

Simple Motion Simulator

CS 260 Assignment 1120

Hieu Le

Introduction

This program seeks to simulate the movements and interactions of particles in a two-dimensional bounded region. For simplicity, these particles are restricted to be identical in size while all collisions are elastic and strictly bilateral. User can generate, remove, clear particles as well as control the simulation.

Capabilities

1. User can start, pause or resume the animation by pressing the Start button located on the left of the button panel. User can exit the program by pressing the letter Q.
2. User can remove the last particle to be added by pressing the Remove button located at the middle of the button panel or by pressing the letter R. Removing particle when the region is empty does not throw any error and is simply ignored.
3. User can remove all existing particles from the region by pressing the Clear button located on the right of the button panel or by pressing the letter C.
4. User can add a new moving particle by clicking on the display area. The initial speed of a particle is fixed while the velocity is randomized. Clicking at locations occupied by existing particles is ignored. Particles can be added regardless of whether the animation is running or not.
5. User can specify the velocity of a new particle by dragging the cursor across the display. While the magnitude remains fixed, the direction of the velocity will be aligned with that of the drag.
6. User can uniformly move all particles up, down, left, right by pressing the corresponding arrow keys.
7. Collisions between one particle with another or with the bounding edge are systematically tracked and resolved in accordance with the laws of mechanics.
8. The movements of particles are continually updated between a given time lapse.

Design Patterns

Strategy Pattern

The CollisionResolver interface defines the behaviors of an engine capable of resolving collisions in two-dimensional space. The concrete ElasticResolver defines one such engine, based

on the assumption that all collisions are perfectly elastic and bilateral. The implementation of this algorithm is hidden from the client which is an object of type `Cartesian2D` and is only determined at runtime.

Decorator Pattern

The `ColoredCircle` class enhances the functionalities of an object of type `Circle2D` by allowing it to be colored when drawn. The `ColoredCircle` class accomplishes that via both inheritance and composition. It implements the `Circle2D` interface whilst also aggregating an object of type `Circle2D`.