

Google Summer of Code 2023 Project Proposal

Sugar on Raspberry Pi

About Me

What is your name?

My name is Simrann Arora and I am a postgraduate student at Indraprastha Institute of Information Technology, Delhi (IIITD) in India.

What is your email address and GitHub username?

My email address is simrann2099@gmail.com and my GitHub username is [simrann20](https://github.com/simrann20).

What is your Matrix IRC username?

My Matrix IRC username is simrann20.

What is your first language? (We have mentors who speak multiple languages and can match you with one of them if you'd prefer.)

My first language of communication is Hindi, however I have full professional proficiency in English.

Where are you located, and what hours (UTC) do you tend to work? (We also try to match mentors by general time zone if possible.)

I am located in New Delhi, India and my time zone is Indian Standard Time (UTC + 5:30). I am planning to work from 6:00 to 14:00 (UTC) but my timings are flexible. I'm very excited to work on this project during the summer and I can surely manage my time and be active when the mentors are available.

Have you participated in an open-source project before? If so, please send us URLs to your profile pages for those projects, or some other demonstration of the work that you have done in open-source. If not, why do you want to work on an open-source project this summer?

Yes, I do have prior experience in open source development. I have contributed to multiple open source projects and it always has been a great experience collaborating with developers and open source enthusiasts around the globe. I also got selected for the prestigious open source program - GirlScript Summer of Code (GSSoC 2020) and was among the top contributors :

(<https://drive.google.com/file/d/1mBHKkFnP28Knz4IFiHHbPdFH6C7YLySF/view>) there

with 10+ pull requests in OpenGenus' Cosmos and other projects. Furthermore, I have been taking part in Hacktoberfest almost every year and contributing to open source projects there. Here are my open source contributions in some of the projects:

OpenGenus/Cosmos

Issues Created:

- <https://github.com/OpenGenus/cosmos/issues/5411>
- <https://github.com/OpenGenus/cosmos/issues/5456>

Pull Requests Merged:

- <https://github.com/OpenGenus/cosmos/pull/5766>
- <https://github.com/OpenGenus/cosmos/pull/5764>
- <https://github.com/OpenGenus/cosmos/pull/5763>
- <https://github.com/OpenGenus/cosmos/pull/5762>
- <https://github.com/OpenGenus/cosmos/pull/5709>
- <https://github.com/OpenGenus/cosmos/pull/5705>
- <https://github.com/OpenGenus/cosmos/pull/5704>
- <https://github.com/OpenGenus/cosmos/pull/5457>
- <https://github.com/OpenGenus/cosmos/pull/5768>
- <https://github.com/OpenGenus/cosmos/pull/5596>
- <https://github.com/OpenGenus/cosmos/pull/5412>

Open-Source-Contributors-JSS/Hacktoberfest2019

- <https://github.com/Open-Source-Contributors-JSS/Hacktoberfest2019/pull/250>

acmbvp/Hacktoberfest

- <https://github.com/acmbvp/Hacktoberfest/pull/274>
- <https://github.com/acmbvp/Hacktoberfest/pull/264>
- <https://github.com/acmbvp/Hacktoberfest/pull/163>

DevsWhoCode/updated-website

- <https://github.com/DevsWhoCode/updated-website/pull/20>
- <https://github.com/DevsWhoCode/updated-website/pull/21>
- <https://github.com/DevsWhoCode/updated-website/pull/34>
- <https://github.com/DevsWhoCode/updated-website/pull/62>
- <https://github.com/DevsWhoCode/updated-website/pull/68>

Convince us that you will be a good fit for this project, by sharing links to your contribution to Sugar Labs.

I am quite passionate about contributing to Sugar Labs and have been rigorously testing Sugar on Raspberry Pi to find bugs, which I will be working on fixing during the GSoC period. I have also been actively collaborating with the mentors and sharing my findings and reports with them. Rather than jumping straight away on making pull requests, I first thoroughly tested out everything and sought feedback from the mentors to ensure that I am at the same page as them and I don't miss out on any possible bugs and addressed them all to make Sugar run impeccably on the Pi. I will be working on the project and plan to make contributions leading up to the GSoC period, that is, even after submitting my proposal.

About your project

We are looking for projects that will enhance the Sugar Learning Platform. Please consider how your project will have an impact on children's learning.

What is the name of your project?

The name of my project is **Sugar on Raspberry Pi**.

Describe your project in 10-20 sentences. What are you making? Who are you making it for, and why do they need it? What technologies (programming languages, etc.) will you be using?

Project Description:

The Raspberry Pi is a system on chip (SoC) computer that can be used for a variety of projects and has been heralded as a great boon to education due to its low cost, flexibility and simplicity. The Raspberry Pi provides a very robust platform for educational institutions to incorporate new technologies in education especially when teaching children how to program and it is because of this functionality of the Pi that it has seen widespread adoption in schools. It would be a marvelous idea if we could get Sugar to perfectly run on the Raspberry Pi as it would provide students and teachers with an intuitive and interactive learning environment.

Aim of the project:

The aim of this project is to identify broken activities and bugs in Sugar on the Raspberry Pi operating system and fix them to make Sugar run perfectly on the operating system. The project also aims at extending the functionalities of the current

activities such as Turtle Blocks, Measure, and Pippy using the sensors that are available on the Raspberry Pi.

Project Goals and Tasks:

The goals and tasks of the project have been listed as follows:

Goals:

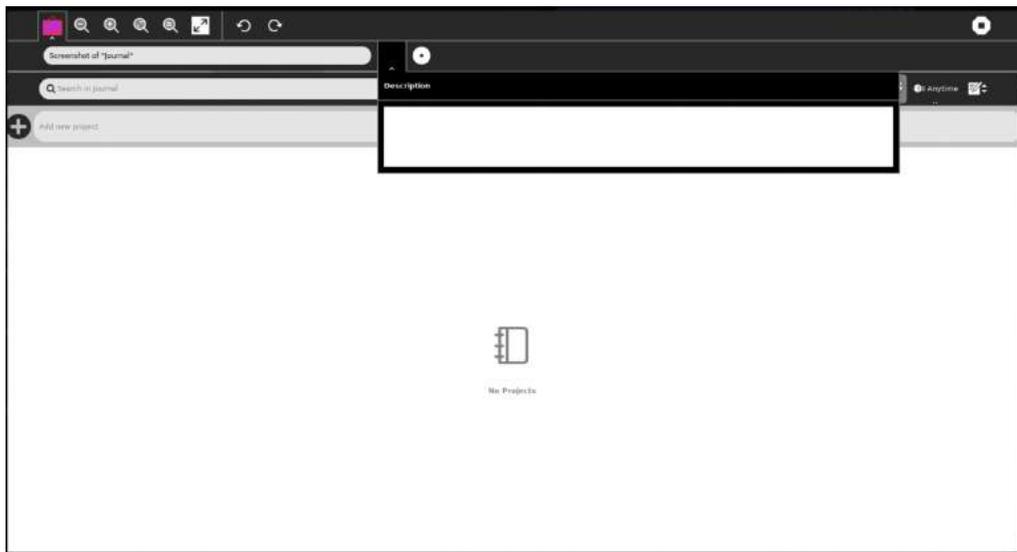
- Identifying the activities that are broken/behaving erratically and fixing them.
- Extending the functionalities for the activities using the sensors available on Raspberry Pi.

Tasks:

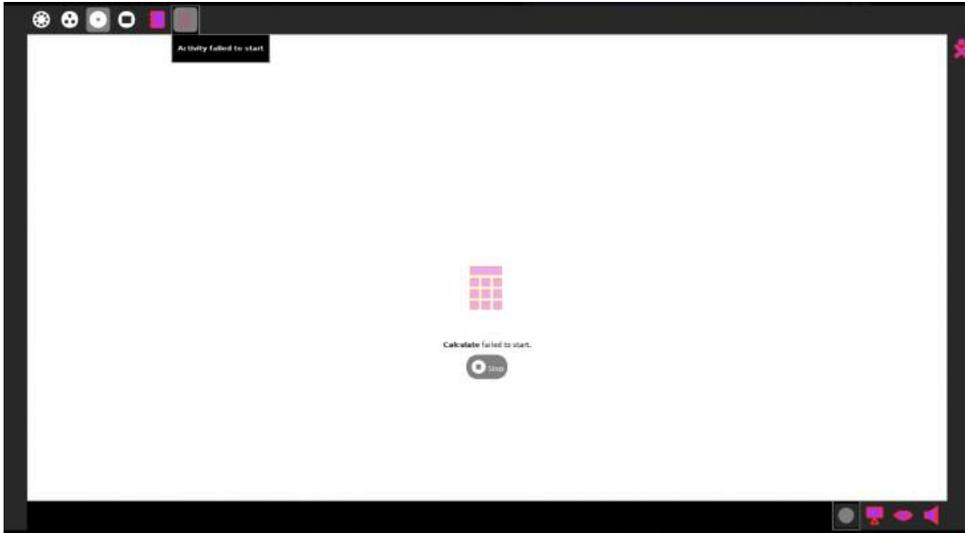
Task 1: Fix all the broken activities/bugs in the Sugar environment.

The following activities have been broken or are displaying erratic behavior:

1. While testing Sugar I found that the Description button does not show any icon and is not visible. Only if someone clicks over the black space, the Description pop up opens. **(Stretch Goal)**



2. The Calculator activity was showing erratic behavior and failed to load after a long delay.



Following were the error logs:

```
(sugar-activity:17899): Gtk-CRITICAL **: 21:22:36.845: gtk_toolbar_remove: assertion 'content_to_remove != NULL' failed
```

```
(sugar-activity:17899): Gtk-CRITICAL **: 21:22:36.846: gtk_toolbar_remove: assertion 'content_to_remove != NULL' failed
```

```
(sugar-activity:17899): Gtk-CRITICAL **: 21:22:36.846: gtk_toolbar_remove: assertion 'content_to_remove != NULL' failed
```

```
(sugar-activity:17899): Gtk-CRITICAL **: 21:22:36.846: gtk_toolbar_remove: assertion 'content_to_remove != NULL' failed
```

```
(sugar-activity:17899): Gtk-CRITICAL **: 21:22:36.846: gtk_toolbar_remove: assertion 'content_to_remove != NULL' failed
```

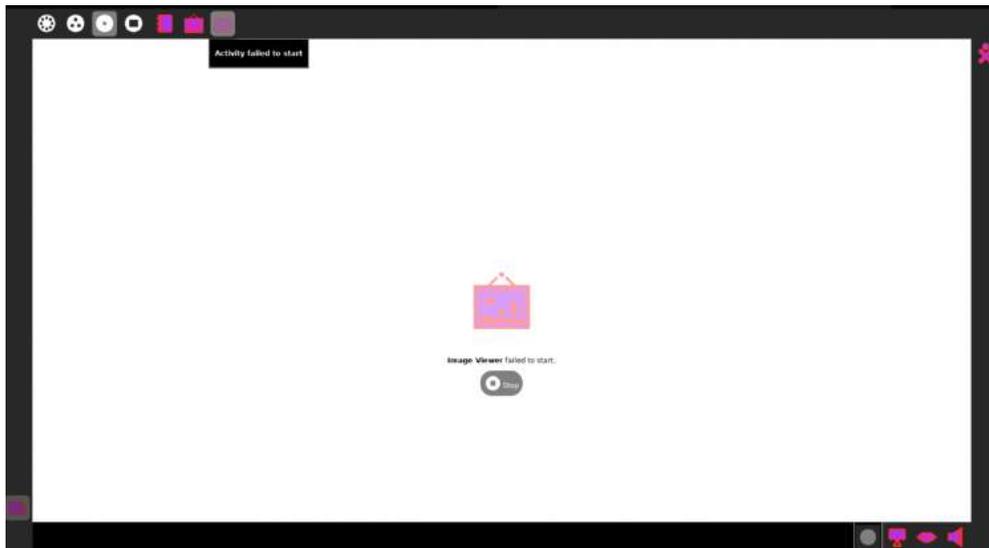
```
(sugar-activity:17899): Gtk-CRITICAL **: 21:22:36.846: gtk_toolbar_remove: assertion 'content_to_remove != NULL' failed
```

```
(sugar-activity:17899): Gtk-CRITICAL **: 21:22:36.846: gtk_toolbar_remove: assertion 'content_to_remove != NULL' failed
```

```
(sugar-activity:17899): Gtk-WARNING **: 21:22:46.719: Don't know color 'Gdk.Color(red=0, green=0, blue=0)'
```

```
(sugar-activity:17899): Gtk-WARNING **: 21:23:12.657: Drawing a gadget with negative dimensions. Did you forget to allocate a size? (node menuitem owner SugarPaletteHeader)
Exited with status 0, pid 17899 data (<open file '<fdopen>', mode 'w' at 0x4c0ed3e8>, '98fb5a25db5beb85835e4f75b415877ac0077cc1')
```

3. Image viewer failed to load as shown in the screenshot:



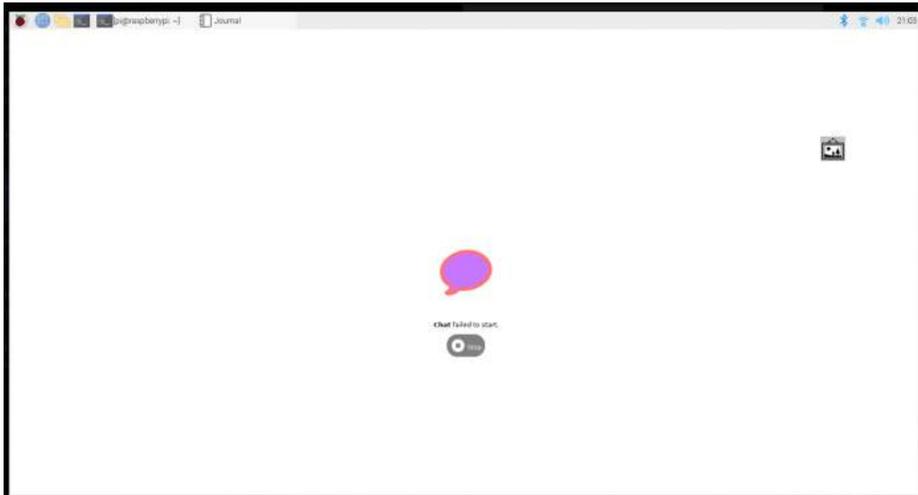
Following were the error logs:

```
Exited with status 0, pid 15014 data (<open file '<fdopen>', mode 'w' at 0x6c3a0180>, '3488725a703565e3762d107678fe34de8e5c5fc2')
```

4. No display icon is there for Projects. It only appears when we hover over the black spot and click on that.



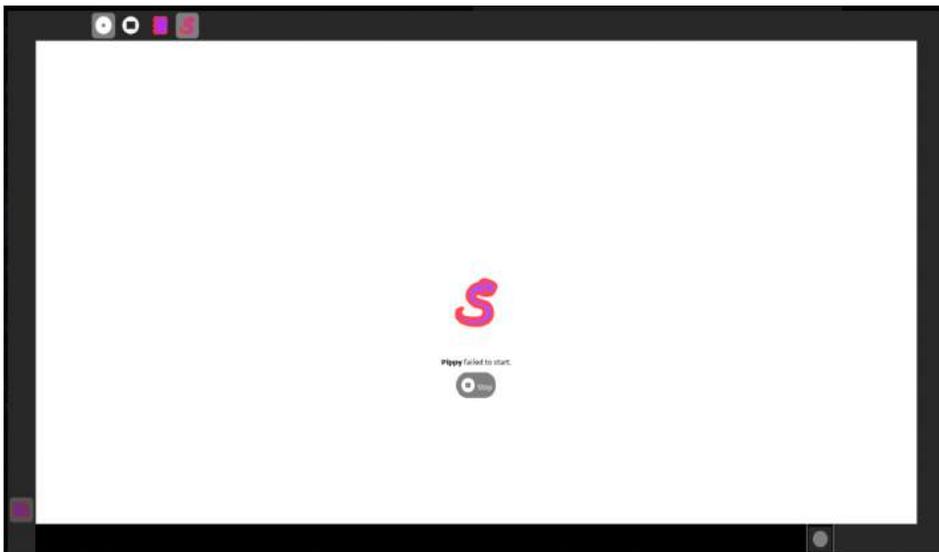
5. Chat activity failed to start.



Following were the error logs:

Exited with status 0, pid 21434 data (<open file '<fdopen>', mode 'w' at 0x4c0ed338>, '801ac2dce6674e6a37e826a6c3c1f6eb2c8058de')

6. Pippy activity failed to start.



Following were the error logs:

warning: check: missing required meta-data: url

warning: check: missing meta-data: either (author and author_email) or (maintainer and maintainer_email) must be supplied

warning: sdist: manifest template 'MANIFEST.in' does not exist (using default file list)

warning: sdist: standard file not found: should have one of README, README.txt

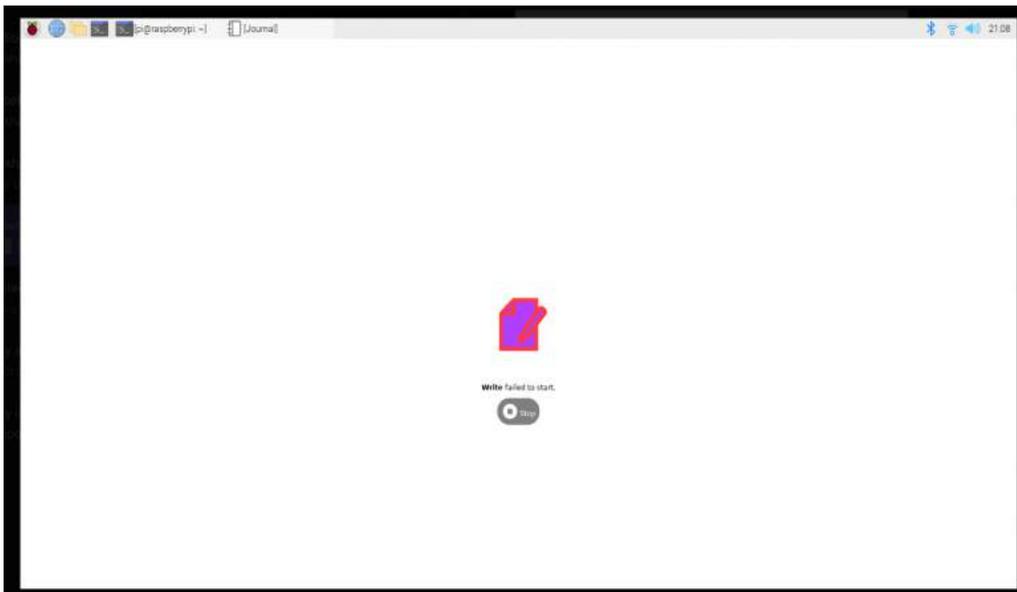
warning: check: missing required meta-data: url

warning: check: missing meta-data: either (author and author_email) or (maintainer and maintainer_email) must be supplied

warning: sdist: manifest template 'MANIFEST.in' does not exist (using default file list)

warning: sdist: standard file not found: should have one of README, README.txt

7. Write activity failed to start:



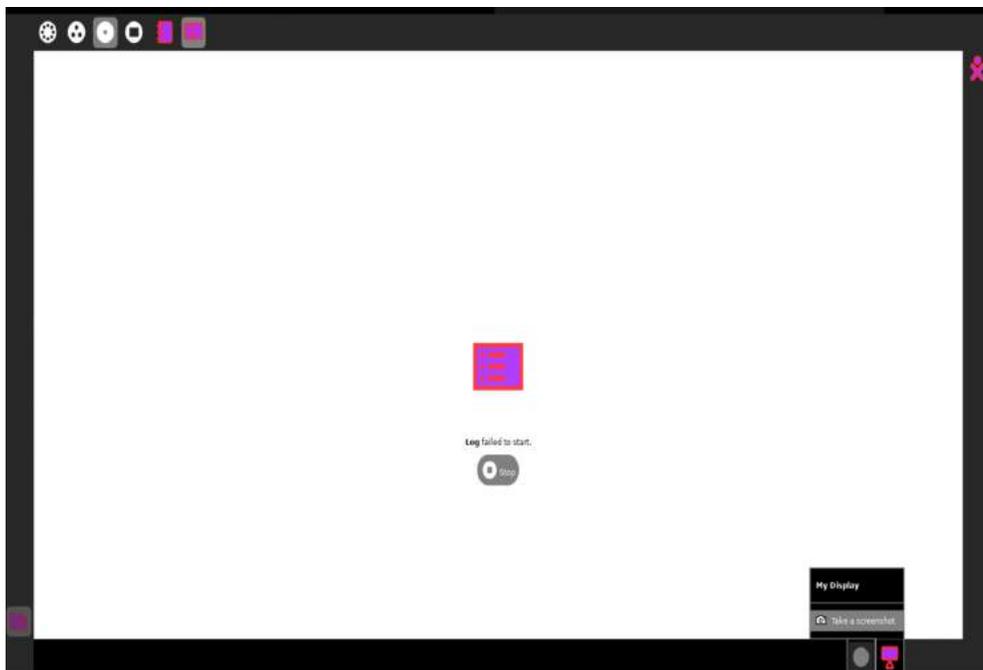
Following were the error logs:

```
1680191869.733093 ERROR root: EXCEPTION: request-clear-area signal not available
1680191869.734055 ERROR root: EXCEPTION: unset-clear-area signal not available
1680191870.235375 ERROR root: Abiword version 3.0
1680191870.247839 ERROR root: Abiword version 3.0
1680191870.258700 ERROR root: Abiword version 3.0
```

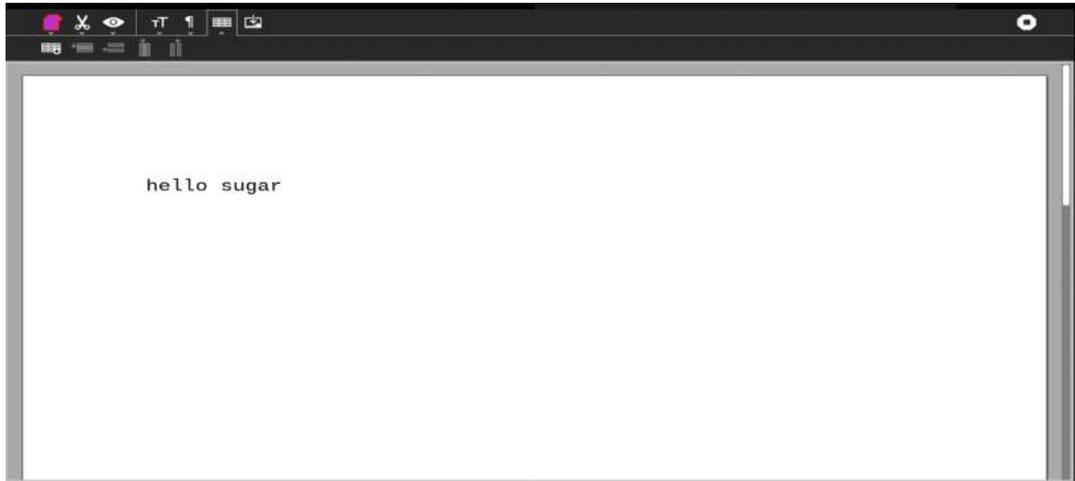


```
button.get_palette().popup(True, button.get_palette().SECONDARY)
AttributeError: 'Palette' object has no attribute 'SECONDARY'
Traceback (most recent call last):
  File "/usr/share/sugar/activities/Write.activity/toolbar.py", line 266, in _table_btn_clicked_cb
    button.get_palette().popup(True, button.get_palette().SECONDARY)
AttributeError: 'Palette' object has no attribute 'SECONDARY'
Traceback (most recent call last):
  File "/usr/share/sugar/activities/Write.activity/toolbar.py", line 266, in _table_btn_clicked_cb
    button.get_palette().popup(True, button.get_palette().SECONDARY)
AttributeError: 'Palette' object has no attribute 'SECONDARY'
Traceback (most recent call last):
  File "/usr/share/sugar/activities/Write.activity/toolbar.py", line 266, in _table_btn_clicked_cb
    button.get_palette().popup(True, button.get_palette().SECONDARY)
AttributeError: 'Palette' object has no attribute 'SECONDARY'
Traceback (most recent call last):
  File "/usr/share/sugar/activities/Write.activity/toolbar.py", line 266, in _table_btn_clicked_cb
    button.get_palette().popup(True, button.get_palette().SECONDARY)
AttributeError: 'Palette' object has no attribute 'SECONDARY'
Traceback (most recent call last):
  File "/usr/share/sugar/activities/Write.activity/toolbar.py", line 266, in _table_btn_clicked_cb
    button.get_palette().popup(True, button.get_palette().SECONDARY)
AttributeError: 'Palette' object has no attribute 'SECONDARY'
Traceback (most recent call last):
  File "/usr/share/sugar/activities/Write.activity/toolbar.py", line 266, in _table_btn_clicked_cb
    button.get_palette().popup(True, button.get_palette().SECONDARY)
AttributeError: 'Palette' object has no attribute 'SECONDARY'
```

8. Log failed to start:



9. When Write activity finally started to work after re-running Sugar, I could not add tables through the add table option.



10. Inside the Sugar terminal the following command was not running as it could not locate the Sugar package :

```
sudo apt install sugar-{abacus,browse,calculate,chart,chat,clock,\
develop,finance,findwords,fototoon,fractionbounce,gears,\
imageviewer,implode,jukebox,letters,log,maze,measure,memorize,\
moon,music-keyboard,paint,physics,pippy,poll,portfolio,read,\
Record,speak,stopwatch,story,terminal,words,write}-activity
```

```
pi@raspberrypi:~$ sudo apt-get install sugar-{abacus,browse,calculate,chart,chat,clock, \
develop,finance,findwords,fototoon,fractionbounce,gears, \
imageviewer,implode,jukebox,letters,log,maze,measure,memorize, \
moon,music-keyboard,paint,physics,pippy,poll,portfolio,read}-activity
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package sugar-{abacus,browse,calculate,chart,chat,clock, \
develop,finance,findwords,fototoon,fractionbounce,gears, \
imageviewer,implode,jukebox,letters,log,maze,measure,memorize, \
moon,music-keyboard,paint,physics,pippy,poll,portfolio,read}-activity
pi@raspberrypi:~$
```

Following were the error logs:

(sugar-activity:15899): Vte-WARNING **: 21:19:12.294: VteSelectionFunc callback ignored.

(sugar-activity:15899): Gtk-CRITICAL **: 21:19:12.514: gtk_widget_get_realized: assertion 'GTK_IS_WIDGET (widget)' failed

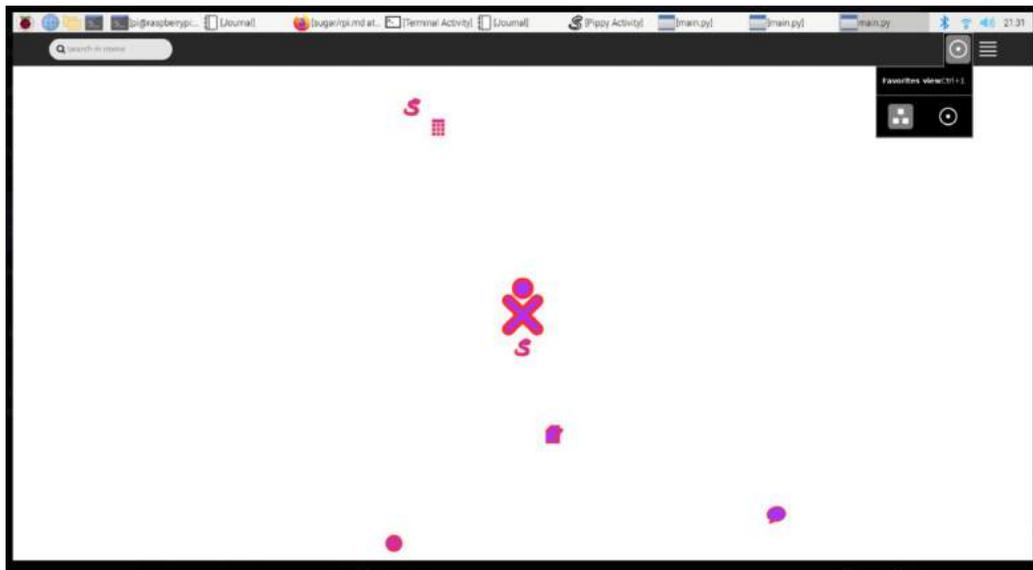
Exited with status 0, pid 15899 data (<open file '<fdopen>', mode 'w' at 0x4c07f440>, 'a0074f05adf84e362341a044ee55667375202efa')

(sugar-activity:25047): Vte-WARNING **: 21:33:05.648: VteSelectionFunc callback ignored.

(sugar-activity:16856): Vte-WARNING **: 21:22:25.051: VteSelectionFunc callback ignored.

Exited with status 0, pid 16856 data (<open file '<fdopen>', mode 'w' at 0x4c07f440>, '77a9e240e7e725b11f893e32ce975c59f6e4347f')

11. The activity icons in this particular view are not properly structured.



Possible reasons of failure and bugs in the Sugar activities:

-When an activity is started by the shell, a timer is set. If the activity doesn't begin before the timer runs out, the shell will display an error message saying "failed to start." However, even if the message appears, there is still a possibility that the slow activity will eventually start.

- The performance of a Raspberry Pi can be affected by the quality of the microSD or SD card being used, which can impact its speed. To measure performance, we can install the "pv" package and use the command ``sudo pv /dev/mmcblk0 > /dev/null``.

-When the activities are first started, GStreamer populates a ``.cache/gstreamer-1.0/registry.armv7l.bin`` file and Python must load all modules, and

calculate import of matplotlib iterates over the available styles. The time taken in performing all this might cause the shell to timeout and report a failed activity.

- Graphics updates can cause slow performance as well. When launching an activity, the pulsing icon in the shell process can compete with the activity starting, and multiple processes running simultaneously can slow down the system.

Work plan to fix activities:

- I will work on fixing the errors during the GSoC period and continue testing for more such bugs that prevent Sugar from running smoothly on the Raspberry Pi.

- In addition to the activities described in this proposal, I will also be testing more essential activities like Turtle Blocks, Measure, etc. by installing them from the command line or downloading them from the Python3 Sugar store: <https://v4.activities.sugarlabs.org>

- To make sure the activities are running on Python version 3 and are updated, I will be installing `sugar-toolkit-gtk3` to ensure they are running smoothly on the platform.

- The file `sugar:src/jarabe/view/launcher.py` sets a default interval of 100 milliseconds for the launcher in Sugar. However, on slower systems, this interval can be increased to 500 milliseconds using gsettings by using the command:

```
...
```

```
gsettings set org.sugarlabs.desktop launcher-interval 1000
```

```
...
```

If this solves the issue, it indicates that the pulsing icon is causing the problem.

- The logs can be checked using `.sugar/default/logs` for previous sessions. For this, `stat` command can be used to compare the activity lifecycle, that is, start, modify and stop. This can further reveal the causes of the failure.

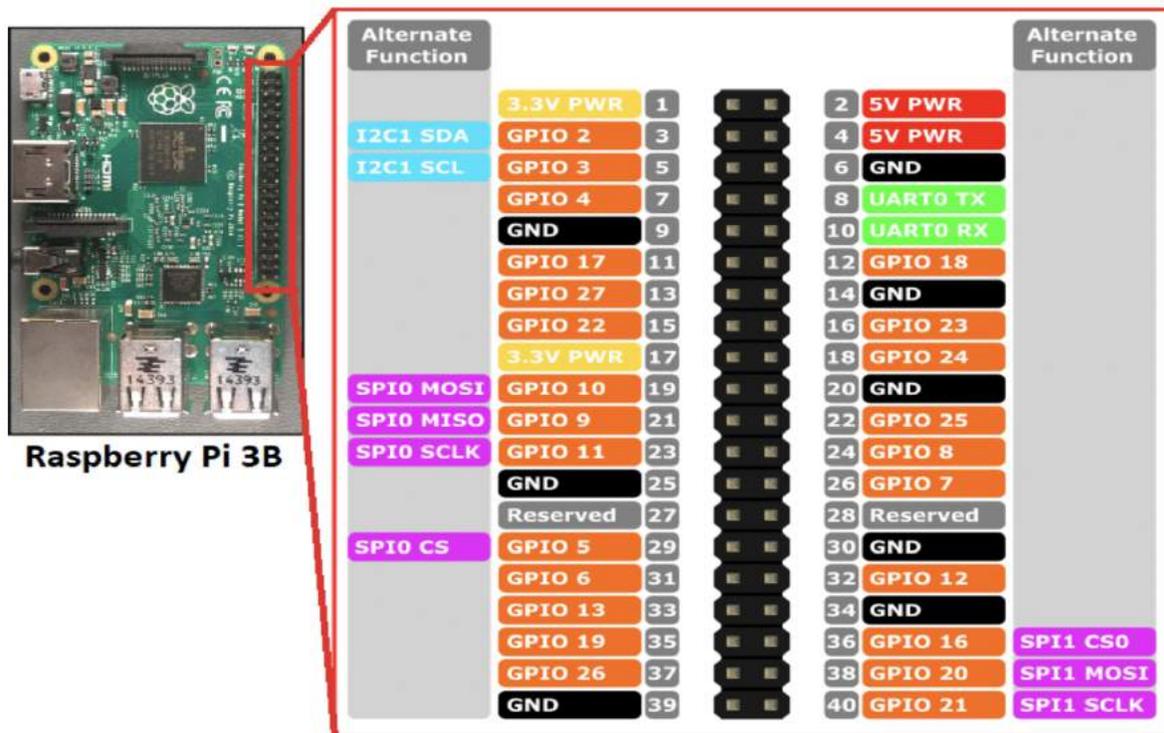
- I will also follow the Sugar Labs maintainer docs on `Modifying Activities` for releasing the fixed activities as shared by Mr. James Cameron here:

<https://github.com/sugarlabs/sugar-docs/blob/master/src/contributing.md#checklist---maintainer>

This is because if we rely on Debian for packaging and releasing activities, it could take a very long time, which would go beyond the GSoC period.

Task 2: Enhancing the functionalities of the current activities such as Turtle Blocks, Measure, and Pippy by using sensors on the Raspberry Pi.

The Raspberry Pi's capabilities can be extended using multiple sensors which can be connected to its pins. The following pin diagram illustrates the built-in features of the RPi which can support a wide range of sensors:



Some of the GPIO (General Purpose Input/Output) pins can be used to connect additional sensors like Camera sensor, Temperature sensor, Light sensor, Motion sensor, Sound sensor, etc.

Extending the functionalities of activities by integrating Raspberry Pi sensors can unleash a whole new dimension to the learning experience for students. Some of the ways through which we can accomplish this for current activities are as follows:

Turtle Blocks:

This is a programming environment in Sugar OS that allows children to create their own interactive art and games using a block-based programming language.

Here are some ways that Turtle Blocks could be used with sensors on Raspberry Pi:

1. **Camera:** The Raspberry Pi Camera module can be used to capture images or video and integrate them into Turtle Blocks projects. For example, students could create a program that captures images from the camera and uses the data to draw a picture or play a game.
2. **Temperature Sensor:** Students could use a temperature sensor connected to the Raspberry Pi to create a program that changes the color of the turtle's shell based on the temperature. As the temperature changes, the color of the turtle's shell would change accordingly.
3. **Sound Sensor:** A sound sensor connected to Raspberry Pi could be used to create an interactive program that responds to noise levels. For example, students could create a program that changes the size or color of a shape on the screen based on the volume of the ambient noise.
4. **Accelerometer:** The Raspberry Pi Sense HAT includes an accelerometer, which can be used to detect changes in motion and orientation. Students could create a program that uses the accelerometer to control the movement of a turtle or other shape on the screen.

Pippy:

This activity can take advantage of the sensors on Raspberry Pi by using Python and the GPIO pins.

1. **Temperature sensor:** A temperature sensor can be used to monitor the temperature of the environment and display it on the Pippy interface. The temperature data can be used to control other aspects of the program, such as changing the color of the interface or playing a sound.

2. **Light sensor:** It can be used to detect the ambient light levels and adjust the program's behavior accordingly. For example, you can create a program that changes the color of the interface based on the light levels or turns on an LED when the light levels are low.
3. **Motion sensor:** Motion sensor can be used to detect movement. For example, you can create a program that plays a sound or displays a message when motion is detected.
4. **Distance sensor:** This sensor can be used to detect the proximity of objects.
5. **Accelerometer:** This can be used to detect the orientation of the Raspberry Pi. For example, you can create a program that changes the color of the interface based on the orientation of the Raspberry Pi.

Measure:

The functionalities of this activity can be extended using the sensors available on Raspberry Pi and it can be a great way to enhance children's learning experience in science and engineering in the following ways:

1. **Temperature sensor:** Temperature sensor can be used to measure the temperature of the environment and display it on the Measure interface.
2. **Light sensor:** This sensor can be used to measure ambient light levels and even collect data that can be used for further analysis. For example, a light sensor can be used to measure the light levels in different areas of a classroom or compare the light levels inside and outside a building.
3. **Sound sensor:** Sound sensor can be used to measure the sound levels in the environment. This can be useful for experiments in acoustics or for monitoring noise levels in a particular area.
4. **Pressure sensor:** This sensor can be used to measure changes in air pressure and record data for analysis. For example, you can use a pressure sensor to measure changes in air pressure as a storm approaches or to measure changes in pressure inside a container.
5. **Gas sensor:** A gas sensor can be used to detect the presence of certain gases in the environment, such as carbon monoxide or methane or any other pollutant. This can be useful for monitoring air quality or for detecting leaks in a gas line.

Chat:

Chat activity allows users to communicate with other users on a network. It can be extended to use sensors such as a microphone or a camera to enable video or voice chat.

Physics:

Physics is a simulation and visualization tool in Sugar that can use various sensors to create physics-based simulations. For example, it can use an accelerometer to simulate the movement of objects in space or a pressure sensor to simulate the effects of air resistance.

Abacus:

The Abacus activity in Sugar is used to help children learn basic arithmetic skills through a virtual abacus. It can also be extended to use Raspberry Pi sensors to create interactive projects that help children learn better.

1. **Temperature-based counting:** The temperature sensor can be used to count the number of times the temperature reaches a certain threshold value. As an example, we can set the temperature to -10 deg celsius and the abacus activity will count each time it hits this defined temperature.
2. **Light-based counting:** In a similar way, we can leverage the light sensor and check every time light is dim or bright.

Similarly we can use Sound and Motion sensors to detect the levels of sound and check when some motion is detected.

I will be discussing more about the functionalities that can be added to the current activities and how many activities need to be focused on with my mentors. My aim would also be to focus on accessibility so that it can reach as many children as possible including the specially abled ones.

What is the timeline for development of your project? The Summer of Code work period is from mid-May to mid-August; tell us what you will be working on each week. (As the summer goes on, you and your mentor will adjust your schedule, but it's good to have a plan at the beginning so you have an idea of where you're headed.)

Following is the timeline that I plan to follow:

Days	Tasks
May 4 th to May 28 th	<p style="text-align: center;">Community Bonding Period</p> <ul style="list-style-type: none"> - Perform testing on more Sugar activities and identify the ones that are broken. - Discuss on ways to fix broken activities with the mentors. - Define goals for the functionalities that need to be extended and what all activities to consider for this.
May 29 th to June 11 th	<ul style="list-style-type: none"> -Assess with mentors whether to use Python 2 or solely Python 3 activities. - Implement the fixes that are discussed with the mentors and get it reviewed for the broken activities.
June 12 th to June 25 th	<ul style="list-style-type: none"> - Discuss the possible activities whose features can be extended using the Raspberry Pi sensors with mentors. - Research on ways to implement the same.
June 26 th to July 9 th	<ul style="list-style-type: none"> - Implement the discussed approach for the activities. - Test it to make sure it is robust and working perfectly.
July 10 th to July 14 th	Midterm Evaluation
July 15 th to July 28 th	<ul style="list-style-type: none"> - Identify other activities whose functionality can be extended. - Work on implementing the support for sensors for those activities.
July 29 th to August 11 th	<ul style="list-style-type: none"> - Implement the support for sensors in those activities.
August 12 th to August 25 th	Wrap up the project and publish blogs on my progress.
August 28 th to September 4 th	Final Evaluation

You and the community

If your project is successfully completed, what will its impact be on the Sugar Labs community?

The entire motive of the "Sugar on Raspberry Pi" project is to offer a more cost-effective alternative to the XO laptop and OLPC's NL3. A single Raspberry Pi unit is priced at around \$40, while a single XO laptop costs around \$200, resulting in a significant price difference between the two options. Even when the cost of keyboards, mouse, and displays that will be connected to the RPi is included, switching to Raspberry Pi will considerably reduce the expenses of the organization using it. Additionally, due to its small form factor, it will provide a more efficient way of using Sugar on displays like a monitor, thereby saving more desk space compared to using a CPU. Being able to run Sugar, which is there to make education available to all children, on a Raspberry Pi is really the best step we can take to provide the privilege of education to everybody.

This will also help Sugar Labs to extend its outreach to the Maker Community and extend the functionalities of various activities as described previously.

What will you do if you get stuck on your project and your mentor isn't around?

In case of a problem, my first approach would be to look for appropriate solutions on the internet. If that does not work, I will seek assistance from other developers on the Matrix channel and post my query on the mailing list. Based on my prior experience, I have observed that the community members are very helpful and responsive, so I am confident that I will receive the necessary support.

How do you propose you will be keeping the community informed of your progress and any problems or questions you might have over the course of the project?

My plan is to create and maintain a progress blog for the project. The blog will contain regular updates on the progress, as well as any obstacles encountered and the methods used to overcome them. To ensure that my mentors are kept up-to-date with my daily progress, I will share links and other relevant updates via email or Matrix IRC when they are online. Additionally, I will update the community on my achievements by sharing milestones through IRC and the mailing list.

Discuss your post GSoC plans. Will you continue contributing to Sugar Labs after GSOC ends ?

I would love to contribute to Sugar Labs even after GSoC ends in any way I could. Sugar Labs is doing some really amazing work out there by doing so much for underprivileged children by making education accessible to them, which is the most important thing that helps a person to grow and experience life ahead. It would be my privilege to be able to contribute to Sugar Labs as much as I could.