

```
1  /**
2   * File Name: combat.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 combat scenes;
6   *               managing 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 //design notes: switch seems to be faster than the if statement combat.
24 //This is original "switched" version rv_8
25 "use strict";
26 window.GAMEAPP.state.Combat = {
27   init: function() {
28     // Data structures
29     //=====
30     GAMEAPP.CmbtTurn = 0;
31     console.log('Intialize Combat Turns: '+GAMEAPP.CmbtTurn)
32   },
33   preload: function() {
34     console.log("loading combat state");
35     this.game.load.image('combat',
36       'assets/images/staticRooms/combat.jpg');
37     this.game.load.atlas('avatar', 'assets/icons/arraIcons.png',
38       'assets/icons/arraIcons.json');
39     this.load.atlas('fireButton',
40       'assets/spriteSheets/mmog-sprites-silver.png',
41       'assets/spriteSheets/mmog-sprites.json');
42     this.load.atlas('attackButton',
43       'assets/spriteSheets/mmog-sprites-silver.png',
44       'assets/spriteSheets/mmog-sprites.json');
45     this.load.atlas('searchButton',
46       'assets/spriteSheets/mmog-sprites-silver.png',
47       'assets/spriteSheets/mmog-sprites.json');
48     this.load.atlas('exchangeButton',
49       'assets/spriteSheets/mmog-sprites-silver.png',
50       'assets/spriteSheets/mmog-sprites.json');
51     this.load.atlas('exitButton',
52       'assets/spriteSheets/mmog-sprites-silver.png',
53       'assets/spriteSheets/mmog-sprites.json');
54     this.load.atlas('cookButton',
55       'assets/spriteSheets/mmog-sprites-silver.png',
56       'assets/spriteSheets/mmog-sprites.json');
57     this.load.spritesheet('Death',
```

```
44     'assets/spriteSheets/ARRADeath/ARRADeath.png', 160, 100);
45     this.load.spritesheet('button',
46     'assets/spriteSheets/mmog-sprites-silver.png', 129, 30);
47     //GAMEAPP.LastRoom = GAMEAPP.CrntRoom;
48
49     },
50
51     create: function() {
52         console.log("starting combat state");
53         var MonsterNdx = MT[GAMEAPP.CrntRoom].Mnstr;
54         var mGold = Number(M[MT[GAMEAPP.CrntRoom].Mnstr].HGold);
55         var mGem = Number(M[MT[GAMEAPP.CrntRoom].Mnstr].HGem);
56         var mFood = Number(M[MT[GAMEAPP.CrntRoom].Mnstr].Food);
57         console.log("Monster #: " + MonsterNdx);
58         console.log("Silver found: " + mGold);
59         console.log("Gems found: " + mGem);
60         console.log("Food found: " + mFood);
61         this.game.add.image(0, 0, 'combat');
62         var rect = new Phaser.Rectangle(90, 110, 196, 155);
63         this.game.physics.arcade.setBounds(90, 110, 196, 155);
64         //Set a neutral background color
65         this.game.stage.backgroundColor = "#000";
66         //Set game to ARCADE physics systemLanguage
67         this.game.physics.startSystem(Phaser.Physics.ARCADE);
68         this.game.renderer.renderSession.roundPixels = true;
69         this.game.world.enableBody = true;
70
71         //Create Room
72         this.Room = this.game.add.group();
73         this.Room.enableBody = true;
74         this.game.physics.enable(this.Room, Phaser.Physics.ARCADE);
75
76         this.NorthWall = this.game.add.sprite(90, 110, box({length: 190, width:
77         : 16, color: '#999'}));
78         this.NorthWall.enableBody = true;
79         this.NorthWall.body.immovable = true;
80         this.NorthWall.visible = false;
81         this.Room.add(this.NorthWall);
82         this.NorthWall.alignTo(rect, Phaser.TOP_CENTER);
83
84         this.SouthWall = this.game.add.sprite(0, 200, box({length: 190, width:
85         : 16, color: '#999'}));
86         this.SouthWall.body.immovable = true;
87         this.SouthWall.visible = false;
88         this.Room.add(this.SouthWall);
89         this.SouthWall.alignTo(rect, Phaser.BOTTOM_CENTER);
90
91         this.WestWall = this.game.add.sprite(200, 16, box({length: 16, width:
92         : 190, color: '#999'}));
93         this.WestWall.body.immovable = true;
94         this.WestWall.visible = false;
95         this.Room.add(this.WestWall);
96         this.WestWall.alignTo(rect, Phaser.RIGHT_CENTER);
97
98         this.EastWall = this.game.add.sprite(90, 16, box({length: 16, width:
99         : 190, color: '#999'}));
100        this.EastWall.body.immovable = true;
101        this.EastWall.visible = false;
102        this.Room.add(this.EastWall);
```

```

97         this.EastWall.alignTo(rect, Phaser.LEFT_CENTER);
98
99         //toolTip (tt) and RoomAlert (ra) text
100        this._toolTip = this.game.add.text(this.game.world.width-100, this
        .game.world.height-90, GAMEAPP.InfoText, GAMEAPP.styleTT);
101        this._toolTip.anchor.set(0.5);
102        GAMEAPP.RoomAlert = "";
103
104        //toolTip (tt) and RoomAlert (ra) text
105        GAMEAPP.RoomAlert = "Beware, a " + M[MT[GAMEAPP.LastRoom].Mnstr].
        Race + " is here!";
106        this._raTxt = this.game.add.text(this.game.world.width-268, 65,
        GAMEAPP.RoomAlert, GAMEAPP.styleRA);
107        this._raTxt.anchor.set(0,0);
108        var attacktxt = this.game.add.text(0, 0, "Attack" , GAMEAPP.
        styleBTN); // "Attack" button text;
109        var firetxt = this.game.add.text(0, 0, "Fire" , GAMEAPP.styleBTN
        ); // "Fire" button text;
110        var exchangetxt = this.game.add.text(0, 0, "Exchange" , GAMEAPP.
        styleBTN);
111        var exittxt = this.game.add.text(0, 0, "Return" , GAMEAPP.styleBTN
        );
112        var searchtxt = this.game.add.text(0, 0, "Search" , GAMEAPP.
        styleBTN);
113        var dodgetxt = this.game.add.text(0, 0, "Dodge" , GAMEAPP.styleBTN
        );
114        var defendtxt = this.game.add.text(0, 0, "Defend" , GAMEAPP.
        styleBTN);
115
116        //Combat narrative HUD
117        var CmbtTurn = 1;
118        this.CTurnNum = this.game.add.text(15, 93, "Combat Turn: #" +String
        (CmbtTurn) , GAMEAPP.styleTT);
119        //combat narrative for player
120        this.Narr1txt = this.game.add.text(35, 265, "xx" , GAMEAPP.
        styleNarrH);
121        //combat narrative for monster
122        this.Narr2txt = this.game.add.text(this.game.world.width-350, 265,
        "xx" , GAMEAPP.styleNarrM);
123
124        //Character HUD Display
125        this.cStmnatxt = this.game.add.text(this.game.world.width-125, 120
        , String(Person[1].ModStmn) , GAMEAPP.styleCmbtH);
126        var cWS = ((CCP.WSRaw * 5) + (CCP.Stmn * 2)) + "%";
127        this.csWStxt = this.game.add.text(this.game.world.width-125, 136,
        String(cWS) , GAMEAPP.styleCmbtH);
128        var cPS = ((CCP.BSRaw * 5) + (CCP.Coor * 2)) + "%";
129        this.csPStxt = this.game.add.text(this.game.world.width-125, 152,
        String(cPS) , GAMEAPP.styleCmbtH);
130        this.csWpntxt = this.game.add.text(this.game.world.width-125, 168,
        String(CCP.WName) , GAMEAPP.styleCmbtH);
131        this.csArmtxt = this.game.add.text(this.game.world.width-125, 185,
        String(CCP.AName) , GAMEAPP.styleCmbtH);
132
133        this.cExptxt = this.game.add.text(this.game.world.width-45, 103,
        String(Person[1].TempScore) , GAMEAPP.styleCmbtH);
134        this.cRenowntxt = this.game.add.text(this.game.world.width-45, 120
        , String(Person[1].Renown) , GAMEAPP.styleCmbtH);
135        this.cGoldtxt = this.game.add.text(this.game.world.width-45, 136,

```

```

String(CCP.HGold) , GAMEAPP.styleCmbtH);
136 this.cFoodtxt = this.game.add.text(this.game.world.width-45, 152,
String(CCP.Food) , GAMEAPP.styleCmbtH);

137
138 //Monster HUD Display
139 this.mStmnatxt = this.game.add.text(this.game.world.width-125, 231
, String(Person[6].ModStmn) , GAMEAPP.styleCmbtM);
140 this.msWStxt = this.game.add.text(this.game.world.width-125, 248,
String(Person[6].WS)+"%" , GAMEAPP.styleCmbtM);
141 this.msPStxt = this.game.add.text(this.game.world.width-125, 265,
String(Person[6].PS)+"%" , GAMEAPP.styleCmbtM);
142 this.msWpntxt = this.game.add.text(this.game.world.width-125, 281,
String(M[MT[GAMEAPP.CrntRoom].Mnstr].Weapn) , GAMEAPP.styleCmbtM);
143 this.msArmtxt = this.game.add.text(this.game.world.width-125, 298,
String(M[MT[GAMEAPP.CrntRoom].Mnstr].Armor) , GAMEAPP.styleCmbtM);
144
145 this.mRenowntxt = this.game.add.text(this.game.world.width-44, 231
, String(Person[6].Renown) , GAMEAPP.styleCmbtM);
146 this.mGoldtxt = this.game.add.text(this.game.world.width-44, 248,
String(M[MT[GAMEAPP.CrntRoom].Mnstr].HGold) , GAMEAPP.styleCmbtM);
147 this.mFoodtxt = this.game.add.text(this.game.world.width-44, 264,
String(M[MT[GAMEAPP.CrntRoom].Mnstr].Food) , GAMEAPP.styleCmbtM);
148
149 //ARRA Character Death animation
150 this.ARRADeath = this.game.add.sprite(1600, 212, 'Death');
151
152 // Here we add a new animation called 'show Death'
153 // Because we didn't give any other parameters it's going to
make an animation from all available frames in the ARRA Death
sprite sheet
154 var showDeath = this.ARRADeath.animations.add('showDeath');
155
156 // And this starts the animation playing by using its key
('showDeath')
157 // 30 is the frame rate (30fps)
158 // true means it will loop when it finishes
159 this.ARRADeath.animations.play('showDeath', 8, true);
160
161 // Attack button deployed off screen
162 this.attackButton = this.game.add.button(this.world.centerX-800,
338, 'button', this._combatRound, this, 2, 1, 0,1);
163 this.attackButton.anchor.set(0.5,0.5);
164 this.attackButton.scale.setTo(0.5,0.5);
165 this.attackButton.addChild(attacktxt).anchor.set(0.5,0.5);
166
167 // Fire button deployed visible only if armed with Missile weapon.
168 this.fireButton = this.add.button(152, 338, 'button', this.
_combatRound, this, 2, 1, 0,1);
169 this.fireButton.anchor.set(0.5,0.5);
170 this.fireButton.scale.setTo(0.5,0.5);
171 this.fireButton.addChild(firetxt).anchor.set(0.5,0.5);
172
173 // Exchange weapons button
174 this.exchangeButton = this.game.add.button(this.world.centerX-800,
361, 'button', this._Exchange, this, 2, 1, 0,1);
175 this.exchangeButton.anchor.set(0.5,0.5);
176 this.exchangeButton.scale.setTo(0.5,0.5);
177 this.exchangeButton.addChild(exchanetxt).anchor.set(0.5,0.5);
178

```

```

179      // Exit button
180      this.exitButton = this.game.add.button(65, 338, 'button', this.
        _GameReturn, this, 2, 1, 0,1);
181      this.exitButton.anchor.set(0.5,0.5);
182      this.exitButton.scale.setTo(0.5,0.5);
183      this.exitButton.addChild(exittxt).anchor.set(0.5,0.5);
184
185      // Search button
186      this.searchButton = this.game.add.button(this.world.centerX-800,
        338, 'button', this._Search, this, 2, 1, 0,1);
187      this.searchButton.anchor.set(0.5,0.5);
188      this.searchButton.scale.setTo(0.5,0.5);
189      this.searchButton.addChild(searchtxt).anchor.set(0.5,0.5);
190
191      // Dodge button - available during missile combat only
192      this.dodgeButton = this.game.add.button(this.world.centerX-800,
        338, 'button', this._Cook, this, 2, 1, 0,1);
193      this.dodgeButton.anchor.set(0.5,0.5);
194      this.dodgeButton.scale.setTo(0.5,0.5);
195      this.dodgeButton.addChild(dodgetxt).anchor.set(0.5,0.5);
196
197      // Defend button - available during melee & hand-to-hand combat
        only
198      this.defendButton = this.game.add.button(this.world.centerX-800,
        338, 'button', this._Cook, this, 2, 1, 0,1);
199      this.defendButton.anchor.set(0.5,0.5);
200      this.defendButton.scale.setTo(0.5,0.5);
201      this.defendButton.addChild(defendtxt).anchor.set(0.5,0.5);
202
203      //enemy character
204      this.enemy = this.game.add.sprite(MT[GAMEAPP.LastRoom].mxPos,MT[
        GAMEAPP.LastRoom].myPos,'avatar');
205      this.enemy.frameName = String(M[MT[GAMEAPP.LastRoom].Mnstr].icon+
        '.bmp');
206      this.enemy.body.collideWorldBounds = true;
207      this.game.physics.enable(this.enemy, Phaser.Physics.ARCADE);
208
209      this.enemy.alignIn(rect, Phaser.BOTTOM_RIGHT);
210
211      GAMEAPP._PUpdate();
212
213      //place character
214      this.player = this.game.add.sprite(90,230,'avatar');
215      this.player.frameName = 'avatar.bmp';
216      this.player.body.collideWorldBounds = true;
217      this.cursor = this.game.input.keyboard.createCursorKeys();
218
219      this.player.alignIn(rect, Phaser.TOP_LEFT);
220
221  },
222
223  update: function(){
224      this.game.physics.arcade.collide(this.player, this.Room);
225      this.game.physics.arcade.collide(this.enemy, this.Room);
226
227      //Not engaged in melee; if player is dead; show it, and stop the
        combat
228      if(Person[1].ModStmn <= 0){
229          this.player.tint = 0xFF0000;

```

```

230     GAMEAPP.RoomAlert = "Defeated ?!";
231     CCP.ModStmn = Person[1].ModStmn;
232     this._raTxt.setText(String(GAMEAPP.RoomAlert));
233     this.attackButton.x = this.world.centerX+800;
234     this.fireButton.x = this.world.centerX+800;
235
236     //Show ARRA Death animation at this.game.world.width-178, 100
237     this.ARRADeath.x = this.game.world.width-178;
238     this.ARRADeath.y = 100;
239     this.ARRADeath.animations.play('showDeath', 8, true);
240
241     }else{
242         this.attackButton.x = this.world.centerX-800;
243         this.defendButton.x = this.world.centerX-800;
244         this.fireButton.x = 240;
245         this.dodgeButton.x = 152;
246         GAMEAPP._MoveChar(this.cursor,this.player);
247     }
248     //if monster dead; show it and stop the combat
249     if(Person[6].ModStmn <= 0){
250         this.enemy.tint = 0xFF0000;
251         //GAMEAPP.RoomAlert = "Victory over the " +
252         M[MT[GAMEAPP.LastRoom].Mnstr].Race + "!";
253         this._raTxt.setText("Victory !");
254         MT[GAMEAPP.CrntRoom].Mnstr = 0;
255         this.attackButton.x = this.world.centerX+800;
256         this.fireButton.x = this.world.centerX+800;
257
258         this.ARRADeath.x = this.game.world.width-178;
259         this.ARRADeath.y = 212;
260         this.ARRADeath.animations.play('showDeath', 8, true);
261
262         //'Killing Blow delivered
263         GAMEAPP.InfoText = "Monster was slain\n... looking in its
264         backpack.\nYou savage all food, gold & gems ... ";
265         CCP.ModStmn = Number(Person[1].ModStmn);
266         CCP.Score += Number(Person[1].TempScore) + Number(Person[6].
267         Coord);
268         Person[1].TempScore = 0;
269         Person[6].Coord = 0;
270         CCP.HGold += Person[6].Gold;
271         Person[6].Gold = 0;
272         CCP.HGem += Person[6].Gem;
273         Person[6].Gem = 0;
274         CCP.Food += Person[6].Food;
275         Person[6].Food = 0;
276         CCP.Reown = Math.ceil((Number(Person[1].Reown)) + Number(
277         Number(Person[6].Reown)/2));
278
279     }else{
280         GAMEAPP._MoveMonster(this.cursor,this.enemy);
281     }
282     this.Narr1txt.setText(String(GAMEAPP.Narr1));
283     this.Narr2txt.setText(String(GAMEAPP.Narr2));
284     this.cStmnatxt.setText(String(Person[1].ModStmn));
285     this.cExptxt.setText(String(Person[1].TempScore));
286     this.mStmnatxt.setText(String(Person[6].ModStmn));
287     this._toolTip.setText(String(GAMEAPP.InfoText));
288     this.game.physics.arcade.overlap(this.player,this.enemy,

```

```

        meleeCombat, null, this);

285
286     },
287     //
288     // =====
289     // Subroutines & Functions - ARRA Combat Module.
290     // =====
291     // Combat Finite State machine - see AI chapter in the book.
292     // Hardcoded percentages according to ARRA game rules.
293     // Refer to FREE Rule book in file downloaded.
294     _CombatSkill: function(EventRoll, TestValue){
295         var results = 20;
296         var Critical = 0;
297         Critical = Math.ceil(TestValue/10)
298
299         if (EventRoll <= Critical){           //Player did a critical
300             results = 10;
301             return results;                   //Return a 10
302         }else if (EventRoll <= TestValue){    //This is a normal hit
303             results = 1;
304             return results;
305         }else if (EventRoll > 94){
306             //Then Player's Fumbled, This is a new
307             //circumstance to include - RQ fubbling. BUT For now,
308             results = 0;
309             return results;
310         }else if (EventRoll > TestValue){    //Player's Missed
311             results = 0;
312             return results;
313         }
314
315     },
316     //
317     // =====
318     _CombatNarrative: function(Who,T){
319
320         var ModTtlAP = Person[T].TotalAP;
321         var AtkWord = "";
322         var PryWord = "";
323         var Narrative = "";
324         var Trash = 1;
325         var Ndx = 0;
326
327         console.log("Who: "+Who+"; Target: "+T)
328
329         //Random Adjectives
330         var AtkWord = "";
331         var PryWord = "";
332         var Ndx = 0;
333         var Atk = 0;
334         var Pry = 0;
335         var Hit = 0;
336         var DamRoll = 0;
337         var ModTtlAP = 2;
338         var TxtTkn = "";
339         var TxtTkn2 = "";
340         var AtkRoll = 101;
341         var PryRoll = 101;
342         var DamRoll = 0;

```

```
343
344     var ThisAdjWord = ["", "", ""];
345
346     var AdjWord = ["poor ", "desperate ", "smooth ", "skilled ", "clever ",
    "quick ", "strong ", "wicked ", "savage ", "firm ", "ragged ", "heavy ",
    "poor ", "desperate ", ""];
347
348     //Random Critical Adjectives
349     var CAdj = ["BRUTAL ", "BRUTAL ", "MORTAL ", "GAPING ", "SEVERE ",
    "CRITICAL ", "GRAVE ", "CRITICAL "];
350
351     //Random Attack Verbs
352     var RAtkVrb = ["slash ", "slash ", "chop ", "slice ", "thrust ", "cut ",
    "hack ", "chop "];
353
354     //Random Parry Words
355     var RPryVrb = ["block ", "block ", "parry ", "block ", "redirect ", "parry
    ", "counter ", "redirect "];
356
357     //Initialize Variables
358
359     Trash = Math.floor((Math.random() * 6 + 1));
360     AtkWord = RAtkVrb[Trash];
361
362
363     Trash = Math.floor((Math.random() * 6 + 1));
364     PryWord = RPryVrb[Trash];
365
366     for (var i=1 ; i < 2 ; i++){
367
368         Ndx = Math.floor((Math.random() * 12 + 1));
369         ThisAdjWord[i] = AdjWord[Ndx];
370     }
371
372     //Start on this person
373     //Get Atk & Parry rolls
374     AtkRoll = Math.floor((Math.random() * 99 + 1));
375
376     //CombatSkill Function return 0,1 or 10 given person AtkRoll and
    Weapon proficiency.
377     Atk = this._CombatSkill(AtkRoll, Person[Who].WS)
378     console.log("WHO: "+Who);
379     console.log("AtkRoll: "+AtkRoll);
380     if(Atk > 0){
381         //then discover amount of damage
382         //Function WeaponDamage returns final resulting damage given
    person's #.
383         //fixed at 2D6
384         //DamRoll = this._GetWpnDamage(2);
385         for(var i=1; i<=2 ;i++){
386             DamRoll += Math.floor((Math.random() * 6 + 1));
387             if (DamRoll<1) {
388                 DamRoll = 1;
389             }
390         }
391
392         console.log("Damage Done: "+DamRoll);
393         if(Atk == 10){
394             Ndx = Math.floor((Math.random() * 6 + 1));
```



```

395         ThisAdjWord[1] = CAdj[Ndx];
396     }
397     }else{
398         console.log("Damage Done: NONE");
399     }
400     //This is the Opponent's Roll
401     PryRoll = Math.floor((Math.random() * 99 + 1));
402
403     //Perform parry for target person
404     Pry = this.CombatSkill(PryRoll, Person[T].PS)
405     console.log("Defender's PS: "+Person[T].PS);
406     console.log("PryRoll: "+PryRoll);
407     if(Pry == 10){
408         Ndx = Math.floor((Math.random() * 6 + 1));
409         ThisAdjWord[2] = CAdj[Ndx];
410         ModTtlAP = (Person[T].TotalAP * 4);
411     }
412
413     //Now Combat Calculations for this Person
414     Trash = 0;
415     Hit = Atk - Pry;
416     console.log("Atk: "+ Atk);
417     console.log("Pry: "+ Pry);
418     console.log("Hit: "+ Hit);
419     console.log("=====");
420
421     //Generate Combat narrative
422     if (Who == 1){
423         TxtTkn = "    You: ";           //Player
424         TxtTkn2 = "Monster: ";         //Monster
425     }else{
426         TxtTkn = "Monster: ";           //Monster's Text Color
427         TxtTkn2 = "    You: ";         //Player's Text Color
428     }
429
430
431     //NEW Switch statement for Combat
432     switch(Hit){
433         case 0:
434             //attack & parry cancell each other
435             Narrative = TxtTkn + ' a ' + ThisAdjWord[1] + AtkWord + '\n';
436             Narrative += TxtTkn2 + ' a ' + ThisAdjWord[2] + PryWord + '\n';
437             Narrative += TxtTkn2 + ' not injured!';
438             break;
439
440         case 1:
441             Trash = DamRoll - ModTtlAP;
442             if (Trash < 0) {
443                 Trash = 0;
444             }
445             Narrative = TxtTkn + ' a ' + ThisAdjWord[1] + AtkWord + '\n';
446             Narrative += TxtTkn2 + ' a failed ' + PryWord + '\n';
447             Narrative += TxtTkn2 + ' wounded! [' + Trash + ' pts]';
448             Person[T].ModStmn -= Number(Trash)
449             console.log("Target: "+T+" stmn = "+Person[T].ModStmn);
450             //this.mStmnatxt.setText(String(Person[6].ModStmn));
451             //this.cStmnatxt.setText(String(Person[1].ModStmn));
452
453             if (Who == 1){

```

```
454         Person[1].TempScore += Number(Trash)
455         // this.cStmnatxt.setText(String(Person[1].TempScore));
456     }
457     break;
458
459     case -1:
460     Narrative = TxtTkn + ' a failed ' + AtkWord + '\n';
461     Narrative += TxtTkn2+" a "+AdjWord[2]+PryWord+" awaits.";
462     break;
463
464     case 9:
465     Trash = ((DamRoll * 2) - ModTtlAP);
466     if (Trash < 0){
467         Trash = 0;
468     }
469
470     Narrative = TxtTkn+" a "+ThisAdjWord[1]+AtkWord+"\n";
471     Narrative += TxtTkn2+" a "+PryWord+" was too late!\n";
472     Narrative += TxtTkn2+" wounded! ["+String(Trash)+" pts]";
473
474     Person[T].ModStmn -= Number(Trash);
475     console.log("Target "+T+" stmn = "+Person[T].ModStmn);
476     // this.mStmnatxt.setText(String(Person[6].ModStmn));
477     //this.cStmnatxt.setText(String(Person[1].ModStmn));
478
479     if (Who == 1){
480         Person[1].TempScore += Number(Trash);
481         //this.cStmnatxt.setText(String(Person[1].TempScore));
482     }
483     break;
484
485     case -9:
486     Trash = (DamRoll - (ModTtlAP * 2));
487     if (Trash < 0){
488         Trash = 0;
489     }
490
491     Narrative = TxtTkn+" a "+ThisAdjWord[1]+AtkWord+"\n";
492     Narrative += TxtTkn2+" but a "+ThisAdjWord[2]+PryWord+"
over-powered!\n";
493     Narrative += TxtTkn2+" wounded? ["+String(Trash)+" pts]";
494
495     Person[T].ModStmn -= Number(Trash);
496     console.log("Target "+T+" stmn = "+Person[T].ModStmn);
497     // this.mStmnatxt.setText(String(Person[6].ModStmn));
498     //this.cStmnatxt.setText(String(Person[1].ModStmn));
499
500     if (Who == 1){
501         Person[1].TempScore += Number(Trash)
502         //this.cStmnatxt.setText(String(Person[1].TempScore));
503     }
504
505     break;
506
507     case 10:
508     Trash = ((DamRoll * 2) - ModTtlAP);
509     if (Trash < 0){
510         Trash = 0;
511     }
```

```

512
513     Narrative = TxtTkn+" a "+ThisAdjWord[1]+" "+AtkWord+"\n";
514     Narrative += TxtTkn2+" block entirely missed!\n";
515     Narrative += TxtTkn2+" wounded ["+String(Trash)+" pts]";
516
517     Person[T].ModStmn -= Number(Trash);
518     console.log("Target "+T+" stmn = "+Person[T].ModStmn);
519     //this.mStmnatxt.setText(String(Person[6].ModStmn));
520     //this.cStmnatxt.setText(String(Person[1].ModStmn));
521
522     if (Who == 1){
523         Person[1].TempScore += Number(Trash)
524         //this.cStmnatxt.setText(String(Person[1].TempScore));
525     }
526     break;
527
528     case -10:
529     Narrative = TxtTkn + ' a missed ' + AtkWord + '\n';
530     Narrative += TxtTkn2+" a "+AdjWord[2]+PryWord+" awaits.";
531     break;
532
533 }
534
535 // return Narrative;
536
537 if(Who == 1){
538     GAMEAPP.Narr1 = String(Narrative);
539
540 }
541 if(Who == 6){
542     GAMEAPP.Narr2 = String(Narrative);
543
544 }
545
546 },
547 //
548 //=====
549 _combatRound: function(game){
550
551     GAMEAPP.CmbtTurn += 1;
552     console.log('combatRound '+GAMEAPP.CmbtTurn);
553     this.CTurnNum.setText( "Combat Turn: #"+String(GAMEAPP.CmbtTurn));
554     //Determine combat round initiative
555     var CmbtInit = Math.floor((Math.random() * 6 + 1));
556
557     //Reset each combat variables
558     var CombatTurnToggle = 0;
559
560     //Who goes first?
561     if (CmbtInit <= 3){
562         GAMEAPP.InfoText = "You've gained the initiative.";
563         this._toolTip.setText(GAMEAPP.InfoText);
564         this._CombatNarrative(1,6);
565         this.Narr1txt.setText(String(GAMEAPP.Narr1));
566         this._CombatNarrative(6,1);
567         this.Narr2txt.setText(String(GAMEAPP.Narr2));
568
569     } else {
570         GAMEAPP.InfoText = "Monster has the combat \n initiative this

```

```

turn.";
571     this._toolTip.setText(GAMEAPP.InfoText);
572     this._CombatNarrative(6,1);
573     this.Narr2txt.setText(String(GAMEAPP.Narr2));
574     this._CombatNarrative(1,6);
575     this.Narr1txt.setText(String(GAMEAPP.Narr1));
576
577 }
578
579 },
580
581 //
582 // =====
583 _Defend: function(){
584     if (Person[6].ModStmn > 0) {
585         GAMEAPP.InfoText = " You focus on defense!\n ... no attack
586         possible ... ";
587     }
588     if (Person[1].ModStmn < 1) {
589         this._GameReturn;
590     }
591 },
592 //
593 // =====
594 _Exchange: function(){
595     if (Person[6].ModStmn > 0) {
596         GAMEAPP.InfoText = "Changing Weapons ...\n You miss your
597         attack and defense?";
598     }
599     if (Person[1].ModStmn < 1) {
600         this._GameReturn;
601     }
602 },
603 //
604 // =====
605 // Return from Combat session
606 _GameReturn: function(){
607     //debug
608     if(Person[6].ModStmn <= 0){
609         console.log("Room #: "+GAMEAPP.CrntRoom+"; Exit combat as
610         victor. ");
611     }else{
612         console.log("Room #: "+GAMEAPP.CrntRoom+"; Exit combat as
613         coward. ");
614         // Ran away while monster was still in the room.
615         CCP.Renown = Person[1].Renown-2;
616         Person[1].Renown = CCP.Renown;
617         GAMEAPP.InfoText = "Running from Combat? \n Renown is "+String
618         (CCP.Renown);
619     }
620     if(Person[1].ModStmn > 0){
621         this.camera.fade(0x000000, 200, false);
622         this.time.events.add(200, function() {
623             //Return to Room Scene rv_3 through rv_8
624             this.game.state.start(MT[GAMEAPP.CrntRoom].RmID);
625         }, this);
626     }else{
627         this.camera.fade(0x000000, 200, false);

```

```

624         this.time.events.add(200, function() {
625             //exitGame defeated Room Scene rv_3 through rv_8
626             this.game.state.start('exitGame');
627         }, this);
628     }
629 },
630
631 //
632 // =====
633 // Prep Frame 3
634 _GetWpnDamage: function (Dice) {
635     var TempT = 0;
636     //Rolls Multiple 6-sided dice
637     for(var i=1; i<Dice ;i++){
638         TempT += Math.floor((Math.random() * 6 + 1));
639     }
640     if (TempT<1) {
641         TempT = 0;
642     }
643     return TempT;
644 },
645
646 //
647 // =====
648 _Search: function(){
649     if (Person[6].ModStmn > 0) {
650         GAMEAPP.InfoText = "WHAT?! NOW ...\n in the middle of Combat?";
651     }
652
653     if(Person[6].ModStmn < 1){
654         GAMEAPP.InfoText = "... looking in its backpack.\nYou savage
all food, gold & gems ... ";
655         CCP.HGold += Person[6].Gold;
656         CCP.HGem += Person[6].Gem;
657         CCP.Food += Person[6].Food;
658         this._GameReturn;
659     }
660
661     if (Person[1].ModStmn < 1) {
662         this._GameReturn;
663     }
664 }
665 //
666 // =====
667 };
668 //
669 // =====
670 //create a box Image (pseudo graphics) for the HTML5 canvas.
671 var box = function(options) {
672     var bxImg = GAMEAPP.game.add.bitmapData(options.length,options.width);
673
674     bxImg.ctx.beginPath();
675     bxImg.ctx.rect(0,0,options.length,options.width);
676     bxImg.ctx.fillStyle = options.color;
677     bxImg.ctx.fill();
678     return bxImg;
679 };
680 //
681 // =====

```

```
681 //melee combat: Dynamic Menu; engaged in melee
682 var meleeCombat = function(player,enemy){
683     this.attackButton.x = 152;
684     this.defendButton.x = 240;
685     this.fireButton.x = this.world.centerX+800;
686     this.dodgeButton.x = this.world.centerX+800;
687     if(Person[6].ModStmn <= 0){
688         this.enemy.tint = 0xFF0000;
689         //GAMEAPP.RoomAlert = "Victory over the " +
690         M[MT[GAMEAPP.LastRoom].Mnstr].Race + "!"
691         this.attackButton.x = this.world.centerX+800;
692         this.fireButton.x = this.world.centerX+800;
693         this.defendButton.x = this.world.centerX+800;
694         this.dodgeButton.x = this.world.centerX+800;
695         //this.cookButton.x = 65;
696         //this.inventoryButton.x = 152;
697         this.searchButton.x = 240;
698     }
699
700     if(Person[1].ModStmn <= 0){
701         this._GameReturn;
702     }
703 }
704 };
705 //
706 // =====
707
708 /* End of file */
709 /* Location: ./js/state/combat.js */
710
711
```