Aim

To simulate the gravitational interactions between multiple bodies in a 2D space using p5.js, and optimize the simulation with a quadtree data structure for efficient force calculation.

Objective

- Develop a simulation of gravitational forces between bodies using the p5.js library.
- Implement a quadtree for efficient spatial partitioning to improve performance in force calculations.
- Visualize the simulation of body movements in real-time on a 2D plane.
- Explore the impact of quadtree-based optimization on the accuracy and efficiency of the n-body simulation.

Summary

This project involves creating an n-body simulation using p5.js, where multiple bodies interact through gravitational forces. The simulation uses a quadtree data structure to optimize the performance by reducing the computational complexity of force calculations between bodies. The code is structured in a modular way with separate files handling the movement of bodies (mover.js), the quadtree logic (quadtree.js), and the overall simulation setup (sketch.js). The project is executed in a web-based environment using HTML, CSS, and JavaScript.

Tools and Libraries Used

- **p5.js**: A JavaScript library for creative coding, used to create the simulation and handle vector calculations.
- **HTML/CSS**: For structuring and styling the webpage.
- **JavaScript**: For implementing the logic of the simulation, including the quadtree and body movement.

Procedure (Code Explanation)

1. **HTML Setup (index.html)**:
    - The HTML file links to the p5.js library and the custom JavaScript files (quadtree.js, mover.js, sketch.js).
    - It structures the webpage to run the simulation within the browser.
2. **Mover Class (mover.js)**:
    - Defines the Mover class, representing each body in the simulation with properties like position, velocity, acceleration, and mass.
    - Implements methods like applyForce, attract, update, and show to handle force application, gravitational attraction, state updates, and visualization of each body.

3. **Quadtree Implementation (quadtree.js)**:
    - Defines the QuadTree, Rectangle, Point, and Circle classes to create a spatial partitioning structure.
    - The quadtree divides the simulation space into quadrants, recursively subdividing areas with many bodies to optimize force calculations.
    - Methods like insert, subdivide, and query help efficiently manage and search for bodies within specific areas, reducing the number of force calculations needed.
4. **Simulation Setup (sketch.js)**:
    - Sets up the canvas and initializes the simulation.
    - Creates instances of the Mover class and adds them to the quadtree.
    - The draw loop continuously updates the positions of the movers, recalculates forces, and redraws the bodies on the canvas.
    - Utilizes the quadtree to efficiently find nearby bodies and apply gravitational forces.

Highlights

- **Quadtree Optimization**: The use of a quadtree significantly reduces the computational load by limiting the number of force calculations to only those bodies within a specific region, enhancing the performance of the simulation.
- **Modular Code Structure**: The code is organized into separate files for different aspects of the simulation, making it easier to maintain and extend.
- **Real-time Visualization**: The simulation runs in real-time, showcasing the dynamic interactions between bodies as they attract and move according to gravitational forces.
- **Efficient Force Calculation**: The combination of p5.js vectors and quadtree-based spatial partitioning ensures that the simulation remains performant even with a large number of bodies.

Conclusion

The project successfully demonstrates an n-body simulation with optimized performance through the use of a quadtree data structure. By reducing the number of necessary force calculations, the simulation can handle a larger number of bodies without sacrificing speed or accuracy. This project serves as an effective example of how spatial partitioning techniques like quadtrees can be applied to complex simulations to improve efficiency.