AKIRA

Building a UX design application from scratch

WORKSHOP

- 20 minutes (ish) presentation
  - Why a new app
  - The differences with other available apps
  - The technology stack
- Demo + Q&A
  - Ideas, suggestions, criticisms, etc.

FOLLOW AKIRA

https://github.com/akiraux/Akira

#akiraux:matrix.org

@akiraux

@akiraux@fosstodon.org

liberapay.com/akiraux

patreon.com/akiraux

WHO AM I

Alessandro, Italian, I love pizza and hate soccer.

- UI/UX Designer
- Full stack developer for 15+ years
- Lead UX Architect at Thunderbird
- FOSS lover and user

Twitter/Mastodon: @alecaddd

Matrix: @aleca:mozilla.org

DISCLAIMER!

All opinions are mine and are not meant to offend or criticize the work of anyone.

GIMP is amazing!

Inkscape is mind blowing!

Krita is a far superior painting app than others!

etc...

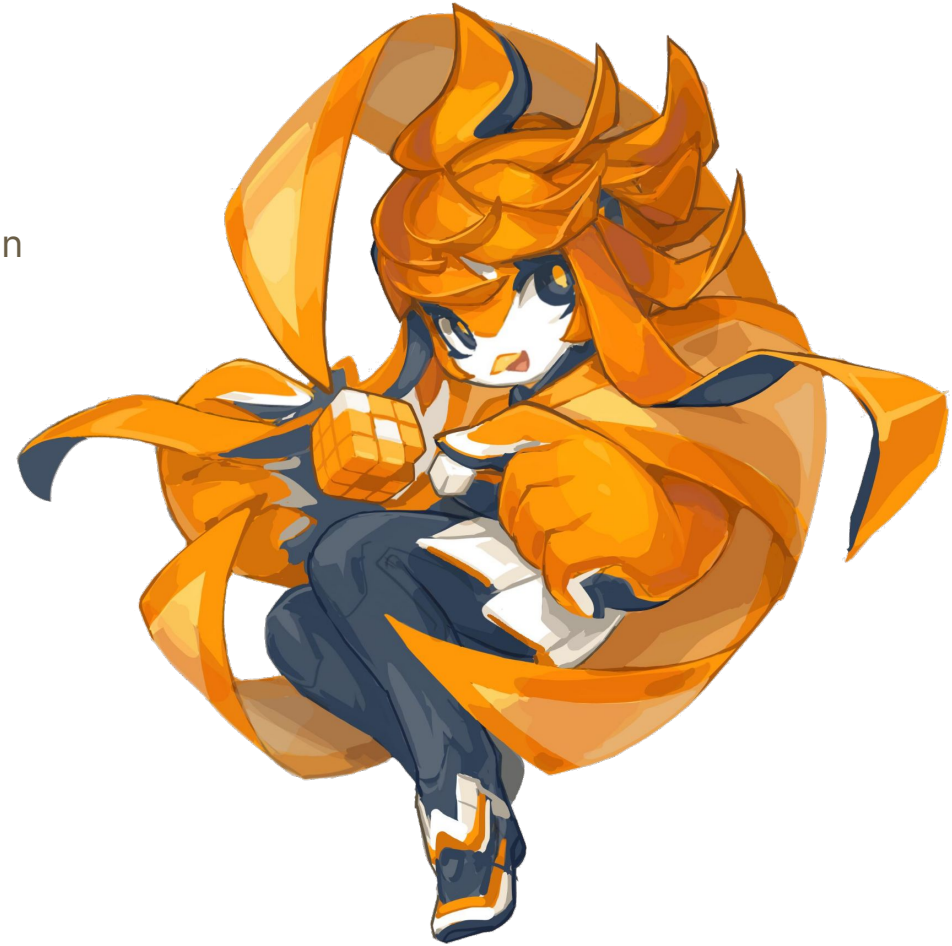
I'm just that guy in a meeting asking "is there a better way to do this?"

WHAT IS IT

2D vector graphical application for UI/UX design

Akira is NOT:

- Photo retouching or manipulation
- Digital painting
- 3D graphics
- Toolkit/Code generator
- Print design...maybe?



FOSS VERSION OF

Akira aims to be the Linux alternative of:

- Sketch
- Figma
- Adobe Xd
- Affinity Designer

IS NOT A REPLACEMENT FOR

- GIMP
- Inkscape
- Krita
- Darktable
- LibreOffice Draw
- Pencil
- Glade
- or any other graphical tool currently available for Linux.

WHY DO WE NEED A NEW APP

Pretty much anything can be done with FOSS applications that can be done with proprietary or closed source applications.

Inkscape

GIMP

Krita

Figma

The problem is not the inability to create something, but rather the speed and efficiency in which you create that “something”.

CHANGES IN THE MARKET

Closed source

Open source

New UX oriented apps

Adobe chickened out

WHAT ARE THE PROBLEMS

Currently available Linux design applications are not a viable solution for a large collaborative professional environment.

- They’re not widely adopted
- They’re not compatible with each other
- The usability and UX paradigms are outdated
- They lack features that are now standard and expected:
  - Reusable (linked) components
  - Built-in version control
  - Multiple Artboards
  - Collaboration and hands off
  - Many more...

FEATURES EXAMPLE

https://uxplanet.org/figma-all-you-need-to-know-156b52b88e54

CONTRIBUTING TO EXISTING APPS

Original attempt to fork Inkscape and slowly contribute upstream, but the complexity of the application was too disconnected from the primary goal I was trying to reach, which was to create a simple and very focused design application with a very narrow objective.

- Massive monolithic code base (my fault)
- Drastic code changes likely rejected
- Drastic UI changes likely rejected
- The goals and new features don’t align with the current scope

THE STACK

The objective was to keep a very simple and easy to install stack, with few dependencies, and a clean and easy to read code base.

- GTK
- Vala
- elementary OS HIG, stylesheet, and icons
- Cairo (via GooCanvas API...for now)

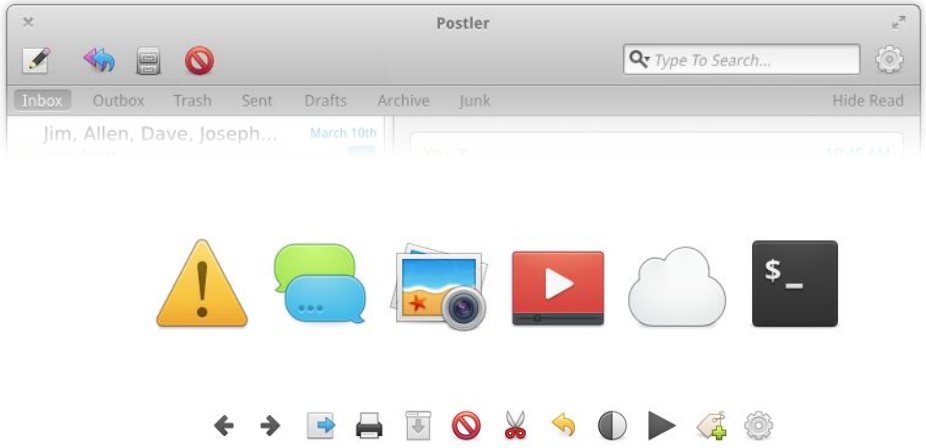
WHY VALA

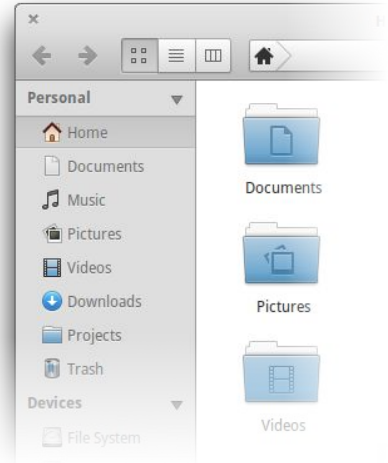
```
public class MyApp : Gtk.Application {
    public MyApp () {
        Object (
            application_id: "com.github.yourusername.yourrepositoryname",
            flags: ApplicationFlags.FLAGS_NONE
        );
    }

    protected override void activate () {
        var main_window = new Gtk.ApplicationWindow (this) {
            default_height = 300,
            default_width = 300,
            title = "Hello World"
        };
        main_window.show_all ();
    }

    public static int main (string[] args) {
        return new MyApp ().run (args);
    }
}
```

WHY ELEMENTARY OS HIG





https://docs.elementary.io/hig/

WHY GOOCANVAS

```c
double x         = 25.6,
       y         = 25.6,
       width         = 204.8,
       height        = 204.8,
       aspect        = 1.0,
       corner_radius = height / 10.0;

double radius = corner_radius / aspect;
double degrees = M_PI / 180.0;

cairo_new_sub_path (cr);
cairo_arc (cr, x + width - radius, y + radius, radius, -90 * degrees, 0 * degrees);
cairo_arc (cr, x + width - radius, y + height - radius, radius, 0 * degrees, 90 * degrees);
cairo_arc (cr, x + radius, y + height - radius, radius, 90 * degrees, 180 * degrees);
cairo_arc (cr, x + radius, y + radius, radius, 180 * degrees, 270 * degrees);
cairo_close_path (cr);

cairo_set_source_rgb (cr, 0.5, 0.5, 1);
cairo_fill_preserve (cr);
cairo_set_source_rgba (cr, 0.5, 0, 0, 0.5);
cairo_set_line_width (cr, 10.0);
cairo_stroke (cr);
```

```c
goo_canvas_rect_model_new (mygroup, 100.0, 100.0, 200.0, 100.0,
                           "stroke-color", "red",
                           "line-width", 5.0,
                           "fill-color", "blue",
                           "border-radius", 10.0,
                           NULL);
```

A SIMPLE AND READABLE SOURCE

```
public class Akira.Lib.Components.BorderRadius : Component {
    public double x { get; set; }
    public double y { get; set; }

    public bool uniform { get; set; }
    public bool autoscale { get; set; }

    public BorderRadius (Items.CanvasItem _item) {
        item = _item;
        x = y = 0.0;

        uniform = true;
        autoscale = false;

        this.notify["x"].connect (update);
        this.notify["y"].connect (update);
    }

    public void update () {
        // We use the X value for both radii in case the radius is set to uniform.
        if (uniform) {
            item.set ("radius-x", x);
            item.set ("radius-y", x);
        }
    }
}
```

IT’S DEMO TIME!

Let’s do this...

```
flatpak remote-add flathub-beta
https://flathub.org/beta-repo/flathub-beta.flatpakrepo

flatpak install akira
```

FOLLOW AKIRA

https://github.com/akiraux/Akira

#akiraux:matrix.org

@akiraux

@akiraux@fosstodon.org

liberapay.com/akiraux

patreon.com/akiraux