```
1   /**
2   Copyright © 1997-2016, Stephen Gose LLC. All rights reserved.
3   Visit our game show case:   http://www.renown-games.com
4   License information:        http://shop.pbmcube.net/
5   Affiliate Information:
    http://www.sgose.com/products/affiliates-program/
6
7    * The above copyright notice and this permission notice shall be
     included in
8    * all copies or substantial portions of the Software.
9    *
10   * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
     EXPRESS OR
11   * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
     MERCHANTABILITY,
12   * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT
     SHALL THE
13   * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
14   * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
     ARISING FROM,
15   * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
     IN
16   * THE SOFTWARE.
17   *
18   *
19   * File Name: demo.js
20   * Live Demo:   http://makingbrowsergames.com/p3gp-book/_p3-2DRooms/
21   * Source Code: http://makingbrowsergames.com/book/
22   * Description: File for controlling and displaying game scenes;
23   *               managing global variables throughout game state.
24
25   * Author: Stephen Gose
26   * Version: 0.0.0.19971027-1  //semantic version: by (0.0.0.date-#)
27   * Phaser Version: 3.16+
28   * Author URL: http://www.stephen-gose.com/
29   * Support: support@pbmcube.com
30   *
31   * \u00A9 Copyright ©  1974-2017 Stephen Gose LLC. All rights reserved.
32   * Redistribution of part or whole of this file and
33   *   the accompanying files is strictly prohibited.
34   *
35   * Do not sell! Do not distribute!
36   *  Please refer to the Terms of Use and
37   *    End Users License Agreement (EULA).
38   *  This is a licensed file with EULA permission.
39   *
40   * Search for "//**TODO**" to tailor this file for your production;
41   *   doing so will void any support agreement.
42   *
43   * Construction References:
44   * http://www.adequatelygood.com/JavaScript-Module-Pattern-In-Depth.html
45   *
     http://labs.phaser.io/edit.html?src=src\scenes\tutorial\scene%20controller
     .js
46   * http://labs.phaser.io/edit.html?src=src\scenes\changing%20scene.js
47   *
     http://www.html5gamedevs.com/topic/37199-extending-phaser-classes-es6-synt
     ax-vs-phaserclass-differences/
48   *
```

```
 */
/*
 * This file is automatically loaded from state/load.js
 * to change default state - change state/load.js at line: 34
 */
"use strict";
var box = {};
var avatar, player, monster, cursors, speed, Tribe;
var Room, NorthWall, EastWall, SouthWall, WestWall;
var DoorGroup, doorN, doorE, doorS, doorW;
var _RmText, newHUD, _RmLabel;

window.Rooms2D.state.demo = new Phaser.Class({

    Extends: Phaser.Scene,
    initialize: function demo() {
        Phaser.Scene.call(this, {
            key: 'demo'
        });
    },

    create: function () {
        // ============
        // Example 4.1: Prototyping Graphics begins
        // ============
        // create a character avatar using the box prototype method
        // player1 = this.physics.add.sprite(64, 64,box({who: this,
        whereX: 64, whereY: 64,length:32,width:32,color: 0x0000FF,
        border: 0xCCCCCC})).setInteractive();
        // create a character avatar using Phaser v3.13.x rectangle method
        player = this.add.rectangle(64, 64, 32, 32, 0x0000FF);
        player.setInteractive().setStrokeStyle(5, 0x3399CC);
        this.physics.add.existing(player);
        player.body.collideWorldBounds = true;
        player.body.onWorldBounds = true;
        player.body.bounce.x = 16;
        player.body.bounce.y = 16;
        console.log("Moving Blue character avatar created as 'player'
        variable.");
        console.log("Player obj: === Ext? " + Object.isExtensible(player));
        //console.log(Object.values(player));
        //console.log(Object.getPrototypeOf(player));
        console.log(Object.getOwnPropertyDescriptors(player));
        console.log(" === End Avatar Object: ========= ");
        // ============
        // Example 4.2: Prototyping Movement Properties
        //    frame refresh and display updates
        // ============
        cursors = this.input.keyboard.createCursorKeys();
        speed = 250;

        // Example 4.2: ends
        // ============

        // Example 4.1: ends
        // ============
        // Example 4.4: World Boundaries Integration begins
        // ============
        // Create Room Walls using rectangles
```

Copyright © 1997-2019, Stephen Gose. All rights reserved

```
105             Room = this.physics.add.staticGroup();
106             //Room = this.physics.add.group();
107 
108             //Room Walls
109             // Creating rectangles; review console in this experiment
110             NorthWall=this.add.rectangle(0, 0, 391, 16, 0x999999).setOrigin(0);
111             EastWall=this.add.rectangle(375, 16.5, 16, 375, 0x999999).
                setOrigin(0);
112             WestWall=this.add.rectangle(0, 16.5, 16, 375, 0x999999).setOrigin(
                0);
113             SouthWall=this.add.rectangle(0, 359, 390, 16, 0x999999).setOrigin(
                0);
114             //console.log("NorthWall obj: === Ext? "+Object.isExtensible(
                NorthWall ));
115             //console.info(NorthWall);
116             console.log("Room external walls created.");
117             Room.addMultiple([NorthWall, EastWall, SouthWall, WestWall]);
118             // Example 4.4: ends
119             // ============
120             // Room Doors configured (not optimized)
121             // Create Room Walls using rectangles
122             doorN = this.add.rectangle(config.width / 3, 0, 60, 18, 0xFFFFFF).
                setOrigin(0.5, 0).setStrokeStyle(3, 0xCCCCCC);
123             doorN.setInteractive({
124                 useHandCursor: true
125             }); //visual clue
126             doorN.name = "North";
127             doorN.enableBody = true;
128             doorN.on('pointerup', function (pointer) {
129                 console.info("DoorN clicked. ");
130                 newRoom(doorN);
131                 resetRoom();
132             }, this);
133             if ((Rooms2D.CrntRoomY == 0) || (Rooms2D.CrntRoomX == 0) || (
                Rooms2D.CrntRoomX == 2)) {
134                 doorN.visible = false;
135                 doorN.setInteractive(false);
136             } else {
137                 doorN.visible = true;
138                 doorN.setInteractive(true);
139             }
140 
141             doorE = this.add.rectangle(372, config.height / 2, 18, 60,
                0xFFFFFF).setOrigin(0, 0.5).setStrokeStyle(3, 0xCCCCCC);
142             doorE.setInteractive({
143                 useHandCursor: true
144             }); //visual clue
145             doorE.name = "East";
146             doorE.enableBody = true;
147             //this.physics.add.existing(doorE);
148             doorE.on('pointerup', function (pointer) {
149                 console.info("DoorE clicked. ");
150                 newRoom(doorE);
151                 resetRoom();
152             }, this);
153             if ((Rooms2D.CrntRoomX == 3) || (Rooms2D.CrntRoomX == 1 && Rooms2D
                .CrntRoomY == 1)) {
154                 doorE.visible = false;
155                 doorE.setInteractive(false);
```

Copyright © 1997-2019, Stephen Gose. All rights reserved

```
156             } else {
157                 doorE.visible = true;
158                 doorE.setInteractive(true);
159             }
160 
161             doorS = this.add.rectangle(config.width / 3, 358, 60, 18, 0xFFFFFF
                ).setOrigin(0.5, 0).setStrokeStyle(3, 0xCCCCCC);
162             doorS.setInteractive({
163                 useHandCursor: true
164             }); //visual clue
165             doorS.name = "South";
166             doorS.on('pointerup', function (pointer) {
167                 console.info("DoorS clicked. ");
168                 newRoom(doorS);
169                 resetRoom();
170             }, this);
171             if ((Rooms2D.CrntRoomY == 3) || (Rooms2D.CrntRoomX == 3)) {
172                 doorS.visible = false;
173                 doorS.setInteractive(false);
174             } else {
175                 doorS.visible = true;
176                 doorS.setInteractive(true);
177             }
178 
179             doorW = this.add.rectangle(0, config.height / 2, 18, 60, 0xFFFFFF
                ).setOrigin(0, 0.5).setStrokeStyle(3, 0xCCCCCC);
180             doorW.setInteractive({
181                 useHandCursor: true
182             }); //visual clue
183             doorW.name = "West";
184             doorW.enableBody = true;
185             doorW.on('pointerup', function (pointer) {
186                 console.info("DoorW clicked. ");
187                 newRoom(doorW);
188                 resetRoom();
189             }, this);
190             if ((Rooms2D.CrntRoomX == 0) || ((Rooms2D.CrntRoomX == 2) && (
                Rooms2D.CrntRoomY == 1))) {
191                 //Left Column doesn't have Western doors
192                 doorW.visible = false;
193                 doorW.setInteractive(false);
194             } else {
195                 doorW.visible = true;
196                 doorW.setInteractive(true);
197 
198             }
199             // include doors into Doors Group
200             DoorGroup = this.physics.add.staticGroup();
201             DoorGroup.addMultiple([doorN, doorE, doorS, doorW]);
202             //debug
203             console.info(DoorGroup);
204 
205             this._toolTip = this.add.text(config.width - 105, 35, Rooms2D.
                InfoText, Rooms2D.styleTT);
206             this._toolTip.setOrigin(0.5, 0.5);
207             this._toolTip.setShadow(3, 3, 'rgba(0,0,0,0.5)', 5);
208             this._toolTip.fontWeight = 'bold';
209 
210             _RmText = "";
```

Copyright © 1997-2019, Stephen Gose. All rights reserved

```
211             _RmText = "Room  #" + Rooms2D.CrntRoom + "\nUse arrow key to move
                or\nClick on doorway.\nGrid: [" + Rooms2D.CrntRoomX + "][" +
                Rooms2D.CrntRoomY + "]\nVisible doorways only. ";
212             newHUD = _RmText;
213             //debug console.log(newHUD);
214             _RmLabel = this.add.text(config.width / 2 - 110, config.height / 2
                 - 30, _RmText, Rooms2D.styleTT);
215             _RmLabel.setOrigin(0.5, 0.5);
216             _RmLabel.setShadow(3, 3, 'rgba(0,0,0,0.5)', 5);
217             _RmLabel.fontWeight = 'bold';
218
219             // if player collides with walls or doors
220             //this.physics.world.collide(player, Room, bumpWall, null, this);
221             this.physics.add.collider(player, Room, bumpWall, null, this);
222             this.physics.add.collider(player, DoorGroup, bumpDoor, null, this);
223
224
225             //Room Exceptions
226             //Hard-coded Error corrections:
227             if (Rooms2D.CrntRoomX < 0) {
228                 Rooms2D.CrntRoomX = 0;
229             }
230             if (Rooms2D.CrntRoomX > 3) {
231                 Rooms2D.CrntRoomX = 3;
232             }
233             if (Rooms2D.CrntRoomY < 0) {
234                 Rooms2D.CrntRoomY = 0;
235             }
236             if (Rooms2D.CrntRoomY > 3) {
237                 Rooms2D.CrntRoomY = 3;
238             }
239
240         },
241
242         update: function () {
243
244             //if(this.spacebar.isDown){
245             //  game.state.start('main');
246             //}
247             // ============
248             // Example 4.3: Movement Arrows Integration begins
249             // NOTE: combination arrow directions are now
250             //    possible with this format
251             // ============
252
253             player.body.velocity.x = 0;
254             player.body.velocity.y = 0;
255
256             // Example 4.3: ends
257             // ============
258             //HUD displayed
259             this._toolTip.setText(Rooms2D.InfoText);
260
261             // Tracking info
262             _RmLabel.setText(newHUD);
263             if (cursors.left.isDown) {
264                 // if the left arrow key is down
265                 player.body.setVelocityX(-speed); // move left
266             }
```

Copyright © 1997-2019, Stephen Gose. All rights reserved

```
267             if (cursors.right.isDown) {
268                 // if the right arrow key is down
269                 player.body.setVelocityX(speed); // move right
270             }
271             if (cursors.down.isDown) {
272                 player.body.setVelocityY(speed); // jump up
273             }
274             if (cursors.up.isDown) {
275                 player.body.setVelocityY(-speed); // jump up
276             }
277         },
278 
279         //
280         //=====================================================
281         //create a box Image (pseudo graphics) for the HTML5 canvas.
282         box: function (options) {
283             //var bxImg = this.add.bitmapData(options.length,options.width);
284             var bxImg = game.add.rectangle(options.length, options.width);
285             bxImg.ctx.beginPath();
286             bxImg.ctx.rect(0, 0, options.length, options.width);
287             bxImg.ctx.fillStyle = options.color;
288             bxImg.ctx.fill();
289             return bxImg;
290         }
291 
292     });
293     //
294     // ============================================================
295     function bumpWall() {
296         player.body.velocity.x = 0;
297         player.body.velocity.y = 0;
298     };
299     //
300     // ============================================================
301     function bumpDoor(player,door) {
302         // if player move into a doorway.
303         player.body.velocity.x = 0;
304         player.body.velocity.y = 0;
305         console.log("LN 265: Bumped into Door "+door.name);
306         // which doorway? Is it visible???
307         // BUG: going through invisible doors? Why?
308         // FIX: if visible, then allow passage; otherwise, stop
309         if(door.visible){
310             newRoom(door);
311             resetRoom();
312         }
313         player.body.velocity.x = 0;
314         player.body.velocity.y = 0;
315     };
316     //
317     // ============================================================
318     //Main Door click handler
319     function newRoom(door) {
320         // 2 Options:
321         // - reset this phase with new room characteristics OR
322         // - have a "repaint" function to adjust the entered room.
323         //
            http://www.html5gamedevs.com/topic/5304-how-to-restartreload-a-state/
324         // Option 1: this.scene.restart();
```

Copyright © 1997-2019, Stephen Gose. All rights reserved

```
325         // Option 2: separation of concerns - new function
326         Rooms2D.LastRoom = Rooms2D.CrntRoom;
327         player.setPosition(64, 64);
328         var LastDoor = door.name;
329         console.log('Last Door Used: ' + door.name);
330 
331         switch (LastDoor) {
332         case "North":
333             //Rooms2D.CrntRoom -= 4; // or GRID_ROWS or MT.length
334             Rooms2D.CrntRoom = Rooms2D.LastRoom - 4;
335             Rooms2D.CrntRoomY -= 1;
336             //Leave via North; enter new room from South-side
337             Rooms2D.pPosX = config.width / 3;
338             Rooms2D.pPosY = 320;
339             break;
340         case "East":
341             Rooms2D.CrntRoom += 1;
342             Rooms2D.CrntRoomX += 1;
343             //Leave via East; Enters new room from the west-side
344             Rooms2D.pPosX = 50;
345             Rooms2D.pPosY = config.height / 2;
346 
347             break;
348         case "South":
349             Rooms2D.CrntRoom += 4; // or GRID_ROWS or MT.length
350             Rooms2D.CrntRoomY += 1;
351             //Leave via South; enter new room from North-side
352             Rooms2D.pPosX = config.width / 3;
353             Rooms2D.pPosY = 50;
354 
355             break;
356         case "West":
357             Rooms2D.CrntRoom -= 1;
358             Rooms2D.CrntRoomX -= 1;
359             //Leave via West; enters new room from east-side
360             Rooms2D.pPosX = 340;
361             Rooms2D.pPosY = config.height / 2;
362 
363             break;
364         }
365 
366         player.setPosition(Rooms2D.pPosX, Rooms2D.pPosY);
367         console.log("New Room #: " + Rooms2D.CrntRoom + "; Door Clicked: " +
        door.name);
368         /**
369         //sfx camera fadein/out
370         this.cameras.main.once('camerafadeincomplete', function (camera) {
371         camera.fadeOut(1000);
372         });
373         this.cameras.main.fadeIn(1000);
374          */
375         resetRoom();
376     };
377     //
378     // ========================================================
379     function resetRoom() {
380         //Room Exceptions
381         //Hard-coded Error corrections for 2DRooms array:
382         if (Rooms2D.CrntRoomX < 0) {
```

Copyright © 1997-2019, Stephen Gose. All rights reserved

```
383             Rooms2D.CrntRoomX = 0;
384         }
385         if (Rooms2D.CrntRoomX > 3) {
386             Rooms2D.CrntRoomX = 3;
387         }
388         if (Rooms2D.CrntRoomY < 0) {
389             Rooms2D.CrntRoomY = 0;
390         }
391         if (Rooms2D.CrntRoomY > 3) {
392             Rooms2D.CrntRoomY = 3;
393         }
394 
395         //redraw the new room; hard-code room door exceptions
396         if ((Rooms2D.CrntRoomY == 0) || (Rooms2D.CrntRoomX == 2)) {
397             doorN.visible = false;
398             doorN.setInteractive(false);
399         } else {
400             doorN.visible = true;
401             doorN.setInteractive(true);
402         }
403         if ((Rooms2D.CrntRoomX == 3) || (Rooms2D.CrntRoomX == 1 && Rooms2D.
        CrntRoomY == 1)) {
404             doorE.visible = false;
405             doorE.setInteractive(false);
406         } else {
407             doorE.visible = true;
408             doorE.setInteractive(true);
409         }
410         if ((Rooms2D.CrntRoomY == 3) || (Rooms2D.CrntRoomX == 2)) {
411             doorS.visible = false;
412             doorS.setInteractive(false);
413         } else {
414             doorS.visible = true;
415             doorS.setInteractive(true);
416         }
417         if ((Rooms2D.CrntRoomX == 0) || ((Rooms2D.CrntRoomX == 2) && (Rooms2D.
        CrntRoomY == 1))) {
418             //Left Column doesn't have Western doors
419             doorW.visible = false;
420             doorW.setInteractive(false);
421         } else {
422             doorW.visible = true;
423             doorW.setInteractive(true);
424         }
425         //update Room HUD information
426         newHUD = "Room  #" + Rooms2D.CrntRoom + "\nUse arrow key to move
        or\nClick on doorway.\nGrid: [" + Rooms2D.CrntRoomX + "][" + Rooms2D.
        CrntRoomY + "]\nVisible doorway: ";
427         //debug console.log("=====");
428         //debug console.log(newHUD);
429         //debug console.log("=====");
430     }
```

Copyright © 1997-2019, Stephen Gose. All rights reserved