

# Phoenix

## interactive 3D data visualization

### Development of an experiment independent javascript event display framework and data format

#### Exercise for Candidate Students

This document describes an exercise that will be used to evaluate those students interested in applying for the “*Phoenix, interactive 3D data visualization--Development of an experiment independent javascript event display framework and data format*” project, included in the Google Summer of Code (GSoC) 2019 program. The detailed description of the project can be found [here](#).

The exercise is divided into two tasks. In order to complete the first one, the student will need to know and combine a set of technologies that are important for the project: the structure of a web page, the [DOM](#), the [HTML](#) and [JavaScript](#) languages, and the [three.js](#) 3D graphics library.

Please follow the guidelines below to go through the exercise and work at a pace that suits you. **Do not hesitate to ask us, the mentors, any question you might have.**

For both tasks we will want to examine the code, and so we encourage you to use an open git repository of your choice (such as GitHub or Bitbucket).

Files required for the test are available on the following google drive:

<https://drive.google.com/drive/folders/1C8xCo0vUjrvvAOY6NsXTD4yK9pakMTAD?usp=sharing>

(These files are taken from the [TrackML](#) challenge).

## TASK 1 : THREE.JS INSTALLATION

- 1 - We want you to use the tools provided by the three.js library to **load and rotate some geometry**. The geometry is provided in the OBJ format and the file ("Pix.obj") can be found in the Google Drive link provided above.
- 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.
- 3 - As a bonus, make it possible to **select objects by clicking them**, and then **display information** about them in the text overlay.

## TASK 2 : CONVERTOR

Make a javascript function to turn the TrackML challenge data provided in the google drive (the files are called `event000000000000-*`) **into a javascript dictionary**, then use this to use **three.js** to draw a curved line, representing the charged particle trajectory (or "*track*").

You can see this as a two-stage process:

- 1 - You take the parameters of a *track* out of the provided data, and store them into a **custom javascript dictionary**
- 2 - You read data from the javascript dictionary you create for the first step, and you **create a curved line in three.js** out of them, representing the original *track*.

The task involves the **design of a convenient data format** for the javascript dictionary, to efficiently and easily store track data. There is a lot of information in the Kaggle dataset that isn't really required - the minimum is just the points (or "hits") which correspond to each track.

For an explanation of the structure of the TrackML data, please look at information provided on the Kaggle page:

<https://www.kaggle.com/c/trackml-particle-identification/data>