GTK4 Exploration - SugarLabs GSoC Proposal

Krish Pandya

Abstract

A comprehensive plan to port Sugar and its core activities from GTK3 to GTK4, ensuring the platform's technical sustainability and future growth.

Table of Contents

GTK4 Exploration - Sugar GSoC Proposal

Personal Information

Full Name:

Krish Pandya

Contact Information

- Email: krish.pandya@students.iiit.ac.in
- GitHub: MostlyKIGuess
- Portfolio: https://mostlykiguess.github.io/

Languages:

Proficient in English, Hindi, and Gujarati (more proficient in English and Gujarati)

Location and Timezone:

Hyderabad, India (UTC +5:30)

Previous Contributions and Experience

Open Source Background

I have been actively contributing to various repositories , with a particular focus on Sugar ecosystem. My open source journey shows my will to improve educational technology .

Sugar Labs Contributions

I have made numerous contributions to Sugar Labs, demonstrating my understanding of the codebase and my ability to work effectively within the Sugar ecosystem:

My Activity Graph on SugarLabs Organization:

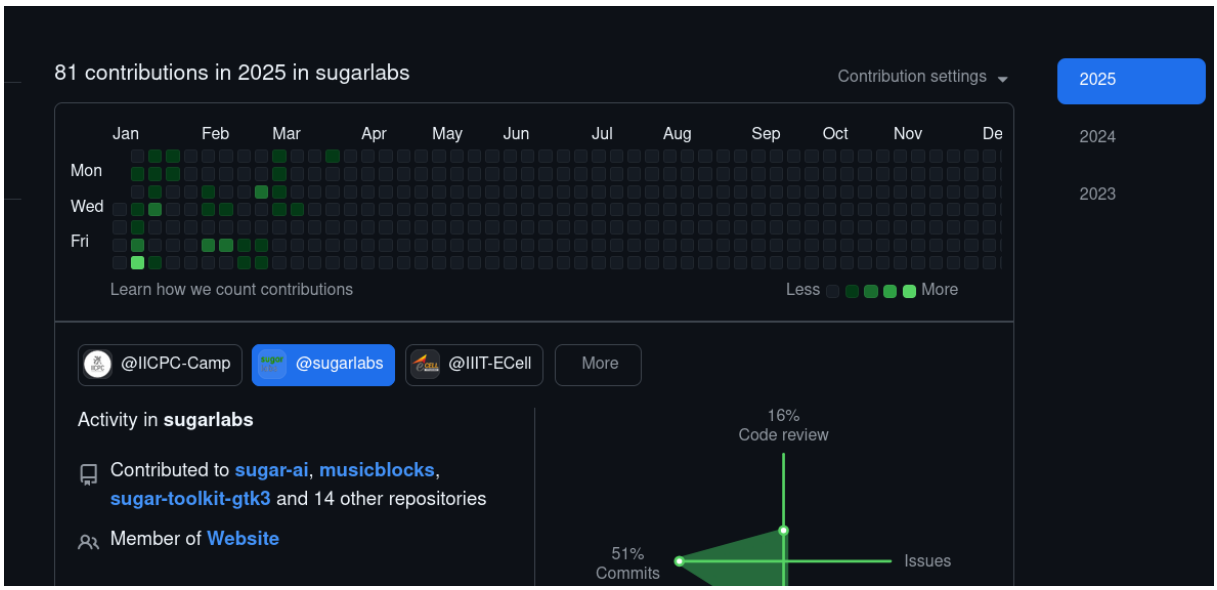

Figure 1: 2025 Activity

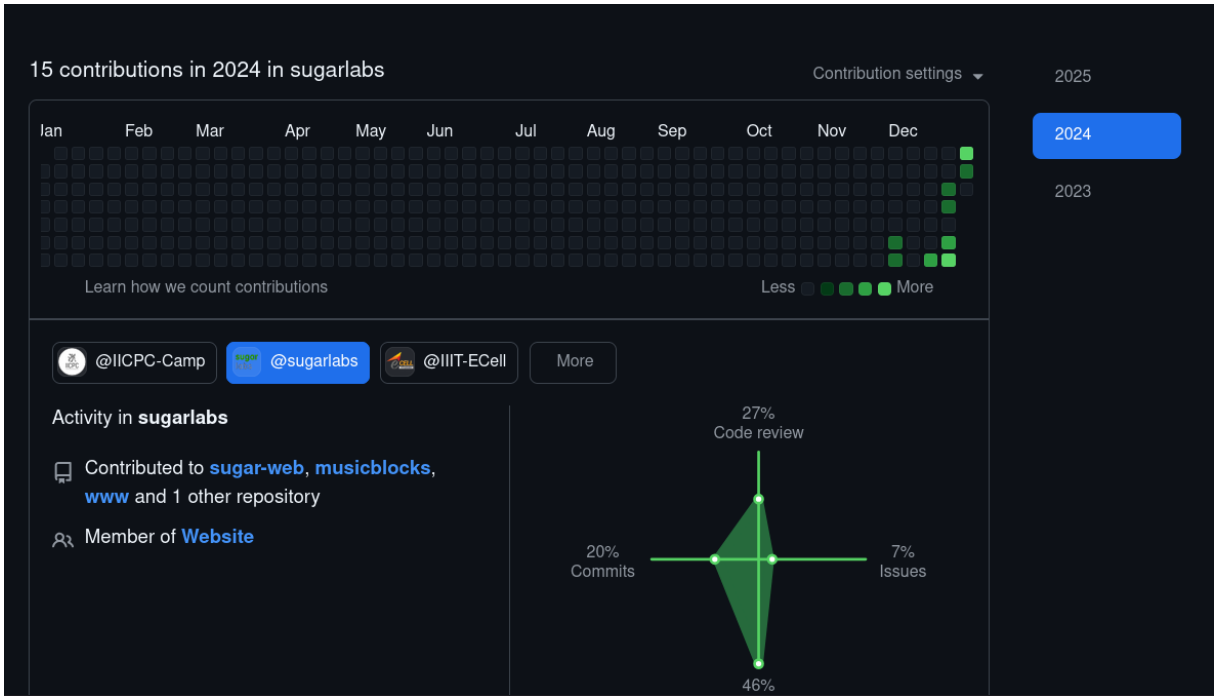

Figure 2: 2024 Activity

1. **GTK4 Port (In Progress)**: I've already begun the critical work of porting Sugar's toolkit from GTK3 to GTK4, with 54 commits (although unstructured) addressing core functionality and components. This includes refactoring the Activity class, Palette system, and various widgets to

use GTK4 APIs. Link Here

2. **Cross-Project Improvements**:

   - Restructured the Sugar-AI codebase to better align with Sugar's architectural needs. Link here (Merged by hand, had to resolve the conflicts, hence it's closed) .
   - Enhanced Activity functionality with improved presence handling in sugar-web (updated the entire library along with proper testing). Links that made on this update:
     - https://github.com/sugarlabs/sugar-web/pull/138.
     - https://github.com/sugarlabs/sugar-web/pull/139.
   - Fixed deprecated ConfigParser methods across multiple projects (sugar-toolkit-gtk3, Frotz, IRC activity).
     - Multiple PRs were in this fix, I will attach a screenshot.

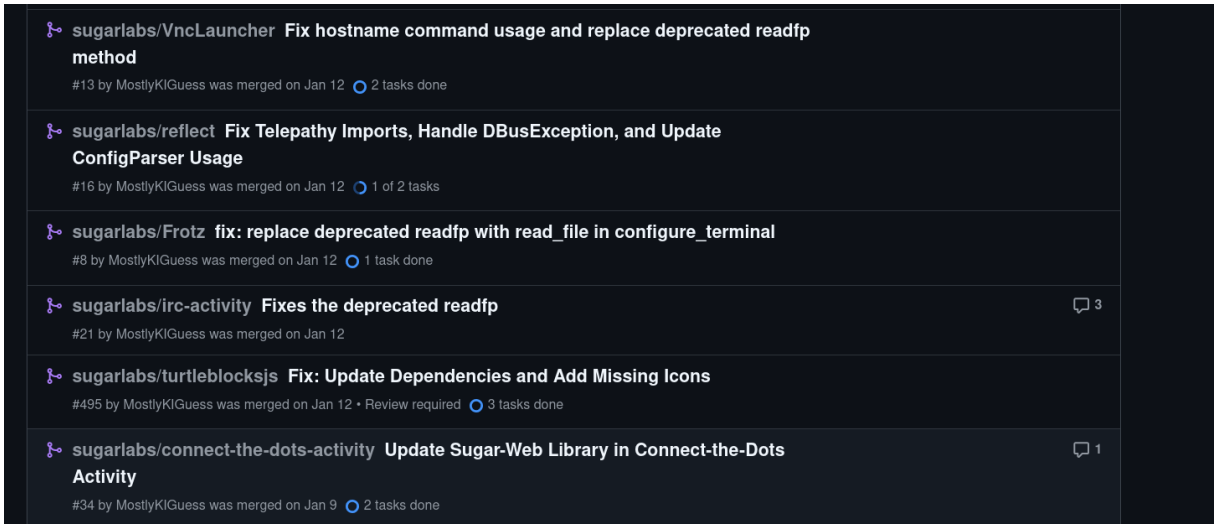

Figure 3: Activity PRs

   -

   - Updated dependencies for better stability in MusicBlocks, TurtleBlocksJS. (Attached in above screenshot)
   - Added extra project ideas and deployed https://sugar-docs-ai.streamlit.app/ (Matrix AI Bot, with Image understanding and RAQ query on Sugar) https://github.com/MostlyKIGuess/Sugar-Docs-AI.
   - Created https://sugardocs.vercel.app/ (Web Preview of Docs). https://github.com/MostlyKIGuess/sugardocs
   - Also created a web-version of Debian OS, which can be used in Sugar to Deploy it's OS for others to experience sugar activities easily. https://github.com/MostlyKIGuess/Sugar-VM-Web

These contributions demonstrate my deep understanding of Sugar's architecture, my ability to work across different components of the ecosystem, and my will to maintain and improve the project's codebase.

Project Overview

Project Scope

I am proposing to complete the port of Sugar and its core activities from GTK3 to GTK4. This involves:

1. Completing the migration of the sugar-toolkit-gtk3 components to GTK4
2. Porting Sugar's shell and UI components to utilize the new toolkit
3. Migrating core activities (the Fructose set) to ensure compatibility with the updated framework
4. Creating comprehensive documentation to guide future activity migrations
5. Developing a "Toolkit Test" activity that demonstrates and tests all migrated components

Project Impact This project is critical for Sugar Labs' future for several key reasons:

1. **Technical Sustainability**: GTK3 is approaching end-of-life, and without migration, Sugar would be dependent on deprecated technology, making maintenance more difficult. And we lack manpower.
2. **Improved Performance**: GTK4 offers performance improvements, particularly for rendering and animations, which will make Sugar run more smoothly on the resource-constrained hardware which are used in educational settings.
3. **Modern Features**: GTK4 provides access to new UI capabilities, including better touch support, improved accessibility, and more consistent theming, enhancing Sugar's usability across diverse environments. We can leverage these features and implement them.
4. **Developer Experience**: A modernized codebase will be more alluring to new contributors, helping to sustain and grow the developer community around Sugar which we need.
5. **Educational Impact**: By ensuring Sugar's continued viability this project directly supports Sugar Labs' educational mission allowing the platform to remain available to learners worldwide.

Technologies Involved

The project will primarily involve:

- **C Programming**: For core toolkit components and low-level integration
- **Python**: For higher-level components and activity porting
- **GTK4**: The primary UI toolkit being migrated to
- **GObject Introspection**: For creating Python bindings to the C-based toolkit
- **Sugar Architecture**: Working extensively with Sugar's component model, activity system, and collaboration framework

Implementation Details

Technical Approach

1. Core Toolkit Migration Strategy

GTK3 → GTK4 Breaking Changes

- **GDK Screen/Display**:
  Replace deprecated `GdkScreen` with `GdkDisplay` (GTK4 Porting Guide: GDK Changes)
  Example: Update screen geometry checks in `sugar3/graphics/device.py`.
- **Event Handling**:
  Migrate from `Gtk.main_do_event()` to `Gtk.EventController` (Event Controllers).
  Critical for Sugar's touchscreen/tablet support in activities like **Paint**.
- **Layout Management**:
  Replace `GtkContainer` APIs with `GtkWidget`'s `set_parent()`/`unparent()` (Widget Hierarchy Changes).
- **CSS Theming**:
  Migrate from `Gtk.StyleContext` to `Gtk.CssProvider` (CSS Changes).

Sugar-Specific Components

- **ActivityWindow**:
  Rewrite using `Gtk.ApplicationWindow` instead of `Gtk.Window` to leverage GTK4's application lifecycle management.
- **Toolbox/Palette**:
  Replace `Gtk.Popover` with `Gtk.PopoverMenu` for better touch compatibility (Popover Migration).
- **Canvas**:
  Migrate from `Gtk.DrawingArea` to `Gtk.Snapshot`-based rendering (Snapshot API).

2. Shell Architecture Changes

Critical GTK4 Adaptations

Component	GTK3 Approach	GTK4 Approach	Docs Link
Window Management	`Gtk.Window` + custom logic	`Gtk.Application` + `Gtk.Native`	Application Class
Drag-and-Drop	`Gtk.drag_*()` APIs	`Gtk.DropTarget`	Drag and Drop
Journal Integration	`Gtk.TreeView`	`Gtk.ColumnView`	List Widgets

Sugar Shell Components

- **Home View**: Port from `Gtk.FlowBox` to `Gtk.GridView` for activity icons.
- **Collaboration**: Update presence system to use `Gio.SocketService` instead of deprecated D-Bus APIs.
- **Frame Management**: Replace `Gtk.Overlay` with `Gtk.Widget`'s new layout system.

Testing Strategy

1. Unit Tests

- **Python**: Use `pytest` for toolkit components (e.g., test `sugar3.graphics.style` theming).
- **C**: Leverage GLib's test framework for low-level widgets.
- **Visual Regression**: Implement screenshot-based testing using GtkTestUtils.

2. Integration Tests

- **Activity Lifecycle**: Verify startup/shutdown sequence with `Gtk.Application` signals.
- **Journal**: Test file I/O using temporary directories (`Gio.File.create_temp()`).

3. CI Pipeline

A[PR Opened] –> B{Run Tests} B –>|Pass| C[Build Debian Package] B –>|Fail| D[Notify via Matrix] C –> E[Test on Ubuntu 24.04 VM]

Documentation Plan

1. Migration Guides

- GTK4 Porting Checklist for Sugar Developers
- **Common Patterns**:

```
# GTK3
box.pack_start(child, expand=True, fill=True, padding=0)

# GTK4
box.append(child)
child.set_hexpand(True)
child.set_halign(Gtk.Align.FILL)
```

2. API Reference

- Auto-generated via gi-docgen for C components.
- Python examples in `sugar-toolkit-gtk4/docs/`.

Risk Management

Risk	Mitigation Strategy
GTK4 API instability	Pin to stable GTK4 releases (≥ 4.14)
Python binding gaps	Contribute upstream to PyGObject as needed
Performance regressions	Profile with Sysprof early in migration

Deliverables

1. **Code**:
    - sugar-toolkit-gtk4 merged
2. **Documentation**:
    - Migration Guide (hosted on GitHub Pages)

    - Toolkit Test Activity (example)

3. **Community**:
    - Communicate via Sugar Matrix

Open Questions

1. Should we maintain **GTK3 compatibility** during transition?
   Proposal: Use `#ifdef` GTK4 conditionals temporarily.
2. How to handle **Wayland support**?
   Next Step: Test under mutter compositor.
3. **Debian Packaging** strategy for GTK4?
   Action: Coordinate with Debian Sugar Team.

Project Timeline

Pre-GSoC Period (Current - May 2025)

- Continue preliminary work on sugar-toolkit-gtk3 components
- Refine migration strategy based on mentors' feedback
- Build and document development environment for efficient testing

Week 1-2: Foundation and Hello World (May 2025)

- Complete migration of essential toolkit components needed for basic activity support
- Focus on Activity class, graphics classes, and core UI elements
- Port Hello World activity to use new GTK4-based toolkit
- Document initial migration patterns and challenges

Week 3-4: Toolkit Components Expansion

- Migrate additional toolkit components including:
    - Complete palette system migration
    - Finish toolbox and toolbar components
    - Implement journal integration with GTK4

- Update test cases for migrated components
- Weekly progress reports and code reviews with mentors

Week 5: First Evaluation Preparation

- Expand Hello World into a “Toolkit Test” activity that exercises all migrated components
- Create comprehensive documentation of migration strategy
- Prepare code for first evaluation
- Address any issues identified in review

First Evaluation Milestone: Functional Hello World activity running on GTK4-based toolkit

Week 6-8: Sugar Shell Migration

- Begin porting Sugar shell components to GTK4
- Focus on view system, activity launching, and window management
- Implement necessary changes to activity lifecycle handling
- Weekly progress reports and code reviews

Week 9-11: Shell Completion and Testing

- Complete Sugar shell migration
- Implement journal integration with GTK4 shell
- Develop and run comprehensive tests
- Begin documentation for activity developers

Week 12: Second Evaluation Preparation

- Address any issues identified in testing
- Refine documentation
- Prepare code for second evaluation
- Plan Fructose activity migration priorities

Second Evaluation Milestone: Functional Sugar environment running on GTK4

Week 13-15: Fructose Activities Migration (If Time Allows / Post-GSoC)

- Port high-priority Fructose activities:
  - Terminal Activity
  - Browse Activity
  - Write Activity
- Document activity-specific migration patterns
- Weekly progress reviews with mentors

Week 15-16: Final Activities and Documentation (If Time Allows / Post-GSoC)

- Complete remaining Fructose activities as time permits
- Finalize all documentation
- Create final migration guide for activity developers
- Prepare final submission

Final Week: Project Wrap-up

- Final testing across all components
- Address any remaining issues
- Complete final report
- Prepare presentation of results

Time Commitment and Availability

Weekly Hours

I will be dedicating approximately 25-45 hours per week to the project. I have no planned absences during the GSoC period, with the following considerations:

- **Early May**: I will have my end-semester exams and some research work will begin, which may limit my availability during the initial weeks
- **July onwards**: I will be on vacation and can commit more than 45 hours per week to accelerate project progress

Progress Reporting Strategy

I will maintain transparency and regular communication throughout the project:

1. **Weekly Progress Reports**: I'll provide detailed updates on completed tasks, challenges faced, and plans for the coming week.
2. **Code Reviews**: I'll submit regular pull requests for review, ensuring code quality and alignment with project goals.
3. **Documentation Updates**: As migration patterns emerge, I'll document them to help both current and future developers.
4. **Community Updates**: I'll participate in community meetings to share progress and gather feedback, as I do regularly.

Post-GSoC Commitment

Ongoing Contributions

My commitment to Sugar Labs extends beyond the GSoC period. After the program ends, I plan to:

1. **Continue Activity Migration**: Work on porting additional activities beyond the project requirements if they weren't met during the GSoC period.
2. **Mentorship**: Help guide new contributors through the now-documented GTK4 migration process. I also plan to assist with AI projects, leveraging my expertise to support newcomers to SugarLabs.
3. **Performance Optimization**: Identify and implement optimizations made possible by GTK4's improved rendering pipeline.
4. **AI Integration**: Continue my work on Sugar-AI, leveraging my deep understanding of the newly migrated codebase to better integrate AI capabilities into the Sugar environment.
5. **Creative Development**: I will continue helping people start their AI projects and expand on creative ideas that newcomers bring to Sugar.

I see this project not as a one-time contribution but as the beginning of a long-term commitment to Sugar Labs' mission of providing quality educational software to learners worldwide.

Why Me?

I believe I am the right candidate for this project because I have a proven record of contributing to Sugar Labs, a deep understanding of its architecture, and a passion for educational technology. My

previous work on GTK4 migration shows my technical capability, while my cross-project contributions demonstrate my adaptability and problem-solving skills.

Additionally, I have strong C foundations reflected on my Data Structures and Algorithms A Grade (the highest grade you can get), my work ethic and ability to work for other organization, I will be going to ICASSP for my work in virtuallabs in which I made visualizers, and I work as a System Administrator which shows my understanding on Linux and management.

Alongside my undergraduate studies, I also do research at https://robotics.iiit.ac.in/ , topics include Relocalization, Computer Vision, and 3D Understanding for Autonomous Systems and Self-Driving Cars. While not directly related to Sugar Labs, these experiences demonstrate my ability to learn, execute complex projects, and deliver results.