Collective Documentation

Aim

The aim of the project was to see **if immersive VR can improve mental well-being and cognitive ability of an individual**. We used **Candle Gazing in VR** as an experiment to check if it can help in meditation and attention span.

Objective

We were asked to work on developing the VR Headset, understand and work on experimentation and analysis. Our objective was to make sure that the focal length and other aspects of the VR Headset were ideal to perform the experimentation.

Approach

Initially, we were asked to work on the focal length of the VR Headset – a google cardboard readymade template was used to make the VR Headset.

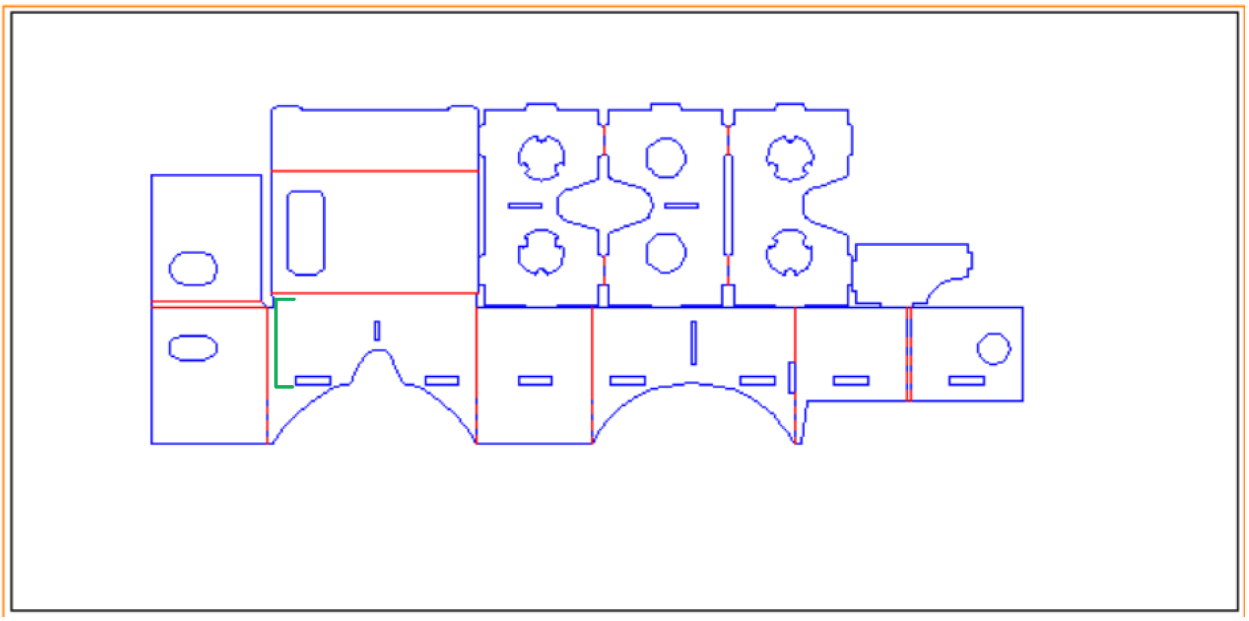
We laser cut the readily available template and started scraping off the parts to help us get the right focal length. We also worked on the scene, by giving inputs and attempting to improve the VR Environment by making it more immersive for meditation.

We wanted to test out this experiment with various combinations of questions and before/after activities to understand its impact on cognitive ability and attention span.

Process of making the prototype

The problem with the initial prototype was the focus. We checked by moving the screen closer and farther to the lens, we saw a difference in the focus. Then we decided to adjust the template so that the scene we see is adjusted and more focussed.

Initial template:

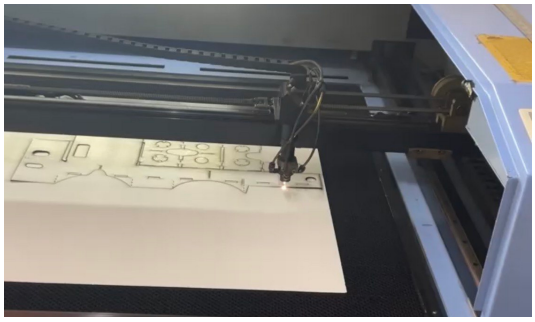
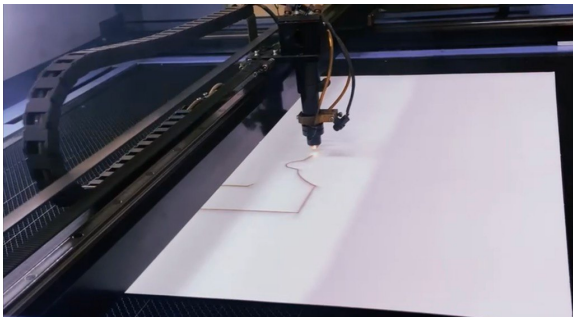


The highlighted green colour is the distance between the lens and the screen

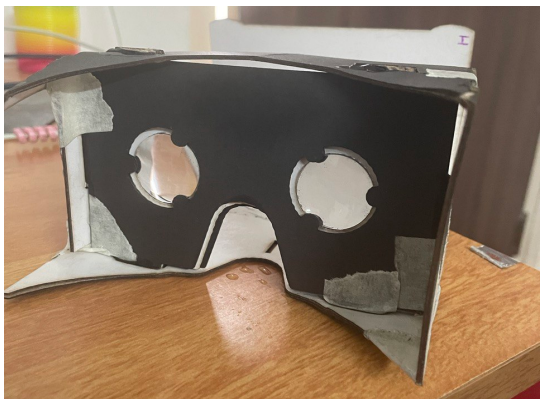
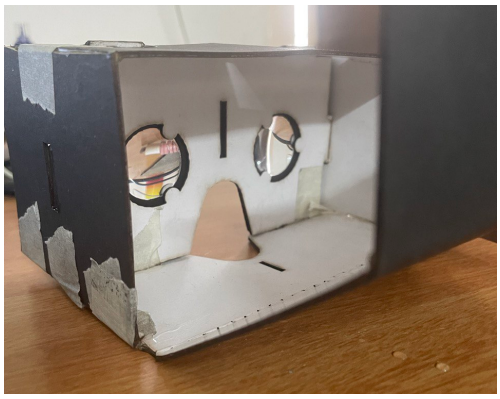
The distance here is adjusted in the template. We cut the template using laser cutter

Material Used: Mill Board
Colour commands

1. Blue - Cut
2. Red- Perforation

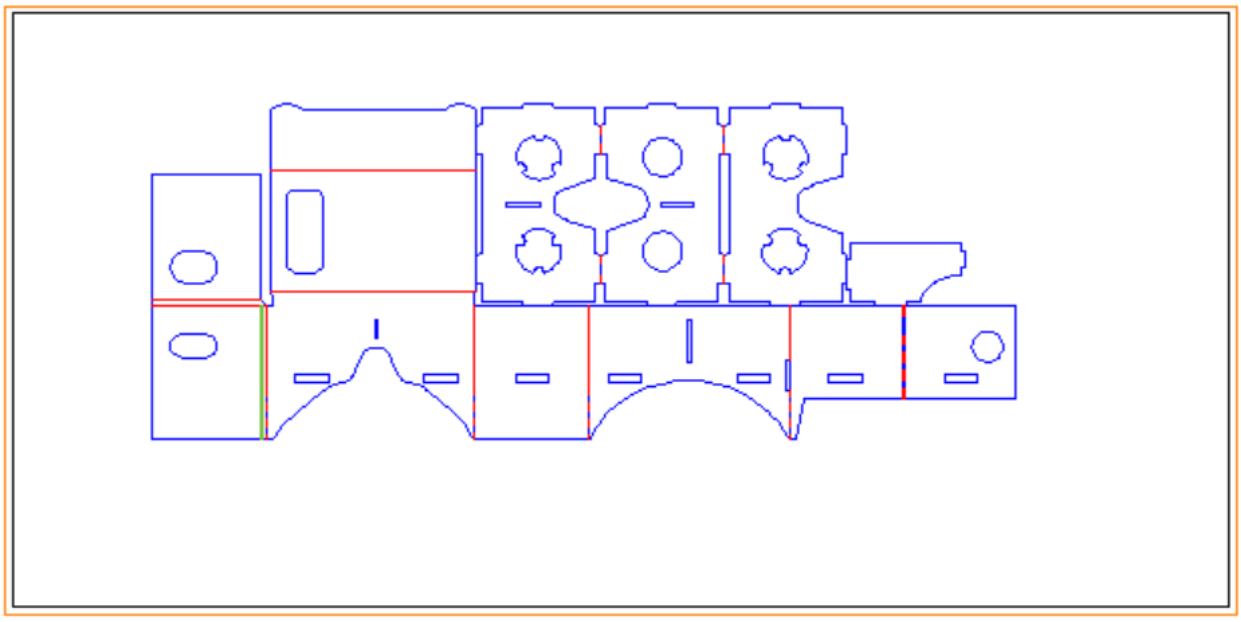
We made 4 cutouts and assembled it each one us with different distances from the lens.

Assembled VR cardboard

With the most accurate headset, we measured the distance with vernier calipers and decided to cut more of the same.

Final Template:



Final template with measurements from vernier callipers

With these measurements and scaling we made three templates and cut them out with a laser cutter. Material used here is also the same i.e, Mill Board.

We assembled and tested these 3 headsets. Labelled as 1,2,3.

Headset 1,2,3

These three headsets are tested with the candle gazing scene. Most feasible and accurate focus was found in headsets 2,3.

With this template we will be making a final accurate prototype with MDF board as the Mill board is not durable.

All the iterations made

Why VR Environment?

Potential benefits and assumptions of candle gazing in VR:

- VR can provide a more immersive nature of VR , which can help users to focus on the candle flame and block out distractions. This can lead to a deeper and more meaningful experience.

- The gentle flickering of the candle flame can be very calming and relaxing. VR can help to amplify this effect, creating a truly peaceful and tranquil environment.

- The quiet and focused atmosphere of a VR candle gazing environment can be ideal for self-reflection and introspection. Users may find themselves pondering their thoughts, feelings, and experiences in a new and deeper way.

- The flickering candle flame can evoke a sense of connection to nature and the natural world. VR can help to amplify this feeling, creating the illusion of sitting by a campfire in the wilderness.

- VR can block out distractions from the real world, which can help participants to focus on the candle flame.

- VR could be used to provide participants with feedback on their candle gazing practice, such as how long they are able to focus on the candle flame without their mind wandering

Design considerations for the scene:

- Creating a dark and quiet environment to get the most out of the experience. It is important to create a dark and quiet environment. This will help users to focus on the candle flame and avoid distractions.

- Using a realistic candle model that has a flickering flame and realistic lighting effects.

- Added other elements of nature to create a more immersive experience.

- The environment was iterated multiple times with modifications to flickering flame , contrast and distance to provide a stress free and relaxed experience .

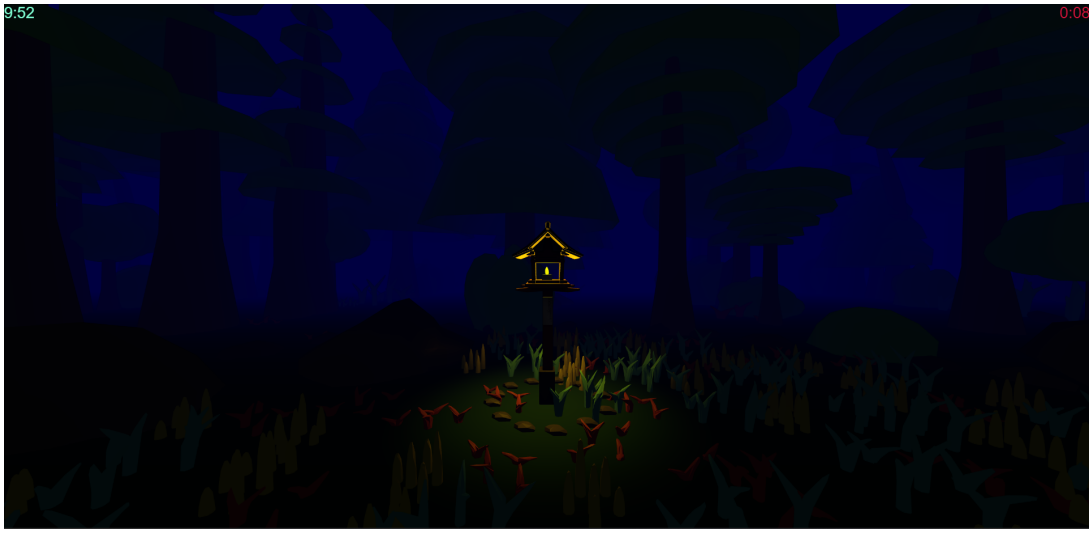


fig. VR scene- Candle Gazing Experiment

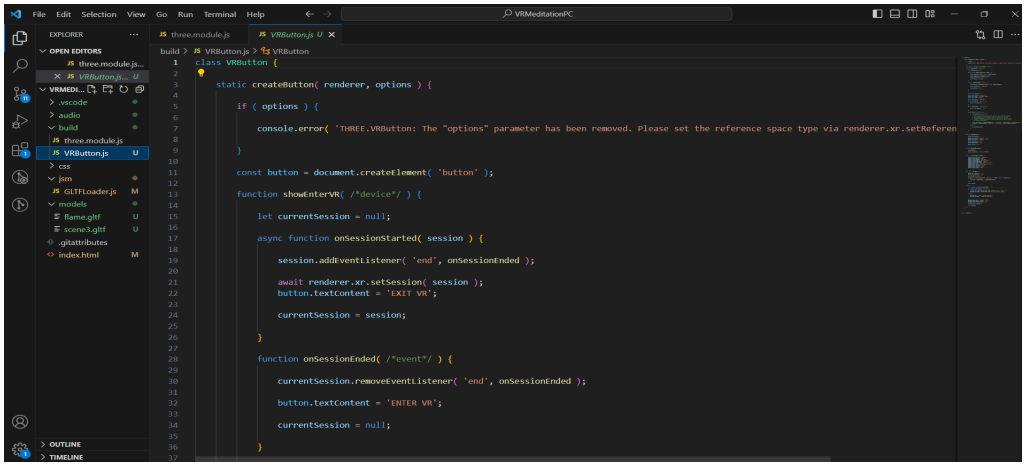


fig. Code snippet of the VR experience - Candle Gazing Experiment

Experiments

Moving forward, our objective was to evaluate the prototypes and determine their functionality. To achieve this, we conducted a comparative study to ascertain whether the Candle gazing activity was more effective in an Immersive VR environment or in a physical setting where a real candle was used for the activity. In order to obtain reliable and substantial data to validate our experiments, we used an EEG (Electroencephalography) Reader to collect raw data.

An EEG machine is a device that records the electrical activity of the brain. It comprises electrodes that, when positioned on an individual's scalp, can capture, and monitor brain activity. These electrodes register the brain's wave patterns, and the EEG device transmits this data to a computer or a cloud server.

We utilised the Brain wave visualizer app to record the data transmitted from the EEG device for our comparative study, focusing on two key metrics: attention and meditation span.
Initially, we conducted a pilot experiment in the VR environment to gain insights into various elements. In this experiment, participants were instructed to gaze for 2 minutes, and their data was recorded using the EEG reader, which was then processed by the brain wave visualizer app.

Subsequently, participants were asked to repeat the same procedure in a real-world environment, with an actual candle, and their data was captured again.

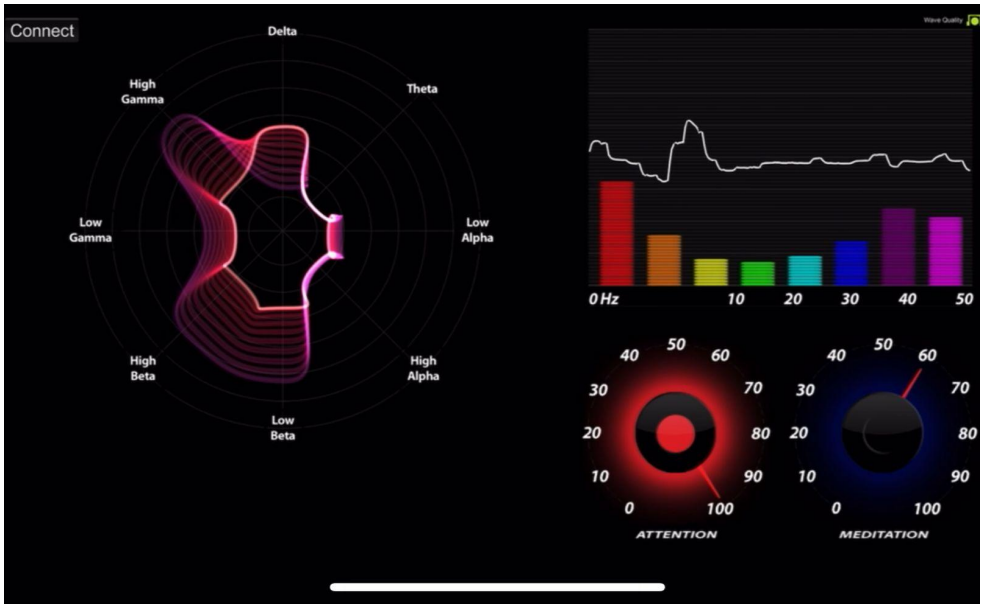


Fig (1)

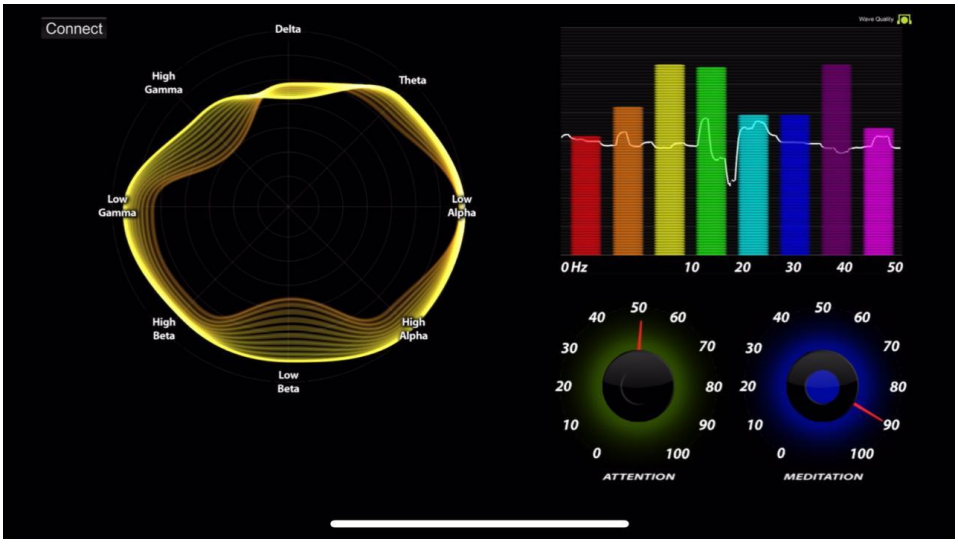


Fig (2)

The figure 1 is the data captured while the user performed the activity through the VR candle gazing environment and the figure 2 represents the data captured while the same activity was done in the real-world environment.

The data captured were:

- ➢ Attention span
- ➢ Meditation level
- ➢ Delta
- ➢ Theta

- ➢ Low Alpha
- ➢ High Alpha
- ➢ Low Beta
- ➢ High Beta
- ➢ Low Gamma
- ➢ High Gamma

As you can see the attention span was higher in the VR environment, but the meditation level was comparatively low. But the same activity done in the real-world environment depicted a contrast reading where the attention span was low, and the mediation level was positively higher. We recorded the entire two minute readings, and we were able to see that the Attention span hit the range of 100 multiple times in the VR environment whereas the same was low in a real-world environment.

The primary goals of this pilot experiment were twofold:

1.To get the raw data from the data captured by the EEG reader to study the experiment.
2.To get the user feedback in order to make it more user friendly and find the area of improvement both in the hardware and software aspects.

Observations

1. This experiment served as a means for us to gain insight into how users experienced the candle gazing activity and to determine whether there was any noticeable change in their state of mind or a sense of relaxation before and after the activity. Specifically, we aimed to compare the impact of candle gazing in a VR environment to that in a real-world setting.

2. The feedback we received from participants following the experiments proved to be invaluable. For instance, one participant noted that they experienced blind spots after the real-world experiment, a phenomenon that was not observed during their VR experience.
3. Another participant mentioned that they encountered difficulty concentrating in a real-world environment due to various external distractions. As this user pointed out, we were able to detect a decline in their attention span whenever they became distracted by the surrounding noise.

Future scope of improvement

VR Environment

After realising even with the nearly perfect focal length , the experience varied from one phone to another . The clarity and experience was better on the iphone due to high resolution . After thinking it over , we decided that we could change the VR environment .Hence in order to develop an enhanced three.js template , we worked on a basic solar system environment to understand the fundamentals of Three.js.

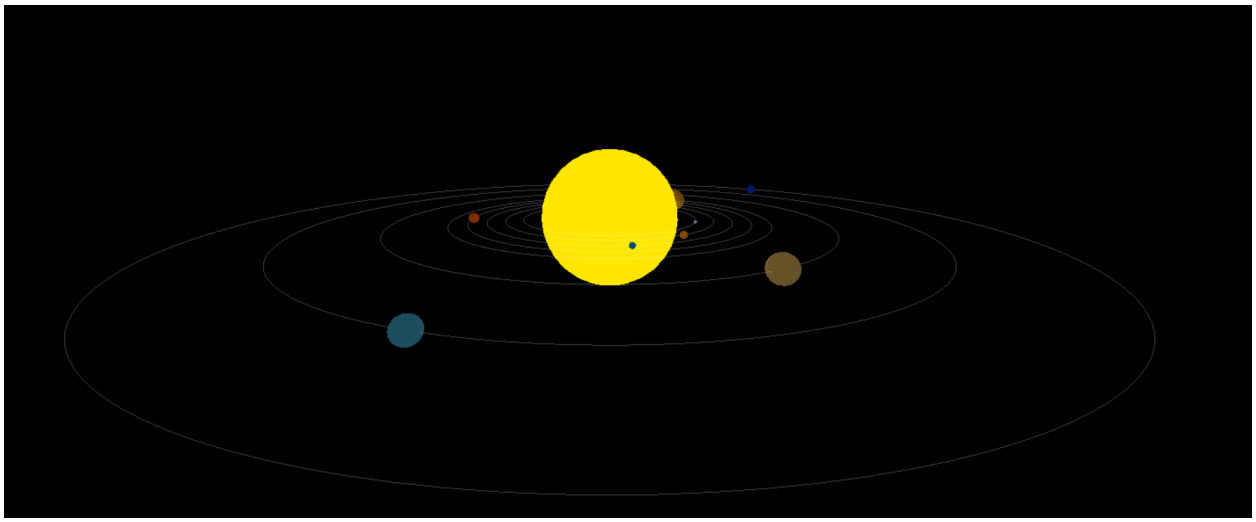
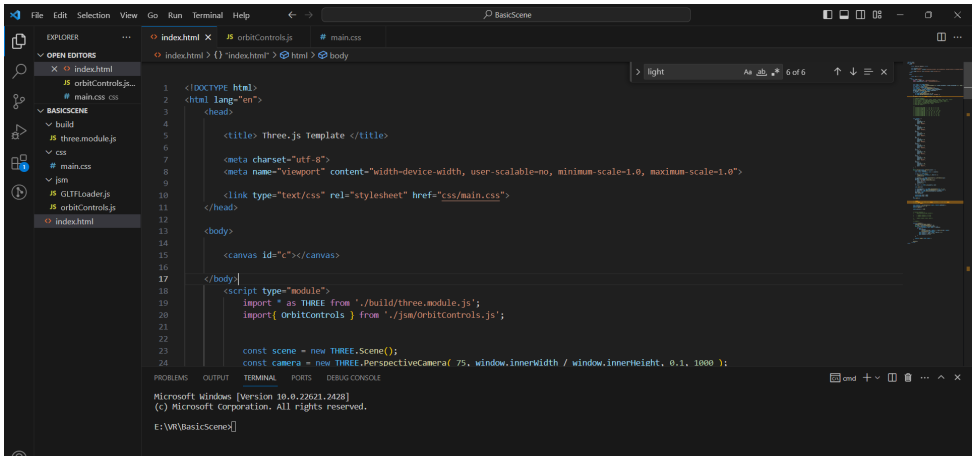


fig. VR scene- Solar system basic scene

This is a Three.js template for creating a simple solar system scene. It includes a sun, eight planets, and their orbits.
The code begins by importing the necessary libraries, including Three.js and OrbitControls.js. OrbitControls.js is used to allow the user to rotate and zoom around the scene.
Next, the scene and camera are created. The scene is where the objects in the solar system will be placed. The camera is used to view the scene.
Later , various components such as clock , canvas etc . are declared

The sun is then created and added to the scene. The sun is a sphere with a material that emits light.
Next, the planets are created. Each planet is a sphere with a different material and orbit radius. The planets are added to the scene and their orbits are added as well.
Finally, the controls for the scene are created. This allows the user to rotate and zoom around the scene.
The code then enters a loop where the scene is rendered and the controls are updated. This loop is necessary to keep the scene animating.



Experimentation

Right now, we are able to use the VR Headset and gather data from the EEG that gives us information about the Attention Span and Meditation of the user. Full-fledged actual tests and controlled experiments are yet to be conducted.

We can bring in other parameters to measure other aspects of the cognitive ability. We can introduce puzzles/activities that induce flow in one's cognition before and after the VR Meditation to see if it actually works.

We can ask users to use this for a set period of time that will get them habituated to this method of meditation and then check their before and after – attention span, cognitive ability and their experience of meditation in VR.

We can primarily test this out in **children and adolescents** – as we have backed up literature review which says that children's attention span is deteriorating and needs intervention in the formative years.Therefore, we can check this in children to see if it does bring a difference in their attention span and cognitive ability.

Right now, the VR Headset has to be held by hand – this is a little strenuous as the user has to hold it the entire time while meditating. Hence, an adjustable strap can be attached to it. This can be made out of a nylon strap material (used in bags for adjusting).

Individual Contribution and Reflective Note

My initial line of inquiry was to test " The role of minimalist puzzles in enhancing cognitive thinking".

Plan of Action

To test the role of minimalist puzzles in enhancing cognitive thinking,

- Recruit participants.
- Have the participants complete a series of cognitive tests.
- Provide the participants with minimalist puzzles to complete on a regular basis for a period of time.
- After the intervention period, have the participants complete the same series of cognitive tests again.
- Compare the participants' performance on the cognitive tests before and after the intervention period.

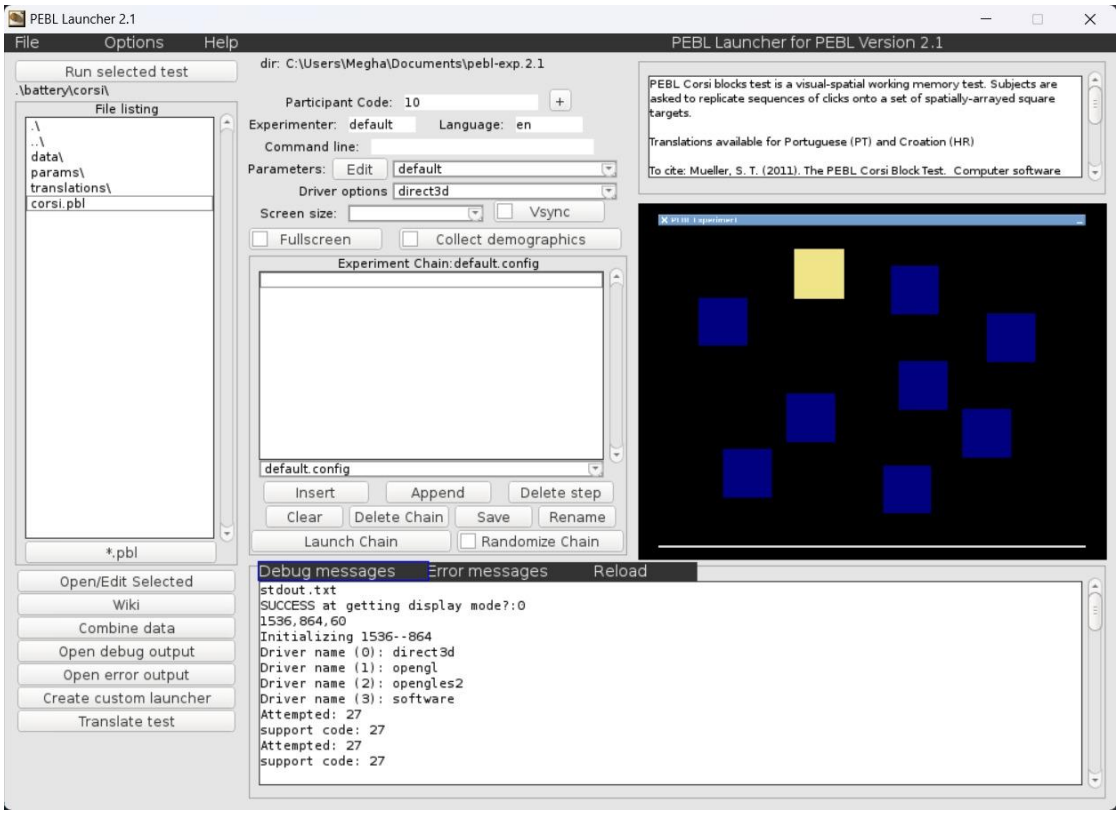
If the participants perform better on the cognitive tests after the intervention period, this would suggest that minimalist puzzles can enhance cognitive thinking.

Pilot

I asked the participants to take up the cognitive test

1. *Corsi Block Tapping Test (CBTT)* :
   It can help in analyzing :
   - *Visual spatial reasoning*: The CBTT requires the participant to visually encode the spatial location of the blocks. This requires the participant to be able to understand and reason about spatial relationships.
   - *Memory*: The CBTT requires the participant to maintain the spatial sequence of the blocks in working memory. This requires the participant to be able to hold information in mind for a short period of time.
   - *Motor skills*: The CBTT requires the participant to execute the correct motor sequence to tap the blocks. This requires the participant to have good motor coordination and control.

The test was conducted in hybrid mode wherein some of them took up the test in classroom environment and some took up the test virtually using anydesk . The software used to run the test was PEBL software.



Results

Sl. no.	Initials	Block Span	Total score	Total Correct Trials	Memory Span
1	RU	6	54	9	5.5
2	LA	6	42	7	4.5
3	DE	5	40	8	5
4	ME	8	104	13	7.5
5	SH	6	54	9	5.5
6	NI	6	54	9	5.5
7	HA	8	88	11	6.5
8	MA			10	6
9	VA	8	88	11	6.5

Here , I got the Memory span values.

Collective group project

As a collective group project , we worked on how VR is impacting attention span and testing if meditating in VR is more beneficial or not.

Designing the working Prototype

To test the attention span in the VR environment , the group project involved an already existing VR environment with a candle gazing experiment . I had to create a number of iterations of Google glasses with a mount board to get a perfect focal length so that the VR is not straining the user's eyes . I created a few prototypes until the focal length was perfect .

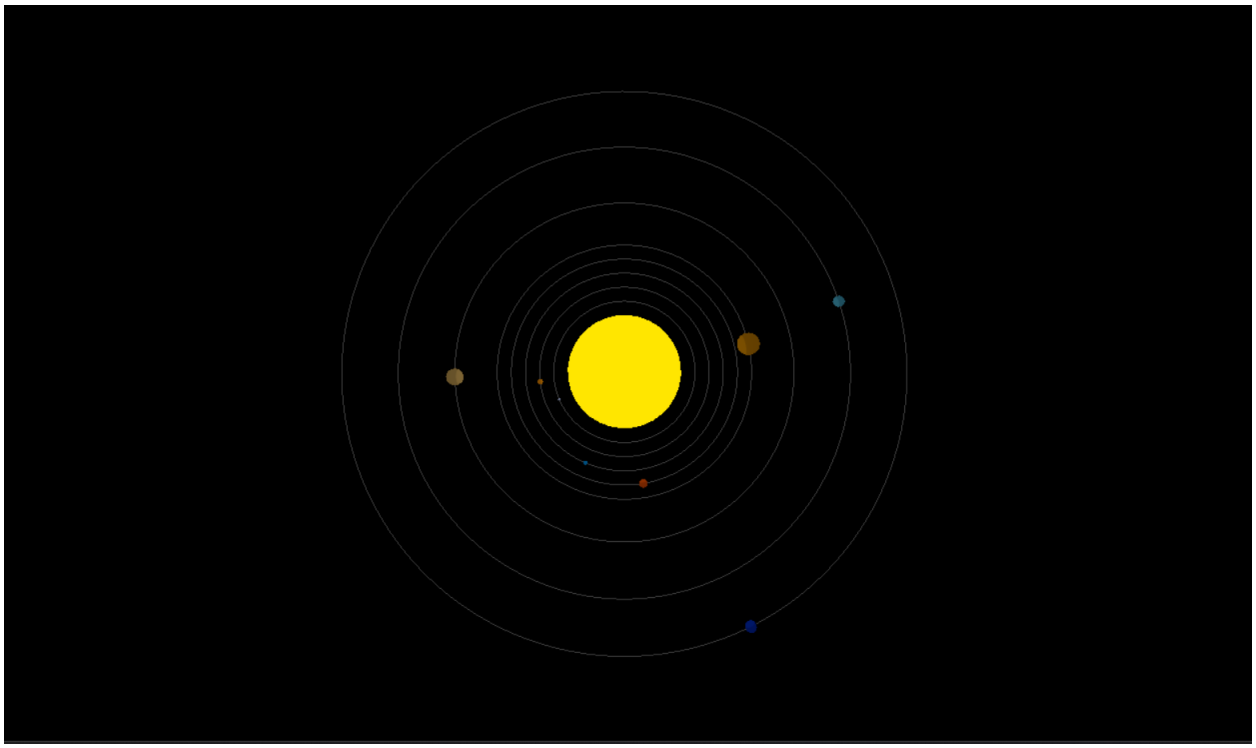
I then realized even with the nearly perfect focal length , the experience varied from one phone to another . The clarity and experience was better on the iphone due to high resolution . After thinking through , we decided that we could change the VR environment . So , I started learning three.js to create a new environment .

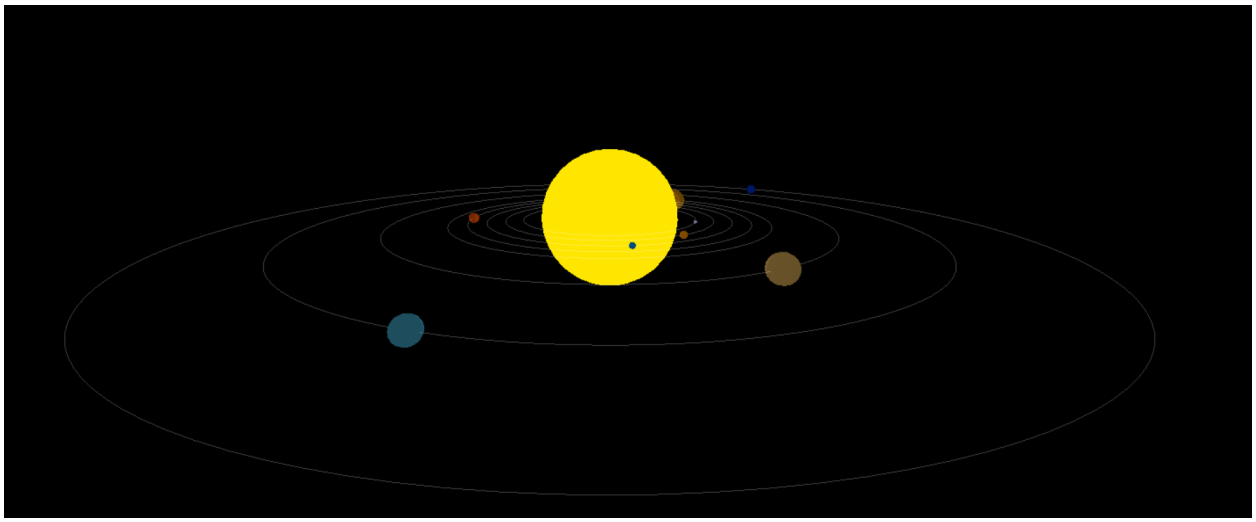
Hardware and Software that I have used :

- Mill Board
- Laser Cutting Machine
- Adobe Illustrator
- Visual Studio Code
- Node.js
- Vite
- Three.js
- Lens
- VR headset template
- EEG Headset

My Three.js basic scene development

To start with Three.js , I went through the Three.js documentation and set up a project . The default project that Three.js has is a rotating cube . Hence , it flashed to me to create a solar system with planets rotating and revolving around the sun .





Three js environment enhancement (focal length)

I created a Three.js template for a simple solar system scene. It includes a sun, eight planets, and their orbits.
The code begins by importing the necessary libraries, including Three.js and OrbitControls.js. OrbitControls.js is used to allow the user to rotate and zoom around the scene.
Next, the scene and camera are created. The scene is where the objects in the solar system will be placed. The camera is used to view the scene.
Later , various components such as clock , canvas etc . are declared

The sun is then created and added to the scene. The sun is a sphere with a material that emits light.
Next, the planets are created. Each planet is a sphere with a different material and orbit radius. The planets are added to the scene and their orbits are added as well.
Finally, the controls for the scene are created. This allows the user to rotate and zoom around the scene.
The code then enters a loop where the scene is rendered and the controls are updated. This loop is necessary to keep the scene animating.

Here is a brief explanation of the key parts of the code:

```
<canvas id="c"></canvas>
```

This creates a canvas element that will be used to render the scene.

```
const scene = new THREE.Scene();
```

This creates a new scene object. The scene is where the objects in the solar system will be placed.

```
const camera = new THREE.PerspectiveCamera( 75, window.innerWidth / window.innerHeight, 0.1, 1000 );
```

This creates a new perspective camera object. The camera is used to view the scene.

```
const renderer  = new THREE.WebGLRenderer({canvas});
```

This creates a new WebGL renderer object. The renderer is used to render the scene to the canvas element.

```
// **CREATING SPHERES**
            // function createSphere(color ,width , height , length , posX , posY , posZ){
              // const geometry = new THREE.SphereGeometry(width, height, length );
             // const material = new THREE.MeshPhysicalMaterial( { color: color } );
            // const sphere = new THREE.Mesh( geometry, material );
            // sphere.position.set(posX, posY , posZ)
            // scene.add( sphere );

            // }
            // createSphere(0xffe600 , 1 , 16 , 16 , 0 , 0 , 0);
            // createSphere(0x00ff00 , 1 , 16 , 16 , 5 , 5 , 0);
            // createSphere(0x00ff00 , 2 , 16 , 16 , 4 , 2 , 0);
            // createSphere(0xffff00 , 1.5 , 16 , 16 , 3 , 10 , 0);
            // createSphere(0x00ff00 , 3 , 16 , 16 , 10 , 15 , 0);
            // createSphere(0x00ff00 , 8 , 16 , 16 , 14 , 2 , 0);
            // createSphere(0x00ff00 , 6 , 16 , 16 , 8 , 5 , 0);
            // createSphere(0x00ff00 , 4 , 16 , 16 , 7 , 12 , 0);
            // createSphere(0x00ff00 , 4 , 16 , 16 , 17 , 8 , 0);
```

This is a code snippet where in I tried to create spheres (planets and sun) in a single function and calling the function to give values for the various parameters like position X , position Y , position Z , color , width , height and length .

```
const sun = new THREE.Mesh(
                new THREE.SphereGeometry(40, 32, 16),
                new THREE.MeshBasicMaterial({ color: '#FFE600' })
```

```
            )
```

This creates a Sun object where we are calling SphereGeometry and MeshBasicMaterial inside Mesh . The SphereGeometry gives the dimensions to the sun and the MeshBasicMaterial gives its color .

```
sun.add(new THREE.PointLight(0xdddddd, 24.0, 1400, 0.7))
```

This adds a point light to the sun. The point light will illuminate the rest of the scene.

```
const planets = {
                mercury: {
                    size: 1,
                    orbitRadius: 50,
                    speed: 100,
                    color: '#C2D0FF',
                },
                venus: {
                    size: 2,
                    orbitRadius: 60,
                    speed: 73,
                    color: '#FF9900',
                },
                earth: {
                    size: 1.5,
                    orbitRadius: 70,
                    speed: 62,
                    color: '#00A3FF',
                },
                mars: {
                    size: 3,
                    orbitRadius: 80,
                    speed: 50,
                    color: '#FF5C00',
                },
                jupiter: {
                    size: 8,
                    orbitRadius: 90,
                    speed: 27,
                    color: '#E59700',
                },
                saturn: {
```

```
                size: 6,
                orbitRadius: 120,
                speed: 20,
                color: '#FFCF72',
            },
            uranus: {
                size: 4,
                orbitRadius: 160,
                speed: 14,
                color: '#61D9FF',
            },
            neptune: {
                size: 4,
                orbitRadius: 200,
                speed: 11,
                color: '#0047FF',
            }
        }
```

This creates an object that contains the information such as size , orbit radius , speed and color of the planet for each planet in the solar system.

```
Object.entries(planets).forEach((planet) => {
            const name = planet[0]
            const { size, orbitRadius, color } = planet[1]

            let pts = new THREE.Path()
                .absarc(0, 0, orbitRadius, 0, Math.PI * 2)
                .getPoints(90)

                                                let   geometry   =   new
THREE.BufferGeometry().setFromPoints(pts)
            geometry.rotateX(Math.PI * 0.5)
            let mesh = new THREE.LineBasicMaterial({
                color: '#ffffff',
                transparent: true,
                opacity: 0.25
            })
            let orbit = new THREE.Line(geometry, mesh)
```

```
                scene.add(orbit)

                   let planetGeometry = new THREE.SphereGeometry(size, 16,
16)
                let planetMesh = new THREE.MeshToonMaterial({ color: color
})
                let pmesh = new THREE.Mesh(planetGeometry, planetMesh)
                pmesh.name = name
                orbit.add(pmesh)

                planets[name].mesh = pmesh
                planets[name].line = orbit
            })
```

This iterates over the planets object and creates a sphere and orbit for each planet. The sphere is added to the scene and the orbit is added to the planet.

```
const controls = new OrbitControls( camera, renderer.domElement );
```

This creates a new OrbitControls object. OrbitControls.js is used to allow the user to rotate and zoom around the scene.

```
function animate() {
                requestAnimationFrame( animate );
                let time = clock.getElapsedTime() / 50
                Object.entries(planets).forEach((planet) => {
                    const { orbitRadius, mesh, speed } = planet[1]
                    if (mesh) {
                        mesh.position
                            .set(Math.cos(time * speed), 0, -Math.sin(time
* speed))
                            .multiplyScalar(orbitRadius)
                        mesh.rotation.y = time * speed - Math.PI * 0.5
                        mesh.rotation.z = Math.PI * 0.5
                        mesh.rotation.x += 0.01
                    }
                })

                renderer.render( scene, camera );
            }
```

This function is used to animate the scene. The function is called repeatedly and it updates the positions of the planets.

```
requestAnimationFrame( animate );
```

This schedules the animate function to be called at the next animation frame. This ensures that the scene is animated smoothly.

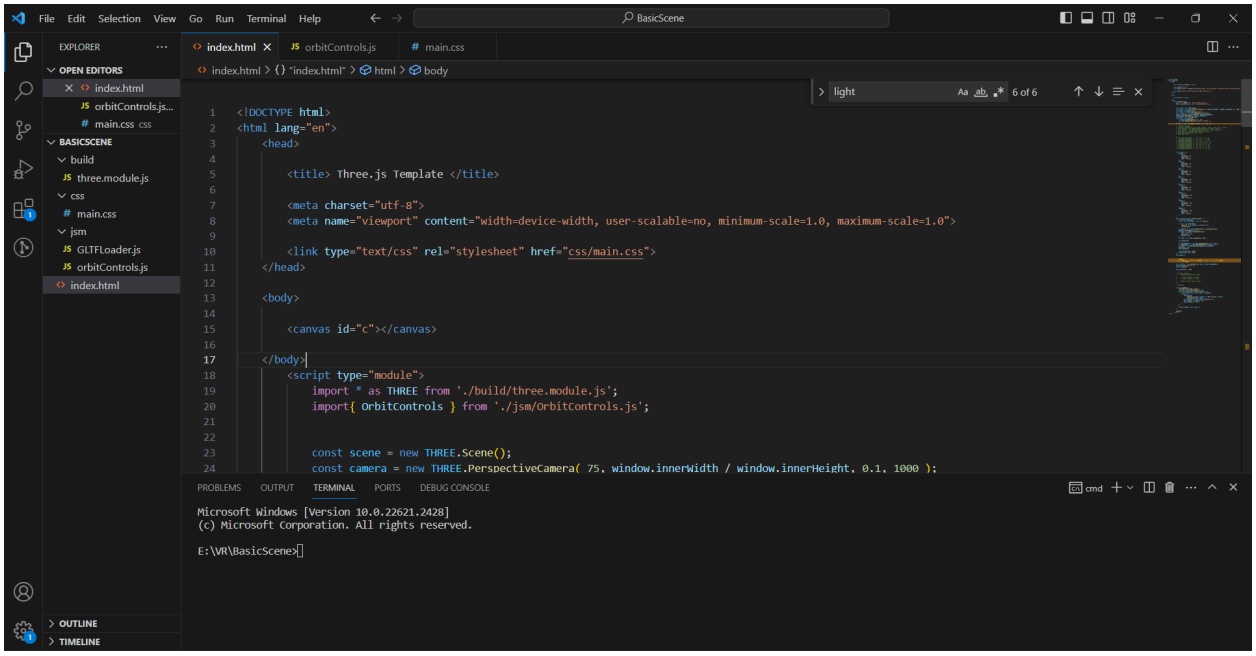


fig . Code snippet of solar system environment

Pilot Experiment for the Prototype

We did a pilot experiment round using the candle gazing VR experience , the nearest to perfect VR headset prototype and an EEG to get values for Attention , meditation , High alpha, Low alpha , High beta , Low beta , High gama , Low gama , delta , theta etc .

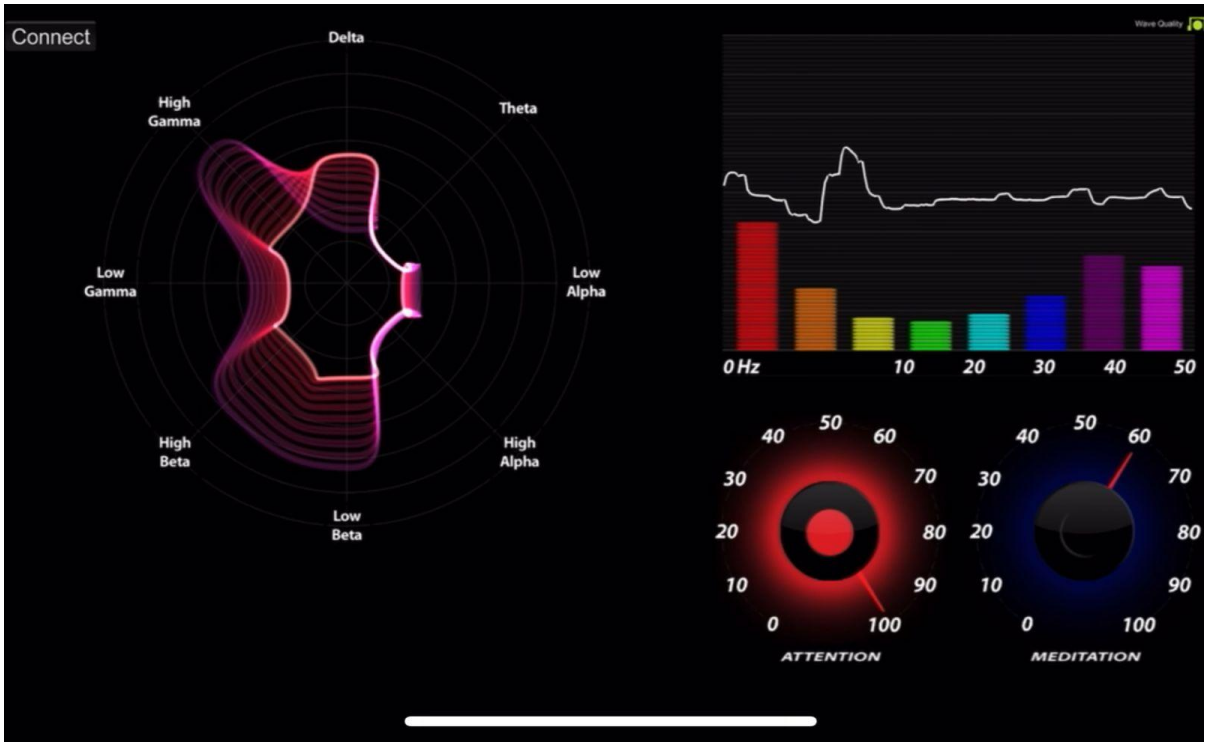


fig . Brainwave Visualizer app

The volunteer (Me) wore EEG headgear properly and the EEG was connected to the Brainwave Visualizer app in the phone via bluetooth . Brainwave Visualizer is an app which gives the readings of the experiment . The experiment was to gaze at the candle in VR for 2 minutes while wearing the EEG headgear and record the results . Later , the experiment was conducted with a similar setup but now in reality . A real candle was asked to be gazed at for 2 minutes while wearing the EEG headgear and record the readings . After the experiment was conducted , We could get the values for Attention .

fig . I am undergoing a real candle gazing experiment

We could also observe that the attention was high in VR compared to real experience while on the other hand , the meditation meter showed that Meditation readings were higher in the real experience when compared to the VR environment .

Work in Progress to find the relation between attention span and memory span

I got the results of Memory span through the pilot experiment that I conducted using the *Corsi Block Tapping Test (CBTT)* , and I got the readings for Attention using the EEG .

My future scope of interest with this is to consider giving the puzzles as a task to test various cognitive abilities before and after the VR intervention .
Things like , *has the attention span enhanced or not ?* can be tested .

fig . Attention meter readings

Now , Considering My plan of action with my initial line of inquiry “role of minimalist puzzles in enhancing cognitive thinking “ was to provide Puzzles itself as the intervention and do a before and after intervention tests using cognitive tests such as *Corsi Block Tapping Test (CBTT)* .
Here ,things like , *is there any enhancement of memory span after puzzle as intervention* ? can be tested .

Memory Span
5.5
4.5
5
7.5
5.5
5.5
6.5
6
6.5

fig . Memory span results

So , on a broader note , I am also trying to find the connection between **Attention span** and **Memory span** and find out more information after analysis .

Future work focus :

- How do working memory and attention span interact?
- How can attention span training improve memory span?
- How can memory training improve attention span?
- What is the relationship between attention span and memory span?