Project A: Tornado, fire, flock and pentagonal dodecahedron

Name: Yipeng Pan
NetID: YPK6670

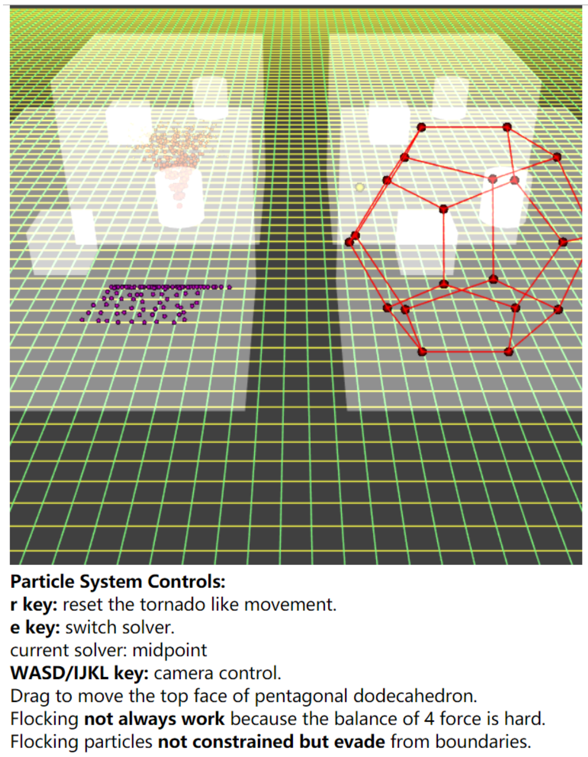
Instruction:

There are four particle systems in the world and constrained by four big cubes and small cubes and cylinders inside. The four systems are tornado, fire, flock and spring.

User Guide:

Press r to start/reset tornado, and drag mouse to move the top face of the pentagonal dodecahedron. Use WASD and IJKL to control the camera. You can press e to switch between different solvers.

Pictures:



Particle System Controls:
r key: reset the tornado like movement.
e key: switch solver.
current solver: midpoint
WASD/IJKL key: camera control.
Drag to move the top face of pentagonal dodecahedron.
Flocking **not always work** because the balance of 4 force is hard.
Flocking particles **not constrained but evade** from boundaries.

Figure 1

Figure 1 is the initial scene, tornado system at back, right side is not started yet, fire system is at back, left side. Flock system is at front, left side started to move automatically, and spring system is at front, right side.

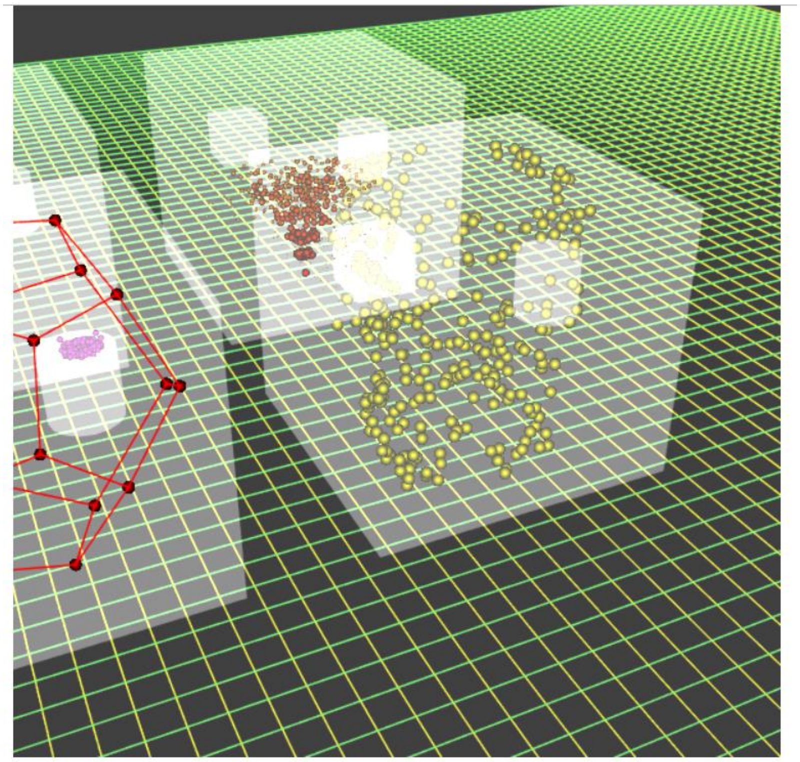


Figure 2 tornado system

Press r on the keyboard to start the tornado system.

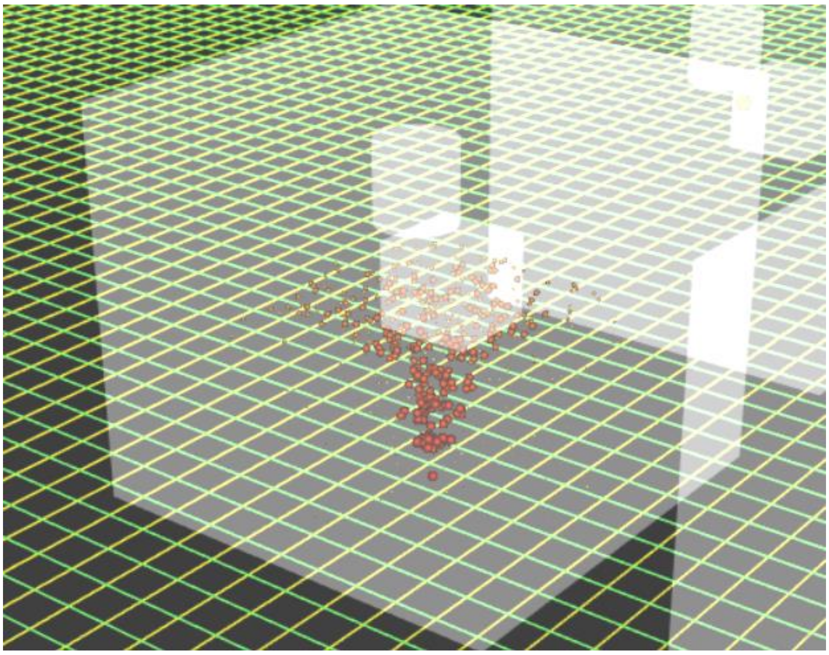


Figure 3 Fire system

Color, mass and size of particles change with time.

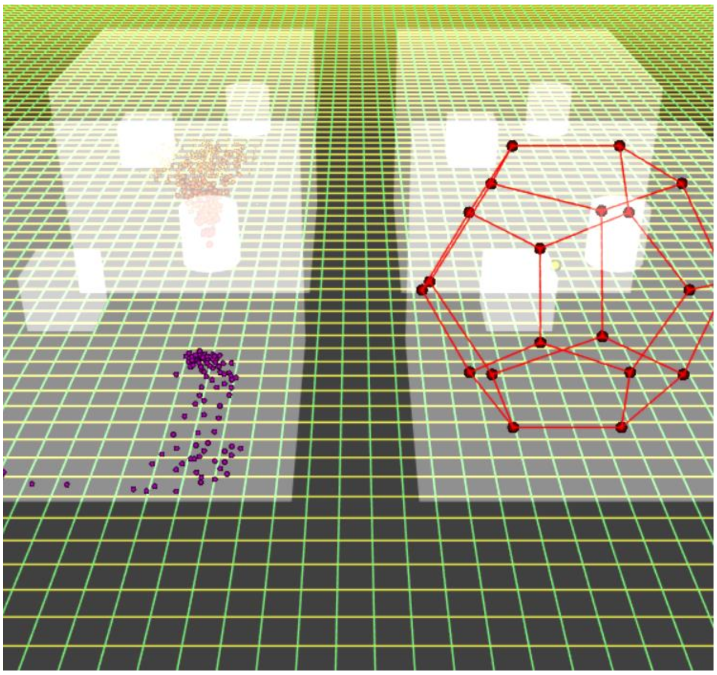


Figure 4 Flocking system

Flocking system not always works, because the balance of four kind of force is too difficult, particles may stopped by separation and gather together but reject to move because of cohesion. Just refresh it! Particles here is not bounce back when hitting the wall by constrain, but evade the wall and change the direction.

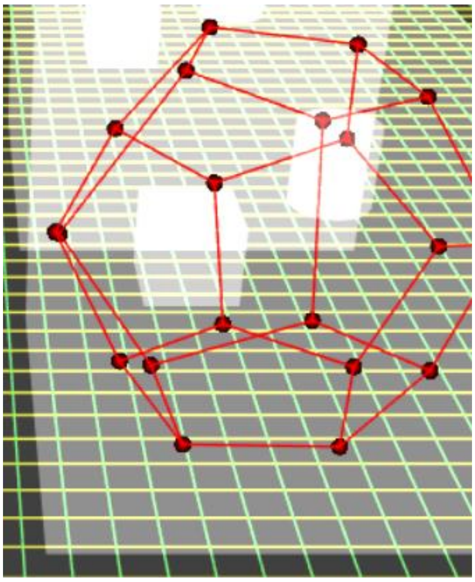


Figure 5 Spring system

Drag on the canvas and release your mouse, you can see springs works. However, even with ‘stable’ solvers, it will still explode if you move it a lot. Of cause, unstable ones will make it easier to explode.

Code guide:

I have a class for particle system called PartSys and it contains most of implementations for these four systems.

For tornado system, the force of tornado is generated by PartSys.portotype.Wind.
For fire system, the reset of died particles and the change of color, mass and size is simply assigned in draw function but not an independent function.
For Flocking system, I have four functions, PartSys.portotype.seperation, cohesion, alignment and evasion representing four forces.
For Spring system, I made a function called PartSys.prototype.springForce and deal with everything about spring.

Besides, dotFinder, swap, solver and doConstraint is shared, midpoint is adopted if current solver is midpoint or explicit midpoint. ApplyForces is only called when starting or resetting tornado system.

I have solver, solver1 and solver2 in the script, but they are almost the same. Solver1 is for flocking system because no gravity and drag in flocking system, and solver2 is for spring system with no gravity.