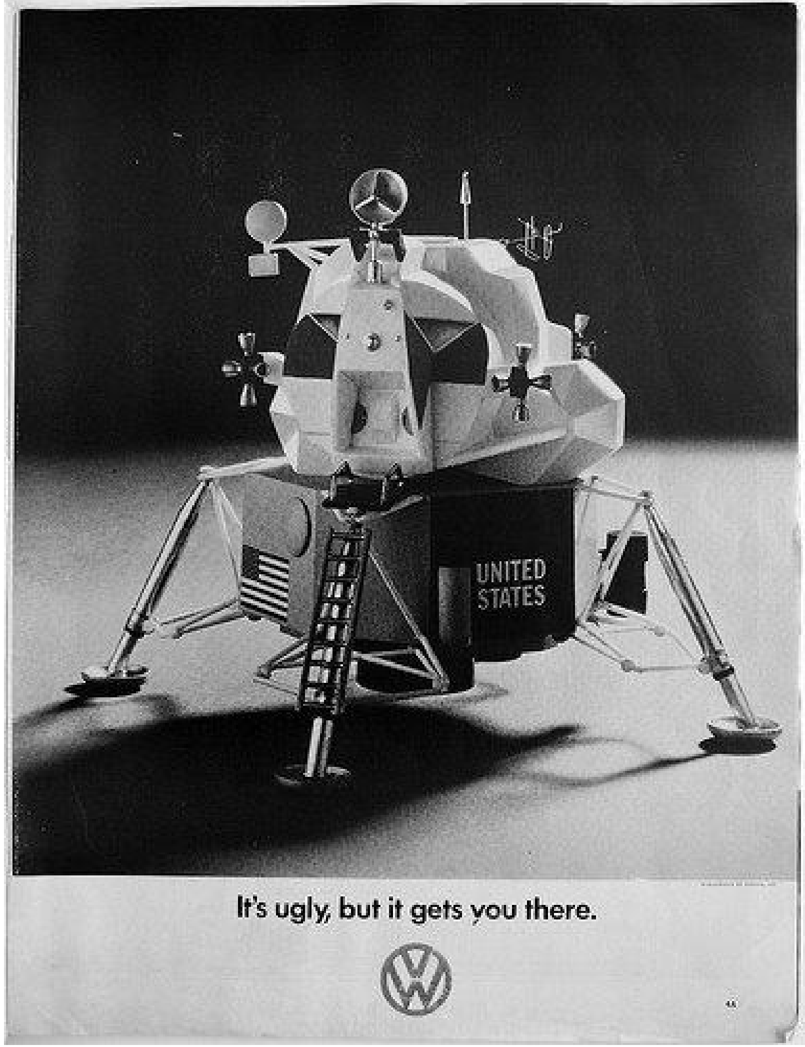
# D ♥ GTK

**Gerald Nunn**

http://www.gexperts.com


UNITED
STATES
It's ugly, but it gets you there.

# Bias Check

- Started with Borland’s Delphi building Desktop Apps in 90s
- Switched to Java and Java Enterprise Edition in 2000
- Some experience with Python for IT automation
- Work for Red Hat as a Middleware Solution Architect
  - No formal involvement with GNOME or GTK
- Biases
  - Prefer strongly and statically typed languages
  - Believe tooling is important
  - Spaces not Tabs :)
- Started using D in late 2015

# GTK+

# What is GTK+?

- A cross-platform GUI Toolkit for Linux, Windows and OSX
- One of the two primary GUI toolkits for Linux
- The toolkit used by the GNOME Desktop Environment
- Primarily written in C with multiple language bindings
  - JavaScript 🐾
  - Python 🐾
  - C++ 🐾
  - D
  - Rust
  - Vala 🐾
  - Pascal
  - Haskall
  - Go (partial)
  - etc...

# GTK+, more than you ever wanted to know

- GTK+ is based around the GLib Object System
    - Aka GObject
- It is reference counted
- GTK+ 3 is current stable version, GTK+ 4 in development
- It uses an object oriented C API
    - Object state is represented by a class specific struct
    - Methods follow convention of {prefix}_{class}_{method}
        - gtk_widget_show
        - gtk_container_add
        - g_settings_set_boolean
    - Relies heavily on inheritance
- GIR files available in XML containing GOBject introspection information

# Writing a GTK+ App

The Usual Suspects…
… from Middle-Earth

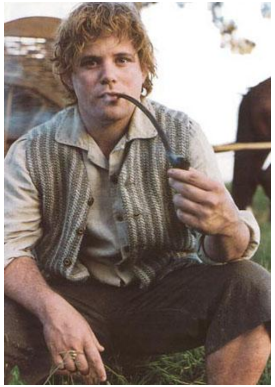
# JavaScript

- Gjs is the JavaScript binding for GNOME
- Based on Mozilla SpiderMonkey
- Used extensively in GNOME Shell

| Pros | Cons |
|---|---|
| Large eco-system of tools and libraries | Dynamically and weakly typed language |
| Easy to learn | Gjs Not widely used outside of GNOME (Shell, Polari, Maps) |
| Memory usage on par with Python | Slow performance |
| | Sub-optimal language design |

# Python

- Very popular choice for writing GTK applications (Lollypop, Terminator, etc..)
- PyGObject provides the GTK language bindings



| **Pros** | **Cons** |
|---|---|
| Large eco-system of tools and libraries | Dynamically typed language |
| Widely used in GTK applications | Reasonably performant but not natively compiled |
| Highly productive and easy to learn | Whitespace as a delimiter |
| First class support for creating GObjects | |

# C

- The lingua franca of GTK
- No bindings required

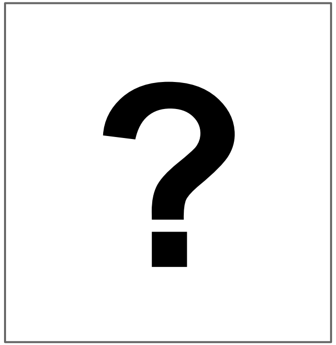
| Pros | Cons |
|---|---|
| Large eco-system of tools and libraries | Not as productive as alternatives |
| Very widely used in GTK applications | High learning curve |
| Native compilation, excellent performance | Inconsistent memory management |
|  | Unsafe |

# C++

- C but better..maybe..
- Bindings provided by gtkmm

| **Pros** | **Cons** |
|---|---|
| Large eco-system of tools and libraries | Not as productive as alternatives |
| Used in a number of GTK applications | Very high learning curve |
| Excellent performance | Very Slow compilation |
| First class support for creating GObjects | Safer than C but not perfect |

# Vala



- Language designed explicitly to support GObject and GTK+ development
- Default language for Elementary OS

| Pros | Cons |
|---|---|
| Easy to learn for people coming from C based languages | Not a general purpose language, single focus |
| Natively compiled, fast performance | Limited tool support |
| Memory management via RC | Cross-compilation to C can make debugging challenging |
| Highly productive with many abstractions available | |

# Writing a GTK App

The Up-and-Comers

# Rust

- New language from Mozilla
- Built on concept of memory safety
- GTK-RS is the binding library

| **Pros** | **Cons** |
|---|---|
| Natively compiled, fast performance | High learning curve, borrow-checker semantics |
| Memory management and safety | Decent but not great tooling |
| Widely hyped language with rapidly growing base of tools | Few GTK applications using language |
| GNOME developers actively engaged | Slow compilation times |
| Excels at libraries and low-level use cases | |

# D

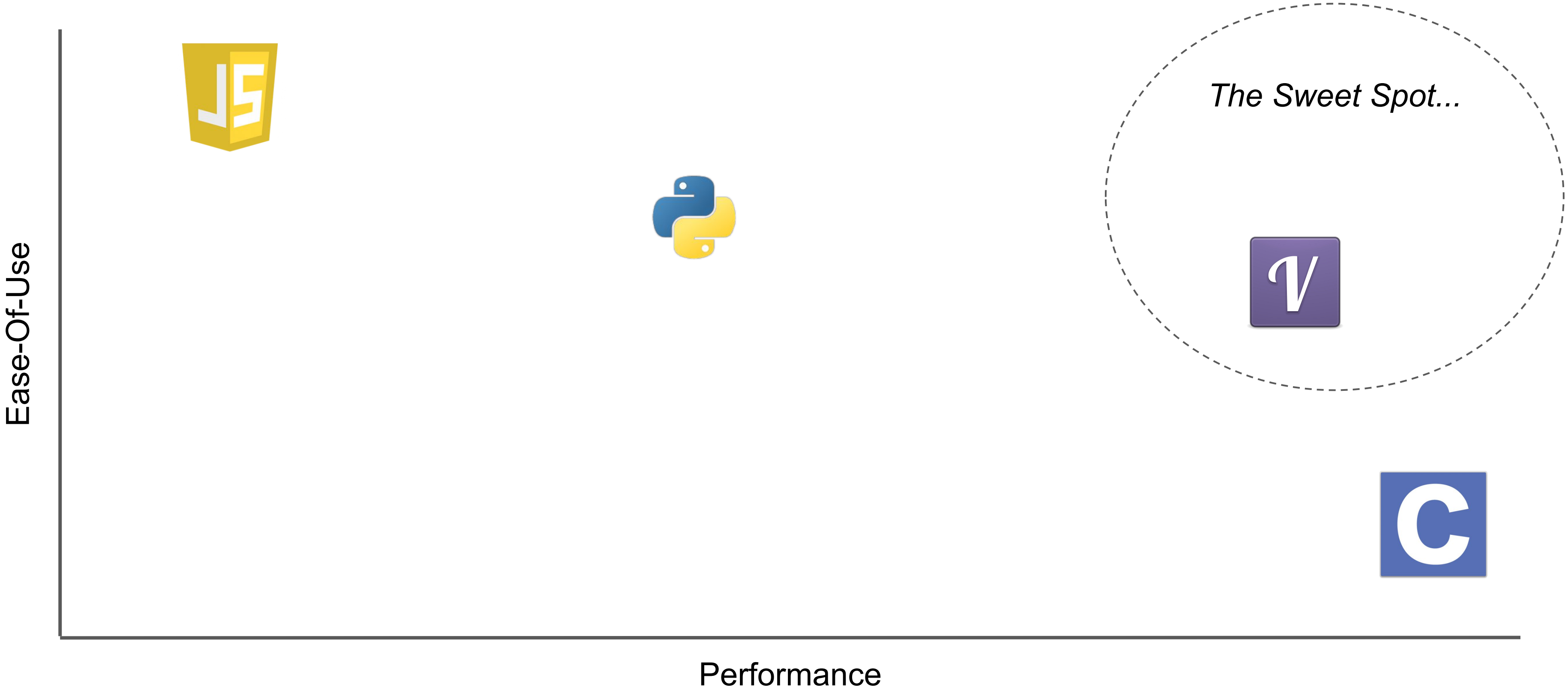
- You already know the story...
- Bindings provided by GtkD

| **Pros** | **Cons** |
|---|---|
| Natively compiled, fast performance | Few GTK applications using language |
| Memory management via GC | Multiple compilers and ABIs |
| Fast compile times | Decent but not great tooling |
| Easy to learn particularly when coming from C family languages | |
| | |

# GTK+ and D

The Argument for D

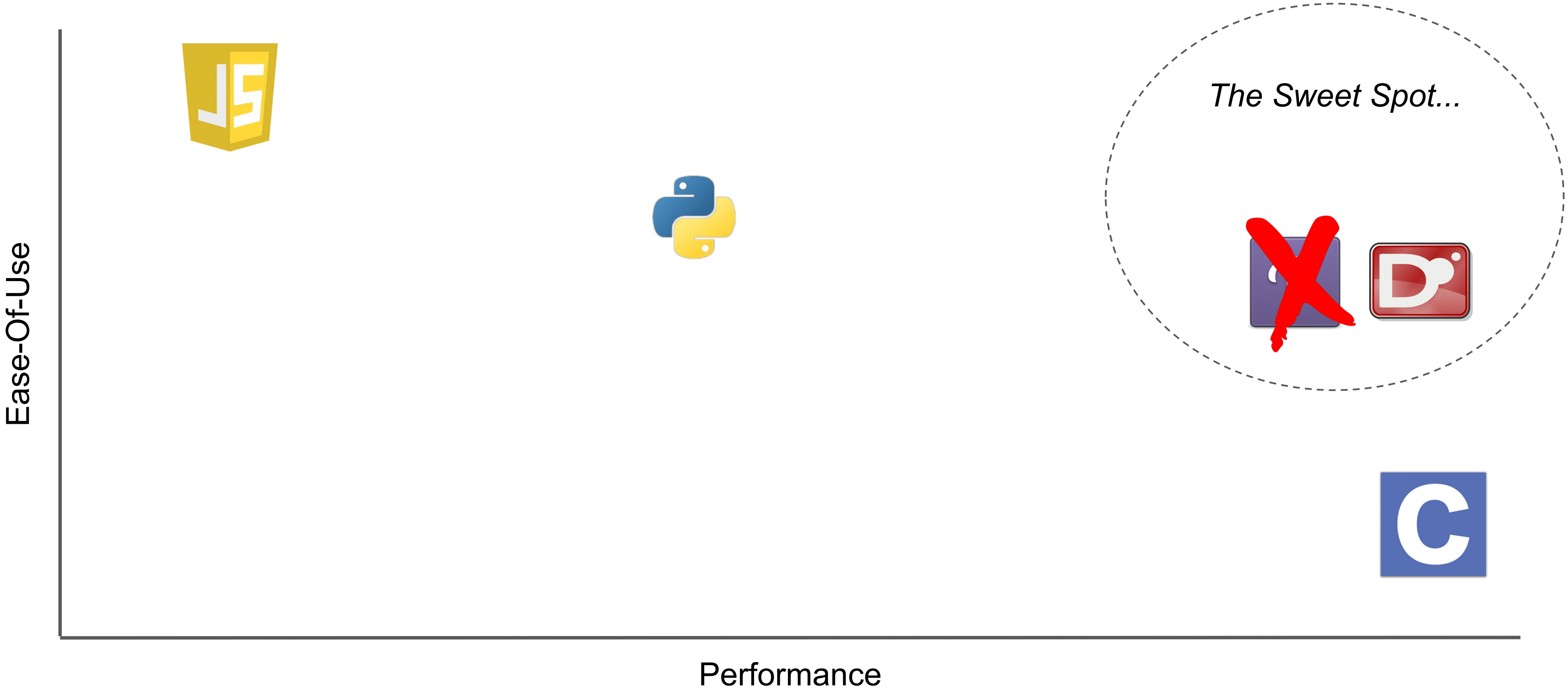
# The Original Vala Use Case



# However, Vala Has No Future (IMHO)

- Few developers
- Little development
- No use outside of GTK+ Development

# My Ideal Future...



*The Sweet Spot...*

Ease-Of-Use

Performance

# Why D for GTK Applications?

- GtkD advantages
    - Full coverage of GTK+ API plus more
    - Generated API based on GObject Introspection
- High performance, fast compilation
- Easy interfacing with C if you need to go low-level
- CTFE potential as C Macros replacement
- GC is productive for GUIs
- Maintenance benefits of static typing
- Easy transition for Vala developers

# Why Not Rust?

✗ High learning curve compared to D

✗ Memory Safety is cool...

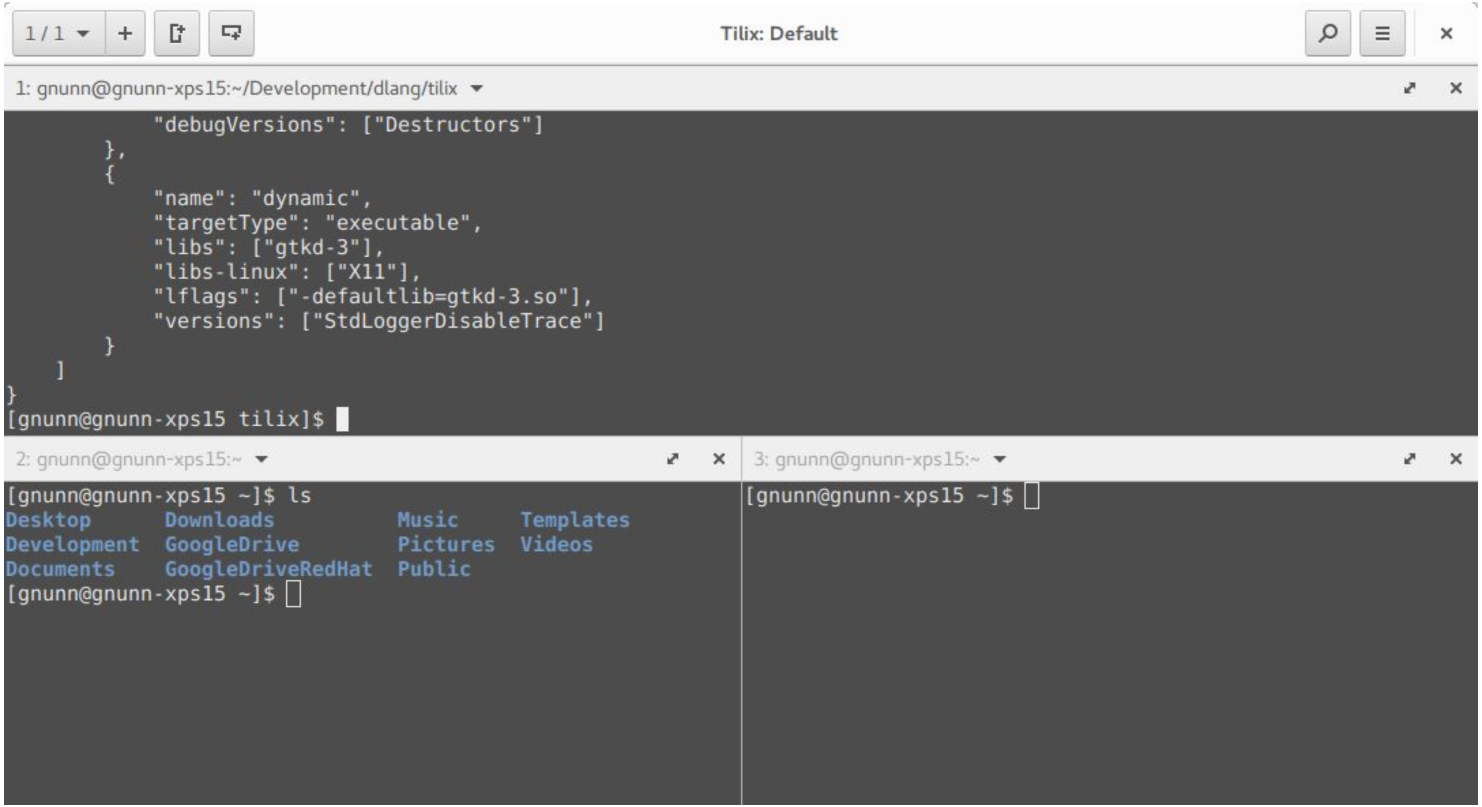
- ○ Is the benefit worth the cost?
- ○ Looks phenomenal when compared to C…
- ○ Less so compared to other modern languages
- ○ Dependencies on unsafe code

✓ Great for Libraries and Low-Level Work

# Success Story

*Tilix Experience Report*

# Tilix



```
            "debugVersions": ["Destructors"]
        },
        {
            "name": "dynamic",
            "targetType": "executable",
            "libs": ["gtkd-3"],
            "libs-linux": ["X11"],
            "lflags": ["-defaultlib=gtkd-3.so"],
            "versions": ["StdLoggerDisableTrace"]
        }
    ]
}
[gnunn@gnunn-xps15 tilix]$
```

```
[gnunn@gnunn-xps15 ~]$ ls
Desktop      Downloads          Music     Templates
Development  GoogleDrive        Pictures  Videos
Documents    GoogleDriveRedHat  Public
[gnunn@gnunn-xps15 ~]$
```

```
[gnunn@gnunn-xps15 ~]$
```

# Tilix - Fast Facts

- Started in November 2015
- Written in D using the GtkD bindings
- Supports GTK+ 3.14 to 3.24
- Complies with GNOME HIG
- Is my second GTK+ program
  - Visual Grep was first, learning experience

# Tilix Successes

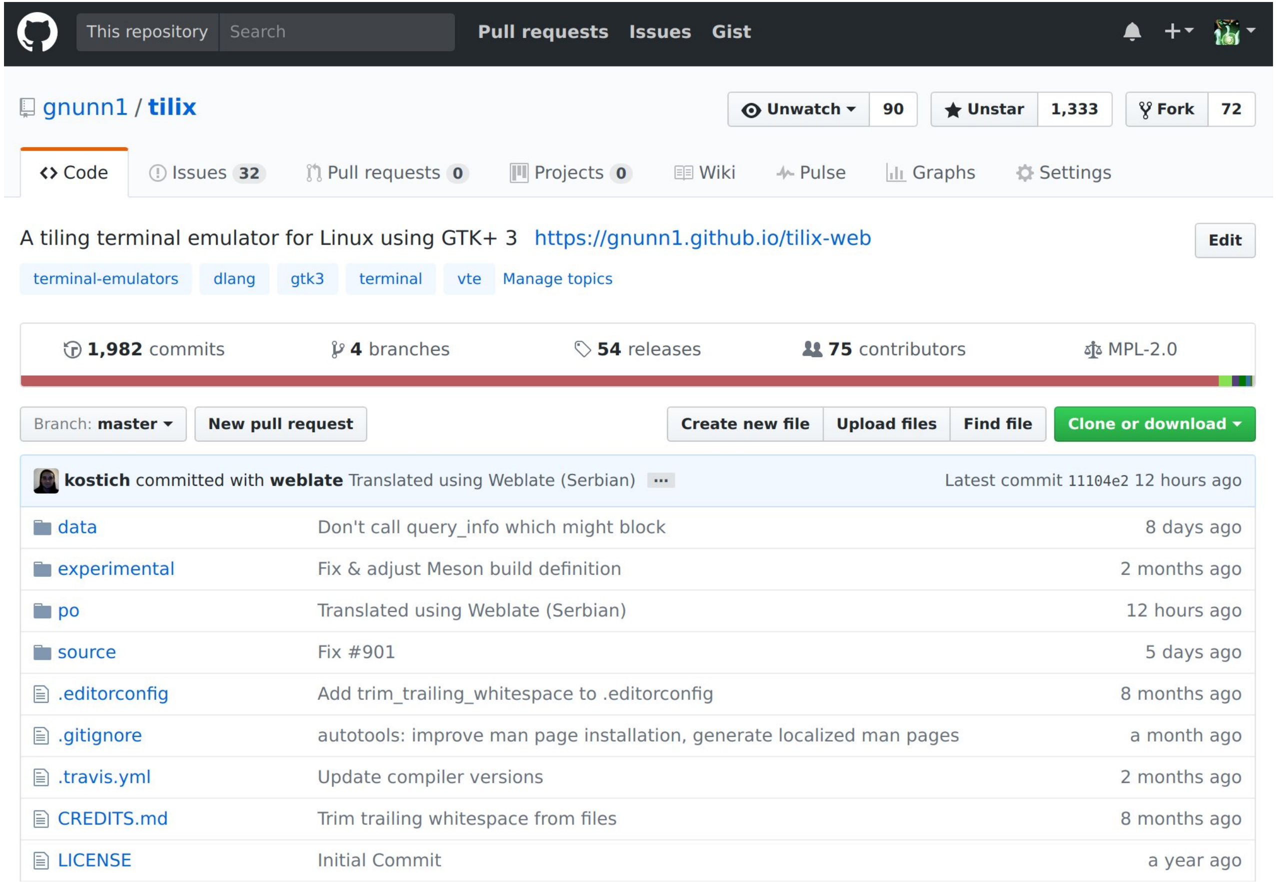
- Packages available for every major Linux Distro
- Official packages for Debian
- Official package for Fedora will be available in Fedora 26
- Default Terminal for Ubuntu Budgie flavor
- Localized in ~25 languages
- Github:
    - 74 Contributors
    - 54 Releases
    - 185 PRs
    - ~ 700 issues, ~670 closed
    - ~1300 stars, 72 forks
    - 2nd most starred D Application on GitHub

# Lessons Learned - The Good Stuff 😊

- GtkD bindings are mature
- Linux packaging is hard, leave it to the experts
- Distro specific bugs are unfortunately common…
- … so be restrictive on what you support
- GNU gettext works great using Vala as input language

# Lessons Learned - Regrets 😔

- Should have made more effort up front to leverage unique D features
- Check trademarks before choosing project name
- Take the high road when interacting with users



A tiling terminal emulator for Linux using GTK+ 3 https://gnunn1.github.io/tilix-web

kostich committed with weblate Translated using Weblate (Serbian) … Latest commit 11104e2 12 hours ago

| | | |
|---|---|---|
| data | Don't call query_info which might block | 8 days ago |
| experimental | Fix & adjust Meson build definition | 2 months ago |
| po | Translated using Weblate (Serbian) | 12 hours ago |
| source | Fix #901 | 5 days ago |
| .editorconfig | Add trim_trailing_whitespace to .editorconfig | 8 months ago |
| .gitignore | autotools: improve man page installation, generate localized man pages | a month ago |
| .travis.yml | Update compiler versions | 2 months ago |
| CREDITS.md | Trim trailing whitespace from files | 8 months ago |
| LICENSE | Initial Commit | a year ago |

https://github.com/gnunn1/tilix

# A Call To Action

# What You Can Do

- Write GTK+ Applications
  - More Apps = More Recognition
  - D & GTK Template App from GNOME Developer Carlos Soriano
    - https://gitlab.com/csoriano/GtkDApp
- Contribute to GtkD Development
  - Create higher level API abstractions
- Contribute To D Development
- Improve Linux packaging for D Applications
  - https://gist.github.com/ximion/fe6264481319dd94c8308b1ea4e8207a

# Questions?

# Desktop Applications

*Yes Matilda, Desktop Applications Are Still Relevant...*

# Desktop Applications - Still Relevant?

“There are three consumer devices that are leading tablets by a large margin: TVs, smartphones, and computers. It seems unlikely that the tablet will ever displace these devices.”

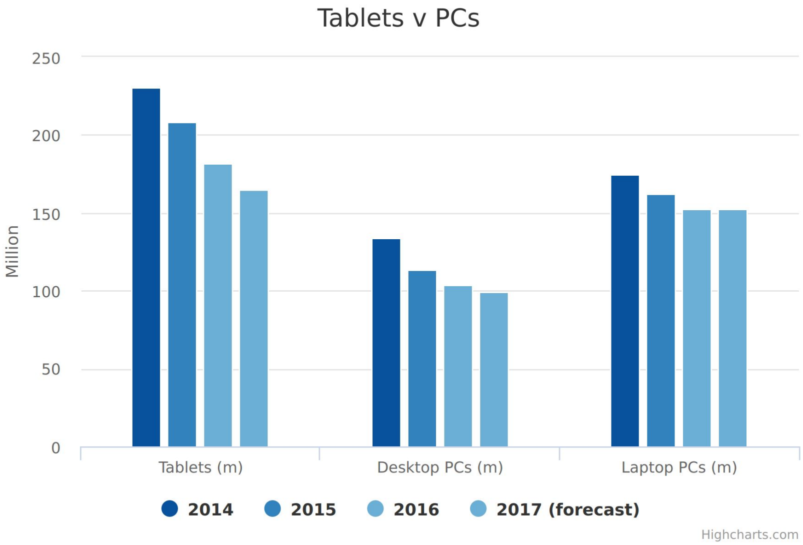
---

**“What about tablets? People say they’re just as good a PC for making content.**

Sure they are, once you connect a keyboard and a mouse to them. Then guess what? That’s a PC too!”

---

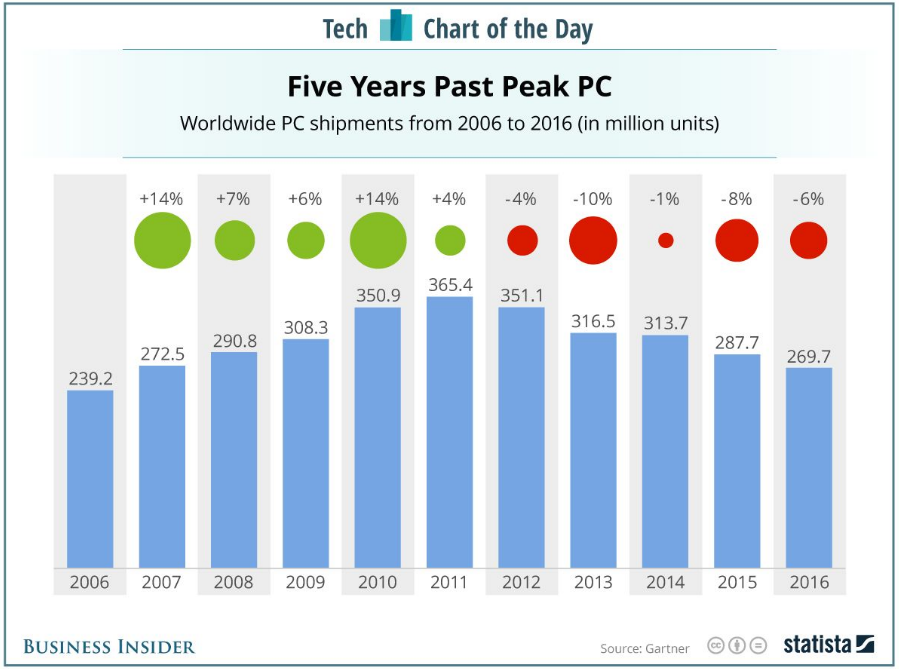
“Well, it boils down to the reality that almost every piece of technology you use and much of the media content you consume, is designed and built on PCs.”



*http://www.telegraph.co.uk/technology/2017/01/11/tablet-sales-fall-third-successive-year-pc-market-stabilises/amp/*

# Desktop Applications

Down but not out....



Source: Gartner

# Desktops Are Evolving

**Convergence?**

**It's a desktop! It's a tablet!**

**The PC is evolving, not disappearing, analysts say**

**Content Creators Need More Power**

# Agenda

- Bias Check
- Desktop Applications, Not Dead Yet...
- What is GTK?
- Writing a GTK App
  - The Usual Suspects
  - The Up-and-Comers
- D and GTK: The Perfect Match
- Experience Report: Tilix

# Tilix - Github Statistics

| | |
|---|---|
| Commits | **Almost 2000** |
| Contributors | **74** |
| Releases | **54** |
| Issues | **~ 700 (~30 open)** |
| Starred | **1316** |

*Second most starred D language project on Github:*
*https://github.com/search?l=D&q=stars%3A%3E1&s=stars&type=Repositories*

# Desktop Market Evolving

Software for Consumers

Software for Professionals