

CCOD
Lab 2 15.09.2016
Akira Imada
Student – Aleksey Trotsiuk (AS - 36)

We have a metro which length is 1000px. We have 2 trains. They move in clockwise direction. Each train has start speed, which is 10. Max speed is 20, minimum speed is 0 (we do not allow to exceed or to be less than 0). On each iteration, speed of a train could be changed (from -2 to 2). We also have break value. If speed ≥ 14 then break = 1. If distance ≤ 50 then break = 9. Default break value – 0.

We have 3 rules

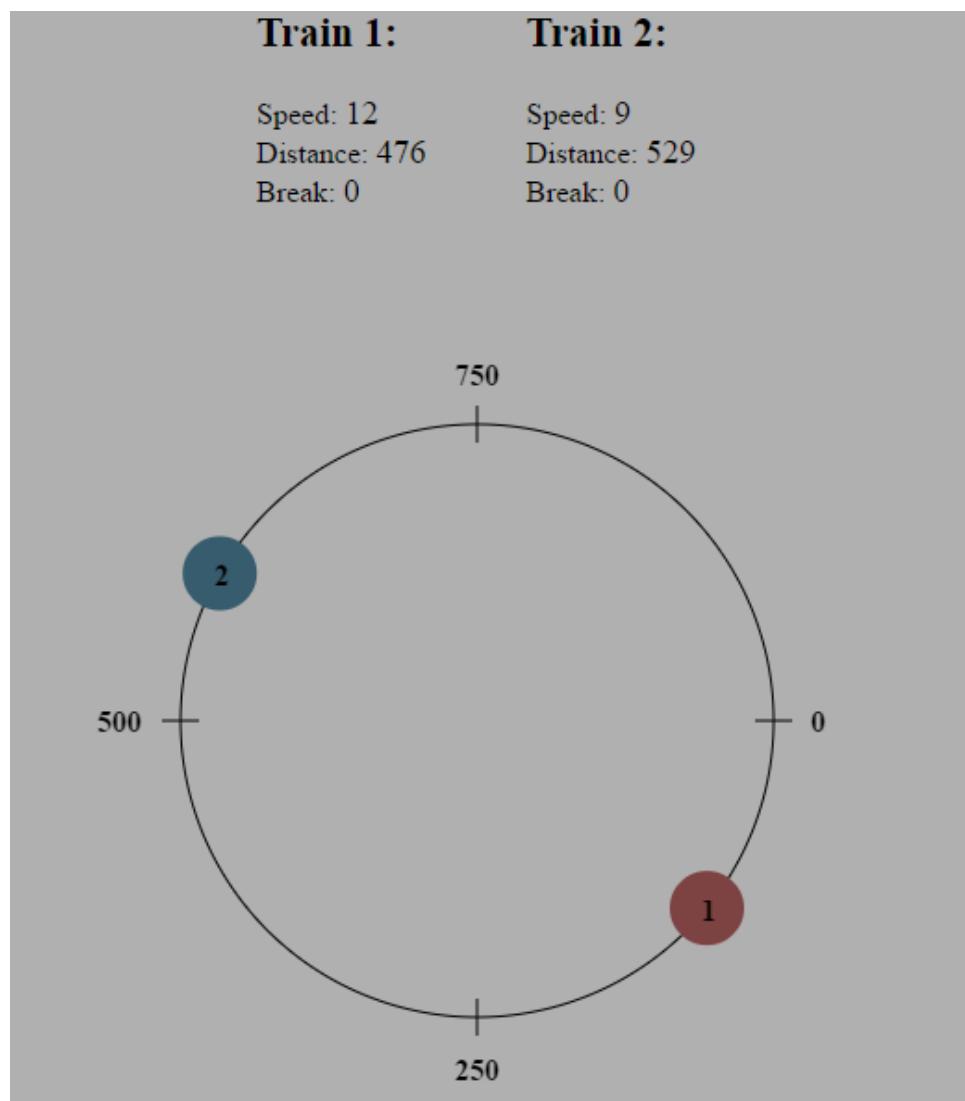
{IF x=medium AND y=small THEN z=strong} OR

{IF x=medium AND y=medium THEN z=medium} OR

{IF x=medium AND y=large THEN z=weak}

Below I attach 5 snapshots of trains:

1.



Train 1:

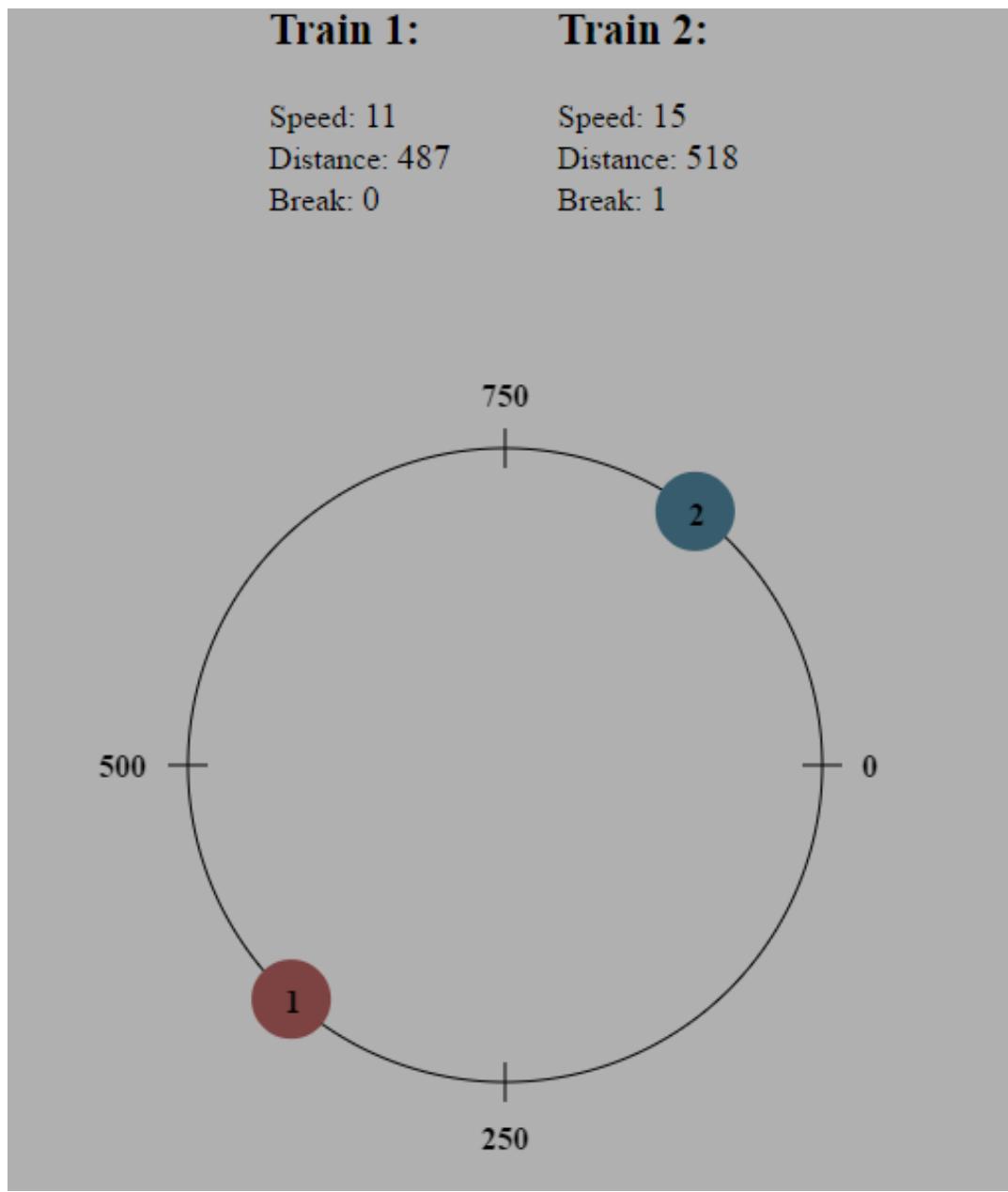
						U
Rule 1		0.5		0		0
	If	x=medium	&	y=small	=>	z=strong
Rule 2		0.5		0.65		0
	If	x=medium	&	y=medium	=>	z=medium
Rule 3		0.5		0.35		0
	If	x=medium	&	y=large	=>	z=week
Max						0

Train 2

						U
Rule 1		0.75		0		0
	If	x=medium	&	y=small	=>	z=strong
Rule 2		0.75		0.35		0
	If	x=medium	&	y=medium	=>	z=medium
Rule 3		0.75		0.65		0
	If	x=medium	&	y=large	=>	z=week
Max						0

As you can see, U is always equal to 0, because z can be 0, 1, 9 (it will be 0 in strong, medium and week states)

2.

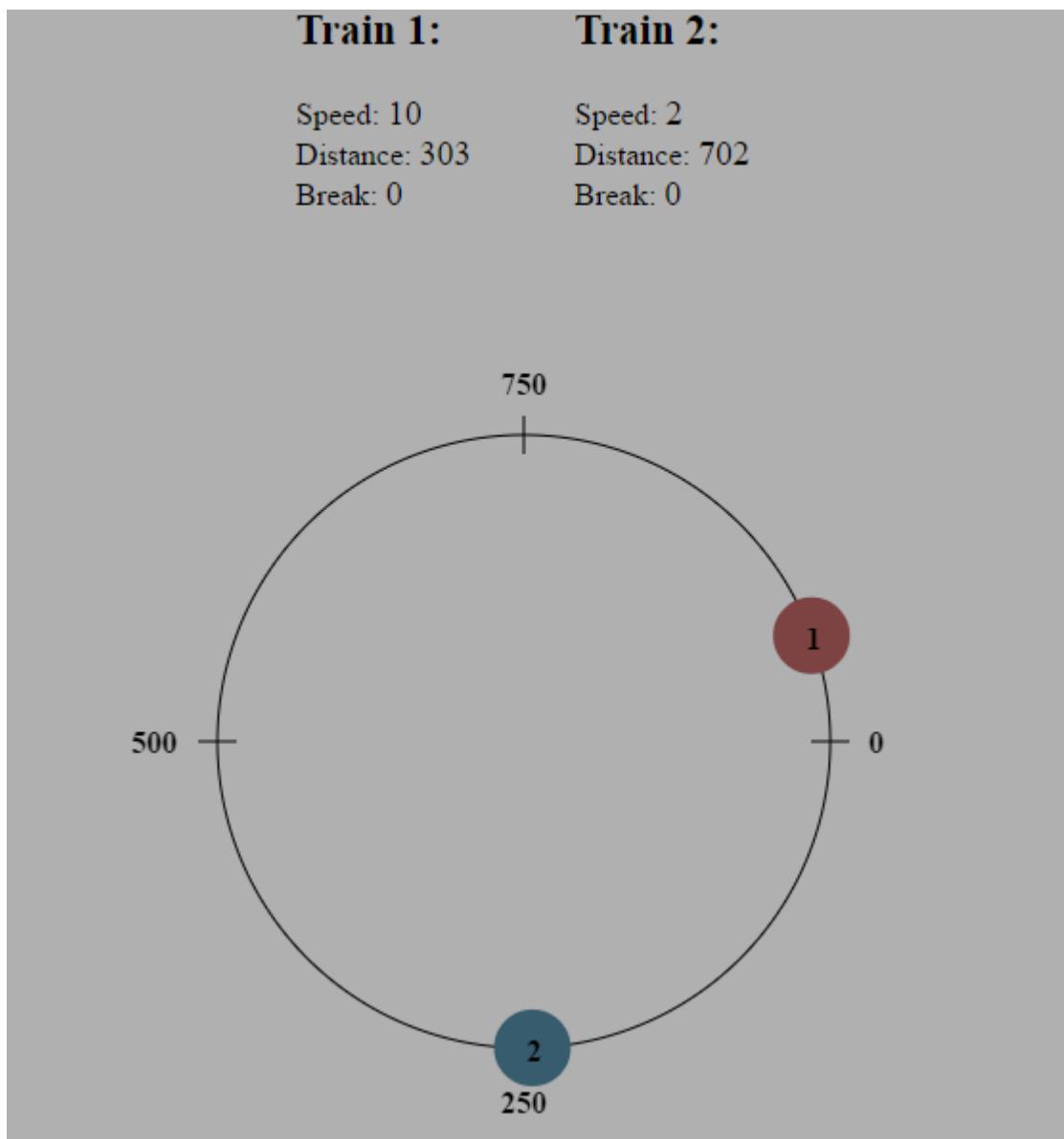


Train 1:

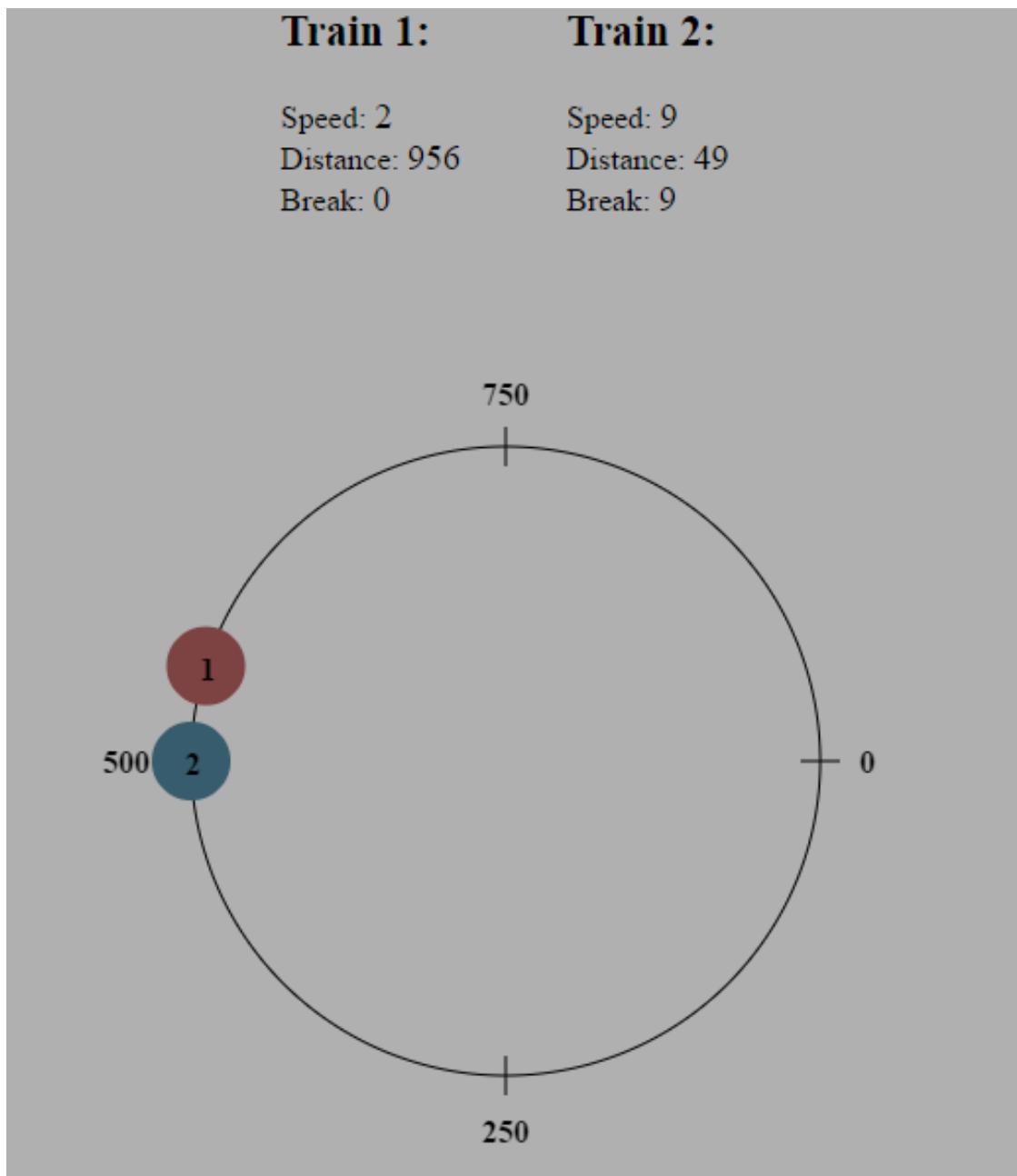
Train 2

						U
Rule 1		0		0		0
	If	x=medium	&	y=small	=>	z=strong
Rule 2		0		0.45		0
	If	x=medium	&	y=medium	=>	z=medium
Rule 3		0		0.55		0
	If	x=medium	&	y=large	=>	z=week
Max						0

3.



4.



Train 1:

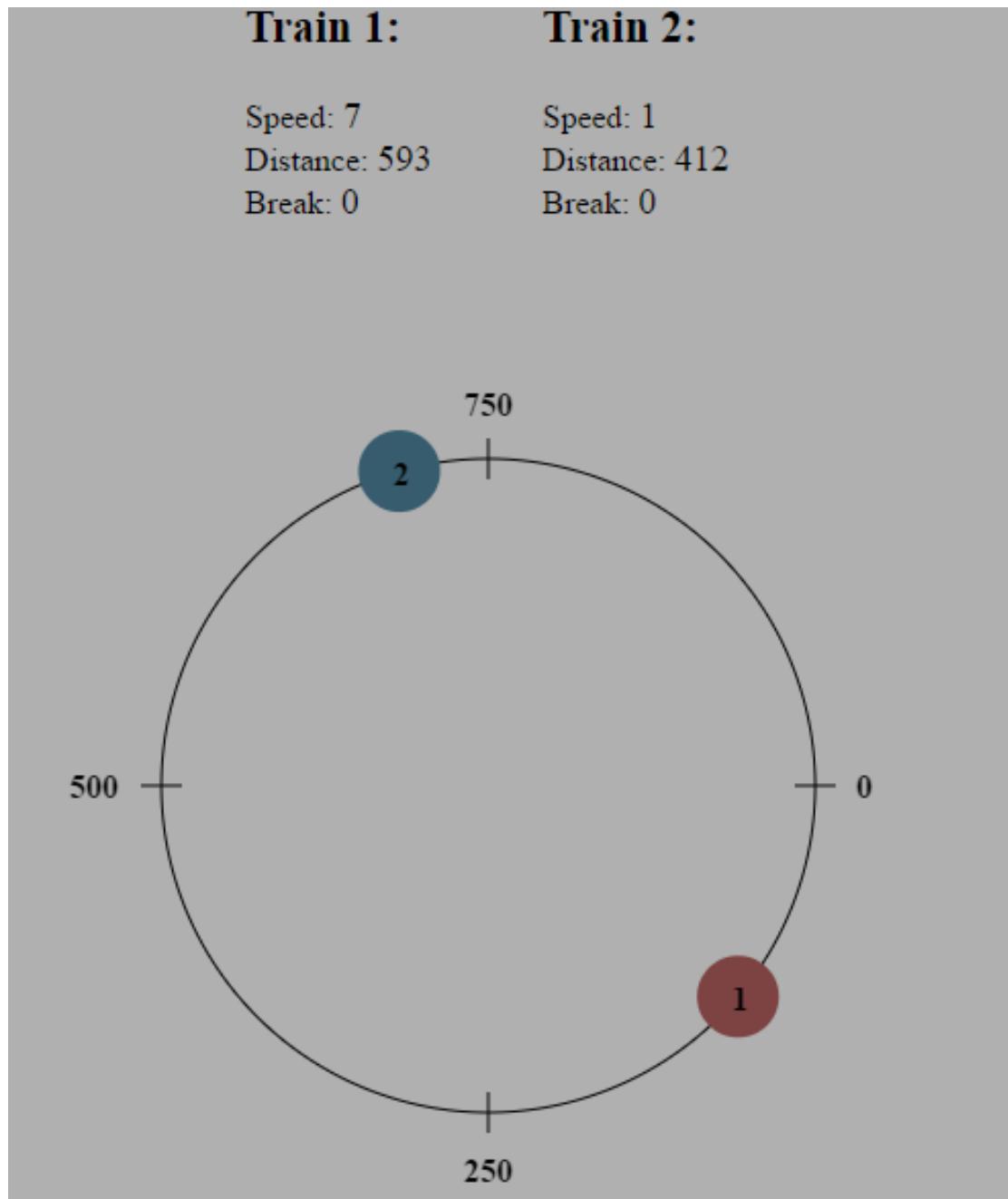
							U
Rule 1	0	0	0	0	0	0	0
	If	x=medium	&	y=small	=>	z=strong	
Rule 2	0	0	0	0	0	0	0
	If	x=medium	&	y=medium	=>	z=medium	
Rule 3	0	0	0	0	0	0	0
	If	x=medium	&	y=large	=>	z=weak	

Max	0
-----	---

Train 2

						U
Rule 1		0.75		0		0
	If	x=medium	&	y=small	=>	z=strong
Rule 2		0.75		0		0
	If	x=medium	&	y=medium	=>	z=medium
Rule 3		0.75		0		0
	If	x=medium	&	y=large	=>	z=week
	Max					0

5.



Train 1:

							U
Rule 1		0.25		0		0	0
	If	x=medium	&	y=small	=>	z=strong	
Rule 2		0.25		0		0	0
	If	x=medium	&	y=medium	=>	z=medium	
Rule 3		0.25		1		0	0
	If	x=medium	&	y=large	=>	z=week	
Max							0

Train 2

							U
Rule 1		0		0		0	0
	If	x=medium	&	y=small	=>	z=strong	
Rule 2		0		1		0	0
	If	x=medium	&	y=medium	=>	z=medium	
Rule 3		0		0		0	0
	If	x=medium	&	y=large	=>	z=week	
Max							0

SOURCE CODES:

[Index.html](#)

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Lab 1</title>
5          <link rel="stylesheet" type="text/css" href="styles.css">
6      </head>
7      <body>
8          <div class="main-content">
9              <div class="trains-info">
10                 <div class="train-content">
11                     <p class="title">Train 1:</p>
12                     <ul class="train-data">
13                         <li>Speed: <p id="train-speed1"></p></li>
14                         <li>Distance: <p id="train-distance1"></p></li>
15                         <li>Break: <p id="train-break1"></p></li>
16                     </ul>
17                 </div>
18                 <div class="train-content">
19                     <p class="title">Train 2:</p>
20                     <ul class="train-data">
21                         <li>Speed: <p id="train-speed2"></p></li>
22                         <li>Distance: <p id="train-distance2"></p></li>
23                         <li>Break: <p id="train-break2"></p></li>
24                     </ul>
25                 </div>
26             </div>
27             <canvas id="canvas" width="500" height="500"></canvas>
28         </div>
29         <script src = "script/index.js"></script>
30     </body>
31 </html>
```

Styles.css

```
1  .main-content {
2      position: relative;
3      text-align: center;
4  }
5
6  .main-content .trains-info .train-content {
7      display: inline-block;
8      text-align: left;
9      padding: 25px;
10 }
11
12 .main-content .trains-info .train-content .title {
13     margin: 0;
14     font-size: 22px;
15     font-weight: bold;
16 }
17
18 .main-content .trains-info .train-content .train-data {
19     list-style: none;
20     padding: 20px 0 0 0;
21     margin: 0;
22 }
23
24 .main-content .trains-info .train-content .train-data p {
25     display: inline;
26     margin: 0;
27     padding: 0;
28     font-size: 18px;
29 }
```

```
1  (function() {
2    "use strict";
3
4    const WIDTH = 1000;
5    const HEIGHT = 1000;
6    const X_CENTER = 250;
7    const Y_CENTER = 250;
8    const RADIUS = 160;
9    const TRAIN_RADIUS = 20;
10   const ARC_ANGLE = 2 * Math.PI;
11   const MOVE_UPDATE = 30;
12   const SPEED_UPDATE = 29;
13   const TRAIN_NUMBERS = 2;
14   const MAX_SPEED = 20;
15   const MIN_SPEED = 0;
16   const BREAK_SPEED = 14;
17   const BREAK_DISTANCE = 50;
18
19   let canvas = document.getElementById("canvas");
20   let context = canvas.getContext("2d");
21   let trains = [{{
22     angle: 0,
23     speed: 10,
24     distance: 500,
25     break: 0,
26     color: "#b55f5f"
27   },
28   {{
29     angle: 180,
30     distance: 500,
31     speed: 10,
32     break: 0,
33     color: "#50869e"
34   }}];
35   let moveTrain;
```

```
36     let changeTrainSpeed;
37
38     init();
39
40     function init() {
41         updateWindow();
42
43         moveTrain = setInterval(move, MOVE_UPDATE);
44         changeTrainSpeed = setInterval(changeSpeed, SPEED_UPDATE);
45     }
46
47     function move() {
48         updateWindow();
49         drawTrains();
50         changeTrainsAngel();
51         if (isBreak()) {
52             stop();
53         }
54     }
55
56     function changeSpeed() {
57         for (let i = 0; i < trains.length; i++) {
58             trains[i].speed += getRandomInt(-2, 3) - trains[i].break;
59             if (trains[i].speed >= BREAK_SPEED || trains[i].distance <= BREAK_DISTANCE) {
60                 if (trains[i].speed >= BREAK_SPEED) {
61                     trains[i].break = 1;
62                 }
63                 if (trains[i].distance <= BREAK_DISTANCE) {
64                     trains[i].break = 9;
65                 }
66             } else {
67                 trains[i].break = 0;
68             }
69
70             if (trains[i].speed > MAX_SPEED) {
71                 trains[i].speed = MAX_SPEED;
72             }
73             if (trains[i].speed < MIN_SPEED) {
74                 trains[i].speed = MIN_SPEED
75             }
76             let trainSpeed = "train-speed" + (i + 1);
77             let trainDistance = "train-distance" + (i + 1);
78             let trainBreak = "train-break" + (i + 1);
79             let speed = document.getElementById(trainSpeed);
80             let distance = document.getElementById(trainDistance);
81             let trBreak = document.getElementById(trainBreak);
82             speed.innerHTML = trains[i].speed;
83             distance.innerHTML = trains[i].distance;
84             trBreak.innerHTML = trains[i].break;
85         }
86     }
87
88     function stop() {
89         updateWindow();
90         drawTrains();
91         clearInterval(moveTrain);
92         clearInterval(changeTrainSpeed);
93     }
94
95     function drawTrains() {
96         for (let i = 0; i < trains.length; i++) {
97             let position = getTrainPosition(trains[i].angle);
98
99             context.beginPath();
100            context.arc(position.x, position.y, TRAIN_RADIUS, 0, ARC_ANGLE);
101            context.fillStyle = trains[i].color;
102            context.fill();
```

```

102         context.fillStyle = "black";
103         context.font = "bold 16px serif";
104         context.textBaseline = "top";
105         context.fillText(i + 1, position.x - 3, position.y - 7);
106     }
107 }
108
109 function getTrainPosition(angle) {
110     let y = RADIUS * Math.sin(toRadian(angle)) + Y_CENTER;
111     let x = RADIUS * Math.cos(toRadian(angle)) + X_CENTER;
112     return {
113         x: x,
114         y: y
115     };
116 }
117
118 function changeTrainsAngel() {
119     for (let i = 0; i < trains.length; i++) {
120         trains[i].angle += trains[i].speed / 5;
121     }
122 }
123
124 function toRadian(angle) {
125     return angle * Math.PI / 180;
126 }
127
128 function isBreak() {
129     trains[0].distance = Math.round(RADIUS * toRadian(trains[1].angle - trains[0].angle));
130     trains[1].distance = Math.round(1005 - trains[0].distance);
131     return trains[0].distance <= 0 || trains[1].distance <= 0;
132 }
133
134 function updateWindow() {
135     context.clearRect(0, 0, WIDTH, HEIGHT);
136     context.beginPath();
137     context.arc(X_CENTER, Y_CENTER, RADIUS, 0, ARC_ANGLE);
138
139     context.fillText(750, X_CENTER - 12, Y_CENTER - RADIUS - 35);
140     context.moveTo(X_CENTER, Y_CENTER - RADIUS - 10);
141     context.lineTo(X_CENTER, Y_CENTER - RADIUS + 10);
142
143     context.fillText(250, X_CENTER - 12, Y_CENTER + RADIUS + 20);
144     context.moveTo(X_CENTER, Y_CENTER + RADIUS - 10);
145     context.lineTo(X_CENTER, Y_CENTER + RADIUS + 10);
146
147     context.fillText(500, X_CENTER - RADIUS - 45, Y_CENTER - 8);
148     context.moveTo(X_CENTER - RADIUS - 10, Y_CENTER);
149     context.lineTo(X_CENTER - RADIUS + 10, Y_CENTER);
150
151     context.fillText(0, X_CENTER + RADIUS + 20, Y_CENTER - 8);
152     context.moveTo(X_CENTER + RADIUS - 10, Y_CENTER);
153     context.lineTo(X_CENTER + RADIUS + 10, Y_CENTER);
154
155     context.stroke();
156 }
157
158 function getRandomInt(min, max) {
159     return Math.floor(Math.random() * (max - min)) + min;
160 }
161
162 })();

```