

به نام خدا

پروژه ی درس گرافیک کامپیوتری استاد مینوفام

بازی پینگ پونگ

تیر 1399

سارا سلطان محمدی

نرگس محمدیاری

```

12
13 function setRenderer() {
14
15     renderer = new THREE.WebGLRenderer({alpha: true});
16     renderer.setPixelRatio(window.devicePixelRatio);
17     renderer.setSize(window.innerWidth, window.innerHeight);
18     document.body.appendChild(renderer.domElement);
19 }
20
21
22 function setCamera() {
23
24     camera = new THREE.PerspectiveCamera(70, window.innerWidth / window.innerHeight, 1, 1000);
25
26     camera.position.x = 0;
27     camera.position.y = 0;
28     camera.position.z = 5;
29 }
30
31
32 function setScene() {
33
34     scene = new THREE.Scene();
35 }

```

ما در ابتدا از تابع `setRender()` برای ایجاد `Render` استفاده میکنیم.

سپس از تابع `setCamera()` برای ایجاد زاویه ی دید و تعیین آن استفاده میکنیم.

در مرحله ی بعد با کمک تابع `setScence` محیط کلی بازی را ایجاد میکنیم.

```

37 function setWorld() {
38     x_map = 5;
39     y_map = 4;
40     x_mile = 0.2;
41     y_mile = 1;
42
43
44     /* ezafe kardane map */
45     var geometry = new THREE.BoxGeometry(x_map, y_map, 0.01);
46     var material = new THREE.MeshPhongMaterial({color: 0xFFC77C, side: THREE.DoubleSide});
47     map = new THREE.Mesh(geometry, material);
48     scene.add(map);
49
50     var surfaceGeometry = new THREE.PlaneGeometry(300,350);
51     var surfaceMaterial = new THREE.MeshBasicMaterial({color: 0x828282, side: THREE.DoubleSide});
52     var surfaceMesh = new THREE.Mesh(surfaceGeometry, surfaceMaterial);
53     scene.add(surfaceMesh);
54
55
56     /* ezafe kardane mile player 1 */
57     var geometry = new THREE.BoxGeometry(x_mile, y_mile, 0.1);
58     var material = new THREE.MeshPhongMaterial({color: 0x005000});
59     player_1 = new THREE.Mesh(geometry, material);
60     player_1.position.x = -x_map / 2;
61     scene.add(player_1);
62
63     /* ezafe kardane mile player 2 */
64     var geometry = new THREE.BoxGeometry(x_mile, y_mile, 0.1);
65     var material = new THREE.MeshPhongMaterial({color: 0xff0000});
66     player_2 = new THREE.Mesh(geometry, material);
67     player_2.position.x = x_map / 2;
68     scene.add(player_2);
69

```

 IDE and Plugin Updates

با کمک تابه `setWorld()` اجسامی و اشیایی که در بازی استفاده کردیم را به بازی اضافه میکنیم. و به طور مثال برای میله ها (راکت ها) با کمک `x` و `y` اندازه هارا معین میکنیم و به آنها رنگ بخصوص میدهیم.

```

82
83     var s = 0;
84
85     function recovery_on_player1(recover_speed) {
86         ball.position.copy(player_1.position);
87         console.log(recover_speed)
88         ball_speed = -recover_speed;
89         s = 0;
90     }
91
92     function recovery_on_player2(recover_speed) {
93         ball.position.copy(player_2.position);
94         console.log(recover_speed)
95         console.log(player_1.position)
96         console.log(player_2.position)
97         console.log(ball.position)
98         ball_speed = -recover_speed;
99         s = 0;
100     }
101
102     function get_random_angle(minimum, maximum) {
103
104         var randomnumber = Math.random() * ( maximum - minimum ) + minimum;
105
106         return randomnumber;
107     }
108
109     var ball_angle = Math.PI;
110     var player2_speed = 0.05;
111

```

در مرحله ی بعد در تابع های recovery موقعیت راکت و سرعت توپ بعد از یک دور بازی تعیین شده است.

Flag تعریف شده تحت عنوان s که وقتی $s=0$ آنگاه توپ رها شده و وقتی $s=1$ توپ نگه داشته شده.

از تابع رندوم برای تعیین زاویه ی برگشت توپ استفاده کردیم.

```

111
112 function animate() {
113
114     requestAnimationFrame(animate);
115
116     ball.position.x += ball_speed * Math.cos(ball_angle);
117     ball.position.y += ball_speed * Math.sin(ball_angle);
118
119
120     /* barkhord player 1 */
121     if (( ball.position.x < player_1.position.x + (x_mile / 2) ) &&
122         ( ball.position.y < ( player_1.position.y + y_mile / 2 ) ) &&
123         ( ball.position.y > ( player_1.position.y - y_mile / 2 ) )) {
124         var snd = new Audio("sounds/ding.wav");
125         snd.play();
126         if (s == 0) {
127             ball.position.x = player_1.position.x + (x_mile / 2);
128             ball_speed = -ball_speed;
129             ball_angle = get_random_angle(-Math.PI / 4, Math.PI / 4);
130         }
131     }
132
133     /* barkhord player 2 */
134     if (( ball.position.x > player_2.position.x - (x_mile / 2) ) &&
135         ( ball.position.y < ( player_2.position.y + y_mile / 2 ) ) &&
136         ( ball.position.y > ( player_2.position.y - y_mile / 2 ) )) {
137         var snd = new Audio("sounds/ding.wav");
138         snd.play();
139         if (s == 0) {
140             ball.position.x = player_2.position.x - (x_mile / 2);
141             ball_speed = -ball_speed;
142             ball_angle = get_random_angle(-Math.PI / 4, Math.PI / 4);
143         }
144     }

```

در ادامه یکی از چالش ها چگونگی تشخیص برخورد توپ با راکت بود که با کمک یکی از منابع (که آنرا ضمیمه کردیم) توانستیم آنرا بنویسیم.

```

245
246 function setEventListenerHandler() {
247     window.addEventListener('keydown', function (e) {
248         keyState[e.keyCode || e.which] = true;
249     }, true);
250
251     window.addEventListener('keyup', function (e) {
252         keyState[e.keyCode || e.which] = false;
253     }, true);
254
255     window.addEventListener('resize', onWindowResize, false);
256 }
257
258
259 function setKeyboardControls() {
260
261     if (keyState[81]) {
262         //q
263         if (player_1.position.y < ( (y_map / 2) - ( y_mile / 2 ))) {
264             player_1.position.y += 0.075;
265         }
266     }
267
268     //a
269     if (keyState[65]) {
270         if (player_1.position.y > ( -(y_map / 2) + ( y_mile / 2 ))) {
271             player_1.position.y -= 0.075;
272         }
273     }
274
275     if (keyState[38]) {
276         if (player_2.position.y < ( (y_map / 2) - ( y_mile / 2 ))) {
277             player_2.position.y += 0.075;
278         }
279     }
280 }

```

سپس با کمک تابع setKeyboardControls() برای انجام عملیات های مختلف روی کیبورد کمک میگیریم.

```

298
299 function setLights() {
300
301     var light = new THREE.AmbientLight(0xffffff);
302     scene.add(light);
303
304
305     var spotLight = new THREE.SpotLight(0xFFC77C);
306     spotLight.position.set(0, 0, 2);
307
308     spotLight.castShadow = true;
309
310     spotLight.shadow.mapSize.width = window.innerWidth;
311     spotLight.shadow.mapSize.height = window.innerHeight;
312
313     scene.add(spotLight);
314 }
315
316 function harder() {
317     player1_score = 0;
318     player2_score = 0;
319     document.getElementById("player1_score").innerHTML = player1_score;
320     document.getElementById("player2_score").innerHTML = player2_score;
321
322     if (s == 0) {
323         ball.position.x = 0;
324         ball.position.y = 0;
325         ball_speed = -0.1;
326
327         ball_angle = Math.PI;
328
329     }
330

```

برای نوردهی از تابع `setLights()` استفاده کردیم و برای مهیج شدن بازی یک قسمت `harder` اضافه کردیم که سرعت توپ را افزایش میدهد.