GSoC'24 Project Proposal
Sugar Labs: Sugarizer 3D Volume activity

Korada Vishal

Basic Details

Full Name:

Korada Vishal

Email and GitHub Username:

Email: korada.vishal.phe22@itbhu.ac.in
GitHub Username: @Vishalk91-4

Your First Language:

My first language is Hindi but I am proficient in English and I am learning French.

Location and Timezone:

Location: Varanasi, Uttar Pradesh, India
Timezone: Indian Standard Time (UTC + 5:30)

Education Details

I cleared the toughest undergraduate examination in India, and the second toughest exam in the world IITJEE and got admission in Indian Institute of Technology (BHU) Varanasi. I started programming in my Freshman year of college. I have got knowledge over a lot of Computer Science topics through the Institute courses of my department like Computer Architecture,

Data Structures, Operating System and Web Development. I have also been contributing to Open Source projects since the latter half of my Freshman year. I am recently learning Gen AI and trying to build some projects in it.

Share links, if any, of previous work on Open Source Projects

I got into GSSoC'23, an Open Source Program and started my journey there. Since then I have evolved as a developer by having a **MEVN** stack and knowing about Cloud Technologies.

I have been contributing to **Sugar Labs** for the past **7 months.**
In this period, I have contributed to various repositories in Sugarizer main project from sugarizer, to sugarizer-doc and sugarizer-server
Contributing from UI Changes, bug fixes, typo fixes, documentation, enhancements and Migrating activities from webL10n to i18next
These are my **contributions** to **Sugar Labs** in Sugarizer

In **Sugarizer Repository**

Pull Request Link	Description	Status
#1360	Fixed the UI of Next and Done Button	Merged
#1522	Fixed Resize issue in description of activities	Merged
#1383	Migrated Abacaderium from webL10n to i18next	Merged
#1385	Migrated Blockrain from webL10n to i18next	Merged
#1386	Made Fototoon Activity Buttons interactive	Merged

#1391	Migrated Chat Activity to i18next	Merged
#1394	Changing the position and color of spinner in QR Activity	Merged
#1400	Replaced webL10n to i18next in Constellation	Merged
#1401	Replaced webL10n to i18next in Fototoon	Merged
#1405	Replaced webL10n to i18next in Clock	Merged
#1408	Replaced webL10n to i18next in Falabracam	Merged
#1423	Migrated Speak Activity to i18next	Merged
#1450	Migrated Sprint Math Activity to i18next	Merged
#1456	Made Grammatical Changes to README file	Merged
#1465	Migrated Speak Activity to i18next	Merged
#1467	Fixed Typos in CODE_OF_CONDUCT .md and credits.md file	Merged
#1473	Migrated GridPaint Activity to i18next	Merged
#1477	Migrated TankOp Activity to i18next	Merged
#1495	Migrated Record	Merged

	Activity to i18next	
#1502	Migrated GetThingsDone to i18next	Merged
#1519	Ctrl + a/A, ctrl + backspace functionalities in v2	Merged
#1522	Using Same button for Next and Done in v2	Merged
#1548	Fixed selection of popup-title in language settings	Merged

In **Sugarizer-doc Repository**

Pull Request Link	Description	Status
#37	Made Language Dropdown visible in Sugarizer website	Merged
#39	In Explore activities, no. of page selection possible now	Merged
#44	Sort By Alphabetical order, in Mobile version present now	Merged
#53	Copyright automation every year	Merged
#55	Updated All Logos in	Merged

	Sugarizer webpage	
#45	Added Carousel for displaying blogs	Open

I have made a total of 25 issues in Sugarizer. All of them can be found here link

Out of the above contributions, I have made significant contributions in porting 14 activities to i18next and now I am completing the TODO for v2 of sugarizer

I have worked on my personal projects for learning purposes, creating a Linux Foundation Mentorship website for easy selection of Organizations for contributors Backend Repo Frontend Repo website link
Also, I have participated in Hacktoberfest '23 and Advent of Code '23, completing both successfully. Following are some of my open-source contributions to different organizations.

- https://gitlab.com/gitlab-org/gitlab/-/merge_requests/139437
- https://github.com/illacloud/illa-builder/pull/2228
- https://github.com/ucsc-ospo/ucsc-ospo.github.io
- https://github.com/COPS-IITBHU/cops-website

All My other contributions can be found here on my GitHub page link

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

I have been an active contributor of **Sugar Labs** since **Aug 2023**, with 25+ PRs over various repositories. With this much time spent over the codebase, I have got a nice understanding of it.

- Pull Request: (28 closed, 1 Open)
- Issues: (18 closed, 7 Open: 2 to be released)

I am willing to be a part of Sugar Labs for Google Summer of Code 2024, because I feel this matches very much to my Technical Stack with it's framework VueJS, and the motive of the organization, the connections which I believe, I can make with such a large Organization with many Apps and Activities being run everyday by thousands, the effect a change in my code could bring is what drives me to work in this organization, sticking to one only.

Prerequisites for Sugarizer VueJS Application

I possess experience in Javascript, HTML5 and VueJS framework. And, I am familiar with the codebase of Sugarizer and have fixed issues for the following in the related repositories.

I have been a MEVN developer, and have already made projects with JavaScript, HTML5 and VueJS framework. Here are some of my work

- **COPS SDG Site**: link
  I am a core team member in Club Of ProgrammerS (COPS) IIT BHU, and have worked in their SDG Site, for software development group I have enhanced the UI of the site, made it more accessible, and made the frontend components in it.
  Technologies Used: VueJS framework, JavaScript, SCSS

- **Portfolio Website**: github
  I built a 3D Portfolio website for myself which has my experiences, about myself, the technologies I know made with ThreeJS, used framer motion, react tilt for cards section, canvas and 3d objects camera and all those effects creations.
  Technologies Used: ThreeJS, JavaScript, CSS, Framer Motion

These are some of the projects, which I have listed, I have some more and over various TechStacks like Golang, did my Advent of Code '24 in Python, Golang, you can find more on my GitHub page link

I have previously contributed to a lot of Sugarizer Activities, and worked around the sugarizer website and it's server

Links to my contributions are listed here, link and in the above section.

To mention some of them :-
Fototoon
https://github.com/llaske/sugarizer/pull/1386
QR
https://github.com/llaske/sugarizer/pull/1394
FoodChain
https://github.com/llaske/sugarizer/pull/1406

I completed both the Sugarizer Vanilla Javascript activity development tutorial and the Sugarizer Vue.js activity development tutorial.

Here is the link to my repository containing those tutorials.

- https://github.com/Vishalk91-4/sugarizer/tree/pawn-issue/activities/Pawn.activity
- https://github.com/Vishalk91-4/sugarizer/tree/pawn-vanilla/activities/Pawn.activity

Video of those tutorials ➖

`Vue JS` ➖
https://discord.com/channels/1078051575580336249/1078054265517506681/1201947911022460928
`Vanilla JS` ➖

https://discord.com/channels/1078051575580336249/1078054265517506681/1217162361069244426

The Making of above tutorials have helped me a lot in understanding the making of new activities, their procedure the components to add, what additional functionality could be implemented and how would such things affect the rest of the codebase

I have also `tested sugarizer`, for `v2`, both using `file://` and also using `docker` to host it locally at `localhost:8080`

Video Link to file:// ➖
https://github.com/Vishalk91-4/sugarizer-links/blob/main/sugarizer-server.webm
Video Link to localhost:8080 ➖
https://github.com/Vishalk91-4/sugarizer-links/blob/main/file_sugarizerserver.webm

Github link to current Activity of Volume3D under development ➖
https://github.com/Vishalk91-4/Volume3D-Activity

Project Details

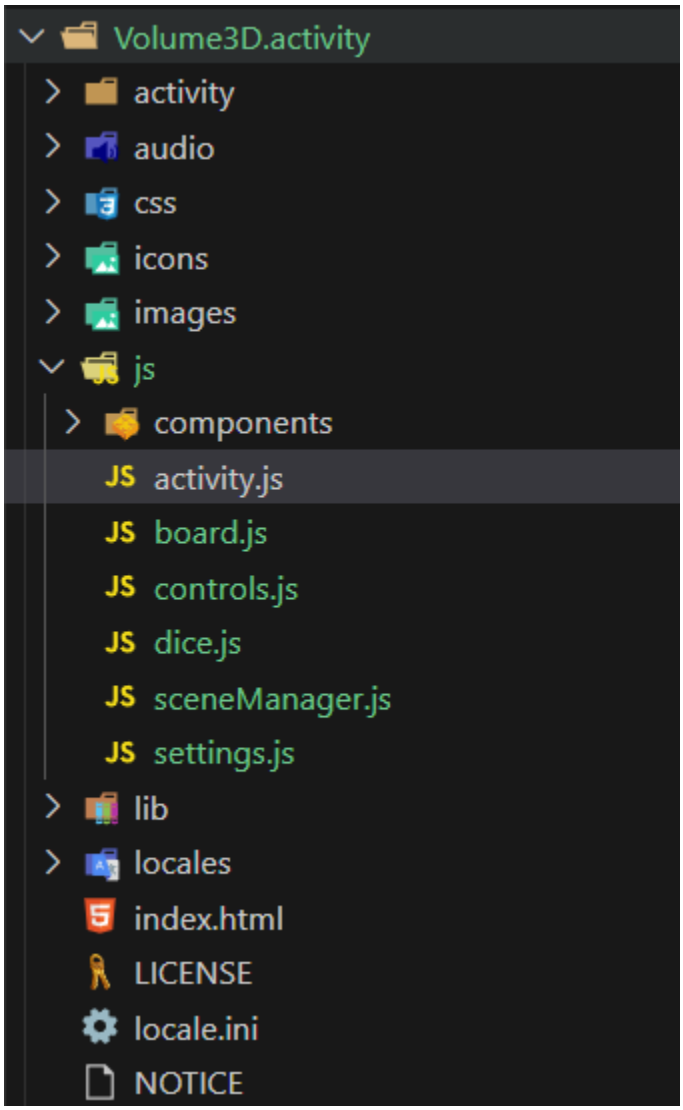
What are you making ?

The aim of this project is to create a new Sugarizer activity to explore volume using dices.

- Having a button with controller type thing where users can change the volume of dices, add or remove volumes and add like which dices they want
- Adding Zoom in / out features
- Adding a board to bound the area of dice rolling, which could be rotated and shaked
- Adding color to dice as buddy colors and to the background like neutral (default), green playmat, red playmat, wood

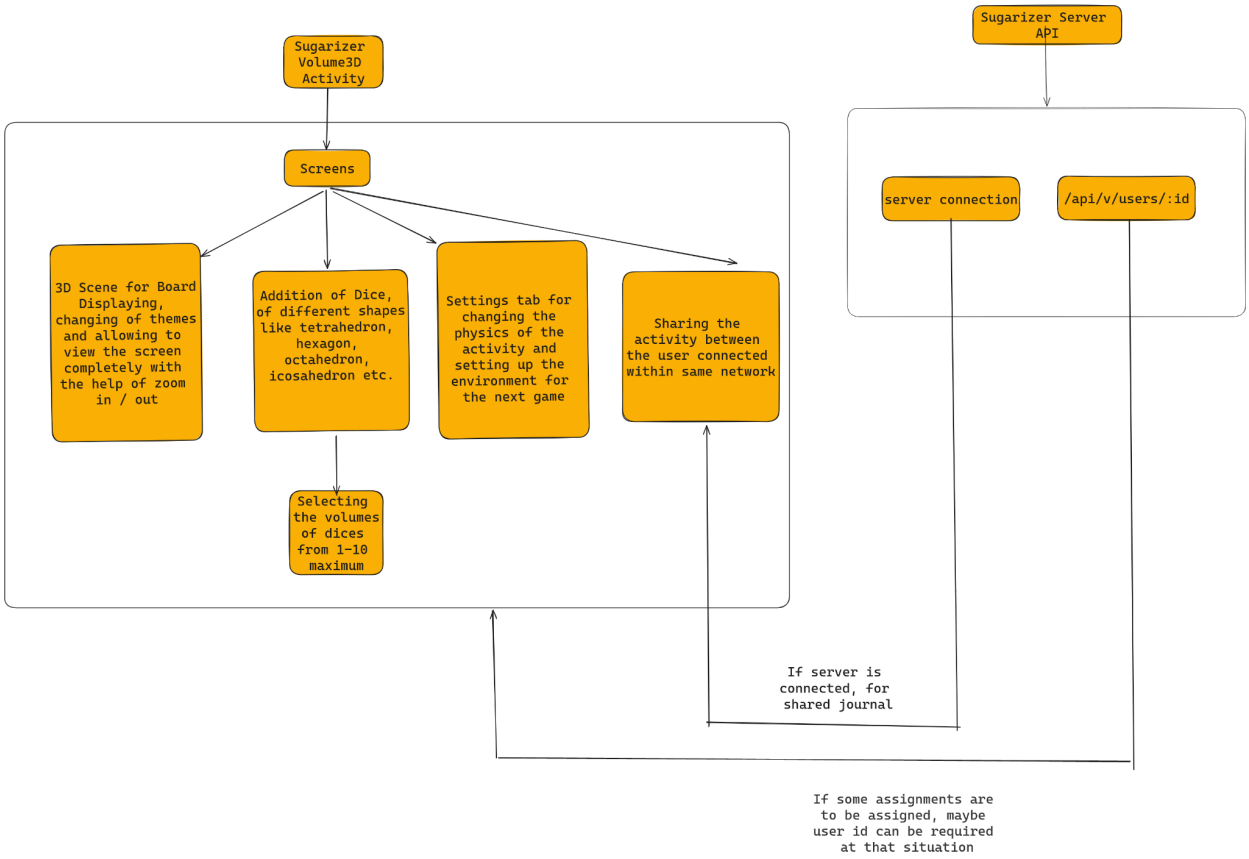
- Adding button where Volume of Dices i.e 4, 6, 8, 10, 12, 20 can be selected and their types like semi-transparent, without number (default), with number can be selected
- Adding a custom setting button where different parameters based on the physics aspect of the board, the pace at which dice rotate and all of those can be controlled

I will be creating a new directory named `3dVolume` in the activities directory of the sugarizer app and there we would be adding rest of our files



The interconnection can be summarized in the image given below

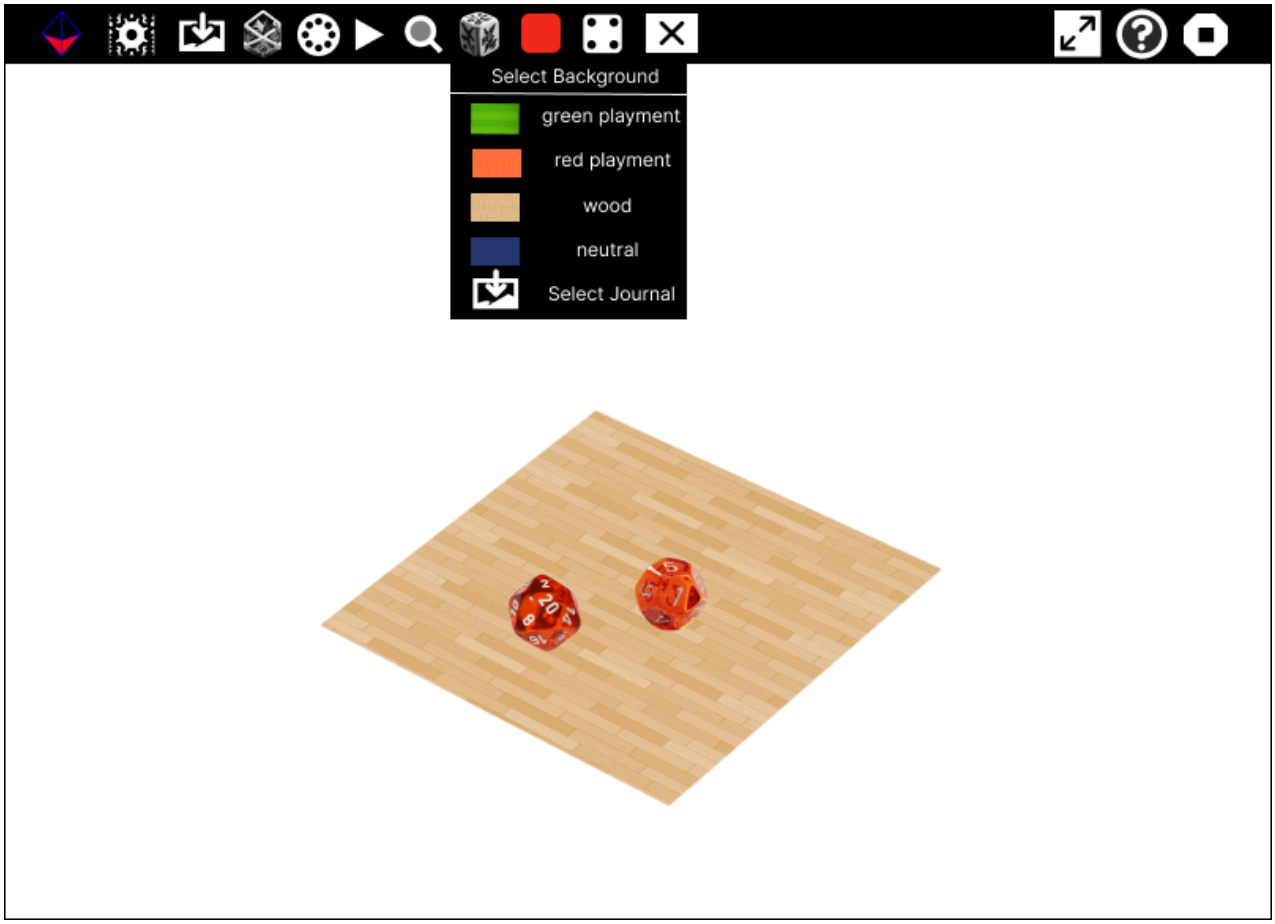
[Sugarizer 3D Volume Project Architecture Link]



Details of how the changes will be written are described below:

❖ Displaying the Board

➢ Displaying the `3D Board`, we would make the button to select background which would have 3 major backgrounds like `neutral (default), green playmat, red playmat, wood and one for selecting from journal` there would be friction depend of the different background



➢ Adding a data function where we add `backgroundOptions` which contain different types of background like wood, neutral, greenPlaymat, redPlaymat and Selecting from Journal
And method `changeBackground` to change the background to whatever the user selects

```
data() {
 return {
    backgroundOptions: [
      { type: 'wood', url: '../images/wood.jpg' },
      { type: 'neutral', url: '../images/neutral.jpg' },
      { type: 'greenPlaymat', url: '../images/greenPlaymat.jpg' },
      { type: 'redPlaymat', url: '../images/redPlaymat.jpg' },
      { type: 'journal', url: '' }
    ],
    currentBackground: 'neutral'
 };
}
```

```
methods: {
 changeBackground(option) {
    const loader = new THREE.TextureLoader();
    loader.load(option.url, texture => {
      this.board.material.map = texture;
      this.board.material.needsUpdate = true;
    });
 }
}
```

➢ Adding a button with which we could `rotate` the complete 3D Scene by 180º, and select the direction whether `left, right, upwards or downwards` and the complete `scene shakes` and `new results` are obtained

```
document.getElementById('rotateButton').addEventListener(
'click', function() {
    rotateScene('right');
    shakeScene();
});
```

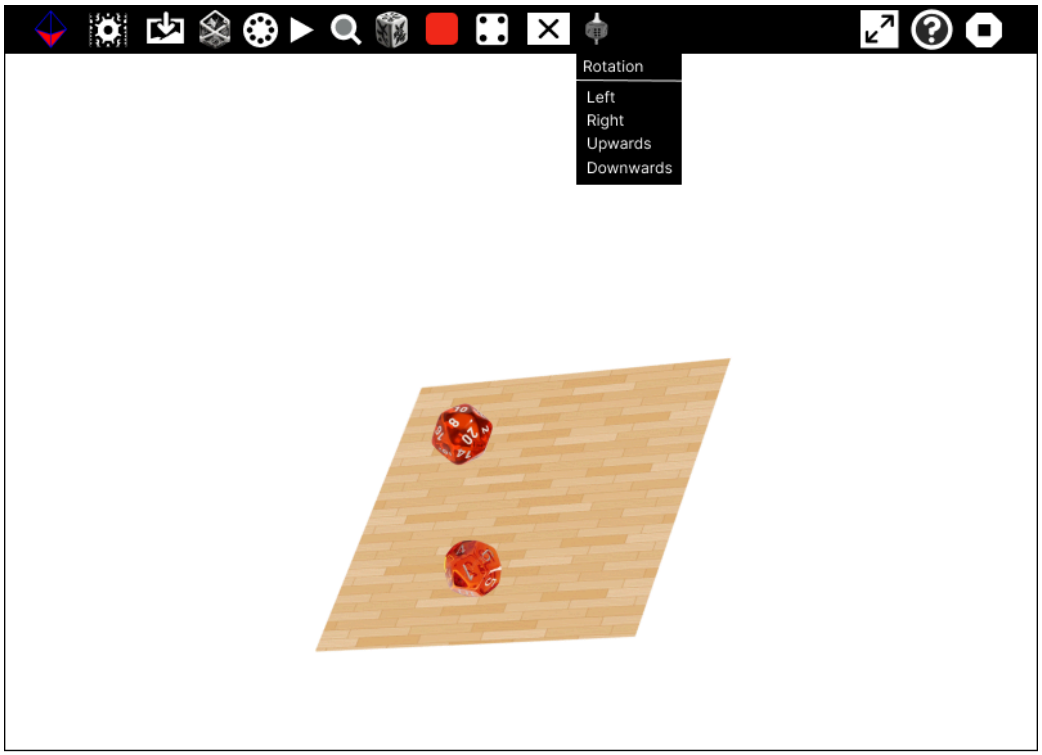
Pseudo Code for Rotation of scene

```
function rotateScene(direction) {
    const rotationAmount = THREE.MathUtils.degToRad(180);
    switch (direction) {
        case 'left':
            scene.rotation.y += rotationAmount;
            break;
        case 'right':
            scene.rotation.y -= rotationAmount;
            break;
        case 'up':
            scene.rotation.x += rotationAmount;
            break;
        case 'down':
            scene.rotation.x -= rotationAmount;
            break;
    }
}
```

Pseudo Code for Shaking it after rotation

```
function shakeScene() {
    const shakeDuration = 100;
    let shakeFrames = 0;
    function shake() {
        if (shakeFrames > 0) {
            scene.rotation.x += (Math.random() - 0.5) * 0.1;
            scene.rotation.y += (Math.random() - 0.5) * 0.1;
            scene.rotation.z += (Math.random() - 0.5) * 0.1;
            shakeFrames--;
            requestAnimationFrame(shake);
        }
    }
    shake();
}
```

UI for Rotation part



❖ Addition of volumes to the Board

➢ Creating a file named Dice.js where we would add all the dices
Export it to the main activity.js which we contain all things together
Here, we are rendering the 3D Scene, listing all the shapes and then animating the same and adding it

➢ First here we are creating a div where we are adding the `3DScene` ,
Then creating a scene in which we are having different shapes like 4, 6, 8, 10, 12, 20 die

```
mounted() {
    const scene = new THREE.Scene();
    const camera = new THREE.PerspectiveCamera(75,
window.innerWidth / window.innerHeight, 0.1, 1000);
    const renderer = new THREE.WebGLRenderer();
    renderer.setSize(window.innerWidth,
window.innerHeight);
    this.$refs.scene.appendChild(renderer.domElement);

    let geometry;
    switch (this.diceType) {
      case 'tetrahedron':
        geometry = new THREE.TetrahedronGeometry(1);
        break;
      case 'octahedron':
        geometry = new THREE.OctahedronGeometry(1);
        break;
    }
```

❖ **Selecting Volumes**

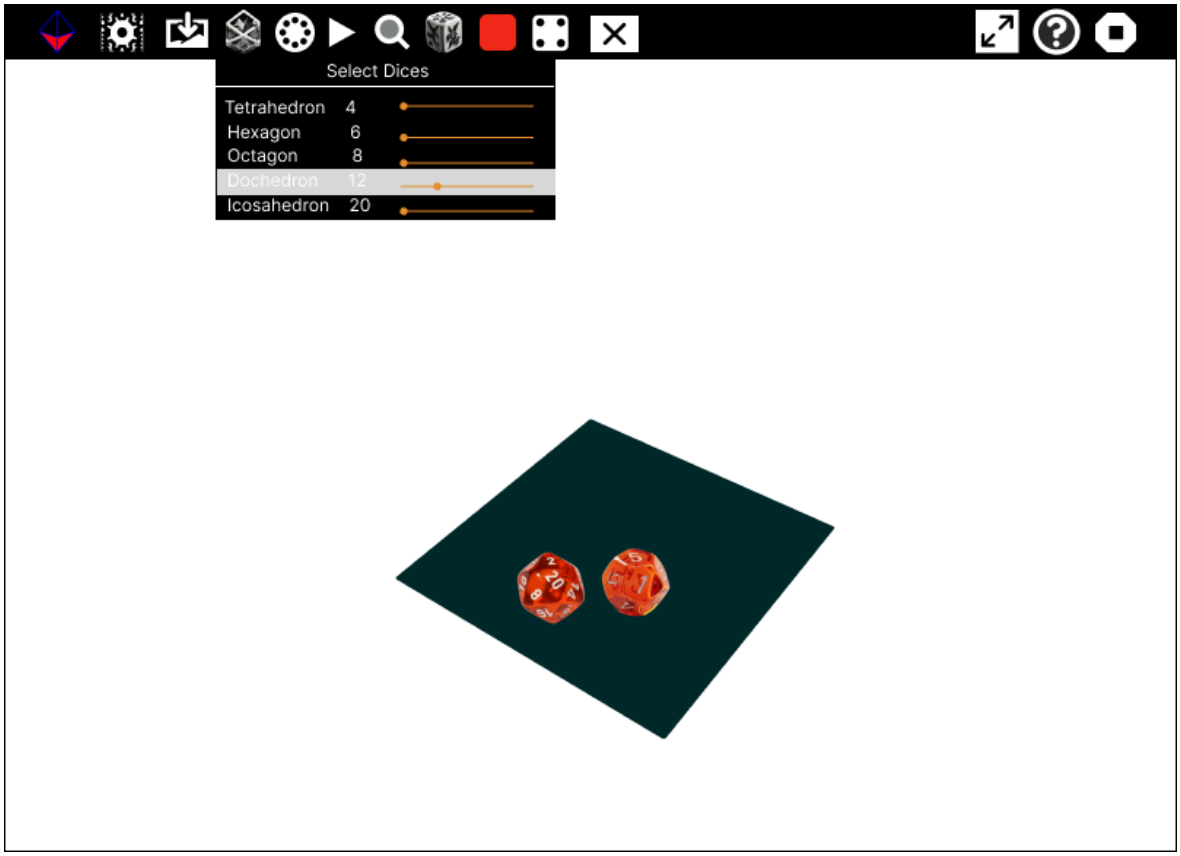
➢ Creating a file like `DiceSelector.js` where there would be a template inside it we would have a div `scene` where we could add a button to add different volumes from selecting different die of different types

```
<template>
 <div>
    <div v-for="(dice, index) in diceTypes" :key="index">
      <input type="checkbox" :id="'dice' + index" v-
model="selectedDice[dice.type]" />
      <label :for="'dice' + index">{{ dice.name }}
</label>
    </div>
    <label>
      Quantity:
      <input type="number" v-model="quantity" min="1"
max="10" />
    </label>
    <button @click="addDice">Add Dice</button>
 </div>
</template>
```

➢ Adding this to the scene, to get dice, which we can select and add more volumes

```
methods: {
    addDice() {
      this.diceTypes.forEach((dice) => {
        if (this.selectedDice[dice.type]) {
          for (let i = 0; i < dice.volume; i++) {
            this.addDiceToScene(dice.type);
          }
        }
      });
    },
    addDiceToScene(diceType) {
      console.log(`Adding ${diceType}`);
    },
  },
  mounted() {
    this.diceTypes.forEach((dice) => {
      this.$set(this.selectedDice, dice.type, false);
    });
  },
};
```

```
addDiceToScene(diceType) {
 const geometry = new THREE.TetrahedronGeometry(1);
 const material = new THREE.MeshBasicMaterial({ color:
0x00ff00 });
 const dice = new THREE.Mesh(geometry, material);
 dice.position.set(Math.random() * 10 - 5, Math.random()
* 10 - 5, Math.random() * 10 - 5);
 this.scene.add(dice);
}
```



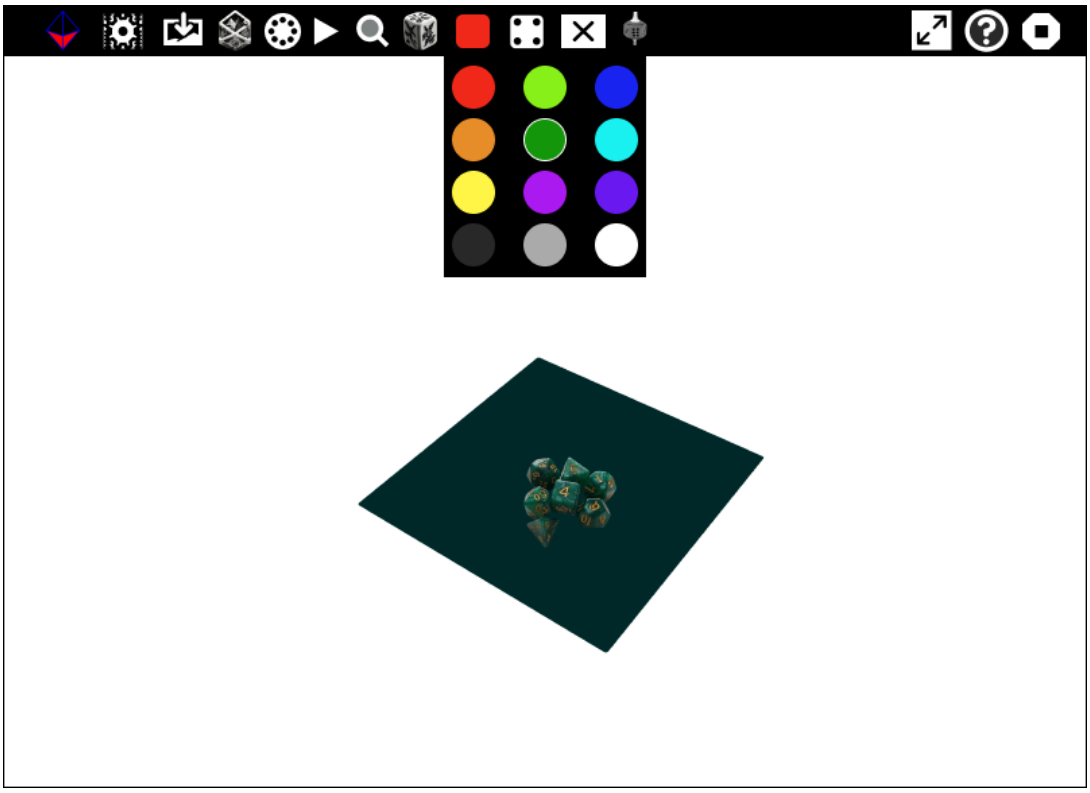
❖ **Adding color to the die**

➢ Colors from the palette presented,
Default is going to be the user's color of the activity

```
const buddyColor = new
THREE.Color(sugarizer.environment.buddyColor);

const diceTypes = [
 { type: 'tetrahedron', name: 'Tetrahedron', color:
buddyColor },
 { type: 'octahedron', name: 'Octahedron', color:
buddyColor },
 // Listing rest of the shapes below here
];
```

```
 const material = new THREE.MeshBasicMaterial({ color:
diceType.color });
 const dice = new THREE.Mesh(geometry, material);
 // Position the dice randomly within the scene
 dice.position.set(Math.random() * 10 - 5, Math.random()
* 10 - 5, Math.random() * 10 - 5);
 scene.add(dice);
```

UI of the Color palette for selection of die color



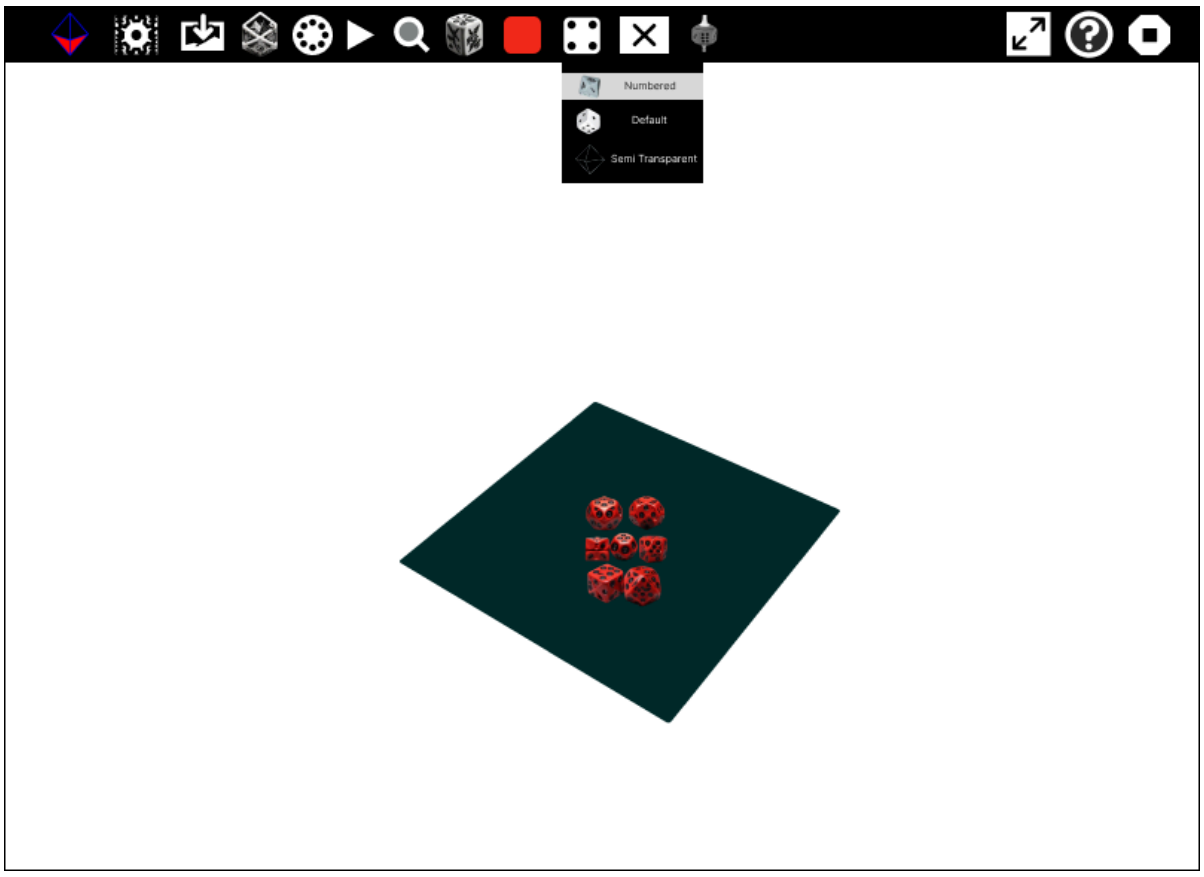
❖ **<u>Adding Structure to the die</u>**

➢ Adding structure like numbered form of die, or with dots or transparent structures of the die

➢ Adding methods for transparent and Number to dice and the other one for dots to the dice
A function for getting face method like which one for at each face for the dice and dots for the same

```
methods: {
   addNumberedDice(diceType) {
       const geometry = this.createDiceGeometry(diceType);
       const material = new THREE.MeshBasicMaterial({ color: 0x000000 });
       const dice = new THREE.Mesh(geometry, material);
       this.scene.add(dice);

       const numbers = this.getNumbersForDiceType(diceType);
       numbers.forEach((number, index) => {
         const textGeometry = new THREE.TextGeometry(number, {
           font: font,
           size: 1,
           height: 0.1,
         });
         const textMesh = new THREE.Mesh(textGeometry, material);
         textMesh.position.set(...this.getFacePosition(index, diceType));
         dice.add(textMesh);
       });
     });
   },
}
```

UI of the dot structure of die



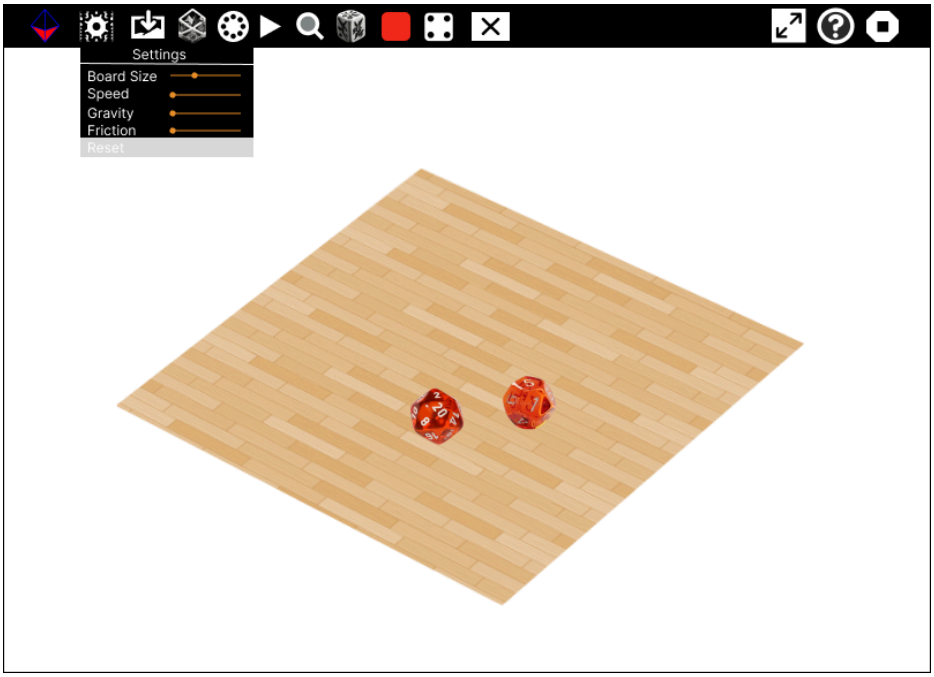
❖ **Settings Button to customize the board game**

- ➢ Adding a settings button with the help of which the user's can customize the board and the dice
- ➢ Adding a method known as applySettings inside of which there would be logic for addition of gravity, friction, speed at which the dice rotates
- ➢ Then integrating these in activity.js where these would be added as a button and the user can change the settings and reset to a default setting which would be present

```
import * as THREE from 'three';
import * as CANNON from 'cannon-es';

export default {
 data() {
    return {
      scene: null,
      camera: null,
      renderer: null,
      world: null,
      gravity: 0.1,
      friction: 0.1,
      speed: 1,
      boardSize: 10,
      showSettings: false,
      dice: [],
    };
 },
```

```
methods: {
   openSettings() {
     this.showSettings = !this.showSettings;
   },
   applySettings() {
     this.camera.position.z = this.boardSize;

     this.world.gravity.set(0, -this.gravity * 9.82, 0);

     this.dice.forEach(dice => {
       dice.rotation.x += this.speed * 0.01;
       dice.rotation.y += this.speed * 0.01;
     });
   },
},
```

```
 mounted() {
    this.scene = new THREE.Scene();
    this.camera = new THREE.PerspectiveCamera(75, window.innerWidth / window.innerHeight,
0.1, 1000);
    this.renderer = new THREE.WebGLRenderer();
    this.renderer.setSize(window.innerWidth, window.innerHeight);
    document.body.appendChild(this.renderer.domElement);

    this.camera.position.z = 5;

    this.world = new CANNON.World();
    this.world.gravity.set(0, -9.82, 0);

    this.animate();
 },
```



❖ Zoom in / out || Touch to move through the 3D Scene

Creating a file Control.js which is where we would add the zoom in/out functionality and then later integrate it in activity.js file

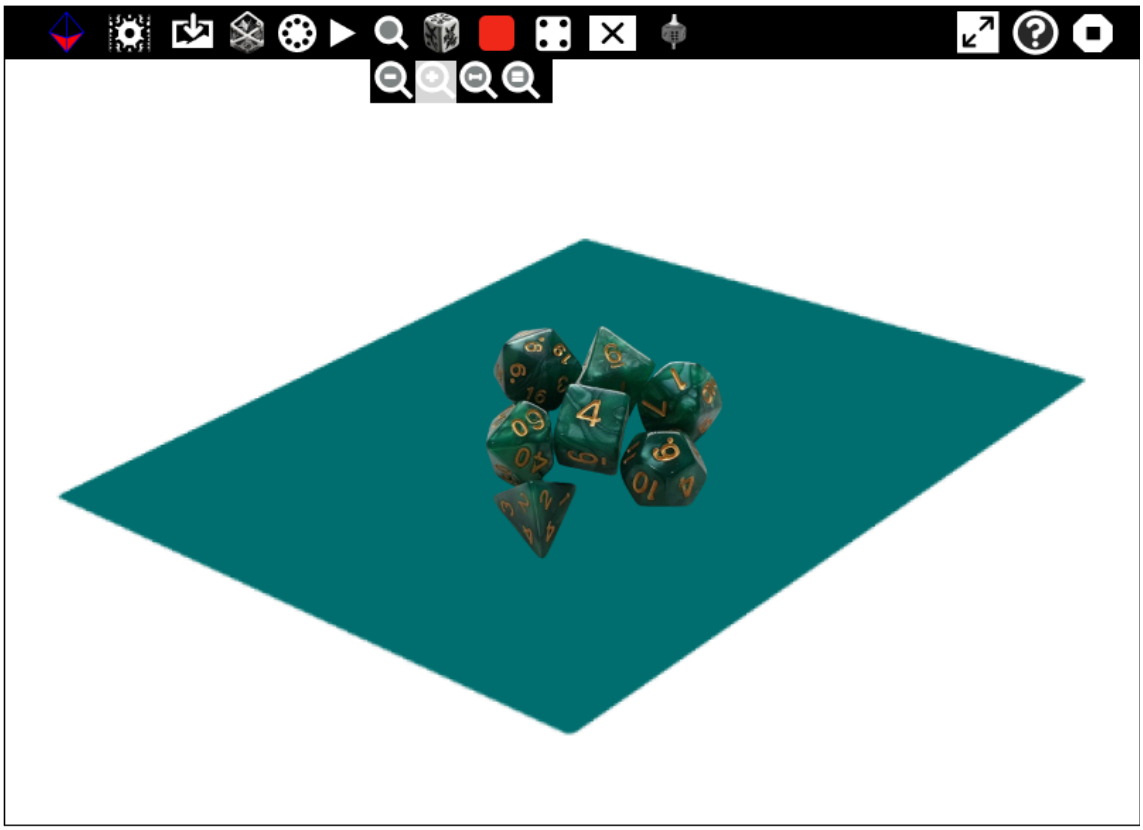
- ➢ Having functions for zoomIn, zoomOut and updateScene to update the view of the user
- ➢ OrbitControls is an add-on, and must be imported to control the camera, for initiation of the user interaction
- ➢ Adding a constructor for rendering the camera objects, and using elements from the OrbitControls for complete user interaction
- ➢ Using the
  .enableRotate : Boolean
  Enable or disable horizontal and vertical rotation of the camera. Default is true.

  .enableZoom : Boolean
  Enable or disable zooming (dollying) of the camera.
- ➢ Initializing the controls inside the main activity.js file

```
import * as THREE from 'three';
import { OrbitControls } from 'three/examples/jsm/controls/OrbitControls';

export class CustomOrbitControls {
 constructor(camera, rendererDomElement) {
    this.controls = new OrbitControls(camera, rendererDomElement);
    this.controls.enableDamping = true;
    this.controls.dampingFactor = 0.25;
    this.controls.enableZoom = true;
    this.controls.zoomSpeed = 1.0;
    this.controls.enableRotate = true;
    this.controls.autoRotate = false;
    this.controls.enablePan = true;
    this.controls.screenSpacePanning = true;
    this.controls.minDistance = 2;
    this.controls.maxDistance = 10;
    this.controls.minPolarAngle = 0;
    this.controls.maxPolarAngle = Math.PI;
    this.controls.minAzimuthAngle = -Infinity;
    this.controls.maxAzimuthAngle = Infinity;
    this.controls.target.set(0, 0, 0);
 }
```

```
import { CustomOrbitControls } from './controls';

 methods: {
    initThree() {
      this.scene = new THREE.Scene();
      this.camera = new THREE.PerspectiveCamera(75, window.innerWidth / window.innerHeight, 0.1, 1000);
      this.renderer = new THREE.WebGLRenderer();
      this.renderer.setSize(window.innerWidth, window.innerHeight);
      document.body.appendChild(this.renderer.domElement);
      this.camera.position.z = 5;
      this.orbitControls = new CustomOrbitControls(this.camera, this.renderer.domElement);
    },
    animate() {
      requestAnimationFrame(this.animate);
      this.renderer.render(this.scene, this.camera);
      this.orbitControls.update();
    },
    zoomIn() {
      this.orbitControls.zoomIn();
    },
    zoomOut() {
      this.orbitControls.zoomOut();
    },
 },
```

UI for the Zoom in / Zoom Out part



❖ Sharing the Activity between connected users

- ➢ Sharing these activity between users connected to the same network and displaying a message whenever shared to show the name of user connected
- ➢ Then whoever throws the die, their name gets displayed and their total is calculated separately
- ➢ Then the one with more score is displayed as a winner with a message, Then an audio respectively for winner and the one who has lost

```
<div v-show="showNetwork">
   <div v-for="(item, index) in networkItems" :key="index" class="network-icon-
container">
      <sugar-icon
        :icon="item.icon"
        :size="constant.sizeNeighbor"
        :colorized="true"
        :classes="item.classes"
        @popupShow="showPopup(item)"
        @popupHide="hidePopup(item)"
      />
   </div>
</div>
```

UI for the Shared Part

❖ Displaying Results

➢ Clicking the Play button to roll the die and after all the die's have been rotated, to compute the total and display it at the end

➢ It also has to be checked like which structure has been selected, since the result can't be computed for the transparent structures

```
methods: {
 rollDice() {
    const dice = this.createDice();
    this.dice.push(dice);
    this.totalResult = 0;

    dice.body.applyImpulse(
      new CANNON.Vec3(Math.random() - 0.5, Math.random() - 0.5, Math.random() - 0.5),
      new CANNON.Vec3(0, 0, 0)
    );
 },
}
```

```
methods: {
 updateDicePositions() {
    this.dice.forEach(dice => {
      dice.mesh.position.copy(dice.body.position);
      dice.mesh.quaternion.copy(dice.body.quaternion);
      // Pseduo logic for calculation of the result
      const face = Math.round(dice.mesh.position.y);
      this.totalResult += face;
    });
    console.log('sugar.getEnvironment.user()' + ' Score is :- ', this.totalResult);
 },
},
```

Testing Plan

- Using Vue Test utils with Jest for testing the vue components created across various files
- I am also thinking of using Selenium Testing for testing methods which I propose to create for various files, which would help in automation of them
- This is just a rough implementation and not a rigid one. These may change once I will start working on the project to test all the possible occurring scenarios. I'll be referring to this article to implement Vue util.js with jest to test my Vue components.

How will it impact Sugar Labs ?

Sugarizer is a learning tool for children. It was conceived to work fully offline, so it could work by calling local files and with electron in a PC, Currently, Sugarizer has some 3D activities like Planets which is educational in it's own way, so is my thought process behind the creation fo 3D Volume Activity for Dice as the students can run through the physics of it also, and we have to make it in such a way so that the childrens enjoy it as 3D activities take lot of CPU resources and memory. I am using Three.JS and CannonJS for it as those are the most popular 3D frameworks, easy to maintain and follow in their updated versions, considering my future prospects it is necessary to have a codebase easier to understand and contribute, to let new contributors comfortably come in.

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

The main part of the project will revolve with coding in Vue.js 3, ThreeJS, CannonJS along with making CSS adjustments to ensure the new component templates align with the current Sugar UI.

Timeline

Break down the entire projects into chunks and tell us what will you work on each week

Days	Tasks
Pre GSoC **Feb 22, 2024 - May 1, 2024**	❖ Learning the working of Vue.JS 3, understanding the files and architecture of Sugarizer ❖ Working towards initial code creation to get setup of the new activity ❖ Contributing to Sugarizer and staying connected with the community
Community Bonding **May 1, 2024 - May 26, 2024**	❖ Discuss the creation of testing for this project and learn about it ❖ Discussing about the new version and features to be added ❖ Working around the tech stack for it
Week 1 **May 27, 2024 - Jun 2, 2024**	❖ Unit testing of files to be done ❖ Writing the order in which files to be created
Week 2 **Jun 3, 2024 - Jun 9, 2024**	❖ Testing of files, an additional buffer taken here, so that the process could be done effectively ❖ Creation of `board.js` file for displaying the board having changing of background

	functionality
Week 3 **Jun 10, 2024 - Jun 16, 2024**	❖ Completion of `board.js` file with rotating screen feature added too
Week 4 **Jun 17, 2024 - Jun 23, 2024**	❖ Initiation of file for addition of volumes on to the board, with feature to select with type of die to be added
Week 5 **Jun 24, 2024 - Jun 30, 2024**	❖ Completion of creation of the file for adding volumes of die as this is one of the most important part of the activity
Week 6 **Jul 1, 2024 - Jul 7, 2024**	❖ Writing methods for addition of color to the die, the color palette and for the buddy color as the initial color of the dices
Mid Phase Evaluation **Jul 8, 2024 - Jul 14, 2024**	❖ Testing of methods using Selenium ❖ Creation of Board to display the area bounded for the dice ❖ Adding Volume onto the dices complete ❖ Adding color to the dices ❖ Code for the above to be cleaned and evaluated so that it doesn't cause future errors
Week 8 **Jul 15, 2024 - Jul 21, 2024**	❖ Initiation file for creation of different types of structures of the dice, like semi-transparent, with numbers etc
Week 9 **Jul 22, 2024 - Jul 28, 2024**	❖ Completion of creation of file for the structures of the dices

Week 10 **Jul 29, 2024 - Aug 4, 2024**	❖ Adding zoom in / out functionalities so that the board and the scene could be viewed as and how the user wants
Week 11 **Aug 5, 2024 - Aug 11, 2024**	❖ Displaying the result of the activity, when the dices are selected without the semi-transparent structure creation of file for displaying result to the user
Week 12 **Aug 12, 2024 - Aug 18, 2024**	❖ Shared activity functionality to be added so that the activity could be shared and more functionalities like winner with audios and such things could be displayed in sharing
Final Phase Evaluation **Aug 19, 2024 - Aug 25, 2024**	❖ Adding the settings file for managing the physics of the activity, if user want to change it and create a new environment he wants for the activity
Final Evaluation Submission **Aug 26, 2024 - Sep 1, 2024**	❖ Writing other small files like audio as needed for the shared part ❖ Cleaning and wrap up the code

*It may not be a task but Handling PR feedback would also be an important thing to work upon every other week
In Aug 2nd week I would be having my semester examinations at that time I would work for a little less duration

How many hours will you spend each week on your project ?

I have my Institute summer vacations starting from May 11 to July 17. In this period I can give about 28-40 hours per week and after college starts, I can give about 20-25 hours per week. I have no other commitments for the summer vacation, so I can devote most of my time to GSoC.

Highlight the work you plan to complete before each evaluation.

`Before Mid Evaluation` Performing unit testing and code coverage, Creation of Board to display the area bound to the dice, Addition of Volume onto the dice and Addition of color to the dice.

`Before End Evaluation` Addition of Structure of the dices, the zoom in / out functionalities, displaying the results after the activity, adding the shared button for multiple users and the settings button for changing the physics of the activity.

How will you report progress between evaluations ?

Every week I would have a meeting with the mentors on discord or on matrix, where I would share my progress.
I will also submit the notes of every meeting and next week tasks on sugar-devel channel
I will be active on GitHub and will contribute regularly to the project, so everyone in the Organization will be able to view my progress.
I am also thinking of writing blogs, to share my progress.

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

I am thinking of working continuously on the project after the project ends. After this project, I would like to work on optimization of other Sugarizer Activities. So that the Web app runs smoothly as designed and students do not face any issue regarding the same. I aim to develop mentorship skills

and the ability to guide others and try to give back to the community by mentoring and guiding others. I hope to mentor future GSoC students, as I improve my skills and become more experienced in the Sugar Labs community.

I am Looking forward to contribute to Sugar Labs, for this summer
Kind Regards