**ESA: Computer In A Room Challenge 3 (CIARC3)**

**Report PART 4: User Console & Evaluation Phase**

**Team name:** Space Transformers

**Team members:**

ANAGNOSTOU NIKOLAOS

LAPPAS NIKOLAOS

MAKRAS ILIAS

MAVROMATIS MARIOS IOANNIS

VITALIS PETROS

Summary Report: Submit a 2–3-page report that includes:

• The logic behind your solution

1)

2)

• How you prioritized the objectives

1. Beginning with the map
2. The first objective (and if same starting time, then choose the one that ends earlier)
3. Zoned objectives handled first over EB or secret
4. EB afterwards if no zoned objective in queue
5. EB waiting the first pings and if no pings and 1 minute passes, then Daily map for a specific time till the next ping
6. daily map along with secret objective scanning (image recognition algorithm running locally due to limited memory inside melvin)
7. if any errors occur, safety handler runs the daily map routine

• Resource management

1. Optimal orbits for less fuel consumption (vel\_calculation)
2. Often battery checks to make sure no battery runout occurs
3. Estimate when acquisition mode is needed (for picture capturing) and make sure it transitions to that mode with as much battery as possible (needed for the task)

• Functionalities implemented in the user console

1. Ηλίας …. (δεν ξέρω τι θα βάλουμε)

• Team performance assessment, including strengths and areas for improvement

1. Best implementation = Daily Map routine (Αναγνώ … )
2. Zoned objectives with enough area = Well structed and handled (like the first aurora)
3. Multithreading logic and automated submissions …
4. Zoned objectives improvements =
   1. handle better the smaller than lens areas, and take into consideration the coverage\_required …,
   2. implement the daily’s map routine, an array of bits, displaying the zoned objective area for optimal capturing technique
5. Beacon routine = when testing we were handling EBs with maximum distance from original point 150 maximum, (variety of orbits, more pings)
   1. needed more orbit changes in order to prune symmetric estimated points,
   2. ignore the area with radius 75 pixel (as center the first estimated point if wrong)
6. Beacon handling helped the daily map, cause while it did not wait for pings, it was capturing images
7. Forgot to implement the border of maximum speed while using narrow lens, and got API errors
8. Exceptions logger did not work with nohup, although we had tested it without it
9. Needed more memory support in order to run the sprite recognition algorithm inside MELVIN and reduce the amount of slots we used

• Diagrams and screenshots illustrating your solution

1. I have two concerning two EBs …
2. Need some from the console
3. Zoned/secret objectives

Code submission:

• Provide all code for both the console and MELVIN

• Ensure the code is well-organized **(using a git versioning tool is recommended)**

• Include **a README file** explaining the code structure and deployment instructions

**Your code and report will account for 60% of your final score in the challenge.**