

```
1  /**
2   * File Name: R2.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.R2 = {
25    preload: function(){
26      console.log("loading Room 2 state");
27      this.game.load.image('R2', 'assets/images/staticRooms/R2.jpg');
28      GAMEAPP.CrntRoom = 2;
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 2 state");
52      this.game.add.image(0, 0, 'R2');
53
54      var rect = new Phaser.Rectangle(57, 112, 250, 190);
55      //Set a neutral background color
56      this.game.stage.backgroundColor = "#000";
57    }
58  }
```

```
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
: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
:16,color:'#999'}));
64 this.SouthWall.body.immovable = true;
65 this.SouthWall.visible = false;
66 this.Room.add(this.SouthWall);
67 this.SouthWall.alignTo(rect, Phaser.BOTTOM_CENTER);
68
69 this.WestWall = this.game.add.sprite(200,16,box({length:16,width:
190,color:'#999'}));
70 this.WestWall.body.immovable = true;
71 this.WestWall.visible = false;
72 this.Room.add(this.WestWall);
73 this.WestWall.alignTo(rect, Phaser.LEFT_CENTER);
74
75 this.EastWall = this.game.add.sprite(90,16,box({length:16,width:
190,color:'#999'}));
76 this.EastWall.body.immovable = true;
77 this.EastWall.visible = false;
78 this.Room.add(this.EastWall);
79 this.EastWall.alignTo(rect, Phaser.RIGHT_CENTER);
80
81 //Room Doors
82 this.doorN = this.game.add.sprite(148,100,'doorButton');
83 //perfectly aligned over placeholder
84 GAMEAPP.doorN = this.doorN;
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(156,286,'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;
```

```

99      this.game.physics.enable(this.doorS, Phaser.Physics.ARCADE);
100     this.doorS.events.onInputDown.add(this.newRoom, this);
101
102     this.doorE = this.game.add.sprite(288,196,'doorButton');
    //perfectly aligned over placeholder
103     GAMEAPP.doorE = this.doorE;
104     this.doorE.name = "East";
105     this.doorE.frameName = 'vertWood.bmp';
106     this.doorE.enableBody = true;
107     this.doorE.body.immovable = true;
108     this.doorE.inputEnabled = true;
109     this.game.physics.enable(this.doorE, Phaser.Physics.ARCADE);
110     this.doorE.events.onInputDown.add(this.newRoom, this);
111
112     //create treasure if needed; check for treasure in this room
113     if(MT[GAMEAPP.CrntRoom].Tres > 0){
114         GAMEAPP.InfoText = "Score: "+CCP.Score+"\n Click on the Door
    to enter.\nThere treasure here! See it?!!";
115         this.Treasure = this.game.add.sprite(200,200,'tres');
116         this.Treasure.frameName = 'treasure.bmp';
117         this.Treasure.body.collideWorldBounds = true;
118         this.Treasure.enableBody = true;
119         this.Treasure.body.immovable = true;
120         this.Treasure.inputEnabled = true;
121         this.game.physics.enable(this.Treasure, Phaser.Physics.ARCADE);
122         //clicking treasure for mobile interaction
123         this.Treasure.events.onInputDown.add(this.CTreasure, this);
124         this.Treasure.alignIn(rect, Phaser.CENTER);
125
126     }
127
128     this.inventorytxt = this.game.add.text(0, 0, "Inventory" , GAMEAPP
    .styleBTN); // "Attack" button text;
129     this.exittxt = this.game.add.text(0, 0, "Exit" , GAMEAPP.styleBTN
    ); // "Attack" button text;
130     this._toolTip = this.game.add.text(this.game.world.width-100, this
    .game.world.height-90, GAMEAPP.InfoText, GAMEAPP.styleTT);
131     this._toolTip.anchor.set(0.5);
132     GAMEAPP.RoomAlert = "";
133
134     // Inventory button
135     this.inventoryButton = this.game.add.button(65, 338, 'button',
    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,
    0.5);
139
140     // Exit button
141     this.exitButton = this.game.add.button(65, 361, 'button', GAMEAPP.
    _ExitGame, this, 2, 1, 0,1);
142     this.exitButton.anchor.set(0.5,0.5);
143     this.exitButton.scale.setTo(0.5,0.5);
144     this.exitButton.addChild(this.exittxt).anchor.set(0.5,0.5);
145
146     //create an opponent; Check for monster in this room
147     if(MT[GAMEAPP.CrntRoom].Mnstr > 0){
148         GAMEAPP.RoomAlert = "Beware, a " + M[MT[GAMEAPP.CrntRoom].
    Mnstr].Race + " is here!"

```

```

149     GAMEAPP.raTxt = this.game.add.text(this.game.world.width-268,
150     65, GAMEAPP.RoomAlert, GAMEAPP.styleRA);
151     GAMEAPP.raTxt.anchor.set(0);
152     this.enemy = this.game.add.sprite(MT[GAMEAPP.CrntRoom].mxPos,
153     MT[GAMEAPP.CrntRoom].myPos, 'avatar');
154     this.enemy.frameName = String(M[MT[GAMEAPP.CrntRoom].Mnstr].
155     icon+'.bmp');
156     this.enemy.body.collideWorldBounds = true;
157     this.game.physics.enable(this.enemy, Phaser.Physics.ARCADE);
158
159     this.enemy.alignIn(rect, Phaser.RIGHT_CENTER);
160
161     GAMEAPP._PUpdate();
162     //clicking fight for mobile interaction
163     // Fight button
164     this.fighttxt = this.game.add.text(0, 0, "Fight" , GAMEAPP.
165     styleBTN);
166     this.fightButton = this.game.add.button(152, 338, 'button',
167     GAMEAPP._CombatCheck, this, 2, 1, 0,1);
168     this.fightButton.anchor.set(0.5,0.5);
169     this.fightButton.scale.setTo(0.5,0.5);
170     this.fightButton.addChild(this.fighttxt).anchor.set(0.5,0.5);
171
172     // Talk button
173     var talktxt = this.game.add.text(0, 0, "Talk" , GAMEAPP.
174     styleBTN);
175     var talkButton = this.game.add.button(152, 361, 'button', this
176     ._talkToMonster, this, 2, 1, 0,1);
177     talkButton.anchor.set(0.5,0.5);
178     talkButton.scale.setTo(0.5,0.5);
179     talkButton.addChild(talktxt).anchor.set(0.5,0.5);
180
181     }else{
182         GAMEAPP.InfoText = " Looking for something to kill?\n Try the
183         next room.";
184     }
185
186     GAMEAPP._RmAdmin(GAMEAPP.CrntRoom);
187
188     //character placement -
189     var entryBy = new Phaser.Point();
190     if(GAMEAPP.LastDoor=="South"){
191         entryBy.copyFrom(this.doorN);
192         GAMEAPP.xPos = entryBy.x+15;
193         GAMEAPP.yPos = entryBy.y+35;
194     }
195     if(GAMEAPP.LastDoor=="North"){
196         entryBy.copyFrom(this.doorS);
197         GAMEAPP.xPos = entryBy.x+15;
198         GAMEAPP.yPos = entryBy.y-35;
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
207     update: function() {
208         GAMEAPP._MoveChar(this.cursor, this.player);
209         this.game.physics.arcade.collide(this.player, this.Room);
210         if (this.game.physics.arcade.intersects(this.player, this.doorN)) {
211             this.newRoom(this.doorN);
212         }
213         if (this.game.physics.arcade.intersects(this.player, this.doorS)) {
214             this.newRoom(this.doorS);
215         }
216         if (this.game.physics.arcade.intersects(this.player, this.doorE)) {
217             this.newRoom(this.doorE);
218         }
219         //Check for monster in this room
220         if (MT[GAMEAPP.CrntRoom].Mnstr > 0) {
221             GAMEAPP._MoveMonster(this.cursor, this.enemy);
222             this.game.physics.arcade.collide(this.enemy, this.Room);
223             this.game.physics.arcade.overlap(this.player, this.enemy,
                GAMEAPP._CombatCheck, null, this);
224
225         }
226         //Check for treasure in this room
227         if (MT[GAMEAPP.CrntRoom].Tres > 0) {
228
229             this.game.physics.arcade.collide(this.player, this.Treasure,
                this.CTreasure, null, this);
230
231         }
232         this._toolTip.setText(GAMEAPP.InfoText);
233     },
234     //
235     // =====
236
237     //
238     // =====
239     //Main ARRA Door click handler
240     newRoom: function(door) {
241         GAMEAPP.LastRoom = GAMEAPP.CrntRoom;
242         GAMEAPP.LastDoor = door.name;
243         console.log('Last Door Used: '+GAMEAPP.LastDoor);
244
245         if (door.name == "North") {
246
247             //debug
248             console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
                +door.name);
249             this.camera.fade(0x000000, 200, false);
250             this.time.events.add(200, function() {
251                 this.game.state.start('R'+MT[GAMEAPP.CrntRoom].North);
252                 //ARRA Main Entrance (hard coded) rv_3 through rv_8
253             }, this);
254         }
255     }

```

```
254
255     if (door.name == "East"){
256
257         //debug
258         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
259             +door.name);
259         this.camera.fade(0x000000, 200, false);
260         this.time.events.add(200, function() {
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     if (door.name == "South" ){
266
267         //debug
268         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
269             +door.name);
269         this.camera.fade(0x000000, 200, false);
270         this.time.events.add(200, function() {
271             this.game.state.start('R'+MT[GAMEAPP.CrntRoom].South);
272             //ARRA Main Entrance (hard coded) rv_3 through rv_8
273         }, this);
274     }
275     //
276     //=====
277     _talkToMonster: function(){
278         GAMEAPP.InfoText = "... mmm, not in the mood to negotiate?!";
279     },
280     //
281     //=====
282     CTreasure: function(){
283         CCP.HGold += MT[GAMEAPP.CrntRoom].Tres;
284         this.Treasure.kill();
285         GAMEAPP.InfoText = "You found $" + MT[GAMEAPP.CrntRoom].Tres + " in
286             the chest.";
286         MT[GAMEAPP.CrntRoom].Tres = 0;
287         this._toolTip.setText("You found $" + MT[GAMEAPP.CrntRoom].Tres + "
288             in the chest.");
289     }
290 };
291 //
292 // =====
293 //create a box Image (pseudo graphics) for the HTML5 canvas.
294 var box = function(options) {
295     var bxImg = GAMEAPP.game.add.bitmapData(options.length, options.
296         width);
297     bxImg.ctx.beginPath();
298     bxImg.ctx.rect(0,0,options.length,options.width);
299     bxImg.ctx.fillStyle = options.color;
300     bxImg.ctx.fill();
301     return bxImg;
302 };
303 //
304 // =====
305 /* End of file */
306 /* Location: ./js/state/R2.js */
```

306

307