

```
1  /**
2   * File Name: R8.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.R8 = {
25    preload: function(){
26      console.log("loading Room 8 state");
27      this.game.load.image('R8', 'assets/images/staticRooms/R8.jpg');
28      GAMEAPP.CrntRoom = 8;
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 8 state");
52      this.game.add.image(0, 0, 'R8');
53      var rect = new Phaser.Rectangle(37, 105, 300, 200);
54      //Set a neutral background color
55      this.game.stage.backgroundColor = "#000";
56      //Set game to ARCADE physics systemLanguage
```

```
46     this.game.physics.startSystem(Phaser.Physics.ARCADE);
47     this.game.renderer.renderSession.roundPixels = true;
48     this.game.world.enableBody = true;
49
50     //Create Room
51     this.Room = this.game.add.group();
52     this.Room.enableBody = true;
53     this.game.physics.enable(this.Room, Phaser.Physics.ARCADE);
54
55     this.NorthWall = this.game.add.sprite(87,270,box({length:270,width:
56     :16,color:'#999'}));
57     this.NorthWall.enableBody = true;
58     this.NorthWall.body.immovable = true;
59     this.NorthWall.visible = false;
60     this.Room.add(this.NorthWall);
61     this.NorthWall.alignTo(rect, Phaser.TOP_CENTER);
62
63     this.SouthWall = this.game.add.sprite(87,270,box({length:270,width:
64     :16,color:'#999'}));
65     this.SouthWall.body.immovable = true;
66     this.SouthWall.visible = false;
67     this.Room.add(this.SouthWall);
68     this.SouthWall.alignTo(rect, Phaser.BOTTOM_CENTER);
69
70     this.WestWall = this.game.add.sprite(88,110,box({length:16,width:
71     190,color:'#999'}));
72     this.WestWall.body.immovable = true;
73     this.WestWall.visible = false;
74     this.Room.add(this.WestWall);
75
76     this.EastWall = this.game.add.sprite(270,110,box({length:16,width:
77     190,color:'#999'}));
78     this.EastWall.body.immovable = true;
79     this.EastWall.visible = false;
80     this.Room.add(this.EastWall);
81
82     //Room Doors
83     this.doorN = this.game.add.sprite(152,99,'doorButton');
84     //perfectly aligned over placeholder
85     this.doorN.name = "North";
86     this.doorN.frameName = 'horizWood.bmp';
87     this.doorN.enableBody = true;
88     this.doorN.body.immovable = true;
89     this.doorN.inputEnabled = true;
90     this.game.physics.enable(this.doorN, Phaser.Physics.ARCADE);
91     this.doorN.events.onInputDown.add(this.newRoom, this);
92
93     this.doorS = this.game.add.sprite(157,287,'doorButton');
94     //perfectly aligned over placeholder
95     GAMEAPP.doorS = this.doorS;
96     this.doorS.name = "South";
97     this.doorS.frameName = 'horizWood.bmp';
98     this.doorS.enableBody = true;
99     this.doorS.body.immovable = true;
100    this.doorS.inputEnabled = true;
101    this.game.physics.enable(this.doorS, Phaser.Physics.ARCADE);
102    this.doorS.events.onInputDown.add(this.newRoom, this);
103
104    this.upStairs = this.game.add.sprite(260,150,'floors');
```

```
99      //perfectly aligned over placeholder
100     this.upStairs.name = "UpStairs";
101     this.upStairs.enableBody = true;
102     this.upStairs.body.immovable = true;
103     this.upStairs.inputEnabled = true;
104     this.game.physics.enable(this.upStairs, Phaser.Physics.ARCADE);
105     this.upStairs.events.onInputDown.add(this.newRoom, this);
106
107     this.downStairs = this.game.add.sprite(260,220,'floors');
108     //perfectly aligned over placeholder
109     this.downStairs.name = "DownStairs";
110     this.downStairs.enableBody = true;
111     this.downStairs.body.immovable = true;
112     this.downStairs.inputEnabled = true;
113     this.game.physics.enable(this.downStairs, Phaser.Physics.ARCADE);
114     this.downStairs.events.onInputDown.add(this.newRoom, this);
115
116     //create treasure if needed; check for treasure in this room
117     if(MT[GAMEAPP.CrntRoom].Tres > 0){
118         GAMEAPP.InfoText = "Score: "+CCP.Score+"\n Click on the Door
119         to enter.\nThere treasure here! See it?!";
120         this.Treasure = this.game.add.sprite(200,200,'tres');
121         this.Treasure.frameName = 'treasure.bmp';
122         this.Treasure.body.collideWorldBounds = true;
123         this.Treasure.enableBody = true;
124         this.Treasure.body.immovable = true;
125         this.Treasure.inputEnabled = true;
126         this.game.physics.enable(this.Treasure, Phaser.Physics.ARCADE);
127         //clicking treasure for mobile interaction
128         this.Treasure.events.onInputDown.add(this.CTreasure, this);
129         this.Treasure.alignIn(rect, Phaser.CENTER);
130     }
131
132     this.inventorytxt = this.game.add.text(0, 0, "Inventory" , GAMEAPP
133     .styleBTN); // "Attack" button text;
134     this.exittxt = this.game.add.text(0, 0, "Exit" , GAMEAPP.styleBTN
135     ); // "Attack" button text;
136     this._toolTip = this.game.add.text(this.game.world.width-100, this
137     .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',
143     GAMEAPP._InventoryRoom, this, 2, 1, 0,1);
144     this.inventoryButton.anchor.set(0.5,0.5);
145     this.inventoryButton.scale.setTo(0.5,0.5);
146     this.inventoryButton.addChild(this.inventorytxt).anchor.set(0.5,
147     0.5);
148
149     // Exit button
150     this.exitButton = this.game.add.button(65, 361, 'button', GAMEAPP.
151     _ExitGame, this, 2, 1, 0,1);
152     this.exitButton.anchor.set(0.5,0.5);
153     this.exitButton.scale.setTo(0.5,0.5);
154     this.exitButton.addChild(this.exittxt).anchor.set(0.5,0.5);
155
156     //create an opponent; Check for monster in this room
```

```

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

```

```

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

```

```
254
255 //
256 // =====
257 //Main ARRA Door click handler
258 newRoom: function(door){
259     GAMEAPP.LastRoom = GAMEAPP.CrntRoom;
260     GAMEAPP.LastDoor = door.name;
261     console.log('Last Door Used: '+GAMEAPP.LastDoor);
262
263     if (door.name == "North"){
264
265         //debug
266         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
267             +door.name);
268         this.camera.fade(0x000000, 200, false);
269         this.time.events.add(200, function() {
270             this.game.state.start('R'+MT[GAMEAPP.CrntRoom].North);
271             //ARRA Main Entrance (hard coded) rv_3 through rv_8
272         }, this);
273     }
274
275 /**
276     if (door.name == "East"){
277
278         //debug
279         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
280             +door.name);
281         this.camera.fade(0x000000, 200, false);
282         this.time.events.add(200, function() {
283             this.game.state.start('R'+MT[GAMEAPP.CrntRoom].East);
284             //ARRA Main Entrance (hard coded) rv_3 through rv_8
285         }, this);
286     }
287
288 */
289     if (door.name == "South" ){
290
291         //debug
292         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
293             +door.name);
294         this.camera.fade(0x000000, 200, false);
295         this.time.events.add(200, function() {
296             this.game.state.start('R'+MT[GAMEAPP.CrntRoom].South);
297             //ARRA Main Entrance (hard coded) rv_3 through rv_8
298         }, this);
299     }
300
301     if (door.name == "DownStairs"){
302
303         //debug
304         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
305             +door.name);
306         this.camera.fade(0x000000, 200, false);
307         this.time.events.add(200, function() {
308             this.game.state.start('R'+MT[GAMEAPP.CrntRoom].Dn);
309             //ARRA Main Entrance (hard coded) rv_3 through rv_8
310         }, this);
311     }
312 }
```

```

304         if (door.name == "UpStairs"){
305
306             //debug
307             console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
308                 +door.name);
309             this.camera.fade(0x000000, 200, false);
310             this.time.events.add(200, function() {
311                 this.game.state.start('R'+MT[GAMEAPP.CrntRoom].Up);
312                 //ARRA Main Entrance (hard coded) rv_3 through rv_8
313             }, this);
314         }
315     /**
316     if (door.name == "West"){
317
318         //debug
319         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked:
320             "+door.name);
321         this.camera.fade(0x000000, 200, false);
322         this.time.events.add(200, function() {
323
324             this.game.state.start('R'+MT[GAMEAPP.CrntRoom].West);
325             //ARRA Main Entrance (hard coded) rv_3 through rv_8
326         }, this);
327     }
328     */
329 },
330 //
331 //=====
332 _talkToMonster: function(){
333     GAMEAPP.InfoText = "... mmm, not in the mood to negotiate?!";
334 },
335 //
336 //=====
337 CTreasure: function(){
338     CCP.HGold += MT[GAMEAPP.CrntRoom].Tres;
339     this.Treasure.kill();
340     GAMEAPP.InfoText = "You found $" + MT[GAMEAPP.CrntRoom].Tres + " in
341     the chest.";
342     MT[GAMEAPP.CrntRoom].Tres = 0;
343     this._toolTip.setText("You found $" + MT[GAMEAPP.CrntRoom].Tres + "
344     in the chest.");
345 }
346 };
347 //
348 // =====
349 //create a box Image (pseudo graphics) for the HTML5 canvas.
350 var box = function(options) {
351     var bxImg = GAMEAPP.game.add.bitmapData(options.length, options.
352         width);
353     bxImg.ctx.beginPath();
354     bxImg.ctx.rect(0,0,options.length,options.width);
355     bxImg.ctx.fillStyle = options.color;
356     bxImg.ctx.fill();
357     return bxImg;
358 };
359 //
360 // =====
361
362 /* End of file */

```

---

```
355  /* Location: ./js/state/R8.js */  
356  
357
```