

```
1  /**
2   * File Name: R5.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.R5 = {
25    preload: function(){
26      console.log("loading Room 5 state");
27      this.game.load.image('R5', 'assets/images/staticRooms/R5.jpg');
28      GAMEAPP.CrntRoom = 5;
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 5 state");
54      this.game.add.image(0, 0, 'R5');
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     //gap filled with this
57     this.interior1Wall = this.game.add.sprite(267,130,box({length:16,
width:70,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:
190,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.alignIn(rect, Phaser.RIGHT_CENTER);
87
88     //Room Doors
89     this.doorN = this.game.add.sprite(114,103,'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.doorW = this.game.add.sprite(25,198,'doorButton');
    //perfectly aligned over placeholder
100    GAMEAPP.doorW = this.doorW;
101    this.doorW.name = "West";
102    this.doorW.frameName = 'vertWood.bmp';
103    this.doorW.enableBody = true;
104    this.doorW.body.immovable = true;
105    this.doorW.inputEnabled = true;
106    this.game.physics.enable(this.doorW, Phaser.Physics.ARCADE);
107    this.doorW.events.onInputDown.add(this.newRoom, this);
108
109    this.downStairs = this.game.add.sprite(267,206,'floors');
    //perfectly aligned over placeholder
110    this.downStairs.name = "DownStairs";
111    this.downStairs.enableBody = true;
112    this.downStairs.body.immovable = true;
113    this.downStairs.inputEnabled = true;
114    this.game.physics.enable(this.downStairs, Phaser.Physics.ARCADE);
115    this.downStairs.events.onInputDown.add(this.newRoom, this);
116
117    this.upStairs = this.game.add.sprite(315,256,'floors');
    //perfectly aligned over placeholder
118    this.upStairs.name = "UpStairs";
119    this.upStairs.enableBody = true;
120    this.upStairs.body.immovable = true;
121    this.upStairs.inputEnabled = true;
122    this.upStairs.angle = 90;
123    this.game.physics.enable(this.upStairs, Phaser.Physics.ARCADE);
124    this.upStairs.events.onInputDown.add(this.newRoom, this);
125
126    //create treasure if needed; check for treasure in this room
127    if(MT[GAMEAPP.CrntRoom].Tres > 0){
128        GAMEAPP.InfoText = "Score: "+CCP.Score+"\n Click on the Door
        to enter.\nThere treasure here! See it?!!";
129        this.Treasure = this.game.add.sprite(200,200,'tres');
130        this.Treasure.frameName = 'treasure.bmp';
131        this.Treasure.body.collideWorldBounds = true;
132        this.Treasure.enableBody = true;
133        this.Treasure.body.immovable = true;
134        this.Treasure.inputEnabled = true;
135        this.game.physics.enable(this.Treasure, Phaser.Physics.ARCADE);
136        //clicking treasure for mobile interaction
137        this.Treasure.events.onInputDown.add(this.CTreasure, this);
138        this.Treasure.alignIn(rect, Phaser.CENTER);
139    }
140
141
142    this.inventorytxt = this.game.add.text(0, 0, "Inventory" , GAMEAPP
    .styleBTN); // "Attack" button text;
143    this.exittxt = this.game.add.text(0, 0, "Exit" , GAMEAPP.styleBTN
    ); // "Attack" button text;
144    this._toolTip = this.game.add.text(this.game.world.width-100, this
    .game.world.height-90, GAMEAPP.InfoText, GAMEAPP.styleTT);
145    this._toolTip.anchor.set(0.5);
146    GAMEAPP.RoomAlert = "";
147
148    // Inventory button
149    this.inventoryButton = this.game.add.button(65, 338, 'button',
```

```

150     GAMEAPP._InventoryRoom, this, 2, 1, 0,1);
151     this.inventoryButton.anchor.set(0.5,0.5);
152     this.inventoryButton.scale.setTo(0.5,0.5);
153     this.inventoryButton.addChild(this.inventorytxt).anchor.set(0.5,
154     0.5);
155
156     // Exit button
157     this.exitButton = this.game.add.button(65, 361, 'button', GAMEAPP.
158     _ExitGame, this, 2, 1, 0,1);
159     this.exitButton.anchor.set(0.5,0.5);
160     this.exitButton.scale.setTo(0.5,0.5);
161     this.exitButton.addChild(this.exittxt).anchor.set(0.5,0.5);
162
163     //create an opponent; Check for monster in this room
164     if(MT[GAMEAPP.CrntRoom].Mnstr > 0){
165         GAMEAPP.RoomAlert = "Beware, a " + M[MT[GAMEAPP.CrntRoom].
166         Mnstr].Race + " is here!"
167         GAMEAPP.raTxt = this.game.add.text(this.game.world.width-268,
168         65, GAMEAPP.RoomAlert, GAMEAPP.styleRA);
169         GAMEAPP.raTxt.anchor.set(0);
170         this.enemy = this.game.add.sprite(MT[GAMEAPP.CrntRoom].mxPos,
171         MT[GAMEAPP.CrntRoom].myPos, 'avatar');
172         this.enemy.frameName = String(M[MT[GAMEAPP.CrntRoom].Mnstr].
173         icon+'.bmp');
174         this.enemy.body.collideWorldBounds = true;
175         this.game.physics.enable(this.enemy, Phaser.Physics.ARCADE);
176
177         //this.enemy.alignIn(rect, Phaser.RIGHT_CENTER-250);
178
179         GAMEAPP._PUpdate();
180         //clicking fight for mobile interaction
181         // Fight button
182         this.fighttxt = this.game.add.text(0, 0, "Fight" , GAMEAPP.
183         styleBTN);
184         this.fightButton = this.game.add.button(152, 338, 'button',
185         GAMEAPP._CombatCheck, this, 2, 1, 0,1);
186         this.fightButton.anchor.set(0.5,0.5);
187         this.fightButton.scale.setTo(0.5,0.5);
188         this.fightButton.addChild(this.fighttxt).anchor.set(0.5,0.5);
189
190         // Talk button
191         var talktxt = this.game.add.text(0, 0, "Talk" , GAMEAPP.
192         styleBTN);
193         var talkButton = this.game.add.button(152, 361, 'button', this
194         ._talkToMonster, this, 2, 1, 0,1);
195         talkButton.anchor.set(0.5,0.5);
196         talkButton.scale.setTo(0.5,0.5);
197         talkButton.addChild(talktxt).anchor.set(0.5,0.5);
198     }else{
199         GAMEAPP.InfoText = " Looking for something to kill?\n Try the
200         next room.";
201     }
202
203     GAMEAPP._RmAdmin(GAMEAPP.CrntRoom);
204
205     //character placement -
206     var entryBy = new Phaser.Point();
207     if(GAMEAPP.LastDoor=="North" || GAMEAPP.LastDoor=="North2"){
208         entryBy.copyFrom(this.doorS);

```

```
197         GAMEAPP.xPos = entryBy.x+15;
198         GAMEAPP.yPos = entryBy.y-35;
199     }
200
201     if(GAMEAPP.LastDoor=="East"){
202         entryBy.copyFrom(this.doorW);
203         GAMEAPP.xPos = entryBy.x+35;
204         GAMEAPP.yPos = entryBy.y+15;
205     }
206
207     if(GAMEAPP.LastDoor=="South"){
208         entryBy.copyFrom(this.doorN);
209         GAMEAPP.xPos = entryBy.x+15;
210         GAMEAPP.yPos = entryBy.y+35;
211     }
212
213     if(GAMEAPP.LastDoor=="West"){
214         entryBy.copyFrom(this.doorE);
215         GAMEAPP.xPos = entryBy.x-35;
216         GAMEAPP.yPos = entryBy.y+15;
217     }
218
219     if(GAMEAPP.LastDoor=="DownStairs"){
220         entryBy.copyFrom(this.upStairs);
221         GAMEAPP.xPos = entryBy.x-35;
222         GAMEAPP.yPos = entryBy.y+15;
223     }
224
225     if(GAMEAPP.LastDoor=="UpStairs"){
226         entryBy.copyFrom(this.downStairs);
227         GAMEAPP.xPos = entryBy.x-35;
228         GAMEAPP.yPos = entryBy.y+15;
229     }
230
231     this.player = this.game.add.sprite(GAMEAPP.xPos,GAMEAPP.yPos,
    'avatar');
232     this.player.frameName = 'avatar.bmp';
233     this.player.body.collideWorldBounds = true;
234     this.cursor = this.game.input.keyboard.createCursorKeys();
235
236 },
237
238 update: function(){
239     GAMEAPP._MoveChar(this.cursor,this.player);
240     this.game.physics.arcade.collide(this.player, this.Room);
241     if(this.game.physics.arcade.intersects(this.player, this.doorN)){
242         this.newRoom(this.doorN);
243     }
244     if(this.game.physics.arcade.intersects(this.player, this.doorW)){
245         this.newRoom(this.doorW);
246     }
247     if(this.game.physics.arcade.intersects(this.player, this.
    downStairs)){
248         this.newRoom(this.downStairs);
249     }
250     if(this.game.physics.arcade.intersects(this.player, this.upStairs
    )){
251         this.newRoom(this.upStairs);
252     }
```

```
253         //Check for monster in this room
254         if (MT[GAMEAPP.CrntRoom].Mnstr > 0) {
255             GAMEAPP._MoveMonster(this.cursor,this.enemy);
256             this.game.physics.arcade.collide(this.enemy, this.Room);
257             this.game.physics.arcade.overlap(this.player,this.enemy,
                GAMEAPP._CombatCheck,null,this);
258
259         }
260         //Check for treasure in this room
261         if (MT[GAMEAPP.CrntRoom].Tres > 0) {
262
263             this.game.physics.arcade.collide(this.player,this.Treasure,
                this.CTreasure,null,this);
264
265         }
266         this._toolTip.setText(GAMEAPP.InfoText);
267     },
268     //
269     // =====
270
271     //
272     // =====
273     //Main ARRA Door click handler
274     newRoom: function(door){
275         GAMEAPP.LastRoom = GAMEAPP.CrntRoom;
276         GAMEAPP.LastDoor = door.name;
277         console.log('Last Door Used: '+GAMEAPP.LastDoor);
278
279         if (door.name == "North"){
280
281             //debug
282             console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
                +door.name);
283             this.camera.fade(0x000000, 200, false);
284             this.time.events.add(200, function() {
285                 this.game.state.start('R'+MT[GAMEAPP.CrntRoom].North);
                //ARRA Main Entrance (hard coded) rv_3 through rv_8
286             }, this);
287         }
288
289         /**
290         if (door.name == "East"){
291
292             //debug
293             console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
                +door.name);
294             this.camera.fade(0x000000, 200, false);
295             this.time.events.add(200, function() {
296
297                 this.game.state.start('R'+MT[GAMEAPP.CrntRoom].East);
                //ARRA Main Entrance (hard coded) rv_3 through rv_8
298             }, this);
299         }
300         /**
301         if (door.name == "South" ){
302
303             //debug
304             console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked:
```

```

        "+door.name);
305     this.camera.fade(0x000000, 200, false);
306     this.time.events.add(200, function() {
307
        this.game.state.start('R'+MT[GAMEAPP.CrntRoom].South);
        //ARRA Main Entrance (hard coded) rv_3 through rv_8
308     }, this);
309 }
310 */
311
312     if (door.name == "West"){
313
314         //debug
315         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
        +door.name);
316         this.camera.fade(0x000000, 200, false);
317         this.time.events.add(200, function() {
318             this.game.state.start('R'+MT[GAMEAPP.CrntRoom].West);
            //ARRA Main Entrance (hard coded) rv_3 through rv_8
319         }, this);
320     }
321
322     if (door.name == "DownStairs"){
323
324         //debug
325         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
        +door.name);
326         this.camera.fade(0x000000, 200, false);
327         this.time.events.add(200, function() {
328             this.game.state.start('R'+MT[GAMEAPP.CrntRoom].Dn);
            //ARRA Main Entrance (hard coded) rv_3 through rv_8
329         }, this);
330     }
331
332     if (door.name == "UpStairs"){
333
334         //debug
335         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
        +door.name);
336         this.camera.fade(0x000000, 200, false);
337         this.time.events.add(200, function() {
338             this.game.state.start('R'+MT[GAMEAPP.CrntRoom].Up);
            //ARRA Main Entrance (hard coded) rv_3 through rv_8
339         }, this);
340     }
341
342 },
343 //
344 //=====
345 _talkToMonster: function(){
346     GAMEAPP.InfoText = "... mmm, not in the mood to negotiate?!";
347 },
348 //
349 //=====
350 CTreasure: function(){
351     CCP.HGold += MT[GAMEAPP.CrntRoom].Tres;
352     this.Treasure.kill();
353     GAMEAPP.InfoText = "You found $" + MT[GAMEAPP.CrntRoom].Tres + " in
        the chest.";

```

```
354         MT[GAMEAPP.CrntRoom].Tres = 0;
355         this._toolTip.setText("You found $" + MT[GAMEAPP.CrntRoom].Tres +
                                "in the chest.");
356     }
357 };
358 //
359 // =====
360 //create a box Image (pseudo graphics) for the HTML5 canvas.
361 var box = function(options) {
362     var bxImg = GAMEAPP.game.add.bitmapData(options.length,options.
        width);
363     bxImg.ctx.beginPath();
364     bxImg.ctx.rect(0,0,options.length,options.width);
365     bxImg.ctx.fillStyle = options.color;
366     bxImg.ctx.fill();
367     return bxImg;
368 };
369 //
370 // =====
371
372 /* End of file */
373 /* Location: ./js/state/R5.js */
374
375
```