

```

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);

```

game.js 1:22 LF UTF-8 JavaScript master 0 files

```

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 ) {

```

game.js 47:49 LF UTF-8 JavaScript master 0 files

```
game.js
89     this.validMoves.push("u"); // up
90 }
91 if ( vector.y < mapConfig.y - 1 ) {
92     this.validMoves.push("s"); // south
93 }
94 if ( vector.y > 0 ) {
95     this.validMoves.push("n"); // north
96 }
97 if ( vector.x < mapConfig.x - 1 ) {
98     this.validMoves.push("e"); // east
99 }
100 if ( vector.x > 0 ) {
101     this.validMoves.push("w"); // west
102 }
103 }
104
105 this.look = function(){
106     let stuff = "";
107     this.scene = "You are in a " + this.name + ", it features a " + this.features[0] + ", and contains: ";
108     displayOutput(this.scene);
109     this.contents.forEach ( function(element) {
110         displayOutput(element);
111     } );
112 }
113 }
114
115
116
117 //Player
118 function Player() {
119     this.init = function() {
120         this.position = new JSVector(0, 0, 0);
121         this.inventory = [];
122     }
123     this.update = function(x, y, z) {
124         this.position = new JSVector(x, y, z);
125     }
126 }
127
128
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 }
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

```
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 {
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

```
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
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("Kept.");
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
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

```