The background features a complex, swirling pattern of white lines on a black surface, resembling a magnetic field or a vortex. A thick white diagonal line cuts across the frame from the bottom-left corner.

CS 184 Final Project

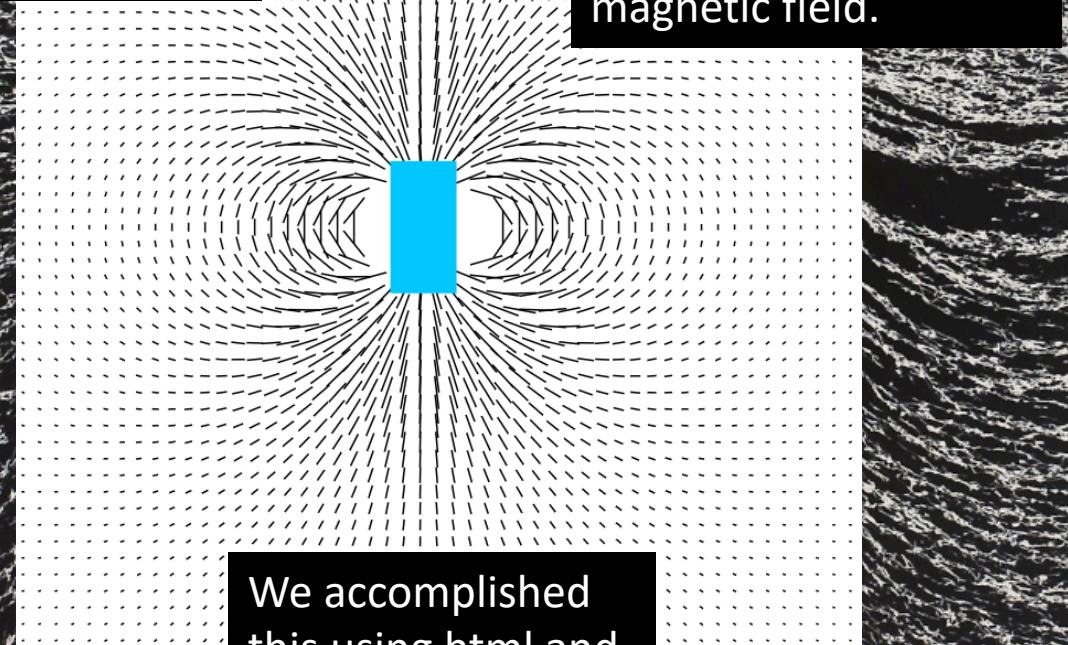
A Two-Dimensional Magnetic Playground

Tristan Cox, Isabel Daniels, Amber-Mae Skutelsky

Primary goal:

To display a two-dimensional simulation of metal shards manipulated by magnetism

We started by implementing a basic visualization of the magnetic field of a bar magnet.



Using position in relation to the dipole moment of the magnet, we were able to simulate the magnetic field.

```
for (var i = 0; i < width; i += stepLength) {  
    for (var j = 0; j < height; j += stepLength) {  
        var br = new Two.Vector((i - center_x) / the_calm_downer, (j - center_y) / the_calm_downer);  
        var r = br.length();  
        if (r > controls.strength/constraint) {  
            continue;  
        }  
        var rhat = new Two.Vector(br.x, br.y);  
        rhat.set(rhat.x / r, rhat.y / r);  
        var m = new Two.Vector(0, 5 * 20 / 6 * controls.strength);  
        var numTermOne = 3 * rhat.dot(m);  
        var numerator = (rhat.multiplyScalar(numTermOne)).subSelf(m);  
        numerator.divideScalar(2 * Math.PI * r * r);  
  
        if (numerator.length() > stepLength * 2) {  
            numerator = numerator.normalize().multiplyScalar(stepLength * 2);  
        }  
        var x2 = i + 0.5 * (numerator.x);  
        var x1 = i - 0.5 * (numerator.x);  
        var y2 = j + 0.5 * (numerator.y);  
        var y1 = j - 0.5 * (numerator.y);  
  
        if (r > 2) {  
            var line = two.makeLine(x1, y1, x2, y2);  
            line.linewidth = 0.5;  
            line.fill = "#881111";  
            line.stroke = "rgba(0, 0, 0, 255)";  
        }  
    }  
}
```

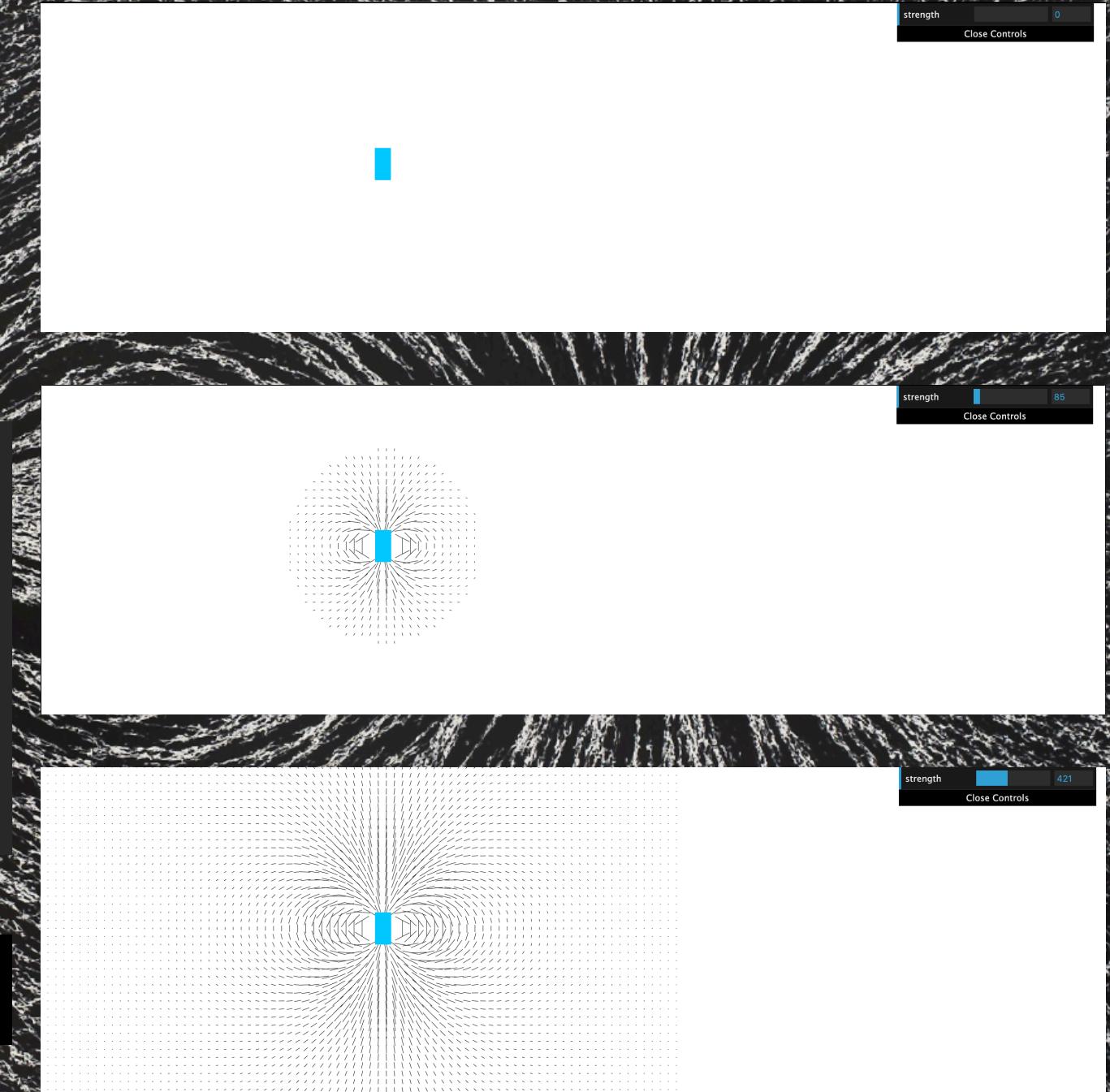
We added a feature where the user can adjust the strength of the magnet:

```
var controls = new function() {
  this.strength = 50;
};

var gui = new dat.GUI();
var strength = gui.add(controls, 'strength', 0, 1000);

render();
strength.onChange(
  render
);
```

We implemented this using dat.gui.js



We introduced a second magnet:

Using a summation of the magnetic field vectors coming from both magnets, we were able to simulate two fields and their effects on each other.

```
File Edit View Search Terminal Help
51     var stepLength = 2;
50     var magnets = new Array(2);
49     for (var i = 0; i < magnets.length; i++) {
48       magnets[i] = new Array(3);
47     }
46     magnets[0] = [center_x, center_y, 50];
45     magnets[1] = [250, 150, 50];
44     var vectors = {};
43     console.log(magnets);
42     var d = new Date();
41     var t = d.getTime();
40     console.log(t);
39     for (var i = 0; i < width; i+= stepLength) {
38       for (var j = 0; j < height; j+=stepLength) {
37         var str = getkey(i, j);
36         var currVect = vectors[str];
35         for (var k = 0; k < magnets.length; k++) {
34           var br = new Two.Vector((i - magnets[k][0])/the_calm_downer, (j - magnets[k][1])/the_calm_downer);
33           var r = br.length();
32           //if (r > magnets[k][2] / 7) {
31             // continue;
30           //}
29           var rhat = new Two.Vector(br.x,br.y);
28           rhat.set(rhat.x / r, rhat.y / r);
27           var m = new Two.Vector(0, 5*20 / 6 * magnets[k][2]);
26           var numTermOne = 3 * rhat.dot(m);
25           var numerator = (rhat.multiplyScalar(numTermOne)).subSelf(m);
24           numerator.divideScalar(2 * Math.PI * r * r);
23
22
21           //if(r > 1.5) {
20             if (currVect == null) {
19               vectors[str] = new Two.Vector();
18             }
17             vectors[str].addSelf(numerator);
16             currVect = vectors[str];
15           //}
14     }
13   if (currVect == null)
12     continue;
11   if (currVect.length() > stepLength * 2)
10     currVect = currVect.normalize().multiplyScalar(stepLength * 2);
9     var x2 = i + 0.5 * (currVect.x);
8     var x1 = i - 0.5 * (currVect.x);
7     var y2 = j + 0.5 * (currVect.y);
6     var y1 = j - 0.5 * (currVect.y);
5
4     var line = two.makeLine(x1, y1, x2, y2);
3     line.lineWidth = 0.5;
2     line.fill = "#881111";
1     line.stroke = "rgba(0, 0, 0, 255)";
81
```

