the right of the top toolbar. It could be left closed forever, or left toggled opened. It would be a nice drawer for more useful tools in the future

I've been playing around with making the UI more responsive and supporting something like a "drawer" for more commands or options on existing commands.

I don't really like it.. Maybe @samanpwbb has ideas?


The problem is it either bumps down the other things in the layout, or it overlaps them. So we'd either need to 1. position the other UI stuff differently or 2. be ok about having them move (eh) or 3. something else.








PanierAvide commented 1 day ago



I'm glad to see the opportunity of integrating indoor editing in main iD version. Feel free to reuse any custom part of iD-indoor, like presets or [level parser](#) (which handles lot of level-related tags) ;)

 **zbycz** added some commits

-  indoor: fix decimal levels regexp + test 3f1803b
-  indoor: tagging - repeat\_on means discreet values 3f5097b
-  indoor: refactor inRange() to indoor file a57c92e
-  indoor: switching floors using decimal values (if exist) 5edb261
-  indoor: build english and presets ✓ b37ccca

