CSCI 2081 Introduction to Software Development - Fall 2024
Homework 05 - Enhancement

Figure 1 - Enhanced Particle System.

Due Date: Wednesday, December 11, 2024 @ 11:59pm (**No grace period** because runs into finals week - This assignment will be graded more like a lab).

Instructions: This homework may be done together with your lab partner (Add partner name and X500 to README.md). You are to decide on one of the enhancements listed below and implement it.

Getting Started: Before you begin this assignment, be sure to become familiar with the following course content:

- **Video Tutorial:** Lecture 28 - Web Services
- **Video Tutorial:** Lecture 31 (Part A) - Sorting
- **Video Tutorial:** Lecture 31 (Part B) - Using Databases
- **Video Tutorial:** Lecture 33 - Docker

Submission: Your submission is entirely on github (code, documentation, branches, pull requests). You will check off your submission with a TA (similar to checking off a lab).

Rubric: You will check off each milestone **with a TA** to get your grade.

- **Milestone 1:** 50%
- **Milestone 2:** 80%
- **Milestone 3:** 90%
- **Milestone 4:** 100%

Choose one of the following Enhancements:

Docker Enhancement: Create a Docker image and container

- **Milestone 1:** Upload a Dockerfile to Github.
- **Milestone 2:** Release your web service (from Homework 4) as a Docker image on Docker Hub.
- **Milestone 3:** Show a TA that your web server is running on a container.
- **Milestone 4:** Show a TA that your Java Particle Simulation uses the container.

Database Enhancement: Use a Database instead of a CSV file.

- **Milestone 1:** Create a database obstacle table that has the same columns as the CSV file for Homework 4.
- **Milestone 2:** Add data to the obstacle table.
- **Milestone 3:** Show that you can add and remove data from the obstacle table using the web service.
- **Milestone 4:** Show that your Java Particle Simulation uses the database table.

Web Client Enhancement: Create a new web client.

- **Milestone 1:** Create a HTML client that allows users to create obstacles (e.g. typing in locations or using web interaction). You may implement this using any web based framework (e.g. React, Angular, WordPress, etc…)
- **Milestone 2:** Add a CSS file to format the HTML. (Or show how the CSS works in your framework)
- **Milestone 3:** Add a javascript file to interact with the web service (Homework 4). Use Postman to generate the javascript code. (Refer to Lecture 32 - User Interfaces for how to click buttons, etc… You may also need to do some other research). (Or show how to call the web service from your web framework)
- **Milestone 4:** Show that obstacles added through the HTML also show up when running the Java Particle Simulation.

Particle Collisions Enhancement: Add efficient collisions

- **Milestone 1:** Before each frame, “sort” the particles by some useful criteria (e.g. by x position, by y position, both, or some other metric - Another idea is to partition your particles into a grid and look at nearby particles). Any type of filtering optimization is fine here (have a way to find nearest neighbors).
- **Milestone 2:** Create an adapter that wraps a particle to treat it like an obstacle (e.g. use the adapter pattern).

- **Milestone 3:** When testing for collisions, use the particle's nearest neighbors (e.g. look to the right and left of the sorted list to limit the amount of possible collisions). Use the adapter from Milestone 2 to check collisions for the filtered list.
- **Milestone 4:** Show a TA that your obstacles bounce off each other.

Testing Enhancement: Test your application

- **Milestone 1:** Create JUnit test classes for the following classes: Obstacle, ObstacleNode, Particle, and Simulation.
- **Milestone 2:** Write at least one unit test for every method that doesn't involve another class.
- **Milestone 3:** Write three or four integration tests for more complicated methods that involve multiple classes.
- **Milestone 4:** Show a TA how your tests can serve as regression tests. (E.g. show how adding potential bugs into the system might make the tests fail)

Other Enhancement: Please reach out to the instructor team (csci2081_f24@umn.edu) if you want to pursue another software development skill.