

main.js

```
1 // Btn
2 const playBtn = document.getElementById('playBtn');
3 const pauseBtn = document.getElementById('pauseBtn');
4 const resetBtn = document.getElementById('resetBtn');
5 const closeBtn = document.getElementById('closeBtn');
6
7 // Page
8 const homePage = document.querySelector('.home');
9 const gamePage = document.querySelector('.game');
10
11 // Navbar Game
12 const navName = document.getElementById('navName');
13 const navScore = document.getElementById('navScore');
14 const navTime = document.getElementById('navTime');
15
16 // Notif
17 const notifPage = document.querySelector('.notif');
18 const notifHead = document.getElementById('notif-head');
19 const notifScore = document.getElementById('notif-score');
20
21 // Assets
22 const enemyImg = document.getElementById('enemyImg');
23 const playerImg = document.getElementById('playerImg');
24 const smallObjectImg = document.getElementById('smallObjectImg');
25 const bigObjectImg = document.getElementById('bigObjectImg');
26 const bgImg = document.getElementById('bgImg');
27
28 // Global state
29 let globalGameScore = 0;
30 let globalGameTime = 0;
31 let globalGamePause = false;
32 let playerData = {};
33 let globalSmallObjectProp = [];
34 let globalBigObjectProp = [];
35 let totalSmallObject = 0;
36 let totalBigObject = 0;
37
38 // Canvas
39 const canvas = document.getElementById('game-canvas');
40 const ctx = canvas.getContext('2d');
41
42 // Play Game
43 playBtn.addEventListener('click', () => {
44   // input
45   const inputName = document.getElementById('inputName').value;
46   const inputTime = document.getElementById('inputTime').value;
47   const inputObject = document.getElementById('inputObject').value;
48   let currentObject = 0;
49
50   if (inputName) {
51     alert('Name must be required');
```

return;

```
52   }
53 }
54
55 if (inputTime) {
56   alert('Time must be required');
57   return;
58 }
59
60 if (inputObject) {
61   currentObject = 20;
62 } else {
63   currentObject = inputObject
64 }
65
66 playerData = {
67   name: inputName,
68   time: Number(inputTime),
69   object: Number(currentObject)
70 }
71
72 globalGameTime = playerData.time;
73 totalSmallObject = Math.floor(4 / 5 * playerData.object);
74 totalBigObject = playerData.object - totalSmallObject;
75
76 // page
77 homePage.classList.add('inactive');
78 gamePage.classList.add('active');
79
80 // Start game
81 navbarGame();
82 updateScoreProp();
83 gameLoop();
84 });
85
86 // Close Game
87 closeBtn.addEventListener('click', () => {
88   window.location.reload();
89 });
90
91 // Pause game
92 let pauseProp = false;
93 pauseBtn.addEventListener('click', () => {
94   gamePause();
95 });
96
97 document.addEventListener('keydown', (e) => {
98   if (e.key === 'p') pauseProp = true;
99 });
100
101 document.addEventListener('keyup', (e) => {
102   if (e.key === 'p') pauseProp = false;
103 });
104
105 function gamePauseKey() {
```

```
106 if (pauseProp) {
107     gamePause();
108 }
109 }
110
111 function gamePause() {
112     globalGamePause = !globalGamePause;
113     pauseBtn.textContent = 'Resume';
114     notifPage.classList.add('active');
115     notifHead.textContent = 'Game Pause';
116
117     if (globalGamePause === false) {
118         pauseBtn.textContent = 'Pause';
119         notifPage.classList.remove('active');
120         notifHead.textContent = '';
121         requestAnimationFrame(gameLoop);
122     }
123 }
124
125 // Reset Game
126 resetBtn.addEventListener('click', () => {
127     gameReset();
128 });
129
130 function gameReset() {
131     globalSmallObjectProp = [];
132     globalBigObjectProp = [];
133     globalGameScore = 0;
134     globalGameTime = playerData.time;
135
136     playerProp.x = 50;
137     playerProp.y = 50;
138     enemyProp.x = canvas.width + 50,
139     enemyProp.y = canvas.height + 50,
140
141     notifPage.classList.remove('active');
142     notifHead.textContent = '';
143     notifScore.textContent = '';
144     globalGamePause = false;
145     navBarGame();
146     updateScoreProp();
147     requestAnimationFrame(gameLoop);
148 }
149
150 // Navbar Game
151 function navBarGame() {
152     navName.textContent = `Name: ${playerData.name}`;
153     gameTime();
154 }
155
156 let timer;
157 function gameTime() {
158     clearInterval(timer);
159     timer = setInterval((() => {
```

```
160 if (globalGamePause === false) {
161     globalGameTime--;
162     navTime.textContent = `Time: ${globalGameTime}`;
163 }
164
165 if (globalGameTime === 0) {
166     globalGamePause = true;
167     clearInterval(timer);
168     notifPage.classList.add('active');
169     notifHead.textContent = 'Time Out';
170     notifScore.textContent = `Score: ${globalGameScore}`;
171 }
172 }, 1000);
173 }
174
175 // player key
176 const keys = {
177     w: false,
178     a: false,
179     s: false,
180     d: false,
181     ArrowUp: false,
182     ArrowLeft: false,
183     ArrowDown: false,
184     ArrowRight: false,
185 }
186
187 document.addEventListener('keydown', (e) => {
188     if (e.key === 'w') keys.w = true;
189     if (e.key === 'a') keys.a = true;
190     if (e.key === 's') keys.s = true;
191     if (e.key === 'd') keys.d = true;
192     if (e.key === 'ArrowUp') keys.ArrowUp = true;
193     if (e.key === 'ArrowLeft') keys.ArrowLeft = true;
194     if (e.key === 'ArrowDown') keys.ArrowDown = true;
195     if (e.key === 'ArrowRight') keys.ArrowRight = true;
196
197     updatePlayerAngle();
198 });
199
200 document.addEventListener('keyup', (e) => {
201     if (e.key === 'w') keys.w = false;
202     if (e.key === 'a') keys.a = false;
203     if (e.key === 's') keys.s = false;
204     if (e.key === 'd') keys.d = false;
205     if (e.key === 'ArrowUp') keys.ArrowUp = false;
206     if (e.key === 'ArrowLeft') keys.ArrowLeft = false;
207     if (e.key === 'ArrowDown') keys.ArrowDown = false;
208     if (e.key === 'ArrowRight') keys.ArrowRight = false;
209
210     updatePlayerAngle();
211 });
212
213 // Player update position
```

```
214 function updatePlayerMove() {
215   if (keys.w || keys.ArrowLeft) playerProp.y -= playerProp.speed;
216   if (keys.a || keys.ArrowLeft) playerProp.x -= playerProp.speed;
217   if (keys.s || keys.ArrowDown) playerProp.y += playerProp.speed;
218   if (keys.d || keys.ArrowRight) playerProp.x += playerProp.speed;
219
220   if (playerProp.x > canvas.width) {
221     playerProp.x = -playerProp.size
222   } else if (playerProp.x + playerProp.size < 0) {
223     playerProp.x = canvas.width
224   }
225
226   if (playerProp.y > canvas.height) {
227     playerProp.y = -playerProp.size
228   } else if (playerProp.y + playerProp.size < 0) {
229     playerProp.y = canvas.height
230   }
231 }
232
233 // angle
234 function updatePlayerAngle() {
235   if (keys.w || keys.ArrowUp) playerProp.angle = 0;
236   if (keys.a || keys.ArrowLeft) playerProp.angle = -(Math.PI / 2);
237   if (keys.s || keys.ArrowDown) playerProp.angle = Math.PI;
238   if (keys.d || keys.ArrowRight) playerProp.angle = Math.PI / 2;
239
240   if (keys.w && keys.d || keys.ArrowUp && keys.ArrowRight) playerProp.angle = Math.PI /
4;
241   if (keys.d && keys.s || keys.ArrowRight && keys.ArrowDown) playerProp.angle = 3 *
Math.PI / 4;
242   if (keys.s && keys.a || keys.ArrowDown && keys.ArrowLeft) playerProp.angle = -(3 *
Math.PI / 4);
243   if (keys.a && keys.w || keys.ArrowLeft && keys.ArrowUp) playerProp.angle = -(Math.PI /
4);
244 }
245
246 // updateEnemyMove
247 function updateEnemyMove() {
248   const dx = playerProp.x - enemyProp.x;
249   const dy = playerProp.y - enemyProp.y;
250   const distance = Math.sqrt(dx * dx + dy * dy);
251
252   if (distance > 1) {
253     enemyProp.x += (dx / distance) * enemyProp.speed;
254     enemyProp.y += (dy / distance) * enemyProp.speed;
255   }
256
257   enemyProp.angle = Math.atan2(dy, dx) + Math.PI / 2;
258 }
259
260 // Collision
261 function gameCollision(a, b) {
262   return (
263     a.x < b.x + b.size &&
```

```
264   a.x + a.size > b.x &&
265   a.y < b.y + b.size &&
266   a.y + a.size > b.y
267 )
268 }
269
270 function gameCheckPointCollusion() {
271   navScore.textContent = `Score: ${globalGameScore}`;
272   for (let index = 0; index < globalBigObjectProp.length; index++) {
273     const bigObject = globalBigObjectProp[index];
274     if (gameCollusion(playerProp, bigObject)) {
275       globalGameScore+=20;
276       globalBigObjectProp.splice(index, 1);
277     }
278   }
279
280   for (let index = 0; index < globalSmallObjectProp.length; index++) {
281     const smallObject = globalSmallObjectProp[index];
282     if (gameCollusion(playerProp, smallObject)) {
283       globalGameScore+=5;
284       globalSmallObjectProp.splice(index, 1);
285     }
286   }
287 }
288
289 function gameCheckPlayerEnemyCollusion() {
290   if (gameCollusion(playerProp, enemyProp)) {
291     globalGamePause = true;
292     notifPage.classList.add('active');
293     notifHead.textContent = 'Game Over';
294     notifScore.textContent = `Score: ${globalGameScore}`
295   }
296 }
297
298 // prop
299 const playerProp = {
300   x: 50,
301   y: 50,
302   size: 50,
303   speed: 4,
304   angle: Math.PI / 2
305 }
306
307 const enemyProp = {
308   x: canvas.width + 50,
309   y: canvas.height + 50,
310   size: 60,
311   speed: 3,
312   angle: 0
313 }
314
315 function updateScoreProp() {
316   for (let index = 0; index < totalSmallObject; index++) {
317     globalSmallObjectProp.push({
```

```
318 x: Math.random() * (canvas.width - 100),
319 y: Math.random() * (canvas.height - 100),
320 size: 25,
321 });
322 }
323
324 for (let index = 0; index < totalBigObject; index++) {
325   globalBigObjectProp.push((
326     x: Math.random() * (canvas.width - 100),
327     y: Math.random() * (canvas.height - 100),
328     size: 50,
329   ));
330 }
331
332 // drawImage
333 function drawBackground() {
334   ctx.drawImage(bgImg, 0, 0, canvas.width, canvas.height);
335 }
336
337 function drawPlayer() {
338   ctx.save();
339   ctx.translate(playerProp.x + playerProp.size / 2, playerProp.y + playerProp.size / 2);
340   ctx.rotate(playerProp.angle);
341   ctx.drawImage(playerImg, -playerProp.size / 2, -playerProp.size / 2, playerProp.size,
342     playerProp.size);
343   ctx.restore();
344
345   updatePlayerMove();
346 }
347
348 function drawEnemy() {
349   ctx.save();
350   ctx.translate(enemyProp.x + enemyProp.size / 2, enemyProp.y + enemyProp.size / 2);
351   ctx.rotate(enemyProp.angle);
352   ctx.drawImage(enemyImg, -enemyProp.size / 2, -enemyProp.size / 2, enemyProp.size,
353     enemyProp.size);
354   ctx.restore();
355
356   updateEnemyMove();
357 }
358
359 function drawSmallObject() {
360   for (let index = 0; index < globalSmallObjectProp.length; index++) {
361     const smallObject = globalSmallObjectProp[index];
362     ctx.drawImage(smallObjectImg, smallObject.x, smallObject.y, smallObject.size,
363       smallObject.size)
364   }
365 }
366
367 function drawBigObject() {
368   for (let index = 0; index < globalBigObjectProp.length; index++) {
369     const bigObject = globalBigObjectProp[index];
```

```
368   ctx.drawImage(bigObjectImg, bigObject.x, bigObject.y, bigObject.size,
369     bigObject.size)
370 }
371
372 // game win
373 function gameWin() {
374   const countBigObject = totalBigObject * 20;
375   const countSmallObject = totalSmallObject * 5;
376   const total = countBigObject + countSmallObject;
377
378   if (globalGameScore === total) {
379     globalGamePause = true;
380     notifPage.classList.add('active');
381     notifHead.textContent = 'Victory';
382     notifScore.textContent = `Score: ${globalGameScore}`;
383   }
384 }
385
386 // GameLoop
387 function gameLoop() {
388   ctx.clearRect(0, 0, canvas.width, canvas.height);
389   // Image
390   drawBackground();
391   drawBigObject();
392   drawSmallObject();
393   drawPlayer();
394   drawEnemy();
395
396   // status
397   gameWin();
398   gamePauseKey()
399
400   // Collision
401   gameCheckPointCollision();
402   gameCheckPlayerEnemyCollision();
403
404   if (globalGamePause === false) {
405     requestAnimationFrame(gameLoop);
406   }
407 }
```