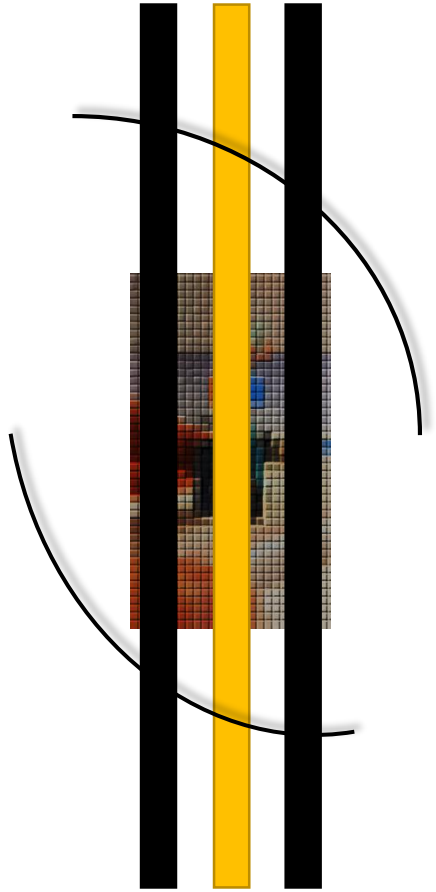
# Ontologies for Spatial Interactions



**Supervisor**:
Vertr.-Prof. Jason Reizner

**Author:**
Zeinab Rahimi

MediaArchitecture
Winter 2023

# CONTENT :

- Virtual Reality (VR), Augmented reality (AR)
- Mixed Reality
- The use of MR and AR
- The Aim
- The Schematic view of the Prototype
- Tools
- The Result and The Code
- References

# Virtual Reality (VR), Augmented reality (AR)

Virtual Reality (VR) is generally perceived as an experience in which the observer is fully immersed in and able to interact with a completely synthetic environment. However, we concentrate in this project on augmented reality(AR), which is a subclass of VR related technologies that merges the real and virtual worlds.

*"Augmented reality (AR) is a technology that allows users to view virtual objects within a real-world environment in real-time. In contrast to virtual reality(VR), where the user and objects are fully integrated in the virtual world, AR systems augment computer-generated virtual images into the physical world. AR systems also have the capacity to allow physical objects in real-world environments to interact with virtual objects***[1]***."*

The three primary characteristics of an AR system, according to Azuma **[2**]are:

1. Combines virtual and real content.
2. Registered in 3D
3. Interactive in real time.

*"Augmented reality sits within the mixed reality(MR) space that connects the real world and the virtual world, as shown in* ***Figure1****. Augmented reality is defined as a system in which a real environment is augmented by virtual content* ***[3],*** *where augmented virtuality is defined as synthetic or virtual world in which the user can interact with both virtual and/or real objects in the same environment***[1]."**

# Mixed Reality

In 1994**, Mixed Reality (MR)** was defined by Milgram and co.(Figure 1) to be a combination of Augmented reality and Augmented virtuality, where the real-world environment was presented with a virtual world. However, this definition only concentrates on the computer graphics part of mixed reality**[3].**

**"***This definition is incomplete nowadays, as MR has advanced since then to include a variety of human-computer interaction methods. Currently, mixed reality applications make use of environmental, spatial sound, gestures, sensory and physical inputs to create an experience for users***[4].**"

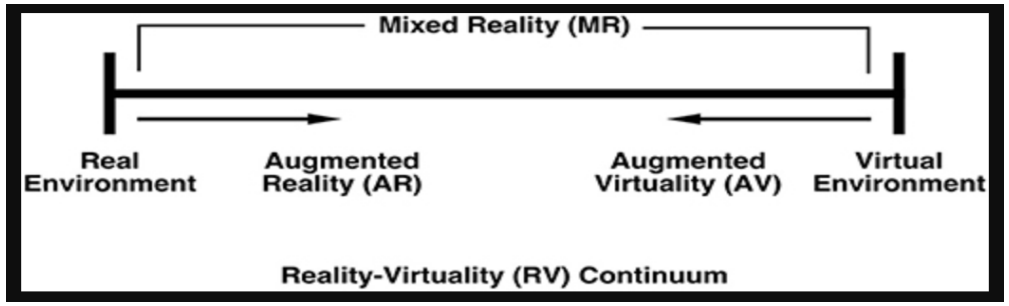



Figure 1. Milgram and Kishino's reality-virtuality continuum (adapted from Milgram et al., 1994).

It is evident that the difference between mixed reality and augmented reality is sometimes unclear**[*4*]**.

In this project, augmented reality refers to virtual objects placed in a physical environment with little or no interaction with them. While mixed reality is where virtual objects placed in the physical users' environment can be interacted with using a variety of input methods.

# The use of MR and AR

There is an infinite number of possible applications for mixed reality. Given the correct set of capabilities, mixed reality can be used in any field for simulation, teaching, practice and much more[1]

The use of AR has been investigated across numerous fields, including medicine: surgical planning [4], and training [5], and education: in teaching geometry [6], teaching civil engineering [7].Also, it has been successfully applied to both mathematics and geometry lessons to provide students with an immersive educational environment [8].In the field of medicine, AR technology has been used to support the teaching and training of medical students, and to plan/perform surgeries[9].

# The Aim:

*"Although augmented reality technology has been used across a diverse range of fields, it is not yet widespread consumer product. One key consideration in the mainstream use of AR technology is the need for an appropriate AR user interface that will allow end-users to interact intuitively and seamlessly with virtual content. There are numerous interaction interfaces, including those that use gesture-based interaction, tangible interaction, and multimodal interaction interfaces. However, these interaction interfaces continue to develop and evolve*[*1*]."

There are three principal elements to augmented reality interactions that contribute to the enhancement of the user experience(Figure 2)[1]:

1)The user,

2)The user interface,

3) The virtual content

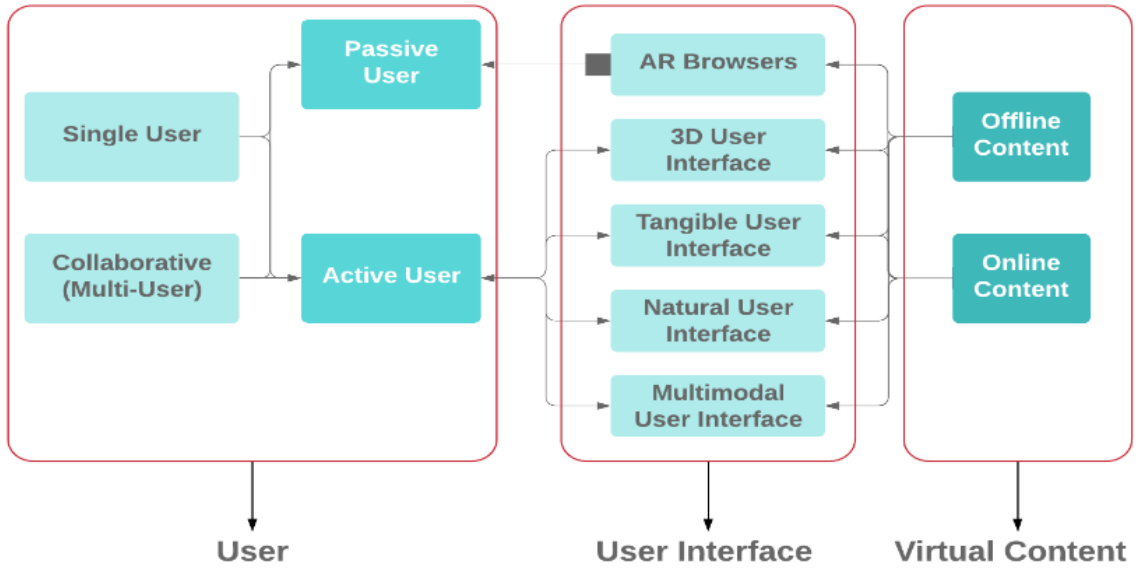



**Figure 2. Augmented reality interaction aspects** [1]

In this project, we will practice developing an interactive tool that allows users to manipulate computer-generated 3D Objects/Models that are developed in a web-based Augmented Reality environment with physical and tangible tools; the IOT device (Arduino Uno) and mouse will be used here.

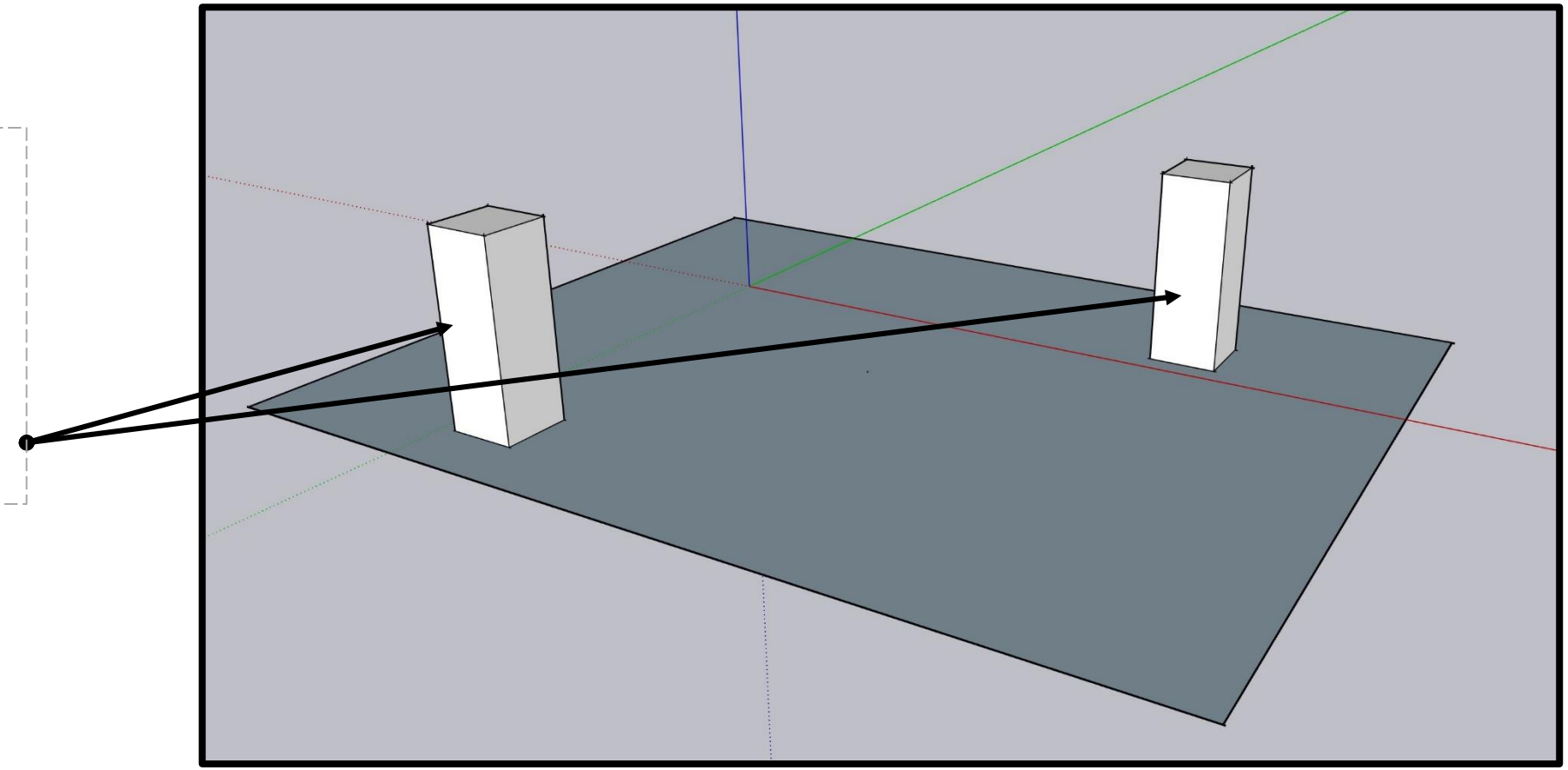
# What kind of interaction?

The target interactions in the scope of this project and the user interface to perform that interaction is listed in table 1.

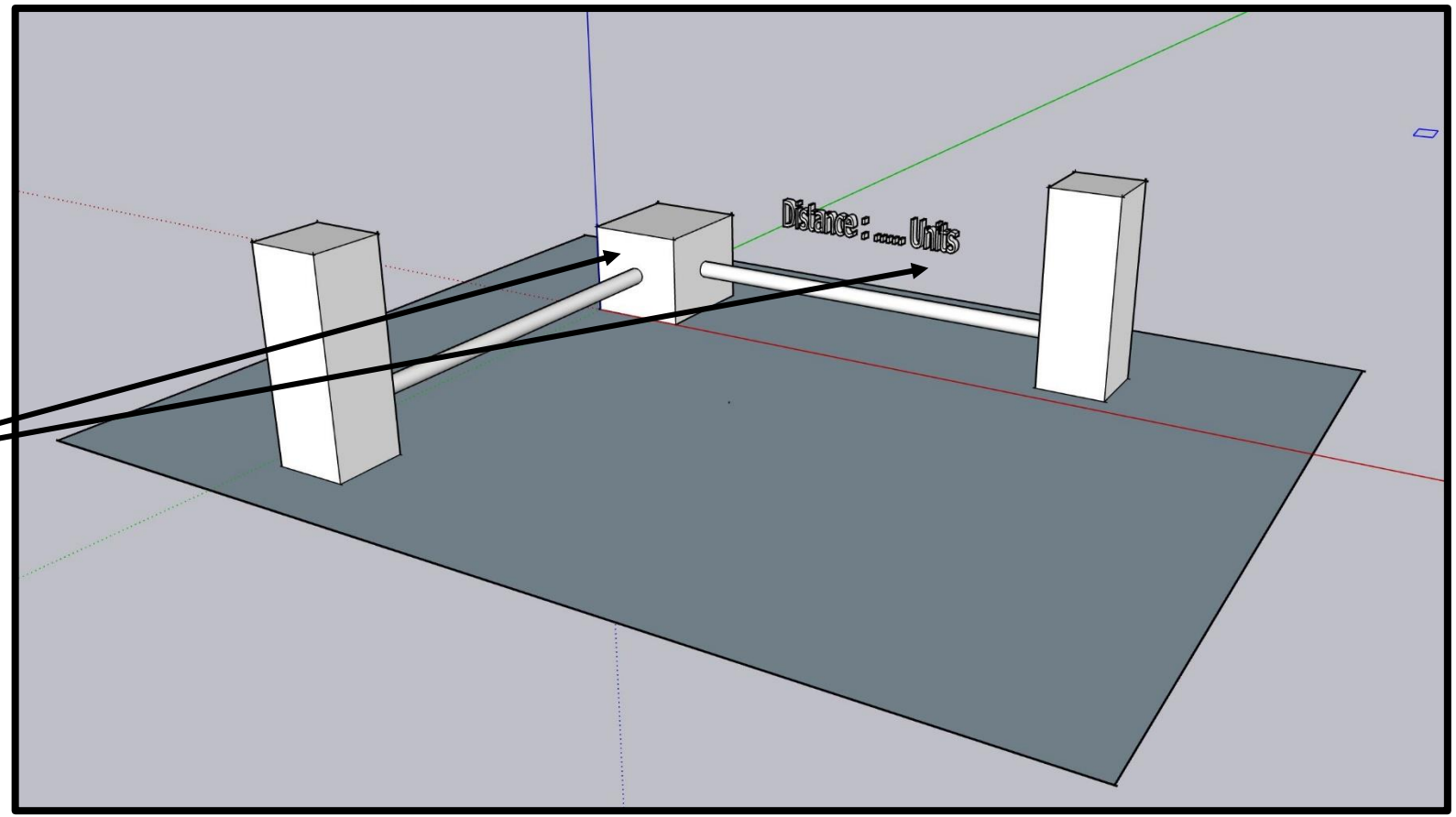
| The Device | The Sensor |
|---|---|
| Arduino Uno | |
| **The Interaction :** | |
| Displaying/Hiding visual data | Touch sensor |
| Changing the scale | Potentiometer |
| Changing the position | Potentiometer |
| **The Device :** | **The Event:** |
| Mouse | |
| **The Interaction :** | |
| Changing the color | Mouse Enter |
| Changing the scale | Mouse Click |

- **The Schematic View of the Prototype**

Two(or more) 3D models, that are already in the scene. They can be real physical models or 3D Objects developed in the AR environment.
In this project we used 3D Objects developed in the AR environment.

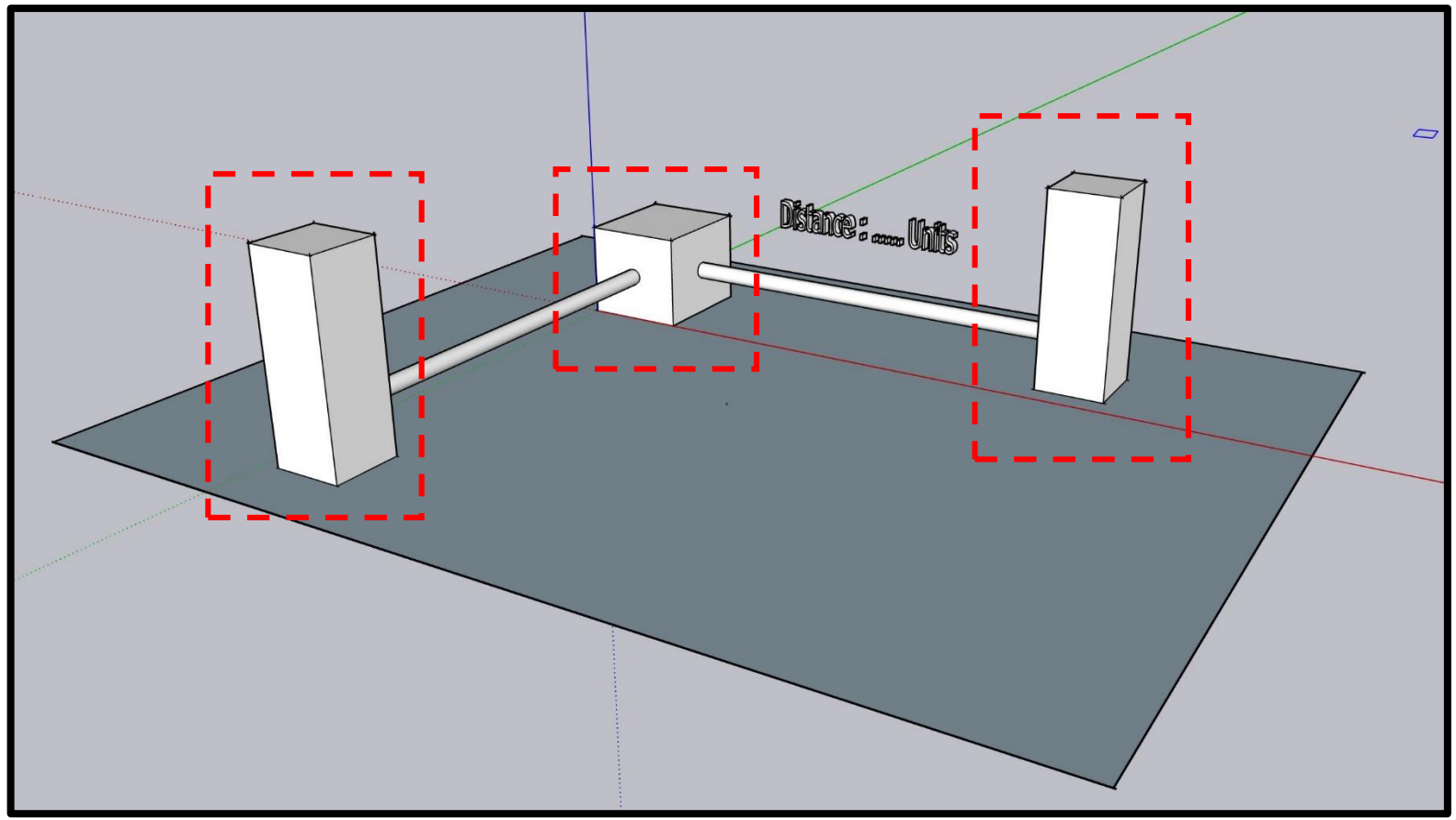


Using Touch sensor to add (display/hide) new 3D Objects and Data (here: the distance between 3D Objects)into the scene.



Perform further manipulations:

Changing Color, scale and the position of the 3D object.



# The Tools:

# Arduino

Arduino is an open-source platform consists of both hardware referred to as a microcontroller and software or Integrated Development Environment (IDE), that can be used to build digital devices. for instance, build low-cost scientific tools, to prove scientific principles such as physics and chemistry, or to get started with programming and robotics. (Arduino, 2023)

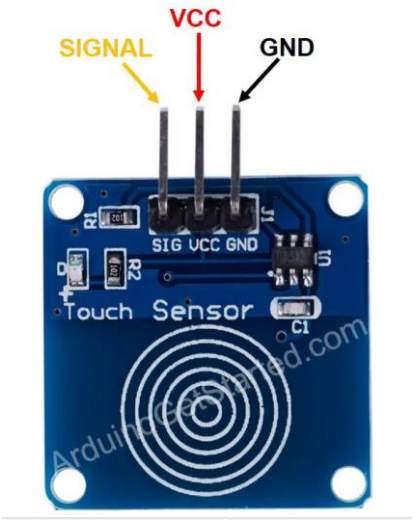
## Development

Arduino can be connected to a variety of IDEs, since it is based on open-source programming. Arduino software code is similar to the C language software code. That being said, to write microcontroller programs, Arduino is considered to be one of the easiest computer programming languages to use. Arduino is predominantly used in designing projects that aim to build Various environmental modules such as Bluetooth, temperature sensor, light sensor, etc. (Arduino, 2023).

# Touch Sensor

Touch sensor has 3 pins:

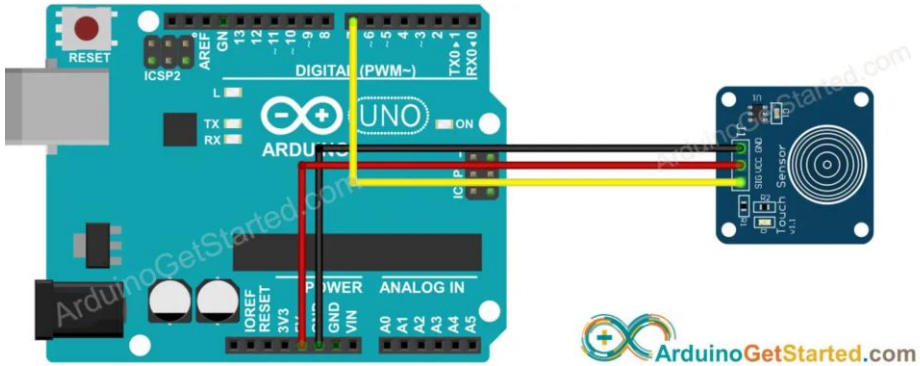
- GND pin: needs to be connected to **GND** (0V)
- VCC pin: needs to be connected to **VCC** (5V or 3.3v)
- SIGNAL pin: is an output pin: **LOW** when it is NOT touched, **HIGH** when it is touched. This pin needs to be connected to Arduino's input pin.




## How It Works

- When the sensor is NOT touched, the sensor's SIGNAL pin is **LOW**
- When the sensor is touched, the sensor's SIGNAL pin is **HIGH**

## Circuit:

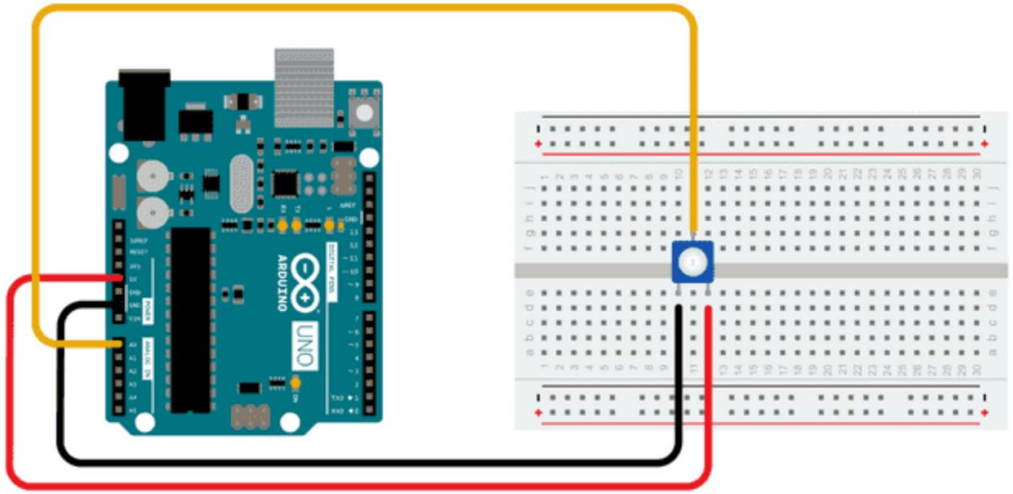


# Potentiometer

A **potentiometer** is a simple mechanical device that provides a varying amount of resistance when its shaft is turned. By passing voltage through a potentiometer and into an analog input on the board, it is possible to measure the amount of resistance produced by a potentiometer (or *pot* for short) as an analog value.

## Circuit

Three wires from the potentiometer should be connected to the board. The first goes from one of the outer pins of the potentiometer to ground. The second goes from the other outer pin of the potentiometer to 5 volts. The third goes from the middle pin of the potentiometer to the analog pin A0.



# The “How”:

To conduct the project we needed to develop an AR 3D environment. In this project we used **web-based AR environment**. In order to develop the AR environment the HTML, CSS, and JavaScript programming languages are used as well as the following libraries:

- AR.js
- THREE.js
- A-Frame.js

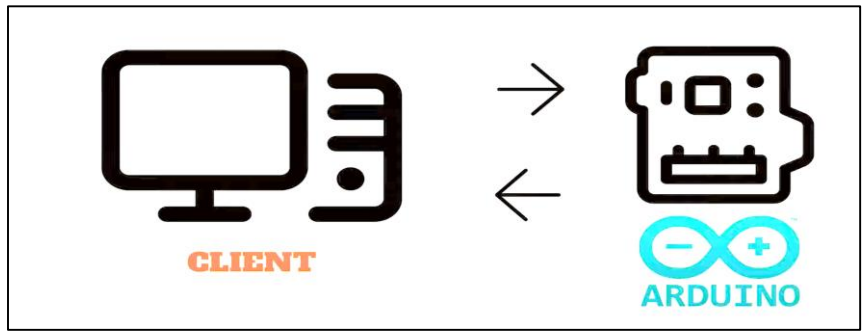
The next step was to set up the physical device; here Arduno uno and the two microcontrollers ( touch sensor and the potentiometer)are used.

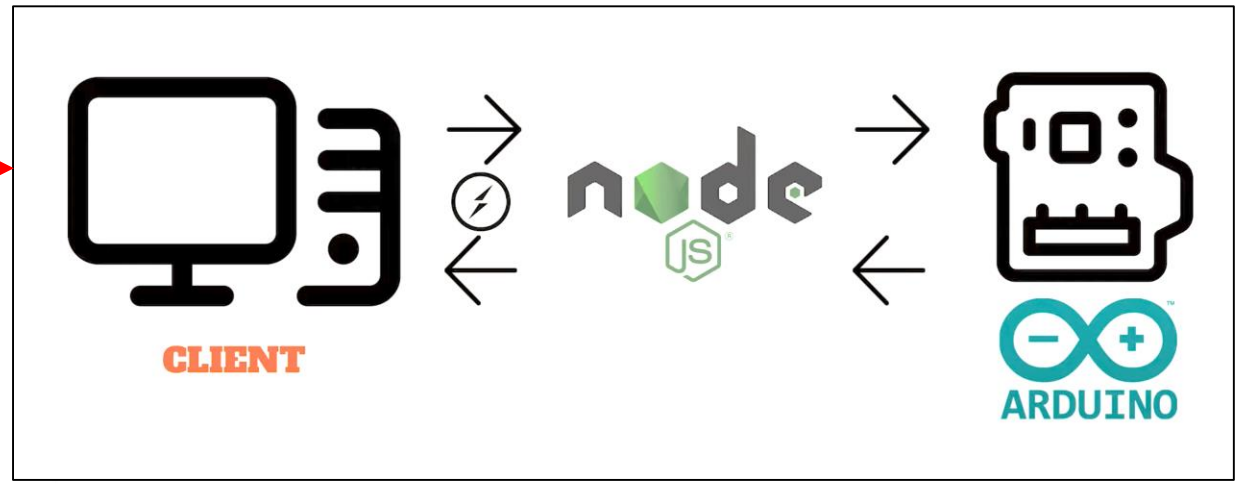
As explained before the touch sensor is used to display and hide Visual data and the potentiometer is used to manipulate the scale of the 3D Object and also changing the position of the 3D Object.

To connect the web based content to the Aduino Uno we needed to develop a local server. To do that we used the Node.Js library and Souket.io and serial port package.

In the following slides the definition of each library and the detailed code of the program is given.

## Communicating from and Arduino  to an HTML/JavaScript Webpage




HTML web page + Socket.io + Node,js + Arduino

Serialport Package

# Working with Web based Augmented reality

**AR.js**

AR.js is a lightweight library for Augmented Reality on the Web, which includes features like Image Tracking, Location based AR and Marker tracking.

**THREE.js**

Three.js is a cross-browser JavaScript library and application programming interface used to create and display animated 3D computer graphics in a web browser using WebGL.

**A-Frame.js**

Frame is a web framework for building virtual reality (VR) experiences. A-Frame is based on top of HTML, making it simple to get started. But A-Frame is not just a 3D scene graph or a markup language; the core is a powerful entity-component framework that provides a declarative, extensible, and composable structure to three.js.

# The Server:

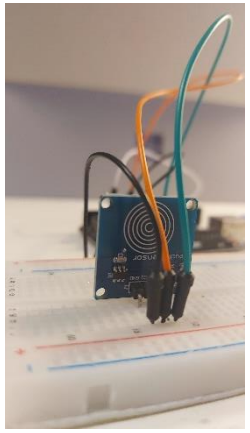
**Node.js**

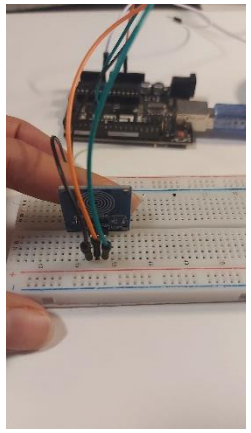
Node.js is a cross-platform, open-source server environment that can run on Windows, Linux, Unix, macOS, and more. Node.js is a back-end JavaScript runtime environment, runs on the V8 JavaScript Engine, and executes JavaScript code outside a web browser.Node.js lets developers use JavaScript to write command line tools and for server-side scripting. The functionality of running scripts server-side produces dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.(Wikipedia)

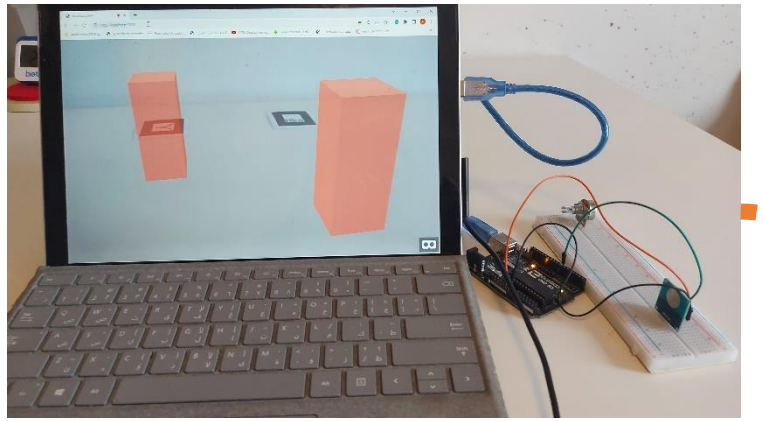
**Socket.IO**

Socket.io is an event-driven library for **real-time web applications**. It enables real-time, bi-directional communication between web clients and servers. It consists of two parts: a client-side library that runs in the browser, and a server-side library for Node.js. Both components have a nearly identical API.
It can be installed with the Node Package Manager (NPM).(Wikipedia)

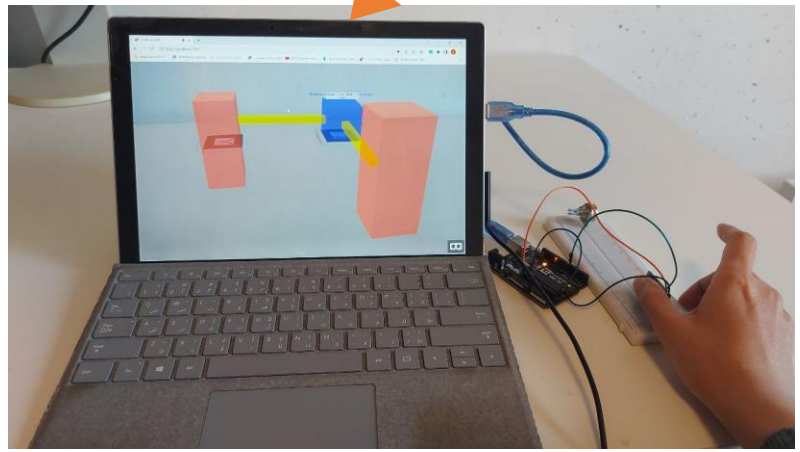
- **The Result and The Code:**

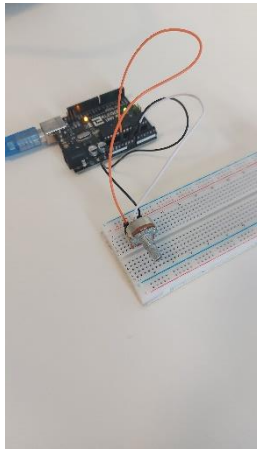


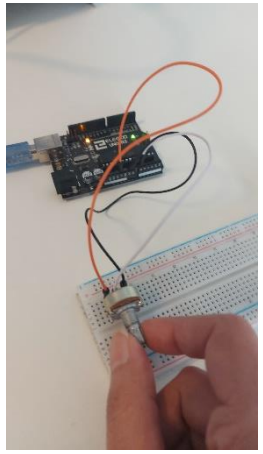


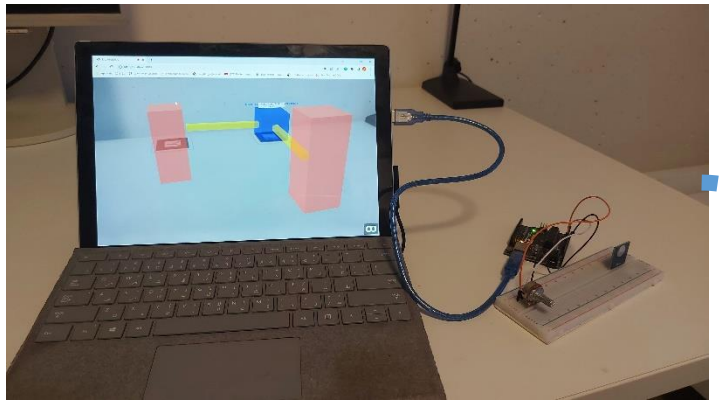
Using Touch Sensor to display/hide visual data



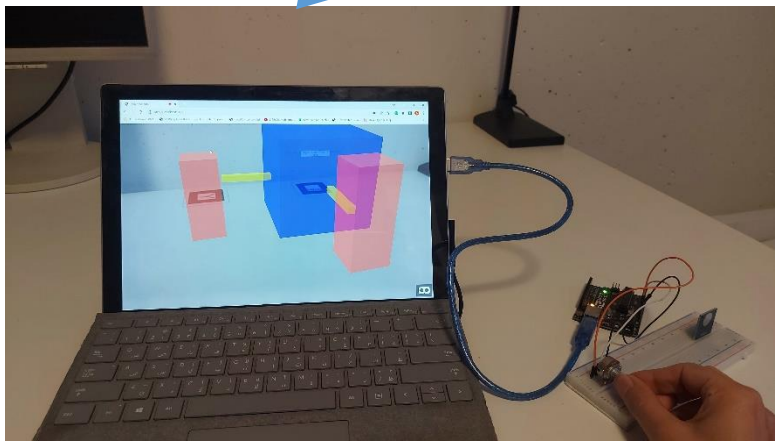
Scan the QR code

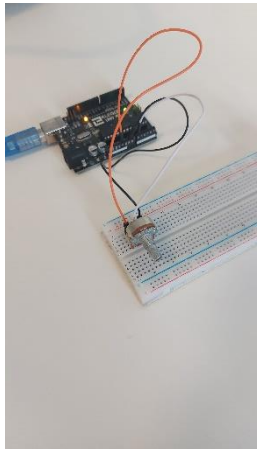


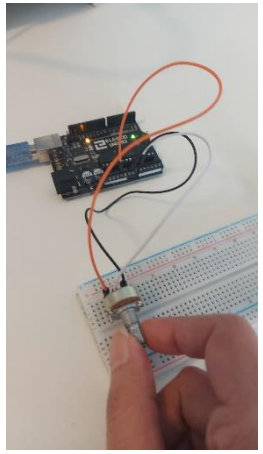


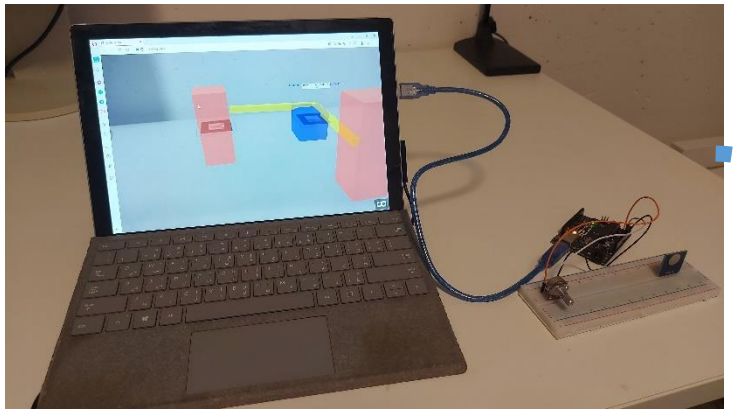
Using Potentiometer to change the Scale of the 3D Object



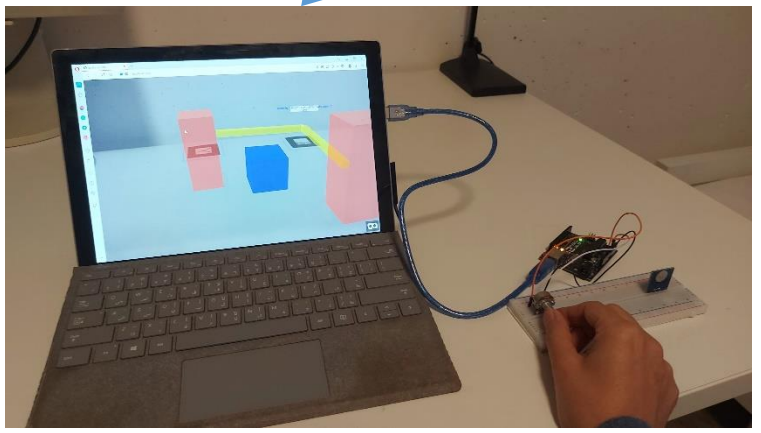
Scan the QR code



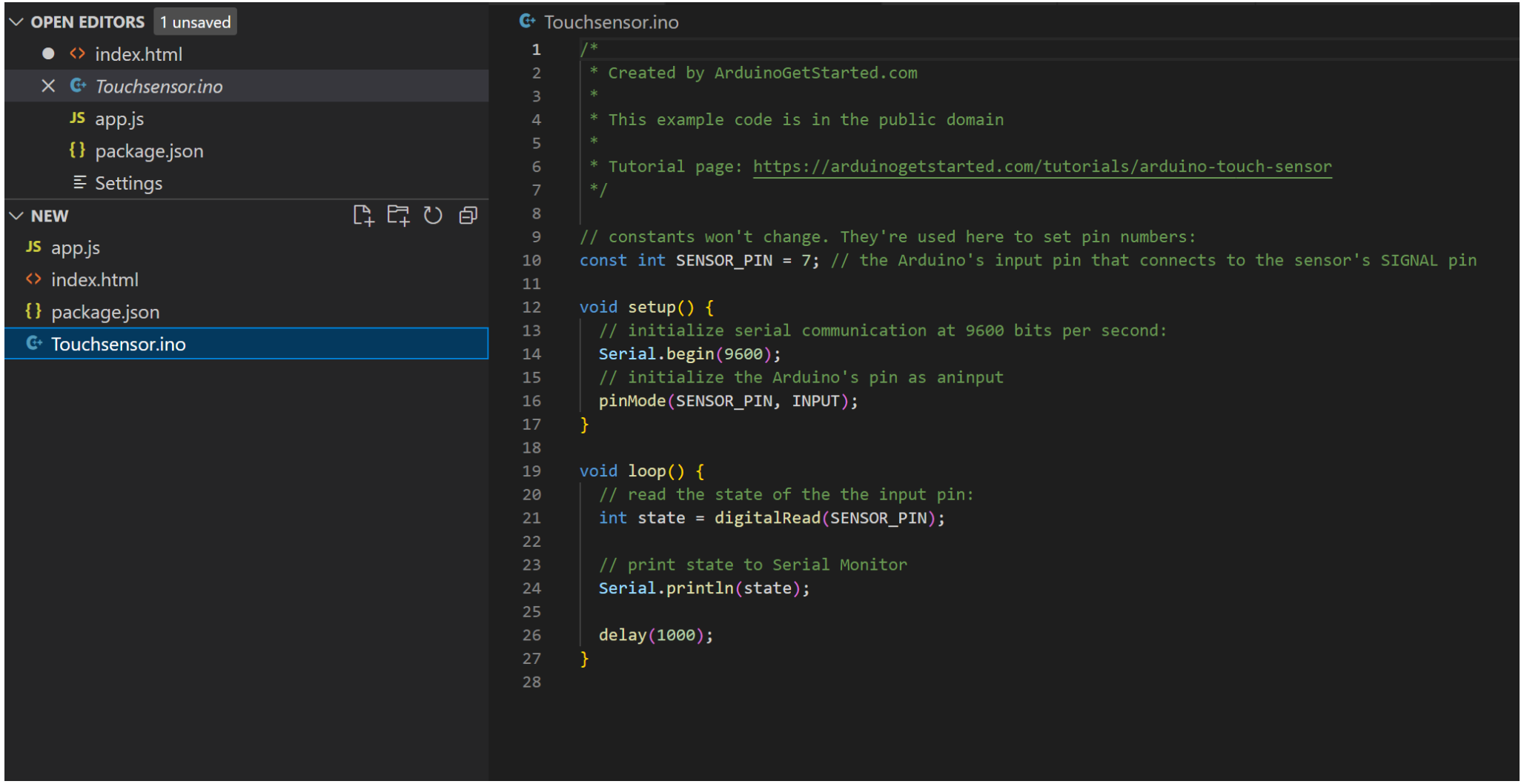


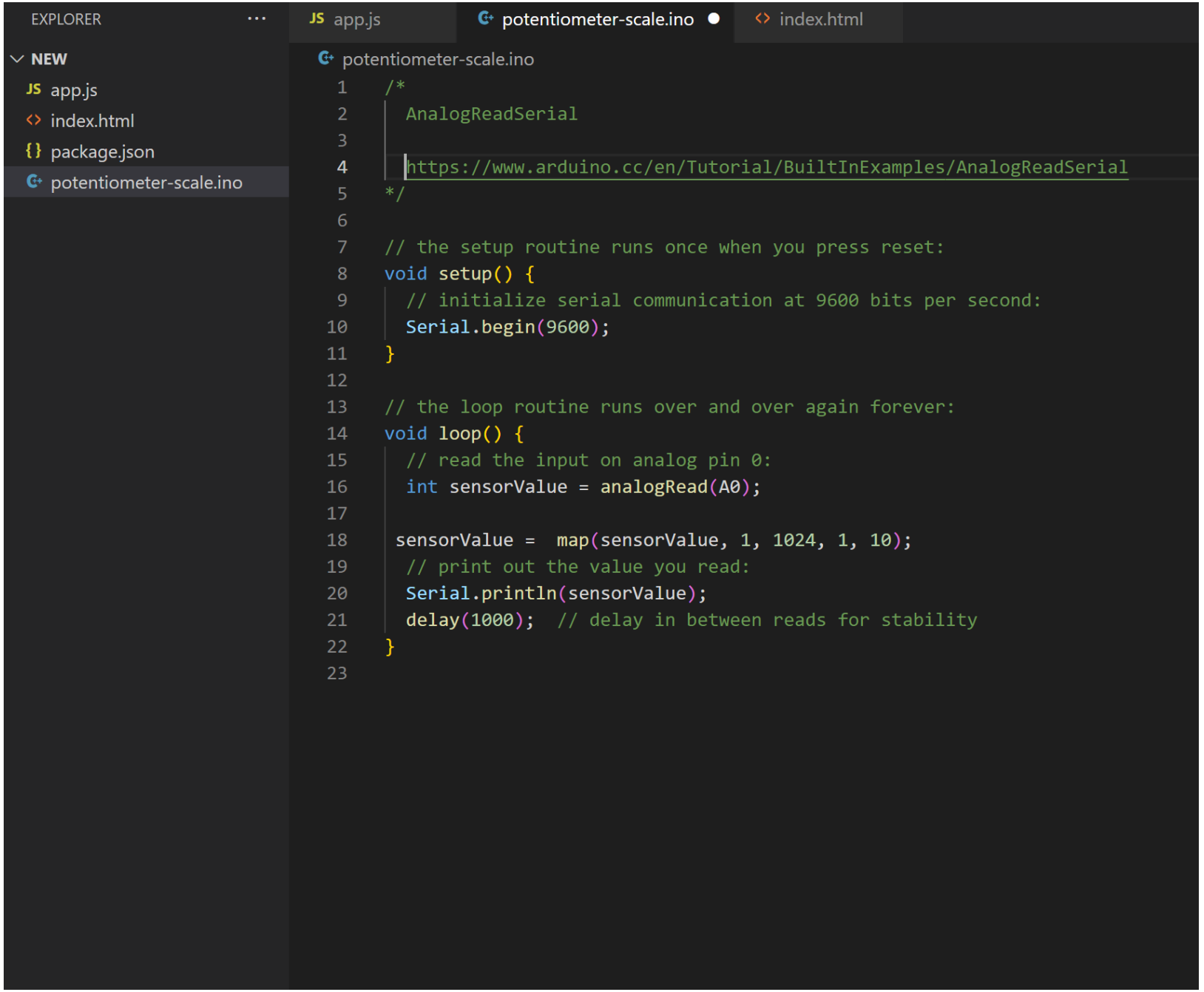
Using Potentiometer to change the Position of the 3D Object

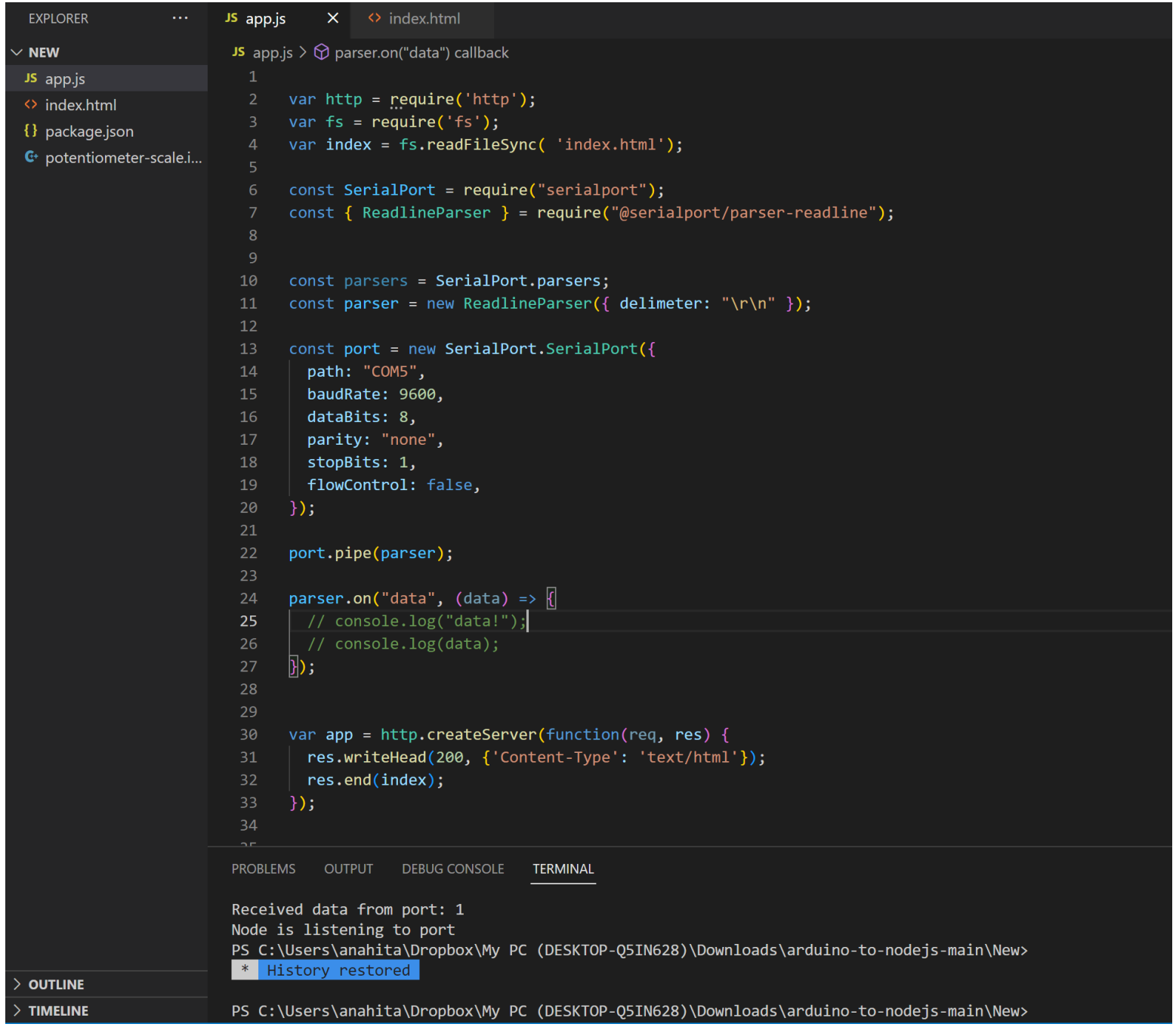


Scan the QR code

# The Code

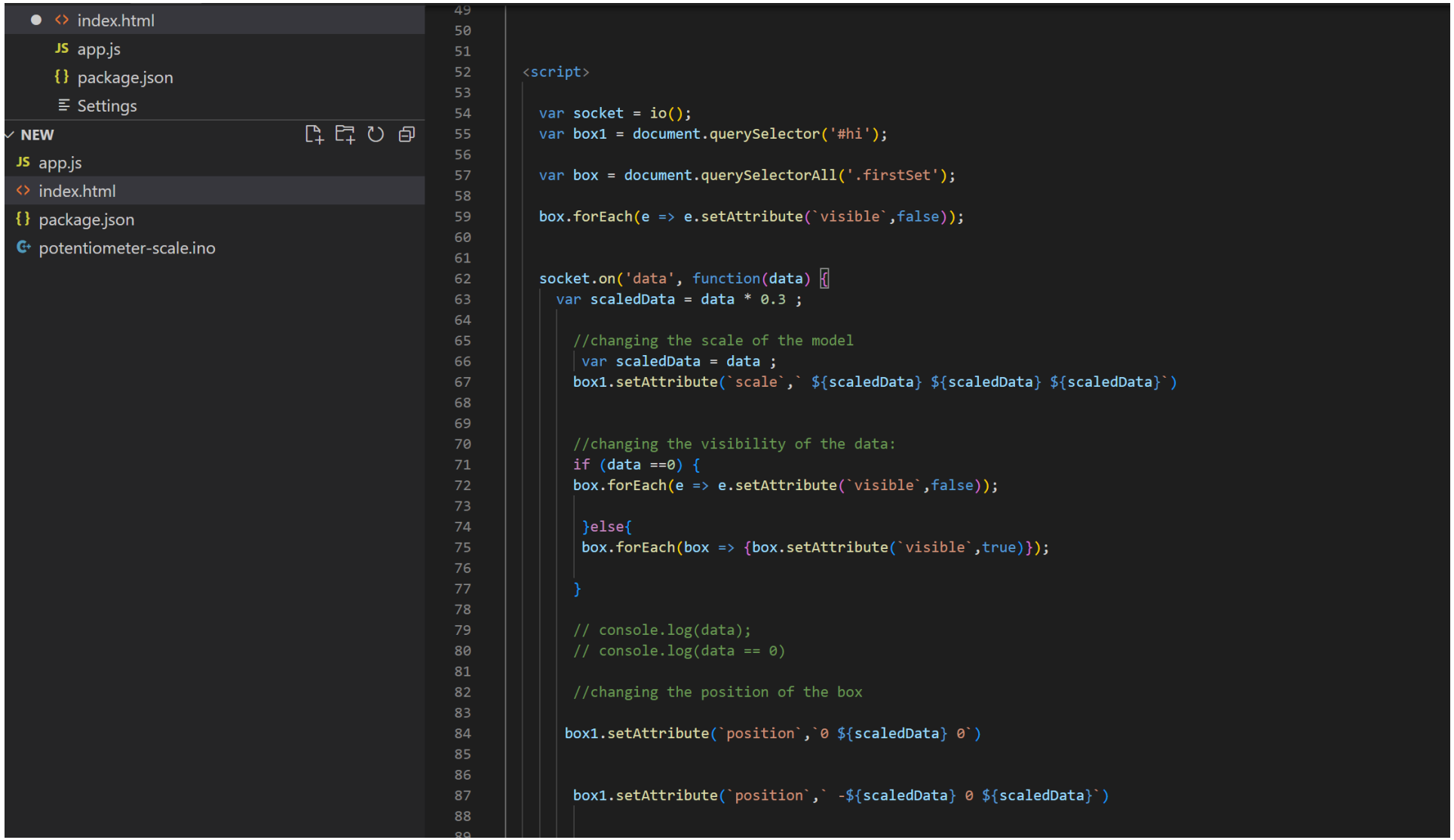

OPEN EDITORS 1 unsaved
index.html
Touchsensor.ino
app.js
package.json
Settings
NEW
app.js
index.html
package.json
Touchsensor.ino
Touchsensor.ino
/*
* Created by ArduinoGetStarted.com
*
* This example code is in the public domain
*
* Tutorial page: https://arduinogetstarted.com/tutorials/arduino-touch-sensor
*/
// constants won't change. They're used here to set pin numbers:
const int SENSOR_PIN = 7; // the Arduino's input pin that connects to the sensor's SIGNAL pin
void setup() {
// initialize serial communication at 9600 bits per second:
Serial.begin(9600);
// initialize the Arduino's pin as aninput
pinMode(SENSOR_PIN, INPUT);
}
void loop() {
// read the state of the the input pin:
int state = digitalRead(SENSOR_PIN);
// print state to Serial Monitor
Serial.println(state);
delay(1000);
}


EXPLORER
NEW
app.js
index.html
package.json
potentiometer-scale.ino
app.js
potentiometer-scale.ino
index.html
potentiometer-scale.ino
/*
AnalogReadSerial
https://www.arduino.cc/en/Tutorial/BuiltInExamples/AnalogReadSerial
*/
// the setup routine runs once when you press reset:
void setup() {
// initialize serial communication at 9600 bits per second:
Serial.begin(9600);
}
// the loop routine runs over and over again forever:
void loop() {
// read the input on analog pin 0:
int sensorValue = analogRead(A0);
sensorValue = map(sensorValue, 1, 1024, 1, 10);
// print out the value you read:
Serial.println(sensorValue);
delay(1000); // delay in between reads for stability
}


EXPLORER
NEW
app.js
index.html
package.json
potentiometer-scale.i...
OUTLINE
TIMELINE
app.js
index.html
app.js > parser.on("data") callback
var http = require('http');
var fs = require('fs');
var index = fs.readFileSync( 'index.html');
const SerialPort = require("serialport");
const { ReadlineParser } = require("@serialport/parser-readline");
const parsers = SerialPort.parsers;
const parser = new ReadlineParser({ delimeter: "\r\n" });
const port = new SerialPort.SerialPort({
path: "COM5",
baudRate: 9600,
dataBits: 8,
parity: "none",
stopBits: 1,
flowControl: false,
});
port.pipe(parser);
parser.on("data", (data) => {
// console.log("data!");
// console.log(data);
});
var app = http.createServer(function(req, res) {
res.writeHead(200, {'Content-Type': 'text/html'});
res.end(index);
});
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Received data from port: 1
Node is listening to port
PS C:\Users\anahita\Dropbox\My PC (DESKTOP-Q5IN628)\Downloads\arduino-to-nodejs-main\New>
* History restored
PS C:\Users\anahita\Dropbox\My PC (DESKTOP-Q5IN628)\Downloads\arduino-to-nodejs-main\New>

index.html > html > body > a-scene > a-marker > a-box#6

```
<!doctype html>
<html>
<head>

  <script src='https://cdnjs.cloudflare.com/ajax/libs/socket.io/2.0.4/socket.io.js'></script>
<!-- include A-Frame obviously -->
<script src="https://aframe.io/releases/0.6.0/aframe.min.js"></script>
<!-- include ar.js for A-Frame -->
<script src="https://jeromeetienne.github.io/AR.js/aframe/build/aframe-ar.js"></script>


</head>

<body style='margin : 0px; overflow: hidden;'>
<a-scene embedded  arjs="debugUIEnabled: false;">
  <!--<a-entity cursor="rayOrigin:mouse"></a-entity>-->

  <a-marker preset='hiro'>
      <a-entity position="0 1.2 0" geometry="primitive: plane; width: 1; height: auto" material="color: #eee"
                text="color: blue; align: center; value: Model No.1, Height : 1 unit, Width : 1 unit, Depth : 1 unit; width: 2; ">
      </a-entity>
      <a-box id = "hi" class= "firstSet" position='0 0.5 0' color = "blue" material='opacity: 0.5;'></a-box>
      <a-box  id = "2" class= "firstSet" position='0 0.5 1.5' color = "yellow" material='opacity: 0.5;' height = .2; depth = 3; width= .2;></a-box>
      <a-box  id = "3"  class= "firstSet" position='-1.5 0.5 0' rotation = ' 0 90 0'; color = "yellow" material='opacity: 0.5;' height = .2; depth = 3; width= .2;></a-box>
      <a-box id = "4" class= "firstSet" position='-3 0.5 3' material='opacity: 0.5;'></a-box> -->
  </a-marker>

<a-marker preset='kanji'>
      <a-entity position="0 2.2 0" geometry="primitive: plane; width: 1; height: auto" material="color: #eee"
      text="color: blue; align: center; value: Model No.2, Height : 2 unit, Width : 1 unit, Depth : 1 unit; width: 2; ">
      </a-entity>

     <a-box  id = "5" position='0 0.5 0'  color = "tomato" material='opacity: 0.5;' height =2.5  animation="startEvents: click; property: scale;
    from: 1 1 1 ; to: 2 2 2; dur: 1000" animation__mouseenter="property: components.material.material.color; type: color; to: blue; startEvents: mouseenter; dur: 500"
    ></a-box>
    <a-box  id = "6" position='-3.5 0.5 3.5'  color = "tomato" material='opacity: 0.5;' height =2.5
     animation__mouseenter="property: components.material.material.color; type: color; to: blue; startEvents: mouseenter; dur: 500"
     animation__mouseleave="property: components.material.material.color; type: color; to: red; startEvents: mouseleave; dur: 500" ></a-box>
    </a-marker>

<!-- define a camera which will move according to the marker position -->
<a-entity camera></a-entity>

</a-scene>
```


index.html
app.js
package.json
Settings
NEW
app.js
index.html
package.json
potentiometer-scale.ino
<script>
var socket = io();
var box1 = document.querySelector('#hi');
var box = document.querySelectorAll('.firstSet');
box.forEach(e => e.setAttribute(`visible`,false));
socket.on('data', function(data) {
var scaledData = data * 0.3 ;
//changing the scale of the model
var scaledData = data ;
box1.setAttribute(`scale`,` ${scaledData} ${scaledData} ${scaledData}`)
//changing the visibility of the data:
if (data ==0) {
box.forEach(e => e.setAttribute(`visible`,false));
}else{
box.forEach(box => {box.setAttribute(`visible`,true)});
}
// console.log(data);
// console.log(data == 0)
//changing the position of the box
box1.setAttribute(`position`,`0 ${scaledData} 0`)
box1.setAttribute(`position`,` -${scaledData} 0 ${scaledData}`)

# References:


1. Yahya Gh., Shamus S. Interaction in Augmented Reality: Challenges to Enhance User ExperienceI.CVARS 2020: Proceedings of the 2020 4th International Conference on Virtual and Augmented Reality Simulations., (2020),39–44.https://doi.org/10.1145/3385378.3385384

2. Azuma, R. T. A survey of augmented reality. Presence: Teleoper. Virtual Environ., 6, 4 (1997), 355--385. DOI=https://doi.org/10.1162/pres.1997.6.4.355

3. Milgram, P. and Kishino, F. A taxonomy of mixed reality visual displays. IEICE Trans. Information Systems, vol. E77-D, no. 12(1994), 1321--1329.

4. Perkins, S. L., Lin, M. A., Srinivasan, S., Wheeler, A. J., Hargreaves, B. A. and Daniel, B. L. A mixed-reality system for breast surgical planning. Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR-Adjunct), Oct. 2017, pp. 269--274. DOI=https://doi.org/10.1109/ISMAR-Adjunct.2017.92

5. Havard, V., Baudry, D., Louis, A. and Mazari, B. Augmented reality maintenance demonstrator and associated modelling. Proc. IEEE Virtual Reality (VR), Mar. 2015 pp. 329--330. DOI=https://doi.org/10.1109/VR.2015.7223429

6. Andrea, R., Lailiyah, S. R., Agus, F. and Ramadiani, R. "Magic Boosed" an elementary school geometry textbook with marker-based augmented reality. TELKOMNIKA Indonesian Journal of Electrical Engineering, 17 (2019), 1242--1249. DOI=http://dx.doi.org/10.12928/telkorrmika.v17i3.11559

7. Turkan, Y., Radkowski, R., Karabulut-Ilgu, A., Behzadan, A. H. and Chen, A. Mobile augmented reality for teaching structural analysis. Advanced Engineering Informatics, 34 (2017), 90—100

8. Kaufmann, H. and Schmalstieg, D. Mathematics and geometry education with collaborative augmented reality. Computers & Graphics, 27, 3 (2003), 339--345. DOI=https://doi.org/10.1016/S0097-8493(03)00028-1

9. Lorensen, W., Cline, H., Nafis, C., Kikinis, R., Altobelli, D. and Gleason, L. Enhancing reality in the operating room. Proc.