Canvas Second Step

小林　統 *

2022 年 4 月 27 日

概要

HTML5 の Canvas の機能を利用して、物理的な表現に取り組んでみましょう。

目次

* 帝京平成大学人文社会学部人間文化学科メディア文化コース

1　はじめに

1.1　読み間違えないでね

ソースコード 1　読み間違えないでね

```
1  数字: 0123456789
2  小文字:abcdefghijklmnopqrstuvwxyz
3  大文字:ABCDEFGHIJKLMNOPQRSTUVWXYZ
4
5  1:イチ
6  l：小文字のエル
7  i：小文字のアイ
8  !:ビックリマーク
9  |:バーティカルバー。Shift と￥を押したもの。
10
11 0:ゼロ
12 o：小文字のオー
13 O：大文字のオー
14
15 .:ピリオド
16 ,:コンマ
```

1.2　注意

- これから出てくるソースコードには、左に「行番号」と呼ばれる番号が出てくるけど、入力する必要ないからね。
- script タグの中で「//」で始まる文は、コメントで、プログラムは読み飛ばすよ。
- コピーできるところはコピーして効率よく入力して行こう
- 徐々に追加されていくから、量が多く見えるけど、平気だよ！
- 改行されていても、行番号が書かれていないところは、1 行だからね。表示上改行されて見えてるだけ

1.3　コンピュータの 2D の座標について

数学では、右に x, 上に y だったけど、
コンピュータでは、左上が原点、右に x, 下に y と考えるので、気をつけよう。

2　Canvas 応用

入門を受けて、いろんなアニメーション作って行こう

2.1　物理的な動き

2.1.1　04-101.html　円を描こう

CanvasFirstStep のおさらい 1。これ間違えると動かないよ。先週のコピってきても許す。

ソースコード 2　円を描こう

```
<!DOCTYPE html>
<html>
        <head>
                <title>Canvas Test</title>
                <script type="text/javascript">
                        //canvas の中身を指し示すもの
                        var ctx;
                        //canvas の width 幅
                        var cw;
                        //canvas の height 高さ
                        var ch;

                        //初期化処理
                        function init(){
                                //ID が mycanvas のものを canvas という変数にいれておく
                                var canvas = document.getElementById('mycanvas');
                                //canvas が存在しないか、canvas の中身がなければ処理終了
                                if(!canvas ||!canvas.getContext){
                                        return false;
                                }

                                //canvas の中身を指し示すものを代入
                                ctx = canvas.getContext('2d');
                                //canvas の幅・高さを代入しておく
                                cw = canvas.width;
                                ch = canvas.height;

                                //実際の描画処理
                                draw();
                        }

                        function draw(){
                                //円の描画
                                ctx.beginPath();
                                ctx.arc(100, 100, 20, 0, 2*Math.PI, false);
                                ctx.fillStyle = '#ff0000';
                                ctx.fill();
```

```
38                      }
39              </script>
40      </head>
41      <body onload="init();">
42              <H1>Canvas Test</H1>
43              <canvas id="mycanvas" width="1000" height="800"></canvas>
44      </body>
45 </html>
```

2.1.2 04-102.html 一定時間ごとに円を描くけど... 動かない

CanvasFirstStep のおさらい 2。円を描くのに 4 行使うから、それを drawCircle という関数にまとめてるよ。

42 行目、45-51 行目あたり。

一定時間で描写してるけど、同じところに同じもの書いてるだけだから動作は変わらないよ。

32 行目あたりが増えてるね。

ソースコード 3 一定時間ごとに円を描くけど... 動かない

```
1  <!DOCTYPE html>
2  <html>
3          <head>
4                  <title>Canvas Test</title>
5                  <script type="text/javascript">
6                          //canvas の中身を指し示すもの
7                          var ctx;
8                          //canvas の width 幅
9                          var cw;
10                         //canvas の height 高さ
11                         var ch;
12
13                         //初期化処理
14                         function init(){
15                                 //ID が mycanvas のものを canvas という変数にいれておく
16                                 var canvas = document.getElementById('mycanvas');
17                                 //canvas が存在しないか、canvas の中身がなければ処理終了
18                                 if(!canvas ||!canvas.getContext){
19                                         return false;
20                                 }
21
22                                 //canvas の中身を指し示すものを代入
23                                 ctx = canvas.getContext('2d');
24                                 //canvas の幅・高さを代入しておく
25                                 cw = canvas.width;
26                                 ch = canvas.height;
27
28                                 //実際の描画処理
29                                 draw();
30
```

```
31                              //一定時間ごとに書き換える 30fps にするために 1000msec/30
                                    = 33msec
32                              setInterval("draw()",33);
33
34                      }
35
36                      function draw(){
37                              //画面をリセットする
38                              ctx.fillStyle = "rgba(255,255,255,1)";
39                              ctx.fillRect(0,0,cw,ch);
40
41                              //円の描画
42                              drawCircle(100, 100, 20, '#FF0000');
43                      }
44
45                      function drawCircle(x,y,scale,color){
46                              //円の描画
47                              ctx.beginPath();
48                              ctx.arc(x, y, scale, 0, 2*Math.PI, false);
49                              ctx.fillStyle = color;
50                              ctx.fill();
51                      }
52              </script>
53      </head>
54      <body onload="init();">
55              <H1>Canvas Test</H1>
56              <canvas id="mycanvas" width="1000" height="800"></canvas>
57      </body>
58 </html>
```

2.1.3 04-103.html 等速度運動

一定のスピードで動くのを等速度運動って言ったよね????

14 行目あたり 46,7 行目あたりかな。

ソースコード 4 等速度運動

```
1  <!DOCTYPE html>
2  <html>
3          <head>
4                  <title>Canvas Test</title>
5                  <script type="text/javascript">
6                          //canvas の中身を指し示すもの
7                          var ctx;
8                          //canvas の width 幅
9                          var cw;
10                         //canvas の height 高さ
11                         var ch;
12
```

```
13                      //移動速度
14                      var speed = 2;
15                      var x = 0;
16
17                      //初期化処理
18                      function init(){
19                              //ID が mycanvas のものを canvas という変数にいれておく
20                              var canvas = document.getElementById('mycanvas');
21                              //canvas が存在しないか、canvas の中身がなければ処理終了
22                              if(!canvas ||!canvas.getContext){
23                                      return false;
24                              }
25
26                              //canvas の中身を指し示すものを代入
27                              ctx = canvas.getContext('2d');
28                              //canvas の幅・高さを代入しておく
29                              cw = canvas.width;
30                              ch = canvas.height;
31
32                              //実際の描画処理
33                              draw();
34
35                              //一定時間ごとに書き換える
36                              setInterval("draw()",33);
37
38                      }
39
40                      function draw(){
41                              //画面をリセットする
42                              ctx.fillStyle = "rgba(255,255,255,1)";
43                              ctx.fillRect(0,0,cw,ch);
44
45                              //円の描画
46                              x += speed;
47                              drawCircle(x, 100, 20, '#FF0000');
48                      }
49
50                      function drawCircle(x,y,scale,color){
51                              //円の描画
52                              ctx.beginPath();
53                              ctx.arc(x, y, scale, 0, 2*Math.PI, false);
54                              ctx.fillStyle = color;
55                              ctx.fill();
56                      }
57              </script>
58      </head>
59      <body onload="init();">
60              <H1>Canvas Test</H1>
```

```
61                 <canvas id="mycanvas" width="1000" height="800"></canvas>
62         </body>
63 </html>
```

2.1.4 04-104.html 摩擦のシミュレーション

目的地を設定して、そこに向かって動いて止まる動きをするよ。

16,47 行目あたり

ソースコード 5 摩擦のシミュレーション

```
1  <!DOCTYPE html>
2  <html>
3          <head>
4                  <title>Canvas Test</title>
5                  <script type="text/javascript">
6                          //canvas の中身を指し示すもの
7                          var ctx;
8                          //canvas の width 幅
9                          var cw;
10                         //canvas の height 高さ
11                         var ch;
12
13                         //移動速度
14                         var speed = 20;
15                         var x = 0;
16                         var target_x = 400;
17
18                         //初期化処理
19                         function init(){
20                                 //ID が mycanvas のものを canvas という変数にいれておく
21                                 var canvas = document.getElementById('mycanvas');
22                                 //canvas が存在しないか、canvas の中身がなければ処理終了
23                                 if(!canvas ||!canvas.getContext){
24                                         return false;
25                                 }
26
27                                 //canvas の中身を指し示すものを代入
28                                 ctx = canvas.getContext('2d');
29                                 //canvas の幅・高さを代入しておく
30                                 cw = canvas.width;
31                                 ch = canvas.height;
32
33                                 //実際の描画処理
34                                 draw();
35
36                                 //一定時間ごとに書き換える
37                                 setInterval("draw()",33);
38
```

```
39                          }
40
41                          function draw(){
42                                  //画面をリセットする
43                                  ctx.fillStyle = "rgba(255,255,255,1)";
44                                  ctx.fillRect(0,0,cw,ch);
45
46                                  //円の描画
47                                  x += (target_x - x) / speed;
48                                  drawCircle(x, 100, 20, '#FF0000');
49                          }
50
51                          function drawCircle(x,y,scale,color){
52                                  //円の描画
53                                  ctx.beginPath();
54                                  ctx.arc(x, y, scale, 0, 2*Math.PI, false);
55                                  ctx.fillStyle = color;
56                                  ctx.fill();
57                          }
58                  </script>
59          </head>
60          <body onload="init();">
61                  <H1>Canvas Test</H1>
62                  <canvas id="mycanvas" width="1000" height="800"></canvas>
63          </body>
64  </html>
```

2.2 Particle というオブジェクトを用いた生成方法

これまでは、Canvas に対して円を書いて！と指示してきたよね。

オブジェクト指向という考え方があって、例えば、車を例にしてみよう。

車に「走れ」「止まれ」「右に曲がって」「左に曲がって」と指示をアクセル・ブレーキ・ハンドルを使って指示をすると、勝手にそう動くよね。別に、エンジンの仕組みを知らなくても人は操作できるわけだ。

この様に、機能を持ったモノを定義して、そこに指示を出していくプログラミング方法を「オブジェクト指向」っていうよ。

また、車というオブジェクトがあったときに、A さんの車、B さんの車、という風に色々な車があるよね。オブジェクトは抽象化された概念なのに対して、A さんの車とかは実体化されたもの (インスタンス) と捉えるよ。

車という概念を定義しておいて、必要に応じてインスタンスを適宜生成することによって、いろいろなことができる様になるよ。

今回は、Particle(粒子) 等オブジェクトを作成して、それのインスタンスをたくさん作ることで面白い表現をしてみよう。

41-49 あたりは追加。それを受けて、67,68 あたりが変わるよ。

42 行目からは、Particle の初期化の話

51 行目からは、Particle に draw()... 描け、と言った時の挙動が定義されています。

59 行目で、Particleっていうオブジェクトから実態としての particle を生成しています。

2.2.1 04-201.html　オブジェクトを利用した等加速度運動

ソースコード 6　　オブジェクトを利用した等加速度運動

```
1  <!DOCTYPE html>
2  <html>
3          <head>
4                  <title>Canvas Test</title>
5                  <script type="text/javascript">
6                          //canvas の中身を指し示すもの
7                          var ctx;
8                          //canvas の width 幅
9                          var cw;
10                         //canvas の height 高さ
11                         var ch;
12
13                         //移動速度
14                         var speed = 20;
15                         var x = 0;
16                         var target_x = 400;
17
18                         //初期化処理
19                         function init(){
20                                 //ID が mycanvas のものを canvas という変数にいれておく
21                                 var canvas = document.getElementById('mycanvas');
22                                 //canvas が存在しないか、canvas の中身がなければ処理終了
23                                 if(!canvas ||!canvas.getContext){
24                                         return false;
25                                 }
26
27                                 //canvas の中身を指し示すものを代入
28                                 ctx = canvas.getContext('2d');
29                                 //canvas の幅・高さを代入しておく
30                                 cw = canvas.width;
31                                 ch = canvas.height;
32
33                                 //実際の描画処理
34                                 draw();
35
36                                 //一定時間ごとに書き換える
37                                 setInterval("draw()",33);
38
39                         }
40
41                         //オブジェクト初期化処理
42                         var Particle = function(scale, color, speed){
43                                 this.scale = scale;
```

```
44                              this.color = color;
45                              this.speed = speed;
46                              this.position = {
47                                      x: 100,
48                                      y: 100
49                              };
50                      };
51                      Particle.prototype.draw = function() {
52                              ctx.beginPath();
53                              ctx.arc(this.position.x, this.position.y, this.scale,
                                    0, 2*Math.PI, false);
54                              ctx.fillStyle = this.color;
55                              ctx.fill();
56                      };
57
58                      //パーティクルの準備
59                      var particle = new Particle(20, "#ff0000", 2);
60
61                      function draw(){
62                              //画面をリセットする
63                              ctx.fillStyle = "rgba(255,255,255,1)";
64                              ctx.fillRect(0,0,cw,ch);
65
66                              //円の描画
67                              particle.position.x += particle.speed;
68                              particle.draw();
69                      }
70
71              </script>
72      </head>
73      <body onload="init();">
74              <H1>Canvas Test</H1>
75              <canvas id="mycanvas" width="1000" height="800"></canvas>
76      </body>
77 </html>
```

2.2.2 04-202.html 配列に Particle を入れてアニメーション

一つだけじゃ、メリットわかりませんね。複数 particle を生成してみましょう。

19 行目、38 行目,78 行目あたり

ソースコード 7 配列に Particle を入れてアニメーション

```
1  <!DOCTYPE html>
2  <html>
3          <head>
4                  <title>Canvas Test</title>
5                  <script type="text/javascript">
6                          //canvas の中身を指し示すもの
```

```
7   var ctx;
8   //canvas の width 幅
9   var cw;
10  //canvas の height 高さ
11  var ch;
12
13  //移動速度
14  var speed = 20;
15  var x = 0;
16  var target_x = 400;
17
18  //パーティクルの準備
19  var density = 100; //パーティクルの密度
20  var particles = [];
21
22  //初期化処理
23  function init(){
24          //ID が mycanvas のものを canvas という変数にいれておく
25          var canvas = document.getElementById('mycanvas');
26          //canvas が存在しないか、canvas の中身がなければ処理終了
27          if(!canvas ||!canvas.getContext){
28                  return false;
29          }
30
31          //canvas の中身を指し示すものを代入
32          ctx = canvas.getContext('2d');
33          //canvas の幅・高さを代入しておく
34          cw = canvas.width;
35          ch = canvas.height;
36
37          //円の初期化
38          for(var i=0;i < density; i++){
39                  particles[i] = new Particle(6, "#FF0000",
                        Math.random()*(4-2)+2);
40                  particles[i].position.x = Math.random()*ch;
41                  particles[i].position.y = Math.random()*cw;
42                  particles[i].draw();
43          }
44
45          //実際の描画処理
46          draw();
47
48          //一定時間ごとに書き換える
49          setInterval("draw()",33);
50
51  }
52
53  //オブジェクト初期化処理
```

```
54                          var Particle = function(scale, color, speed){
55                                  this.scale = scale;
56                                  this.color = color;
57                                  this.speed = speed;
58                                  this.position = {
59                                          x: 100,
60                                          y: 100
61                                  };
62                          };
63
64                          Particle.prototype.draw = function() {
65                                  ctx.beginPath();
66                                  ctx.arc(this.position.x, this.position.y, this.scale,
                                        0, 2*Math.PI, false);
67                                  ctx.fillStyle = this.color;
68                                  ctx.fill();
69                          };
70
71
72                          function draw(){
73                                  //画面をリセットする
74                                  ctx.fillStyle = "rgba(255,255,255,1)";
75                                  ctx.fillRect(0,0,cw,ch);
76
77                                  //円の描画
78                                  for(var i=0;i < density; i++){
79                                          particles[i].position.x += particles[i].speed;
80                                          particles[i].draw();
81
82                                          if(particles[i].position.x > cw) particles[i].
                                                position.x -= cw;
83                                  }
84                          }
85
86
87                  </script>
88          </head>
89          <body onload="init();">
90                  <H1>Canvas Test</H1>
91                  <canvas id="mycanvas" width="1000" height="800"></canvas>
92          </body>
93  </html>
```

2.2.3 04-203.html **色を変更**

円を生成する時に色を適当に指定しましょう。

86 行目あたり getRandomColor(),getRandomAlpha() という関数を追加

39 行目あたり

ソースコード 8　色を変更

```
<!DOCTYPE html>
<html>
        <head>
                <title>Canvas Test</title>
                <script type="text/javascript">
                        //canvas の中身を指し示すもの
                        var ctx;
                        //canvas の width 幅
                        var cw;
                        //canvas の height 高さ
                        var ch;

                        //移動速度
                        var speed = 20;
                        var x = 0;
                        var target_x = 400;

                        //パーティクルの準備
                        var density = 100; //パーティクルの密度
                        var particles = [];

                        //初期化処理
                        function init(){
                                //ID が mycanvas のものを canvas という変数にいれておく
                                var canvas = document.getElementById('mycanvas');
                                //canvas が存在しないか、canvas の中身がなければ処理終了
                                if(!canvas ||!canvas.getContext){
                                        return false;
                                }

                                //canvas の中身を指し示すものを代入
                                ctx = canvas.getContext('2d');
                                //canvas の幅・高さを代入しておく
                                cw = canvas.width;
                                ch = canvas.height;

                                //円の初期化
                                for(var i=0;i < density; i++){
                                        particles[i] = new Particle(6, "rgba("+
                                            getRandomColor()+","+getRandomColor
                                            ()+","+getRandomColor()+","+
                                            getRandomAlpha()+")", Math.random
                                            ()*(4-2)+2);
                                        particles[i].position.x = Math.random()*ch;
                                        particles[i].position.y = Math.random()*cw;
                                        particles[i].draw();
                                }
```

```
44
45                      //実際の描画処理
46                      draw();
47
48                      //一定時間ごとに書き換える
49                      setInterval("draw()",33);
50
51              }
52
53              //オブジェクト初期化処理
54              var Particle = function(scale, color, speed){
55                      this.scale = scale;
56                      this.color = color;
57                      this.speed = speed;
58                      this.position = {
59                              x: 100,
60                              y: 100
61                      };
62              };
63
64              Particle.prototype.draw = function() {
65                      ctx.beginPath();
66                      ctx.arc(this.position.x, this.position.y, this.scale,
                            0, 2*Math.PI, false);
67                      ctx.fillStyle = this.color;
68                      ctx.fill();
69              };
70
71
72              function draw(){
73                      //画面をリセットする
74                      ctx.fillStyle = "rgba(255,255,255,1)";
75                      ctx.fillRect(0,0,cw,ch);
76
77                      //円の描画
78                      for(var i=0;i < density; i++){
79                              particles[i].position.x += particles[i].speed;
80                              particles[i].draw();
81
82                              if(particles[i].position.x > cw) particles[i].
                                    position.x -= cw;
83                      }
84              }
85
86              function getRandomColor(){
87                      return Math.floor(Math.random()*255);
88              }
89              function getRandomAlpha(){
```

```
90                              return Math.random();
91                      }
92
93              </script>
94      </head>
95      <body onload="init();">
96              <H1>Canvas Test</H1>
97              <canvas id="mycanvas" width="1000" height="800"></canvas>
98      </body>
99 </html>
```

2.2.4 04-204.html 大きさを変更

91 行目あたり、getRandomScale()っていう関数を追加

39 行目あたり

ソースコード 9 大きさを変更

```
1  <!DOCTYPE html>
2  <html>
3          <head>
4                  <title>Canvas Test</title>
5                  <script type="text/javascript">
6                          //canvas の中身を指し示すもの
7                          var ctx;
8                          //canvas の width 幅
9                          var cw;
10                         //canvas の height 高さ
11                         var ch;
12
13                         //移動速度
14                         var speed = 20;
15                         var x = 0;
16                         var target_x = 400;
17
18                         //パーティクルの準備
19                         var density = 100; //パーティクルの密度
20                         var particles = [];
21
22                         //初期化処理
23                         function init(){
24                                 //ID が mycanvas のものを canvas という変数にいれておく
25                                 var canvas = document.getElementById('mycanvas');
26                                 //canvas が存在しないか、canvas の中身がなければ処理終了
27                                 if(!canvas ||!canvas.getContext){
28                                         return false;
29                                 }
30
31                                 //canvas の中身を指し示すものを代入
```

```
32                      ctx = canvas.getContext('2d');
33                      //canvas の幅・高さを代入しておく
34                      cw = canvas.width;
35                      ch = canvas.height;
36
37                      //円の初期化
38                      for(var i=0;i < density; i++){
39                              particles[i] = new Particle(getRandomScale(),
                                    "rgba("+getRandomColor()+","+
                                    getRandomColor()+","+getRandomColor
                                    ()+","+getRandomAlpha()+")", Math.random
                                    ()*(4-2)+2);
40                              particles[i].position.x = Math.random()*ch;
41                              particles[i].position.y = Math.random()*cw;
42                              particles[i].draw();
43                      }
44
45                      //実際の描画処理
46                      draw();
47
48                      //一定時間ごとに書き換える
49                      setInterval("draw()",33);
50              }
51
52              //オブジェクト初期化処理
53              var Particle = function(scale, color, speed){
54                      this.scale = scale;
55                      this.color = color;
56                      this.speed = speed;
57                      this.position = {
58                              x: 100,
59                              y: 100
60                      };
61              };
62
63              Particle.prototype.draw = function() {
64                      ctx.beginPath();
65                      ctx.arc(this.position.x, this.position.y, this.scale,
                            0, 2*Math.PI, false);
66                      ctx.fillStyle = this.color;
67                      ctx.fill();
68              };
69
70
71              function draw(){
72                      //画面をリセットする
73                      ctx.fillStyle = "rgba(255,255,255,1)";
74                      ctx.fillRect(0,0,cw,ch);
```

```
75
76                                  //円の描画
77                                  for(var i=0;i < density; i++){
78                                          particles[i].position.x += particles[i].speed;
79                                          particles[i].draw();
80
81                                          if(particles[i].position.x > cw) particles[i].
                                                position.x -= cw;
82                                  }
83                          }
84
85                          function getRandomColor(){
86                                  return Math.floor(Math.random()*255);
87                          }
88                          function getRandomAlpha(){
89                                  return Math.random();
90                          }
91                          function getRandomScale(){
92                                  return (Math.random()*(8-3))+3;
93                          }
94
95                  </script>
96          </head>
97          <body onload="init();">
98                  <H1>Canvas Test</H1>
99                  <canvas id="mycanvas" width="1000" height="800"></canvas>
100         </body>
101 </html>
```

2.3 より高度な表現

2.3.1 04-301.html 一定の方向に等加速度運動

54,57,58 行目で speed の代わりに vx,vy が増えているね。2 次元の速度を持たせるよ。
これを受けて、39 行目も変更
updateっていう処理を 71 行目で追加して、83 行目たりでそれを呼び出している。

ソースコード 10 一定の方向に等加速度運動

```
1  <!DOCTYPE html>
2  <html>
3          <head>
4                  <title>Canvas Test</title>
5                  <script type="text/javascript">
6                          //canvas の中身を指し示すもの
7                          var ctx;
8                          //canvas の width 幅
9                          var cw;
10                         //canvas の height 高さ
```

```
11                      var ch;
12
13                      //移動速度
14                      var speed = 20;
15                      var x = 0;
16                      var target_x = 400;
17
18                      //パーティクルの準備
19                      var density = 100; //パーティクルの密度
20                      var particles = [];
21
22                      //初期化処理
23                      function init(){
24                              //ID が mycanvas のものを canvas という変数にいれておく
25                              var canvas = document.getElementById('mycanvas');
26                              //canvas が存在しないか、canvas の中身がなければ処理終了
27                              if(!canvas ||!canvas.getContext){
28                                      return false;
29                              }
30
31                              //canvas の中身を指し示すものを代入
32                              ctx = canvas.getContext('2d');
33                              //canvas の幅・高さを代入しておく
34                              cw = canvas.width;
35                              ch = canvas.height;
36
37                              //円の初期化
38                              for(var i=0;i < density; i++){
39                                      particles[i] = new Particle(getRandomScale(),
                                            "rgba("+getRandomColor()+","+
                                            getRandomColor()+","+getRandomColor
                                            ()+","+getRandomAlpha()+")", 5, 1);
40                                      particles[i].position.x = Math.random()*ch;
41                                      particles[i].position.y = Math.random()*cw;
42                                      particles[i].draw();
43                              }
44
45                              //実際の描画処理
46                              draw();
47
48                              //一定時間ごとに書き換える
49                              setInterval("draw()",33);
50
51                      }
52
53                      //オブジェクト初期化処理
54                      var Particle = function(scale, color, vx, vy){
55                              this.scale = scale;
```

```
56                              this.color = color;
57                              this.vx = vx;
58                              this.vy = vy;
59                              this.position = {
60                                      x: 100,
61                                      y: 100
62                              };
63                      };
64
65                      Particle.prototype.draw = function() {
66                              ctx.beginPath();
67                              ctx.arc(this.position.x, this.position.y, this.scale,
                                    0, 2*Math.PI, false);
68                              ctx.fillStyle = this.color;
69                              ctx.fill();
70                      };
71                      Particle.prototype.update = function(){
72                              this.position.x += this.vx;
73                              this.position.y += this.vy;
74                              this.draw();
75                      };
76
77
78                      function draw(){
79                              //画面をリセットする
80                              ctx.fillStyle = "rgba(255,255,255,1)";
81                              ctx.fillRect(0,0,cw,ch);
82
83                              //円の描画
84                              for(var i=0;i < density; i++){
85                                      particles[i].update();
86                              }
87                      }
88
89                      function getRandomColor(){
90                              return Math.floor(Math.random()*255);
91                      }
92                      function getRandomAlpha(){
93                              return Math.random();
94                      }
95                      function getRandomScale(){
96                              return (Math.random()*(8-3))+3;
97                      }
98
99              </script>
100     </head>
101     <body onload="init();">
102             <H1>Canvas Test</H1>
```

```
103                 <canvas id="mycanvas" width="1000" height="800"></canvas>
104         </body>
105 </html>
```

2.3.2 04-302.html 重力のシミュレーション

54 行目に gv:gravity が増えているね…73 行目あたりで速度に重力加速度が増えている。
これを受けて、54,59 行目あたりも増えてるよ。

ソースコード 11 重力のシミュレーション

```
1  <!DOCTYPE html>
2  <html>
3          <head>
4                  <title>Canvas Test</title>
5                  <script type="text/javascript">
6                          //canvas の中身を指し示すもの
7                          var ctx;
8                          //canvas の width 幅
9                          var cw;
10                         //canvas の height 高さ
11                         var ch;
12
13                         //移動速度
14                         var speed = 20;
15                         var x = 0;
16                         var target_x = 400;
17
18                         //パーティクルの準備
19                         var density = 100; //パーティクルの密度
20                         var particles = [];
21
22                         //初期化処理
23                         function init(){
24                                 //ID が mycanvas のものを canvas という変数にいれておく
25                                 var canvas = document.getElementById('mycanvas');
26                                 //canvas が存在しないか、canvas の中身がなければ処理終了
27                                 if(!canvas ||!canvas.getContext){
28                                         return false;
29                                 }
30
31                                 //canvas の中身を指し示すものを代入
32                                 ctx = canvas.getContext('2d');
33                                 //canvas の幅・高さを代入しておく
34                                 cw = canvas.width;
35                                 ch = canvas.height;
36
37                                 //円の初期化
38                                 for(var i=0;i < density; i++){
```

```
39                          particles[i] = new Particle(getRandomScale(),
                                "rgba("+getRandomColor()+","+
                                getRandomColor()+","+getRandomColor
                                ()+","+getRandomAlpha()+")", 5, 1, 0.4);
40                          particles[i].position.x = Math.random()*ch;
41                          particles[i].position.y = Math.random()*cw;
42                          particles[i].draw();
43                  }
44
45                  //実際の描画処理
46                  draw();
47
48                  //一定時間ごとに書き換える
49                  setInterval("draw()",33);
50
51          }
52
53          //オブジェクト初期化処理
54          var Particle = function(scale, color, vx, vy, gv){
55                  this.scale = scale;
56                  this.color = color;
57                  this.vx = vx;
58                  this.vy = vy;
59                  this.gv = gv;
60                  this.position = {
61                          x: 100,
62                          y: 100
63                  };
64          };
65
66          Particle.prototype.draw = function() {
67                  ctx.beginPath();
68                  ctx.arc(this.position.x, this.position.y, this.scale,
                        0, 2*Math.PI, false);
69                  ctx.fillStyle = this.color;
70                  ctx.fill();
71          };
72          Particle.prototype.update = function(){
73                  this.vy += this.gv;
74                  this.position.x += this.vx;
75                  this.position.y += this.vy;
76                  this.draw();
77          };
78
79
80          function draw(){
81                  //画面をリセットする
82                  ctx.fillStyle = "rgba(255,255,255,1)";
```

```
83                              ctx.fillRect(0,0,cw,ch);
84
85                              //円の描画
86                              for(var i=0;i < density; i++){
87                                      particles[i].update();
88                              }
89                      }
90
91                      function getRandomColor(){
92                              return Math.floor(Math.random()*255);
93                      }
94                      function getRandomAlpha(){
95                              return Math.random();
96                      }
97                      function getRandomScale(){
98                              return (Math.random()*(8-3))+3;
99                      }
100
101             </script>
102     </head>
103     <body onload="init();">
104             <H1>Canvas Test</H1>
105             <canvas id="mycanvas" width="1000" height="800"></canvas>
106     </body>
107 </html>
```

2.3.3 04-303.html 床の跳ね返りをシミュレーション

77 行目あたりで、跳ね返る様に設定している。

ソースコード 12 床の跳ね返りをシミュレーション

```
1  <!DOCTYPE html>
2  <html>
3          <head>
4                  <title>Canvas Test</title>
5                  <script type="text/javascript">
6                          //canvas の中身を指し示すもの
7                          var ctx;
8                          //canvas の width 幅
9                          var cw;
10                         //canvas の height 高さ
11                         var ch;
12
13                         //移動速度
14                         var speed = 20;
15                         var x = 0;
16                         var target_x = 400;
17
```

```
18                      //パーティクルの準備
19                      var density = 100; //パーティクルの密度
20                      var particles = [];
21
22                      //初期化処理
23                      function init(){
24                              //ID が mycanvas のものを canvas という変数にいれておく
25                              var canvas = document.getElementById('mycanvas');
26                              //canvas が存在しないか、canvas の中身がなければ処理終了
27                              if(!canvas ||!canvas.getContext){
28                                      return false;
29                              }
30
31                              //canvas の中身を指し示すものを代入
32                              ctx = canvas.getContext('2d');
33                              //canvas の幅・高さを代入しておく
34                              cw = canvas.width;
35                              ch = canvas.height;
36
37                              //円の初期化
38                              for(var i=0;i < density; i++){
39                                      particles[i] = new Particle(getRandomScale(),
                                            "rgba("+getRandomColor()+","+
                                            getRandomColor()+","+getRandomColor
                                            ()+","+getRandomAlpha()+")", 5, 1, 1);
40                                      particles[i].position.x = Math.random()*ch;
41                                      particles[i].position.y = Math.random()*cw;
42                                      particles[i].draw();
43                              }
44
45                              //実際の描画処理
46                              draw();
47
48                              //一定時間ごとに書き換える
49                              setInterval("draw()",33);
50
51                      }
52
53                      //オブジェクト初期化処理
54                      var Particle = function(scale, color, vx, vy, gv){
55                              this.scale = scale;
56                              this.color = color;
57                              this.vx = vx;
58                              this.vy = vy;
59                              this.gv = gv;
60                              this.position = {
61                                      x: 100,
62                                      y: 100
```

```
63                              };
64                      };
65
66                      Particle.prototype.draw = function() {
67                              ctx.beginPath();
68                              ctx.arc(this.position.x, this.position.y, this.scale,
                                    0, 2*Math.PI, false);
69                              ctx.fillStyle = this.color;
70                              ctx.fill();
71                      };
72                      Particle.prototype.update = function(){
73                              this.vy += this.gv;
74                              this.position.x += this.vx;
75                              this.position.y += this.vy;
76                              this.draw();
77                              if(this.position.x > cw) this.position.x -= cw;
78                              if(this.position.y > ch) this.vy = -this.vy;
79                      };
80
81
82                      function draw(){
83                              //画面をリセットする
84                              ctx.fillStyle = "rgba(255,255,255,1)";
85                              ctx.fillRect(0,0,cw,ch);
86
87                              //円の描画
88                              for(var i=0;i < density; i++){
89                                      particles[i].update();
90                              }
91                      }
92
93                      function getRandomColor(){
94                              return Math.floor(Math.random()*255);
95                      }
96                      function getRandomAlpha(){
97                              return Math.random();
98                      }
99                      function getRandomScale(){
100                             return (Math.random()*(8-3))+3;
101                     }
102
103             </script>
104     </head>
105     <body onload="init();">
106             <H1>Canvas Test</H1>
107             <canvas id="mycanvas" width="1000" height="800"></canvas>
108     </body>
109 </html>
```

2.3.4 04-304.html とある点からの吹き出し

getRandomVelocity() を 102 行目あたりで定義。39 行目で利用。

40 行目あたりで左右の真ん中、上から 1/4 の位置からスタートする様に設定してる。

ソースコード 13 とある点からの吹き出し

```
1  <!DOCTYPE html>
2  <html>
3          <head>
4                  <title>Canvas Test</title>
5                  <script type="text/javascript">
6                          //canvas の中身を指し示すもの
7                          var ctx;
8                          //canvas の width 幅
9                          var cw;
10                         //canvas の height 高さ
11                         var ch;
12
13                         //移動速度
14                         var speed = 20;
15                         var x = 0;
16                         var target_x = 400;
17
18                         //パーティクルの準備
19                         var density = 100; //パーティクルの密度
20                         var particles = [];
21
22                         //初期化処理
23                         function init(){
24                                 //ID が mycanvas のものを canvas という変数にいれておく
25                                 var canvas = document.getElementById('mycanvas');
26                                 //canvas が存在しないか、canvas の中身がなければ処理終了
27                                 if(!canvas ||!canvas.getContext){
28                                         return false;
29                                 }
30
31                                 //canvas の中身を指し示すものを代入
32                                 ctx = canvas.getContext('2d');
33                                 //canvas の幅・高さを代入しておく
34                                 cw = canvas.width;
35                                 ch = canvas.height;
36
37                                 //円の初期化
38                                 for(var i=0;i < density; i++){
39                                         particles[i] = new Particle(getRandomScale(),
                                               "rgba("+getRandomColor()+","+
                                               getRandomColor()+","+getRandomColor
                                               ()+","+getRandomAlpha()+")",
```

```
                                getRandomVelocity(), getRandomVelocity(),
                                1);
40                      particles[i].position.x = cw/2;
41                      particles[i].position.y = ch/4;
42                      particles[i].draw();
43              }
44
45              //実際の描画処理
46              draw();
47
48              //一定時間ごとに書き換える
49              setInterval("draw()",33);
50
51      }
52
53      //オブジェクト初期化処理
54      var Particle = function(scale, color, vx, vy, gv){
55              this.scale = scale;
56              this.color = color;
57              this.vx = vx;
58              this.vy = vy;
59              this.gv = gv;
60              this.position = {
61                      x: 100,
62                      y: 100
63              };
64      };
65
66      Particle.prototype.draw = function() {
67              ctx.beginPath();
68              ctx.arc(this.position.x, this.position.y, this.scale,
                    0, 2*Math.PI, false);
69              ctx.fillStyle = this.color;
70              ctx.fill();
71      };
72      Particle.prototype.update = function(){
73              this.vy += this.gv;
74              this.position.x += this.vx;
75              this.position.y += this.vy;
76              this.draw();
77              if(this.position.x > cw) this.position.x -= cw;
78              if(this.position.y > ch) this.vy = -this.vy;
79      };
80
81
82      function draw(){
83              //画面をリセットする
84              ctx.fillStyle = "rgba(255,255,255,1)";
```

```
85                              ctx.fillRect(0,0,cw,ch);
86
87                              //円の描画
88                              for(var i=0;i < density; i++){
89                                      particles[i].update();
90                              }
91                      }
92
93                      function getRandomColor(){
94                              return Math.floor(Math.random()*255);
95                      }
96                      function getRandomAlpha(){
97                              return Math.random();
98                      }
99                      function getRandomScale(){
100                             return (Math.random()*(8-3))+3;
101                     }
102                     function getRandomVelocity(){
103                             return Math.random()*20-10;
104                     }
105
106             </script>
107     </head>
108     <body onload="init();">
109             <H1>Canvas Test</H1>
110             <canvas id="mycanvas" width="1000" height="800"></canvas>
111     </body>
112 </html>
```

2.3.5 04-305.html 線の追加

22 行目で線の密度や、配列を追加。

88 行目から Line というオブジェクトを追加

142 行目あたりで getRandomX(),getRandomY(),getRandomWidth() を追加

48 行目で線を初期化、128 行目で更新をしている。

ソースコード 14 線の追加

```
1  <!DOCTYPE html>
2  <html>
3          <head>
4                  <title>Canvas Test</title>
5                  <script type="text/javascript">
6                          //canvas の中身を指し示すもの
7                          var ctx;
8                          //canvas の width 幅
9                          var cw;
10                         //canvas の height 高さ
11                         var ch;
```

```
12
13                      //移動速度
14                      var speed = 20;
15                      var x = 0;
16                      var target_x = 400;
17
18                      //パーティクルの準備
19                      var density = 100; //パーティクルの密度
20                      var particles = [];
21
22                      var density_line = 10;
23                      var lines = [];
24
25                      //初期化処理
26                      function init(){
27                              //ID が mycanvas のものを canvas という変数にいれておく
28                              var canvas = document.getElementById('mycanvas');
29                              //canvas が存在しないか、canvas の中身がなければ処理終了
30                              if(!canvas ||!canvas.getContext){
31                                      return false;
32                              }
33
34                              //canvas の中身を指し示すものを代入
35                              ctx = canvas.getContext('2d');
36                              //canvas の幅・高さを代入しておく
37                              cw = canvas.width;
38                              ch = canvas.height;
39
40                              //円の初期化
41                              for(var i=0;i < density; i++){
42                                      particles[i] = new Particle(getRandomScale(),
                                            "rgba("+getRandomColor()+","+
                                            getRandomColor()+","+getRandomColor
                                            ()+","+getRandomAlpha()+")",
                                            getRandomVelocity(), getRandomVelocity(),
                                            1);
43                                      particles[i].position.x = cw/2;
44                                      particles[i].position.y = ch/4;
45                                      particles[i].draw();
46                              }
47                              //線の初期化
48                              for(var j=0;j < density_line; j++){
49                                      lines[j] = new Line(getRandomX(), getRandomY
                                            (), getRandomVelocity(), getRandomVelocity
                                            (),getRandomX(), getRandomY(),
                                            getRandomVelocity(), getRandomVelocity(),"
                                            rgba("+getRandomColor()+","+getRandomColor
                                            ()+","+getRandomColor()+","+
```

```
                                getRandomAlpha()+")",getRandomWidth(),
                                1);
50                      lines[j].draw();
51              }
52
53              //実際の描画処理
54              draw();
55
56              //一定時間ごとに書き換える
57              setInterval("draw()",33);
58
59      }
60
61      //オブジェクト初期化処理
62      var Particle = function(scale, color, vx, vy, gv){
63              this.scale = scale;
64              this.color = color;
65              this.vx = vx;
66              this.vy = vy;
67              this.gv = gv;
68              this.position = {
69                      x: 100,
70                      y: 100
71              };
72      };
73      Particle.prototype.draw = function() {
74              ctx.beginPath();
75              ctx.arc(this.position.x, this.position.y, this.scale,
                        0, 2*Math.PI, false);
76              ctx.fillStyle = this.color;
77              ctx.fill();
78      };
79      Particle.prototype.update = function(){
80              this.vy += this.gv;
81              this.position.x += this.vx;
82              this.position.y += this.vy;
83              this.draw();
84              if(this.position.x > cw) this.position.x -= cw;
85              if(this.position.y > ch) this.vy = -this.vy;
86      };
87
88      var Line = function(x1, y1, vx1, vy1, x2, y2, vx2, vy2, color
                , width, gv){
89              this.position1 = {x:x1, y:y1};
90              this.v1 = {x:vx1, y:vy1};
91              this.position2 = {x:x2, y:y2};
92              this.v2 = {x:vx2, y:vy2};
93              this.color = color;
```

```
94                      this.width = width;
95                      this.gv = gv;
96              };
97              Line.prototype.draw = function(){
98                      ctx.strokeStyle = this.color;
99                      ctx.lineWidth = this.width;
100                     ctx.beginPath();
101                     ctx.moveTo(this.position1.x, this.position1.y);
102                     ctx.lineTo(this.position2.x, this.position2.y);
103                     ctx.stroke();
104             };
105             Line.prototype.update = function(){
106                     this.v1.y += this.gv;
107                     this.v2.y += this.gv;
108                     this.position1.x += this.v1.x;
109                     this.position1.y += this.v1.y;
110                     this.position2.x += this.v2.x;
111                     this.position2.y += this.v2.y;
112                     this.draw();
113                     if(this.position1.y > ch) this.v1.y = -this.v1.y;
114                     if(this.position2.y > ch) this.v2.y = -this.v2.y;
115             };
116
117             function draw(){
118                     //画面をリセットする
119                     ctx.fillStyle = "rgba(255,255,255,1)";
120                     ctx.fillRect(0,0,cw,ch);
121
122                     //円の描画
123                     for(var i=0;i < density; i++){
124                             particles[i].update();
125                     }
126
127                     //線の描画
128                     for(var j=0;j < density_line; j++){
129                             lines[j].update();
130                     }
131             }
132
133             function getRandomColor(){
134                     return Math.floor(Math.random()*255);
135             }
136             function getRandomAlpha(){
137                     return Math.random();
138             }
139             function getRandomScale(){
140                     return (Math.random()*(8-3))+3;
141             }
```

```
142                         function getRandomX(){
143                                 return Math.random()*cw;
144                         }
145                         function getRandomY(){
146                                 return Math.random()*ch;
147                         }
148                         function getRandomWidth(){
149                                 return Math.random()*10;
150                         }
151                         function getRandomVelocity(){
152                                 return Math.random()*20-10;
153                         }
154
155                 </script>
156         </head>
157         <body onload="init();">
158                 <H1>Canvas Test</H1>
159                 <canvas id="mycanvas" width="1000" height="800"></canvas>
160         </body>
161 </html>
```

2.3.6 04-306.html 三角形の追加

25 行目で三角形の密度や、配列を追加。

124 行目から三角形というオブジェクトを追加

56 行目で三角形を初期化、176 行目で更新をしている。

ソースコード 15 三角形の追加

```
1  <!DOCTYPE html>
2  <html>
3          <head>
4                  <title>Canvas Test</title>
5                  <script type="text/javascript">
6                          //canvas の中身を指し示すもの
7                          var ctx;
8                          //canvas の width 幅
9                          var cw;
10                         //canvas の height 高さ
11                         var ch;
12
13                         //移動速度
14                         var speed = 20;
15                         var x = 0;
16                         var target_x = 400;
17
18                         //パーティクルの準備
19                         var density = 100; //パーティクルの密度
20                         var particles = [];
```

```
21
22                      var density_line = 10;
23                      var lines = [];
24
25                      var density_tri = 10;
26                      var tris = [];
27
28                      //初期化処理
29                      function init(){
30                              //ID が mycanvas のものを canvas という変数にいれておく
31                              var canvas = document.getElementById('mycanvas');
32                              //canvas が存在しないか、canvas の中身がなければ処理終了
33                              if(!canvas ||!canvas.getContext){
34                                      return false;
35                              }
36
37                              //canvas の中身を指し示すものを代入
38                              ctx = canvas.getContext('2d');
39                              //canvas の幅・高さを代入しておく
40                              cw = canvas.width;
41                              ch = canvas.height;
42
43                              //円の初期化
44                              for(var i=0;i < density; i++){
45                                      particles[i] = new Particle(getRandomScale(),
                                            "rgba("+getRandomColor()+","+
                                            getRandomColor()+","+getRandomColor
                                            ()+","+getRandomAlpha()+")",
                                            getRandomVelocity(), getRandomVelocity(),
                                            1);
46                                      particles[i].position.x = cw/2;
47                                      particles[i].position.y = ch/4;
48                                      particles[i].draw();
49                              }
50                              //線の初期化
51                              for(var j=0;j < density_line; j++){
52                                      lines[j] = new Line(getRandomX(), getRandomY
                                            (), getRandomVelocity(), getRandomVelocity
                                            (),getRandomX(), getRandomY(),
                                            getRandomVelocity(), getRandomVelocity()," 
                                            rgba("+getRandomColor()+","+getRandomColor
                                            ()+","+getRandomColor()+","+
                                            getRandomAlpha()+")",getRandomWidth(),
                                            1);
53                                      lines[j].draw();
54                              }
55                              //三角形の初期化
56                              for(var k=0;k<density_tri;k++){
```

```
57                              tris[k] = new Tri(getRandomX(), getRandomY(),
                                     getRandomVelocity(), getRandomVelocity(),
                                    getRandomX(), getRandomY(),
                                    getRandomVelocity(), getRandomVelocity(),
                                    getRandomX(), getRandomY(),
                                    getRandomVelocity(), getRandomVelocity(),"
                                    rgba("+getRandomColor()+","+getRandomColor
                                    ()+","+getRandomColor()+","+
                                    getRandomAlpha()+")",getRandomWidth(),
                                    1);
58                              tris[k].draw();
59                      }
60
61                      //実際の描画処理
62                      draw();
63
64                      //一定時間ごとに書き換える
65                      setInterval("draw()",33);
66
67              }
68
69              //オブジェクト初期化処理
70              var Particle = function(scale, color, vx, vy, gv){
71                      this.scale = scale;
72                      this.color = color;
73                      this.vx = vx;
74                      this.vy = vy;
75                      this.gv = gv;
76                      this.position = {
77                              x: 100,
78                              y: 100
79                      };
80              };
81              Particle.prototype.draw = function() {
82                      ctx.beginPath();
83                      ctx.arc(this.position.x, this.position.y, this.scale,
                            0, 2*Math.PI, false);
84                      ctx.fillStyle = this.color;
85                      ctx.fill();
86              };
87              Particle.prototype.update = function(){
88                      this.vy += this.gv;
89                      this.position.x += this.vx;
90                      this.position.y += this.vy;
91                      this.draw();
92                      if(this.position.x > cw) this.position.x -= cw;
93                      if(this.position.y > ch) this.vy = -this.vy;
94              };
```

```
95
96                          var Line = function(x1, y1, vx1, vy1, x2, y2, vx2, vy2, color
                                , width, gv){
97                              this.position1 = {x:x1, y:y1};
98                              this.v1 = {x:vx1, y:vy1};
99                              this.position2 = {x:x2, y:y2};
100                             this.v2 = {x:vx2, y:vy2};
101                             this.color = color;
102                             this.width = width;
103                             this.gv = gv;
104                         };
105                         Line.prototype.draw = function(){
106                             ctx.strokeStyle = this.color;
107                             ctx.lineWidth = this.width;
108                             ctx.beginPath();
109                             ctx.moveTo(this.position1.x, this.position1.y);
110                             ctx.lineTo(this.position2.x, this.position2.y);
111                             ctx.stroke();
112                         };
113                         Line.prototype.update = function(){
114                             this.v1.y += this.gv;
115                             this.v2.y += this.gv;
116                             this.position1.x += this.v1.x;
117                             this.position1.y += this.v1.y;
118                             this.position2.x += this.v2.x;
119                             this.position2.y += this.v2.y;
120                             this.draw();
121                             if(this.position1.y > ch) this.v1.y = -this.v1.y;
122                             if(this.position2.y > ch) this.v2.y = -this.v2.y;
123                         };
124                         var Tri = function(x1, y1, vx1, vy1, x2, y2, vx2, vy2, x3, y3
                                , vx3, vy3, color, width, gv){
125                             this.position1 = {x:x1, y:y1};
126                             this.v1 = {x:vx1, y:vy1};
127                             this.position2 = {x:x2, y:y2};
128                             this.v2 = {x:vx2, y:vy2};
129                             this.position3 = {x:x3, y:y3};
130                             this.v3 = {x:vx3, y:vy3};
131                             this.color = color;
132                             this.width = width;
133                             this.gv = gv;
134                         };
135                         Tri.prototype.draw = function(){
136                             ctx.strokeStyle = this.color;
137                             ctx.lineWidth = this.width;
138                             ctx.beginPath();
139                             ctx.moveTo(this.position1.x, this.position1.y);
140                             ctx.lineTo(this.position2.x, this.position2.y);
```

```
141                         ctx.lineTo(this.position3.x, this.position3.y);
142                         ctx.lineTo(this.position1.x, this.position1.y);
143                         ctx.stroke();
144                 };
145                 Tri.prototype.update = function(){
146                         this.v1.y += this.gv;
147                         this.v2.y += this.gv;
148                         this.v3.y += this.gv;
149                         this.position1.x += this.v1.x;
150                         this.position1.y += this.v1.y;
151                         this.position2.x += this.v2.x;
152                         this.position2.y += this.v2.y;
153                         this.position3.x += this.v3.x;
154                         this.position3.y += this.v3.y;
155                         this.draw();
156                         if(this.position1.y > ch) this.v1.y = -this.v1.y;
157                         if(this.position2.y > ch) this.v2.y = -this.v2.y;
158                         if(this.position3.y > ch) this.v3.y = -this.v3.y;
159                 };
160
161                 function draw(){
162                         //画面をリセットする
163                         ctx.fillStyle = "rgba(255,255,255,1)";
164                         ctx.fillRect(0,0,cw,ch);
165
166                         //円の描画
167                         for(var i=0;i < density; i++){
168                                 particles[i].update();
169                         }
170
171                         //線の描画
172                         for(var j=0;j < density_line; j++){
173                                 lines[j].update();
174                         }
175                         //三角形の描画
176                         for(var k=0;k < density_tri; k++){
177                                 tris[k].update();
178                         }
179                 }
180
181                 function getRandomColor(){
182                         return Math.floor(Math.random()*255);
183                 }
184                 function getRandomAlpha(){
185                         return Math.random();
186                 }
187                 function getRandomScale(){
188                         return (Math.random()*(8-3))+3;
```

```
189                         }
190                         function getRandomX(){
191                                 return Math.random()*cw;
192                         }
193                         function getRandomY(){
194                                 return Math.random()*ch;
195                         }
196                         function getRandomWidth(){
197                                 return Math.random()*10;
198                         }
199                         function getRandomVelocity(){
200                                 return Math.random()*20-10;
201                         }
202
203                 </script>
204         </head>
205         <body onload="init();">
206                 <H1>Canvas Test</H1>
207                 <canvas id="mycanvas" width="1000" height="800"></canvas>
208         </body>
209 </html>
```

以上