Meet the Bliss Launcher -/e/OS' default launcher

/e/ Developers Blog Nov 3, 2019 · 6 min read

The world of Smartphones is divided between two forces …no not the Empire and the Resistance but iOS and Android. There were some bit characters in this mix. For example Microsoft 's mobile phone OS or the Blackberry OS but over the years these other Operating Systems have almost but disappeared from the Smartphone galaxy.

One of the biggest difference between iOS and Android OS is the level of 'personalization' that is possible on the Android platform. Android allow their users a great deal of control on the look and feel of their phone. Leading the way is the app called the Launcher. The launcher on the Android phones lets you customize your phone screen User Interface UI as per your requirements.

All android phones come with a default launcher — the launcher that comes with the Stock ROM. Google Now is an examples of a default launcher that comes along with the Stock ROM. Having said that there are a multitude of launcher which can be downloaded from various app repositories like FDroid or Aurora Store. Some of the popular launchers are Nova Launcher, Pixel Launcher, Evie Launcher… the list is endless.

Today we w long with the /e/ OS.

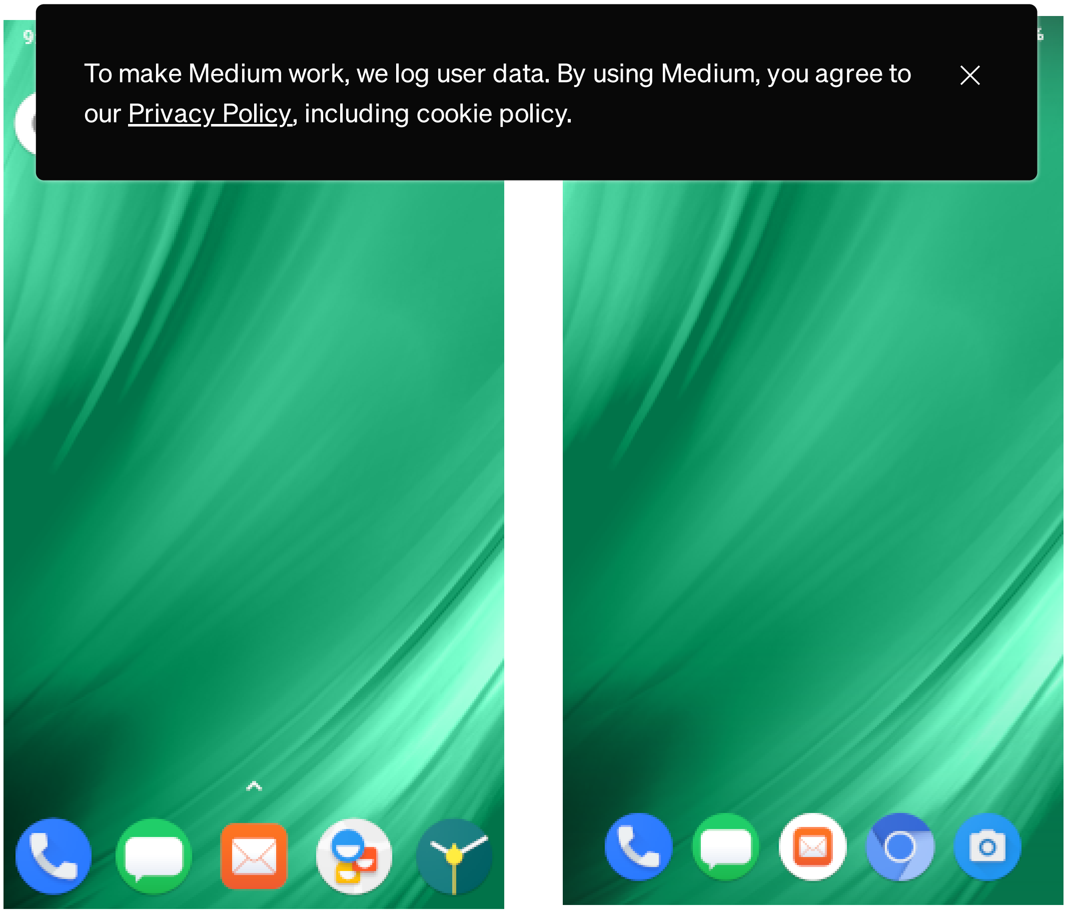
We will also from scratch.
The latest version of the Launcher is version 1.2.3.
Handing the narration over to Amit…

Introducing Bliss Launcher

Hi, I am Amit. I have been working with /e/ as an Android Engineer since its inception. I started to work on the Bliss Launcher in Feb/March of 2018.

Bliss Launcher at its core aims to provide the user an easy to understand and use interface. /e/OS is a privacy-first OS. In an era of unethical data mining, when people are not educated enough about privacy (especially moms and dads), it’s most important to provide them with a UX that’s easy to use and control.

Most existing launchers in android have a very common visual design. This would include having an app drawer, a few screens to arrange your favorite apps and widgets as visible in the screen-shots below :



Nova Launcher and the Pixel Launcher ...two of the popular launchers for Android Smartphones

Most of these launchers are derived from the same codebase ie AOSP Launcher. BlissLauncher was written from scratch. There is a reason behind this decision. The main reason for not using existing Launcher source was that it follows a very different way for loading apps, widgets and shortcuts. Each launcher model is written to suite the OS's UX but in our case, for the /e/OS, it was not going to help. Another issue was if we used the existing AOSP Launcher codebase, we would have to refactor lots of source to fulfill our requirements.

The downsides of writing it from scratch is the time it takes to figure out new launcher features. As android does not have a specific documentation on what to take care of when creating a launcher it becomes difficult to handle specific use cases until a user reports a bug.
Example of such cases — User disables some apps by going into settings, handling apps that are installed on sdcard and handling managed profile.

Tools used in making of BlissLauncher

- Progra
- Editor –
- RxJava and RxAndroid for making things reactive
- Retrofit and OkHttp for loading search suggestions from network
- Room Persistence library for database

and

- Android support library.

There are few other libraries that were used like Calligraphy, EventBus.
We also used the lineage-sdk to implement a weather widget in the BlissLauncher.

Most of the things that are used as a references in BlissLauncher are from AOSP Launcher2 or 3.
Comments in the Launcher3 code are much better than documentation available at https://developer.android.com !!
For those interested in creating a launcher of their own, I would suggest that they have a look at the Launcher3 code.
A word of caution it is not easy to navigate and go through the source!

State of Art

The main screen on the /e/OS with Bliss Launcher

In its current form the Bliss Launcher looks like this on the /e/ OS screen.

The screen has a 4 x 4 grid which holds 16 default apps.

Additional apps are visible on a subsequent screen.

By swiping from the left the screen as visible here shows up. This screen has the following features

- SEARCH — typing the first character will display if the app is already installed on the phone else fetch its reference on the web
- APP SUGGESTIONS — the top 4 most used apps on the phone will show up in this widget
- WEATHER — This widget is installed by default . The weather widget needs to be configured the first time the OS is set up to start showing the local weather. The weather widget cannot be removed as of now.
- EDIT — This button allows the addition of more widgets to the screen

Additional screens can be accessed when swiped from right to left.

By default the /e/OS comes with a set of default system apps.

When you install new apps these get arranged on the additional screens.

A word of caution here from T/e/am : /e/OS is a google-free, user privacy respecting OS. By adding apps like WhatsApp and Facebook the user compromises his / her personal data on these social media platforms. So add new apps at your own risk.

Some of the features of the Bliss Launcher are :

- App Icons are easily accessible on the screen .
- Little or no learning curve for a user who wants to use the /e/OS
- Easily uninstall non — system apps
- One place for having important widgets like weather info or emails.
- Swipe down on any screen for searching apps
- Notification badge for notifying if there is any pending notification of any app.

To delete a non system app on the BlissLauncher

To delete a non system app on the BlissLauncher

- Long press a non-system app
- This will make all the uninstallable apps jiggle
- Apps that jiggle will have a small '-' minus sign on top right of the icon
- Tap the '-' sign on the app you want to uninstall
- Select OK in the pop up which will ask you to confirm the uninstall request
- The app will be uninstalled when you press OK
- A toast appears with the message that the App has been uninstalled.

Updates for the Bliss Launcher

I am planning a few updates to the Launcher in the next versions.
The next major update version will be V2 of the BlissLauncher . It will be

- More stability
- More reactive
- Have lots of new features!!

At present, I am doing a complete refactoring of the BlisLauncher source code. The purpose of this refactoring exercise is :

- Better C
- Implem

You can view the current progress by going to this refactor-mvi branch.

Here is a list of a few features and modifications that are intended to be part of this big upcoming update:

- Pure Kotlin modules
- MVI architecture
- Blurred background behind opened folder and widget panel.
- Horizontal Scrolling of recent apps just like AOSP/Trebuchet Launcher.
- Enable Reordering of widgets.
- Multi-Language support
- Widget preview to avoid confusion in two different widgets of same app

Additional Resources for Reference:

Bliss Launcher Source code

Bliss Launcher Icon Pack

/e/ OS Source Code

Android Open Source Software Android Launcher Bliss Launcher