

```
1  /**
2   * File Name: R1.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.R1 = {
25    preload: function(){
26      console.log("loading Room 1 state");
27      GAMEAPP.CrntRoom = 1;
28      this.game.load.image('R1', 'assets/images/staticRooms/R1.jpg');
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 1 state");
52      this.game.add.image(0, 0, 'R1');
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 //var doorS = this.game.add.sprite(160,290,'doorButton');
83 //perfectly aligned over placeholder
84 this.doorsS = this.game.add.sprite(160,290,'doorButton');
85 //perfectly aligned over placeholder
86 GAMEAPP.doorsS = this.doorsS;
87 this.doorsS.name = "South";
88 this.doorsS.frameName = 'horizWood.bmp';
89 this.doorsS.enableBody = true;
90 this.doorsS.body.immovable = true;
91 this.doorsS.inputEnabled = true;
92 this.game.physics.enable(this.doorsS, Phaser.Physics.ARCADE);
93 this.doorsS.events.onInputDown.add(this.newRoom, this);
94
95 //Room Doors
96 var doorW = this.game.add.sprite(30,180,'doorButton');
97 //perfectly aligned over placeholder
98 doorW.name = "West";
99 doorW.frameName = 'vertMetal.bmp';
100 doorW.body.immovable = true;
```

```

98     doorW.inputEnabled = true;
99     doorW.events.onInputDown.add(this.newRoom, this);
100
101     //create treasure if needed; check for treasure in this room
102     if(MT[GAMEAPP.CrntRoom].Tres > 0){
103         GAMEAPP.InfoText = "Score: "+CCP.Score+"\n Click on the Door
104         to enter.\nThere treasure here! See it?!!";
105         this.Treasure = this.game.add.sprite(200,200,'tres');
106         this.Treasure.frameName = 'treasure.bmp';
107         this.Treasure.body.collideWorldBounds = true;
108         this.Treasure.enableBody = true;
109         this.Treasure.body.immovable = true;
110         this.Treasure.inputEnabled = true;
111         this.game.physics.enable(this.Treasure, Phaser.Physics.ARCADE);
112         //clicking treasure for mobile interaction
113         this.Treasure.events.onInputDown.add(this.CTreasure, this);
114         this.Treasure.alignIn(rect, Phaser.CENTER);
115     }
116
117     this.inventorytxt = this.game.add.text(0, 0, "Inventory" , GAMEAPP
118     .styleBTN);
119     this.exittxt = this.game.add.text(0, 0, "Exit" , GAMEAPP.styleBTN
120     );
121     this._toolTip = this.game.add.text(this.game.world.width-100, this
122     .game.world.height-90, GAMEAPP.InfoText, GAMEAPP.styleTT);
123     this._toolTip.anchor.set(0.5);
124     GAMEAPP.RoomAlert = "";
125
126     // Inventory button
127     this.inventoryButton = this.game.add.button(65, 338, 'button',
128     GAMEAPP._InventoryRoom, this, 2, 1, 0,1);
129     this.inventoryButton.anchor.set(0.5,0.5);
130     this.inventoryButton.scale.setTo(0.5,0.5);
131     this.inventoryButton.addChild(this.inventorytxt).anchor.set(0.5,
132     0.5);
133
134     // Exit button
135     this.exitButton = this.game.add.button(65, 361, 'button', GAMEAPP.
136     _ExitGame, this, 2, 1, 0,1);
137     this.exitButton.anchor.set(0.5,0.5);
138     this.exitButton.scale.setTo(0.5,0.5);
139     this.exitButton.addChild(this.exittxt).anchor.set(0.5,0.5);
140
141     //create an opponent; Check for monster in this room
142     if(MT[GAMEAPP.CrntRoom].Mnstr > 0){
143         GAMEAPP.RoomAlert = "Beware, a " + M[MT[GAMEAPP.CrntRoom].
144         Mnstr].Race + " is here!"
145         GAMEAPP.raTxt = this.game.add.text(this.game.world.width-268,
146         65, GAMEAPP.RoomAlert, GAMEAPP.styleRA);
147         GAMEAPP.raTxt.anchor.set(0);
148         this.enemy = this.game.add.sprite(MT[GAMEAPP.CrntRoom].mxPos,
149         MT[GAMEAPP.CrntRoom].myPos, 'avatar');
150         this.enemy.frameName = String(M[MT[GAMEAPP.CrntRoom].Mnstr].
151         icon+'.bmp');
152         this.enemy.body.collideWorldBounds = true;
153         this.game.physics.enable(this.enemy, Phaser.Physics.ARCADE);
154
155         this.enemy.alignIn(rect, Phaser.RIGHT_CENTER);

```

```

146
147     GAMEAPP._PUpdate();
148     //clicking fight for mobile interaction
149     // Fight button
150     this.fighttxt = this.game.add.text(0, 0, "Fight" , GAMEAPP.
        styleBTN);
151     this.fightButton = this.game.add.button(152, 338, 'button',
        GAMEAPP._CombatCheck, this, 2, 1, 0,1);
152     this.fightButton.anchor.set(0.5,0.5);
153     this.fightButton.scale.setTo(0.5,0.5);
154     this.fightButton.addChild(this.fighttxt).anchor.set(0.5,0.5);
155
156     // Talk button
157     var talktxt = this.game.add.text(0, 0, "Talk" , GAMEAPP.
        styleBTN);
158     var talkButton = this.game.add.button(152, 361, 'button', this
        ._talkToMonster, this, 2, 1, 0,1);
159     talkButton.anchor.set(0.5,0.5);
160     talkButton.scale.setTo(0.5,0.5);
161     talkButton.addChild(talktxt).anchor.set(0.5,0.5);
162
163     }else{
164         GAMEAPP.InfoText = " Looking for something to kill?\n Try the
            next room.";
165     }
166
167     GAMEAPP._RmAdmin(GAMEAPP.CrntRoom);
168
169     //character placement - R1 exception use case
170     var entryBy = new Phaser.Point();
171     if(GAMEAPP.LastDoor!="Main Entrance"){
172         entryBy.copyFrom(this.doorS);
173         GAMEAPP.xPos = entryBy.x+15;
174         GAMEAPP.yPos = entryBy.y-35;
175         this.player = this.game.add.sprite(GAMEAPP.xPos,GAMEAPP.yPos,
            'avatar');
176     }else{
177         this.player = this.game.add.sprite(GAMEAPP.xPos,GAMEAPP.yPos,
            'avatar');
178         this.player.alignIn(rect, Phaser.LEFT_CENTER);
179     }
180
181     //this.player =
182     this.game.add.sprite(GAMEAPP.xPos,GAMEAPP.yPos,'avatar');
183     this.player.frameName = 'avatar.bmp';
184     this.player.body.collideWorldBounds = true;
185     this.cursor = this.game.input.keyboard.createCursorKeys();
186
187 },
188
189 update: function(){
190     GAMEAPP._MoveChar(this.cursor,this.player);
191     this.game.physics.arcade.collide(this.player, this.Room);
192     if(this.game.physics.arcade.intersects(this.player, this.doorS)){
193         this.newRoom(this.doorS);
194     }
195     //Check for monster in this room
196     if (MT[GAMEAPP.CrntRoom].Mnstr > 0) {

```

```

197         GAMEAPP._MoveMonster(this.cursor, this.enemy);
198         this.game.physics.arcade.collide(this.enemy, this.Room);
199         this.game.physics.arcade.overlap(this.player, this.enemy,
        GAMEAPP._CombatCheck, null, this);
200
201     }
202     //Check for treasure in this room
203     if (MT[GAMEAPP.CrntRoom].Tres > 0) {
204
205         this.game.physics.arcade.collide(this.player, this.Treasure,
        this.CTreasure, null, this);
206
207     }
208     this._toolTip.setText(GAMEAPP.InfoText);
209 },
210 //
211 // =====
212
213 //
214 // =====
215 //Main ARRA Door click handler
216 // this is an exception use case
217 newRoom: function(door){
218     GAMEAPP.LastRoom = GAMEAPP.CrntRoom;
219     GAMEAPP.LastDoor = door.name;
220     console.log('Last Door Used: '+GAMEAPP.LastDoor);
221
222     if (door.name == "South"){
223
224         //debug
225         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
        +door.name);
226         this.camera.fade(0x000000, 200, false);
227         this.time.events.add(200, function() {
228             this.game.state.start('R'+MT[GAMEAPP.CrntRoom].South);
                //ARRA Main Entrance (hard coded) rv_3 through rv_8
229         }, this);
230     }else{
231         GAMEAPP.InfoText = "West door is jammed shut!";
232         console.log("New Room #: "+GAMEAPP.CrntRoom+"; Door Clicked: "
        +door.name);
233
234     }
235
236 },
237 //
238 // =====
239 _talkToMonster: function(){
240     GAMEAPP.InfoText = "... mmm, not in the mood to negotiate?!";
241 },
242 //
243 // =====
244 CTreasure: function(){
245     CCP.HGold += MT[GAMEAPP.CrntRoom].Tres;
246     this.Treasure.kill();
247     GAMEAPP.InfoText = "You found $" + MT[GAMEAPP.CrntRoom].Tres + " in
        the chest.";
248     MT[GAMEAPP.CrntRoom].Tres = 0;
249     this._toolTip.setText("You found $" + MT[GAMEAPP.CrntRoom].Tres + "

```

```
        in the chest.");
250     }
251 };
252 //
253 // =====
254 //create a box Image (pseudo graphics) for the HTML5 canvas.
255 var box = function(options) {
256     var bxImg = GAMEAPP.game.add.bitmapData(options.length,options.
        width);
257     bxImg.ctx.beginPath();
258     bxImg.ctx.rect(0,0,options.length,options.width);
259     bxImg.ctx.fillStyle = options.color;
260     bxImg.ctx.fill();
261     return bxImg;
262 };
263 //
264 // =====
265
266 /* End of file */
267 /* Location: ./js/state/R1.js */
268
269
```