sugarlabs

Google Summer of Code 2023 Proposal

Improve and Maintain 13 Sugar Activity

Basic Details

Name	Riya Jain
Email	riya1jain567@gmail.com
Github Username	jriyyya
Primary Language	English, Hindi
Location	India
Time Zone	Indian Standard Time (UTC+5:30)
Portfolio	bit.ly/jriyyya

Why Sugarlabs

I first came across sugar while browsing organisations which had been a part of GSoC 2022, It caught my eye in the first glance as my father is the director of an educational group of schools and the ideas behind sugarlabs and its products were focused on improving the children, student and teacher experiences, which was reflected, at least in my opinion, the most by Sugar, the desktop environment developed by sugarlabs. I once held a small session with about 15 students of my father's school in their computer lab, where I made then have a small interaction with sugarizer, mostly, the children appeared to be interested and after personally talking to them, they said they'd really love it if this was a part of their regular school activities. This motivated me to be a part of sugarlabs and contribute to sugar and help grow this open source project so that the children who are currently getting to use this may have a better experience. And call it selfish, but it just makes me feel extreme gratitude working towards something of a societal goal.

Furthermore, a major reason besides all these is that my skill set aligns with the sugar desktop environment perfectly. I've been working with python for a long time and most of the sugar environment relies on python3. I also have made a few personal projects using python and I had used pygame as a GUI library, and that's what sugar uses. Also, I've become familiar with the coding style and usual codebases of activities used in sugar as I've been working with these since January 2023.

Past Contribution in Sugarlabs

My contribution to Sugarlabs organisation till now

S.No.	Activity Name	Pull Request	Status
1.	make-them-fall-activity	**Refactored code, added new Button class, reduced re-renders**	**In Progress**
2.	classify-cats	**Features and bug fixes**	Merged
3.	block-party-activity	**Feature: Show highscore**	Merged
4.	tictactoe	**Feature: Added text to show current turn**	Merged
5.	sugar-toolkit-gtk3	**Fixed issue: Deprecated module optparse is used**	Merged
6.	stick-hero-activity	**Bug Fix: Initially Fruit score showed highscore**	Merged
7.	stick-hero-activity	**Bug Fix: Bug in perfect score**	Merged
8.	activity-abacus	**Bug Fix: Top left bead overflows on reset**	Merged
9.	flipsticks	**Fix Flake8 warnings**	Merged
10.	falabracman-activity	**Fixed Flake8 warnings** **More fixes of Flake8 warnings**	Merged
11.	distance-activity	**Fixed Flake8 warnings**	Merged
12.	flipsticks	**Changed Division operator to Floor division**	Merged
13.	cookie-search-activity	**Fixed Flake8 Warnings** **Flake8 Fixes**	Merged
14.	falabracman-activity	**Fixed PyGIWarning**	Merged
15.	stick-hero-activity	**Corrected a comment**	Merged
16.	block-party-activity	**Updated README.md**	Merged
17.	musicblocks	**Corrected a bash command in README.md**	Merged
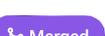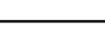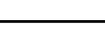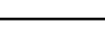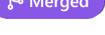
Project Details

Sugar has a lot of activities, with 250+ on GitHub, and more elsewhere. These have scope for improvement; bugs, features, updated human translations, and release. This project will involve working on 13 activities (or more) to improve them for a better user experience.

I have shortlisted these activities for spending my summer time with

1. classify-cats

Introducing a new game mode

In this new game mode, a grid of four different coloured cats with different counts will be displayed. One of the cats will have an odd count while the others will have even counts or one of them will have an even count and all others have an odd count. The player will be presented with three options to choose from in order to identify the cat which has the count as required by the question (even or odd)

This can be implemented by adding a new mode to the GameType and then creating a section for the new gamemode in each callback function. This would require tinkering with how the dividers are drawn as the drawing function for it is separate.

Choose the cat with **ODD** count
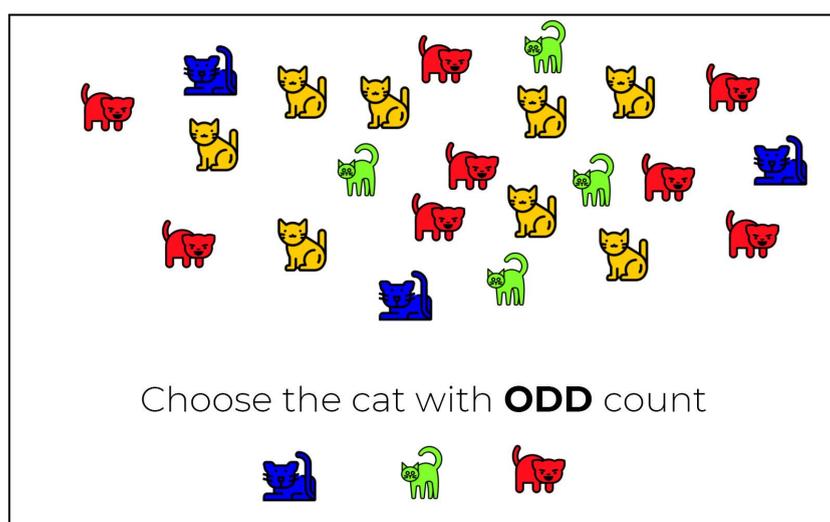
2. memorize-activity

Game over Screen:

To enhance the user experience and provide a sense of accomplishment, an end game screen with stats for a memorize-activity can be implemented. This end game screen can include various components such as, time taken to complete, score, high score and a button to replay

-> The time taken to complete the game can be displayed on the end game screen. It can also motivate the user to try to beat their previous time in subsequent games.
-> The score can be calculated by assigning a 5 point value to each correct match and deducting a point value for each incorrect guess. The time taken can also be considered, to add more weightage to the score.
-> Furthermore, including the high score on the end game screen provides an additional level of motivation for the user

3. ball-and-brick-activity

Power ups feature

The activity can use power ups as many breakout based games have this feature. A few powerups I had in mind which could be implemented are:

-> Extra ball -> 2 balls in the game at a time
-> Longer bat -> Increase the size of the bat
-> Smaller bat -> Bad luck powerup (reduces bat size making the game harder)
-> Extra life

The bat sizes could be implemented by modifying the scroller_w property
And life could be implemented by just adding to the life counter.
Extra ball can be instantiated by having a self.ball_2 just like self.ball and ball_2 will not have motion and will not be visible until the powerup is obtained.

4. block-party-activity

Ghost piece Feature

A ghost piece will be a translucent representation of the current block that shows where it will land if the player drops it instantly. This feature particularly helps players who want to plan their moves ahead of time and increase their gameplay speed

This would replicate the functionality of drop_fgiure method of BlockParty class and would instead just be used to calculate the new coordinates, and then the coordinates would be replaced with -1 in the board, and instead of rendering a colour, -1 would exceptionally render an outlined box..

Score fixing

As of now, no incentive is provided to the player for breaking a row of blocks, its track is being kept as 'lines' but it has no effect on the score and thus won't affect the high score if the player makes one so we can provide an extra score for it.
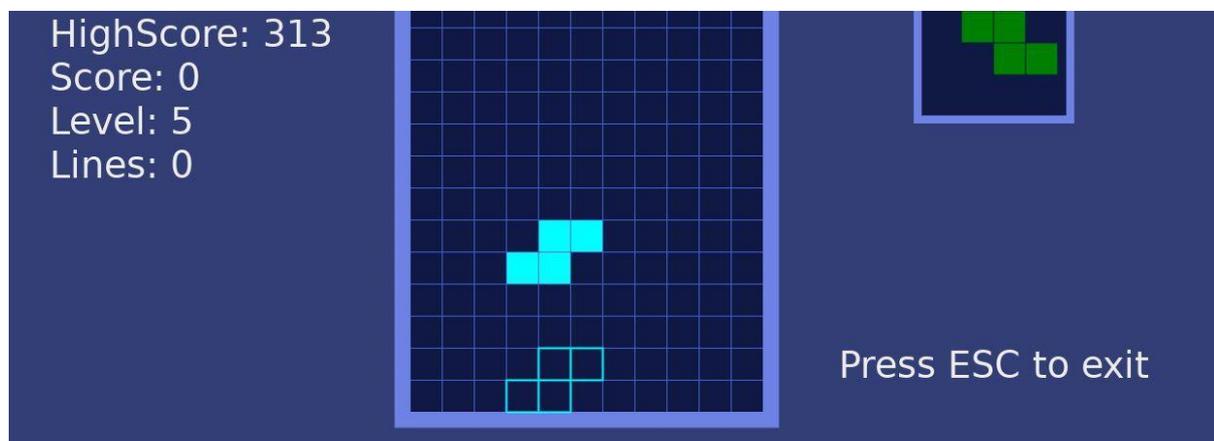
5. falabracman-activity

Fix Randomness:

To reduce the likelihood of encountering impossible path obstacles that prevent players from completing the word.

This occurs because the random generation of obstacles can result in a path that is impossible to navigate, which negatively impacts the user experience.

This can be implemented by having an algorithm which only chooses coordinates at least at a certain minimum distance.

Multilingual Support:

In changing the language from the menu, only the words which are used in gameplay are changed. The images like 'you lost' are still in spanish which should be updated according to the language selected

6. math-hurdler

Adding Difficulty levels:

The game could benefit from having different difficulty levels that the player can choose from, depending on their skill level and preference. This would add more variety and replay value to the game. The point value can be determined based on the difficulty level of the game, with harder levels assigning higher point values. This can be implemented as an option in the sugar toolbar which will produce a dropdown for the various difficulties. The difficulty selected will involve modifications to the generate_question function and the vx property of the MathHurdler class as the ramping would be much quicker in higher difficulty levels.

UI-enhancement:

The horse in the activity seems to be floating in mid-air, and the fetch is not aligned with the ground, which can be confusing for the player.

Secondly, The score is not very practical in terms of readability and visibility. It should be displayed in a more prominent and legible manner

Thirdly, the questions take up a lot of screen space and looks cluttered

7. sonic-jump-activity

Adding Help Animation:
The help page can be navigated through a button on the welcome screen. This improvement involves creating an animated tutorial that guides users through the gameplay.
This will be implemented in a manner similar to how it is done in activities like stick hero activity.

Enhancing the Welcome Screen and Game Over Screen UI:
This involves redesigning the user interface (UI) of the Home Screen and Game Over screens to make them more visually appealing, intuitive, and engaging.

8. hittheballs-activity

Implementing Game loop
A game loop can really do well with this activity as currently the game ends as soon as the correct answer is chosen. Implementing a game loop would mean that when the correct answer is chosen, points would be added to a score counter and the score will be displayed within the activity.

Feature: Score and Highscore
With the implementation of a game loop, the activity can also include a proper score system and a high score feature, the score will be implemented as specified above and the high score will be implemented using pickle and I will save this highscore in a file in the sugar activity root directory.

9. mancala-Activity

Feature: 2-player mode

The activity does not currently offer a feature that allows players to compete against each other on the same device i.e a two-player mode. The game only provides an option to play against an AI opponent.

To make this feature accessible, an option can be added to the toolbar that lets players choose between playing against an AI opponent or another human player. With this feature, players would have the flexibility to enjoy the game in either mode, depending on their preferences and availability of another player

This can be implemented by adding player counterparts to the "wizard" methods of the Manc and switching between the modes will determine if the wizard functionalities are called or second player functionalities are called .

10. countries-activity &
11. across-and-down-activity

Drawing the letters for the background image

As this issue is blocking the release of the activity, it is a critical priority to fix it. Currently, the alphabets in the background are drawn by using a png image and this poses quite a few problems such as the image not being able to maintain its scale in accordance to the rotated screens.

This can be avoided by placing the alphabets as font renders instead of using a png, since both these activities are in pygame it would be relatively easy to pull this off by creating a letter class and applying a little trigonometry to place them in a circular manner using pygame blit.

12. slider-puzzle-branch

Fixing Collaboration
In the current version of the activity, there is a problem with collaboration as the game state is not shared between players, and each instance displays a different puzzle. I intend on fixing this during my GSOC period.

Porting to Python3
Once the collaboration fixing is done, the activity now can successfully be ported to python3. This is because Python 2 is no longer being supported and is becoming increasingly obsolete, making it difficult to maintain and update the activity in the long term.

13. CowBulls-activity

Adding Collaboration
The activity currently does not support collaboration, and it can be added to it by sharing the state between different devices through the CollabWrapper library, which is implemented in many of the other sugar activities.

Note:
I might change the work to be done mentioned in the above section, with the guidance from the mentors.

Tools and Technologies
Python, PyGame, GTK3 and other activity specific modules.

Project Type
The Project type is large (350 hours)

How will it impact Sugarlabs

Upon the successful completion of this project, at least 13 sugar activities will have a better user experience. As user experience plays a vital role in any project, with the accomplishment of it, a user will be able to have a better interactive environment and will have fun exploring it.

Timeline

Community Bonding Period 4th May - 28th May	➢ Getting to know the code base of the selected activities ➢ Exploring and understanding Collabwrapper. ➢ Learning about the previous contribution of the activities whose port to python3, gtk3 etc. were not completed.
Week 1 29th May - 4th June	➢ Start working on the 1st activity i.e **classify-cats**, involving a new game mode in it. ➢ Take review from the mentors and apply changes if any are requested.
Week 2 5th June - 11th June	➢ Begin the work of the 2nd, **memorize-activity**, which involves implementing an end game screen. ➢ First implementing a end game screen and adding score to it, followed by taking reviews from mentors ➢ Later adding a high score to it. ➢ Proceeding with addition of time taken to it and completing it with a replay button.
Week 3 12th June - 18th June	➢ Start working on the 3rd and 4th activity. ➢ The 3rd activity, **ball-and-brick-activity** involves a powerup feature and the 4th activity, **block-party-activity,** involves ghost piece feature and score calculation fix. ➢ Get review on the work of 3rd activity and

	simultaneously fixing the score calculation of the 4th activity. ➢ Complete the task of the 4th activity.
Week 4 19th June - 25th June	➢ Get feedback on the 4th activity and look into any bugs or issues (if any) of the work which is already done in the first 4 activities. ➢ Start working on the 5th activity, **falabracman-activity**, which would first involve fixing the randomness and later adding multilingual support. ➢ Get a review on the work done in the 5th activity.
Week 5 26th June - 2nd July	➢ Begin the UI enhancement of the 6th activity, **math-hurdler** proceeding with adding different difficulty levels to it. ➢ Get feedback from mentors at every stage in the UI enhancement. And later getting overall feedback for the feature of different levels
Week 6 3rd July - 9th July	➢ Proceeding to the UI enhancement of the 7th Activity, **sonic-jump-activity** and getting the reviews from mentors. ➢ Later adding an animation for help in Sonic-jump-activity. ➢ Fixing issues (if any) and test the new features and maintenance done in the span of these 6 weeks.
Week 7 **Mid Term evaluation** 10th July - 16th July	➢ By this time, a total of 7 activities will be ready to be submitted for the Mid term evaluation. ➢ Continuing the work with the start of implementing Game loop in **hit-the-balls-activity.** ➢ Get feedback from mentors and complete the other task of adding score and highscore to it.
Week 8	➢ Proceeding to the 9th Activity which is

17th July - 23rd July	**mancala-activity**, involving adding a 2-player game mode. ➢ Receive feedback from the mentors and do changes accordingly if any are requested.
Week 9 24th July - 30th July	➢ Start with the 10th and 11th activity, which involves changing the background image to drawing the alphabets, of two activities, namely **across-and-down-activity** and **countries-activity** ➢ Get review from the mentor on the done task. ➢ Fix any bugs or issues noticed till now in these last 4 activities.
Week 10 31st July - 6th August	➢ Proceeding with understanding the code base of the 12th activity, **slider-puzzle-branch**, and trying to figure out the issue because of which the collaboration is broken. ➢ Fixing the collaboration later this week and getting feedback from the mentors regarding the same. ➢ Start working on the port to python3 for this activity
Week 11 7th August - 13th August	➢ Complete the port to python3 and get reviews from the mentors. ➢ Start working on the collaboration of the 13th activity, **cowandBulls**.
Week 12 14th August - 20th August	➢ Complete with the collaboration and get feedback from the mentors regarding the same. ➢ Look for any bugs or issues in the last 2 activities and fix them. ➢ Review all the work done till now
Final Evaluation **21st August - 28th August**	➢ By now a total of 13 activities will be ready for the final evaluation. ➢ Document the final work

Schedule

The GSoC timeline is mostly in sync with my university's summer break and thus will allow me ample time to work on my project. Even after the break ends there would be no tests or examinations as well as the classes would be asynchronous leaving me enough time to work.

My awake hours are between 10:00 IST (4:30 UTC) to 1:00 IST (19:30 UTC), and I would be reachable any time between them.

Due to unexpected situations, if I take a day off, I will inform the mentors in such cases.

Post GSoC plan

There are many other activities which need maintenance as there are more than 250 activities, so I wish to continue contributing to sugar and improve and modify them.

Other Commitments

➔ Due to minor university assessments in mid July, I won't be able to devote enough time to the project during that time. To make up for it, I will devote more time to the project during the other weeks so as to not interfere with deadlines.

➔ Sugarlabs is the only organisation I am applying for.