Exploration in Magical Particle Systems

Project A report for COMP_SCI 351-2 CG

Haoyu Wei - HWJ9151

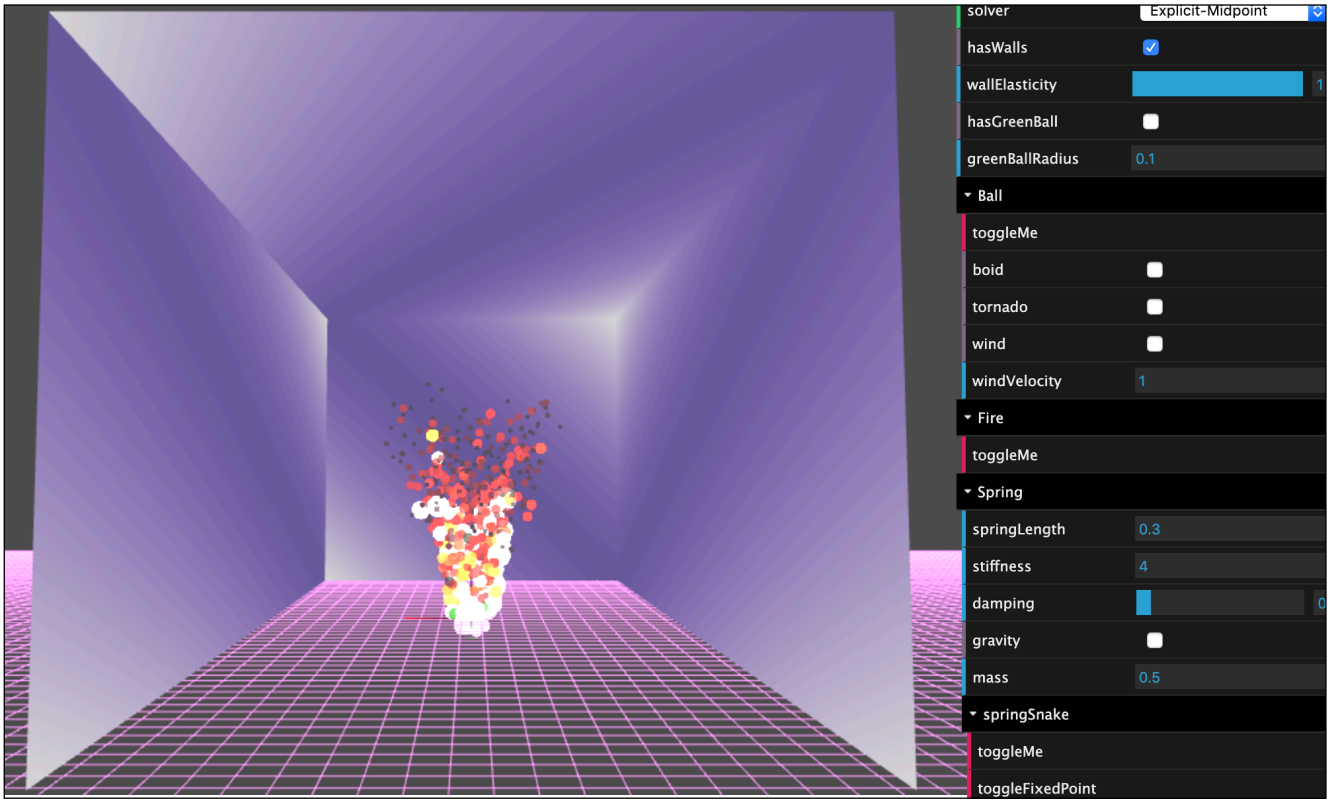

Figure 1. A screenshot of user interface.

Introduction

In Project A, I build a world consists of 4 particles systems: A fire system with 600 particles, A tornado-like force on 600 bouncy balls, Boids particles and A spring system. The boids are also designed to be attracted by mouse controlled green constraint ball while maintaining collision detection as a constraint ball. You can drag the ball and see boids following it. And the spring system contains a spring snake with adjustable number of balls connected by springs and a tetrahedron

connected by spring. And the springs are explicitly drawn to show the force. Gravity and mass of the balls can also be adjusted easily through user control bar.

As usual, the world can be explored by navigating our camera through the scene to view from different angles with key events.

There are two types of constraints implemented in the systems: wall constraints including four removable walls and ground plane, and a ball constraint using one mouse controlled sphere.

I've also implemented 7 solvers: Explicit Euler, Explicit Midpoint, Implicit Euler, Iterative Implicit Euler, Iterative Implicit Midpoint, Velocity Verlet and Semi-implicit solver.

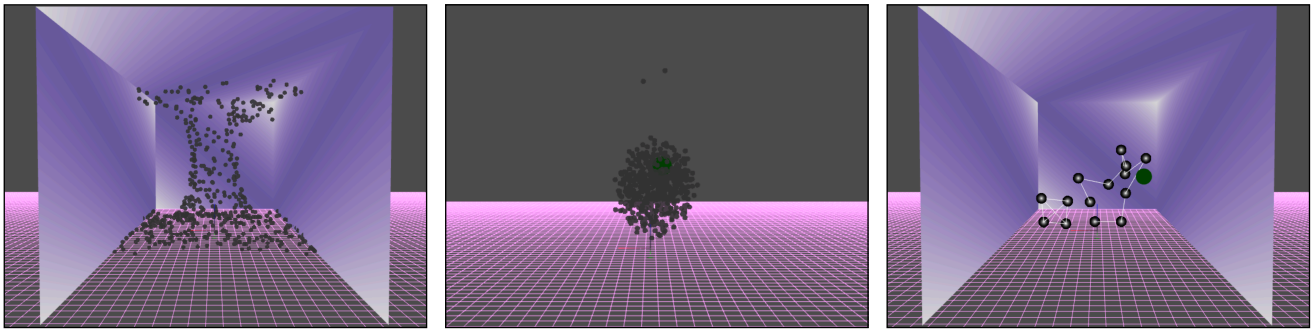

Figure 2. (Left) Tornado of half transparent particles. (Middle) Boid of particles attracted around the green ball. (Right) A spring snake and a spring tetrahedron without gravity.

User Guide

The user guide consists of 5 parts: General control, Ball system control, Fire control, and Spring control.

- Press 'H' or 'Close Controls' to hide the control bar.

General Control:

- Switch **Solvers**: Select from list of solvers in the floating control bar.
- Toggle **Wall Constraints**: Click the checkbox of 'hasWalls' in the control bar.
- Change **Wall Elasticity** (Kbouncy): Slide the bar or input a new value in the 'wallElasticity' column in the control bar.
- Toggle **Ball Constraint**: Click the checkbox of 'hasGreenBall' in the control bar. The ball can be controlled by mouse drag. You can drag it freely everywhere to let it affect other particle's movement.
- Change **Constraint Ball Radius**: Input new radius value in the 'greenBallRadius' column.

❖ **Camera** Navigation: Press 'W''A''S''D' to turn or tilt the camera's aiming direction up/left/down/right without moving the camera. Press ←↑→↓ to move the camera horizontally/vertically without changing the aiming direction of camera. Press +/- to move the camera forward/backward in gaze direction.

Ball System Control:

Reminder: Better not turn on multiple forces together. This will cause over-energetic balls.

❖ **Toggle** Ball System: Show or hide the bouncy balls by clicking the 'toggleMe' button in the folder of 'Ball' in the control bar.

❖ Toggle **Boid** Behavior: Switch on or off boid behaviors by clicking the checkbox of 'boid'. After turning on boid, you can drag your mouse to move the green constraint ball and the boids are all attracted to the ball. See Figure 2 (middle), Figure 4.

❖ Toggle **Tornado** Force: Switch on or off tornado force by clicking the checkbox of 'tornado'. You will see effect like Figure 2 (left).

❖ Constant **Wind** Force: Switch on or off constant wind force by clicking the checkbox of 'wind'. The velocity of wind can be adjusted in the control bar.

Fire System:

❖ **Toggle** Fire System: Show or hide the fire by clicking the 'toggleMe' button in the folder of 'Fire' in the control bar. See Figure 1.

Spring System:

❖ **Toggle** Spring System: Show or hide the spring snake or spring tetrahedron by clicking the 'toggleMe' button in the folder of 'springSnake' or 'springTetrahedron' in the control bar. See Figure2 (right), Figure 5, and Figure 6.

❖ Adjust Spring **Properties**: Adjust spring length, stiffness, damping coefficients and the ball mass by changing the values in the corresponding columns. Turn on or off gravity by clicking the checking box.

❖ **Toggle Fixed Point** of Spring Snake: The spring snake has one ball fixed in a certain position by default. You can unfix it by clicking the button 'toggleFixedPoint'.

❖ Adjust **Number of Balls** in Spring Snake: Input an integer in 'number' column or slide the bar to change. Maximum is 10.

Code Guide

❖ File Structure:

The file structure is in Figure 3. In the js folder, the Javascript files beginning with 'C' are JS classes, where CPartSys.js includes all the functions applied to the particle system. And StateSpaceBouncyBall3D.js contains my main entrance of the project. VBO-Lib.js is a file containing all my VBO boxes.

1. StateSpaceBouncyBall3D.js:

The makeXXX() functions constructs rigid bodies in the system. And the drawXXX() functions correspondingly draw the objects that are made.

The control bar is implemented using Dat.gui, written in main() function in this file.

2. VBO-Lib.js:

Each VBO box has its own shaders and similar functions. In this way, the systems will not mess with others' shaders.

3. CPartSys.js:

In the particle system class, we initiate different particle system using initXXX() functions. For each system, three state vectors S0, S1, SM are initiated with their time derivatives S0dot, S1dot, SMdot, and forces and constraints (walls) are also initiated.

Then every system will go through the functions applyAllForces(), dotMaker(), solver(), doConstraints(), drawMe(), and stateVecSwap(). These functions are called by draw() functions in each VBO box.

Each function is well commented in the code.

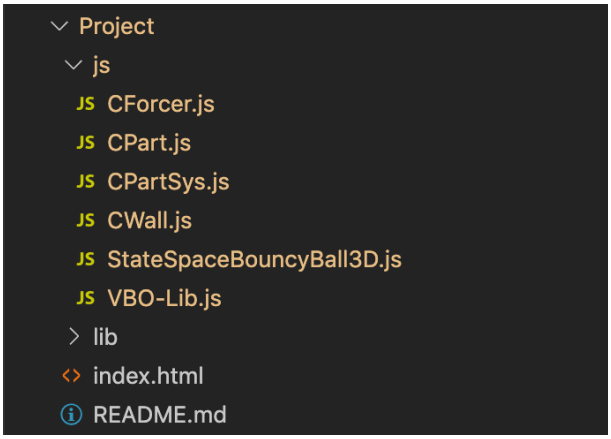

Figure 3. File Structure.

Result

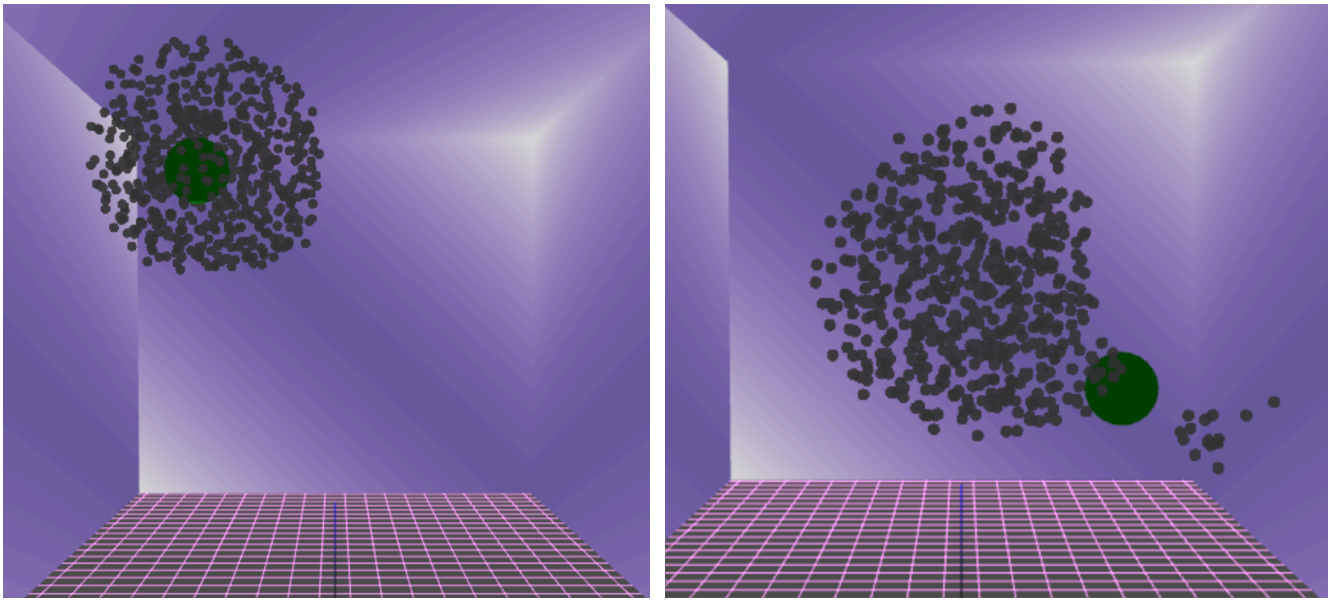

Figure 4. (Left) Green Ball on the left. (Right) Green Ball moved to the right.

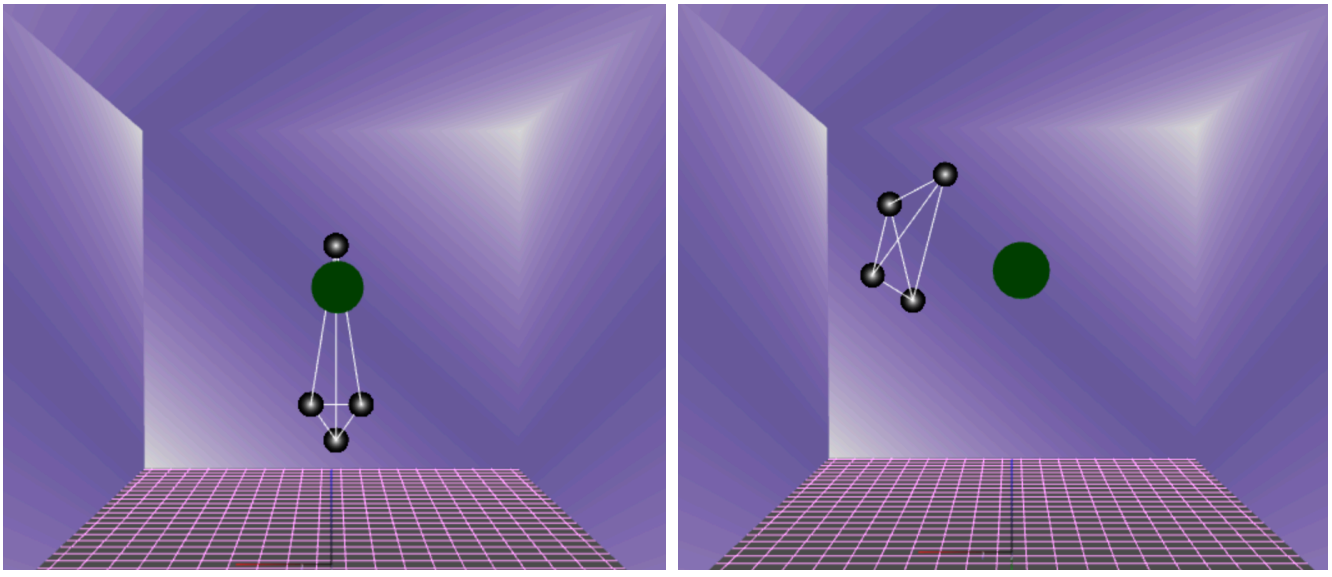

Figure 5. (Left) Spring Tetrahedron with gravity. (Spring length: 0.2, stiffness: 12.7, damping: 0.1, mass: 0.27). (Right) Without gravity.

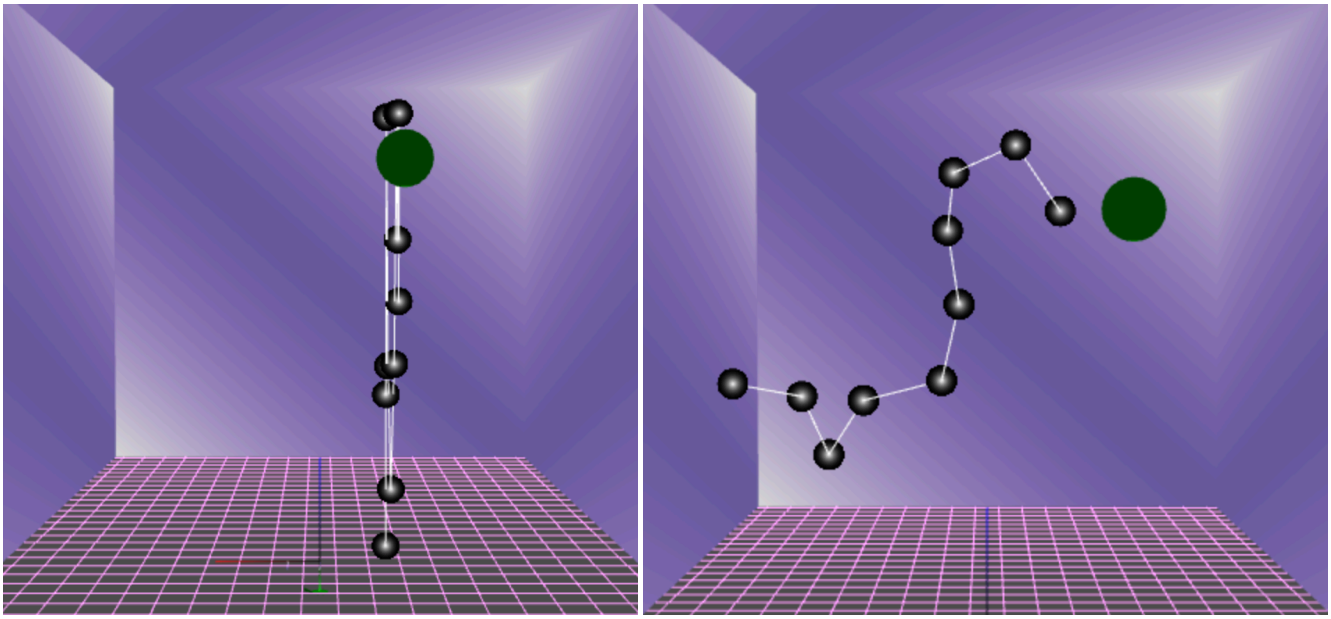

Figure 6. (Left) Spring snake with gravity. (Spring length: 0.2, stiffness: 12.7, damping: 0.1, mass: 0.27). (Right) Without gravity.