

Where do we all look at in the ESO observatories?

ESOp2.0 - R. THOMAS

This project aims at creating a prototype of 'live' display of ESO telescope during the night. We will start with a single telescope.

You can start with 2D and if you have time go with a 3D representation.

For the telescope:

A queue of observation has been prepared. The queue contains 4 OBs. The OBs are, for the sake of this project, simple configuration file with ra, dec, exposure time (we do not deal with overheads here).

The staralt graph let you show the program of the queue.

1- Let's observe.....

You will run the observation. Meaning that you have to create a program to simulate the evolution of the position of the telescope on the sky every second (or any time interval of your choice). When the first OB is finished, the second OB will start etc....

hint:s -Astropy will help you dealing with the coordinates.

-To make things a bit easier, you can write the position of the telescope in a file and update this file at any time interval of your choice. [open('position.txt', 'a'), to append to a file].

2- Let's live plot!

The idea here is to create a real time animation.

You are going to create a 2D (and 3D if you have time) animated plot that you will update each time a new position of the sky is added to the file you created in 1.

Ideally the plot should have:

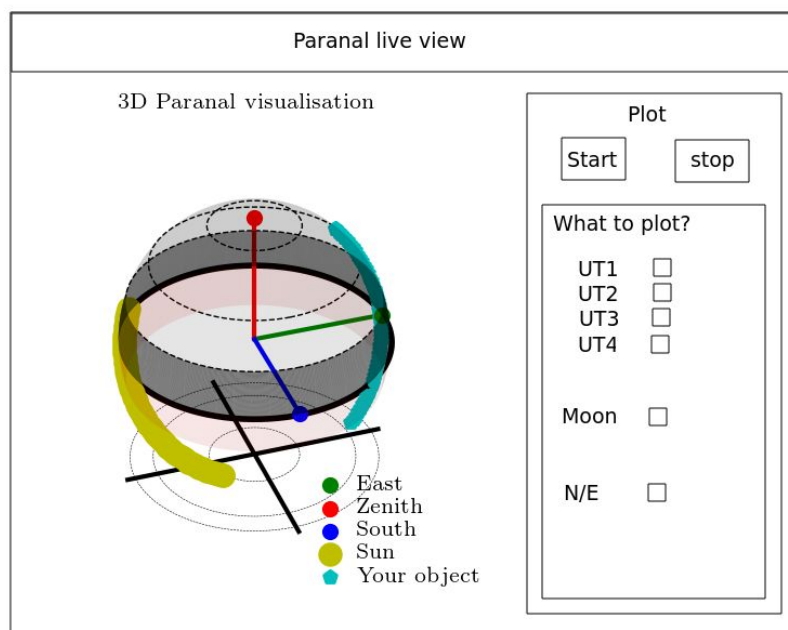
- The real time position of the telescope on the sky. (One point with a well identifiable marker). This is the point that you update at each iteration.
- The full trajectory of the on-going OB (simple line, transparent.)
- The full trajectory of the OB that comes just after.
- The 20 degree pointing limit.
- North and East.
- The moon (optional)

hints: -When you start the program, you are going to enter sort of infinite loop...In each iteration, you will check of the position you are already at is the same as the last position of the position file. If it is indeed the same, you wait 1 sec and have a look again, if it is not, you update the position on the plot.

- From the position you are looking at, you compute the remaining time of the OB and plot the trajectory of the next OB (in a different color).

3- GUI!

We did not see that during the camp but if you would like it you can try to put this in a graphical interface. This interface could like this.



hints: To make this GUI you can use the qt5 library and https://matplotlib.org/examples/user_interfaces/embedding_in_qt5.html which will show you how to embed matplotlib graphs in qt5.