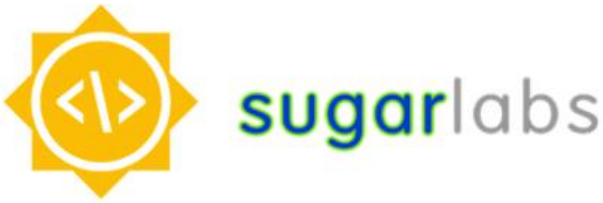


Sugar Labs 

Google Summer of Code 2025

Project Information:

Name: Sugarizer Human Activity Pack (here)
Length: 175 Hours
Difficulty: Medium
Mentor: Lionel Laské
Assisting Mentor: Samarth Bagga

Student Details:

Full Name: Rahul Pandhrinath Shendre
Email: rahulshendre789@gmail.com
GitHub: rahulshendre
LinkedIn: Rahul Shendre
Preferred Language: I'm proficient in English for communication, both spoken and written
Location: Pune (Maharashtra), India

Time Zone: Indian Standard Time (IST) (UTC +05:30)
Phone Number: +91 9730919748
Institution: Vishwakarma Institute of Technology, Pune
Program: B.Tech in Information Technology
Stage of completion: 2nd Year (expected May 2027)

About Me:

I have been a tech enthusiast since childhood, downloading cracked games, booting Windows for my friends when we were in school, and simulating online gaming experience on the same network while being offline. I have done a lot of stuff when I came to tech in general. My experience of open source has been pretty epic, even though I recently got to know about **open source**. I have loved VLC media player for as long as I can remember and have recently been obsessed with VS Code, my go-to IDE when it comes to coding.

Since last year, I have been involved in Web development. I am a web-dev coordinator at GedIT, the coding club of our college, made some pretty cool projects, participated, and was a finalist in State-level and National-level Hackathons, I also do some Leetcode Problems in my free time.

Before coding, I was a freelance video editor on Fiverr and have worked for multiple YouTube channels, one of them being Grey Fang, a channel with over 200k subscribers and many more local customers. this was for around 2 years.
This not only helped me financially but also **taught me how** to work with clients, meeting their expectations and delivering on time, etc.

During this period, I developed a deep understanding of highly **complex software** like Adobe Premiere Pro, Adobe After Effects, and Davinci Resolve.

Programming Languages: JavaScript, TypeScript, Python, C, C++
Libraries/Frameworks: React, Bootstrap, TailwindCSS, Node.js, Express.js, Next.js, Three.js, Paper.js, Tensorflow.
Databases: MongoDB, SQL
Tools/Platforms: Git, GitHub, Vercel, AWS, VS Code, Postman.
Software: Blender

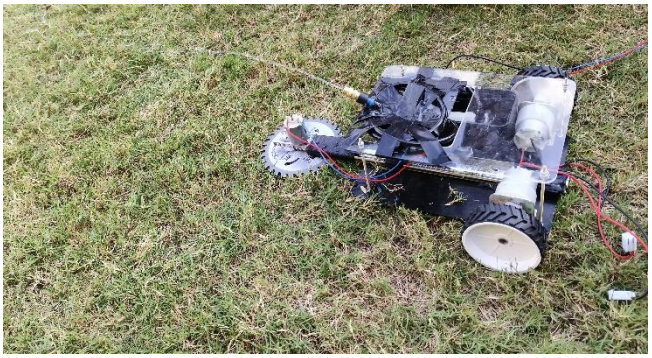
Previous Projects:

Projects are something that teach **real-life implementation** of the stuff you learn. Luckily,

in our college, **project**-based learning is highly promoted. In the past 3 semesters, I have made several projects with my group. Here are the most fun ones –

1. Multifeatured Lawn Mower by Using Arduino UNO-

- This project was a mixture of both **hardware and software**. For the hardware, we used Arduino Uno boards, which are programmed using the Arduino IDE, which is an **open-source** project.
- The main features of the model were, it was controlled by a phone, the app was developed by us, it had a cutter to cut the grass, a brush which cleans the grass after cutting and a water spray to water the grass if needed.



- Helped us learn about the **integration** of hardware with software, logic of the app, and the Arduino board, **all three had to be integrated seamlessly**.

2. Document verification using Pinata

- Code the web app leverages **Pinata** it is a web3 media management platform and **IPFS** (InterPlanetary File System) pinning service that simplifies the storage and sharing of content on the **decentralized web**, offering tools for fast and secure uploads, dedicated gateways, and token-gated experiences.
- This was a pure software project we did.

3. Cloudburst Prediction and Breast Cancer Detection

- The Cloudburst prediction model is a project that revolves around a simple ui and a **highly accurate ML** model, which predicts if a certain place will experience a cloudburst according to its weather conditions in **real time.**
- The Breast Cancer detection using Histopathology is also a project that utilizes **hybrid ML models** to predict whether a person has breast cancer or not, based on the histopathology images.

4. Enhanced Detection of Deep Fakes: Exploring Advanced Approaches

- In the last semester, we published a paper in the **IEEE conference**, a world-renowned technical professional organization dedicated to advancing technology.
- The paper was a survey paper in which my group surveyed over **100 research papers** on the models used for deepfake detection in detail and came to a conclusion about which one is the best model for the detection of deepfakes.

These projects helped me understand how to lead a team, collaborate, and build great stuff with other people.

Why SugarLabs?:

Since I was a kid, I have always relied on the internet to learn pretty much everything, not just fun skills like learning how to solve a cube but also how to score well in examinations. The internet has taught me everything I know today, that too free of cost. In the past decade, since I have been using the internet, I have never spent money on any course, any tool any video… because it was freely available. This is something for which I am beyond grateful for. The same thing has happened with my 10-year-old brother; he too, relies on the internet for his learning. in fact, I encourage him to do so. This is what open source is right, free to use high level software and technology for everyone.

Recently when I stumbled upon the Sugarizer app, I loved the idea of kids being able to learn such interesting stuff freely and in such an interactive manner, I showed some of the activities to my little brother and he was super happy to use them.

My main reason for working with Sugar labs, specifically Sugarizer is to be able to create something which will be used by kids all over the world, my brother his friends and their friends. Contributing to such a big initiative is just simply epic.

My contributions on Sugar Labs will extend beyond GSOC, the idea of helping kids learn
freely and in such an interactive way resonates with me deeply.

Project Details

Project Title

Sugarizer Human activity pack

Activity Logo-

Video Demos-

Human activity's demo-
https://drive.google.com/file/d/10G4_c52d6uRzfdE2ZCUoYWrLnapzsBdj/view?usp=sharing

Stickman activity's demo-
https://drive.google.com/drive/folders/1PbfVnbFJa1-aiAb2rDxCzCJ6KOUPWS7Z?usp=sharing

Link to pawn activity video (posted on the Discord channel) –

https://discord.com/channels/1078051575580336249/1078051576284975226/1337203497904508988

Current size-

3D Human activity.js file – 900 -1000 lines of code
Stickman animator activity.js file – 600-700 lines of code

What are you making?

The Human body activity pack comprises of two activities first one being the **3D Human Body** activity is a Sugarizer activity made to teach its users about human body by implementing a way to toggle between different layers **3D** of the body – **Skeleton layer, Muscle layer and Organs layer**. It provides an interactive platform where users can **visualize, interact and learn** playfully and engagingly.

Implementation of intuitive controls and the paint feature, the activity encourages hands-on exploration, enabling users to zoom in for detailed examination and interact with various parts of the body. **Whether adjusting colors, selecting different model types, or using tools like learn, paint, doctor, and tour,** users have the freedom to choose how they want to learn according to their preferences.

The different models provide a way to learn and explore **all the body parts of the human body**, not just from the outside but also from the inside.

Furthermore, the activity promotes collaborative learning by facilitating shared experiences among connected users. Students can engage with peers, exchange ideas, and also compete in the same, There is also a **leaderboard** which will display the scores of the players according to their answer for the questions that arise when they shift to the doctor mode.

I have used the best assests suitable for the activity, **simple and open source models which are free to use** and editing them according to how you see fit, opens a whole new door for even better use of them.

The second activity is the **Stickman animation activity**. The goal of this activity is to provide users with a **2D stickman figure** that can be manipulated in any way the user prefers. This is achieved by moving the dots placed at the joints of the stickman. **These poses can be captured, and by playing them in sequence, you create an animation. This animation can be exported as a video file.**

Users can play and pause the frames while animating the stickman for better feedback, changing speed also helps in creating better animations and an access to fun templates makes the experience even better, user just chooses the animation he wants, eg: he chooses running then all the poses and frames which make up running will automatically load in the tab of selected poses and just hit play the stickman starts to run.

The most exciting feature and the one I am hyped about is the **Sharing and Collaboration** in the Stickman activity, it will be **real-time sharing of the moves you and the other players do**, interaction by using your own characters will take the experience to a whole new level.
Once on the same canvas, user have the freedom to do anything they wish, waving, dance battles, goofing around, animations using their templates to do anything they want.

Importing a **photo of a pose and having the stickman turn into the same pose is also a key feature of the activity**. it will require Tensorflow.js in browser, Currently I am exploring this feature of the activity.

How will it impact Sugar Labs?

Adding the **Human activity Pack** to Sugar Labs will provide user to learn about the human body in a very fun and interactive manner and being able to play around with stickman and interacting with other players will enhance their creativity and overall being, it will provide users with a specialized platform for exploring and manipulating Human body's Layers – **Skeletal, Muscular and Organs**. Existing activities using 3D models, such as the 3D volume activity and Planets activity, which concentrates on 3D shapes and Planets, this new activity will facilitate interactive learning and experimentation with different Layers of the body. It will cater to diversify not only the learning styles but also the **overall 3D learning activities of Sugarizer while encouraging collaboration**; it enriches the educational experience within Sugarizer while deepening comprehension of the body's understanding and **basic animation principles.**

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

The primary programming language for this project is JavaScript, selected for its versatility. The additional software and frameworks utilized include:

- **Blender**: Used for **creating** and **refining 3D models**. Blender was chosen for its powerful modeling capabilities, open-source nature, and compatibility with various export formats. It allows for precise asset creation, ensuring **high-quality 3D elements for the 3D Human Body activity**.

- **Three.js**: Employed for **displaying all the 3D Model**s. Three.js was chosen for its open-source nature, lightweight footprint, and **proven track record** in other Sugarizer activities like the 3D volume activity Planets Activity.

- **Paper.js**: Used for handling vector graphics and **animations** in the **Stickman Animation activity**. Paper.js was chosen due to **its lightweight nature**, ease of use for interactive drawings, and ability to efficiently manage complex shape manipulations. It simplifies working with paths, curves, and **dynamic stick figure animations, making it ideal for keyframe-based movement.**

Blender is used as a stand alone software to edit, create export the 3d models required for 3D human body activity based on the needs of the project.

Implemented Features

I have been able to successfully implement almost all features mentioned in the Ideas-2025.md file for this activity. Below are the features which I have integrated into the activity:

3D Human Body Activity

1. Identifying the missing assets for the activity

The most important part for the completion of this activity, the **assets which suit the needs for the project and are open source or at-least free to use even if they are licensed**, there was a skeleton model already implemented by [Samarth Bagga], so I had to find the muscles/ skin model and the organ model.

Majority of my time was actually spent in creating the models, not from scratch though, As mentioned in the about me section previously I used to work with video editing software so picking up on blender was not that hard. I loaded the models in blender and edited them accordingly.

As the activity is based around teaching kids about different body parts the model should be made of **different meshes** to be able to work and interact with them like coloring and identifying them separately.

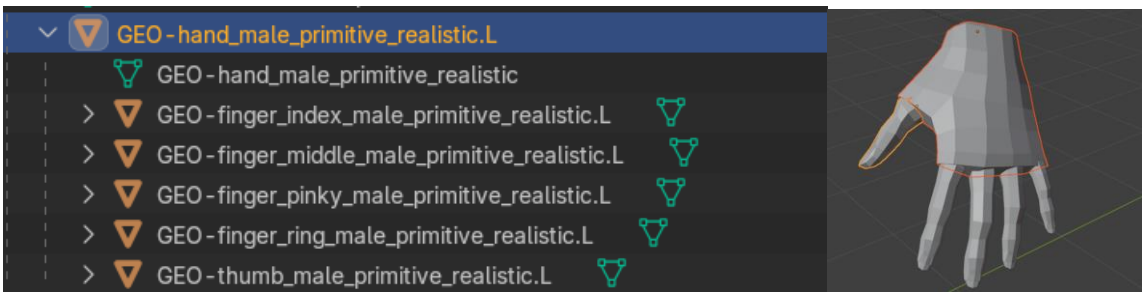

```
  {
    "mesh": 0,
    "name": "SK_Head_Material.001mat_0"
  },
  {
    "children": [
      6
    ],
    "matrix": [
      100.0,
      0.0,
      0.0,
      0.0,
      0.0,
      99.9999999999999,
      4.371138828673792e-06,
      0.0,
      0.0,
      -4.371138828673792e-06,
      99.9999999999999,
      0.0,
      -51.67522430419922,
      195.07705688476563,
      -7.321474075317383,
      1.0
    ],
    "name": "SK_RArmDown"
  },
  {
    "mesh": 1,
    "name": "SK_RArmDown_Material.001mat_0"
  },
```

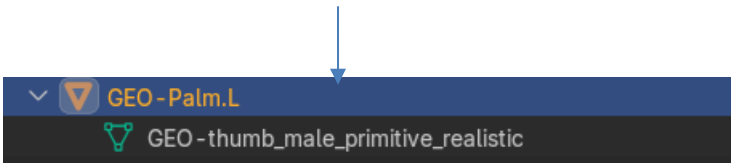

In the skeleton model there are 18 different meshes/ parts/ bones.

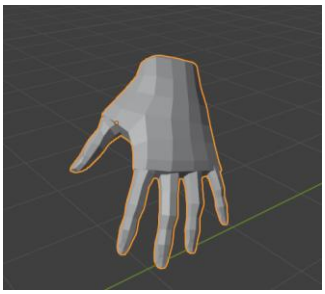
```
[
  { "name": "Head", "mesh": "SK_Head_Material001mat_0", "position": [-0.04, 7.80, 0.56] },
  { "name": "Ribcage", "mesh": "SK_Side_Material001mat_0", "position": [-0.06, 5.16, 0.48] },
  { "name": "Right Arm Down", "mesh": "SK_RArmDown_Material001mat_0", "position": [-2.22, 2.65, -0.12] },
  { "name": "Right Arm Up", "mesh": "SK_RArmUp_Material001mat_0", "position": [-1.46, 4.79, -0.44] },
  { "name": "Right Hand", "mesh": "SK_HandR_Material001mat_0", "position": [-2.38, 1.35, 0.15] },
  { "name": "Left Arm Down", "mesh": "SK_LArmDown_Material001mat_0", "position": [2.11, 2.70, -0.35] },
  { "name": "Left Arm Up", "mesh": "SK_LArmUp_Material001mat_0", "position": [1.43, 4.85, -0.43] },
  { "name": "Left Hand", "mesh": "SK_HandL_Material001mat_0", "position": [2.36, 1.12, -0.20] },
  { "name": "Right Leg Up", "mesh": "SK_RLegUp_Material001mat_0", "position": [-0.86, 0.26, 0.09] },
  { "name": "Right Leg Down", "mesh": "SK_RLegDown_Material001mat_0", "position": [-0.54, -2.93, -0.12] },
  { "name": "Left Leg Up", "mesh": "SK_LLegUp_Material001mat_0", "position": [0.92, 0.33, 0.08] },
  { "name": "Left Leg Down", "mesh": "SK_LLegDown_Material001mat_0", "position": [0.43, -2.95, -0.15] },
  { "name": "Right Foot", "mesh": "SK_RFoot_Material001mat_0", "position": [-0.73, -4.97, 0.24] },
  { "name": "Left Foot", "mesh": "SK_LFoot_Material001mat_0", "position": [0.55, -4.87, 0.25] },
  { "name": "Coccyx", "mesh": "SK_Coccyx_Material001mat_0", "position": [-0.05, 2.03, -0.59] },
  { "name": "Spine", "mesh": "SK_Spine_Material001mat_0", "position": [-0.11, 3.05, -0.08] },
  { "name": "Right Clavicle", "mesh": "SK_RClavicle_Material001mat_0", "position": [-0.87, 5.88, -0.60] },
  { "name": "Left Clavicle", "mesh": "SK_LClavicle_Material001mat_0", "position": [0.84, 5.87, -0.64] }
```


For editing and storing the models in **.gltf** format I have used **Blender software** and Threejs for interaction and loading. For the muscle model I was able to edit this free model which and converted this into something which we could use for the activity.

This has a **lot of meshes, around 50**, which gives a lot of freedom in editing the model however you like. For example, I can merge the meshes of the fingers to make the palm

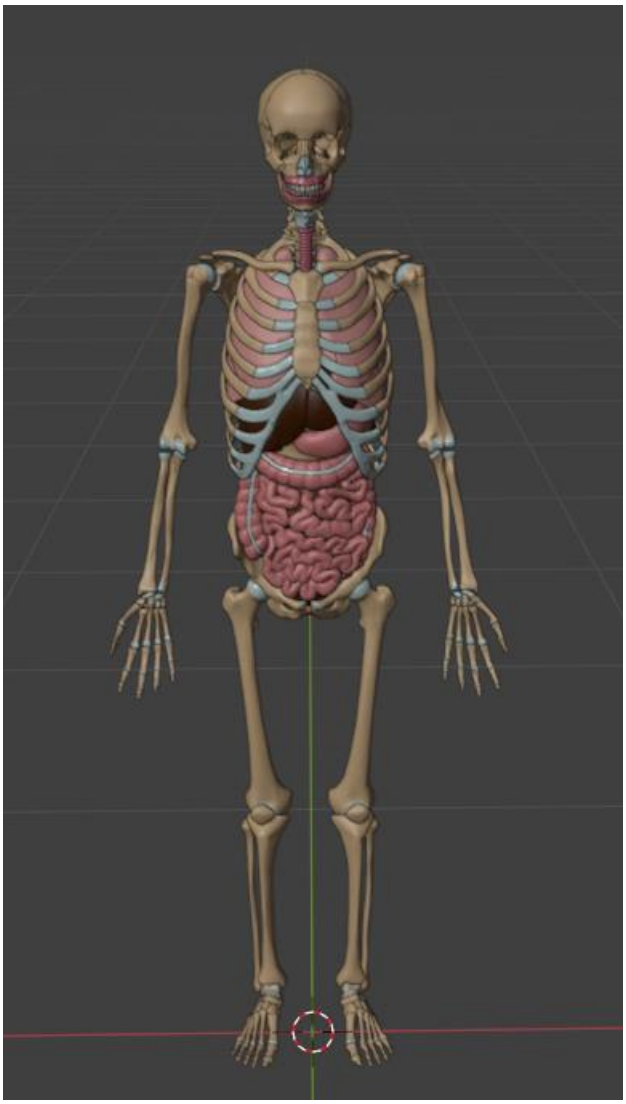




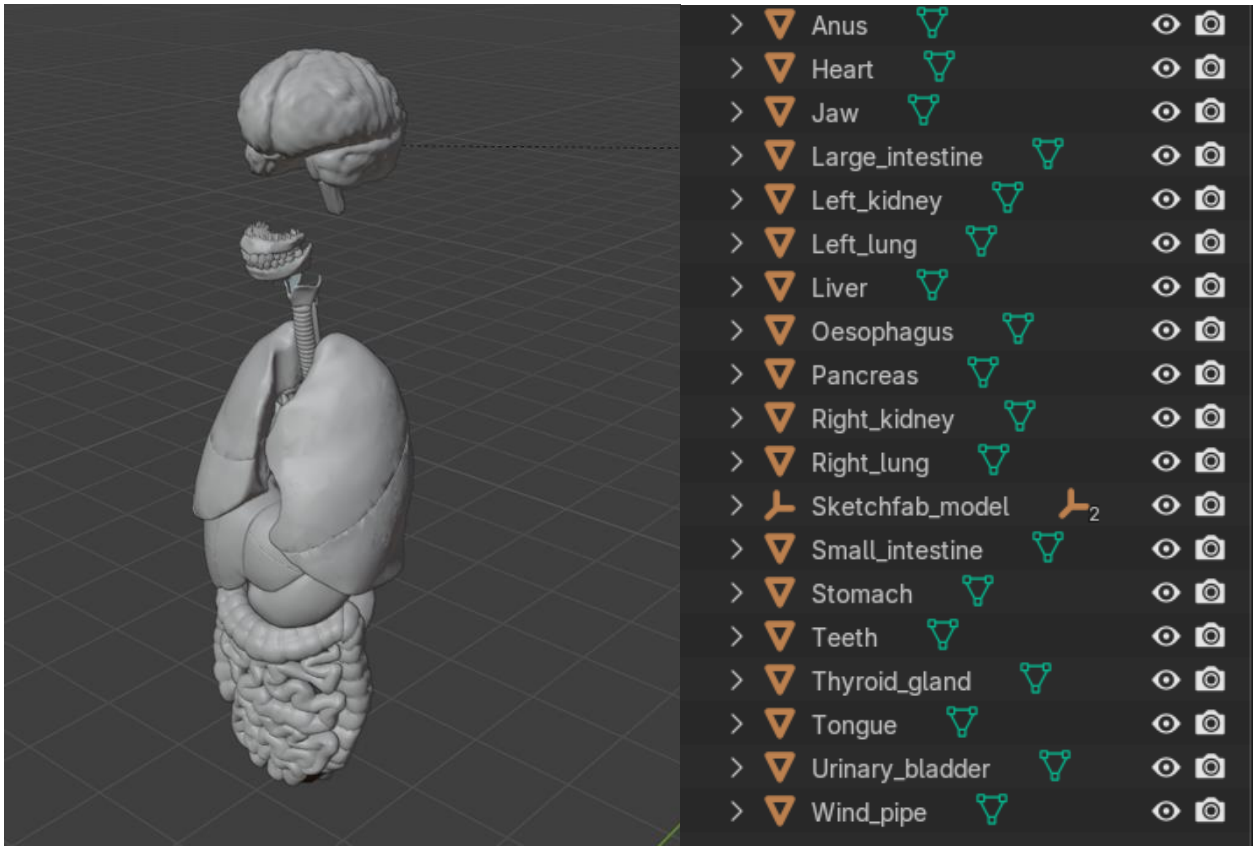


In the first one, fingers and the other part are different from each other, in the second one I have merged the meshes together to make the palm as one, the same thing can be done to all parts of the model.

Similarly, we have the **organs model**, this was extracted from a very complex model of both bones and the organs.



This had different textures for each and every organ and the brain were missing, adding the brain and making the texture same for all, made this the best asset which could be used for the human activity's organs section.

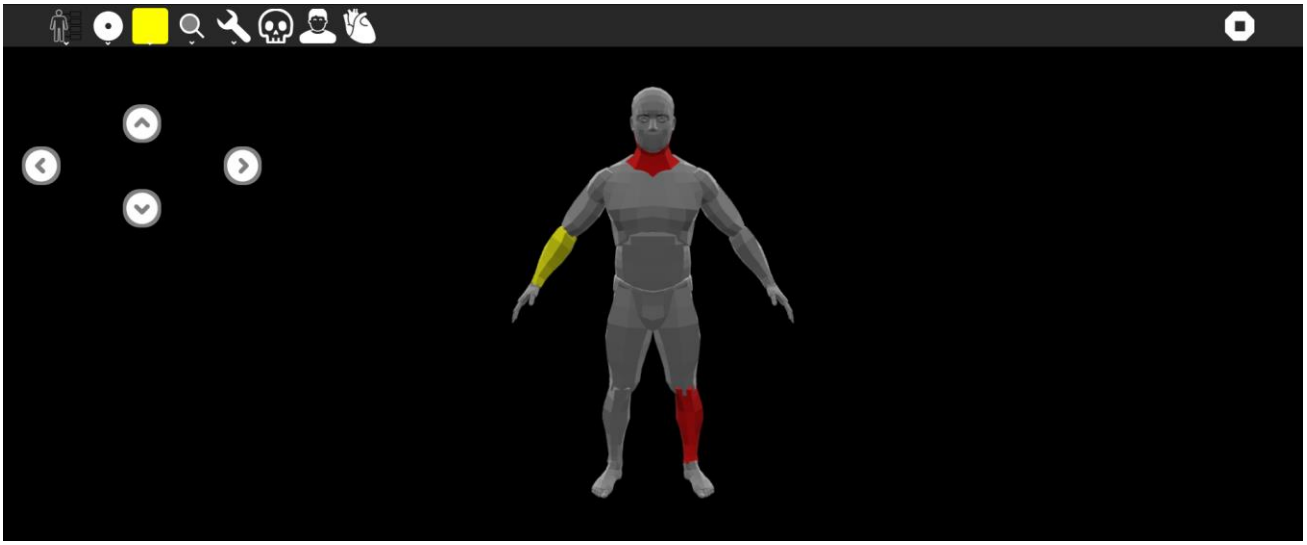


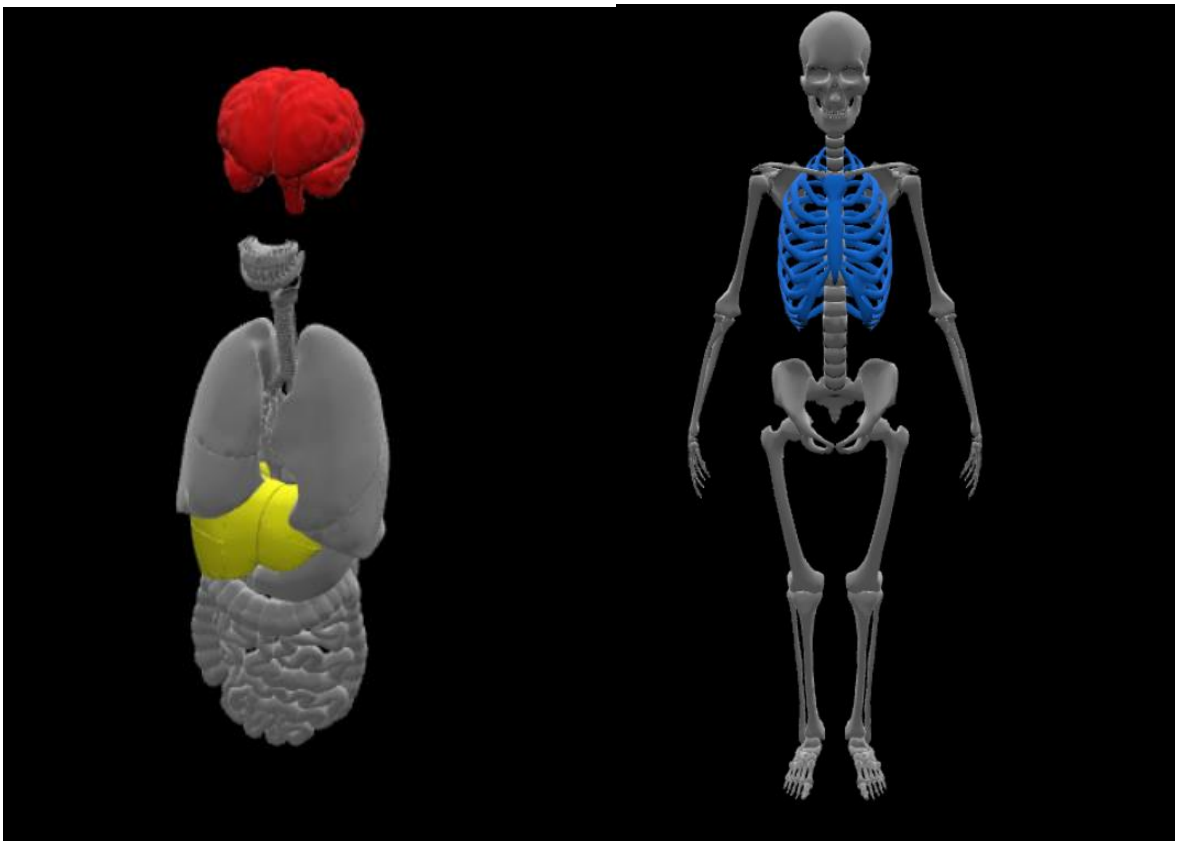
As of now this **has 19 different meshes/ organs, which can be further reduced or increased according to our needs.**

2. Integrating these layers and a way to change them

   Link to the video demo of the feature implemented – https://cdn.discordapp.com/attachments/1078051576284975226/1350709084105736283/human_activity_progress_2.mp4?ex=67ef745c&is=67ee22dc&hm=f0813bc40a8964ee3e69bded36895931e4333c43bf016c3e94aebbcad9c689ed&

   This feature gives the user **3 options/ 3 models** to choose from the skeleton, muscles and organs, the user can easily **toggle** and interact with all the three models.

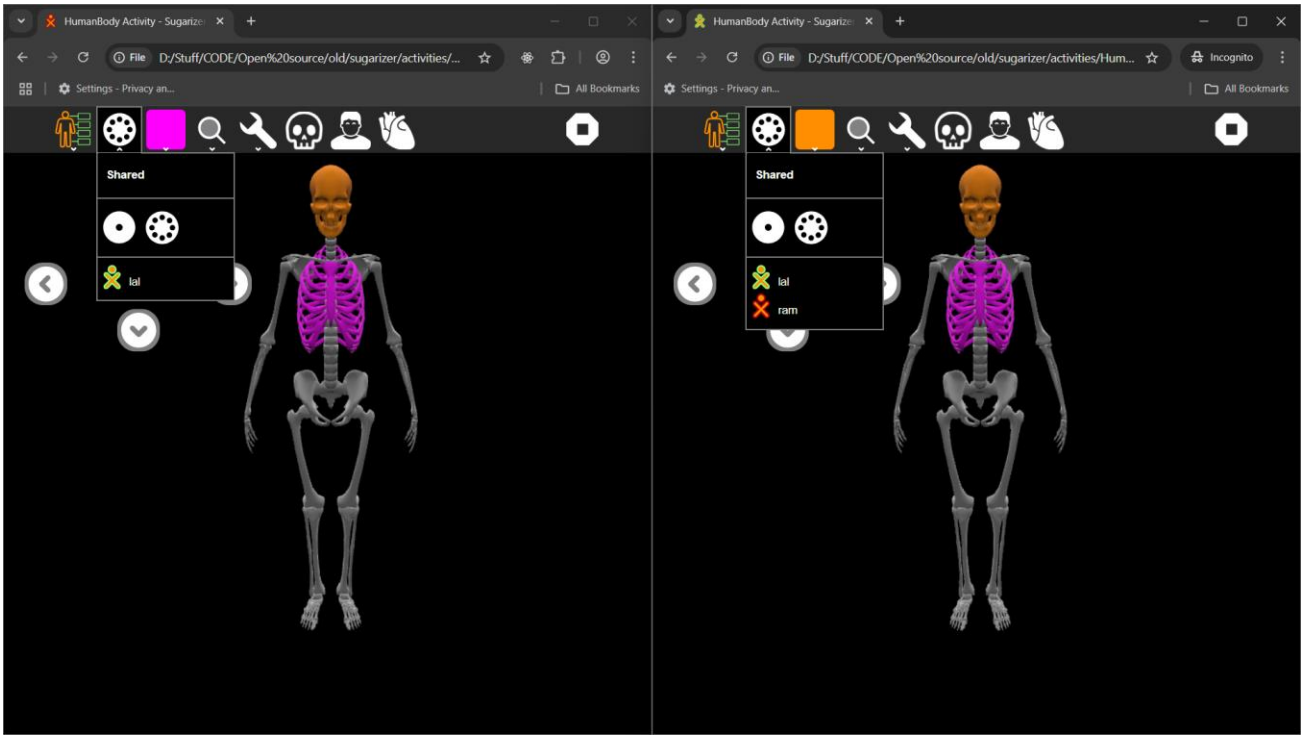




```
// Event listener for loading the skin/muscle model
document.getElementById("skin-button").addEventListener("click", () => {
  // Remove the currently displayed model and skeleton
  removeModel(currentModel);
  removeModel(skeleton);

  // Load and add the new skin/muscle model
  loader.load(
    "models/skin/skin.gltf",
    function (gltf) {
      const skinModel = gltf.scene;
      skinModel.traverse((node) => {
        if (node.isMesh) {
          node.userData.originalMaterial = node.material.clone();
        }
      });

      // Position and scale the skin/muscle model
      skinModel.position.set(0, -3.5, 0);
      skinModel.scale.set(2.35, 2.35, 2.35);

      scene.add(skinModel);
      currentModel = skinModel; // Update the current model
      console.log("Skin model loaded successfully.");
    },
    function (xhr) {
      console.log((xhr.loaded / xhr.total) * 100 + "% loaded");
    },
    function (error) {
      console.error("An error occurred while loading the model:", error);
    }
  );
});
```

Uses simple event listener, whenever the buttons for different models are clicked the existing are removed and the new one is loaded on the canvas.

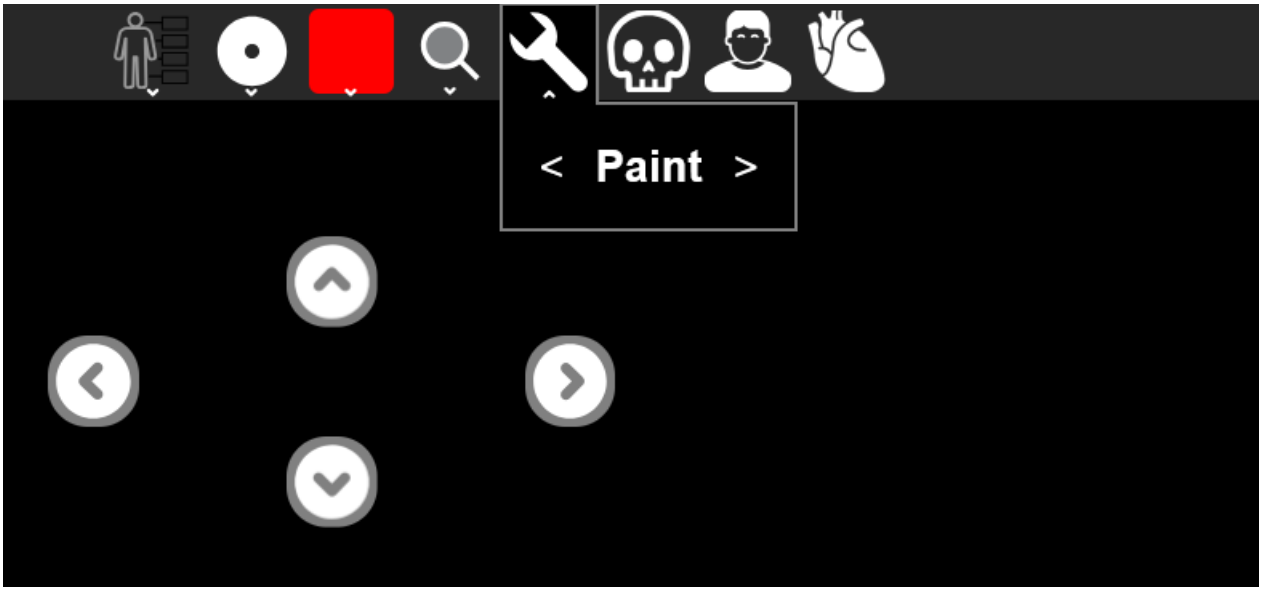
3. Sharing



Users can **share the activity on the server**, allowing others to join and participate. After joining, users can freely **interact** with the models and use all the functionalities, currently this has only been implemented in the skeleton model and I am working on using the collaboration for other models as well.

This feature is an integral part of the Sugarizer and has been implemented by using the presence palette.

4. Reviewing the Toolbar and pop ups-



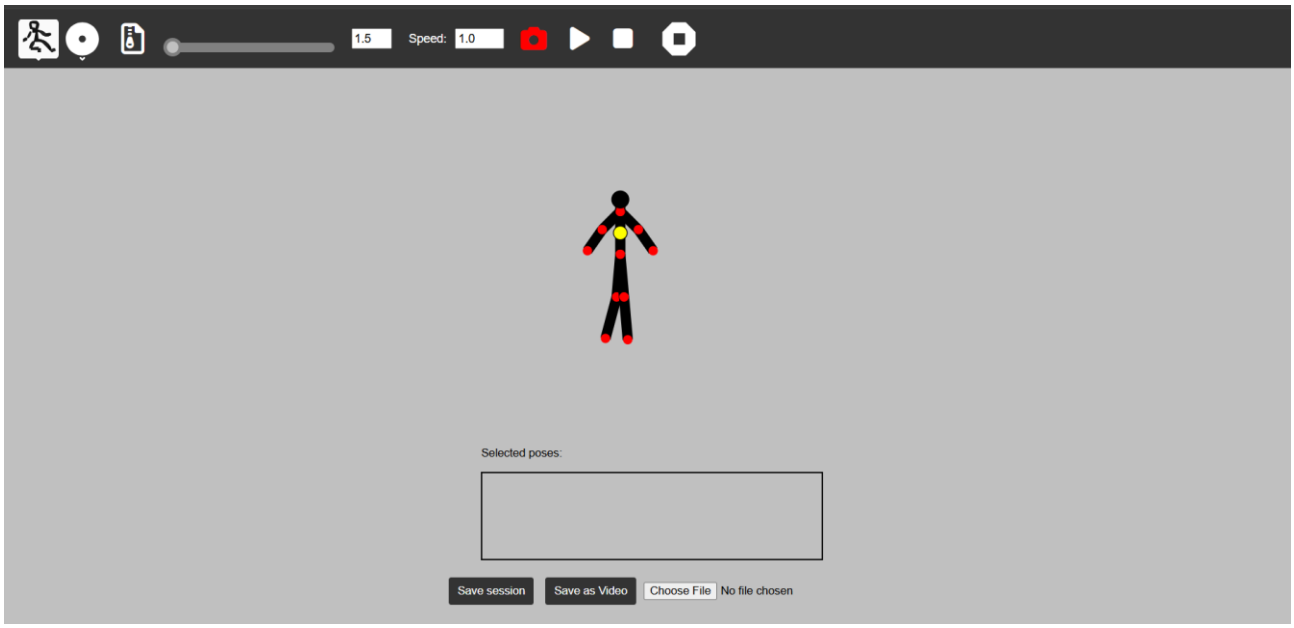
Currently the toolbar has the icon of the activity, the network button, the color palette the zoom palette, the toolbar and the popups are for learn, tour and doctor mode – which were already implemented.

What I added were 3 other buttons to toggle between models – the skeleton, the muscle and the organ's button.
Some other popups that can be added could be while switching between the models .

Stickman Animation Activity

1. Put the stickman figure in different poses by moving dots-

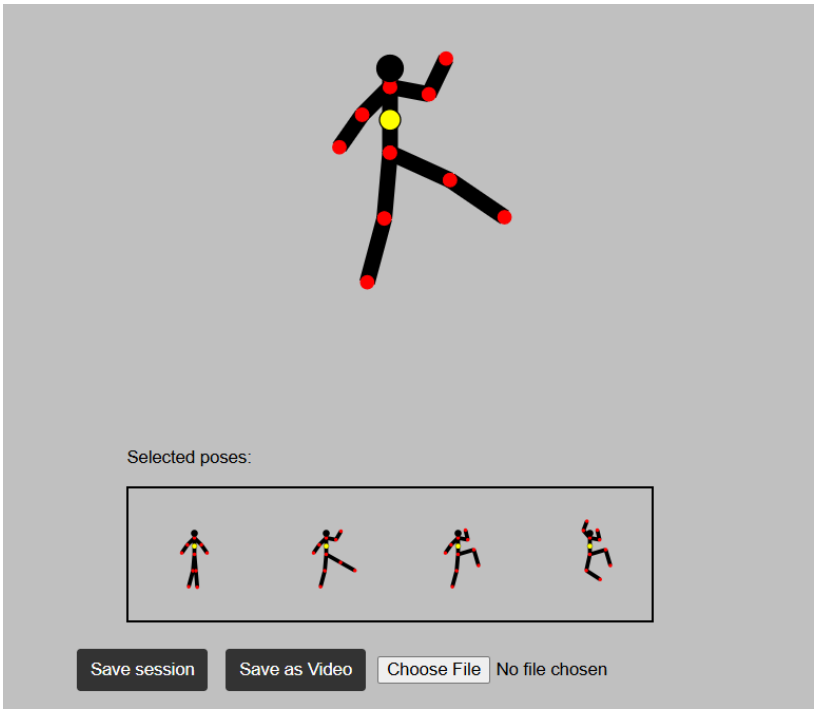
Users can change the move the body parts of the stickman through the by simply selected the part they want to move and move the part to 360 degrees. **The default pose is the one shown below and the red dots represent the joints of the stickman and yellow, is the dot that if selected will help you move the whole stickman to anywhere across the canvas**.



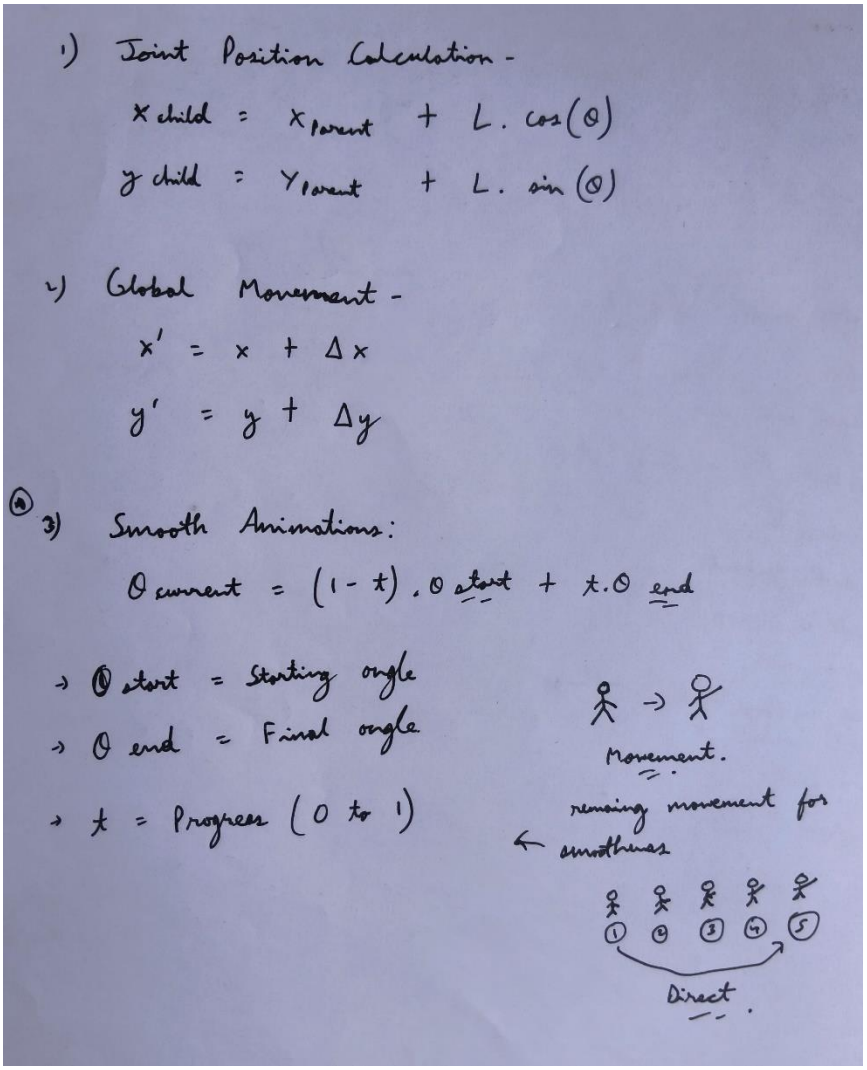
The toolbar consists of the Logo of the stickman activity, the network button, a templates button, a slider which shows the length of the animation, a speed section, a camera icon, which **captures** the current pose and stores it in **the selected poses** section, then hit the **play** button the poses play with a **smooth** animation. Then we have the **pause** button and the stop button of the activity. Also a save **session** feature which exports all the poses in a JSON file, which could be used later, and a save as video feature too, saving the animation created in a **video**.

These are the initial stages of the toolbar; it needs a lot more features and improvements.

After recording the poses the poses are automatically stores in the selected poses Tab and give out smoother animations using the **interpolation logic, explained below.**



Logic behind the movements and animation:



The whole stickman works on a parent child relation, in essence when the parent is moving the child will also be moved and the position of the child will always adhere to the position of the parent

1. Joint Position calculation -

```
function Transform(theta,t,x){
    /**
     * Homogeneous transfomration
     * theta: angle [degree]
     * t    : translation vector
     * x    : point
     */
    let ang = theta*Math.PI/180;
    let M = [[Math.cos(ang), -Math.sin(ang), t[0]],
             [Math.sin(ang),  Math.cos(ang), t[1]],
             [0.0,                0.0,      0.0]]

    let tfvec = [M[0][0]*(x[0]+t[0]) + M[0][1]*(x[1]+t[1]),
                 M[1][0]*(x[0]+t[0]) + M[1][1]*(x[1]+t[1])]

    return tfvec
}

export function angle(X1,X2){
   return (2*Math.PI + Math.atan2(X2[1]-X1[1],X2[0]-X1[0]))%(2*Math.PI)
}
```

This function is responsible for the **Joint position calculation:**

1. Converts theta from degrees to **radians**.
2. Constructs the 2D transformation matrix.
3. Computes the new position of the point after transformation.

The basic idea of it has been given in the handwritten document attached.

2. Global Movement –

```
// callback to change pose when joint is moved
tool.onMouseDrag = function (event) {
    if (appState.selectedPart != null) {
        let location = event.point
        let currJointAngle = Node.angle(appState.selectedPart.parent.endPosg,appState.
        selectedPart.endPosg)
        let newJointAngle  = Node.angle(appState.selectedPart.parent.endPosg,[event.
        point.x,event.point.y])
        let diff1 = (2*Math.PI + (currJointAngle - newJointAngle))%(2*Math.PI)
        let diff2 = (currJointAngle - newJointAngle)
        let diff  = 0
        if(diff1 > Math.abs(diff2)){
            diff = diff2
        }else{
            diff = diff1
        }
        appState.selectedPart.UpdateJointAngle((360 + appState.selectedPart.angle -
        diff*180/Math.PI)%360)
        Node.UpdatePose(stickman)
        appState.update = true
    }
}
```

- Checks if a joint is selected (appState.selectedPart).
- Gets the new mouse position (event.point).
- Calculates the current and new joint angles.
- Computes the angle difference for smooth rotation.
- Updates the joint's angle in degrees.
- Applies the new pose to the stickman (Node.UpdatePose(stickman)).
- Marks for rendering update (appState.update = true).

Again, the basic idea of this is in the handwritten document.

3. Smooth Animations-

How It Works:

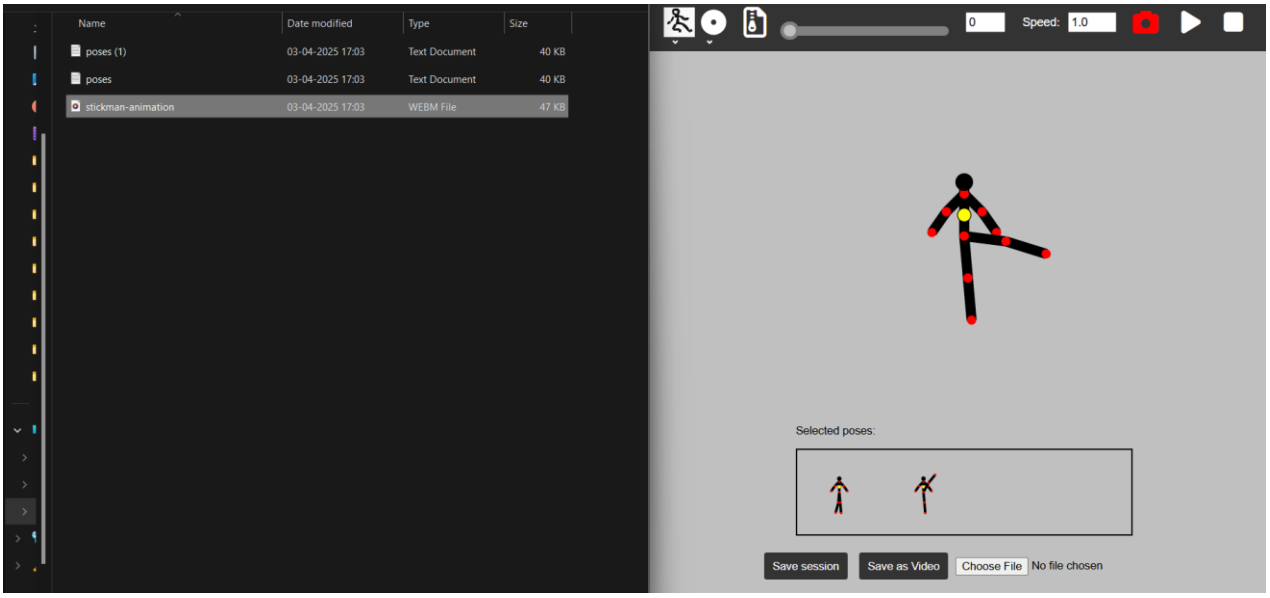
1. **Finds two recorded poses**:
    - `j1` and `j2` are the values of a joint at two different times (`t1` and `t2`).
    - It gets these values from `appState.recordedPoses`.
2. **Calculates the difference**:
    - For **position values** (`posX`, `posY`), the difference is simple: `j2 - j1`.
    - For **angles**, it finds the smallest rotation difference (avoiding large jumps).
3. **Interpolates the value**:
    - It linearly blends between `j1` and `j2` based on time `t`.
    - This ensures smooth transitions from one pose to the next.

This function is crucial for **smooth motion** between recorded frames.

```
function InterpolatePoses(t){

    let interpolateJoints = function(i,t,name){
        let j1 = appState.recordedPoses[i-1][1][name]
        let j2 = appState.recordedPoses[i][1][name]
        let t1 = appState.recordedPoses[i-1][0]
        let t2 = appState.recordedPoses[i][0]
        let diff = 0
        if(name == "posX" || name == "poxY"){
            diff = j2-j1
        }else{
            let diff1 = (360 + (j2 - j1)) % 360
            let diff2 = (360 + (j1 - j2)) % 360
            if (diff1 > diff2) {
                diff = -diff2
            } else {
                diff = diff1
            }
        }
        let j  = j1 + diff*(t - t1)/(t2-t1)
        return j
    }
```

You can also refer to the document for the formula of this.

2. Saving as video and saving the poses themselves-



The poses shown in the selected poses tab, which are captured by the camera button above and can be stored in a **txt document in json format as and the animation will be stored as a video.**

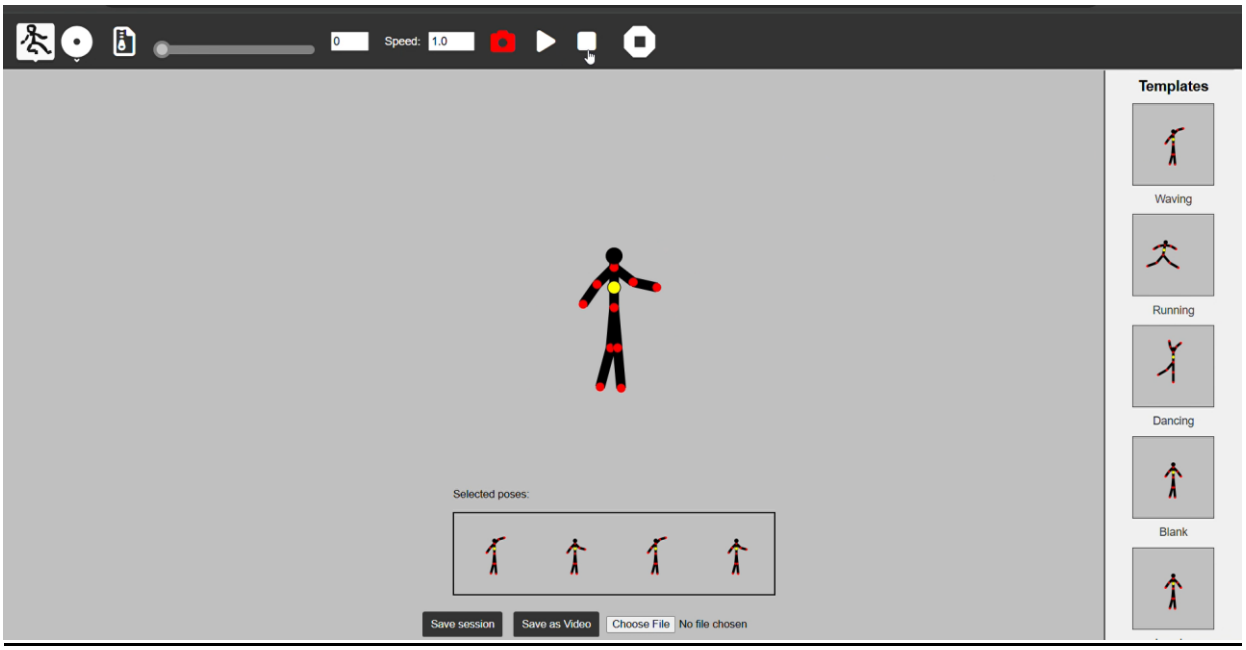
The functions below are responsible for capturing poses, exporting them as video and txt.

```
function snapOnClick() {
    let poset = parseFloat(appState.slider.value);
    let availindex = -1;
    for (let i = 0; i < appState.recordedPoses.length; ++i) {
        if (poset == appState.recordedPoses[i][0]) {
            availindex = i;
            break;
        }
    }
    var shot = canvas.toDataURL("image/png");
    let pose = {};
    GetCurrentPose(stickman, pose);
    if (availindex < 0) {
        appState.recordedPoses.push([poset, pose, shot]);
    } else {
        appState.recordedPoses[availindex][1] = pose;
        appState.recordedPoses[availindex][2] = shot;
    }
    appState.recordedPoses.sort((a, b) => a[0] - b[0]);
    showRecoredPoses();
    UpdatePose(stickman, defaultPose);
    appState.currentPose = defaultPose;
}
```

```
document.getElementById("download").addEventListener("click", () => {
    if (appState.recordedPoses.length > 0) download(JSON.stringify(appState.
    recordedPoses), "poses.txt", "text/plain");
    else alert("No recorded poses");
});
```

```
function startRecording() {
    if (!canvas.captureStream) {
        alert("Your browser does not support canvas.captureStream().");
        return;
    }
    const stream = canvas.captureStream(60);
    mediaRecorder = new MediaRecorder(stream, { mimeType: "video/webm" });
    mediaRecorder.ondataavailable = function (event) {
        if (event.data.size > 0) recordedChunks.push(event.data);
    };
    mediaRecorder.onstop = function () {
        const blob = new Blob(recordedChunks, { type: "video/webm" });
        const url = URL.createObjectURL(blob);
        const a = document.createElement("a");
        a.href = url;
        a.download = "stickman-animation.webm";
        a.click();
        URL.revokeObjectURL(url);
        recordedChunks = [];
        appState.isRecording = false;
    };
    mediaRecorder.start();
    appState.isRecording = true;
}
```

3. Access to a list of fun and exciting templates-



Link to the video demo of the same - https://cdn.discordapp.com/attachments/1078051576284975226/1355217099265675346/stickman_update.mp4?ex=67ef6007&is=67ee0e87&hm=bccff6c52e7e16cdd19cf19baa9d85e6138bd96db61d639ec2c79ffab526b08b&

Whenever the template's button, 3rd from left, is clicked it opens up a templates tab which show a frame of the pose of the stickman and also the name of the template below.

The templates work like this--

```
const templates = {
    Running: [
        { time: 0, pose: { leftKnee: 130, leftFoot: 30, rightKnee: 50, rightFoot: -20, torso: -80,
        leftElbow: -110, leftWrist: -20, rightElbow: 90, rightWrist: 20, neck: 5, posX: 150, posY:
        200 } },
        { time: 0.5, pose: { leftKnee: 90, leftFoot: 10, rightKnee: 90, rightFoot: 0, torso: -85,
        leftElbow: -130, leftWrist: -10, rightElbow: 110, rightWrist: 10, neck: 0, posX: 200, posY:
        200 } },
        { time: 1, pose: { leftKnee: 50, leftFoot: -20, rightKnee: 130, rightFoot: 30, torso: -80,
        leftElbow: 90, leftWrist: 20, rightElbow: -110, rightWrist: -20, neck: 5, posX: 250, posY:
        200 } },
        { time: 1.5, pose: { leftKnee: 90, leftFoot: 0, rightKnee: 90, rightFoot: 10, torso: -85,
        leftElbow: 110, leftWrist: 10, rightElbow: -130, rightWrist: -10, neck: 0, posX: 300, posY:
        200 } },
    ],
    Dancing: [
        { time: 0, pose: { leftKnee: 90, leftFoot: 0, rightKnee: 130, rightFoot: 20, torso: -90,
        leftElbow: -30, leftWrist: 20, rightElbow: 30, rightWrist: 20, neck: 5, posX: 200, posY: 180 } },
        { time: 0.5, pose: { leftKnee: 130, leftFoot: 20, rightKnee: 90, rightFoot: 0, torso: -90,
        leftElbow: -90, leftWrist: 10, rightElbow: 90, rightWrist: 10, neck: 0, posX: 200, posY: 200 } },
        { time: 1, pose: { leftKnee: 90, leftFoot: 0, rightKnee: 130, rightFoot: 20, torso: -90,
        leftElbow: -30, leftWrist: 20, rightElbow: 30, rightWrist: 20, neck: -5, posX: 200, posY:
        180 } },
        { time: 1.5, pose: { leftKnee: 130, leftFoot: 20, rightKnee: 90, rightFoot: 0, torso: -90,
        leftElbow: -90, leftWrist: 10, rightElbow: 90, rightWrist: 10, neck: 0, posX: 200, posY: 200 } },
    ],
```

- **Structure**: Each template is an array of objects, where each object has:
    - time: A float representing the timestamp in the animation (e.g., 0, 0.5, 1).
    - pose: An object specifying the angles of each joint (e.g., leftKnee, torso) and the root position (posX, posY).
- **Purpose**: These define keyframes for an animation. For example, "Running" has four keyframes over 1.5 seconds, simulating a running motion.

  In summary the logic of accessing and playing the templates works like this
    - **Validation**: Checks if the template exists in templates.
    - **Reset**: Clears appState.recordedPoses to ensure only the new template’s poses are stored.
    - **Iteration**: Loops through each { time, pose } object in the template (e.g., templates["Running"]).
    - **Screenshot Generation**: Calls generatePoseScreenshot to render the stickman with the pose and capture a screenshot.
    - **Storage**: Adds an array [time, pose, shot] to appState.recordedPoses.
    - **Sorting**: Sorts appState.recordedPoses by time to maintain chronological order.
    - **UI Update**: Calls showRecoredPoses to reflect the updated poses in the "image-panel".

Features to be implemented from the ideas list in Ideas-2025.md

3d Human Body activity -

1. Implementing the shared mode for the doctor mode.

   The Doctor mode aims to play a **question-and-answer game**, currently the names of the players and leaderboard are present but they don't work, I will work on the shared mode for all the three models. Users will be able to use the **doctor mode for all the three models simultaneously and answer the questions which are being asked and they will get points for correct answers and accordingly, their scores will be calculated.**

2. Review of pop-ups and toolbar UI.

   Currently the toolbar contains almost everything it might need, there are buttons for every task the activity aims to achieve, but **lacks some pop ups,** I will add more pop-ups where we need them**, e.g.: when switching the modes from tour to paint, there will be a pop-up saying mode switched from Tour to Paint.** Pop-ups for switching between the 3 models, and more pop ups for different features, tour, doctor, etc.

3. Localize the activity.

   I will localize the activity using the i18next framework like I did in the Pawn activity. To ensure comprehensive localization, I will collaborate with French and Spanish speakers within the community to obtain accurate translations for their respective languages. Additionally, I will personally incorporate translation support for Hindi, thereby catering to a wider audience and enhancing accessibility for users from diverse linguistic backgrounds.

Stickman animation activity -

1. Share and collaborate.

   Similar to how the share mode of the 3d volume activity works, providing real time canvas update and sharing, the stickman animation activity aims to provide a common canvas for all the users, the users can see real time movement of the stickman of other players, right now the user can connect but can't really interact.

2. Import a photo of a human body to create a stickman in the same pose.

   This feature aims to provide a simple drag-and-drop feature where you can drop a picture of a human doing a certain pose, and the same pose will be replicated by the stickman. This will need a web-based ML framework such as Tensorflow.js, which will be added to detect the poses.

Other features that can be implemented

3d Human Body activity -

1. Functionality for users to learn more about the part they want.

   To implement this functionality, I will add a learn more button, in the toolbar, when clicked it will share short information on the selected part, e.g.: if the user selects skull, the information will be, “As brain is one of the most critical parts and weakest part of the body, we need to protect it with something hard and strong so the Skull is used to protect the brain from external damage and harm”

2. Functionality for users to see just that organ/ part

   Enabling users to be able to see only one single part instead of the whole model will give them a deeper understanding and more freedom over the parts, their textures and their size can be easily explored in this feature as the canvas will only hold a single element.

3. Storing the current view of the activity in the journal.

   I will add the functionality to store the current painted body parts and some details about the competition with other users, such as scores, winner, date, etc. Currently it lacks this implementation for the muscle model and the organs model.

4. Improve the multi-user presence functionality.

   I will add the multi-user presence for not just the doctor mode, but for all the other modes as well. currently, the shared mode works only for the skeleton model; by referring to the same, I will add it for the other models as well.

Stickman Animation activity –

1. Chat bubbles and pre-written messages.

   The feature of chat is already implemented in the Chat activity, so we don’t need to implement the chat activity here from scratch the main idea of chat bubble will be to convey small messages, e.g.: if I want to ask the other user to do a dance battle, I can’t, I don’t have a way to say that to him, implementing pre-build messages, like “Hi”, “let’s dance”, etc.. will make the activity more interactive and will take collaboration to a whole other level.

2. Similarity with the user's sugar logo

   As of now, the stickman’s are by default black in color, which won’t be a problem if I am alone and not on the server. But if I am connected the canvas will have multiple stickmen which will make it difficult to identify the user and their stickmen, so we

could either change the whole color of the stickman to match with the sugar logo, or simply the head of the stickman to match the logo. This will be super helpful in identifying the players.

3. Sharing of other users' poses.
   As the activity will be joined and collaborated on by multiple users, each one will have their creativity and ideas, If someone comes up with an impressive animation, they should be able to share it with the users. Right now, I have added the feature that stores the poses in a JSON file. The user should be able to share this file with their peers over the server. If selected, I would love to work on this feature in the coding period of GSOC.

Timeline

Pre GSoC period April 8 – April 29	Taking a deep dive into the work done till now and exploring the existing codebase of the 3d Volume activity to learn more about Three.js, useful in the Human body activity While simultaneously learning about Paper.js, from the official documentation and online resources, it will be heavily used in the Stickman Animator activity.
Community Bonding Period (May 8 – June 1)	During this period, I will interact with the mentors and the community and try to help other members as well. Asking for any features they would want to have in the activity. Deciding the best approach to handle events and exploring other features by discussing them with the mentors and the community.
Week 1 (June 2 – June 9)	Implement the shared mode for the doctor mode for all the three models, skeleton, muscles, and organs, simultaneously working on completing the leaderboard feature.

Week 2 (June 9 – June 15)	Checking if any glitches or bugs exist in the activity and taking feedback from the mentors. Provide the weekly progress report to the mentors.
Week 3 (June 16 – June 22)	Localizing the activity and releasing a partially completed activity, and asking community members for feedback, fixing any more bugs if they exist. Provide the weekly progress report to the mentors.
Week 4 (June 23 – June 29)	Starting on the stickman activity's key feature which is the share and collaborate, users being able to share a single canvas and getting real time updates on the position of other players. Finalizing the 3d human activity, by fixing all the bugs and glitches. Provide the weekly progress report to the mentors.
Week 5 (June 30 – July 6)	Working on the integration of Tensorflow in this activity to detect the poses from a human image uploaded to the activity. Arrange a demo session with the mentors to showcase the activity.

Week 6 (July 7 – July 13)	Testing both the activity. Fix bugs and corner cases within the activity for the midterm evaluation. Arrange a meeting with the mentors to ask for feedback before submitting the activity for the midterm evaluation.
Mid Term Evaluation (July 14)	Deliverables – Complete activity with all features mentioned in the Ideas-2025.md
Week 7 (July 15 – July 21)	Implement the features requested by the community during the community bonding period. Provide the weekly progress report to the mentors.
Week 8 (July 22 – July 28)	Feature to learn more about the selected body part in the 3D human activity, by giving a detailed and easy-to-understand text, mentioned in the other features to be implemented above Provide the weekly progress report to the mentors.
Week 9 (July 29 – August 4)	Working on the chat functionality with the help of chat bubbles in the Stickman Animator activity, also looking into matching the sugar logo's color with the head/ body of the stickman, for identification. Provide the weekly progress report to the mentors.

Week 10 (August 5 – August 11)	Release the activity for community engagement and soliciting feedback. Encourage active participation from the community to rigorously engage with the activity and identify potential errors. Conduct follow-up surveys to uncover any previously overlooked corner cases.
Week 11 (August 12 – August 18)	Fix the new corner cases and work on the feedback provided by the mentors and the community. Add the functionality for users to edit the environment. Provide the weekly progress report to the mentors.
Week 12 (August 19 – August 25)	Make the activity production ready. Engage with the mentors for the final time and seek feedback and incorporate any changes they wish to see. Localize the activity using the i18next framework.
End Term Evaluation	Deliverables – Fully developed activity incorporating both the initially proposed features and subsequent requests, polished to a production-ready state, and refined based on feedback from both the community and mentors.

Engagements during Summer

I plan to dedicate 30-40 hours per week to the project and will be most active between Friday and Sunday. I will have my End-semester exams between 6 May - 25 May i.e. the community bonding period, and will be able to dedicate 2-3 hours a day during that period, after that I will be having summer holidays and I have no prior engagement in summer, all of the summer will be dedicated to Sugarizer.

Highlight the work you plan to complete before each evaluation.

1. Mid Term Evaluation Deliverables - Complete activity with all features mentioned in the Ideas-2025.md.

2. End Term Evaluation Deliverables - Fully developed activity incorporating both the initially proposed features and additional requests, polished to a production-ready state, and refined based on feedback from both the community and mentors.

How will you report progress between evaluations?

I will be updating the mentors daily on Discord and Matrix chat and demonstrating my work through biweekly meetings. Additionally, I will write a blog every 2 weeks on about my progress and share it on my social media profiles.

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

After my completion of GSoC, my plan is to thoroughly review the issues and pull requests raised by other developers. In addition, I will explore new features to address. My primary goal is to add more functionality to the whole Human Activity Pack to make it more robust and feature-rich.

I am committed to maintaining the Human Activity Pack and adding more activities related to the activity, in the long term, and becoming a lifelong member of the Sugar Labs community to help others in the field of open-source.

Along with my work on the Human activity pack, I also plan to explore and contribute to other Sugar Labs projects, including the programming framework of Music Blocks v4, Sugar, and Music Blocks v3, after the GSoC period.

Conclusion:

Thank you for reading. I have provided a detailed overview of my project and how I plan to execute it. For GSoC 2025, my main goal is to further enhance my understanding of the project by building on my practical experience and research.

As for the technology, I am well-versed in all the necessary technologies required for this project. I have extensive experience working with Javascript and Blender, and other technologiesolutions. I am confident that I can complete this project within the given timeline and take full responsibility for implementing all the crucial and valuable features that will take the Human Activity Pack to the next level.

I am 100% dedicated to SugarLabs and have no plans whatsoever to submit a proposal to any other organization.