sugarlabs

GSoC Proposal : Math Games for Sugar Labs.

Basic Details :

FULL NAME	:	ALKAIF GAJDHAR
AGE	:	20
EMAIL	:	alkaifgajdhar@gmail.com
GITHUB	:	kaif_builds
TIMEZONE	:	Indian Standard Time (UTC +5:30)
CONTACT No.	:	8827111136
DEGREE	:	Undergraduate in Computer Science

SKILLS:

Proficient in Python, JavaScript, and game development, with a strong understanding of both frontend and backendlogic.
Experienced in AI prompting, UI/UX design, and graphic design.

INTERESTS:

Passionate about AI-driven learning, game development, and UI/UX design. Enthusiastic about creating interactive educational tools
.

1. Why Sugar Labs?

The selection of Sugar Labs as a potential GSoC project reflects a strategic decision to contribute to an established open-source platform dedicated to interactive and engaging learning. With its activity-based approach and intuitive interface, Sugar Labs has transformed the way users interact with educational technology, fostering creativity and exploration through hands-on learning. Participating in GSoC 2025 with Sugar Labs offers an exceptional opportunity to contribute to a well-established platform that has made significant strides in open- source education and digital learning tools.

This project will allow me to collaborate with experienced developers, refine my technical and design skills, and create AI-powered, interactive math games that enhance the learning experience. By leveraging Python, GTK, and modern UI/UX principles, I aim to develop engaging activities that align with Sugar Labs’ mission—making learning more accessible, enjoyable, and impactful.

2. About Project :

Math Games for Sugar Labs: Expanding Interactive Learning with AI and Engaging Puzzles.

ABSTRACT:

This Google Summer of Code project aims to expand Sugar Labs' learning platform by developing eight interactive math games that promote logical reasoning, problem-solving, and AI-driven learning. These activities will provide an engaging, hands-on approach to mathematics while aligning with Sugar’s exploratory and collaborative learning philosophy.

By integrating AI-based challenges and interactive puzzles, this project will enhance user engagement and concept retention. Through community collaboration and iterative development, it will introduce high-quality, reusable activities, making math more interactive and enjoyable within the Sugar ecosystem..

TECHNICAL DETAILS:

The crux of this Google Summer of Code initiative is the development and integration of eight interactive math games into the Sugar Labs platform. These activities will incorporate logical puzzles, problem-solving tasks, and AI-driven challenges, enhancing the platform’s interactive and exploratory learning environment.

To ensure seamless integration, the project will utilize Python, GTK, TypeScript, and JavaScript, maintaining compatibility with Sugar’s activity-based learning framework. Games such as Number Detective and Sorting Hat AI will leverage machine learning techniques like rule-based logic and k-Nearest Neighbors (k-NN) classification to introduce AI-driven gameplay elements.

The successful execution of this project will require a strong understanding of game logic, UI/UX design principles, and AI integration. Close collaboration with Sugar Labs mentors and the open-source community will be essential to refine the game mechanics, optimize performance, and align with Sugar’s educational goals. The development process will follow an iterative approach, incorporating feedback-driven improvements and rigorous testing to ensure a smooth and engaging user experience.

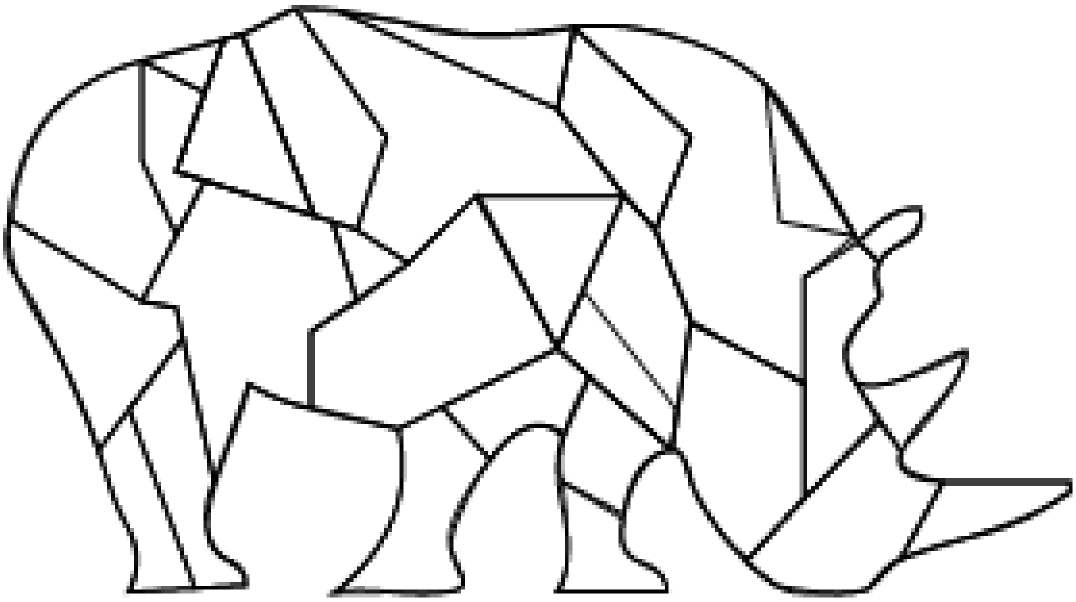
DELIVERABLES:

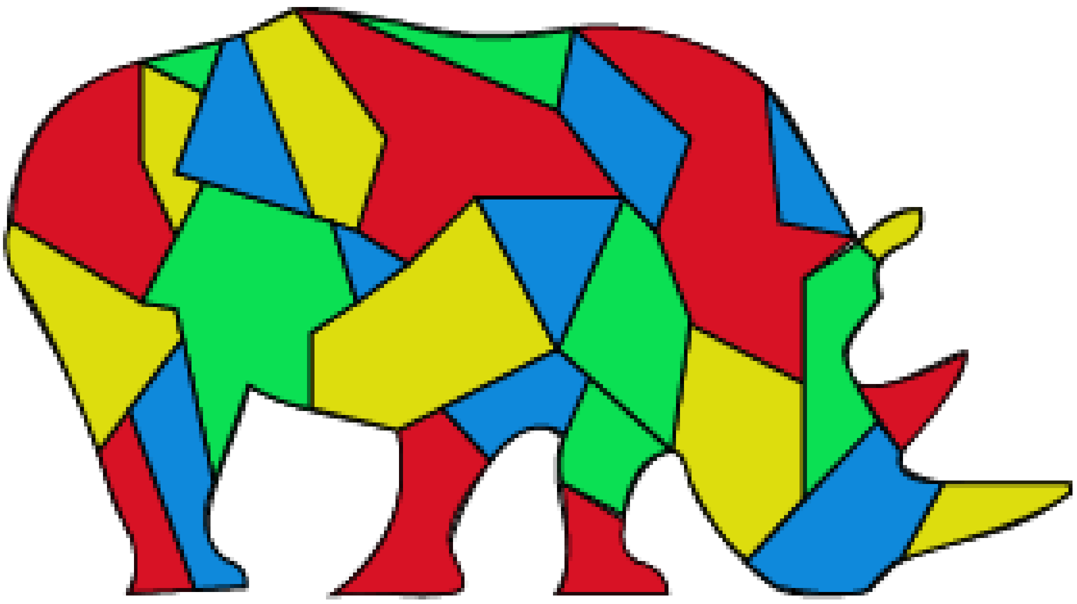
The following games will be developed:

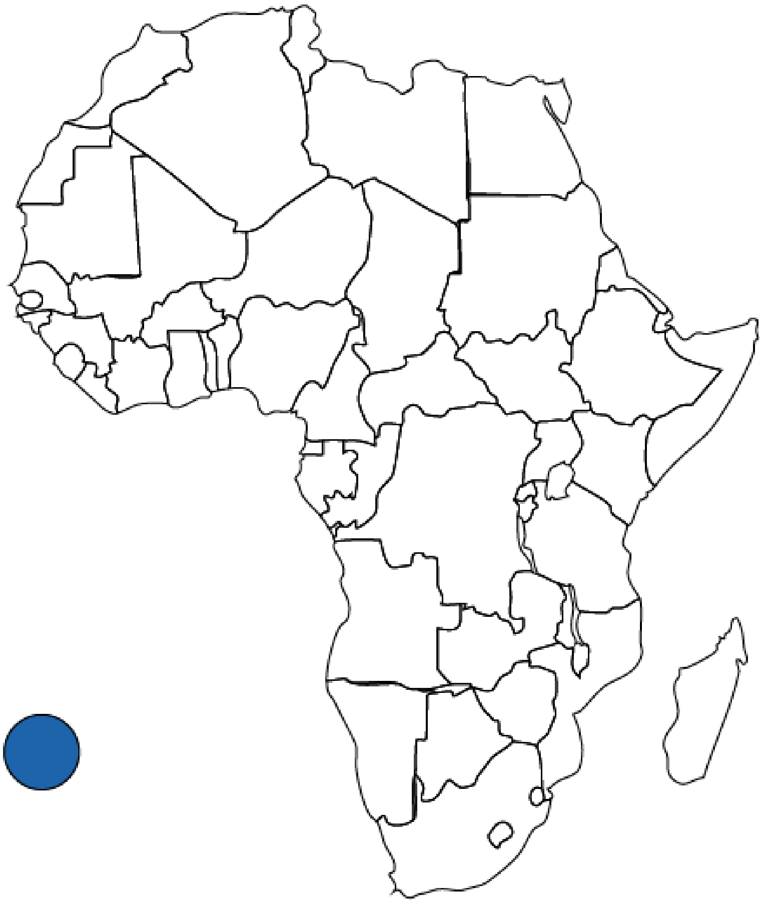
1. **Four Color Map Game**

- Based on the Four Color Theorem.
- Teaches children about map coloring and graph theory.

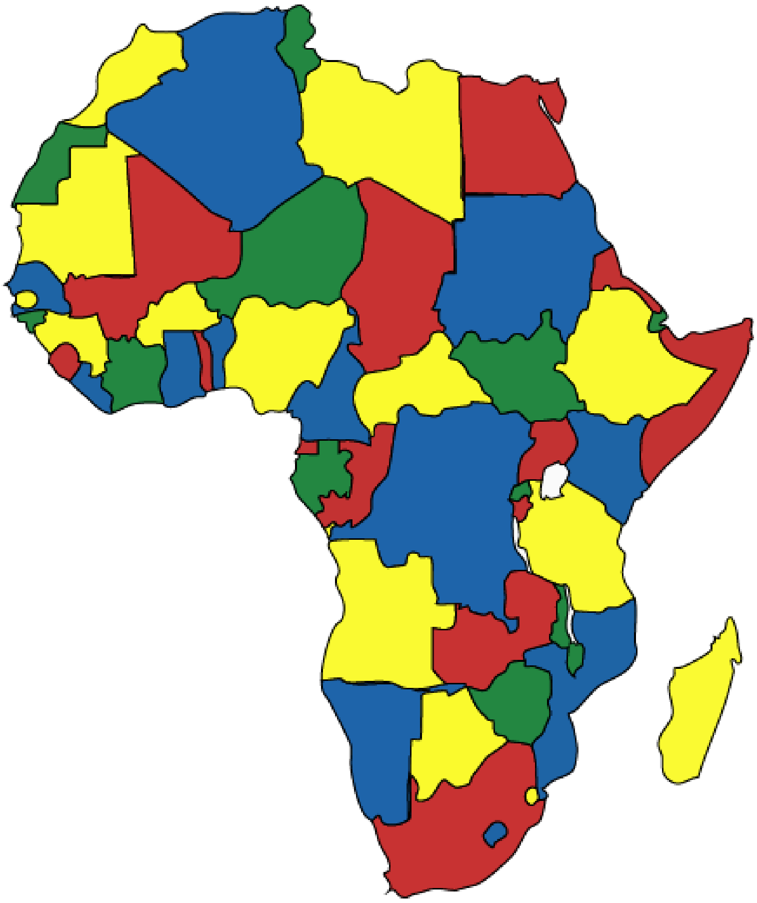
Here is a basic prototype I designed in Adobe Illustrator for the Four Color Map Game. To make the game more interactive and visually engaging, I incorporated the shape of a rhinoceros and the African continent as the playing areas, demonstrating how the four-color theorem can be applied in a fun and educational way.







🔴 🟡 🟢 🔵



Implementation Plan for the Four Colour Map Game

1. **Development Environment Setup**

To begin development, I will set up the necessary tools and frameworks required for a Sugar Activity:

- Install Python and Sugar Toolkit to develop within the Sugar ecosystem.
- sudo apt install python3 python3-sugar3
- Initialize a new Sugar Activity using the sugar-activity create command:
- sugar-activity create fourcolourmap
- cd fourcolourmap
- Set up a virtual environment to manage dependencies and ensure compatibility with Sugar's framework.

2. **UI/UX Design with Adobe Illustrator**

To create a visually appealing and interactive game interface:

- Design an interactive map in Adobe Illustrator using distinct vector regions.
- Export as an SVG for rendering in the game.
- Create an intuitive UI with a color palette, reset button, and drag-and-drop function.

3. Game Logic & Rendering

- Use GTK and Cairo to render the SVG map and enable region selection.
- Implement color selection and application, enforcing the Four Colour Theorem.
- Highlight incorrect moves and provide instant feedback.

4. UI Components & Features

- Color Palette – Four predefined colors for selection.
- Reset Button – Allows restarting the game.
- Hint System – Provides guidance for solving the puzzle.

5. Testing & Optimization

- Run the game in Sugar Emulator:
- bash
- CopyEdit
- sudo apt install sugar-emulator
- Ensure smooth performance, accurate click detection, and UI responsiveness.
- Optimize rendering and color detection for better usability.

6. **Packaging & Submission**

- Bundle the activity in Sugar’s .xo format.
- Submit for review and iterate based on mentor feedback.

Conclusion:

This structured plan ensures a well-designed and engaging educational activity, helping children understand the Four Colour Theorem interactively while refining my skills in UI/UX design, game development, and Sugar activity programming.

2. Broken Calculator

- A puzzle game using restricted mathematical operations to reach target numbers.
- Develops problem-solving and creative mathematical thinking

Implementation Plan for the Broken Calculator Game

1. Development Setup

Initialize a new Sugar Activity

2. UI/UX Design

- Design the calculator interface in Adobe Illustrator, incorporating broken keys to reflect the challenge.
- Export as an SVG for rendering in the game.
- Develop an intuitive UI that includes:
  - A calculator layout with missing or disabled keys.
  - A display screen for calculations.
  - A restart button to reset the challenge.

3. Game Logic & Rendering

- Use GTK and Cairo to render the calculator and handle interactions.
- Implement interactive buttons where only certain numbers and operations function.
- Develop core logic:
  - Generate a random target number.
  - Allow users to reach it using a limited set of keys.
  - Validate input and check if the correct target number is achieved.

4. UI Components & Features

- **Non-functional Keypad – Some numbers and operations are missing.**
- **Display Screen – Shows user inputs and calculations.**
- **Reset Button – Allows restarting the game.**

5. Testing & Optimization

- Run the game in Sugar Emulator:
- Ensure smooth performance, accurate input handling, and a responsive UI.
- Optimize number operations and game logic for an engaging experience.

6. Packaging & Submission

- Bundle the activity in Sugar's .xo format.
- Submit for review and iterate based on mentor feedback.

Conclusion:

This structured approach ensures an engaging and educational experience that sharpens mathematical reasoning while also refining my skills in UI/UX design, game development, and Sugar activity programming.

3.Soma Cubes Game

- A 3D puzzle game requiring players to assemble shapes using Soma Cubes.
- Improves spatial reasoning and geometric problem-solving.

Implementation Plan for the Soma Cubes Game

1. Development Setup

- Initialize a new Sugar activity:
- Set up a virtual environment for managing dependencies.

2. UI/UX Design

- Design 2D representations of the Soma cube pieces using Adobe Illustrator.
- Export the pieces as SVG files for rendering in the game.
- Develop an intuitive UI, including:
- A drag-and-drop system for placing cube pieces.
- A grid-based workspace to assist alignment.
- A reset button to clear the board.

- **3. Game Logic & Rendering**

- Use GTK and Cairo to render the 2D pieces and allow interaction.
- Implement drag-and-drop mechanics for players to position and rotate pieces.
- Check for correct assembly by verifying the arrangement of pieces.
- Provide instant feedback when the puzzle is solved correctly.

- **4. UI Components & Features**
-
- Piece Selection Panel – Displays the seven Soma cube pieces.
- Drag-and-Drop Interaction – Allows players to move and rotate pieces.
- Reset Button – Resets the board for a new attempt.

5. Testing & Optimization

- Run the game in the Sugar Emulator to test functionality:
- sugar-emulator
- Ensure smooth drag-and-drop mechanics, accurate piece placement, and UI responsiveness.
- Optimize performance for seamless interactions.

6. Packaging & Submission

- Bundle the activity in Sugar’s .xo format:
- Submit the game for mentor review and iterate based on feedback.

Conclusion

- By following this structured plan, I will create an engaging Soma Cubes puzzle that enhances problem-solving skills through interactive play. This project will refine my skills in UI/UX design, game development, and Sugar activity programming while making learning fun

4. Fifteen Puzzle

- A classic sliding puzzle where players arrange numbered tiles correctly.
- Strengthens logical sequencing and strategic planning.

Implementation Plan for the Fifteen Puzzle Game

. Development Setup

- Initialize a new Sugar Activity
- Set up the environment to manage dependencies.

2. UI/UX Design

- Design a 4×4 grid puzzle layout in Adobe Illustrator.
- Export the design as an SVG file for smooth rendering.
- Create an intuitive drag-and-slide interface to allow seamless tile movement.

3. Game Logic & Rendering

- Load and render the SVG grid using GTK and Cairo.
- Implement logic for detecting valid tile moves and shifting tiles accordingly.
- Ensure a randomized solvable puzzle generation at the start of each game.
- Track the number of moves and display completion time for user feedback.

4. UI Components & Features

- Sliding Mechanism – Allows smooth tile movement into empty spaces.

- Move Counter & Timer – Displays the number of moves and time taken to complete the puzzle.

5. Testing & Optimization

- Run the activity within the Sugar environment to ensure proper functioning.
- Test for:
  - Correct tile movement detection.
  - Smooth transitions between tile shifts.
  - Puzzle solvability checks before shuffling.
- Optimize rendering and minimize lag for better user experience.

6. Packaging & Submission

- Bundle the game in Sugar’s .xo format for deployment.
- Submit the activity for review and feedback, iterating based on suggestions.

Conclusion

This structured plan ensures the development of a fun and educational puzzle game, enhancing problem-solving skills while refining my expertise in UI/UX design, game logic implementation, and Sugar activity programming.

Here is the detailed implementation plan for the first four games, showcasing my structured approach. The rest of the games will follow a similar methodology, which I will refine further after discussing with mentors.

Below is a brief overview of their implementation plans.

Euclid's Game – Players iteratively subtract the smaller number from the larger one, reinforcing the concept of the greatest common divisor. The game logic will handle input validation and provide hints for optimal moves.

Odd Scoring – A strategic number-based game where players move a chip leftward, aiming to achieve an even step count. The system will enforce valid moves and display real-time progress.

Make an Identity – Players manipulate mathematical expressions to form a given identity. A dynamic equation parser will validate inputs and provide interactive feedback.

Number Detective – A pattern-recognition game where players identify hidden numerical rules. The game will feature randomized puzzles and provide hints to guide learning.

Sorting Hat AI – A classification-based game where players assign items to correct categories, integrating basic AI-driven suggestions for learning reinforcement.

TIMELINE:

Community Bonding Period (Before Coding Starts)

May 25 – June 9, 2025

- Familiarize myself with Sugar Labs' development guidelines and codebase.
- Engage with mentors and the community to refine implementation plans.
- Finalize the UI/UX designs for all games using Adobe Illustrator.

Weeks 1-2 – Initial Development & Core Implementations

June 10 – June 23, 2025

- Develop the Four Colour Map Game with basic functionality, including color selection and validation.
- Implement the Broken Calculator logic for interactive arithmetic challenges.
- Begin working on the Soma Cubes game, implementing shape placement and validation mechanics.
- Conduct early-stage testing for bugs and usability issues.

Weeks 3-4 – Expanding Functionality & Refinement

June 24 – July 7, 2025

- Complete Soma Cubes gameplay interactions and piece arrangements.
- Develop the Fifteen Puzzle, ensuring smooth tile movement and win conditions.

- Implement Odd Scoring, including game logic and turn-based mechanics.
- Gather feedback from mentors and refine existing games.

Weeks 5-8 – Mid-Term Evaluation & Further Development

July 8 – August 4, 2025

- Complete development of Euclid’s Game, refining number selection and game logic.
- Implement Make an Identity and Number Detective, ensuring they follow intuitive gameplay mechanics.
- Optimize all games for performance, responsiveness, and accessibility within the Sugar environment.

Weeks 9-12 – Final Features, Optimization & Submission

August 5 – August 19, 2025)

- Conduct thorough testing to ensure all games function smoothly.
- Write detailed documentation for users and future contributors.
- Package the activities in the Sugar .xo format and submit for review.

Address final mentor feedback before submission.

- ## COMMITMENTS:

I would like to inform the mentors that my university examinations are expected to take place between late April and early May. However, I am fully prepared for them, and I anticipate that my GSoC timeline will not be disrupted for more than six to seven days. During this period, I will remain actively engaged in researching and refining my project approach.

Beyond this, I want to emphasize that I am fully committed to this project and willing to put in extra effort, time, and dedication to ensure its success. Being a part of Sugar Labs and contributing to this initiative is a top priority for me, and I am ready to give my best to make a meaningful impact. If necessary, I will compensate for any lost time by working additional hours after my examinations.

- ## INTRODUCTION:

I am a passionate and dedicated developer with a strong enthusiasm for technology, coding, and interactive design. As the head of the graphics team in my college's media club, I have worked extensively on digital vector design, UI/UX development, and visual storytelling, creating engaging visual assets, user interfaces, and interactive experiences. My deep understanding of composition, color theory, and user engagement allows me to design interfaces that are not only aesthetically appealing but also highly intuitive. Beyond design, I have been fascinated by games and interactive learning since childhood, which has driven me to explore game mechanics, AI-driven experiences, and digital education tools. My technical expertise includes Python, GTK, TypeScript, and JavaScript, aligning seamlessly with this project's requirements. By integrating my design sensibilities with my programming skills, I aim to create engaging and interactive math activities that foster exploration, problem-solving, and AI-based learning.

- ## CONCLUSION:

This Google Summer of Code project aims to develop and integrate eight interactive math activities into the Sugar Labs learning platform, enhancing its educational value and engagement. By leveraging Python, GTK, TypeScript, and JavaScript, this project will introduce AI-driven learning experiences, logic-based puzzles, and problem-solving challenges, expanding Sugar’s interactive ecosystem.

The success of this project will depend on thorough research, seamless integration with existing Sugar activities, and iterative development based on community feedback. By combining my technical expertise in coding and UI/UX design, I am committed to creating high-quality, reusable activities that align with Sugar Labs’ mission of fostering exploratory and interactive learning.