

```
1  /**
2   * File Name: R15.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.R15 = {
25    preload: function(){
26      console.log("loading Room 15 state");
27      this.game.load.image('R15', 'assets/images/staticRooms/R15.jpg');
28      GAMEAPP.CrntRoom = 15;
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.spritesheet('floors',
38        'assets/images/floor/downStairs.bmp', 40, 50);
39      this.load.atlas('inventoryButton',
40        'assets/spriteSheets/mmog-sprites-silver.png',
41        'assets/spriteSheets/mmog-sprites.json');
42      this.load.atlas('exitButton',
43        'assets/spriteSheets/mmog-sprites-silver.png',
44        'assets/spriteSheets/mmog-sprites.json');
45      this.load.atlas('fightButton',
46        'assets/spriteSheets/mmog-sprites-silver.png',
47        'assets/spriteSheets/mmog-sprites.json');
48      this.load.atlas('talkButton',
49        'assets/spriteSheets/mmog-sprites-silver.png',
50        'assets/spriteSheets/mmog-sprites.json');
51    },
52    create: function(){
53      console.log("starting Room 15 state");
54      this.game.add.image(0, 0, 'R15');
55      var rect = new Phaser.Rectangle(85, 160, 200, 80);
56      //Set a neutral background color
```

```
45     this.game.stage.backgroundColor = "#000";
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 - aligned to smaller rectangle
52     this.Room = this.game.add.group();
53     this.Room.enableBody = true;
54     this.game.physics.enable(this.Room, Phaser.Physics.ARCADE);
55
56     this.NorthWall = this.game.add.sprite(0,0,box({length:270,width:16
57     ,color:'#999'}));
58     this.NorthWall.enableBody = true;
59     this.NorthWall.body.immovable = true;
60     this.NorthWall.visible = false;
61     this.Room.add(this.NorthWall);
62     this.NorthWall.alignTo(rect, Phaser.TOP_CENTER);
63
64     this.SouthWall = this.game.add.sprite(0,0,box({length:270,width:16
65     ,color:'#999'}));
66     this.SouthWall.body.immovable = true;
67     this.SouthWall.visible = false;
68     this.Room.add(this.SouthWall);
69     this.SouthWall.alignTo(rect, Phaser.BOTTOM_CENTER);
70
71     this.WestWall = this.game.add.sprite(0,0,box({length:16,width:190,
72     color:'#999'}));
73     this.WestWall.body.immovable = true;
74     this.WestWall.visible = false;
75     this.Room.add(this.WestWall);
76     this.WestWall.alignTo(rect, Phaser.LEFT_CENTER);
77
78     this.EastWall = this.game.add.sprite(0,0,box({length:16,width:190,
79     color:'#999'}));
80     this.EastWall.body.immovable = true;
81     this.EastWall.visible = false;
82     this.Room.add(this.EastWall);
83     this.EastWall.alignTo(rect, Phaser.RIGHT_CENTER);
84
85     //Room Doors
86     this.doorN = this.game.add.sprite(90,153,'doorButton');
87     //perfectly aligned over placeholder
88     GAMEAPP.doorN = this.doorN;
89     this.doorN.name = "North";
90     this.doorN.frameName = 'horizWood.bmp';
91     this.doorN.enableBody = true;
92     this.doorN.body.immovable = true;
93     this.doorN.inputEnabled = true;
94     this.game.physics.enable(this.doorN, Phaser.Physics.ARCADE);
95     this.doorN.events.onInputDown.add(this.newRoom, this);
96
97     this.downStairs = this.game.add.sprite(268,190,'floors');
98     //perfectly aligned over placeholder
99     this.downStairs.name = "DownStairs";
100    this.downStairs.enableBody = true;
101    this.downStairs.body.immovable = true;
102    this.downStairs.inputEnabled = true;
103    this.game.physics.enable(this.downStairs, Phaser.Physics.ARCADE);
```

```
98         this.downStairs.events.onInputDown.add(this.newRoom, this);
99
100         //create treasure if needed; check for treasure in this room
101         if(MT[GAMEAPP.CrntRoom].Tres > 0){
102             GAMEAPP.InfoText = "Score: "+CCP.Score+"\n Click on the Door
to enter.\nThere treasure here! See it?!!";
103             this.Treasure = this.game.add.sprite(200,200,'tres');
104             this.Treasure.frameName = 'treasure.bmp';
105             this.Treasure.body.collideWorldBounds = true;
106             this.Treasure.enableBody = true;
107             this.Treasure.body.immovable = true;
108             this.Treasure.inputEnabled = true;
109             this.game.physics.enable(this.Treasure, Phaser.Physics.ARCADE);
110             //clicking treasure for mobile interaction
111             this.Treasure.events.onInputDown.add(this.CTreasure, this);
112             this.Treasure.alignIn(rect, Phaser.CENTER);
113
114         }
115
116         this.inventorytxt = this.game.add.text(0, 0, "Inventory" , GAMEAPP
.styleBTN); // "Attack" button text;
117         this.exittxt = this.game.add.text(0, 0, "Exit" , GAMEAPP.styleBTN
); // "Attack" button text;
118         this._toolTip = this.game.add.text(this.game.world.width-100, this
.game.world.height-90, GAMEAPP.InfoText, GAMEAPP.styleTT);
119         this._toolTip.anchor.set(0.5);
120         GAMEAPP.RoomAlert = "";
121
122         // Inventory button
123         this.inventoryButton = this.game.add.button(65, 338, 'button',
GAMEAPP._InventoryRoom, this, 2, 1, 0,1);
124         this.inventoryButton.anchor.set(0.5,0.5);
125         this.inventoryButton.scale.setTo(0.5,0.5);
126         this.inventoryButton.addChild(this.inventorytxt).anchor.set(0.5,
0.5);
127
128         // Exit button
129         this.exitButton = this.game.add.button(65, 361, 'button', GAMEAPP.
_ExitGame, this, 2, 1, 0,1);
130         this.exitButton.anchor.set(0.5,0.5);
131         this.exitButton.scale.setTo(0.5,0.5);
132         this.exitButton.addChild(this.exittxt).anchor.set(0.5,0.5);
133
134         //create an opponent; Check for monster in this room
135         if(MT[GAMEAPP.CrntRoom].Mnstr > 0){
136             GAMEAPP.RoomAlert = "Beware, a " + M[MT[GAMEAPP.CrntRoom].
Mnstr].Race + " is here!"
137             GAMEAPP.raTxt = this.game.add.text(this.game.world.width-268,
65, GAMEAPP.RoomAlert, GAMEAPP.styleRA);
138             GAMEAPP.raTxt.anchor.set(0);
139             this.enemy = this.game.add.sprite(MT[GAMEAPP.CrntRoom].mxPos,
MT[GAMEAPP.CrntRoom].myPos,'avatar');
140             this.enemy.frameName = String(M[MT[GAMEAPP.CrntRoom].Mnstr].
icon+'.bmp');
141             this.enemy.body.collideWorldBounds = true;
142             this.game.physics.enable(this.enemy, Phaser.Physics.ARCADE);
143
144             GAMEAPP._PUpdate();
145             //clicking fight for mobile interaction
```

```
146 // Fight button
147 this.fighttxt = this.game.add.text(0, 0, "Fight" , GAMEAPP.
    styleBTN);
148 this.fightButton = this.game.add.button(152, 338, 'button',
    GAMEAPP._CombatCheck, this, 2, 1, 0,1);
149 this.fightButton.anchor.set(0.5,0.5);
150 this.fightButton.scale.setTo(0.5,0.5);
151 this.fightButton.addChild(this.fighttxt).anchor.set(0.5,0.5);
152
153 // Talk button
154 var talktxt = this.game.add.text(0, 0, "Talk" , GAMEAPP.
    styleBTN);
155 var talkButton = this.game.add.button(152, 361, 'button', this
    ._talkToMonster, this, 2, 1, 0,1);
156 talkButton.anchor.set(0.5,0.5);
157 talkButton.scale.setTo(0.5,0.5);
158 talkButton.addChild(talktxt).anchor.set(0.5,0.5);
159
160 }else{
161     GAMEAPP.InfoText = " Looking for something to kill?\n Try the
        next room.";
162 }
163
164 GAMEAPP._RmAdmin(GAMEAPP.CrntRoom);
165
166 //character placement -
167 var entryBy = new Phaser.Point();
168 if(GAMEAPP.LastDoor=="North" || GAMEAPP.LastDoor=="North2"){
169     entryBy.copyFrom(this.doorS);
170     GAMEAPP.xPos = entryBy.x+15;
171     GAMEAPP.yPos = entryBy.y-35;
172 }
173
174 if(GAMEAPP.LastDoor=="East"){
175     entryBy.copyFrom(this.doorW);
176     GAMEAPP.xPos = entryBy.x+35;
177     GAMEAPP.yPos = entryBy.y+15;
178 }
179
180 if(GAMEAPP.LastDoor=="South"){
181     entryBy.copyFrom(this.doorN);
182     GAMEAPP.xPos = entryBy.x+15;
183     GAMEAPP.yPos = entryBy.y+35;
184 }
185
186 if(GAMEAPP.LastDoor=="West"){
187     entryBy.copyFrom(this.doorE);
188     GAMEAPP.xPos = entryBy.x-35;
189     GAMEAPP.yPos = entryBy.y+15;
190 }
191
192 if(GAMEAPP.LastDoor=="DownStairs"){
193     entryBy.copyFrom(this.upStairs);
194     GAMEAPP.xPos = entryBy.x-35;
195     GAMEAPP.yPos = entryBy.y+15;
196 }
197
198 if(GAMEAPP.LastDoor=="UpStairs"){
199     entryBy.copyFrom(this.downStairs);
```

```

200         GAMEAPP.xPos = entryBy.x-35;
201         GAMEAPP.yPos = entryBy.y+15;
202     }
203
204     this.player = this.game.add.sprite(GAMEAPP.xPos,GAMEAPP.yPos,
    'avatar');
205     this.player.frameName = 'avatar.bmp';
206     this.player.body.collideWorldBounds = true;
207     this.cursor = this.game.input.keyboard.createCursorKeys();
208
209 },
210
211 update: function(){
212     GAMEAPP._MoveChar(this.cursor,this.player);
213     this.game.physics.arcade.collide(this.player, this.Room);
214     if(this.game.physics.arcade.intersects(this.player, this.doorN)){
215         this.newRoom(this.doorN);
216     }
217     if(this.game.physics.arcade.intersects(this.player, this.
    downStairs)){
218         this.newRoom(this.downStairs);
219     }
220     //Check for monster in this room
221     if (MT[GAMEAPP.CrntRoom].Mnstr > 0) {
222         GAMEAPP._MoveMonster(this.cursor,this.enemy);
223         this.game.physics.arcade.collide(this.enemy, this.Room);
224         this.game.physics.arcade.overlap(this.player,this.enemy,
    GAMEAPP._CombatCheck,null,this);
225
226     }
227     //Check for treasure in this room
228     if (MT[GAMEAPP.CrntRoom].Tres > 0) {
229
230         this.game.physics.arcade.collide(this.player,this.Treasure,
    this.CTreasure,null,this);
231
232     }
233     this._toolTip.setText(GAMEAPP.InfoText);
234 },
235 //
236 // =====
237
238 //
239 // =====
240 //Main ARRA Door click handler
241 newRoom: function(door){
242     GAMEAPP.LastRoom = GAMEAPP.CrntRoom;
243     GAMEAPP.LastDoor = door.name;
244     console.log('Last Door Used: '+GAMEAPP.LastDoor);
245
246     if (door.name == "North"){
247
248         //debug
249         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
    +door.name);
250         this.camera.fade(0x000000, 200, false);
251         this.time.events.add(200, function() {
252             this.game.state.start('R'+MT[GAMEAPP.CrntRoom].North);
    //ARRA Main Entrance (hard coded) rv_3 through rv_8

```

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

```
301     GAMEAPP.InfoText = "... mmm, not in the mood to negotiate?!";
302 },
303 //
304 //=====
305 CTreasure: function() {
306     CCP.HGold += MT[GAMEAPP.CrntRoom].Tres;
307     this.Treasure.kill();
308     GAMEAPP.InfoText = "You found $" + MT[GAMEAPP.CrntRoom].Tres + " in
the chest.";
309     MT[GAMEAPP.CrntRoom].Tres = 0;
310     this._toolTip.setText("You found $" + MT[GAMEAPP.CrntRoom].Tres + "
in the chest.");
311 }
312 };
313 //
314 // =====
315 //create a box Image (pseudo graphics) for the HTML5 canvas.
316 var box = function(options) {
317     var bxImg = GAMEAPP.game.add.bitmapData(options.length, options.
width);
318     bxImg.ctx.beginPath();
319     bxImg.ctx.rect(0, 0, options.length, options.width);
320     bxImg.ctx.fillStyle = options.color;
321     bxImg.ctx.fill();
322     return bxImg;
323 };
324 //
325 // =====
326
327 /* End of file */
328 /* Location: ./js/state/R15.js */
329
330
```