Project A: Particle Systems

Yifei Ge Net ID: ygt052

User's Guide

Please use WSAD to move the camera in four directions, Q for higher and E for lower.

Please use IKJL to for turning the camera UP DOWN LEFT RIGHT.

Please use the Set String Position input area to relocate the string and press submit to watch it bounce.

Please use Frame Rate input area and submit button to reset the frame rate. The highest effective frame rate differs from computer to computer.

Please use the Multiple Solver buttons to set different solver types.

This program allows 6 different solver types to be chosen, they are Explicit and Implicit Euler, Explicit and Implicit MidPoint, Adams-Bashforth and the Verlet. Each solver behaves differently; the Explicit Euler and Explict MidPoint solvers are not stable.

The available constraints are the ground, red wall, yellow pillar, and the blue ball.

Please use N to create a force field to attract free particles to a left point of the screen.

Please use n to create a counter direction force field to create a boom like effect.

Code Guide

Key data structure:

Five Float 32 Arrays:

s0[]: The current state of all particles.

s1[]: The next state of all particles.

sp[]: The previous state of all particles.

gndVerts[]: All information to create the ground.

cube []: All points to create the walls and the pillar.

spheVerts[]: All information to draw the ball on screens.

Key Functions:

Draw function: This function also works as solver function, calls the four apply force functions, and applies the selected solver using s0[], s1[] and sp[], and check for all particle constraints using wall1() wall2() and wall3() function, and then puts the final result to the s0[], finally calls the PartSys_render function to draw everything.

Wall 1, wall2, wall3 functions: This three functions work as the constraints for all the particles, wall1() creates the solid four red wall effect and the ground. Wall2() makes the pillar constraint. Wall3() functions the blue ball. All constraints resist all particles to get through and reflect them, all wall effect directly modifies the s0[] to show.

Applyforce1() Function: This function serves the first 3000 particles, these particles are initially render randomly in the sky and will fall in the effect of gravity like a rain. This function also create the force filed at the center of a left corner of the ground, when user press N, create an attraction force field and when user press n, create a repulsive force field. The force field works simply, affects all particles, and the direction of the force is to the center or counter the center of the force field.

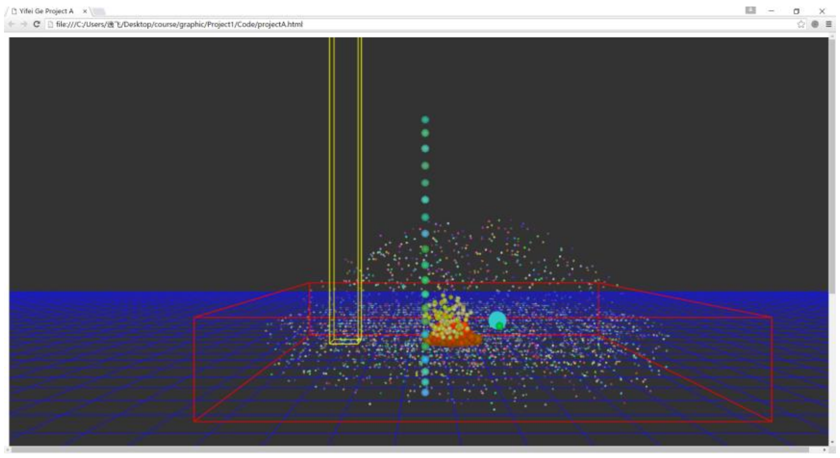
Applyforce2() Function: This function serves the next 3000 particles; these particles create the red fire effect. These particles are render randomly around a center point at the ground, and besides the gravity, they are also affected by a force upwards and a little force to the center line of the fire. These force will make particles raises and gather to the center like a fire shape. These particles will burn out their mass during their life time, and the mass will not only effect the force effect but also effect the particles' color. The less mass they have, the faster they move and the redder the become. As long as the mass is burnt out (less than 0.1), I assume these particle dies and recreate a new one at the original state (randomly on the ground, not too red and has full weight).

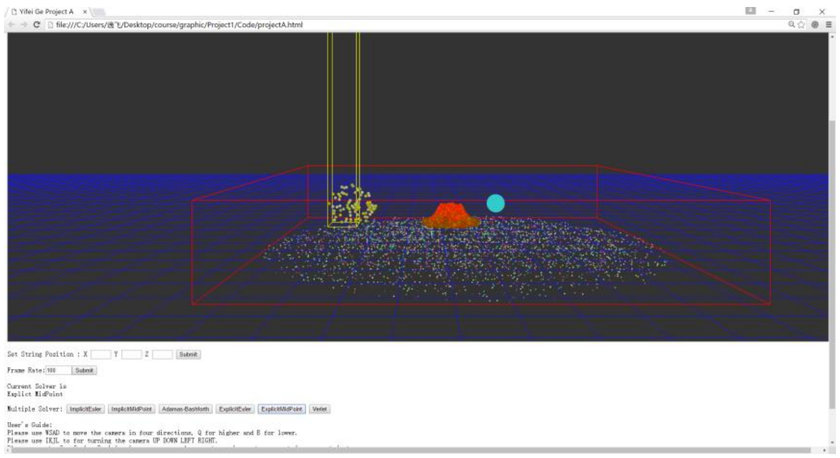
Applyforce3() Function: This function serves the next 90 particles for the boids effect. These particles are yellow and follow the leading bird, a bigger green particle. They are effected by a force to the center of these particles, which allows all particles moves together which is "cohesion". And for each particle, it continually calculates the distance with all other particles in the boids system, as long as the distance is too close (), it will be effected by a force at the counter direction, so they will not crush together, and this is the "separation". The "alignment" works as the leading bird has an attraction to all other particles, so they will follow the movement of the leading bird. "Evasion" shows on the blue ball, as long as a particle in the systems get too closer to the blue ball, it will get a force that push it far from the blue ball, and this function let the whole flocking evasion the blue ball.

Applyforce4() Function: This function affects the last 20 particles in the systems and create a string like stuff. One particle is set to be located in a certain and will not move, just like we hold one side of a string. And the other particles can be replaced arbitrarily to any position with the help of submit () function (this function will relocate the string linearly to the user set positon and then release it, just like we take the other end of the string to a certain place and then release it). All particles in this systems are effected the force between two adjacent particles, this value and direction of each force is determined by the distance between them(F=k(d'-d)).

Results

Some pictures are intentionally capture in a low frame rate, for being able to catch the fast movement.


Yifei Ge Project A
file:///C:/Users/逸飞/Desktop/course/graphic/Project1/Code/projectA.html



Set String Position : X Y Z

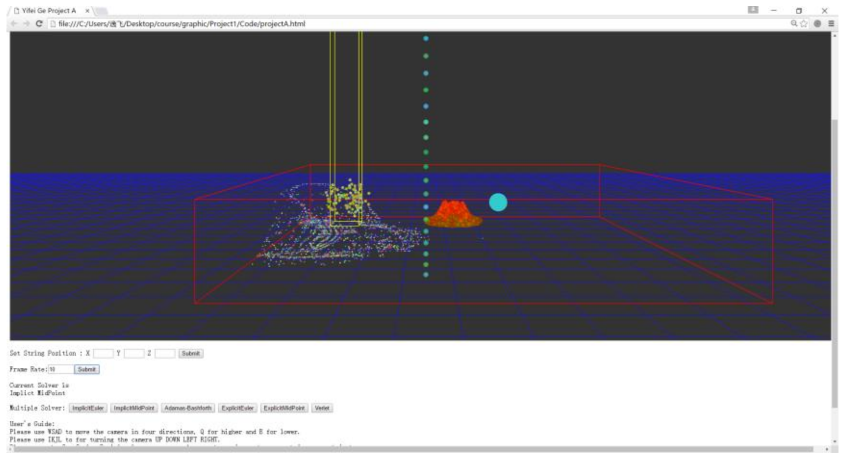
Frame Rate: 100

Current Solver is
Explict MidPoint

Multiple Solver:

User's Guide:
Please use WSAD to move the camera in four directions, Q for higher and E for lower.
Please use IKJL to for turning the camera UP DOWN LEFT RIGHT.

The different solver type, the instable solver will cause the string to explode, only the top point left.



Set String Position : X Y Z

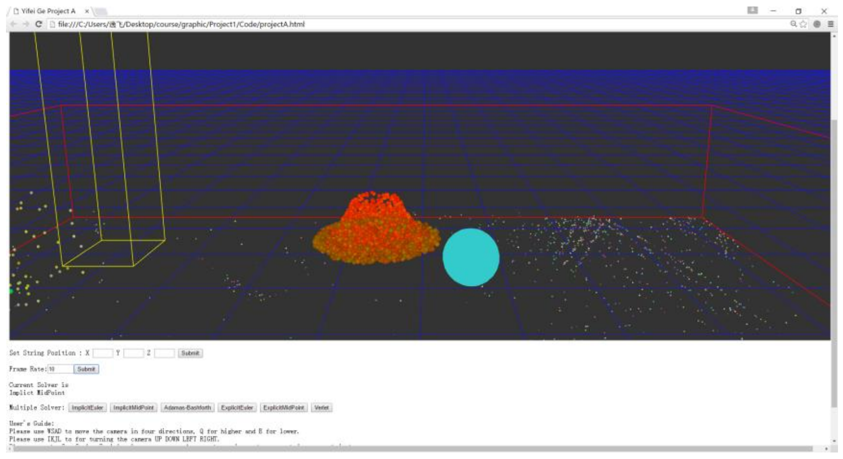
Frame Rate: 10

Current Solver is
Implict MidPoint

Multiple Solver:

User's Guide:
Please use WSAD to move the camera in four directions, Q for higher and E for lower.
Please use IKJL to for turning the camera UP DOWN LEFT RIGHT.

The force field that causing the explode effect, and you can see particles reflect to the yellow pillar and the red walls.



Set String Position : X Y Z

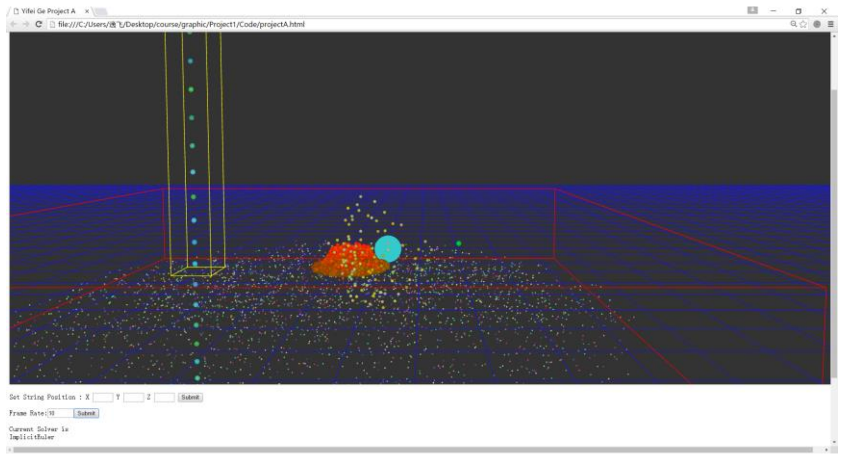
Frame Rate: 10

Current Solver is
Implict MidPoint

Multiple Solver:

User's Guide:
Please use WSAD to move the camera in four directions, Q for higher and E for lower.
Please use IKJL to for turning the camera UP DOWN LEFT RIGHT.

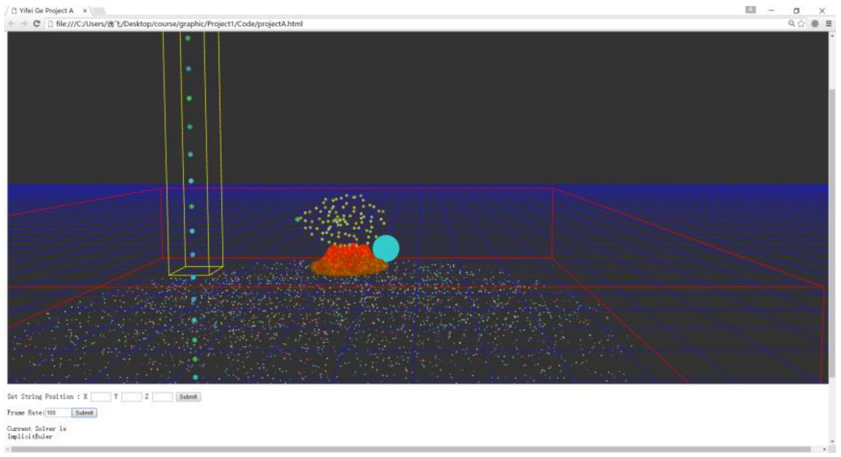
The fire effect, it is obvious that particles change color and diameter during it burns and looks quite like a real flame. This picture also shows that the camera can be reset easily.


Yifei Ge Project A
file:///C:/Users/逸飞/Desktop/course/graphic/Project1/Code/projectA.html
Submit
Submit

Set String Position : X Y Z

Frame Rate: 10

Current Solver is
ImplicitEuler

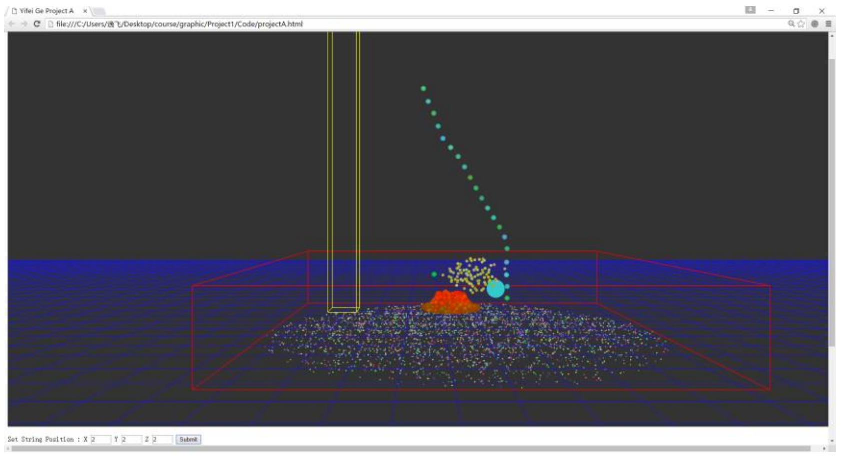

Yifei Ge Project A
file:///C:/Users/逸飞/Desktop/course/graphic/Project1/Code/projectA.html

Set String Position : X Y Z Submit

Frame Rate: 100 Submit

Current Solver is
ImplicitEuler

The boids system, it is clearly the yellow particles are following the green head bird at the same time evasion the blue ball and still keeps in a general shape.


Yifei Ge Project A
file:///C:/Users/逸飞/Desktop/course/graphic/Project1/Code/projectA.html

Set String Position : X 2 Y 2 Z 2 Submit

The string system, you can see the string is pulled to somewhere and now is bouncing back.