```
//Sky Gastinel
//13 February 2020
var num = 1000;
var vx = new Array(num);
var vy = new Array(num);
var x = new Array(num);
var y = new Array(num);
var ax = new Array(num);
var ay = new Array(num);

var magnetism = 10.0; //Attraction strength Negative value means
repulsion.
var radius = 1 ; //Radius of the circle to draw
var gensoku = 0.95; // Slow down the movement of particles

var ball;
var mode = 1;

function setup(){
  createCanvas(windowWidth,windowHeight);
  noStroke();
  fill(0);
  ellipseMode(RADIUS);
  background(0);
  blendMode(ADD);

  for(var i =0; i< num; i++){
    x[i] = random(width);
    y[i] = random(height);
    vx[i] = 0;
    vy[i] = 0;
    ax[i] = 0;
    ay[i] = 0;
  }

  ball = new Ball(random(0, windowWidth),
random(0,windowHeight/2), 2, 2, 2)

}

function keyPressed() {
  console.log(keyCode)
  if(keyCode === 32 &&
    mode === 1){
    mode = 2;
```

```
  }else if(keyCode === 32 &&
    mode === 2){
    mode = 1;
  }
}

function draw(){

  textSize(20);
  text('Press space bar to make the particles follow your
mouse', 10, 30)

  var positionX, postionY;

  if(mode === 1){
    positionX = ball.loc.x;
    positionY = ball.loc.y;
  }
  if(mode === 2){
    positionX = touchX;
    positionY = touchY;
  }


  fill(0,0,0);
  rect(0,0,width,height);

  for(var i=0; i<num; i++){
    var distance = dist(positionX, positionY, x[i], y[i]);
//dist(x1,y1,x2,y2) //Function to find the distance between two
points
    // Acceleration is inversely proportional to the square of
the distance from the center of gravity.
    if(distance > 3){ //Don't update acceleration if too close
to mouse
      ax[i] = magnetism * (positionX - x[i]) / (distance *
distance);
      ay[i] = magnetism * (positionY - y[i]) / (distance *
distance);
    }
    vx[i] += ax[i]; // Increase speed vx by ax per frame.
    vy[i] += ay[i]; // Increase speed vy by ay per frame.

    vx[i] = vx[i]*gensoku;
    vy[i] = vy[i]*gensoku;
```

```
    x[i] += vx[i];  //Advance vy pixels per frame.
    y[i] += vy[i];  // Advance vy pixels per frame.

    var sokudo = dist(0,0,vx[i],vy[i]); // Calculate velocity
from X and Y components of velocity
    var r = map(sokudo, 0, 5, 0, 255); //Calculate color
according to speed
    var g = map(sokudo, 0, 5, 64, 255);
    var b = map(sokudo, 0, 5, 128, 255);
    fill(r, g, b, 32);
    ellipse(x[i], y[i], radius, radius);
  }

  ball.run();

}

class Ball {
  constructor(x, y, w, dx, dy) {
    this.loc = createVector(x, y);
    this.w = w
    this.vel = createVector(dx, dy);
  }

  run() {
    this.update();
    this.checkEdges();
    this.render();
  }

  update() {
    this.loc.x = this.loc.x + this.vel.x;
    this.loc.y = this.loc.y + this.vel.y
  }

  checkEdges() {
    if(this.loc.x > windowWidth - this.w/2 ||
      this.loc.x < this.w/2) {
        this.vel.x = -1*this.vel.x
    }
    if(this.loc.y > windowHeight - this.w/2 ||
      this.loc.y < this.w/2) {
        this.vel.y = -1*this.vel.y
    }
```

```
  }

  render() {
    fill(255, 223, 41, 50);
    ellipse(this.loc.x, this.loc.y, this.w);
  }

}
```

I changed this code by adding a ball class, of a ball that bounces around the screen. By pressing the spacebar, you can choose whether the particles follow your mouse or the ball.