# Final Project Phase 4 Requirements

The intent of this phase is to complete Defense Daemon to the best of your team’s abilities. Since the missile defense system is sufficiently complicated, there is no clear delineation marking completion. Your final presentation must address the performance of your system in a quantified, objective way. Your group will have up to 12 minutes for their final presentation. The final presentation should cover the same attributes as earlier phases, but this must heavily focus on the results / performance. Imagine each team has built a missile defense system, and your quantified analysis will serve as the selling points to potential buyers (me) of this system. Whose system would I prefer to go with? Your results ought to speak for themselves – imagine you are trying to sell your solution!

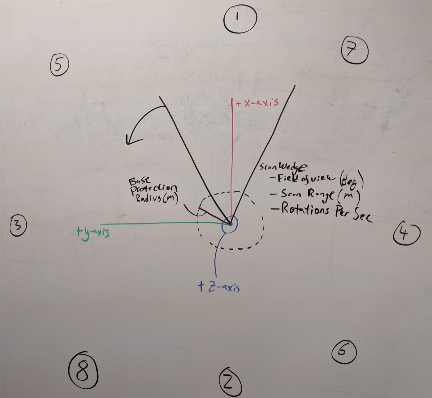
# Total Annihilation

To evaluate your system in a manner similar to your competitors’ missile systems, each team must create a test environment. The test bed, codenamed “Total Annihilation”, is illustrated below.

Key parameters:  
**Base Protection Radius (BPR)**: Distance from origin that must be protected by the MDS (Missile Defense System). Default to 15 meters. The image below shows the radar co-located with the base’s origin, but this is not necessarily the case, and should not be relied upon for your solution to work properly.

**Scan/Sensing Wedge Params**:

1. Field of View: Default to 20 degrees.
2. Scan Range: Default to 200 meters.
3. Rotations per Second: Default to 1 rotation per second – When enabled, the wedge should spin automatically, once per second.
4. Enemy Missile flight time: The amount of time the enemy missile is in flight from launch to impact



During your final demo, you \*must\* be able to dynamically vary the above parameters via your conf file, press CRTL-R, and have the new changes take effect immediately.

In the image above, the circled numbers around your base represent enemy launch missile locations. Different Enemy Attack scenarios will launch a different number of enemy missiles based on the requested Enemy Attack. A keypress should be able to automatically begin each Enemy Attack Phase listed below. Each enemy launch will target a random location \*inside\* the BPR of your base.

1. Enemy Attack Phase 1 – One enemy missile launches from circle 1
2. Enemy Attack Phase 2 – One enemy missile launches from circles 1,2
3. Enemy Attack Phase 3 – One enemy missile launches from circles 1,2,3,4
4. Enemy Attack Phase 4 – One enemy missile launches from circles 1,2,3,4,5,6,7,8
5. Enemy Attack Phase 5 – 16 missiles are launched, each separated by 360 / 16 degrees (22.5 deg)
6. Enemy Attack Phase 6 – 32 missiles are launched, each separated by 360 / 32 degrees (11.25 deg)
7. TOTAL ANNIHILATION 7 – 64 missiles are launched, each separated by 360 / 64 degrees (5.625 deg)

To show the efficacy of your system, you should augment the visualization with virtual indicators that determine when a red missile is, or is not, eliminated. For example, if a red missile is struck by a defensive missile at some location, a semi-transparent sphere placed at the defensive missile’s point of explosion should subsume the head of the inbound red missile’s flight path. The radius of the sphere should equal the defensive missile’s blast radius.

\*Ensure camera placement is optimal to view all inbound attacks (directly overhead is, generally, \*not\* optimal – be creative with how well you visualize the attack).

# Quantitative Analysis

1. To quantify performance, each team must conduct a series of identical tests and provide quantitative performance metrics and qualitative analysis on any shortcomings. Consider the following test parameters.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Rot Per Sec (Deg) (Assume 60 fps) | Wedge Angle (Deg) | Scan Range (m) | Qty Inbound Red | Time of Red Arrival (sec) | Time to Lock (sec) | Successfully eliminated? |
| 1 | 20 | 50 | 1,2,3,4 | 1,2,3,4,5 |  |  |
| 1.2 |  | 100 | 1,2,3,4 |  |  |  |
| 1.4 |  | 200 | 1,2,3,4 |  |  |  |
| 1.6 |  | 300 | 1,2,3,4 |  |  |  |
| 1.8 |  | 400 | 1,2,3,4 |  |  |  |
| 2.0 |  | 500 | 1,2,3,4 |  |  |  |
|  |  | 600 | 1,2,3,4 |  |  |  |
|  |  | 700 | 1,2,3,4 |  |  |  |
|  |  | 800 | 1,2,3,4 |  |  |  |
|  |  | 900 | 1,2,3,4 |  |  |  |
|  |  | 1000 | 1,2,3,4 |  |  |  |

When an enemy missile is launched, assume they are launched 10 meters beyond the scan range. Since all missiles are launched from the same distance, we will vary the “Time of Red Arrival” to adjust each Red’s parabolic trajectory.

1. Considering the combinatorics above, testing each permutation would require automation. To test the 5 “Time of Red Arrival” scenarios, that would require at least 15 seconds. This would be multiplied by the number of inbound scenarios run, multiplied by the number of scan ranges, etc. The attributes listed in this paragraph would “require” about 11 minutes to complete. If the wedge angle and Rot Per Sec begin to vary, that 11 minutes will quickly grow to 44 minutes if 2 rotations and 2 wedge angles were tested against all previous possibilities.
2. You may alter the above parameters, with justification, to thoroughly quantify the performance of your system – take it to and then beyond, its breaking point.
3. Overall, how well does your system perform?
4. What are your system’s limitations?
5. What would your team advertise your system’s operational capabilities to be?
6. At which point does your system determine you have a sufficiently good “trajectory lock” to launch your defense missile?
7. How did you automate the testing or go about collecting your quantified results?