Michael Telahun

CECS 622- Spring - HW1

Dr. Elmaghraby

Definitions

- **Modeling**
  - Modeling as defined by Oxford is (1) the work of a fashion model (2) the art or activity of making three-dimensional models.
    - The root word model is said to come from the Latin word modulus meaning *to measure*.
  - Italian definition of modello (model): (1) the construction that reproduces, usually on a reduced scale, the exact shapes and characteristics of a work in the design phase, for illustrative/experimental purposes (2) a term suitable for providing a convenient scheme of reference points for the purpose of reproduction or imitations, sometime of emulation.
  - Engineering definition: a representation of something; it captures not all attributes of the represented thing, but rather only those seeming relevant.
- **Simulation**
  - Simulation as defined by Oxford is (1) the imitation of a situation or process (2) the action of pretending; deception (3) the production of a computer model of something, especially for the purpose of study
    - The root word is said to come from the Latin word similis meaning *like* or *fake*
  - One definition I found was to pretend to have a feeling
  - A simulation engineer is described as a person who is responsible for creating virtual models to test various kinds of machines and vehicles in virtual environments
  - Siemens' definition: sets of mathematical equations representing the behavior of the system in a physical domain of interest

I thought these were all valid definitions. Simulation and modeling seems to cover a broad spectrum of applications that can be realized in many different forms. I like the references to art and emulation.

Comparing Real vs. Theoretical Simulation Steps

The published paper I found was from MDPI the title is *Modeling and Simulation of Collision-Causing Derailment to Design the Derailment Containment Provision Using a Simplified Vehicle Model*. The paper addresses the issue of train derailment, specifically in South Korea, but mentions statistics and regulations from other countries. In brief their contribution is related to building a simulation framework that takes advantage of simplifying models. This can drastically lower the computational cost of performing a simulation while still giving meaningful results. One of these simplified models is related to the frame for a train car which could need many displacement vectors in several directions. The ability to reduce the computational complexity for this task is very valuable as the train they are trying to simulate could several cars long or just a few, traveling very fast, and moving in 3D space.

The people who worked on this research appeared to address nearly all of the simulation steps described in the book. They explain the problem and explain how they formulated some of their ideas for reducing the complexity of modeling. They discuss the different evaluation metrics they wanted to gather by doing these simulations and therefore had clear goals. The data collected was from several sources including the South Korea and UK. They conceptualized their models based on real world discrete events and the changing nature of the environment a train moves through. The model translation, verification, and validation were done using data from real train data and a train test track used against their simulated models. They discuss the results they generated for the vibration frequency of the wheelbase with the different trials on their test track. They were able to run tests on their track for derailment. The last few simulation design steps were not addressed as well as the others. It is understandable as testing real train derailment to capture test data would be very expensive, but the authors mention this in their conclusions. They mention that they superficially want to perform more tests to further refine the design of the simulation and to evaluate design loads at different locations of the train car. In my novice opinion I think they did a really good job of following simulation guidelines.

3D Pursuit of Two Spaceships

To begin I should say I took this opportunity to develop my simulation in mostly JavaScript. I did end up using HTML to give a very basic visual display of the output. I did this mostly to improve my JavaScript. I have included the simulation files in the hw1 folder you can run it just by opening the html file in a browser. A printout of the simulation is in figure 1.

The simulation program I wrote was designed to perform a chase based on random movements of a ship. My idea was to simulate a scenario where the two ships were battling because it would seem if two ships were fighting in space since there is much more freedom to move around fighting back would be a viable approach for the “mouse” in the scenario. Because I made this assumption I used random coordinates every time a ship moved to simulate the idea of the “cat chasing a mouse who is trying to run away” and “a mouse who is trying to maneuver away from a cat to counterattack or counter pursue”. Since both had lasers, they both can destroy each other. Random coordinates were used because they to someone who is running away it may consider a strategy to trick the pursuer into thinking you are frantically trying to escape without a plan. Although this is genuinely not the case in my simulation the idea was to model this characteristic of “seemingly random behavior of the enemy”.

The hints provided for the two-dimensional distance and conversion functions for the angle between cat and mouse locations were very useful. I found three-dimensional equivalents for these equations and then adapted them to the model I wanted to create. I had to do the cylindrical calculations from my Cartesian coordinate system which was 10x10x10. This space was chosen because it was small and easier to manage. I believe that linearly increasing the size of the space would have a linear effect on the distance and angle. In my tests changing the space to 1000x1000x1000 would reduce the angle size by about 100 and increase the distance by a similar factor. The difference in my simulation would be a matter of picking different distance and angle values for an attack to occur. In my simulation I set three levels of conditions for a successful attack. The first criterion was to have a distance less than or equal to five and an angle less than or equal to three. The second criterion induces more randomness to my model as someone trying to escape yet attacking would feel a lot of pressure firing, perhaps their only shot. The ship firing must roll a 1 in range {1, 2 ,3} in order to successfully attack the other ship. The last condition was to remove the roll if the attacking ship had a distance less than three and angle less than two from the other ship. This is to an assumption in my model that if the attacker was close and had a good line of fire, they would probably hit the other ship.

To perform calculations, I converted from Cartesian coordinates cylindrical coordinates first. I did my distance calculations using a form of Euclidean distance. The equation I used to find the angle was done using the difference in each dimension and then solving for the angle α:

$$\alpha = \text{atan2}(\text{y}, \sqrt{x^2 + z^2})$$

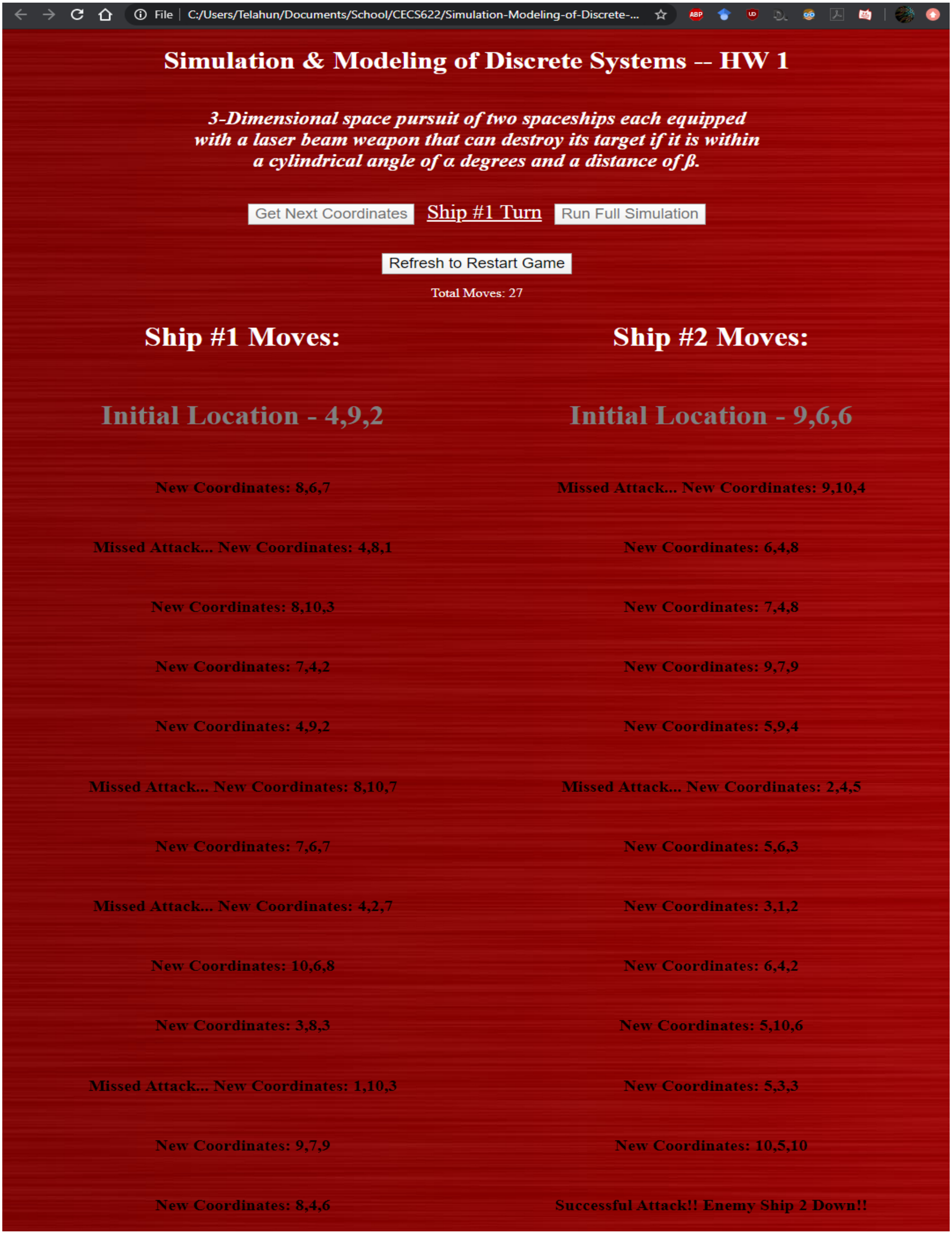

Figure 1. Execution of the simulation for 3D pursuit of Two Spaceships