

```
1  <html>
2    <head>
3      <title>Tank Simulation</title>
4
5      <script language="javascript">
6        //
7
8        //makes variables
9        var invaderSpeedX = 5;
10       var invaderSpeedY = 40;
11
12       var invaderHealth = 10;
13
14       var colorOne = "#0000FF";
15
16       //defines the boundaries of the canvas as varaibles to make it easier to hot
17       swap
18       var canvasBoundX = 700;
19       var canvasBoundY = 500;
20
21       var speed = 6;
22
23       var xDef = 350;
24       var yDef = 450;
25
26       var xInvaders = 350;
27       var yInvaders = 50;
28
29       var targetX = Math.floor(Math.random() * (canvasBoundX - 100)) + 50;
30       var targetY = Math.floor(Math.random() * (canvasBoundY - 100)) + 50;
31       var targetActive = true;
32
33       var currentMovement = "none";
34       var previousMovement = "none";
35
36       var bulletShot = false;
37       var bulletX = 0;
38       var bulletY = 0;
39       var bulletDirection = "none";
40
41       var timestamp = 0;
42       var totalTime = 60000;
43       var totalShots = 0;
44       var targetsHit = 0;
45
46       var isOn = false;
47       //used for timer to determine if it is on or not
48
49       //adds keyboard listener
50       window.addEventListener("keydown", function(event){
51         //changes keyboard input into diretions
52         switch(event.key){
53           case "a":
54             currentMovement = "left";
55             break;
56           case "d":
57             currentMovement = "right";
58             break;
59           case "w":
60             currentMovement = "up";
61             break;
62           case "s":
63             currentMovement = "down";
64             break;
65           default:
66             currentMovement = "none";
67             break;
```

```

67         }
68
69         if(currentMovement != "none")
70             previousMovement = currentMovement;
71
72     }, true);
73
74     window.addEventListener("keyup", function(event){
75         currentMovement = "none";
76
77     }, true);
78
79
80     //fill background and turn on timer when the body gets initialized
81     function initialize(){
82         var canvas = document.getElementById("myCanvas");
83         var context = canvas.getContext("2d");
84
85         context.fillStyle="#ADD8E6";
86         context.fillRect(0, 0, canvasBoundX, canvasBoundY);
87
88         turnOn();
89     }
90
91     //periodically called ever 20ms to update the game
92     function update(){
93         resetBackground();
94         writeText(timestamp / 1000, 10, 50);
95         writeText(totalTime / 1000, 10, 100);
96         writeText("Total Shots: " + totalShots, 500, 50);
97         writeText("Points: " + targetsHit, 500, 90);
98         moveDef();
99         drawTarget();
100        moveBullet();
101        timestamp += 20;
102        totalTime -= 20;
103
104        if(totalTime <= 0){
105            toggleTimer();
106            resetBackground();
107            writeText("Game Over!", canvasBoundX / 2, canvasBoundY / 2);
108        }
109    }
110
111    //writes text on the screen given what to write and the location
112    function writeText(text, x, y){
113        var canvas = document.getElementById("myCanvas");
114        var context = canvas.getContext("2d");
115
116        context.font = "30px Arial";
117        context.fillStyle = "#FF0000";
118        context.fillText(text, x, y);
119    }
120
121    //moves the bullet across the screen based on the direction
122    function moveBullet(){
123        if(bulletShot == true){
124            var canvas = document.getElementById("myCanvas");
125            var context = canvas.getContext("2d");
126
127            if(bulletDirection == "left")
128                bulletX -= 10;
129            else if(bulletDirection == "right")
130                bulletX += 10;
131            else if(bulletDirection == "down")
132                bulletY += 10;
133            else if(bulletDirection == "up")

```

```

134         bulletY -= 10;
135     else
136         bulletX -= 10;
137
138     context.fillStyle = "#FF0000";
139     context.beginPath();
140     context.arc(bulletX, bulletY, 8, 0, 2 * Math.PI, true);
141     context.closePath();
142     context.fill();
143
144     //if the bullet hits anything reset it
145     if(bulletY <= 10 || bulletY >= canvasBoundY - 10 || bulletX <= 10 ||
bulletX >= canvasBoundX - 10 || hitTarget()){
146         context.strokeStyle = "#ADD8E6";
147         context.stroke();
148         bulletShot = false;
149     }
150
151     //if it hits the target specifically then reset the target
152     if(hitTarget()){
153         targetActive = false;
154         targetsHit++;
155         placeTarget();
156     }
157 }
158 }
159
160 //places the target to the given coordinate
161 function drawTarget(){
162     var canvas = document.getElementById("myCanvas");
163     var context = canvas.getContext("2d");
164
165     if(targetActive){
166         context.fillStyle = "#FF0000";
167         context.beginPath();
168         context.arc(targetX, targetY, 16, 0, 2 * Math.PI, true);
169         context.closePath();
170         context.fill();
171
172         context.fillStyle = "#FFFFFF";
173         context.beginPath();
174         context.arc(targetX, targetY, 12, 0, 2 * Math.PI, true);
175         context.closePath();
176         context.fill();
177
178         context.fillStyle = "#FF0000";
179         context.beginPath();
180         context.arc(targetX, targetY, 8, 0, 2 * Math.PI, true);
181         context.closePath();
182         context.fill();
183
184         context.fillStyle = "#FFFFFF";
185         context.beginPath();
186         context.arc(targetX, targetY, 4, 0, 2 * Math.PI, true);
187         context.closePath();
188         context.fill();
189
190         if(timestamp > 1500){
191             targetActive = false;
192             placeTarget();
193         }
194     }
195 }
196
197 //redraws the target at a different location
198 function placeTarget(){
199     if(!targetActive){

```

```

200         targetX = Math.floor(Math.random() * (canvasBoundX - 100)) + 50;
201         targetY = Math.floor(Math.random() * (canvasBoundY - 100)) + 50;
202
203         timestamp = 0;
204         targetActive = true;
205     }
206 }
207
208 //toggles firing the bullet
209 function fireBullet(){
210     if(!bulletShot){
211         bulletX = xDef;
212         bulletY = yDef;
213         bulletDirection = previousMovement;
214         totalShots++;
215     }
216
217     bulletShot = true;
218
219 }
220
221 //moves the tank based on the given direction
222 function moveDef(){
223     switch(currentMovement){
224         case "left":
225             if(xDef > 20)
226                 xDef -= speed;
227             moveDefender(xDef, yDef, previousMovement);
228             break;
229         case "right":
230             if(xDef < canvasBoundX - 20)
231                 xDef += speed;
232             moveDefender(xDef, yDef, previousMovement);
233             break;
234         case "up":
235             if(yDef >= 20)
236                 yDef -= speed;
237             moveDefender(xDef, yDef, previousMovement);
238             break;
239         case "down":
240             if(yDef <= canvasBoundY - 20)
241                 yDef += speed;
242             moveDefender(xDef, yDef, previousMovement);
243             break;
244         default:
245             moveDefender(xDef, yDef, previousMovement);
246             break;
247     }
248 }
249
250
251 //used to reset the background to avoid smearing
252 function resetBackground(){
253     var canvas = document.getElementById("myCanvas");
254     var context = canvas.getContext("2d");
255     //paint the background of the canvas
256     context.fillStyle="#ADD8E6";
257     context.fillRect(0, 0, canvasBoundX, canvasBoundY);
258 }
259
260 //places the defender to a given location and moves the turret based on the
direction
261 function moveDefender(x, y, direction){
262     var canvas = document.getElementById("myCanvas");
263     var context = canvas.getContext("2d");
264
265     var directX = 0;

```

```

266         var directY = 0;
267
268         context.fillStyle = colorOne;
269         context.beginPath();
270         context.arc(x, y, 15, 0, 2 * Math.PI, true);
271         context.closePath();
272         context.fill();
273
274         if(direction == "up")
275             directY = -1;
276         else if(direction == "down")
277             directY = 1;
278         else if(direction == "left")
279             directX = -1;
280         else if(direction == "right")
281             directX = 1;
282
283         context.fillStyle = colorOne;
284         context.beginPath();
285         context.arc(x + (15 * directX), y + (15 * directY), 7.5, 0, 2 * Math.PI,
286             true);
287         context.closePath();
288         context.fill();
289     }
290
291     //returns true if the bullet is within the hitbox of the target
292     function hitTarget(){
293         return Math.abs(bulletX - targetX) < 16 && Math.abs(bulletY - targetY) <
294             16;
295     }
296
297     //toggles the timer
298     function toggleTimer(){
299         //if the timer is on the turn it off, if its not then turn it on
300         isOn ? turnOff() : turnOn();
301     }
302
303     //manually turns on the timer
304     function turnOn(){
305         //turns the timer on
306         timer = setInterval("update()", 20);
307         isOn = true;
308     }
309
310     //manually turns off the timer
311     function turnOff(){
312         //turns the timer off
313         clearInterval(timer);
314         isOn = false;
315     }
316 }
317
318 </script>
319 </head>
320
321 <body onload="initialize()" onclick="fireBullet()">
322     <center>
323         <h1>Tank Simulation</h1>
324
325     </br>
326     </br>
327
328     <canvas id="myCanvas" width="700" height="500"
329         style="border:2px solid rgb(195, 195, 195);">
330         Your browser does not support the canvas element
331     </canvas>
332
333 </br>
334 </br>

```

```
331         </center>
332     </body>
333 </html>
```