

```

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   specialIngredient: "banana slug"
13 }
14 var rooms = ["hall", "dungeon", "waterfall", "kitchen", "dining area", "primitive bathroom", "cavern", "dingy library", "musty cellar", "empty vault", "putrid crypt"];
15 var objects = ["shiny sword", "rusty dagger", "mysterious potion", "broken crystal", "flashlight", "key", "battery-operated lantern", "mom's spaghetti", "banana slug"];
16 var features = ["trap door", "window to nowhere", "refrigerator", "portal", "rickety staircase", "hole", "squeaky door"];
17
18
19
20 function init() {
21   displayOutput("You are in a room.");
22   displayOutput(" ");
23   game.init();
24   player.init();
25   animate();
26 }
27
28
29
30 function animate() {
31   requestAnimationFrame(animate);
32 }
33
34
35
36 // Process Input
37 function checkInput() {
38   var input = document.getElementById("bar").value;
39   displayOutput(">>>" + input);
40   if ( input.toUpperCase() === input ) {
41     displayOutput("DON'T YOU DARE RAISE YOUR VOICE AT ME");
42   }
43   input = input.toLowerCase();
44   doCommand(input);
45   displayOutput(" ");
46   document.getElementById("bar").value = "";
47   window.scrollTo(0,document.body.scrollHeight);
48   if ( !player.win ) {
49     // be normal
50   } else if ( player.win && player.inventory.includes(gameConfig.specialIngredient) ) {
51     displayOutput("THIS CRETIN FINALLY MADE IT");
52     displayOutput("I NEVER THOUGHT THIS DAY WOULD COME");
53     displayOutput("HONEY, COME LOOK");
54     displayOutput("Aw, won't you play again, sweetheart?");
55     displayOutput(" ");
56     displayOutput(" ");
57     displayOutput(" ");
58     displayOutput(" ");
59     document.getElementById("bar").value = "";
60     window.scrollTo(0,document.body.scrollHeight);
61     init();
62   } else if ( player.win && !player.inventory.includes("key") ) {
63     displayOutput("You came so close");
64     displayOutput("But alas...");
65     displayOutput("...you do not have the secret ingredient");
66     displayOutput("Keep trying I suppose");
67     displayOutput("*whispers under breath*");
68     displayOutput("But there's really no hope");
69     displayOutput(" ");
70     document.getElementById("bar").value = "";
71     window.scrollTo(0,document.body.scrollHeight);
72     player.win = false;
73   }
74 }
75
76
77
78 //Game
79 function Game() {
80   this.init = function() {
81     this.map = [];
82     this.escape = new JSVector( generateRandomInt(mapConfig.x), generateRandomInt(mapConfig.y), generateRandomInt(mapConfig.z) );
83     for (var i = 0; i < mapConfig.x; i++) { //across
84       this.map.push([]);
85       for (var j = 0; j < mapConfig.y; j++) { //down
86         this.map[i].push([]);
87         for(var k = 0; k < mapConfig.z; k++) { //level
88           var room = new Room();
89           room.init(rooms[generateRandomInt(rooms.length)], new JSVector(i, j, k));
90           this.map[i][j].push(room);
91         }
92       }
93     }
94   }
95 }

```

```

95 }
96
97
98
99 //Room
100 function Room() {
101     this.init = function(name, vector) {
102         this.name = name;
103         this.location = vector;
104         this.contents = [];
105         this.features = [];
106         this.validMoves = [];
107         this.scene = "";
108
109         this.contents.push(objects[generateRandomInt(objects.length)]);
110         this.features.push(features[generateRandomInt(features.length)]);
111
112         if ( vector.z < mapConfig.z - 1 ) {
113             this.validMoves.push("d"); // down
114         }
115         if ( vector.z > 0 ) {
116             this.validMoves.push("u"); // up
117         }
118         if ( vector.y < mapConfig.y - 1 ) {
119             this.validMoves.push("s"); // south
120         }
121         if ( vector.y > 0 ) {
122             this.validMoves.push("n"); // north
123         }
124         if ( vector.x < mapConfig.x - 1 ) {
125             this.validMoves.push("e"); // east
126         }
127         if ( vector.x > 0 ) {
128             this.validMoves.push("w"); // west
129         }
130     }
131
132     this.look = function(){
133         let stuff = "";
134         this.scene = "You are in a " + this.name + ", it features a " + this.features[0] + ", and contains: ";
135         displayOutput(this.scene);
136         if ( this.contents.length === 0 ) {
137             displayOutput("nothing.")
138         } else
139             this.contents.forEach ( function(element) {
140                 displayOutput(element);
141             } );
142     }
143 }
144
145
146
147 //Player
148 function Player() {
149     this.init = function() {
150         this.position = new JSVector(0, 0, 0);
151         this.inventory = [];
152         this.win = false;
153     }
154     this.update = function(x, y, z) {
155         this.position = new JSVector(x, y, z);
156         if ( this.position.x === game.escape.x && this.position.y === game.escape.y && this.position.z === game.escape.z ) {
157             this.win = true;
158         }
159     }
160 }
161
162
163
164 // Compute Input
165 function doCommand(input) {
166     if ( input.includes("inventory") || input.includes("stuff") ) {
167         player.inventory.forEach(function(element){
168             displayOutput(element);
169         });
170         if ( !player.inventory.length ) {
171             displayOutput("You do not have any material possessions.");
172         }
173     } else if ( input.includes("east") || (input === "e") ) {
174         if (game.map[player.position.x][player.position.y][player.position.z].validMoves.includes("e")) {
175             player.update(player.position.x + 1, player.position.y, player.position.z);
176             game.map[player.position.x][player.position.y][player.position.z].look();
177         } else {
178             displayOutput("You imbecile!");
179             displayOutput("Are you trying to get a concussion? Honestly.");
180             displayOutput("Last I checked, you are not able to go through walls.");
181         }
182     } else if ( input.includes("west") || (input === "w") ) {
183         if (game.map[player.position.x][player.position.y][player.position.z].validMoves.includes("w")) {
184             player.update(player.position.x - 1, player.position.y, player.position.z);
185             game.map[player.position.x][player.position.y][player.position.z].look();
186         } else {
187             displayOutput("You trilobite!");
188             displayOutput("Are you trying to get a concussion? Honestly.");
189         }
190     }
191 }

```

```

189     displayOutput("Last I checked, you are not able to go through walls.");
190 }
191 } else if ( input.includes("north") || (input === "n") ) {
192     if (game.map[player.position.x][player.position.y][player.position.z].validMoves.includes("n")) {
193         player.update(player.position.x, player.position.y - 1, player.position.z);
194         game.map[player.position.x][player.position.y][player.position.z].look();
195     } else {
196         displayOutput("YOU FOOL!");
197         displayOutput("Are you trying to get a concussion? Honestly.");
198         displayOutput("Last I checked, you are not able to go through walls.");
199     }
200 } else if ( input.includes("south") || (input === "s") ) {
201     if (game.map[player.position.x][player.position.y][player.position.z].validMoves.includes("s")) {
202         player.update(player.position.x, player.position.y + 1, player.position.z);
203         game.map[player.position.x][player.position.y][player.position.z].look();
204     } else {
205         displayOutput("Ya dumb meatball!");
206         displayOutput("Are you trying to get a concussion? Honestly.");
207         displayOutput("Last I checked, you are not able to go through walls.");
208     }
209 } else if ( input.includes("up") || (input === "u") ) {
210     if (game.map[player.position.x][player.position.y][player.position.z].validMoves.includes("u")) {
211         player.update(player.position.x, player.position.y, player.position.z - 1);
212         game.map[player.position.x][player.position.y][player.position.z].look();
213     } else {
214         displayOutput("I smell a QNASHGAB (it's a real word--look it up) ");
215         displayOutput("Are you trying to get a concussion? Honestly.");
216         displayOutput("Last I checked, you are not able to go through the ceiling.");
217     }
218 } else if ( input.includes("down") || (input === "d") ) {
219     if (game.map[player.position.x][player.position.y][player.position.z].validMoves.includes("d")) {
220         player.update(player.position.x, player.position.y, player.position.z + 1);
221         game.map[player.position.x][player.position.y][player.position.z].look();
222     } else {
223         displayOutput("Back in my day, noodles weren't THIS stupid!");
224         displayOutput("Are you trying to get a concussion? Honestly.");
225         displayOutput("Last I checked, you are not able to go through the floor.");
226     }
227 } else if ( input.includes("look") ) {
228     game.map[player.position.x][player.position.y][player.position.z].look();
229 } else if ( input.includes("drop") ) {
230     let valid = false;
231     let has = false;
232     objects.forEach(function(element) {
233         if ( input.includes(element) ) { // valid
234             valid = true;
235             if ( player.inventory.includes(element) ) { // player has
236                 has = true;
237                 player.inventory.splice( player.inventory.indexOf(element), 1 ); // remove from inventory
238                 game.map[player.position.x][player.position.y][player.position.z].contents.push(element); // add to room
239             }
240         }
241     });
242     if ( !valid ) {
243         displayOutput("I do not know what that is.");
244     } else if ( !has ) {
245         displayOutput("You do not have that.");
246     } else {
247         displayOutput("Dropped.");
248     }
249 } else if ( input.includes("keep") ) {
250     if ( player.inventory.length === gameConfig.maxPlayerInventory ) {
251         displayOutput("You are too weak to carry anything else.");
252     } else {
253         let valid = false;
254         let here = false;
255         objects.forEach(function(element) {
256             if ( input.includes(element) ) { // valid
257                 valid = true;
258                 if ( game.map[player.position.x][player.position.y][player.position.z].contents.includes(element) ) { // in room
259                     here = true;
260                     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 );
261                     player.inventory.push(element); // add to inventory
262                 }
263             }
264         });
265         if ( !valid ) {
266             displayOutput("I do not know what that is.");
267         } else if ( !here ) {
268             displayOutput("You do not see that.");
269         } else {
270             displayOutput("Kept.");
271         }
272     }
273 } else if ( (input === "restart") || (input === "again") || (input === "new game") || (input === "quit") ) {
274     displayOutput("I knew you would give up, you miserable shrimp. QUITTER!");
275     displayOutput(" ");
276     displayOutput(" ");
277     displayOutput(" ");
278     displayOutput(" ");
279     init();
280 } else if ( input === "help" ) {
281     displayOutput("#disappointed sigh#");
282     displayOutput("Seriously? I hope you know life doesn't work like this.");

```

```

277     displayOutput(" ");
278     displayOutput(" ");
279     init();
280 } else if ( input === "help" ) {
281     displayOutput("disappointed sigh");
282     displayOutput("Seriously? I hope you know life doesn't work like this.");
283     displayOutput("HELP:");
284     displayOutput("north, south, east, west, up, down: directions");
285     displayOutput("inventory: list of items in inventory");
286     displayOutput("drop: removes specified item from inventory");
287     displayOutput("keep: adds specified item to inventory");
288     displayOutput("look: see what is around you");
289     displayOutput("You can figure out the rest on your own.");
290 } else {
291     displayOutput("elongated exasperated sigh");
292     displayOutput("English, please.");
293 }
294 }
295
296
297 //Functionality
298 function generateRandomInt(max){
299     return Math.floor(Math.random() * max);
300 }
301
302 function JSVector(x, y, z){
303     this.x = x;
304     this.y = y;
305     this.z = z;
306 }
307
308 function displayOutput(output){
309     document.getElementById("story").appendChild(document.createTextNode(output));
310     document.getElementById("story").appendChild(document.createElement("br"));
311 }
312
313 window.addEventListener("keypress", function(event){
314     switch(event.keyCode){
315         case 13:
316             checkInput();
317             break;
318         default:
319             return;
320     }
321     event.preventDefault();
322 }, true);
323

```