

```
1  /**
2   * File Name: R3.js
3   * Live Demo: http://www.adventurers-of-renown.com/GAMEAPP/index.html
4   * Source Code: http://makingbrowsergames.com/book/
5   * Description: File for controlling and displaying room scenes; managing
6   * global variables throughout game state.
7   * Author: Stephen Gose
8   * Version: 0.0.0.8
9   * Author URL: http://www.stephen-gose.com/
10  * Support: support@pbmcube.com
11  *
12  * Copyright © \u00A9 1974-2017 Stephen Gose LLC. All rights reserved.
13  *
14  * Do not sell! Do not distribute!
15  * This is a licensed permission file. Please refer to Terms of Use
16  * and End Users License Agreement (EULA).
17  * Search for [ /**TODO** ] to tailor this file for your own use and
18  * will void any support agreement.
19  *
20  * Redistribution of part or whole of this file and
21  * the accompanying files is strictly prohibited.
22  */
23  "use strict";
24  window.GAMEAPP.state.R3 = {
25    preload: function(){
26      console.log("loading Room 3 state");
27      GAMEAPP.CrntRoom = 3;
28      this.game.load.image('R3', 'assets/images/staticRooms/R3.jpg');
29      this.game.load.atlas('doorButton', 'assets/images/doors/Doors.png',
30        'assets/images/doors/Doors.json');
31      this.game.load.atlas('avatar', 'assets/icons/arraIcons.png',
32        'assets/icons/arraIcons.json');
33      this.game.load.atlas('tres', 'assets/icons/arraIcons.png',
34        'assets/icons/arraIcons.json');
35      this.load.spritesheet('button',
36        'assets/spriteSheets/mmog-sprites-silver.png', 129, 30);
37      this.load.atlas('inventoryButton',
38        'assets/spriteSheets/mmog-sprites-silver.png',
39        'assets/spriteSheets/mmog-sprites.json');
40      this.load.atlas('exitButton',
41        'assets/spriteSheets/mmog-sprites-silver.png',
42        'assets/spriteSheets/mmog-sprites.json');
43      this.load.atlas('fightButton',
44        'assets/spriteSheets/mmog-sprites-silver.png',
45        'assets/spriteSheets/mmog-sprites.json');
46      this.load.atlas('talkButton',
47        'assets/spriteSheets/mmog-sprites-silver.png',
48        'assets/spriteSheets/mmog-sprites.json');
49    },
50    create: function(){
51      console.log("starting Room 3 state");
52      this.game.add.image(0, 0, 'R3');
53
54      var rect = new Phaser.Rectangle(37, 105, 300, 200);
55      //Set a neutral background color
56      this.game.stage.backgroundColor = "#000";
57    }
58  }
```

```
46 //Set game to ARCADE physics systemLanguage
47 this.game.physics.startSystem(Phaser.Physics.ARCADE);
48 this.game.renderer.renderSession.roundPixels = true;
49 this.game.world.enableBody = true;
50
51 //Create Room
52 this.Room = this.game.add.group();
53 this.Room.enableBody = true;
54 this.game.physics.enable(this.Room, Phaser.Physics.ARCADE);
55
56 //gap filled with this
57 this.interior1Wall = this.game.add.sprite(30,177,box({length:40,
width:16,color:'#AAA'}));
58 this.interior1Wall.enableBody = true;
59 this.interior1Wall.body.immovable = true;
60 this.interior1Wall.visible = false;
61 this.Room.add(this.interior1Wall);
62
63 this.NorthWall = this.game.add.sprite(87,270,box({length:270,width
:16,color:'#999'}));
64 this.NorthWall.enableBody = true;
65 this.NorthWall.body.immovable = true;
66 this.NorthWall.visible = false;
67 this.Room.add(this.NorthWall);
68 this.NorthWall.alignTo(rect, Phaser.TOP_CENTER);
69
70 this.SouthWall = this.game.add.sprite(87,270,box({length:270,width
:16,color:'#999'}));
71 this.SouthWall.body.immovable = true;
72 this.SouthWall.visible = false;
73 this.Room.add(this.SouthWall);
74 this.SouthWall.alignTo(rect, Phaser.BOTTOM_CENTER);
75
76 this.WestWall = this.game.add.sprite(200,16,box({length:16,width:
210,color:'#999'}));
77 this.WestWall.body.immovable = true;
78 this.WestWall.visible = false;
79 this.Room.add(this.WestWall);
80 this.WestWall.alignTo(rect, Phaser.LEFT_CENTER);
81
82 this.EastWall = this.game.add.sprite(90,270,box({length:16,width:
190,color:'#999'}));
83 this.EastWall.body.immovable = true;
84 this.EastWall.visible = false;
85 this.Room.add(this.EastWall);
86 this.EastWall.alignTo(rect, Phaser.RIGHT_CENTER);
87
88 //Room Doors
89 this.doorN = this.game.add.sprite(80,177,'doorButton');
//perfectly aligned over placeholder
90 GAMEAPP.doorN = this.doorN;
91 this.doorN.name = "North";
92 this.doorN.frameName = 'horizWood.bmp';
93 this.doorN.enableBody = true;
94 this.doorN.body.immovable = true;
95 this.doorN.inputEnabled = true;
96 this.game.physics.enable(this.doorN, Phaser.Physics.ARCADE);
97 this.doorN.events.onInputDown.add(this.newRoom, this);
98
```

```

99         this.doorE = this.game.add.sprite(318,224,'doorButton');
           //perfectly aligned over placeholder
100        GAMEAPP.doorE = this.doorE;
101        this.doorE.name = "East";
102        this.doorE.frameName = 'vertWood.bmp';
103        this.doorE.enableBody = true;
104        this.doorE.body.immovable = true;
105        this.doorE.inputEnabled = true;
106        this.game.physics.enable(this.doorE, Phaser.Physics.ARCADE);
107        this.doorE.events.onInputDown.add(this.newRoom, this);
108
109        this.doorW = this.game.add.sprite(168,112,'doorButton');
           //perfectly aligned over placeholder
110        GAMEAPP.doorW = this.doorW;
111        this.doorW.name = "West";
112        this.doorW.frameName = 'vertWood.bmp';
113        this.doorW.enableBody = true;
114        this.doorW.body.immovable = true;
115        this.doorW.inputEnabled = true;
116        this.game.physics.enable(this.doorW, Phaser.Physics.ARCADE);
117        this.doorW.events.onInputDown.add(this.newRoom, this);
118
119        //create treasure if needed; check for treasure in this room
120        if(MT[GAMEAPP.CrntRoom].Tres > 0){
121            GAMEAPP.InfoText = "Score: "+CCP.Score+"\n Click on the Door
           to enter.\nThere treasure here! See it?!!";
122            this.Treasure = this.game.add.sprite(200,200,'tres');
123            this.Treasure.frameName = 'treasure.bmp';
124            this.Treasure.body.collideWorldBounds = true;
125            this.Treasure.enableBody = true;
126            this.Treasure.body.immovable = true;
127            this.Treasure.inputEnabled = true;
128            this.game.physics.enable(this.Treasure, Phaser.Physics.ARCADE);
129            //clicking treasure for mobile interaction
130            this.Treasure.events.onInputDown.add(this.CTreasure, this);
131            this.Treasure.alignIn(rect, Phaser.CENTER);
132
133        }
134
135        this.inventorytxt = this.game.add.text(0, 0, "Inventory" , GAMEAPP
           .styleBTN); // "Attack" button text;
136        this.exittxt = this.game.add.text(0, 0, "Exit" , GAMEAPP.styleBTN
           ); // "Attack" button text;
137        this._toolTip = this.game.add.text(this.game.world.width-100, this
           .game.world.height-90, GAMEAPP.InfoText, GAMEAPP.styleTT);
138        this._toolTip.anchor.set(0.5);
139        GAMEAPP.RoomAlert = "";
140
141        // Inventory button
142        this.inventoryButton = this.game.add.button(65, 338, 'button',
           GAMEAPP._InventoryRoom, this, 2, 1, 0,1);
143        this.inventoryButton.anchor.set(0.5,0.5);
144        this.inventoryButton.scale.setTo(0.5,0.5);
145        this.inventoryButton.addChild(this.inventorytxt).anchor.set(0.5,
           0.5);
146
147        // Exit button
148        this.exitButton = this.game.add.button(65, 361, 'button', GAMEAPP.
           _ExitGame, this, 2, 1, 0,1);

```

```

149     this.exitButton.anchor.set(0.5,0.5);
150     this.exitButton.scale.setTo(0.5,0.5);
151     this.exitButton.addChild(this.exittxt).anchor.set(0.5,0.5);
152
153     //create an opponent; Check for monster in this room
154     if(MT[GAMEAPP.CrntRoom].Mnstr > 0){
155         GAMEAPP.RoomAlert = "Beware, a " + M[MT[GAMEAPP.CrntRoom].
156             Mnstr].Race + " is here!"
157         GAMEAPP.raTxt = this.game.add.text(this.game.world.width-268,
158             65, GAMEAPP.RoomAlert, GAMEAPP.styleRA);
159         GAMEAPP.raTxt.anchor.set(0);
160         this.enemy = this.game.add.sprite(MT[GAMEAPP.CrntRoom].mxPos,
161             MT[GAMEAPP.CrntRoom].myPos, 'avatar');
162         this.enemy.frameName = String(M[MT[GAMEAPP.CrntRoom].Mnstr].
163             icon+'.bmp');
164         this.enemy.body.collideWorldBounds = true;
165         this.game.physics.enable(this.enemy, Phaser.Physics.ARCADE);
166
167         this.enemy.alignIn(rect, Phaser.RIGHT_CENTER);
168
169         GAMEAPP._PUpdate();
170         //clicking fight for mobile interaction
171         // Fight button
172         this.fighttxt = this.game.add.text(0, 0, "Fight" , GAMEAPP.
173             styleBTN);
174         this.fightButton = this.game.add.button(152, 338, 'button',
175             GAMEAPP._CombatCheck, this, 2, 1, 0,1);
176         this.fightButton.anchor.set(0.5,0.5);
177         this.fightButton.scale.setTo(0.5,0.5);
178         this.fightButton.addChild(this.fighttxt).anchor.set(0.5,0.5);
179
180         // Talk button
181         var talktxt = this.game.add.text(0, 0, "Talk" , GAMEAPP.
182             styleBTN);
183         var talkButton = this.game.add.button(152, 361, 'button', this
184             ._talkToMonster, this, 2, 1, 0,1);
185         talkButton.anchor.set(0.5,0.5);
186         talkButton.scale.setTo(0.5,0.5);
187         talkButton.addChild(talktxt).anchor.set(0.5,0.5);
188     }else{
189         GAMEAPP.InfoText = " Looking for something to kill?\n Try the
190             next room.";
191     }
192
193     GAMEAPP._RmAdmin(GAMEAPP.CrntRoom);
194
195     //character placement -
196     var entryBy = new Phaser.Point();
197     if(GAMEAPP.LastDoor=="North" || GAMEAPP.LastDoor=="North2"){
198         entryBy.copyFrom(this.doorS);
199         GAMEAPP.xPos = entryBy.x+15;
200         GAMEAPP.yPos = entryBy.y-35;
201     }
202
203     if(GAMEAPP.LastDoor=="East"){
204         entryBy.copyFrom(this.doorW);
205         GAMEAPP.xPos = entryBy.x+35;
206         GAMEAPP.yPos = entryBy.y+15;
207     }

```

```

199
200     if (GAMEAPP.LastDoor=="South") {
201         entryBy.copyFrom(this.doorN);
202         GAMEAPP.xPos = entryBy.x+15;
203         GAMEAPP.yPos = entryBy.y+35;
204     }
205
206     if (GAMEAPP.LastDoor=="West") {
207         entryBy.copyFrom(this.doorE);
208         GAMEAPP.xPos = entryBy.x-35;
209         GAMEAPP.yPos = entryBy.y+15;
210     }
211
212     this.player = this.game.add.sprite(GAMEAPP.xPos,GAMEAPP.yPos,
    'avatar');
213     this.player.frameName = 'avatar.bmp';
214     this.player.body.collideWorldBounds = true;
215     this.cursor = this.game.input.keyboard.createCursorKeys();
216
217
218 },
219
220 update: function() {
221     GAMEAPP._MoveChar(this.cursor,this.player);
222     this.game.physics.arcade.collide(this.player, this.Room);
223     if (this.game.physics.arcade.intersects(this.player, this.doorN)) {
224         this.newRoom(this.doorN);
225     }
226     if (this.game.physics.arcade.intersects(this.player, this.doorE)) {
227         this.newRoom(this.doorE);
228     }
229     if (this.game.physics.arcade.intersects(this.player, this.doorW)) {
230         this.newRoom(this.doorW);
231     }
232     //Check for monster in this room
233     if (MT[GAMEAPP.CrntRoom].Mnstr > 0) {
234         GAMEAPP._MoveMonster(this.cursor,this.enemy);
235         this.game.physics.arcade.collide(this.enemy, this.Room);
236         this.game.physics.arcade.overlap(this.player,this.enemy,
    GAMEAPP._CombatCheck,null,this);
237
238     }
239     //Check for treasure in this room
240     if (MT[GAMEAPP.CrntRoom].Tres > 0) {
241
242         this.game.physics.arcade.collide(this.player,this.Treasure,
    this.CTreasure,null,this);
243
244     }
245     this._toolTip.setText(GAMEAPP.InfoText);
246 },
247 //
248 // =====
249
250 //
251 // =====
252 //Main ARRA Door click handler
253 newRoom: function(door){
254     GAMEAPP.LastRoom = GAMEAPP.CrntRoom;

```

```
255     GAMEAPP.LastDoor = door.name;
256     console.log('Last Door Used: '+GAMEAPP.LastDoor);
257
258     if (door.name == "North"){
259
260         //debug
261         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
262         +door.name);
263         this.camera.fade(0x000000, 200, false);
264         this.time.events.add(200, function() {
265             this.game.state.start('R'+MT[GAMEAPP.CrntRoom].North);
266             //ARRA Main Entrance (hard coded) rv_3 through rv_8
267         }, this);
268     }
269
270     if (door.name == "East"){
271
272         //debug
273         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
274         +door.name);
275         this.camera.fade(0x000000, 200, false);
276         this.time.events.add(200, function() {
277             this.game.state.start('R'+MT[GAMEAPP.CrntRoom].East);
278             //ARRA Main Entrance (hard coded) rv_3 through rv_8
279         }, this);
280     }
281
282     /**
283     if (door.name == "South" ){
284
285         //debug
286         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
287         +door.name);
288         this.camera.fade(0x000000, 200, false);
289         this.time.events.add(200, function() {
290
291             this.game.state.start('R'+MT[GAMEAPP.CrntRoom].South);
292             //ARRA Main Entrance (hard coded) rv_3 through rv_8
293         }, this);
294     }
295
296     */
297
298     if (door.name == "West"){
299
300         //debug
301         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
302         +door.name);
303         this.camera.fade(0x000000, 200, false);
304         this.time.events.add(200, function() {
305             this.game.state.start('R'+MT[GAMEAPP.CrntRoom].West);
306             //ARRA Main Entrance (hard coded) rv_3 through rv_8
307         }, this);
308     }
309
310 },
311 //
312 //=====
313 _talkToMonster: function(){
314     GAMEAPP.InfoText = "... mmm, not in the mood to negotiate?!";
315 },
316 //
```

```
305 //=====
306 CTreasure: function(){
307     CCP.HGold += MT[GAMEAPP.CrntRoom].Tres;
308     this.Treasure.kill();
309     GAMEAPP.InfoText = "You found $" + MT[GAMEAPP.CrntRoom].Tres + " in
the chest.";
310     MT[GAMEAPP.CrntRoom].Tres = 0;
311     this._toolTip.setText("You found $" + MT[GAMEAPP.CrntRoom].Tres + "
in the chest.");
312 }
313 };
314 //
315 // =====
316 //create a box Image (pseudo graphics) for the HTML5 canvas.
317 var box = function(options) {
318     var bxImg = GAMEAPP.game.add.bitmapData(options.length,options.
width);
319     bxImg.ctx.beginPath();
320     bxImg.ctx.rect(0,0,options.length,options.width);
321     bxImg.ctx.fillStyle = options.color;
322     bxImg.ctx.fill();
323     return bxImg;
324 };
325 //
326 // =====
327
328 /* End of file */
329 /* Location: ./js/state/R3.js */
330
331
```