Google Summer of Code

2024

sugarlabs

Sugarizer 3D Volume activity

sugarizer

Project Proposal

Harsh Singh

Mentors: Lionel Laské

Table of Contents

1. Personal Information

Basic Information

Full Name: Harsh Singh

Gender: Male

Nationality: Indian | IST (UTC +5:30)

Contact Information

Phone: +91 6386250212

Emergency: +91 7587792076

Email: 22je0388@iitism.ac.in

Social Media

Github: hharshas

Portfolio: https://shorturl.at/yzC49

LinkedIn:harsh-singh-69523a193

Medium:@22je0388

Educational Information

Institute: IIT(ISM) Dhanbad

Degree: B.Tech

Major: Chemical Engineering

Expected Graduation: 2024

I am currently a second year student at Indian Institute of Technology (Indian School of Mines), Dhanbad (India) pursuing a degree of Bachelor of Technology. I am a member of Cyberlabs in the web division and I have been working with **Javascript, React Js, Node Js, Vue Js** for over a year and with **Three Js** for more than six months.

I really love working with Three Js and have been involved in quite a few projects with it. I have a huge respect for open-source culture and I see it as an ultimate opportunity to learn. I believe that learning becomes very easy and effective when we work in a community. Further It's one of my dreams to be a part of open-source projects that your friends and other users will finally use afterwards .i.e you are making an effect with your code in this society. This has always fascinated me to contribute to open-source. My main motive for participating in Google Summer of Code 2024 is to deep dive in this open source world.

2. Relevant Skills

Languages:	JavaScript, C++, Python, C#, GDScript
Library:	React, ThreeJs, Jquery
Framework:	Vue Js
Database:	PostgreSQL, MongoDB
Version Control:	Git

3. Project Information

Organization: Sugarlabs
Project: **Sugarizer 3D Volume activity**

Sugarizer is a free/libre learning platform for children. It is an application including a set of pedagogic activities (more than 50) for children. Each activity proposes a type of work: read, write, explore, create, simulate or play. The objective of this project is to create a new Sugarizer activity to explore volume using dices.

The activity features a dynamic interface allowing users to interact with the **virtual game board** effortlessly. Users can view and **manipulate game volumes** with ease, utilizing intuitive controls such as zooming in/out and rotating the board to get a better perspective. Customization options abound, including the ability to select colors for facets and text, choose from various volume sizes and types, and set the board background to suit their preferences. Additionally, the app supports **multi-user collaboration**, enabling connected users to share activities and collectively engage with the board. Finally, for volumes displaying numbers, **a concise total** is displayed once the shaking action ceases, offering users a clear summary of the current game state.

4. My Objectives

1. Create the 3D scene.
2. Add more core-functionality.
3. It should be Lightweight and compatible with all Sugarizer platforms (textures and 3d models need to be optimized).
4. Convert it into VueJs (if time is left).

5. Abstract

Sugarizer Activities are built using JavaScript, JQuery and the Vue.js framework. This activity needs to be integrated with a 3D JavaScript library. I decided to choose **Vanilla Js** and **Three.js** as it has a large community and a lot of third party plugins. Furthermore, the Planets activity also uses it. I am using **Cannon.js** as the physics library. I will use **Sugar's Presence.js** for the collaborative feature and **Journal** to handle datastore. I am also determined to migrate it later on to **Vue.js** after completing the project (If sufficient time will be left) . During the GSoC period, I would continue with what I have created till now, ensuring a **highly scalable**, **flexible**, and **maintainable code architecture**. Also, if necessary, I will be upgrading the existing Sugar-web Library to make the code most understandable and easy to debug for future collaborators.

6. Architecture Diagram

The user will interact with the 3D-volume activity with the Toolbar Palettes. Any user interaction like clicking a button will trigger an action. These actions update an object named `teal` in the global scope (see milestone and implementation to understand code). I am writing code in this way because declaring the global variable `$t`will make it easier for me later on to convert it into Vue Js.

These changes in `teal` object will cause changes in the user interface and 3D scenes after comparing the differences in the old and new virtual DOM.

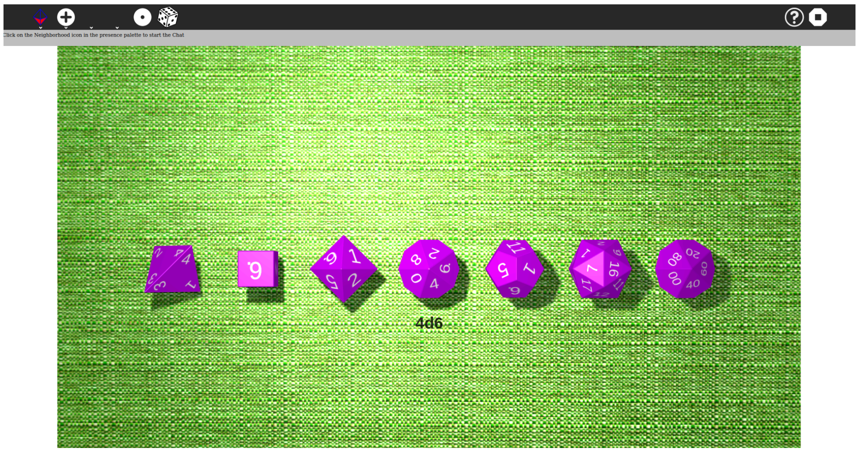
7. Milestones and the implementations

NOTE: This website is only for demo. All the icons and features shown are just samples and subject to modifications as per the suggestion of mentors. I can also create whole new components and their behavior from scratch. This is only for demonstration purposes and an experiment for me to explore the beauty of this GSOC task.

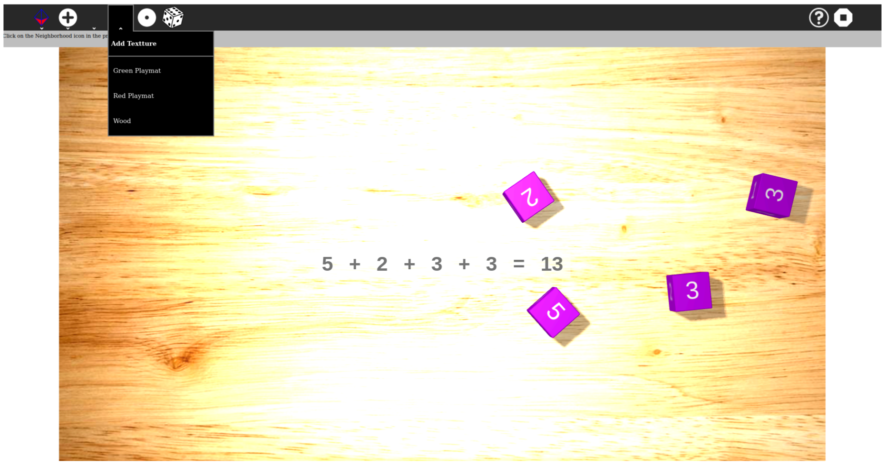
1. **Create the 3D scene.**

   I've created 2 scenes.

   1.1 Home scene where dice are placed up on the board. A simple interface that shows selected dice counts and their volume types.
   Volume types : Tetrahedron, Cube, Octahedron, Dodecahedron, Icosahedron.

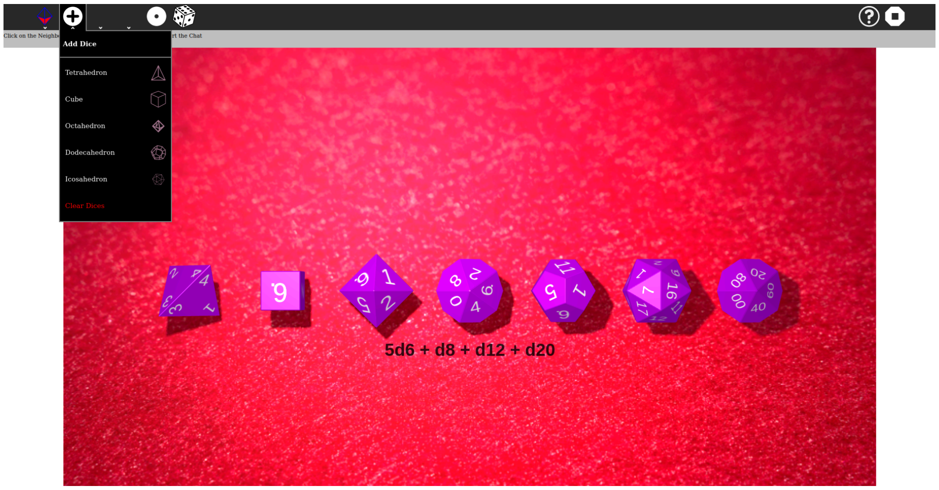

Click on the Neighborhood icon in the presence palette to start the Chat

1,2 Playground Scene where users can throw selected dice in any direction (swipe along the screen). Further an interface for the total score on the center. In descriptions of activity `Shake board` feature was mentioned, I have replaced accelerator with throw dice feature just for demonstration purposes.


Click on the Neighborhood icon in the p

2. **Add more core-functionality.**

   Toolbar Palettes where users can select dice, clear dice, select dice color and select board color. A Toolbar icon to throw dice and a Network icon for live collaboration.



3. **It should be Lightweight and compatible with all Sugarizer platforms.**

   All the Textures are compressed and the 3D models are created out of ` `mesh` ` from scratch rather than being imported from any external source.

4. **Convert it into VueJs (if time is left).**

   I have defined an object ``teal`` and a variable ``$t`` in the global scope so that,

```
window.teal = {};
window.$t = window.teal;
```

It can afterwards be created to Vue.component's data easily.

```
data() {
   return {
      Key : Value,
   }
},
```

How to check the top side ?

I want to share this as I have researched and learned a lot during this demo.

The last thing to do is checking the top side of each dice once the rolling is finished. Aside of adding elements like a button to thrH the dice and place to show the score, we need to:

- capture the moment of stillness for each dice roll
- check the final rotation and get the top side number

Quaternions are a mathematical way to represent rotations in 3D space.

While creating dice I call this method .i.e

```
        dice.body.quaternion.setFromAxisAngle(new CANNON.Vec3(axis.x,  axis.y, axis.z),
axis.a * Math.PI * 2);
```

This code sets the orientation (`quaternion`) of the physics body dice.body by specifying an axis of rotation and an angle of rotation.

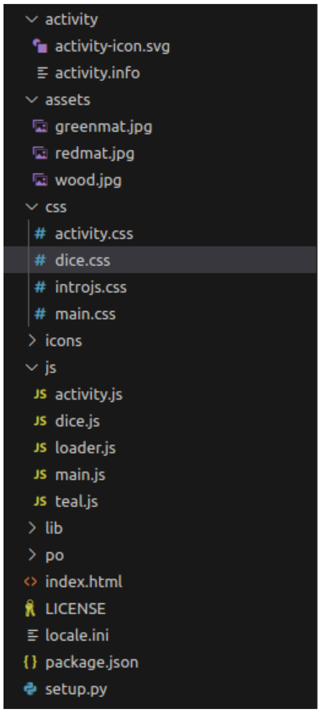
```

for (var i = 0, l = dice.geometry.faces.length; i < l; ++i) {
            var face = dice.geometry.faces[i];
            if (face.materialIndex == 0) continue;
            var angle =
face.normal.clone().applyQuaternion(dice.body.quaternion).angleTo(vector);
            if (angle < closest_angle) {
                closest_angle = angle;
                closest_face = face;
            }
        }
```

And after the stillness for each dice roll, I iterate over faces of the dice geometry and find the smallest angle to the predefined vector (`vector`) i.e. pointing along the z-direction. This face is considered the "closest" face to the vector's direction.

Reference : [codrops](), [aqandrew]()

Current folder structure:-



```
v activity
    activity-icon.svg
    activity.info
v assets
    greenmat.jpg
    redmat.jpg
    wood.jpg
v css
  # activity.css
  # dice.css
  # introjs.css
  # main.css
> icons
v js
  JS activity.js
  JS dice.js
  JS loader.js
  JS main.js
  JS teal.js
> lib
> po
<> index.html
  LICENSE
  locale.ini
{} package.json
  setup.py
```

Current progress you can see :-

1. Hosted website: [gsoc-sugar.vercel.app]
2. Codebase : [github].

Further I am aware that there are many things that are left and need to be done according to [Sugarizer 3d-volume-activity] . I am in constant communication with mentor [Lionel Laské] and will continue to work so that we can meet the desired requirements for this project.

The list of project's must ToDos that are in my mind for now are illustrated in the table below.

ToDos	Description
Complete understanding of the Task and refinement with mentors	Interacting with the mentor to grab every minute details and finalize the work structure
Click to delete the Dices	I have made a Palette to click the entire set of selected dice. I will add features to click and delete the dice one by one.
Handle multi-user with presence	Every connected user can interact with the board.
Localize the activity	Sugarizer will detect the language of your browser and use this language for the UI and the activities.
Friction dependencies to the background texture of the board.	We can create our own friction with conditions using Raycaster from Cannon.js library.
Refine and improve overall with mentor feedback + additional tasks	After completing each tasks, I will take feedback from mentors and improve accordingly
Integrate a tutorial	I will use IntroJs tour component for creating step-by-step users onboarding tours.

8. Timeline

- This timeline is flexible and can be changed according to mentors' suggestions

Time Frame	Milestones
Community Bonding **(**1 May - 26 May**)**	● Communicate with mentors and know the project at its core ● Realize any unambiguous part of proposal and fix it ● Check for any further dependency needs and explore them ● Discuss with mentors the rough flow of project ● Update the readme and set-up instructions to help fellow present/future contributors
First Week (27 May - 2 June)	● Finalize the workflow and timeline with mentors ● Get started with the implementation of project ● Start working on improvements of current demo
Second Week (3 June - 9 June)	● Add new features like zoom, rotate, colorpicker, volume type and volume size. ● Complete all the works related to 3D scenes in the activity ● Start developing UI Designs
Third Week (10 June - 16 June)	● Coding the UI designs and integrating it with 3D scenes ● Discuss with mentors and implement suggested features
Fourth Week (17 June - 23 June)	● Using Presence Js for users to share activity. ● Using Canon Js to include accelerator (if necessary)
Fifth Week (24 July - 30 June)	● Improve Physics in activity so that it doesn't show any minor bugs and improve functionalities in the activity to make 3D scenes fast and compatible.
Sixth Week (1 July - 7 July)	● Add tutorial using IntroJs ● Handle journal and datastore (if required in some way)

FIRST EVALUATION (12 July - 16 July)	**Mentors and Students submit their evaluation of one another**
Seventh Week (13 July - 19 July)	● Work on the changes that mentor mentioned in the evaluation
Eighth Week (20 July - 26 July)	● Localizing the activity ● Will take help of the community to get all language translations. ● Start working on migration to VueJs (if activity seems to be in the verge of completion)
Ninth Week (27 July - 2 Aug)	● Will take review about design types of dice and the board. ● Production build and deployment setting changes to the activity (If needed) ● Continue to migrate to Vue Js
Tenth Week (3 Aug - 9 Aug)	● Finalizing the project and buffer time to fix any issues left ● Refine and improve features with mentors' feedback ● Develop additional features suggested ● By then, try to migrate the entire 3d scenes and physics to the VueJs.
Eleventh Week (10 Aug - 18 Aug)	● Constantly communicate to the mentors to refine the features. ● Wrapping up the VueJS migrations to a satisfactory end result.
FINAL WEEK(19 Aug - 26 Aug)	**Students submit their code and project summaries**
FINAL EVALUATION(26 Aug - 2 Sep)	**Code reviewed by the mentors**

I will keep on updating my progress weekly to my mentors and incorporate their feedback and suggestions as and when required. I am confident enough to complete the project within the given time frame. I have also kept a buffer week before final evaluations for any critical situations.

9. Questions

NOTE: These questions are taken from the Sugar Labs GSOC's "Template.md".

9.1. Basic Details

- **Full Name**

  Harsh Singh

- **Email and GitHub Username**

  22je0388@iitism.ac.in

- **Your first language**

  Hindi, also i am lot fluent with English

- **Location and Timezone**

  Varanasi, UttarPradesh. UTC+5:30

- **Share links, if any, of your previous work on open source projects -**

  I have explored the Palisadoes Foundation, Fossology and Scorelabs earlier but I haven't contributed to any. Most of them were just for me to learn before coming to this project. I learned `Jest` for testing from Palisadoes, `Rest API` management from Fossology and `React` in ScoreLabs. Here is my forked open source repository in which I was invested earlier.

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

  As I have mentioned earlier, Contributing to open-source is somewhat of a dream, I mean the pride in the feeling that my code will cause an impact in the lives of millions of people who will use it is unparalleled. I was watching Sugar Labs repositories earlier and it gave me a vibe of `Edubuntu` "An ubuntu Education edition for school labs / programs". All of my school's computers had Edubuntu installed. As a kid, I used to spend a lot of time with Potato guy, Tux paints, Tux Math and so on. Contributing to SugarLabs feels like contributing to my childhood self, furthermore Sugarizer launched a 3D project this year that I love doing. I have learned Three js, Theater Js and React Three Fiber earlier. Here are some examples of Three js works Portfolio and A 3D product demo. This project affords me the opportunity to apply my acquired knowledge. Sugar-web library also fascinates me and i want to know a lot about insight details of presence js or journal. I have speciality in MERN stack, Vue Js, a bit of Next Js. I want to learn more about Cordova. I think these curiosities and crave for knowledge had already made me a fit candidate for this project.

For my earlier contributions to SugaLabs, I am dedicating a specific coloumn below. You can go through it.

9.1. Project Details

- **What are you making ?**

  The objective of this project is to create a new Sugarizer activity to explore volume using dices. Users can interact with volumes of different shapes and types. This activity aims toward the learning of 3D geometries along with dice simulation. Users can online collaborate and insert Dices of different shapes to the board. Users can see the sum of top of all the dice as a result after the accelerator stops. They have the ability to rotate and zoom for the better view preference and can select the facet color along with the board background.

- **How will it impact Sugar Labs ?**

  Sugar Labs activities cover a wide range of middle school syllabus and hence it provides a good foundation towards basic concepts like for

  Math : - Sprint math, calculate and abacus etc.

  Physics : - Physics Js, Planet etc

  Programing : - Scratch etc

  Sugar Labs have less repositories around Three Dimensional portrays. Students need to visualize planes, volumes and axes before running into math topics like coordinate geometry. Further the collaboration feature will make it a fun place for virtual dice rolling simulation.

- **What technologies (programming languages, etc. will you be using ?**

  I will be using Three Js library for the 3d scenes as it has more plugins, a wider community and further I also want to contribute there in future so better using it now. For physics I will be using the Cannon JS library as Three Js and Cannon Js go well together and also I have some previous work experience with it. Obviously the programming language will be Vanilla Js but I am planning to convert this entire activity to Vue Js later on. I may use Canvas Js later on for better textures but it doesn’t seem necessary to me now.

10. Other Deliverables

Contribution to SugarLabs : Apart from the deliverables mentioned above, I would also get more involved in the Sugar Labs ecosystem by contributing to it and helping other contributors.

Non-coding tasks : There are several miscellaneous non-code tasks that I would like to take up in favor of giving back to the community such as mentoring.

Adding more new technologies, improving the Sugar-web library with newer innovations.

11. Future Scope

I wish to remain as an active maintainer of the Sugarizer repository and other Sugar Labs repositories. I will also address all the future bugs encountered and feature additions.

12. Previous Contributions

A complete list of PRs can be viewed Pull Requests, Issues

Sugarizer dev : Blockrain Improvement

Sugarizer feature / v2 : #1561 (Todo task given by mentor)

Till now 2 merged commits and 2 open commits…

I have also completed the **Pawn Activity Tutorial** of the Sugarizer activity. Here is the Github Repository.

13. Experience

- Technical Organizer of Srijan: I was a part of the technical team who made the frontend and backend of the cultural fest website of our college. A vite.js website in React where we implemented Razorpay payment and started the use of forms along with a backend running on Express.js. The styling was done through MUI and Tailwind CSS along with Framer Motion for animations.

Tech Stack: React, Vite along with Tailwind and MUI in the frontend

Website: Srijan '24
Repository: Srijan_Frontend

14. Time Commitments

I have my summer break from 6 May to 18 July 2024, so I don't have any extra burden of any college exams. I will be able to dedicate 35-40 hours of weekly effort to this project. Even if something comes up in between, then I will make sure to keep the mentors updated and try to complete the work in advance.

Thank You