

```
game.js
1 window.onload = init;
2 // Global Vars
3 var game = new Game();
4 var player = new Player();
5 var mapConfig = {
6   x: 4,
7   y: 3,
8   z: 2
9 }
10 var gameConfig = {
11   maxPlayerInventory: 3
12 }
13 var rooms = ["hall", "dungeon", "waterfall", "kitchen", "dining area", "primitive bathroom", "cavern"];
14 var objects = ["shiny sword", "rusty dagger", "mysterious potion", "broken crystal", "flashlight", "key", "battery-operated lantern", "mom's spaghetti", "banana slug"];
15 var features = ["trap door", "window", "refrigerator", "portal", "rickety staircase", "hole", "door"];
16
17
18
19 function init() {
20   displayOutput("You are in a room.");
21   displayOutput(" ");
22   game.init();
23   player.init();
24   animate();
25 }
26
27
28
29 function animate() {
30   requestAnimationFrame(animate);
31 }
32
33
34
35 // Process Input
36 function checkInput() {
37   var input = document.getElementById("bar").value;
38   displayOutput(">>>" + input);
39   if ( input.toUpperCase() === input ) {
40     displayOutput("DON'T YOU DARE RAISE YOUR VOICE AT ME!");
41   }
42   input = input.toLowerCase();
43
44   doCommand(input);
45   displayOutput(" ");
46   document.getElementById("bar").value = "";
47   window.scrollTo(0,document.body.scrollHeight);
48 }
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
game.js
45 displayOutput(" ");
46 document.getElementById("bar").value = "";
47 window.scrollTo(0,document.body.scrollHeight);
48 }
49
50
51
52 //Game
53 function Game() {
54   this.init = function() {
55     this.map = [];
56     for (var i = 0; i < mapConfig.x; i++) { //across
57       this.map.push([]);
58       for (var j = 0; j < mapConfig.y; j++) { //down
59         this.map[i].push([]);
60         for(var k = 0; k < mapConfig.z; k++) { //level
61           var room = new Room();
62           room.init(rooms[generateRandomInt(rooms.length)], new JSVector(i, j, k));
63           this.map[i][j].push(room);
64         }
65       }
66     }
67   }
68 }
69
70
71
72 //Room
73 function Room() {
74   this.init = function(name, vector) {
75     this.name = name;
76     this.location = vector;
77     this.contents = [];
78     this.features = [];
79     this.validMoves = [];
80     this.scene = "";
81
82     this.contents.push(objects[generateRandomInt(objects.length)]);
83     this.features.push(features[generateRandomInt(features.length)]);
84
85     if ( vector.z < mapConfig.z - 1 ) {
86       this.validMoves.push("d"); // down
87     }
88     if ( vector.z > 0 ) {
89       this.validMoves.push("u"); // up
90     }
91     if ( vector.y < mapConfig.y - 1 ) {
92
93
94
95
96
97
98
99
100
```

```

game.js
83     this.validMoves.push("u"); // up
84 }
85 if ( vector.y < mapConfig.y - 1 ) {
86     this.validMoves.push("s"); // south
87 }
88 if ( vector.y > 0 ) {
89     this.validMoves.push("n"); // north
90 }
91 if ( vector.x < mapConfig.x - 1 ) {
92     this.validMoves.push("e"); // east
93 }
94 if ( vector.x > 0 ) {
95     this.validMoves.push("w"); // west
96 }
97 }
98
99 this.look = function(){
100     let stuff = "";
101     this.scene = "You are in a " + this.name + ", it features a " + this.features[0] + ", and contains: ";
102     displayOutput(this.scene);
103     this.contents.forEach ( function(element) {
104         displayOutput(element);
105     } );
106 }
107 }
108 }
109
110 //Player
111 function Player() {
112     this.init = function() {
113         this.position = new JSVector(0, 0, 0);
114         this.inventory = [];
115     }
116     this.update = function(x, y, z) {
117         this.position = new JSVector(x, y, z);
118     }
119 }
120
121 // Compute Input
122 function doCommand(input) {
123     if ( input.includes("inventory") || input.includes("stuff") ) {
124         player.inventory.forEach(function(element){
125             displayOutput(element);
126         });
127     }
128 }

```

```

game.js 91:40
129
130 // Compute Input
131 function doCommand(input) {
132     if ( input.includes("inventory") || input.includes("stuff") ) {
133         player.inventory.forEach(function(element){
134             displayOutput(element);
135         });
136     }
137 }

```

```

game.js
133 player.inventory.forEach(function(element){
134     displayOutput(element);
135 });
136 if ( !player.inventory.length ) {
137     displayOutput("You do not have any material possessions.");
138 }
139 } else if ( input.includes("east") || (input === "e") ) {
140     if (game.map[player.position.x][player.position.y][player.position.z].validMoves.includes("e")) {
141         player.update(player.position.x + 1, player.position.y, player.position.z);
142         game.map[player.position.x][player.position.y][player.position.z].look();
143     } else {
144         displayOutput("You imbecile");
145         displayOutput("Are you trying to get a concussion? Honestly.");
146         displayOutput("Last I checked, you are not able to go through walls.");
147     }
148 } else if ( input.includes("west") || (input === "w") ) {
149     if (game.map[player.position.x][player.position.y][player.position.z].validMoves.includes("w")) {
150         player.update(player.position.x - 1, player.position.y, player.position.z);
151         game.map[player.position.x][player.position.y][player.position.z].look();
152     } else {
153         displayOutput("You trilobite");
154         displayOutput("Are you trying to get a concussion? Honestly.");
155         displayOutput("Last I checked, you are not able to go through walls.");
156     }
157 } else if ( input.includes("north") || (input === "n") ) {
158     if (game.map[player.position.x][player.position.y][player.position.z].validMoves.includes("n")) {
159         player.update(player.position.x, player.position.y - 1, player.position.z);
160         game.map[player.position.x][player.position.y][player.position.z].look();
161     } else {
162         displayOutput("YOU FOOL");
163         displayOutput("Are you trying to get a concussion? Honestly.");
164         displayOutput("Last I checked, you are not able to go through walls.");
165     }
166 } else if ( input.includes("south") || (input === "s") ) {
167     if (game.map[player.position.x][player.position.y][player.position.z].validMoves.includes("s")) {
168         player.update(player.position.x, player.position.y + 1, player.position.z);
169         game.map[player.position.x][player.position.y][player.position.z].look();
170     } else {
171         displayOutput("Ya dumb meatball");
172         displayOutput("Are you trying to get a concussion? Honestly.");
173         displayOutput("Last I checked, you are not able to go through walls.");
174     }
175 } else if ( input.includes("up") ) {
176     if (game.map[player.position.x][player.position.y][player.position.z].validMoves.includes("u")) {
177         player.update(player.position.x, player.position.y, player.position.z - 1);
178         game.map[player.position.x][player.position.y][player.position.z].look();
179     } else {

```

```

game.js
177 player.update(player.position.x, player.position.y, player.position.z - 1);
178 game.map[player.position.x][player.position.y][player.position.z].look();
179 } else {
180   displayOutput("I smell a GNASHGAB (it's a real word--look it up) ");
181   displayOutput("Are you trying to get a concussion? Honestly.");
182   displayOutput("Last I checked, you are not able to go through the ceiling.");
183 }
184 } else if ( input.includes("down") ) {
185   if (game.map[player.position.x][player.position.y][player.position.z].validMoves.includes("d")) {
186     player.update(player.position.x, player.position.y, player.position.z + 1);
187     game.map[player.position.x][player.position.y][player.position.z].look();
188   } else {
189     displayOutput("Back in my day, noodles weren't THIS stupid");
190     displayOutput("Are you trying to get a concussion? Honestly.");
191     displayOutput("Last I checked, you are not able to go through the floor.");
192   }
193 } else if ( input.includes("look") ) {
194   game.map[player.position.x][player.position.y][player.position.z].look();
195 } else if ( input.includes("drop") ) {
196   let valid = false;
197   let has = false;
198   objects.forEach(function(element) {
199     if ( input.includes(element) ) { // valid
200       valid = true;
201       if ( player.inventory.includes(element) ) { // player has
202         has = true;
203         player.inventory.splice( player.inventory.indexOf(element), 1 ); // remove from inventory
204         game.map[player.position.x][player.position.y][player.position.z].contents.push(element); // add to room
205       }
206     }
207   });
208   if ( !valid ) {
209     displayOutput("I do not know what that is.");
210   } else if ( !has ) {
211     displayOutput("You do not have that.");
212   } else {
213     displayOutput("Dropped.");
214   }
215 } else if ( input.includes("keep") ) {
216   if ( player.inventory.length === gameConfig.maxPlayerInventory ) {
217     displayOutput("You are too weak to carry anything else.");
218   } else {
219     let valid = false;
220     let here = false;
221     objects.forEach(function(element) {
222       if ( input.includes(element) ) { // valid
223         valid = true;

```

```

game.js 179:13
LF UTF-8 JavaScript master 0 files

game.js
221 objects.forEach(function(element) {
222   if ( input.includes(element) ) { // valid
223     valid = true;
224     if ( game.map[player.position.x][player.position.y][player.position.z].contents.includes(element) ) { // in room
225       here = true;
226       game.map[player.position.x][player.position.y][player.position.z].contents.splice( game.map[player.position.x][player.position.y][player.position.z].contents.indexOf(element), 1 );
227       player.inventory.push(element); // add to inventory
228     }
229   }
230 });
231 if ( !valid ) {
232   displayOutput("I do not know what that is.");
233 } else if ( !here ) {
234   displayOutput("You do not see that.");
235 } else {
236   displayOutput("Keep.");
237 }
238 }
239 } else if ( (input === "restart") || (input === "again") || (input === "new game") || (input === "quit") ) {
240   displayOutput("I knew you would give up, you miserable shrimp. QUITTER");
241   displayOutput(" ");
242   displayOutput(" ");
243   displayOutput(" ");
244   displayOutput(" ");
245   init();
246 } else if ( input === "help" ) {
247   displayOutput("*disappointed sigh*");
248   displayOutput("Seriously? I hope you know life doesn't work like this.");
249   displayOutput("HELP:");
250   displayOutput("  north, south, east, west, up, down: directions");
251   displayOutput("  inventory: list of items in inventory");
252   displayOutput("  drop _: removes specified item from inventory");
253   displayOutput("  keep _: adds specified item to inventory");
254   displayOutput("  look: see what is around you");
255   displayOutput("  You can figure out the rest on your own.");
256 } else {
257   displayOutput("*elongated exasperated sigh*");
258   displayOutput("English, please.");
259 }
260 }
261
262 //Functionality
263 function generateRandomInt(max){
264   return Math.floor(Math.random() * max);
265 }
266
267
game.js 223:24
LF UTF-8 JavaScript master 0 files

```

```

game.js
243     displayOutput(" ");
244     displayOutput(" ");
245     init();
246 } else if ( input === "help" ) {
247     displayOutput("*disappointed sigh*");
248     displayOutput("Seriously? I hope you know Life doesn't work like this.");
249     displayOutput("HELP:");
250     displayOutput("    north, south, east, west, up, down: directions");
251     displayOutput("    inventory: list of items in inventory");
252     displayOutput("    drop _: removes specified item from inventory");
253     displayOutput("    keep _: adds specified item to inventory");
254     displayOutput("    look: see what is around you");
255     displayOutput("    You can figure out the rest on your own.");
256 } else {
257     displayOutput("*elongated exasperated sigh*");
258     displayOutput("English, please.");
259 }
260 }
261
262
263 //Functionality
264 function generateRandomInt(max){
265     return Math.floor(Math.random() * max);
266 }
267
268 function JSVector(x, y, z){
269     this.x = x;
270     this.y = y;
271     this.z = z;
272 }
273
274 function displayOutput(output){
275     document.getElementById("story").appendChild(document.createTextNode(output));
276     document.getElementById("story").appendChild(document.createElement("br"));
277 }
278
279 window.addEventListener("keypress", function(event){
280     switch(event.keyCode){
281         case 13:
282             checkInput();
283             break;
284         default:
285             return;
286     }
287     event.preventDefault();
288 }, true);
289

```