

GNOME UX Hackfest 2017: Notes

Non-exhaustive notes taken by Allan Day at the GNOME UX Hackfest in London, November 2017.

Table of contents

[Activities Overview](#)

[Fullscreen windows and window tiling](#)

[App picker](#)

[Window titlebars](#)

[First boot experience](#)

[Login/unlock](#)

[Goals](#)

[Constraints](#)

[Issues](#)

[Discussion](#)

[Next Steps](#)

Activities Overview

Main issues we identified:

- Window switching is old-fashioned. Most people are focused on apps; experience on mobile is a strong influence here.
- Current application launching design is sub-optimal:
 - The grid of launchers doesn't offer user control. App folders can't be manipulated.
 - As the initial "apps view", the launcher grid is overwhelming.
 - The button that launches the application view is ambiguous.
 - The grid doesn't facilitate or encourage exploration or discovery.
- The activities overview would benefit from a clearer spatial model.

Fullscreen windows and window tiling

Mac:

- Workspaces can be closed using an X button. This moves any windows that the workspace contains to the previous workspace.
- Each fullscreen window is given its own dedicated workspace.
 - No other windows are permitted on the workspace.
 - The workspace's close button is replaced by an "un-fullscreen" button - this un-fullscreen's the window, removes its dedicated workspace, and moves it to the previous workspace.
- How would this approach work with non-dynamic workspaces?
- Side-by-side windows are treated in the same way as fullscreen ones - but they must be fullscreened to be side-by-side.

Thumbnails aren't currently a good way to identify workspaces. (They would be good for fullscreen windows though.)

- Elementary OS shows an app icon for each window instead of a thumbnail. Could be a good approach?

App picker

Schema that we want to go for:

- Favourites that can be customized by the user, includes modifiable folders
- Suggested apps (can be smart about this - combine recent, frequent, day of the week, time of day, recently installed, etc)
- App basement - flat alphabetical grid of apps

Should there be a limit on the number of favourites:

- Jimmac - yes
- What about large screens, screen orientation changes? Could end up with a lot of space if there's a limit. Although, perhaps the grid should scale to fit the display.
- We seemed to settle on there not being a limit...

Paging vs scrolling

- Paging better suited to spatial memory. Hasn't worked so well for the app grid, due to the fact that it's a single long sequence - a change results in every subsequent app shifting position.
- Could use paging for favourites, where the order is only going to change due to user manipulation, and scrolling for the app basement. Favourites and the basement will need to be visually distinct in order for this to work.
- Remember that search results will need scrolling.
- Need to experiment with scaling the grid depending on the screen size - ideally needs to keep the pages the same irrespective of the display.

How to add an app to favourites that isn't installed?

- Ability to favourite an app from within Software?
- Automatically add newly installed apps to the favourites? (Probably not.)
- Link to Software from the favourites view?

Window titlebars

Issue - traditional window titlebars are too big, don't fit with GNOME design conventions. They're the norm for non-native apps.

Potential approach: non-native apps could use client-side decorations and implement window controls as part of their app windows.

- This would require that we provide a way for them to tell where to show their window controls, which controls to show, etc.
- Documentation and design guidelines for non-native apps would help.
- Next step - mockups for what non-native apps could look like on GNOME.
- It would be especially good to target Electron for this effort.

First boot experience

Endless have been doing work to improve their first run experience. Their idea is to use multiple approaches to educate users over their first 10 sessions or so:

- First session - short, dedicated tutorial which covers the 5 or 6 things that users really need to know

- Subsequent sessions - popups provide information as the user explores or events happen (example: when notifications have been waiting for the first time)
- Thereafter: help app and documentation

The tutorial serves an educational role, but it's also more than that.

- It sets the tone and establishes a relationship.
- It's also an opportunity to do marketing and branding, and to advertise features.

After first run people are often looking for something to do, particularly if it's a new device. Potential to link into common initial activities after first boot:

- Install apps
- Try a fun application (Cheese is a good example)
- Personalise your session - wallpaper, avatar, favourite apps, etc

New installs or user accounts don't always mean that it's a user who's new to GNOME:

- A first run tutorial requires an opt out
- Should stop showing information popups if several of them are dismissed in sequence

Login/unlock

Goals

- Protect privacy
- Protect from stray input
- Notifications - something is happening, what's happened
- Media controls
- Date and time (useful from a distance)
- Personalisation (wallpaper)
- Login as another user
- Low friction? (Get to the session without effort)
- Should be accessible
- Advertise and allow using guest session
- Usage scenarios:
 - One user
 - Small number of users

- Networked environments - includes shared office workstations with small numbers of users and labs/libraries with many
- Kiosks
- View system status
- Advertise available authentication methods - fingerprint, smartcards, PIN
- Integrate with disk encryption
- Don't use more power than necessary
- Show login messages from admins
- Allow selecting a session type
- Support autologin (no password)
- Be an effective brand touch point and defining microinteraction

Constraints

- Shouldn't be susceptible to brute force attacks. (This is why we wait after an unsuccessful login attempt; however this can be frustrating if it's an honest mistake. Using a delay after 3 unsuccessful login attempts might be better.)

Issues

- Discoverability of the shield
- Screen wake up when notifications come in - can be very bright, uses power, how useful is it?
- Not delightful, boring
- Inconsistency between unlock and login can be disorientating
- Lack of spatial model
- Notification list is constrained
- Unclear when you can type in your password (people often want to type before their display has woken up); use a sound for this?
- If you start typing your password before the screen has woken up, you don't get to see the notifications that have come in while the screen has been locked

Discussion

In cases where there are a small number of users - up to around 5 or so - it's possible to rely on profile pictures to identify users.

- Requires that the default profile pictures are good
- Elementary generates unique images based on the user's name - a combination of colours and initials

Can be good to place the user's name and profile picture next to the password field, for association:



Notifications on the lock screen:

- Primarily goal is to provide a summary of what's happened while you've been away
 - They can also help someone decide whether they want to unlock - it's a way of inspecting what's happening without logging in
- Current presentation is quite noisy - a lot of visuals and text for just a small amount of information
- Could be desirable just to show the icon for each app that has sent notifications, along with the number of notifications for that app
- What would the interaction look like for opening a notification (ie. jumping to the source after unlock)? If you start typing your password before you see the icon, it gets awkward.
- Maybe it could be possible to expand each notification to get a bit of information about it, directly on the lock screen? (Unclear what the privacy implications of this would be.)
- However, we do currently show previous notifications after unlock - the last three get replayed. Maybe this means that it isn't so important to interact with the notifications, and that showing detail about each one on the lock screen isn't as important.

For unlock, there's utility in having the user name visible from the very beginning, rather than after you've stepped through - it allows someone to identify who is logged in without having to attempt to unlock.

It's good to advertise the guest session on the lock screen:

- It makes the feature visible
- There's an argument to present the guest session as something that's different from a user account, since it's not actually a user and behaves differently

There's value in keeping login and unlock the same - otherwise it can be disorientating.

- If they are different, make sure they look different. Avoid the uncanny valley.

Tension between:

1. Showing everything on screen and having it immediate, flat, and lacking friction
2. More elegant and focused views, but split up into several steps

The grey texture background can still have a role when selecting a user - outside of the personalised user's space. Maybe it just needs to be richer, less flat.

Next Steps

The following require detailed interaction mockups, high-resolution visual mockups and motion mockups:

- Activities overview. Include:
 - Experiment with scaling the app grid depending on the screen size - ideally needs to keep the pages the same irrespective of the display
 - Interaction for adding to favourites, drag and drop between pages
 - Sketch what the Activities Overview designs would look like for Ubuntu
 - Transitions into and out of the app picker
- Workspaces, fullscreen, tiling
- Login/unlock
 - Transitions for Tobias's user selection concept
- First run experience - overall interaction flow, how the pieces fit together

Mockups for what non-native apps could look like on GNOME, with integrated window controls.