Maintaining 6 activities
SugarLabs GSoC 2022 Proposal

Basic Details

- **Name:** Sparsh Goenka
- **Email:** sparshgoenka@gmail.com
- **GitHub Username:** sparshg
- **Primary Languages:** English, Hindi, Punjabi
- **Location and Timezone:** India (IST)

Why SugarLabs?

I am a first-year undergraduate student pursuing Computer Science at Birla Institute of Technology & Science, Pilani. I am fascinated by the idea of open sourced projects, and how anyone can access and contribute to the project's code. I liked Sugar Labs' approach that gives you a platform to provide educational opportunities to children. I also tried testing sugar activities with some primary school students, with help of my friends at NSS (BITS Pilani).

Past work on open source projects

- I have been contributing to Manim (community edition) for the past year, a community-maintained Python framework for creating precise mathematical animations programmatically, as demonstrated in the videos of 3Blue1Brown.

  Here is some of my past work related to this.

- I have been contributing to SugarLabs for the past month, trying to constantly improve sugar activities and get the hang of the codebase. My past work till now includes contributions to:
    - TurtlePond
    - Recall
    - Yupana
    - Flip
    - BlockParty

Project Details

Sugar has a lot of activities, with 250+ on GitHub, and more elsewhere. I would like to improve and maintain 6 sugar activities (or more), fixing various bugs and adding features to improve the user experience. To shortlist few, I would like to take up the following activities:

- ## Math Hurdler

The horse in this activity jumps over the hurdles when the correct answer is clicked. I want to improve the horse sprite and add proper running animation to it. Also, the horse remains in the air once you click on the correct answer. I would like to fix this by increasing the speed of the hurdle below once correct answer is given. Also, UI can be significantly improved by remaking sprites and adding textures, like for grass and sky.

- ## JAmath

In this, the numbers seem to blend with the background, making them harder to see. Background can be added to the numbers to provide better contrast. Also, I would like to add the feature where you can type on the keyboard to answer instead of having to point and click.

- ## Number Rush

This activity still runs on python 2 and requires porting to python 3. I would also like to add a gravity effect to the falling balls, making them look more realistic. The paddle currently, snaps to 4 places only. I would like to add a feature to make it move smoothly, something similar to paddle in Atari Breakout game.

- ## Recall

In this activity, I would like to centralize the UI elements. The game directly starts without giving any instructions. I will add some basic alerts/instructions so that the user knows what to do in each mode. Also, add a difficulty selector which will increase the number of images slowly or more rapidly in subsequent levels.

- ## Flappy Bird

In this, I would like to add a parallax effect for the background and also add mouse button for tap functionality. To add the parallax illusion, background needs to be scrolled at a slower pace as the game proceeds. An option to choose difficulties can be added to vary the gap between the upper and lower pipes.

- ## Block Party

For this activity, I would like to draw a grid of squares, that will indicate where exactly the block is going to drop. Also, I would like to improve on the “speedup on down key held” mechanic I added in a past PR, as the speedup begins in next tick which resulted in a bit of latency.

Tools and technologies:

Python, PyGame, GTK 3, other activity specific modules.

How will it impact Sugar labs?

This project will provide users with a better overall experience and also help to keep the code up to date for atleast 6 fully functional activities with strong pedagogical values. Also, as Python 2 is no longer supported, porting activities to Python 3 will help to release them on ASLOv4.

Timeline

Community Bonding Period

During this period, I will interact with my allotted mentor and look at the code for these activities and parts of code I can improve on. Also, in this period, I will try to understand the codebase in-depth and tinker around more.

Coding Period

- **Week 1 & 2 (June 13 - June 27)**
    - Work on Math Hurdler activity.
    - Discuss the bugs and features planned with the allotted mentors and community. For example, speeding up of hurdle once the correct answer is selected. The new animation for horse and textures for sprites, etc.
    - Implement in code according to discussion.
    - Resolve any unforeseeable errors.

- **Week 3 & 4 (June 27 - July 11)**
    - Work on JAmath activity.
    - Try to improve overall UI in the first week, adding background to numbers for better visibility and discuss the “type to answer” feature with community and mentor.
    - Add the "type to answer" feature as discissed in second week.
    - Take feedback for other improvements if time allows.
- **Week 5 & 6 (July 11 - July 25)**
    - Work on Number Rush activity.
    - Port activity to python 3.
    - Implement gravity to the balls, which would make the activity much more enjoyable.
    - This can be done by incrementing velocity each frame by some constant making the balls accelerate (because acceleration is the derivative of velocity).

Phase 1 Evaluation

By this time, I aim to complete 3 activities implementing all the features and bug fixes.

- **Week 7 & 8 (July 25 - August 8)**
    - Work on Recall activity.
    - As mentioned, centralize the UI to make it look more refined and also add the instructions to describe how to use the activity.
    - Add more images to for first two levels by adding a difficulty selector.
- **Week 9 & 10 (August 8 - August 22)**
    - Work on Flappy Bird activity.
    - Add parallax effect to background in the first week. This can be done by slowly scrolling the background as the game proceeds.

    - Add mouse to tap feature in the second week.
    - Work on more bug fixes and features if sufficient time remains.

- **Week 11 & 12 (August 22 - September 5)**
    - Work on Block Party activity.
    - Draw gridlines in the background in the first week.
    - Try to improve the controls for speedup key for the second week.
    - Optionally work on more bug fixes and features if sufficient time remains.

Final Evaluation (September 5 - September 12)

I aim to complete all the activities by now and the project should get completed by now

- Following such schedule, 6 activities can be worked upon in the span of 12 weeks, spending on average 2 weeks on each activity.
- This project will provide users with a better overall experience and also help to keep the code up to date.

Schedule

- My awake hours would usually be between 10 AM IST (4:30 AM UTC) to 1 AM IST (7:30 PM UTC) and I'm comfortable working anytime during this period.
- During this period, I will have mid-semester examinations between June 29 - July 02 and end-semester exams between August 03 - August 13. I may not be that active around this period. Exam dates for the next semester are not known yet, but I will inform at the latest.
- To report the progress between evaluations, I will create a GitHub markdown for the relevant contributions I made to the activities.
- After GSoC, I wish to continue contributing to various activities and try to keep them up to date.