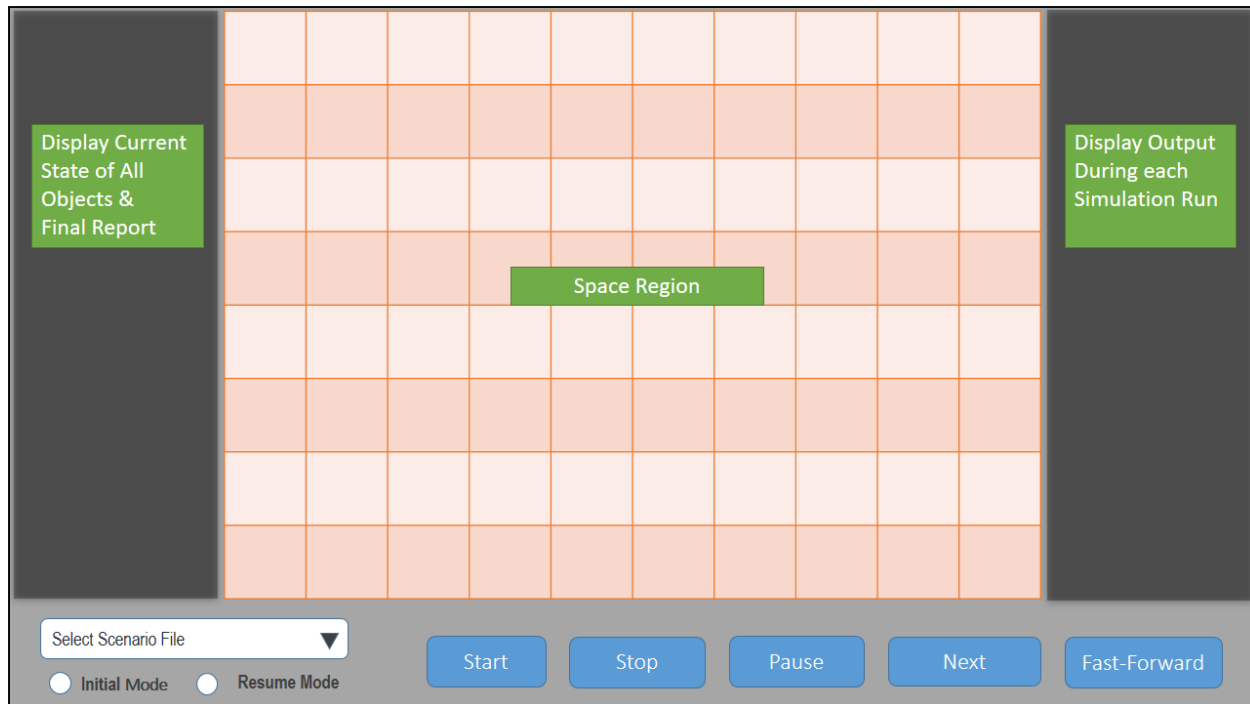


## Proposal for Star Simulation

### 1. First Modification : Graphical User Interface (GUI)

Currently, the system only provides a Command-Line Interface (CLI), users can only track the status of simulation from the text-based output while restraining the establishment of a high-level perspective. Users usually expect to have an easy-to-use and straight-forwarded game interface that they can interact with. Therefore, we proposed a Graphical User Interface (GUI) to offer users an interactive and “user-friendly” system which allows them to get visual feedback for each simulation run. The payoff is that the GUI makes the system easy to operate and more intuitive, while the potential risks are that the GUI design might increase the cognitive loads of the users and exaggerates the potential human errors in the system.

The GUI is a new capability and it will be a web-based application created by using React as a frontend language and Spring Boot for RESTful web services. The main features of the GUI contain display state of the space region and output for each simulation run, provide the final report and add buttons to control the workflow. Figure 1 shows the mockup for this new web application GUI. By using this intuitive interface, users only need to click a button to run the simulator instead of using the command line.



**Figure 1. Mockup For Web Application**

There are two phases of development involved, which needs 15 days of 2 members to implement. In the first phase, we plan to display the state of the space region for each simulator run. The midterm “exam” would be passed if the simulator screen shows exactly the correct state for each simulation run. In

the second phase, the main task is to display output for each simulation run and the final report and add control buttons, such as “STOP”, “FAST-FORWARD” and “NEXT”. The final “exam” is to check if the output is displayed correctly and if the control button works appropriately.

## **2. Second Modification : Data Persistence**

The second brand new capability is to fully stop the system in the middle of the simulation at an arbitrary point and resume the program from the same point as needed. Currently, the system always asks for an input file to begin the simulation. When closing the simulation, no data is preserved, users must start from the beginning to finish the respective simulation.

In order to accomplish the functionality, a *Pause* button will be added. When pressing the *Pause* button, the system saves the current state and time-out the simulation. Besides, two different modes will be introduced into the system, initial mode, and resume mode. If the initial mode is chosen, the user will have to input a configuration file, otherwise, the simulation would start from the last saved state. Please be aware that the users can only choose the resume mode if the pause button was hit before that, unless the entire simulation was completely shut down.

In some cases, users need to keep track of a small step in the middle of the entire simulation, which helps the understanding of the logic and facilitates the further development, or in the other case, it can be a random interruption. This improvement allows them to stop the simulation temporarily and resume later. The payoff is that this improvement expands the sustainability of the program as well as reinforce the controllability for the users. The potential risk, on the other hand, is that it might mitigate its advantage when a different user resumes from the last pause point.

It approximately takes two developers 2 days to complete the new feature. Lower acceptance criteria would be the simulation can only be paused at the end of each turn. Higher acceptance criteria would be the user can pause the simulation at any arbitrary point.

## **3. Third Modification : Fuel & Energy Impacts**

The third brand new capability is to include the fuel and energy impacts for drones. Currently, the drone has unlimited energy to take action as long as it is active. “Recharge” action will be implemented as part of this functionality, which refuels the drones. Drones will automatically take recharge action if running out of energy. Drones will carry a number of units of fuel but not exceeding the maximum number of fuel in the beginning. Each action costs a different unit of fuel, for example, Pass costs 0 fuel, Steer and scan cost 1 fuel, and thrust costs 1,2,3 fuel based on the steps the drones actually thrust.

Users who have the aspirations to realism and want to test the fuel leverage will find this improvement very useful. We will allow users to input the starting fuel units as needed in the configuration phase. The payoff is that this improvement builds up a more realistic system and the potential risk is that the constraint might reduce the seamlessness of the simulation.

It may take two developers 8 hours to complete the new feature. The midterm “exam” is to check if the pure random strategy and user input strategy can handle the fuel and energy impacts, while the final “exam” is to check if smart strategy always offers an optimized solution in the fuel constraint.