

```
1  /**
2   * File Name: R13.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.R13 = {
25    preload: function() {
26      console.log("loading Room 13 state");
27      this.game.load.image('R13', 'assets/images/staticRooms/R13.jpg');
28      GAMEAPP.CrntRoom = 13;
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.image('floors', 'assets/images/floor/downStairs.bmp', 40,
38        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 13 state");
54      this.game.add.image(0, 0, 'R13');
55      var rect = new Phaser.Rectangle(37, 105, 300, 200);
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
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(87,270,box({length:270,width
57 :16,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.alignIn(rect, Phaser.TOP_CENTER);
63
64     this.SouthWall = this.game.add.sprite(67,268,box({length:270,width
65 :16,color:'#999'}));
66     this.SouthWall.body.immovable = true;
67     this.SouthWall.visible = false;
68     this.Room.add(this.SouthWall);
69
70     this.WestWall = this.game.add.sprite(53,110,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(280,110,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     this.EastWall.alignTo(rect, Phaser.RIGHT_CENTER);
82
83     //Room Doors
84     this.doorE = this.game.add.sprite(210,133,'doorButton');
85     //perfectly aligned over placeholder
86     GAMEAPP.doorE = this.doorE;
87     this.doorE.name = "East";
88     this.doorE.frameName = 'vertWood.bmp';
89     this.doorE.enableBody = true;
90     this.doorE.body.immovable = true;
91     this.doorE.inputEnabled = true;
92     this.game.physics.enable(this.doorE, Phaser.Physics.ARCADE);
93     this.doorE.events.onInputDown.add(this.newRoom, this);
94
95     this.upStairs = this.game.add.sprite(222,216,'floors');
96     //perfectly aligned over placeholder
97     this.upStairs.name = "UpStairs";
98     this.upStairs.enableBody = true;
99     this.upStairs.body.immovable = true;
100    this.upStairs.inputEnabled = true;
101    //this.upStairs.angle = 90;
102    this.game.physics.enable(this.upStairs, Phaser.Physics.ARCADE);
103    this.upStairs.events.onInputDown.add(this.newRoom, this);
```

```

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

```

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

```

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

```

```
252     }
253     */
254
255     if (door.name == "East"){
256
257         //debug
258         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
259         +door.name);
260         this.camera.fade(0x000000, 200, false);
261         this.time.events.add(200, function() {
262             this.game.state.start('R'+MT[GAMEAPP.CrntRoom].East);
263             //ARRA Main Entrance (hard coded) rv_3 through rv_8
264         }, this);
265     }
266
267     /**
268     if (door.name == "South" ){
269
270         //debug
271         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
272         +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: "
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].West);
292             //ARRA Main Entrance (hard coded) rv_3 through rv_8
293         }, this);
294     }
295
296     */
297     if (door.name == "UpStairs"){
298
299         //debug
300         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
301         +door.name);
302         this.camera.fade(0x000000, 200, false);
303         this.time.events.add(200, function() {
304             this.game.state.start('R'+MT[GAMEAPP.CrntRoom].Up);
305             //ARRA Main Entrance (hard coded) rv_3 through rv_8
306         }, this);
307     }
308
309     },
310     //
311     //=====
312     _talkToMonster: function(){
313         GAMEAPP.InfoText = "... mmm, not in the mood to negotiate?!";
314     },
315 }
```

```
301      //
302      //=====
303      CTreasure: function() {
304          CCP.HGold += MT[GAMEAPP.CrntRoom].Tres;
305          this.Treasure.kill();
306          GAMEAPP.InfoText = "You found $" + MT[GAMEAPP.CrntRoom].Tres + " in
the chest.";
307          MT[GAMEAPP.CrntRoom].Tres = 0;
308          this._toolTip.setText("You found $" + MT[GAMEAPP.CrntRoom].Tres +
in the chest.");
309      }
310  };
311      //
312      //=====
313      //create a box Image (pseudo graphics) for the HTML5 canvas.
314      var box = function(options) {
315          var bxImg = GAMEAPP.game.add.bitmapData(options.length, options.
width);
316          bxImg.ctx.beginPath();
317          bxImg.ctx.rect(0, 0, options.length, options.width);
318          bxImg.ctx.fillStyle = options.color;
319          bxImg.ctx.fill();
320          return bxImg;
321      };
322      //
323      //=====
324
325      /* End of file */
326      /* Location: ./js/state/R13.js */
327
328
```