

CSCI 2081 Introduction to Software Development - Fall 2024

Homework 02 - Interactive Particles

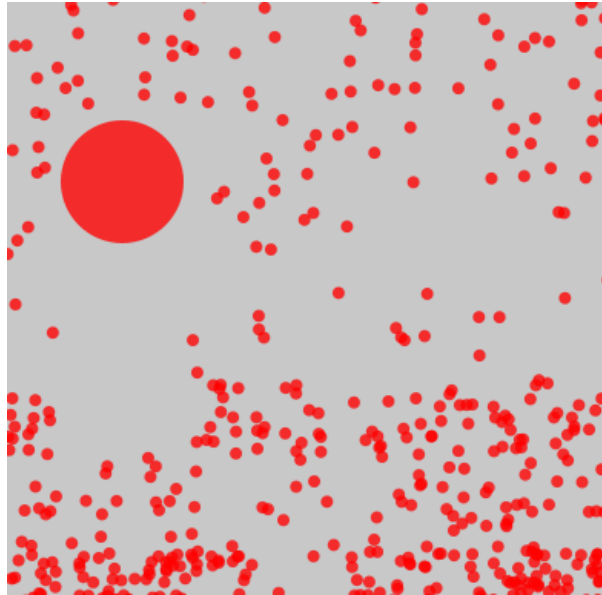


Figure 1 - Interactive Visualization of Particle System

Due Date: Sunday, October 13, 2024 @ 11:59pm

Instructions: This homework assignment enables us to interact with our particle simulation through computer graphics. Your task is to create an interactive visualization that allows the creation of particles and obstacles within a Processing program. The main goal is to manage dynamic particles and obstacles. Here are the tasks:

- **Interactive Visualization** - Learn how to draw particles and interact with a simulation using the keyboard and mouse.
- **Dynamic Particle Array** - Dynamically add / remove particles using a dynamic array inside a simulation. When a user presses “+” on the keyboard, 50 particles are added to the system. When a user presses “-” on the keyboard, 50 particles are removed. The array should double and copy every time more space is needed.
- **Obstacle Linked List** - Maintain a linked list of obstacles. Every time a user left clicks a new obstacle is added to the linked list at the mouse position. Whenever a user left clicks on an obstacle, it is removed from the linked list.

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

- **Video Tutorial (Lecture 11 - Interactive Graphics (Processing Tutorial)):**
https://umn.zoom.us/rec/share/I94U8IVfbbIpmLI0x1xU8593h1pbMpXPXZBqXZwAcIxB3sO4kENrhahdRfiVAZ6s.iET_DzJ4vJ3ZbJMt - This will help you get started with the visualization.
- **Lecture 10 - Lists (Slides)** - This lecture will give you a background of how to work with dynamic arrays and linked lists.

Submission: Once complete, submit all your java files (Simulation.java, Obstacle.java, Particle.java, ObstacleNode.java, Vector2DMath.java), and pde (main.pde - Processing sketch file) files for your solution to Gradescope. You may submit as many times as you like.

Goal: Learn how to implement and use a dynamic array and a linked list in the context of interactive visualization.

Grading Rubric: The grading rubric will be released Wednesday, October 9, 2024.

Points for this assignment will be awarded based on the relative importance of each topic, as outlined below. (High to Low).

- **High:** Dynamic Array
- **Medium:** Linked List
- **Low:** Visualization

UML Diagram of our Design

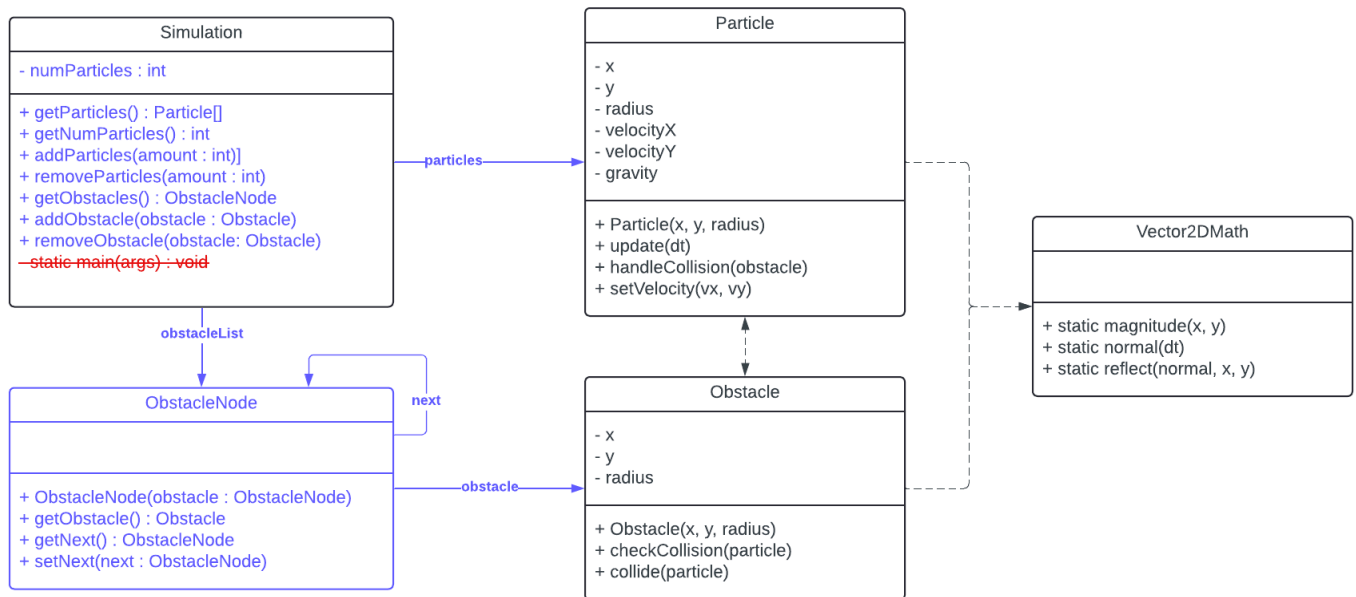


Figure 2 - UML Class Diagram showing our proposed solution. **Items in red have been removed from Homework 1.** **Items in blue have been added from Homework 1.**

Task 1: Dynamic Particles

Task: Create a user-defined dynamic array of particles (e.g. don't use ArrayList or other collection).

- Study the UML above and modify your simulation class from homework 1 to hold an array of particles.
 - Remove the static void main(...) method from the simulation class as our new entry point will be from Processing.
 - Add an array that holds a fixed number of particles.
 - Add the getters and add / remove methods.
- In the addParticles(...) method, if the array is too small you should do the following (example in Lecture 10 - Lists).
 - Create a new array twice the size.
 - Copy the old array into the new array.
 - Set the array to the new array.
 - Modify the numParticles.

3. `removeParticles(...)` should reduce the number of particles. For this homework you do not need to reduce the size of the array.

Task 2: Obstacle Linked List

Task: Your task is to create a user-defined obstacle linked list (e.g. don't use `ArrayList` or other collection).

1. Create the `ObstacleNode` class and implement the obstacle methods in the simulation and the `ObstacleNode` class.
2. In the `addObstacle(...)` method, add the obstacle to the end of the list.
3. In the `removeObstacle(...)` method, find the obstacle in the linked list and remove it.

Task 3: Visualize the Particles!

Task: Your task is to visualize the particles using Processing

1. Refer to the video lecture (Lecture 11 - Interactive Graphics (Processing Tutorial)). It gives you a start on how to draw particles and obstacles.
2. Create a new processing sketch. Inside the sketch create the `setup()` and `draw()` functions.
3. Create a `Simulation` object. Inside the `draw()` function call `simulation.getParticles()` and update each particle. Then draw each particle.
4. Get the obstacle list, and call `handle collision` for each obstacle and each particle.
5. Respond to keyboard input. If "+" is pressed, add 50 particles. If "-" is pressed, remove 50 particles.
6. Respond to mouse input. If the "left button" is clicked, add a new obstacle at the mouse location. If the "right button" is clicked, find the obstacle the mouse is over and remove it.