

Final Report: XMonad

Practical Course - Contributing to an Open-Source Project
Chair for Logic and Verification
Department of Informatics
Technical University of Munich

Yecine Megdiche

March 16, 2021

Contents

1	Introduction	3
2	Preliminary report: community feedback	3
3	My Contributions and interaction with the community	4
3.1	Communication	4
3.2	Contribution summary	5
3.2.1	Refactoring the test-suite	5
3.2.2	Improving some documentation	5
3.2.3	Adding a fix for Java applications	5
3.2.4	Adding Screen specific loggers	6
3.2.5	Giving feedback to other contributions	6
3.2.6	Working on the interaction with status bars	6
3.3	In-depth look: Status bars	7
3.3.1	Status Bar support: the released version	7
3.3.2	Creating a new module: <code>XMonad.Hooks.StatusBar</code>	7
4	Feedback to the xmonad community	10
5	Conclusion and final comments	12
A	Pull Request Summary	20

1 Introduction

Xmonad is a dynamically-tiling X11 window manager that is written and configured in Haskell [1]. Being a window manager, it controls how the windows are placed on the screen and what they look [2]. It is designed to be lightweight, minimal, and extremely customizable [1]. The main characteristic of xmonad, being tiling, means that the windows are placed in non-overlapping tiles. Xmonad does the tiling *dynamically*, meaning according to predefined rules and layouts. Figure 1 shows my personal machine running xmonad.

The project was announced in April 2007 [3], and its development is still active. There are two main parts of the project: The first one is `xmonad` itself, which is the core of the window manager. It provides the bindings to the X11 system, as well as some minimal usage and customization examples. It is quite stable and does change much (contributions are generally only bug fixes; no new features are added) [4]. The second part is `xmonad-contrib`, which is the library of third party extensions to xmonad. It is much more active and has significantly more modules. Contributions here are much more encouraged [5].

The xmonad project has a flat governance structure: there is a team of maintainers who take care of reviewing pull-requests and managing releases. An IRC channel¹ and a less active mailing list² are used for communication [6].

2 Preliminary report: community feedback

After publishing the preliminary report [6], I (IRC: `mc47`) sent it in the IRC channel to get feedback from the community. The maintainers agreed with the points mentioned about the website being outdated, and we had a discussion on whether alternatives to the IRC channel should be provided. I wanted to use a more modern communication platform, such as Zulip, and my argument was that I could not see what was discussed when I was offline. After some back and forth, we settled on providing publicly accessible conversation logs on `IRC Browse`.³ The main argument against using more modern platforms was that it is less convenient for the maintainers, and that there were not many complaints with using IRC. The logs solved my issue, as accessing the conversation was now possible [7].

¹`#xmonad @ chat.freenode.org`

²<https://mail.haskell.org/cgi-bin/mailman/listinfo/xmonad>

³<https://ircbrowse.tomsmeding.com/browse/xmonad>

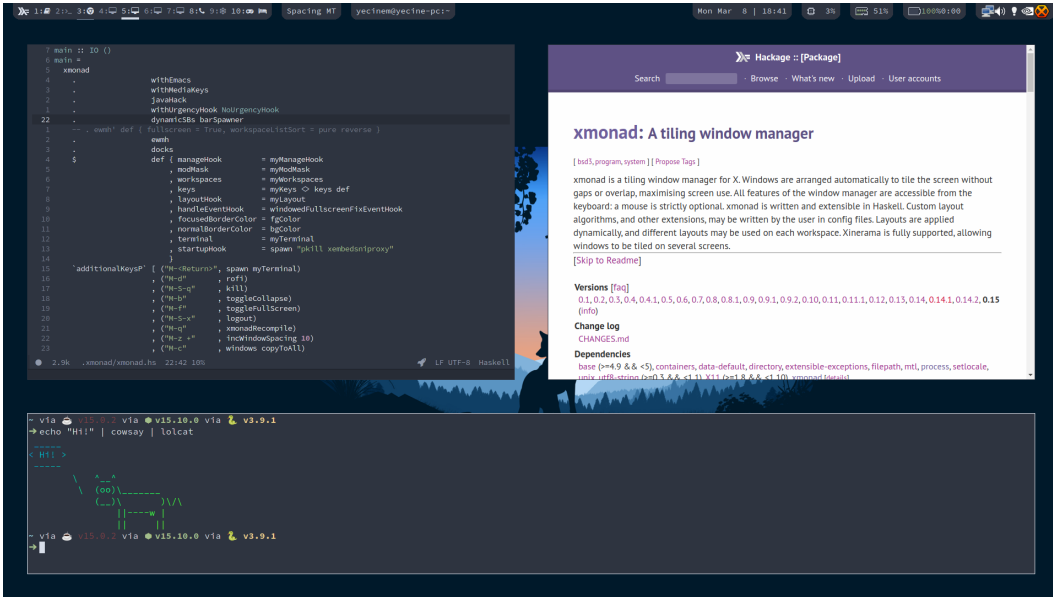


Figure 1: Xmonad. Visible programs: xmbarr and trayer (top), doom emacs (left), chromium (right) and termit (bottom)

3 My Contributions and interaction with the community

3.1 Communication

The communication with the community and the maintainers happened on the IRC channel and on GitHub⁴ through issues and pull requests. GitHub was used for asynchronous communication, such as reviewing pull requests and discussing issues. On the IRC channel, I tried to first discuss any non-trivial changes in code before implementing them [8]. It is also used to follow up on GitHub discussions, when more community feedback is needed: GitHub conversations sometimes became too long, and it was unlikely to get a third opinion there.⁵ Along with more experienced users and maintainers, I also tried to help new users debug their configurations or answer their questions. Due to the small community and the low volume of messages on the IRC, it had casual feeling: people just chatted about Haskell, open-source, and random topics.

⁴My nickname: TheMC47 <https://github.com/TheMC47>

⁵The discussion in the pull-request #443 for example was continued on the IRC Channel, since the comments were too long for anyone else to follow and provide an opinion. https://github.com/xmonad/xmonad-contrib/pull/443#discussion_r565684461

3.2 Contribution summary

In total, I submitted 10 pull requests and opened one issue on `xmonad/xmonad-contrib`, and one issue and one pull request on `xmonad/xmonad`. Two pull requests are still in review and the rest have been merged. On average, it took 12.6 days for a pull request to be merged⁶ and a 1.6 to get a first review. The longest conversation on GitHub was on the pull request #443 with 89 comments [9], and on average, each pull request had 21.1 comments. Not including myself, the average pull request had 2 participants, peaking at 4 participants in #434 [10]. Appendix A shows a complete overview of all the pull requests submitted.

3.2.1 Refactoring the test-suite

Some modules in `xmonad-contrib` came with their own tests, but it was unclear how to run them [11]. The tests used different (outdated) versions of QuickCheck, which made them incompatible. They were also not part of the CI. I wanted to create a proper test-suite, and I discussed my plan in the IRC channel [8]. I then tried to fix these problems and make the tests to work well together [12]. I coordinated with the maintainer Sibi Prabakaran, who was simultaneously working on the CI [13], to make sure the tests were passing. Since the tests became easily runnable and integrated in the development pipeline, they were more visible and got more attention: for example, the maintainer Tomáš Janoušek optimized some tests and improved the test-suite [14].

3.2.2 Improving some documentation

The contribution instructions as well as the list of maintainers needed some improvement: there were some dead links in the list of maintainers, as well as important contribution resources that were not linked.⁷ I opened an issue to address these problems, and later updated the missing links and added instructions on how to run the tests [15, 16]. The information was also corrected afterwards by Tomáš Janoušek in #257 [17].

3.2.3 Adding a fix for Java applications

It is known that Java GUI applications misbehave with `xmonad` and other non-reparenting window managers [18]. Solutions are known and described in the `xmonad` wiki [19] but they were considered “too hacky” to be included as a library function. The `XMonad.Util.Hacks` module was created to contain similar workarounds, so that they are more visible, discoverable and easier to add to one’s configuration

⁶For the unmerged pull requests, the date of the submission of the report, the 16th of March 2021, is assumed to be the date of the merge.

⁷In particular, the documentation for `XMonad.Doc.Developing` <http://hackage.haskell.org/package/xmonad-contrib-0.16/docs/XMonad-Doc-Developing.html>

[20]. Hence, I added a function `javaHack` that provides one of the known fixes to this problem [21]. Before submitting this issue, I asked if it was sufficient for most scenarios, and Brandon Allbery (IRC: `geekosaur`), one of the maintainers, explained the origin of the bug [22].

3.2.4 Adding Screen specific loggers

Xmonad communicates with external programs (like status bars) when its state changes (for example, changing the focused window, or moving to a new workspace) [23, 24]. The information sent to these programs can be configured and extended in many ways, one of which is through `Loggers` [25]. The provided loggers in `XMonad.Util.Loggers` were not suitable for multiple screens, as they do not track which screen is currently active. I added screen-specific loggers that only log the information of the screens they are bound to in the pull request #448. That pull request also provided a way to combine loggers and a minor fix in the documentation.[26]

3.2.5 Giving feedback to other contributions

On two occasions, I provided feedback on contributions that I did not author. The first was #471, which fixes encoding issues that broke my personal configuration⁸ [27]. The second was the Pull Request #399, where the maintainer Tomáš Janoušek wanted some comments on the usability and the documentation of his work from a user's perspective.[28, 29]

3.2.6 Working on the interaction with status bars

This was the biggest contribution by far and it spanned many pull requests and issues. This will be discussed in more detail in the next section; however, here is a quick summary:

- Implemented restarting status bars that use property-based logging [30, 10, 31]
- Applied linter suggestions to `XMonad.Hooks.DynamicLog` [32]
- Added a new module `XMonad.Hooks.StatusBar` with support for multiple dynamic status bars, which replaces the functionality in `XMonad.Hooks.DynamicLog` and `XMonad.Hooks.DynamicBars` [9, 33, 34]

This contribution provides a new way for xmonad users to interact with status bars, which was only possible through the valuable feedback from and discussions with the community.

⁸<https://github.com/xmonad/xmonad-contrib/pull/471#issuecomment-782709583>

3.3 In-depth look: Status bars

3.3.1 Status Bar support: the released version

It is useful to use `xmonad` along with status bars, like `xmobar`⁹ or `dzen`¹⁰. These programs offer a way to visualize the state of the window manager: how many workspaces are used? Which workspace is active? What is the title of the focused window? etc.[24] `Xmonad` provides a configurable `logHook` that is executed whenever its state changes. To communicate with these programs, there are two methods: either use pipes to write to the standard input, or use X11 property-logging.¹¹ Pipes are known to be error-prone [35], so using property logging is preferred [36].

Interacting with these programs is facilitated with the module `XMonad.Hooks.DynamicLog` [24]. It offers a pretty-printing abstraction through the `PP` type, which facilitates creating a string representation of the window manager’s state. The module also provides a helper function, `statusBar`, which takes care of launching a status bar and configuring the `logHook`.

It is a useful module, but rather limited: only one status bar with pipe-based logging is supported in the released version. Code 1 shows a sample configuration file demonstrating how users typically use multiple status bars with the released version.¹² Another module called `XMonad.Hooks.DynamicBars` offers support for dynamic status bars, that respond to changes in screen configuration (e.g. when a new screen is plugged in) [37]. It has similar problems as `XMonad.Hooks.DynamicLog`: only one status bar with pipe-based logging per screen. It also needs better documentation: there were no examples included.

3.3.2 Creating a new module: `XMonad.Hooks.StatusBar`

Getting familiar with the code

I expressed in the IRC channel that I wanted to work on bringing multiple status bar support to `xmonad`. The maintainer `slotThe` (IRC: `Solid`) mentioned `XMonad.Hooks.DynamicBars`, which I found hard to use and was not exactly what I looked for. Other community members mentioned some popular configs that use “the manual” way described previously, which I also did not prefer.¹³ `slotThe` mentioned an issue they were working on and asked if I had any ideas: they previously added property-based support for `XMonad.Hooks.DynamicLog`, but there was a problem with restarting the status bars with `xmonad` [36]. At a later point, I discussed with Tomáš Janoušek how we can tackle this problem and I then summarized our discussion in

⁹<https://github.com/jaor/xmobar>

¹⁰<https://github.com/robm/dzen>

¹¹<https://tronche.com/gui/x/xlib/window-information/properties-and-atoms.html>

¹²A popular configuration is the one from Derek Taylor, a content creator who made videos about `xmonad` on his YouTube channel `DistroTube`, can be found here <https://gitlab.com/dwt1/dotfiles/-/blob/a8a3ee798c6a4687f89b0ad1dd974b04d570f057/.xmonad/xmonad.hs#L882>

¹³Attached IRC Logs

```

-- Imports not shown
pp1, pp2, pp3 :: PP -- The pretty printing options
-- definitions omitted

-- Commands to launch the status bars
-- Two status bars on the main screen, one on the secondary
mainTop, mainBottom, secondary :: String
-- definitions omitted

main :: IO ()
main = do
  -- launching the bars, and getting handles to their
  -- standard input
  mainTopH    <- spawnPipe mainTop
  mainBottomH <- spawnPipe mainBottom
  secondaryH  <- spawnPipe secondary
  -- launching xmonad
  xmonad def
    -- setting the logHook to print to each bar
    {logHook = dynamicLogWithPP pp1{ppOutput = hPutStrLn mainTopH}
      >> dynamicLogWithPP pp2{ppOutput = hPutStrLn mainBottomH}
      >> dynamicLogWithPP pp3{ppOutput = hPutStrLn secondaryH}
    }

```

Listing 1: 3 Status bars, manual approach

the issue #433 [30]. I wanted to work on this issue, since it seemed like a good first issue: The difficult part was finding a suitable approach, the implementation was straightforward. Tomáš and `slotThe` also kindly provided code snippets from their personal configurations that were used to do something similar.^{14, 15}

With that, I submitted the pull request #434 [10]. After their review, `slotThe` found a problem with the initial approach. There was a long discussion on GitHub where different approaches were suggested by different contributors and maintainers, after which we settled on a working approach.¹⁶ Through this pull request, I got more familiar with the code base, especially how to extend the state of the `xmonad` and how the utilities in `XMonad.Hooks.DynamicLog` work.

Creating a new interface

In the git version of `xmonad-contrib`, other functions were added to `XMonad.Hooks.DynamicLog` to support property-based logging. By now, there were 4 ways to add status bars, which were not composable with each other. Moreover, they all had a hardcoded action to toggle the visibility of the status bar: This was to provide a similar interface as the released `statusBar` function [38]. I suggested we add another function that supports customizable actions. After discussing it with the maintainers `slotThe`, who added the functions supporting property-based communication, and Tomáš Janoušek, we settled on creating an abstraction over status bars that can be easily customized, instead of adding more parameters to the existing functions [39]. It is worth noting that the response time was quick, which facilitated the exchange of ideas.

I used this idea and submitted the pull request #443. A consequence of this abstraction was that the support for multiple status bars was easier to add, since status bars are now composable. This meant that `xmonad` users can now have multiple status bars with less boilerplate. The pull request got a first review from `slotThe` two days after its submission [9]. Two weeks after the submission, Tomáš Janoušek left another review. We had a long discussion on GitHub to try and improve the interface: my first implementation tried to mimic to existing functions, which negatively affected the composability of the status bars¹⁷. The final implementation hides the unnecessary details like setting up the `logHook` and starting the status bars. I also noticed that this new abstraction provides a way to create dynamic status bars, like `XMonad.Hooks.DynamicBars`, that are more flexible and easier to use. I submitted the pull request #463 as a draft (as it needed #443 to be merged first) to show how this could be implemented [33]. Tomáš Janoušek provided a first review for this, even though it was still a draft. Listing 2 shows the new way to have multiple

¹⁴<https://github.com/xmonad/xmonad-contrib/issues/433#issuecomment-744838991>

¹⁵IRC Solid

¹⁶<https://github.com/xmonad/xmonad-contrib/pull/434#issuecomment-751517793>

¹⁷In more details: my first attempt included a keybinding to toggle the status bar, and keybindings cannot be composed. This meant we needed special functions to combine the status bars. By dropping the keybinding, composition became natural and `Data.Monoid` could be used.

dynamic status bars, with no boilerplate.

```
-- definitions from Listing 1 omitted

-- A user defined function from screen
-- identifiers to status bars.
bar :: ScreenId -> IO StatusBarConfig
-- Two status bars on screen 0...
bar 0 = statusBarPipe mainTop (pure pp1)
      <> statusBarPipe mainBottom (pure pp2)
-- ... one on screen 1...
bar 1 = statusBarPipe secondary (pure pp3)
-- ... and nothing for the rest
bar _ = mempty

main :: IO ()
-- Dynamic status bars, with no boilerplate!
main = xmonad . dynamicSBs bar $ def
```

Listing 2: #463 Dynamic status bars through the new interface

After the final implementation was reviewed, Tomáš Janoušek suggested giving the new interface its own module `XMonad.Hooks.StatusBar`, and revert `XMonad.Hooks.DynamicLog` to its release state [40]. We discussed the details in the IRC channel with Brandon Allbery (IRC: `geekosaur`), `slotThe` (IRC: `Solid`), and Tomáš Janoušek, who posted a summary of the discussion on GitHub.¹⁸ Finally I submitted the pull request #465 to make the necessary changes, which is still in review at the time of writing [34]. I also asked new users to try the new interface and give me feedback when they asked about “low hanging fruits” (e.g. issues that are easy for beginners) that they could work on [41].

4 Feedback to the xmonad community

The xmonad community and maintainers are open to new suggestions and ideas: they were open to discuss alternative communication platforms after reading my preliminary report, and decided to start logging the IRC conversations.¹⁹ One maintainer, Tomáš Janoušek, tried other communication platforms like Discord, Matrix and Slack, to see if there are any xmonad communities there [42]. New users also regularly get help there and the community is rather friendly and welcoming.

¹⁸<https://github.com/xmonad/xmonad-contrib/pull/443#issuecomment-777041434>

¹⁹<https://ircbrowse.tomsmeding.com/browse/xmonad>

There is also a motivation to work on updating the website, and create newer guides on how to use the window manager [43, 44]. Moreover, the maintainer Tomáš Janoušek recently got access to the Twitter-account²⁰ and the xmonad subreddit²¹ from the old maintainers, which is a positive step [45, 46]. The response time on the IRC channel is also short enough that it allows for chatting. For example, I recently wanted to debug a problem I was having, and got feedback and workarounds quickly [47].

However, there is room for improvement. Xmonad users could benefit from more regular releases: the Git version of xmonad contains bug fixes in both the core package [48] and the contrib package [49], as well as new features and modules that are being asked for by users [50]. Having more frequent releases means that users will not have to build the xmonad packages from source, which would be more convenient.

As mentioned in the preliminary report, xmonad has no clear governance structure. In particular, there is no clear definition of the roles of the different maintainers, or how to solve conflicts, should they ever arise. This was unfortunate, as one such conflict happened between two maintainers: Peter Simons, opposed a change made Tomáš Janoušek.²² The exchange escalated, and Peter Simons then submitted another Pull Request and merged it without first discussing it with anyone [51]. This caused more problems and other maintainers did not appreciate this behavior, and protested merging the change.²³ It was unclear how the issue would be resolved, and two senior maintainers, namely Brandon S. Allbery (GitHub: @geekosaur) and Brent Yorgey (GitHub: @byorgey) were asked to intervene. Due to personal circumstances, Brent Yorgey could only respond a month after being mentioned [52], and he later agreed that the behavior of Peter Simons was unjustified, and removed him from the core xmonad team [53]. This exchange was demotivating for Tomáš Janoušek, and he needed time off the project. That was unfortunate, since he is one of the few maintainers who still actively review pull requests and submit patches to xmonad. He even apologized to me for not being able to review my pull requests [54]. Fortunately, after almost 3 months since the start of the issue, both maintainers had a call and resolved the conflict, and could find a middle ground for their initial technical disagreement [55]. This is an example of a problem that might face an open-source community, and the absence of communication rules and conflict-resolution processes make overcoming them harder.

The xmonad project also has a low “bus factor”.²⁴ I mainly got feedback regarding my pull requests from Tomáš and slotThe, with Brandon Allbery providing explanations and help on the IRC channel. By the end of the semester, getting feedback was slower, because the mentioned maintainers were busy. This might be discouraging for newer contributors who do not use IRC, for example, and do not know that the

²⁰<https://twitter.com/xmonad>

²¹<https://www.reddit.com/r/xmonad/>

²²<https://github.com/xmonad/xmonad/pull/260#pullrequestreview-564693631>

²³<https://github.com/xmonad/xmonad/pull/261#issuecomment-757527545>

²⁴The bus factor is defined as the number of people that need to be hit by a bus, for a project to stop or stall. [56]

maintainers are busy. Having more active maintainers would not only mean that pull requests and issues get feedback sooner, but also that the more neglected parts (like the website, assuring more frequent releases) get more attention, thereby increasing the bus factor.

Adding to that, there are no issues that are labeled as “Good First Issue” or “Low Hanging Fruit”. This might discourage newer users from contributing, as they do not know which issues to start on. I made the mistake of submitting an easy pull request towards the end of the semester [21], which might have been a good candidate for a “good first issue” suitable for potential new contributors, as it only took a couple of minutes to implement. A list of good first issues will not be created without effort: maintainers and contributors must make sure to create suitable candidates and resist working on them themselves. They should also only consider issues that they themselves would enjoy working on to make sure they are interesting enough. Even if this is time consuming, the reward of gaining a new motivated contributor makes it worth while [56].

There are also a lot of forgotten contributions. Interesting features were submitted a couple of years ago, and the review process either stopped or did not begin. One such contribution was an attempt to re-design²⁵ the xmonad-website²⁶, that did not get enough feedback or reaction from the maintainers [57, 58]. However, the maintainers are trying to follow-up on some of these pull requests: the authors of some older contributions got pinged to see if they are still interested on continuing their work, and some even got another review and got merged [59, 60].

5 Conclusion and final comments

Xmonad still gets new users without any effort from the maintainers to promote the software on social media or other platforms. There are also waves of activities through the years: new users learn about xmonad somewhere and submit features or find bugs while creating their configuration. Once that is done, there is less motivation to continue contributing.

Through the practical course, I enjoyed working on xmonad, not only because it is one of the most important tools that I use all the time, but also because of the friendly and welcoming community and a codebase that is pleasant to work with. I am also grateful that I got to work on features and issues that had a visible consequence, like working on more loggers or creating the new interface to interact with status bars. One important lesson I learned through the course of my contributions, was that open-source development is less intimidating than it seems, and maintainers and communities are generally willing to provide help when asked.


Even though the window manager was released almost 14 years ago, there is still room for new features and content like guides and tutorials. I certainly plan to keep

²⁵<https://tsier.github.io/xmonad-web/index.html>

²⁶<https://xmonad.org/>

contributing to xmonad, and hopefully help new users set up their configuration and maybe even make their first open-source contribution to xmonad!

References

- [1] XMonad's Website. <https://xmonad.org/>.
- [2] Arch Linux Wiki. Window Manager. https://wiki.archlinux.org/index.php/window_manager, 2020. Online; accessed 15 March 2021.
- [3] Spencer Janssen , Don Stewart , and Jason Creigh. [Haskell-cafe] ANNOUNCE: xmonad 0.1. <https://mail.haskell.org/pipermail/haskell-cafe/2007-April/024754.html>, Apr 2007.
- [4] xmonad/xmonad Repository. GitHub. <https://github.com/xmonad/xmonad>.
- [5] xmonad/xmonad-contrib Repository. GitHub. <https://github.com/xmonad/xmonad-contrib>.
- [6] Yecine Megdiche. Preliminary Report: XMonad. *Practical Course – Contributing to an Open-Source Project*, Dec 2020.
- [7] IRC Exchange. See attached file .
- [8] Logs of the xmonad IRC at chat.freenode.org. 2020-12-04 09:00:30 mc47 "Hello! Is there a reason the test suite in xmonad-contrib is...". <https://ircbrowse.tomsmeding.com/browse/xmonad?id=301×tamp=1607072430#t1607072430>. Online; accessed 15 March 2021.
- [9] Yecine Megdiche. xmonad/xmonad-contrib, Pull Request 443: Refactor X.H.DynamicLog and multiple loggers support. <https://github.com/xmonad/xmonad-contrib/pull/443>.
- [10] Yecine Megdiche. xmonad/xmonad-contrib, pull request 434: Restarting status bars on restart. <https://github.com/xmonad/xmonad-contrib/pull/434>.
- [11] Ivan. xmonad/xmonad-contrib, issue 381: How to run automated tests? <https://github.com/xmonad/xmonad-contrib/issues/381>.
- [12] Yecine Megdiche. xmonad/xmonad-contrib, pull request 417: Refactor the test-suite. <https://github.com/xmonad/xmonad-contrib/pull/417>.
- [13] Sibi Prabakaran. xmonad/xmonad-contrib, pull request 425: Add github actions for test. <https://github.com/xmonad/xmonad-contrib/pull/425>.
- [14] Tomáš Janoušek. xmonad/xmonad-contrib, Pull Request 452: tests: Use hspecc as test driver (prep for #407 #430 #440 unit tests); speed up. <https://github.com/xmonad/xmonad-contrib/pull/452>.

- [15] Yecine Megdiche. xmonad/xmonad, Issue 248: Update MAINTAINERS.md.
<https://github.com/xmonad/xmonad/issues/248>.
- [16] Yecine Megdiche. xmonad/xmonad, Pull Request 253: Updated CONTRIBUTING.md and MAINTAINERS.md.
<https://github.com/xmonad/xmonad/pull/253>.
- [17] Tomáš Janoušek. xmonad/xmonad, Pull Request 257: Sync MAINTAINERS.md with reality on GitHub.
<https://github.com/xmonad/xmonad/pull/257>.
- [18] Arch Linux Wiki. Java, Troubleshooting, Gray window, applications not resizing with WM, menus immediately closing.
https://wiki.archlinux.org/index.php/Java#Gray_window,_applications_not_resizing_with_WM,_menus_immediately_closing.
Online; accessed 15 March 2021.
- [19] XMonad users' wiki. – Haskell Wiki. Xmonad/Frequently asked questions, Troubleshooting, Problems with Java applications, Applet java console.
https://wiki.haskell.org/Xmonad/Frequently_asked_questions#Problems_with_Java_applications.2C_Applet_java_console. Online; accessed 15 March 2021.
- [20] ElKowar. xmonad/xmonad-contrib, issue 403: Add new module xmonad.util.hacks (formerly xmonad.hooks.windowedfullscreenfix).
<https://github.com/xmonad/xmonad-contrib/pull/403>.
- [21] Yecine Megdiche. xmonad/xmonad-contrib, Pull Request 464: Added javaHack.
<https://github.com/xmonad/xmonad-contrib/pull/464>.
- [22] Logs of the xmonad IRC at chat.freenode.org. 2021-02-08 17:04:10 geekosaur "mc47, the LG3D hack is specific to Oracle Java...".
<https://ircbrowse.tomsmeding.com/browse/xmonad?id=15829×tamp=1612803850#t1612803850>. Online; accessed 15 March 2021.
- [23] XMonad.Core Documentation. Hackage: The Haskell Package Repository.
<https://hackage.haskell.org/package/xmonad-0.15/docs/XMonad-Core.html#v:logHook>. Online; accessed 15 March 2021.
- [24] XMonad.Hooks.DynamicLog Documentation. Hackage: The Haskell Package Repository. <https://hackage.haskell.org/package/xmonad-contrib-0.16/docs/XMonad-Hooks-DynamicLog.html>. Online; accessed 15 March 2021.
- [25] XMonad.Util.Loggers Documentation. Hackage: The Haskell Package Repository. <https://hackage.haskell.org/package/xmonad-contrib-0.16/docs/XMonad-Util-Loggers.html>. Online; accessed 15 March 2021.

- [26] Yecine Megdiche. xmonad/xmonad-contrib, Pull Request 448: More loggers: A ‘maybe’-like combinator, and screen-specific loggers.
<https://github.com/xmonad/xmonad-contrib/pull/448>.
- [27] Tomáš Janoušek. xmonad/xmonad-contrib, Pull Request 471: Fix encoding issues in spawnPipe/dynamicLogString/xmonadPropLog.
<https://github.com/xmonad/xmonad-contrib/pull/471>.
- [28] Logs of the xmonad IRC at chat.freenode.org. 2021-03-01 09:20:07 Test 399.
<https://ircbrowse.tomsmeding.com/browse/xmonad?id=20166×tamp=1614590407#t1614590407>. Online; accessed 15 March 2021.
- [29] Tomáš Janoušek. xmonad/xmonad-contrib, pull request 399: (wip) refactor ewmh window activation.
<https://github.com/xmonad/xmonad-contrib/pull/399>.
- [30] Yecine Megdiche. xmonad/xmonad-contrib, issue 433: Restart xmobar/dzen without relying on pipes.
<https://github.com/xmonad/xmonad-contrib/issues/433>.
- [31] Yecine Megdiche. xmonad/xmonad-contrib, Pull Request 439: Added cleanupStatusBars to the startup-hook.
<https://github.com/xmonad/xmonad-contrib/pull/439>.
- [32] Yecine Megdiche. xmonad/xmonad-contrib, Pull Request 449: Cleanup: XMonad.Hooks.DynamicLog.
<https://github.com/xmonad/xmonad-contrib/pull/449>.
- [33] Yecine Megdiche. xmonad/xmonad-contrib, Pull Request 463: Dynamic Status Bars support for XMonad.Hooks.StatusBar.
<https://github.com/xmonad/xmonad-contrib/pull/463>.
- [34] Yecine Megdiche. xmonad/xmonad-contrib, Pull Request 465: Split XMonad.Hooks.DynamicLog.
<https://github.com/xmonad/xmonad-contrib/pull/465>.
- [35] Tomáš Janoušek. xmonad/xmonad-contrib, pull request 400: X.h.dynamiclog: Postpone statusbar start until after xmonad is recompiled.
<https://github.com/xmonad/xmonad-contrib/pull/400>.
- [36] slotThe. xmonad/xmonad-contrib, issue 408: X.h.dynamiclog: Start recommending property-based logging.
<https://github.com/xmonad/xmonad-contrib/pull/408>.
- [37] XMonad.Hooks.DynamicBars Documentation. Hackage: The Haskell Package Repository. <https://hackage.haskell.org/package/xmonad-contrib-0.16/docs/XMonad-Hooks-DynamicBars.html>. Online; accessed 15 March 2021.

- [38] CHANGES.md: unkown release, Bug Fixes and Minor Changes, XMonad.Hooks.DynamicLog. xmonad/xmonad-contrib, commit 44c5759. <https://github.com/xmonad/xmonad-contrib/blob/44c575930aef0475a5a0140b7399da10990a59db/CHANGES.md>.
- [39] Logs of the xmonad IRC at chat.freenode.org. 2020-12-28 13:31:41 another makeStatusBar. <https://ircbrowse.tomsmeding.com/browse/xmonad?id=6372×tamp=1609162301#t1609162301>. Online; accessed 15 March 2021.
- [40] Logs of the xmonad IRC at chat.freenode.org. 2021-02-10 18:26:55 XMonad.Hook.StatusBar. <https://ircbrowse.tomsmeding.com/browse/xmonad?id=16346×tamp=1612981615#t1612981615>. Online; accessed 15 March 2021.
- [41] Logs of the xmonad IRC at chat.freenode.org. 2021-03-03 20:11:41 mc47 "If you want, you could give <https://github.com/xmonad/xmonad-contrib/pull/465> a try ". <https://ircbrowse.tomsmeding.com/browse/xmonad?id=20926×tamp=1614802301#t1614802301>. Online; accessed 15 March 2021.
- [42] Logs of the #xmonad IRC @ chat.freenode.org. 2021-02-09 18:24:55 Liskni_si "mc47: btw I experimented with the "modern" communication...". <https://ircbrowse.tomsmeding.com/browse/xmonad?id=16023×tamp=1612895095#t1612895095>. Online; accessed 15 March 2021.
- [43] Logs of the xmonad IRC at chat.freenode.org. 2021-02-22 15:42:54 Solid "this really motivates me to continue working on that tutorial". <https://ircbrowse.tomsmeding.com/browse/xmonad?id=19136×tamp=1614008574#t1614008574>. Online; accessed 15 March 2021.
- [44] Logs of the #xmonad IRC @ chat.freenode.org. 2021-02-09 18:32:51 iskni_si "I think the most impactful thing we could do is..". <https://ircbrowse.tomsmeding.com/browse/xmonad?id=16043×tamp=1612895571#t1612895571>. Online; accessed 15 March 2021.
- [45] Logs of the xmonad IRC at chat.freenode.org. 2021-01-30 14:10:37 Liskni_si "good news btw, I pinged dons thru twitter...". <https://ircbrowse.tomsmeding.com/browse/xmonad?id=13941×tamp=1612015837#t1612015837>. Online; accessed 15 March 2021.
- [46] Logs of the xmonad IRC at chat.freenode.org. 2021-02-17 20:00:45 Liskni_si "good news: dons recovered password to the @xmonad...". <https://ircbrowse.tomsmeding.com/browse/xmonad?id=18324×tamp=1613592045#t1613592045>. Online; accessed 15 March 2021.
- [47] Logs of the #xmonad IRC @ chat.freenode.org. 2021-03-08 16:06:55 mc47 "Does anyone have an idea why YouTube videos, when...".

- <https://ircbrowse.tomsmeding.com/browse/xmonad?id=22116×tamp=1615219615#t1615219615>. Online; accessed 15 March 2021.
- [48] slotThe. xmonad/xmonad, Pull Request 258: Revise XDG handling. <https://github.com/xmonad/xmonad/pull/258>.
- [49] slotThe. xmonad/xmonad-contrib, Pull Request 441: Revise XDG handling. <https://github.com/xmonad/xmonad-contrib/pull/441>.
- [50] Logs of the #xmonad IRC @ chat.freenode.org. 2021-02-03 21:34:06 <endiruna> "hi! new to xmonad. coming from i3. is there...". <https://ircbrowse.tomsmeding.com/browse/xmonad?id=14878×tamp=1612388046#t1612388046>. Online; accessed 15 March 2021.
- [51] Peter Simons. xmonad/xmonad, Pull Request 261: stack.yaml: switch from Stackage LTS to Nightly. <https://github.com/xmonad/xmonad/pull/261>.
- [52] Logs of the #xmonad IRC @ chat.freenode.org. 2021-01-26 17:25:25 byorgey "Liskni_si, Solid: how can I help?". <https://ircbrowse.tomsmeding.com/browse/xmonad?id=12933×tamp=1611681925#t1611681925>. Online; accessed 15 March 2021.
- [53] Logs of the xmonad IRC at chat.freenode.org. 2021-02-20 10:52:09 <byorgey> "I just removed...". <https://ircbrowse.tomsmeding.com/browse/xmonad?id=18812×tamp=1613818329#t1613818329>. Online; accessed 15 March 2021.
- [54] Logs of the #xmonad IRC @ chat.freenode.org. 2021-01-12 14:03:05 Liskni_si "mc47: I probably won't get to taking a look any time soon". <https://ircbrowse.tomsmeding.com/browse/xmonad?id=9783×tamp=1610460185#t1610460185>. Online; accessed 15 March 2021.
- [55] Logs of the #xmonad IRC @ chat.freenode.org. 2021-03-06 10:07:44 Liskni_si "good news everyone". <https://ircbrowse.tomsmeding.com/browse/xmonad?id=21729×tamp=1615025264#t1615025264>. Online; accessed 15 March 2021.
- [56] Stephan Kulla. Motivation in Open Source Projects. Personal Communication, as a part of the Open Source Lab, The Chair for Logic and Verification at the Technical University of Munich, <https://www21.in.tum.de/teaching/osp/WS20/index.html>, Dec 2020.
- [57] Wisnu Adi Nurcahyo. xmonad/xmonad-web, issue 5: Beautify xmonad official website. <https://github.com/xmonad/xmonad-web/issues/5>.
- [58] Wisnu Adi Nurcahyo. xmonad/xmonad-web, issue 6: Creates a special branch for hakyll application. <https://github.com/xmonad/xmonad-web/issues/6>.

- [59] Matt Kingston. xmonad/xmonad-contrib, pull request 222: Actions.easymotion feature added. <https://github.com/xmonad/xmonad-contrib/pull/222>.
- [60] Ivan Malison. xmonad/xmonad-contrib, pull request 121: X.a.dynamicworkspacenames: Add dynamic workspace names code. <https://github.com/xmonad/xmonad-contrib/pull/121>.

A Pull Request Summary

PR Number	Days till first review	Days till merged	Lines added	Lines removed	Comments	Number of participants
#253	1	2	33	17	5	3
#417	0	3	374	192	11	4
#434	0	12	64	5	53	5
#439	0	0	1	1	2	2
#443	2	30	224	41	89	4
#448	4	8	79	5	23	3
#449	0	0	12	13	0	2
#463	3	-	126	29	23	4
#464	2	2	23	1	0	2
#465	4	-	943	886	5	2