Step 1: Understanding Three.js

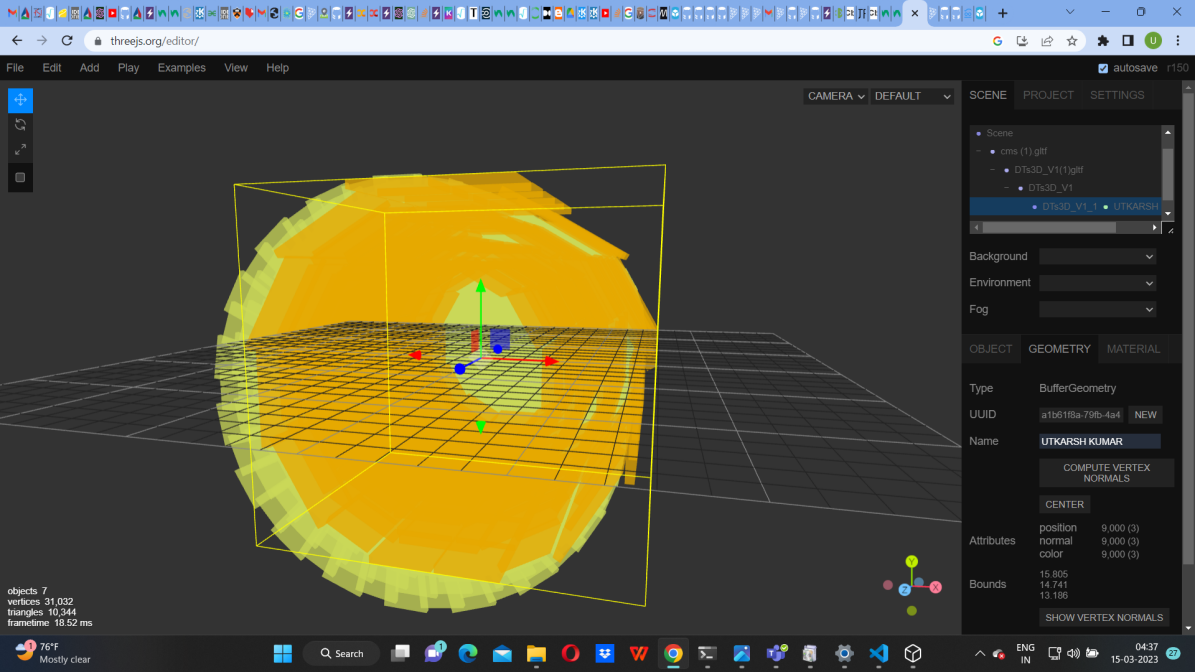
**NAME -UTKARSH KUMAR**

**EMAIL[kumarutkarsh946@gmail.com](mailto:kumarutkarsh946@gmail.com)**

**1 - We want you to use the tools provided by the Three.js library to load and rotate example detector geometry. The geometry is provided in the glTF format and the file (“cms.gltf”) can be found in the Google Drive link provided above.**

**Ans)**

**Just view**

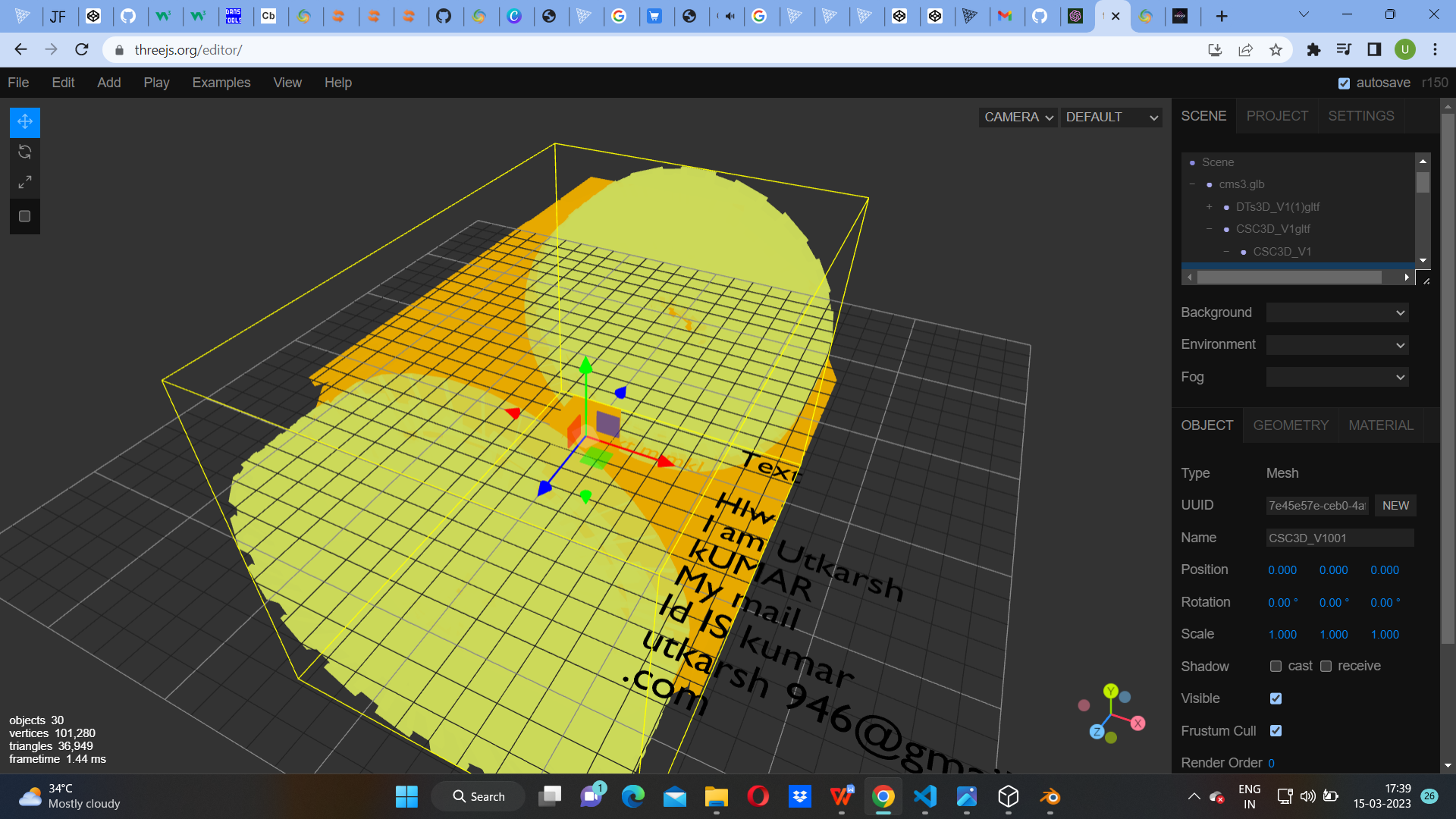


2 - You should next **overlay some text** on the page. The actual text doesn’t matter - the point is to show that you can work with the Three.js library and also traditional HTML layout.

**ANS)**

**The Html layout are**

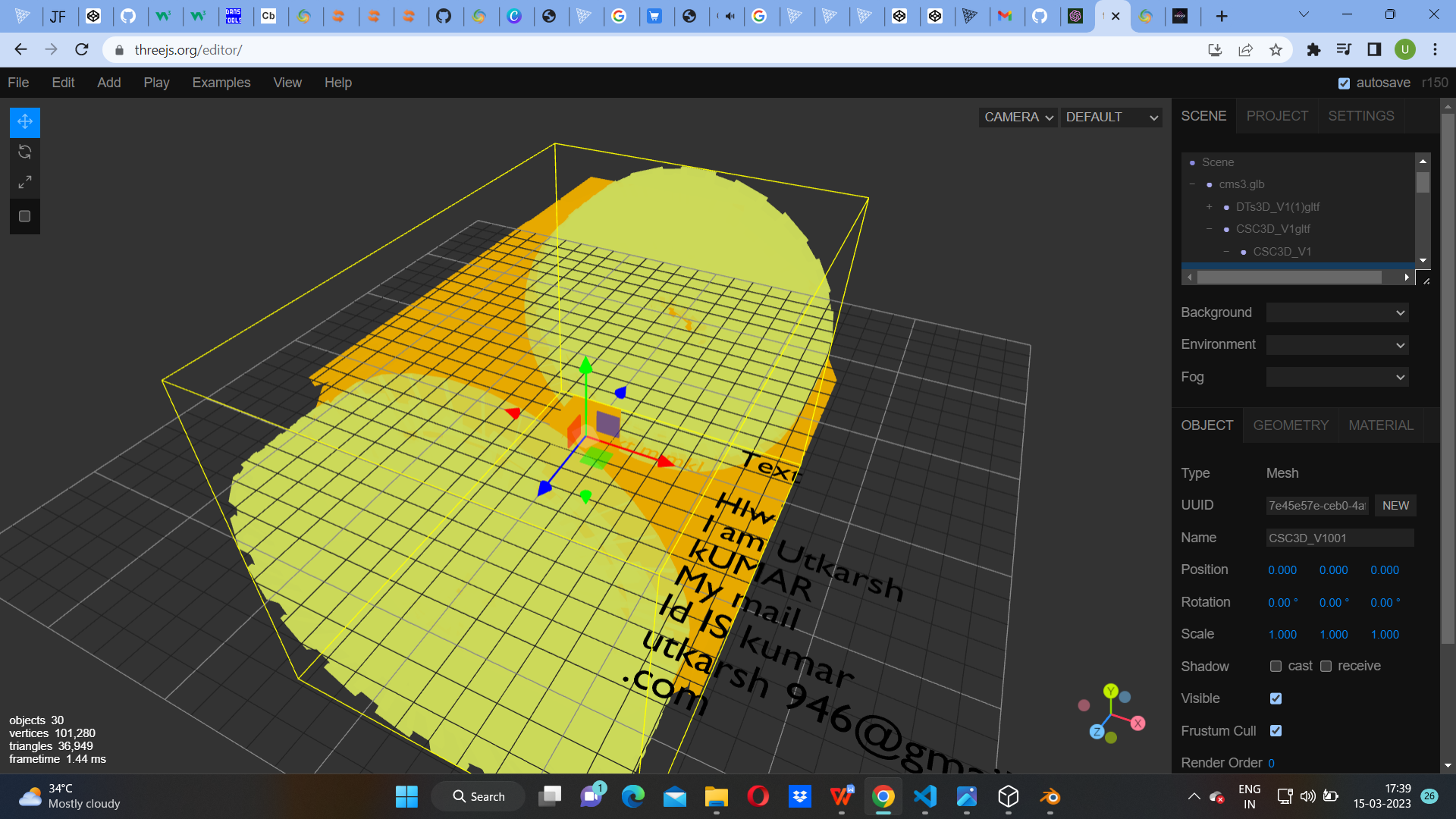
**Yes I did it but in josn**



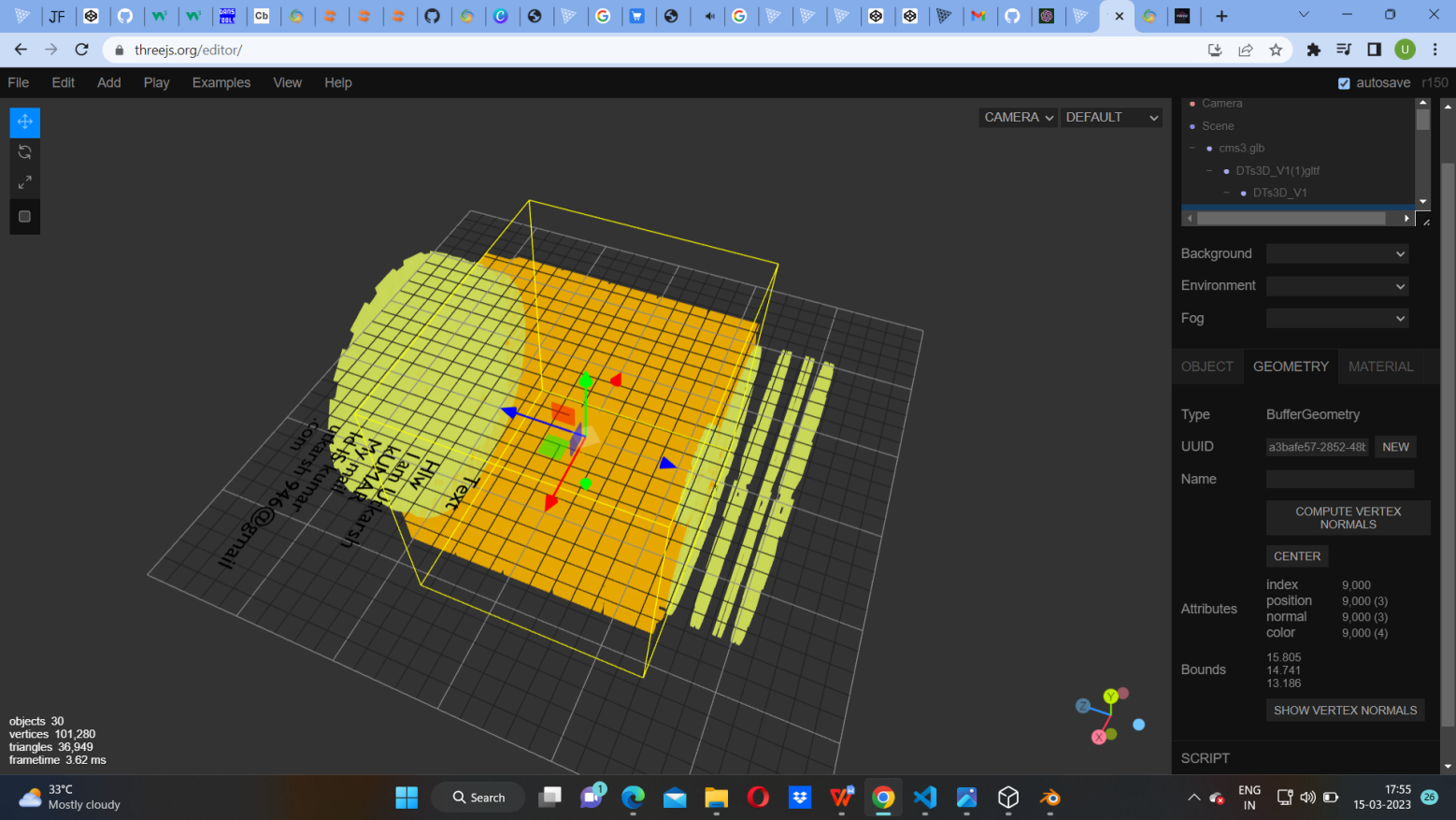
3 - As a bonus, make it possible to **select objects by clicking them**, and then **display information** about them in the text overlay.

**Ans) Selecting object by clicking on them and show information of it**

**Object View OF THE OUTER PART.**



**Geometry View**

****

*4. Optional:* Can you identify what different parts of the detector do?

A presentation with clues can be found here: [https://www.desy.de/~garutti/LECTURES/ParticleDetectorSS12/L1\_Introduction\_HEPdetectors.pdf](https://www.desy.de/~garutti/LECTURES/ParticleDetectorSS12/L1_Introduction_HEPdetectors.pdf" \t "_blank)

**ANS)THE PART VIEWS ARE**

**The hemisphere like structure is called**

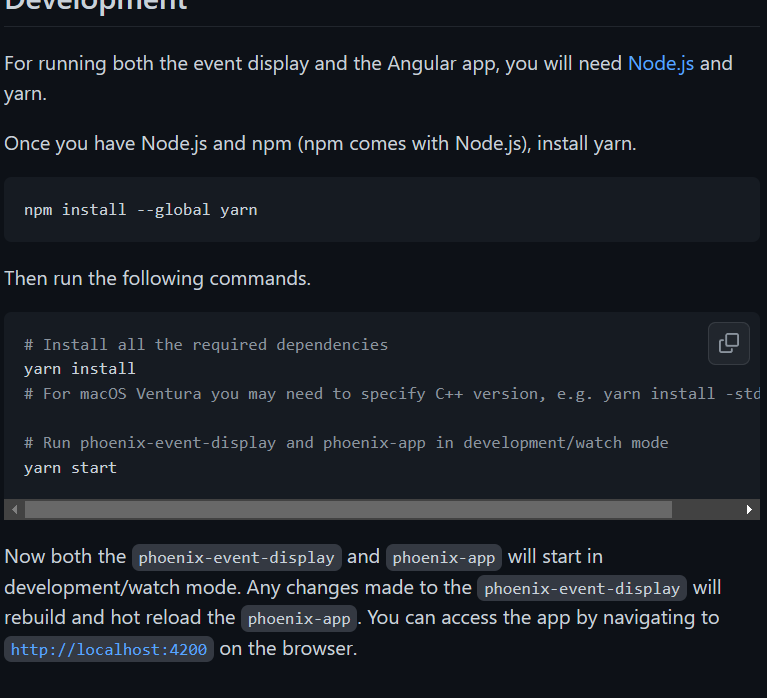
THE OUTER PART

Step 2: Checking and running Phoenix

You should download and set up Phoenix, following the instructions:

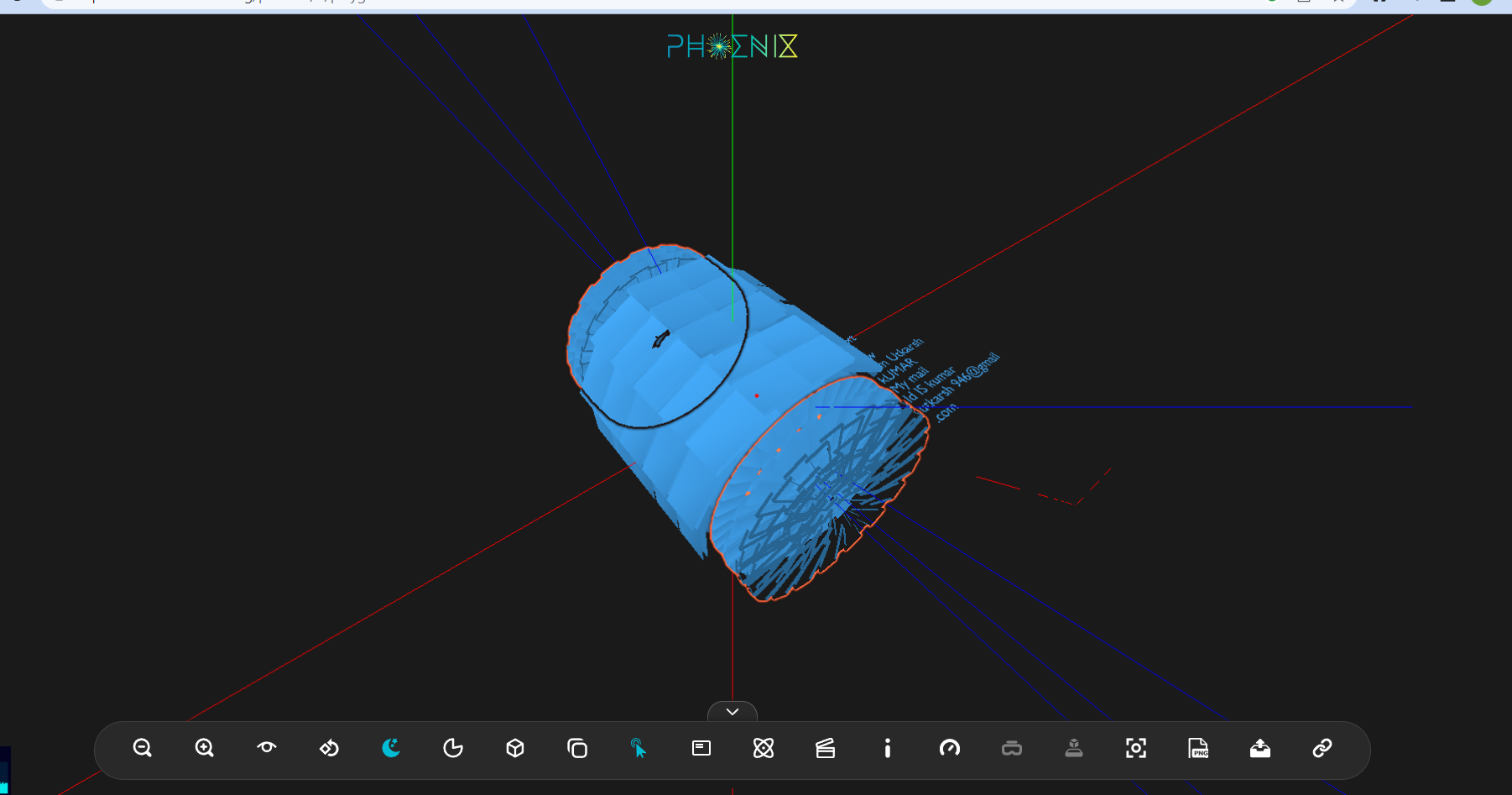
[https://github.com/HSF/phoenix#development](https://github.com/HSF/phoenix" \l "development" \t "_blank)

YES I DOWNLOADED PHOENIX BY USING



Next we would like you to make a change to the “view options” of the phoenix iconbar:

[https://github.com/HSF/phoenix/blob/main/guides/users.md#the-phoenix-iconbar](https://github.com/HSF/phoenix/blob/main/guides/users.md" \l "the-phoenix-iconbar" \t "_blank)



**This two side**

**ANS)-positron emission tomography(END CAP)**

**THE FUNCTION IS e-e+->HZ**

**THE MIDDLE PART IS**

**ANS)**

**Barrel**

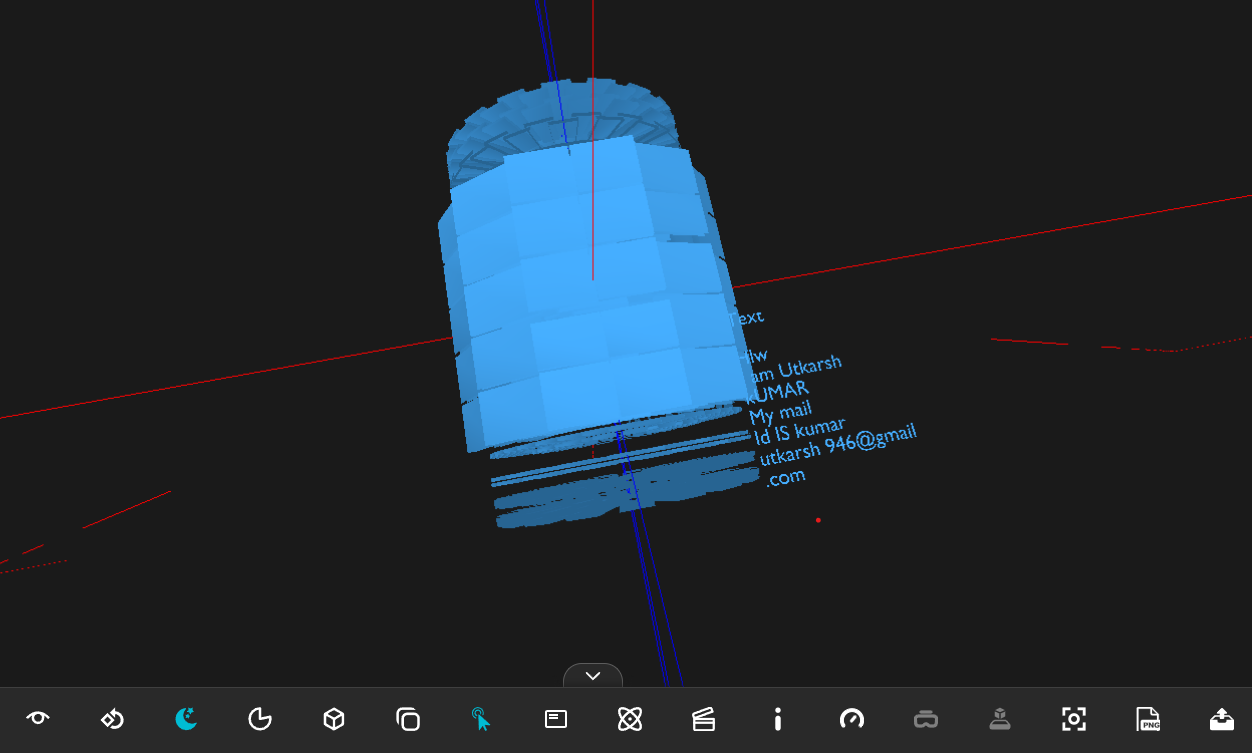
**Function of it is**

**Detection of electron with 16.9 MeV initial energy. It spirals about 36 times in the magnetic field. At the end of the visible track the energy has decreased to 12.4 MeV. from the visible path length (1030cm) the energy loss by ionization is calculated to be 2.8MeV.**  
**Yes I did it**

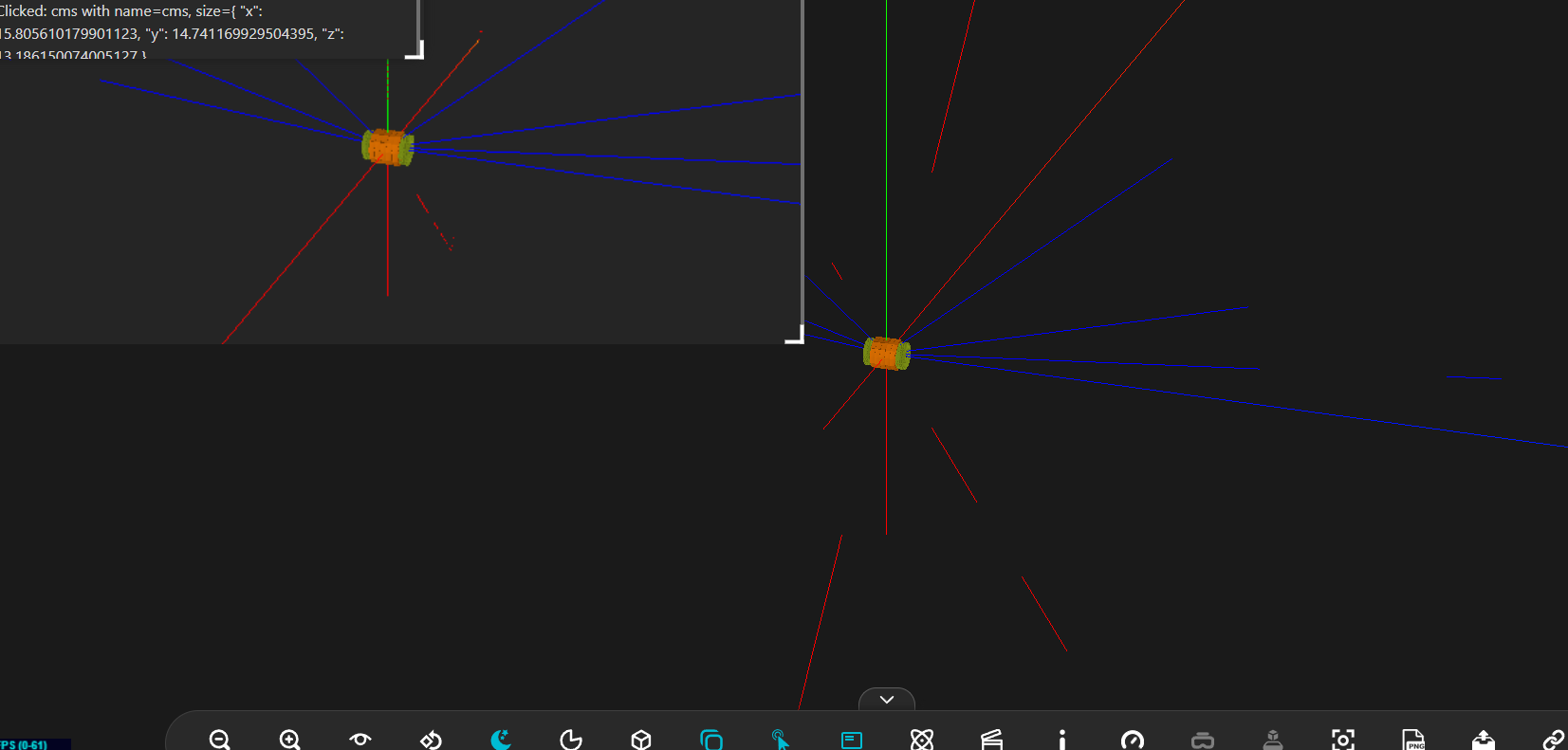
**Currently there are options to ‘Show Grid’, ‘Show Axis’ etc. The ‘Grid’ isn’t really a grid, it’s currently a set of lines in eta and phi directions, which is very useful for understanding physics events, but not so helpful in determining where objects are in cartesian coordinates.**

**ANS) THE DIFFERENT AXIS ARE**

GRIDS ARE

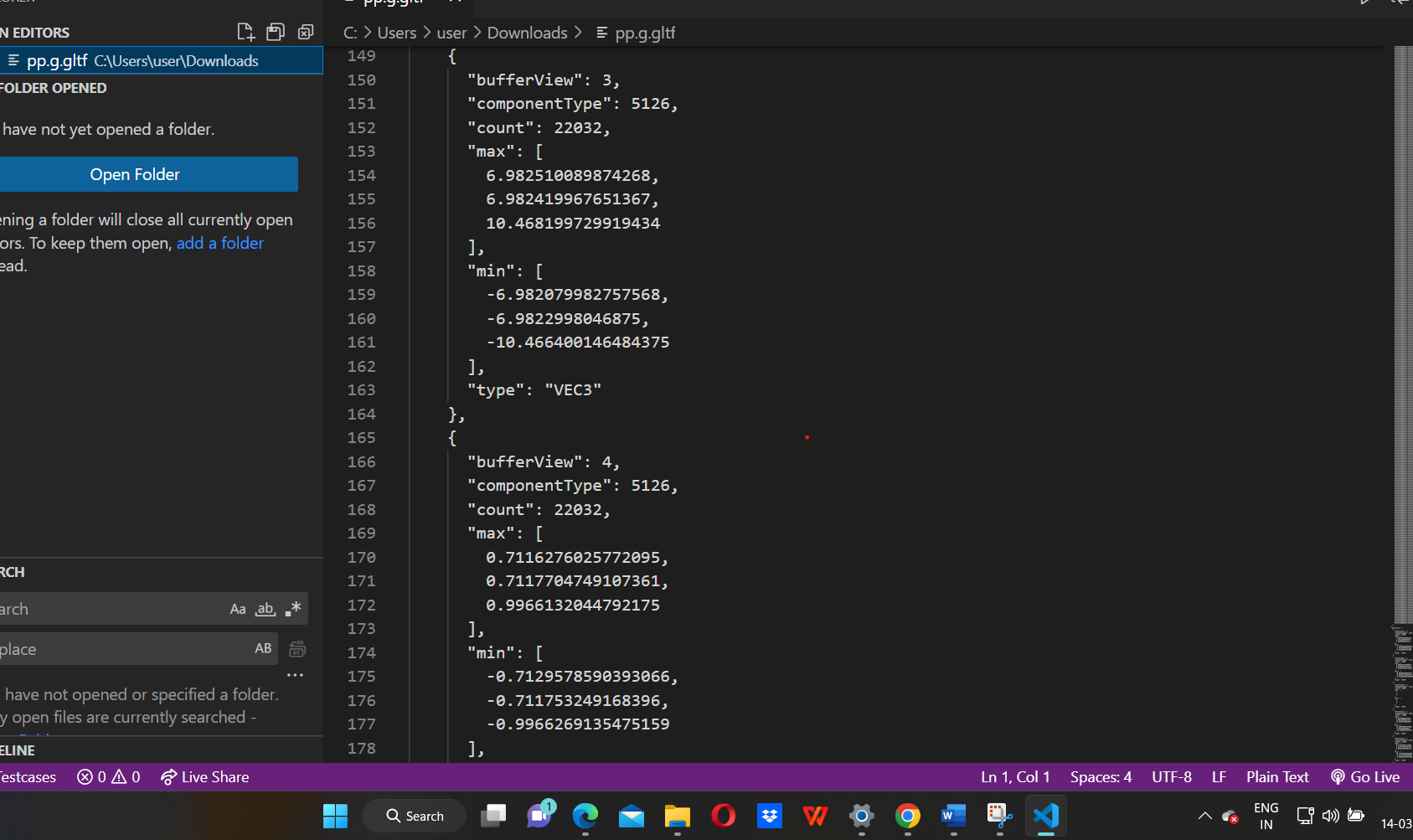


ANOTHER VIEW IS

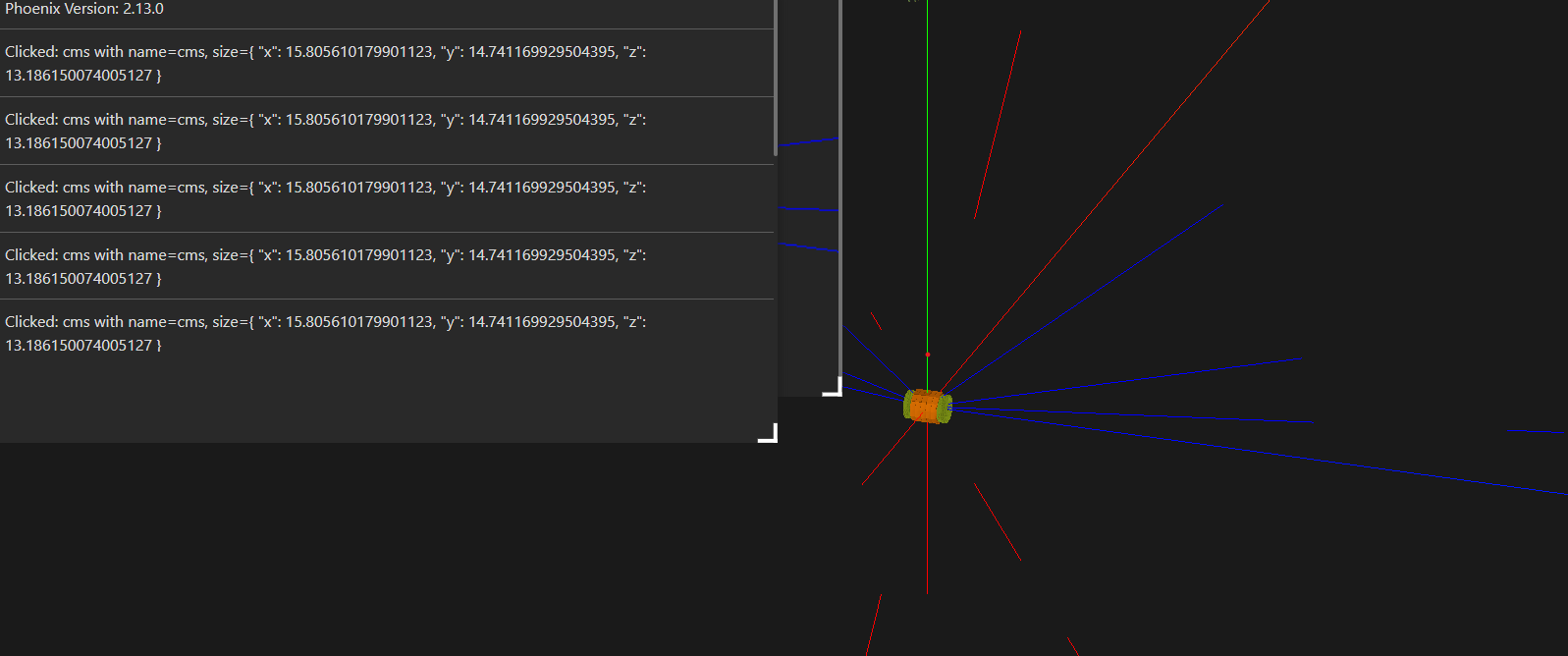


**We would like you to**

1. **Replace the existing eta/phi ‘grid’ with a proper cartesian (x,y,z) grid. Experiment with different grid pitches, and if possible, have labels to show coordinates on the axes.**
2. **YES I CHANGE THE POSITION OF COORDINATE**



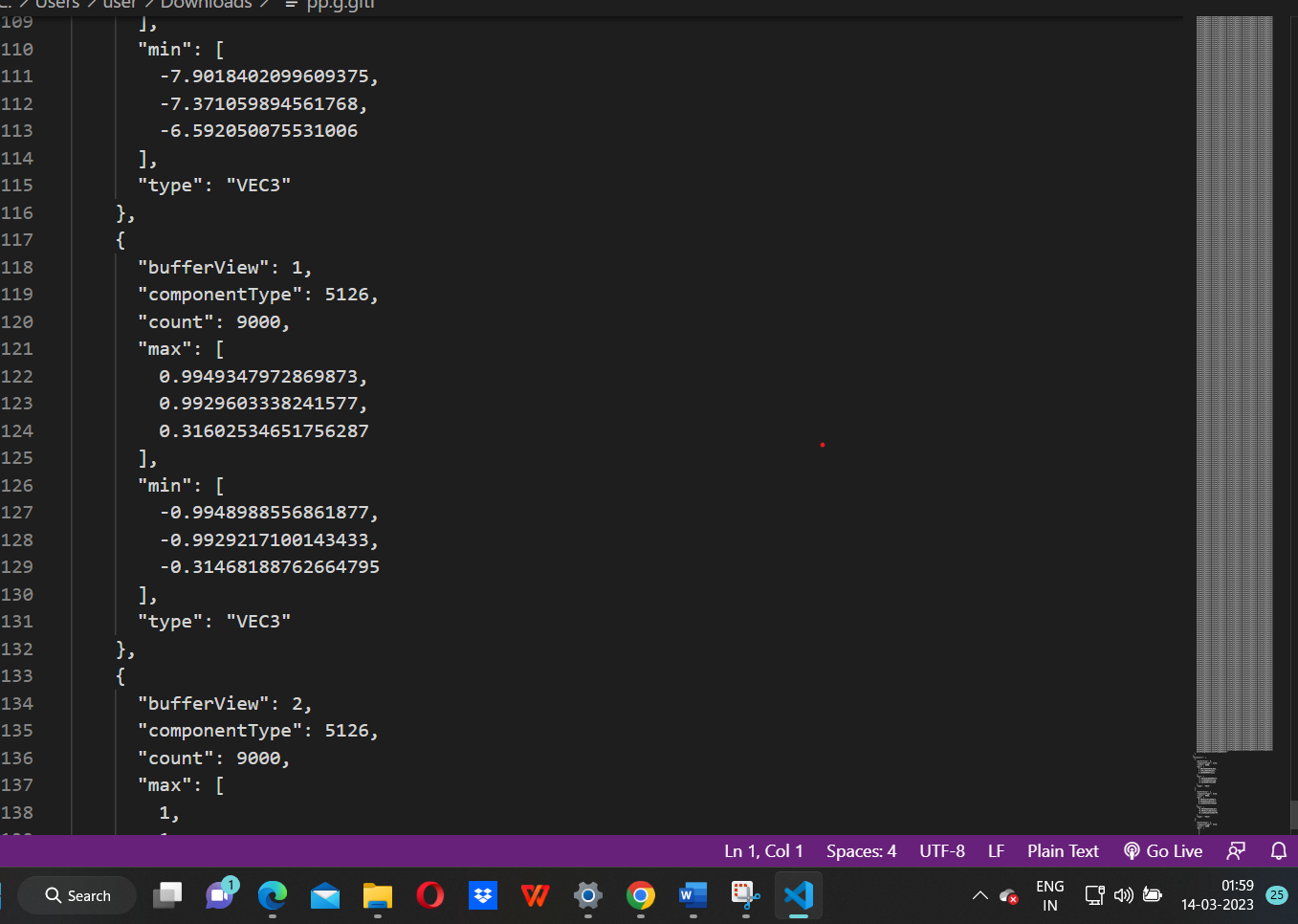
**THE NEW POINT VIEW ARE**



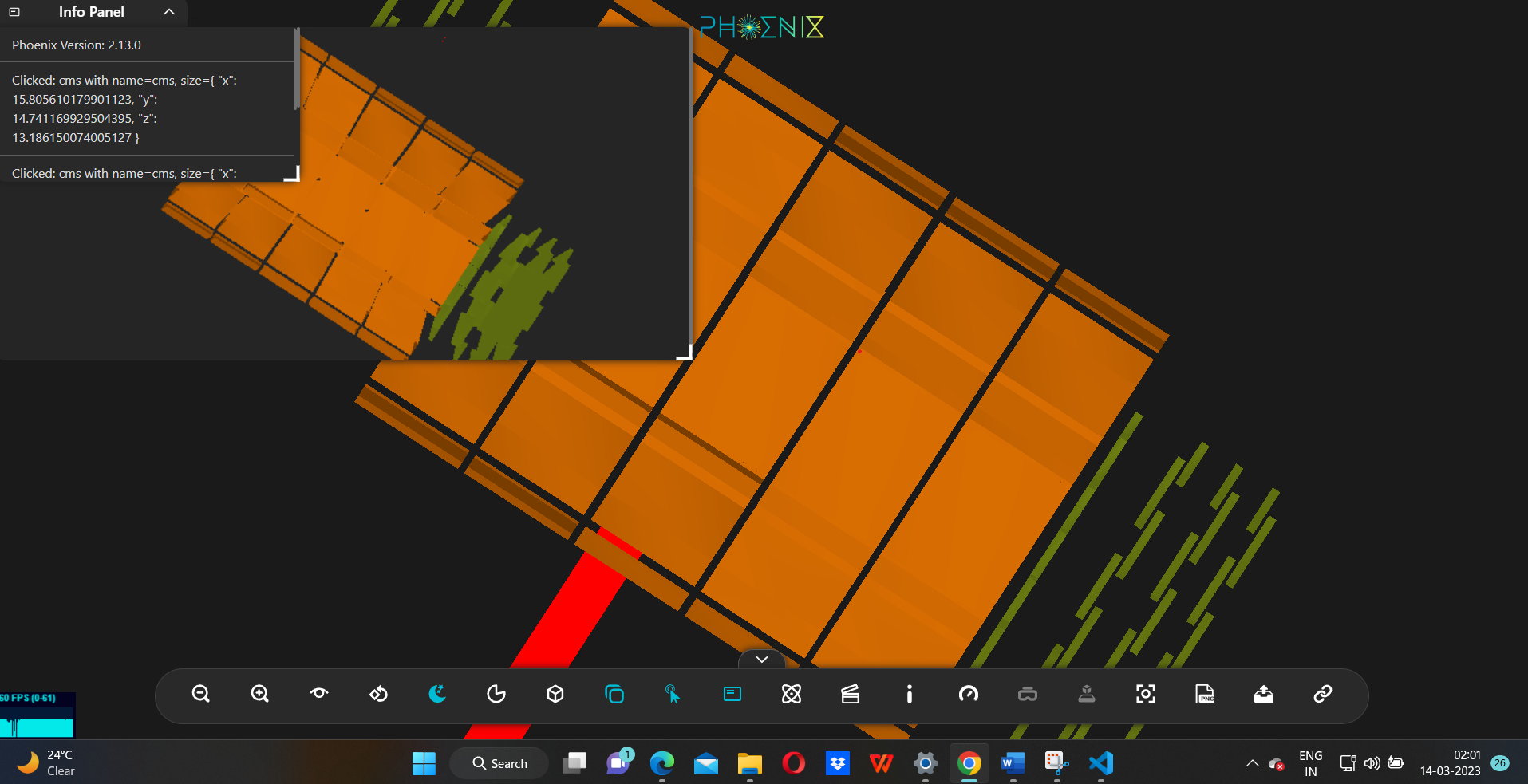
1. **(Optionally) add back the eta/phi grid and modify the menu so this is shown with “show eta/phi lines” and the new cartesian grid is shown with ‘Show Grid’.**

**ANS)**

**YES I CHANGE THE ETA/PHI LIENS**



**New grid is**



The goal here is to add  a new way to help users see where objects are in space. It should be functional, and attractive.

**Some notes:**

* There’s some documentation on the general UI here:

[https://github.com/HSF/phoenix/blob/master/guides/developers/event-display.md#uimanager](https://github.com/HSF/phoenix/blob/master/guides/developers/event-display.md" \l "uimanager" \t "_blank)

* We also have detailed documentation in the API docs… this in particular will be useful:
  + [https://hepsoftwarefoundation.org/phoenix/api-docs/classes/UIManager.html#setShowGrid](https://hepsoftwarefoundation.org/phoenix/api-docs/classes/UIManager.html" \l "setShowGrid" \t "_blank)
* Finally, the view options are defined here: