**Completed tasks:**

* **Arduino and Gyrometer Integration**: This feature seamlessly incorporates data from Arduino and a Gyrometer to capture accurate information about the aircraft's pitch, yaw, and roll, enhancing the user's experience with real-time flight data.
* **3D Map Visualization**: The project offers a dynamic three-dimensional map display, providing a visually engaging and immersive representation of the flight environment, enabling users to interact with the surroundings in a lifelike manner.
* **Interactive Marker Management**: Users can easily add or remove markers on the map by clicking on a dedicated marker icon. This feature allows for the precise marking of specific points of interest during flight simulations.
* **Effortless Path Drawing**: The Line String tool simplifies the process of creating paths by enabling users to place points along their desired route. The tool automatically connects these points, resulting in a continuous and clearly defined path.
* **Intuitive Point and Marker Removal**: By simply double-clicking on any point or marker on the map, users can instantly remove it. This feature ensures that unwanted elements are easily eliminated without interrupting the path's continuity.

**To-do:**

* Installed and built openCV successfully
* Installed CMake
* Installed GStreamer (These 3 installations are done but need to make them work)
* Live camera feed to be integrated in the project output
* Code with GStreamer imports ‘gi’ module, works only in MSYS2 MingW64 app
* To run, you need to move into the directory in mingw bash and run:

>> python file\_name.py

**Documentation to refer:**

* How to compile OpenCV with GStreamer Ubuntu & Windows]

>> <https://galaktyk.medium.com/how-to-build-opencv-with-gstreamer-b11668fa09c>

* Qt with openCV [YouTube]

>> <https://www.youtube.com/watch?v=yVeXXuGnnxs>