Controlled Chaos .pde

이찬, 최윤지, 정승현

documentation

Concept

A partical physics model you can control.

Ideas:

- Gravity.
- Black hole?
- Grid?
- Bouncy walls
- Friction

Parameters

Controllable parameters (From GUI)

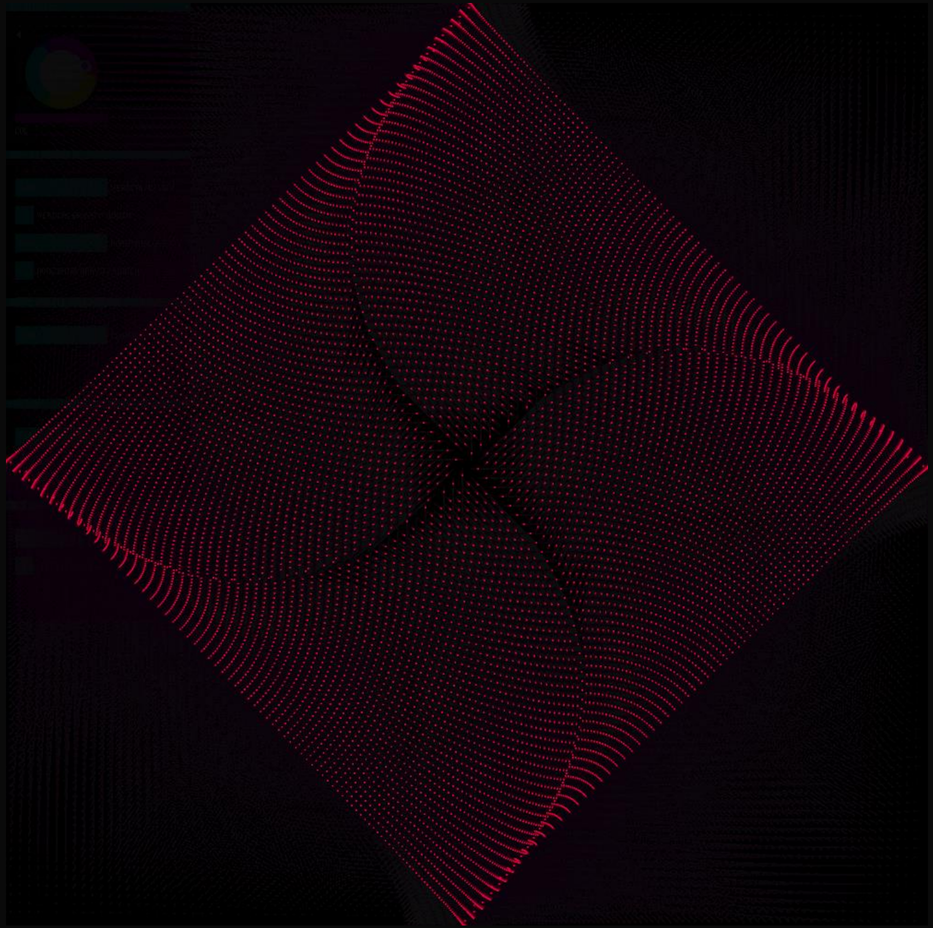
- color col
- float cenGrav
- float verGrav
- float horGrav
- float friction
- int ballSize
- boolean multSpeed

Controllable parameters (Hidden)

- int ballNum (removed from GUI because it was not that useful)

color col

...color.

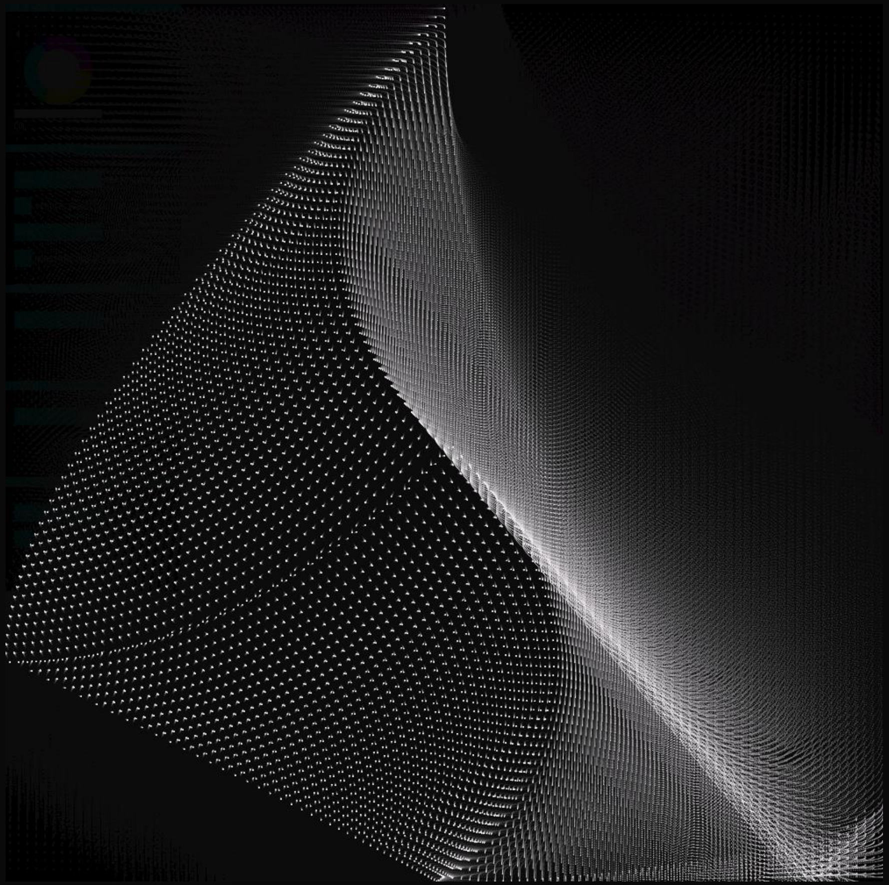


float verGrav, horGrav

Vertical Gravity and Horizontal Gravity.

- Minimum: -1.0
- Maximum: 1.0

- Negative verGrav: reverse Gravity.
- Negative horGrav: leftwards Gravity.
- Positive horGrav: rightwards Gravity.

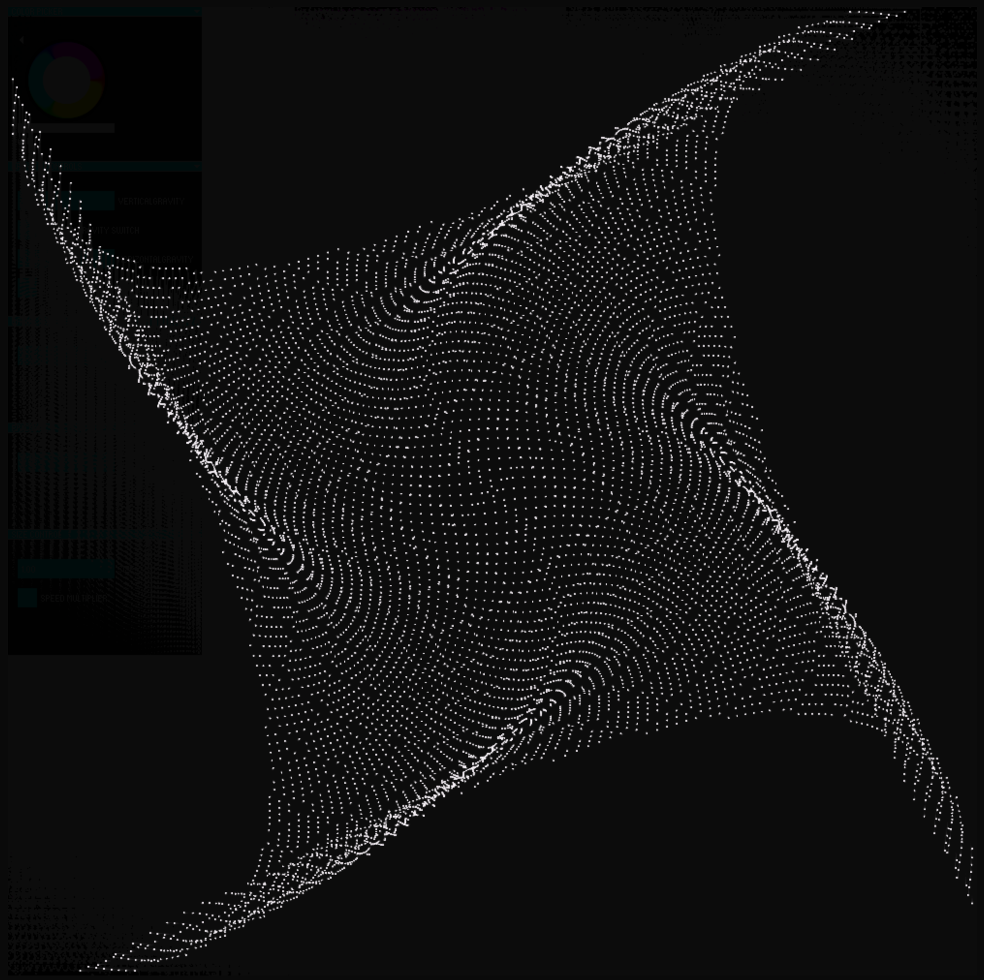


float friction

Friction (how particles slow down).

- Minimum: 0
- Maximum: 0.1

- GUI scales these values from 0 to 1.

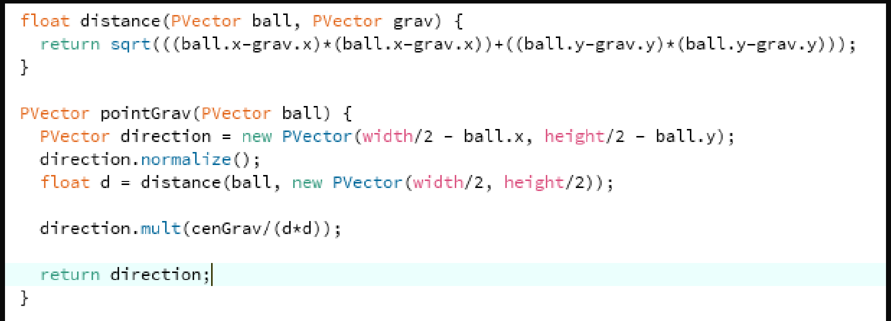


float cenGrav

Single Point Gravity (Experimental)

- Minimum: 0
- Maximum: 1000

- GUI scales these values from 0 to 1.
- Vector normalizing required.
- When a particle hits the center, it gains incredible velocity, which had to be manually limited. Still it is very chaotic, and thus, experimental.

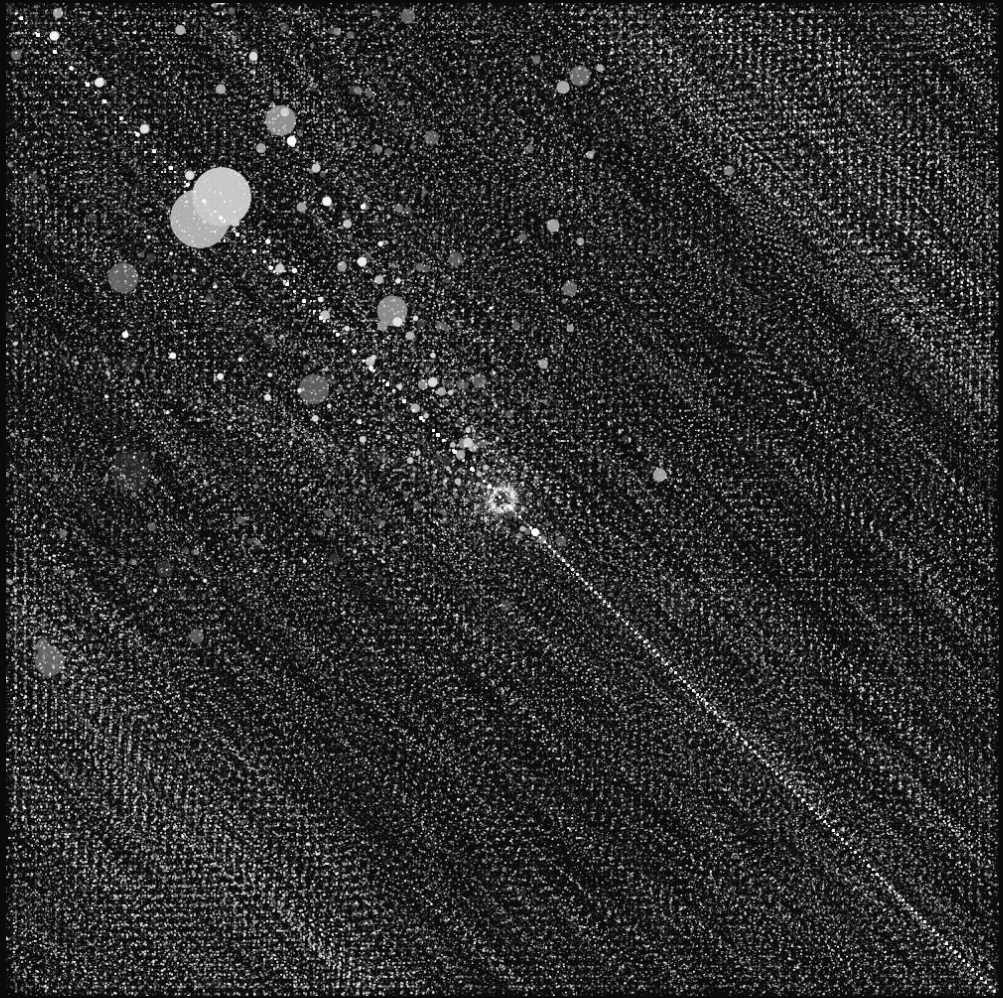


```
float distance(PVector ball, PVector grav) {
  return sqrt(((ball.x-grav.x)*(ball.x-grav.x))+((ball.y-grav.y)*(ball.y-grav.y)));
}

PVector pointGrav(PVector ball) {
  PVector direction = new PVector(width/2 - ball.x, height/2 - ball.y);
  direction.normalize();
  float d = distance(ball, new PVector(width/2, height/2));

  direction.mult(cenGrav/(d*d));

  return direction;
}
```

boolean multSpeed

Multiplies ballSize and velocity to generate the size of each particle.

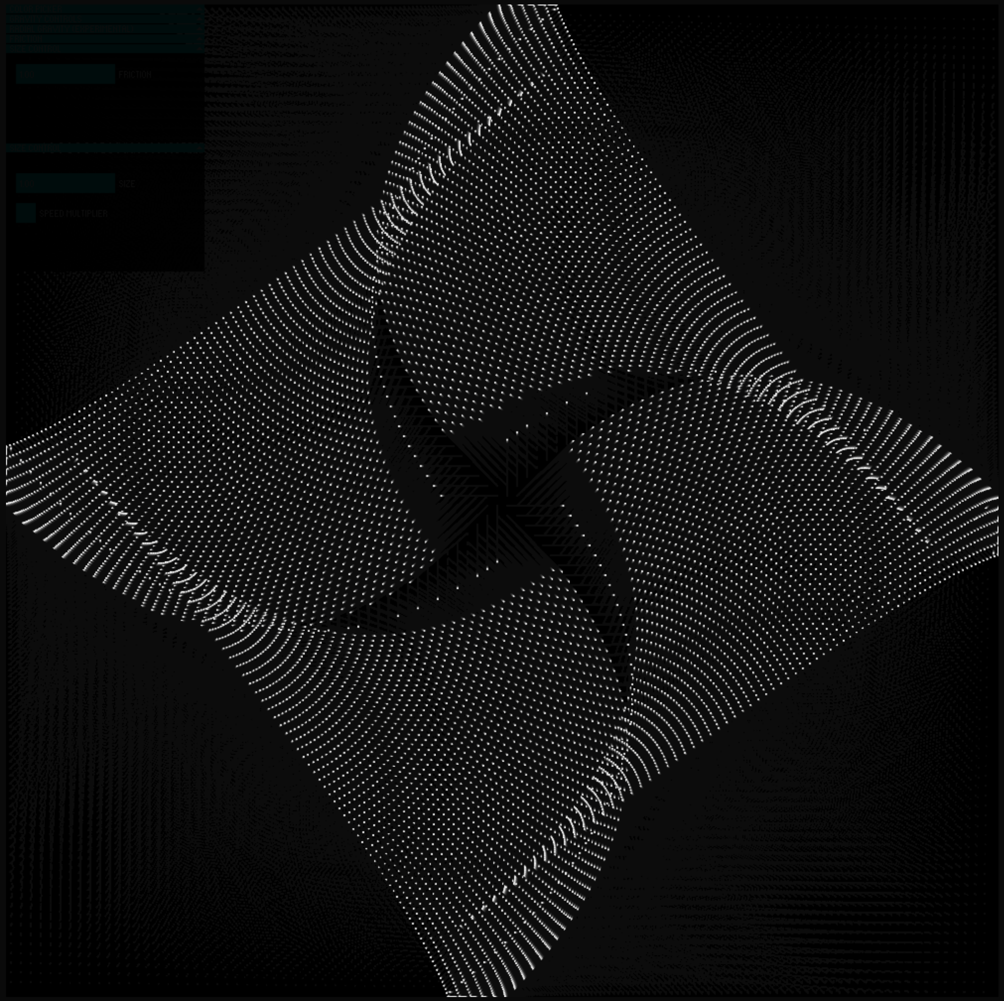
- Fast moving particles are more visible, and particles that do not move are invisible.



Extras

'a' to hide GUI
's' to show GUI
'p' to take screenshot

Any other key to reset particle array.



1.00 FRICTION

1.00 SIZE

Controlled Chaos .pde

이찬, 최윤지, 정승현

documentation