

```
1  /**
2   * File Name: R7.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.R7 = {
25    preload: function(){
26      console.log("loading Room 7 state");
27      this.game.load.image('R7', 'assets/images/staticRooms/R7.jpg');
28      GAMEAPP.CrntRoom = 7;
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 7 state");
52      this.game.add.image(0, 0, 'R7');
53      var rect = new Phaser.Rectangle(45, 125, 270, 150);
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     //gap filled with this
56     this.interior1Wall = this.game.add.sprite(115,250,box({length:16,
57     width:40,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(0,0,box({length:270,width:16
64     ,color:'#999'}));
65     this.NorthWall.enableBody = true;
66     this.NorthWall.body.immovable = true;
67     this.NorthWall.visible = false;
68     this.Room.add(this.NorthWall);
69     this.NorthWall.alignTo(rect, Phaser.TOP_CENTER);
70
71     this.SouthWall = this.game.add.sprite(0,0,box({length:270,width:16
72     ,color:'#999'}));
73     this.SouthWall.body.immovable = true;
74     this.SouthWall.visible = false;
75     this.Room.add(this.SouthWall);
76     this.SouthWall.alignTo(rect, Phaser.BOTTOM_CENTER);
77
78     this.WestWall = this.game.add.sprite(0,0,box({length:16,width:160,
79     color:'#999'}));
80     this.WestWall.body.immovable = true;
81     this.WestWall.visible = false;
82     this.Room.add(this.WestWall);
83     this.WestWall.alignTo(rect, Phaser.LEFT_CENTER);
84
85     this.EastWall = this.game.add.sprite(0,0,box({length:16,width:160,
86     color:'#999'}));
87     this.EastWall.body.immovable = true;
88     this.EastWall.visible = false;
89     this.Room.add(this.EastWall);
90     this.EastWall.alignTo(rect, Phaser.RIGHT_CENTER);
91
92     //Room Doors
93     this.doorN = this.game.add.sprite(53,206,'doorButton');
94     //perfectly aligned over placeholder
95     this.doorN.name = "North";
96     this.doorN.frameName = 'horizWood.bmp';
97     this.doorN.enableBody = true;
98     this.doorN.body.immovable = true;
99     this.doorN.inputEnabled = true;
100    this.game.physics.enable(this.doorN, Phaser.Physics.ARCADE);
101    this.doorN.events.onInputDown.add(this.newRoom, this);
102
103    this.doorS = this.game.add.sprite(53,205,'doorButton');
104    //perfectly aligned over placeholder
```

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

```

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

```

```

    'avatar');
200     this.player.frameName = 'avatar.bmp';
201     this.player.body.collideWorldBounds = true;
202     this.cursor = this.game.input.keyboard.createCursorKeys();
203
204 },
205
206 update: function(){
207     GAMEAPP._MoveChar(this.cursor,this.player);
208     this.game.physics.arcade.collide(this.player, this.Room);
209     if(this.game.physics.arcade.intersects(this.player, this.doorN)){
210         this.newRoom(this.doorN);
211     }
212
213     if(this.game.physics.arcade.intersects(this.player, this.doorS)){
214         this.newRoom(this.doorS);
215     }
216     //Check for monster in this room
217     if (MT[GAMEAPP.CrntRoom].Mnstr > 0) {
218         GAMEAPP._MoveMonster(this.cursor,this.enemy);
219         this.game.physics.arcade.collide(this.enemy, this.Room);
220         this.game.physics.arcade.overlap(this.player,this.enemy,
221             GAMEAPP._CombatCheck,null,this);
222     }
223     //Check for treasure in this room
224     if (MT[GAMEAPP.CrntRoom].Tres > 0) {
225
226         this.game.physics.arcade.collide(this.player,this.Treasure,
227             this.CTreasure,null,this);
228     }
229     this._toolTip.setText(GAMEAPP.InfoText);
230 },
231 //
232 // =====
233
234 //
235 // =====
236 //Main ARRA Door click handler
237 newRoom: function(door){
238     GAMEAPP.LastRoom = GAMEAPP.CrntRoom;
239     GAMEAPP.LastDoor = door.name;
240     console.log('Last Door Used: '+GAMEAPP.LastDoor);
241 /**
242     if (door.name == "North"){
243
244         //debug
245         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked:
246             "+door.name);
247         this.camera.fade(0x000000, 200, false);
248         this.time.events.add(200, function() {
249
250             this.game.state.start('R'+MT[GAMEAPP.CrntRoom].North);
251             //ARRA Main Entrance (hard coded) rv_3 through rv_8
252             }, this);
253     }
254 */
255 /**

```

```
253         if (door.name == "East"){
254
255             //debug
256             console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
257                         +door.name);
258             this.camera.fade(0x000000, 200, false);
259             this.time.events.add(200, function() {
260
261                 this.game.state.start('R'+MT[GAMEAPP.CrntRoom].East);
262                 //ARRA Main Entrance (hard coded) rv_3 through rv_8
263             }, this);
264         }
265     */
266
267     if (door.name == "South" ){
268
269         //debug
270         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
271                     +door.name);
272         this.camera.fade(0x000000, 200, false);
273         this.time.events.add(200, function() {
274             this.game.state.start('R'+MT[GAMEAPP.CrntRoom].South);
275             //ARRA Main Entrance (hard coded) rv_3 through rv_8
276         }, this);
277     }
278
279     /**
280     if (door.name == "West"){
281
282         //debug
283         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
284                     +door.name);
285         this.camera.fade(0x000000, 200, false);
286         this.time.events.add(200, function() {
287
288             this.game.state.start('R'+MT[GAMEAPP.CrntRoom].West);
289             //ARRA Main Entrance (hard coded) rv_3 through rv_8
290         }, this);
291     }
292     */
293     },
294     //
295     //=====
296     _talkToMonster: function(){
297         GAMEAPP.InfoText = "... mmm, not in the mood to negotiate?!";
298     },
299     //
300     //=====
301     CTreasure: function(){
302         CCP.HGold += MT[GAMEAPP.CrntRoom].Tres;
303         this.Treasure.kill();
304         GAMEAPP.InfoText = "You found $" + MT[GAMEAPP.CrntRoom].Tres + " in
305         the chest.";
306         MT[GAMEAPP.CrntRoom].Tres = 0;
307         this._toolTip.setText("You found $" + MT[GAMEAPP.CrntRoom].Tres + "
308         in the chest.");
309     }
310 };
311 //
```

```
302      // =====
303      //create a box Image (pseudo graphics) for the HTML5 canvas.
304      var box = function(options) {
305          var bxImg = GAMEAPP.game.add.bitmapData(options.length,options.
              width);
306          bxImg.ctx.beginPath();
307          bxImg.ctx.rect(0,0,options.length,options.width);
308          bxImg.ctx.fillStyle = options.color;
309          bxImg.ctx.fill();
310          return bxImg;
311      };
312      //
313      // =====
314
315      /* End of file */
316      /* Location: ./js/state/R7.js */
317
318
```