

```
1  /**
2   * File Name: combat-ifStatements-Original.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 "if statements" version 1 through 3
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('inventoryButton',
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 inventorytxt = this.game.add.text(0, 0, "Inventory" , 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 cooktxt = this.game.add.text(0, 0, "Cook" , GAMEAPP.styleBTN
        );
114
115        //Combat narrative HUD
116        var CmbtTurn = 1;
117        this.CTurnNum = this.game.add.text(15, 93, "Combat Turn: #"+String
        (CmbtTurn) , GAMEAPP.styleTT);
118        //combat narrative for player
119        this.Narr1txt = this.game.add.text(35, 265, "xx" , GAMEAPP.
        styleNarrH);
120        //combat narrative for monster
121        this.Narr2txt = this.game.add.text(this.game.world.width-350, 265,
        "xx" , GAMEAPP.styleNarrM);
122
123        //Character HUD Display
124        this.cStmnatxt = this.game.add.text(this.game.world.width-125, 120
        , String(Person[1].ModStmn) , GAMEAPP.styleCmbtH);
125        var cWS = ((CCP.WSRaw * 5) + (CCP.Stmn * 2)) + "%";
126        this.csWStxt = this.game.add.text(this.game.world.width-125, 136,
        String(cWS) , GAMEAPP.styleCmbtH);
127        var cPS = ((CCP.BSRaw * 5) + (CCP.Coar * 2)) + "%";
128        this.csPStxt = this.game.add.text(this.game.world.width-125, 152,
        String(cPS) , GAMEAPP.styleCmbtH);
129        this.csWpntxt = this.game.add.text(this.game.world.width-125, 168,
        String(CCP.WName) , GAMEAPP.styleCmbtH);
130        this.csArmtxt = this.game.add.text(this.game.world.width-125, 185,
        String(CCP.AName) , GAMEAPP.styleCmbtH);
131
132        this.cExptxt = this.game.add.text(this.game.world.width-45, 103,
        String(Person[1].TempScore) , GAMEAPP.styleCmbtH);
133        this.cRenowntxt = this.game.add.text(this.game.world.width-45, 120
        , String(Person[1].Renown) , GAMEAPP.styleCmbtH);
134        this.cGoldtxt = this.game.add.text(this.game.world.width-45, 136,
        String(CCP.HGold) , GAMEAPP.styleCmbtH);
135        this.cFoodtxt = this.game.add.text(this.game.world.width-45, 152,

```

```
String(CCP.Food) , GAMEAPP.styleCmbtH);

136
137 //Monster HUD Display
138 this.mStmnatxt = this.game.add.text(this.game.world.width-125, 231
, String(Person[6].ModStmn) , GAMEAPP.styleCmbtM);
139 this.msWStxt = this.game.add.text(this.game.world.width-125, 248,
String(Person[6].WS)+"%" , GAMEAPP.styleCmbtM);
140 this.msPStxt = this.game.add.text(this.game.world.width-125, 265,
String(Person[6].PS)+"%" , GAMEAPP.styleCmbtM);
141 this.msWpntxt = this.game.add.text(this.game.world.width-125, 281,
String(M[MT[GAMEAPP.CrntRoom].Mnstr].Weapn) , GAMEAPP.styleCmbtM);
142 this.msArmtxt = this.game.add.text(this.game.world.width-125, 298,
String(M[MT[GAMEAPP.CrntRoom].Mnstr].Armor) , GAMEAPP.styleCmbtM);
143
144 this.mRenowntxt = this.game.add.text(this.game.world.width-44, 231
, String(Person[6].Renown) , GAMEAPP.styleCmbtM);
145 this.mGoldtxt = this.game.add.text(this.game.world.width-44, 248,
String(M[MT[GAMEAPP.CrntRoom].Mnstr].HGold) , GAMEAPP.styleCmbtM);
146 this.mFoodtxt = this.game.add.text(this.game.world.width-44, 264,
String(M[MT[GAMEAPP.CrntRoom].Mnstr].Food) , GAMEAPP.styleCmbtM);
147
148 //ARRA Character Death animation
149 this.ARRADeath = this.game.add.sprite(1600, 212, 'Death');
150
151 // Here we add a new animation called 'show Death'
152 // 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
153 var showDeath = this.ARRADeath.animations.add('showDeath');
154
155 // And this starts the animation playing by using its key
("showDeath")
156 // 30 is the frame rate (30fps)
157 // true means it will loop when it finishes
158 this.ARRADeath.animations.play('showDeath', 8, true);
159
160 // Attack button deployed off screen
161 this.attackButton = this.game.add.button(this.world.centerX-800,
this.game.world.height-100, 'button', this._combatRound, this, 2,
1, 0,1);
162 this.attackButton.anchor.set(0.5,0.5);
163 this.attackButton.scale.setTo(0.5,0.5);
164 this.attackButton.addChild(attacktxt).anchor.set(0.5,0.5);
165
166 // Fire button deployed visible only if armed with Missile weapon.
167 this.fireButton = this.add.button(this.world.centerX-60, this.game
.world.height-80, 'button', this._combatRound, this, 2, 1, 0,1);
168 this.fireButton.anchor.set(0.5,0.5);
169 this.fireButton.scale.setTo(0.5,0.5);
170 this.fireButton.addChild(firetxt).anchor.set(0.5,0.5);
171
172 // Inventory button
173 this.inventoryButton = this.game.add.button(this.world.centerX-220
, this.game.world.height-100, 'button', this._Inventory, this, 2,
1, 0,1);
174 this.inventoryButton.anchor.set(0.5,0.5);
175 this.inventoryButton.scale.setTo(0.5,0.5);
176 this.inventoryButton.addChild(inventorytxt).anchor.set(0.5,0.5);
177
```

```
178 // Exit button
179 this.exitButton = this.game.add.button(this.world.centerX-220,
this.game.world.height-80, 'button', this._GameReturn, this, 2, 1,
0,1);
180 this.exitButton.anchor.set(0.5,0.5);
181 this.exitButton.scale.setTo(0.5,0.5);
182 this.exitButton.addChild(exittxt).anchor.set(0.5,0.5);
183
184 // Search button
185 this.searchButton = this.game.add.button(this.world.centerX-140,
this.game.world.height-100, 'button', this._Search, this, 2, 1, 0,
1);
186 this.searchButton.anchor.set(0.5,0.5);
187 this.searchButton.scale.setTo(0.5,0.5);
188 this.searchButton.addChild(searchtxt).anchor.set(0.5,0.5);
189
190 // Cook button
191 this.cookButton = this.game.add.button(this.world.centerX-140,
this.game.world.height-80, 'button', this._Cook, this, 2, 1, 0,1);
192 this.cookButton.anchor.set(0.5,0.5);
193 this.cookButton.scale.setTo(0.5,0.5);
194 this.cookButton.addChild(cooktxt).anchor.set(0.5,0.5);
195
196 //enemy character
197 this.enemy = this.game.add.sprite(MT[GAMEAPP.LastRoom].mxPos,MT[
GAMEAPP.LastRoom].myPos,'avatar');
198 this.enemy.frameName = String(M[MT[GAMEAPP.LastRoom].Mnstr].icon+
'.bmp');
199 this.enemy.body.collideWorldBounds = true;
200 this.game.physics.enable(this.enemy, Phaser.Physics.ARCADE);
201
202 this.enemy.alignIn(rect, Phaser.BOTTOM_RIGHT);
203
204 GAMEAPP._PUpdate();
205
206 //place character
207 this.player = this.game.add.sprite(90,230,'avatar');
208 this.player.frameName = 'avatar.bmp';
209 this.player.body.collideWorldBounds = true;
210 this.cursor = this.game.input.keyboard.createCursorKeys();
211
212 this.player.alignIn(rect, Phaser.TOP_LEFT);
213
214 },
215
216 update: function(){
217     this.game.physics.arcade.collide(this.player, this.Room);
218     this.game.physics.arcade.collide(this.enemy, this.Room);
219
220     //Not engaged in melee; if player is dead; show it, and stop the
    combat
221     if(Person[1].ModStmn <= 0){
222         this.player.tint = 0xFF0000;
223         GAMEAPP.RoomAlert = "Defeated ?!";
224         CCP.ModStmn = Person[1].ModStmn;
225         this._raTxt.setText(String(GAMEAPP.RoomAlert));
226         this.attackButton.x = this.world.centerX+800;
227         this.fireButton.x = this.world.centerX+800;
228     }
```

```

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

```

```

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

```

```
340 //Random Critical Adjectives
341 var CAdj = ["BRUTAL ", "BRUTAL ", "MORTAL ", "GAPING ", "SEVERE ",
  "CRITICAL ", "GRAVE ", "CRITICAL "];
342
343 //Random Attack Verbs
344 var RAtkVrb = ["slash ", "slash ", "chop ", "slice ", "thrust ", "cut "
  , "hack ", "chop "];
345
346 //Random Parry Words
347 var RPryVrb = ["block ", "block ", "parry ", "block ", "redirect ",
  "parry ", "counter ", "redirect "];
348
349 //Initialize Variables
350
351 Trash = Math.floor((Math.random() * 6 + 1));
352 AtkWord = RAtkVrb[Trash];
353
354
355 Trash = Math.floor((Math.random() * 6 + 1));
356 PryWord = RPryVrb[Trash];
357
358 for (var i=1 ; i < 2 ; i++){
359
360     Ndx = Math.floor((Math.random() * 12 + 1));
361     ThisAdjWord[i] = AdjWord[Ndx];
362 }
363
364 //Start on this person
365 //Get Atk & Parry rolls
366 AtkRoll = Math.floor((Math.random() * 99 + 1));
367
368 //CombatSkill Function return 0,1 or 10 given person AtkRoll and
  Weapon proficiency.
369 Atk = this._CombatSkill(AtkRoll, Person[Who].WS)
370 console.log("WHO: "+Who);
371 console.log("AtkRoll: "+AtkRoll);
372 if(Atk > 0){
373     //then discover amount of damage
374     //Function WeaponDamage returns final resulting damage given
  person's #.
375     //fixed at 2D6
376     DamRoll = this._GetWpnDamage(2);
377     for(var i=1; i<=2 ;i++){
378         DamRoll += Math.floor((Math.random() * 6 + 1));
379         if (DamRoll<1) {
380             DamRoll = 1;
381         }
382     }
383
384
385 //CombatSkill Function return 0, 1 or 10 given person AtkRoll and
  Weapon proficiency.
386 Atk = this._CombatSkill(AtkRoll, Person[Who].WS)
387 console.log("WHO: "+Who);
388 console.log("AtkRoll: "+AtkRoll);
389 console.log("Damage Done: "+DamRoll);
390 if(Atk == 10){
391     Ndx = Math.floor((Math.random() * 6 + 1));
392     ThisAdjWord[1] = CAdj[Ndx];
```



```

393     }
394
395     //This is the Opponent's Roll
396     PryRoll = Math.floor((Math.random() * 99 + 1));
397
398     //Perform parry for target person
399     Pry = this._CombatSkill(PryRoll, Person[T].PS)
400     console.log("Defender's PS: "+Person[T].PS);
401     console.log("PryRoll: "+PryRoll);
402     if(Pry == 10){
403         Ndx = Math.floor((Math.random() * 6 + 1));
404         ThisAdjWord[2] = CAdj[Ndx];
405         ModTtlAP = (Person[T].TotalAP * 4);
406     }
407
408     //Now Combat Calculations for this Person
409     Trash = 0;
410     Hit = Atk - Pry;
411     console.log("Atk: "+ Atk);
412     console.log("Pry: "+ Pry);
413     console.log("Hit: "+ Hit);
414     console.log("=====");
415
416     //Generate Combat narrative
417     if (Who == 1){
418         TxtTkn = "You: "; //Player
419         TxtTkn2 = "Monster: "; //Monster
420     }else{
421         TxtTkn = "Monster: "; //Monster's Text Color
422         TxtTkn2 = "You: "; //Player's Text Color
423     }
424
425     if (Hit == 0){
426         //attack & parry cancell each other
427         Narrative = TxtTkn + ' a ' + ThisAdjWord[1] + AtkWord + '\n';
428         Narrative += TxtTkn2 + ' a ' + ThisAdjWord[2] + PryWord + '\n';
429         Narrative += TxtTkn2 + ' not injured!';
430     }
431
432     if (Hit == 1){
433         Trash = DamRoll - ModTtlAP;
434         if (Trash < 0) {
435             Trash = 0;
436         }
437         Narrative = TxtTkn + ': a ' + ThisAdjWord[1] + AtkWord + '\n';
438         Narrative += TxtTkn2 + ': a failed ' + PryWord + '\n';
439         Narrative += TxtTkn2 + ': wounded! [' + Trash + ' pts]';
440         Person[T].ModStmn -= parseInt(Trash,10);
441         console.log("Target "+Who+" stmn = "+Person[T].ModStmn);
442         this.mStmnatxt.setText(String(Person[6].ModStmn));
443         this.cStmnatxt.setText(String(Person[1].ModStmn));
444
445         if (Who == 1){
446             Person[1].TempScore += Trash;
447             this.cStmnatxt.setText(String(Person[1].TempScore));
448         }
449     }
450     if (Hit == -1){
451         Narrative = TxtTkn+" a failed "+AtkWord+"\n";

```

```
452         Narrative += TxtTkn2+" a "+AdjWord[2]+PryWord+" awaits.";
453     }
454
455     if (Hit == 9){
456         Trash = ((DamRoll * 2) - ModTtlAP);
457         if (Trash < 0){
458             Trash = 0;
459         }
460
461         Narrative = TxtTkn+" a "+ThisAdjWord[1]+AtkWord+"\n";
462         Narrative += TxtTkn2+" a "+PryWord+" was too late!\n";
463         Narrative += TxtTkn2+" wounded! ["+String(Trash)+" pts]";
464
465         Person[T].ModStmn -= parseInt(Trash,10);
466         console.log("Target "+Who+" stmn = "+Person[T].ModStmn);
467         this.mStmnatxt.setText(String(Person[6].ModStmn));
468         this.cStmnatxt.setText(String(Person[1].ModStmn));
469
470         if (Who == 1){
471             Person[1].TempScore += Trash;
472             this.cStmnatxt.setText(String(Person[1].TempScore));
473         }
474     }
475
476
477     if (Hit == -9){
478         Trash = (DamRoll - (ModTtlAP * 2));
479         if (Trash < 0){
480             Trash = 0;
481         }
482
483         Narrative = TxtTkn+" a "+ThisAdjWord[1]+AtkWord+"\n";
484         Narrative += TxtTkn2+" but a "+ThisAdjWord[2]+PryWord+"
over-powered!\n";
485         Narrative += TxtTkn2+" wounded? ["+String(Trash)+" pts]";
486
487         Person[T].ModStmn -= parseInt(Trash,10);
488         console.log("Target "+Who+" stmn = "+Person[T].ModStmn);
489         this.mStmnatxt.setText(String(Person[6].ModStmn));
490         this.cStmnatxt.setText(String(Person[1].ModStmn));
491
492         if (Who == 1){
493             Person[1].TempScore += Trash;
494             this.cStmnatxt.setText(String(Person[1].TempScore));
495         }
496     }
497
498
499     if (Hit == 10){
500         Trash = ((DamRoll * 2) - ModTtlAP);
501         if (Trash < 0){
502             Trash = 0;
503         }
504
505         Narrative = TxtTkn+" a "+ThisAdjWord[1]+" "+AtkWord+"\n";
506         Narrative += TxtTkn2+" block entirely missed!\n";
507         Narrative += TxtTkn2+" wounded ["+String(Trash)+" pts]";
508
509         Person[T].ModStmn -= parseInt(Trash,10);
```

```

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

```

```

568      //
569      // =====
570      _Cook: function() {
571          if (Person[6].ModStmn > 0) {
572              GAMEAPP.InfoText = " Can't eat in front of the monster!\n ...
                    it's rude ... and makes them\n crazy with hunger!";
573          }
574          if (Person[1].ModStmn < 1) {
575              this._GameReturn;
576          }
577      },
578      //
579      // =====
580      // Return from Combat session
581      _GameReturn: function() {
582          //debug
583          if(Person[6].ModStmn <= 0){
584              console.log("Room #: "+GAMEAPP.CrntRoom+"; Exit combat as
                    victor. ");
585          }else{
586              console.log("Room #: "+GAMEAPP.CrntRoom+"; Exit combat as
                    coward. ");
587              // Ran away while monster was still in the room.
588              CCP.Renown = Person[1].Renown-2;
589              Person[1].Renown = CCP.Renown;
590              GAMEAPP.InfoText = "Running from Combat? \n Renown is "+String
                    (CCP.Renown);
591          }
592          if(Person[1].ModStmn > 0){
593              this.camera.fade(0x000000, 200, false);
594              this.time.events.add(200, function() {
595                  //Return to Room Scene rv_3 through rv_8
596                  this.game.state.start(MT[GAMEAPP.CrntRoom].RmID);
597              }, this);
598          }else{
599              this.camera.fade(0x000000, 200, false);
600              this.time.events.add(200, function() {
601                  //exitGame defeated Room Scene rv_3 through rv_8
602                  this.game.state.start('exitGame');
603              }, this);
604          }
605      },
606      //
607      // =====
608      // Prep Frame 3
609      _GetWpnDamage: function (Dice) {
610          var TempT = 0;
611          //Rolls Multiple 6-sided dice
612          for(var i=1; i<Dice ;i++){
613              TempT += Math.floor((Math.random() * 6 + 1));
614          }
615          if (TempT<1) {
616              TempT = 0;
617          }
618          return TempT;
619      },
620      //
621      //
622      //

```

```

623 // =====
624 _Inventory: function(){
625     if (Person[6].ModStmn > 0) {
626         GAMEAPP.InfoText = "WHAT?! NOW ...\n in the middle of Combat?";
627     }
628     if (Person[1].ModStmn < 1) {
629         this._GameReturn;
630     }
631 },
632
633 //
634 // =====
635 _Search: function(){
636     if (Person[6].ModStmn > 0) {
637         GAMEAPP.InfoText = "WHAT?! NOW ...\n in the middle of Combat?";
638     }
639
640     if(Person[6].ModStmn < 1){
641         GAMEAPP.InfoText = "... looking in its backpack.\nYou savage
642         all food, gold & gems ... ";
643         CCP.HGold += Person[6].Gold;
644         CCP.HGem += Person[6].Gem;
645         CCP.Food += Person[6].Food;
646         this._GameReturn;
647     }
648     if (Person[1].ModStmn < 1) {
649         this._GameReturn;
650     }
651 }
652 //
653 // =====
654 };
655 //
656 // =====
657 //create a box Image (pseudo graphics) for the HTML5 canvas.
658 var box = function(options) {
659     var bxImg = GAMEAPP.game.add.bitmapData(options.length,options.width);
660
661     bxImg.ctx.beginPath();
662     bxImg.ctx.rect(0,0,options.length,options.width);
663     bxImg.ctx.fillStyle = options.color;
664     bxImg.ctx.fill();
665     return bxImg;
666 };
667 //
668 // =====
669 //melee combat: Dynamic Menu; engaged in melee
670 var meleeCombat = function(player,enemy){
671     this.attackButton.x = this.world.centerX-60;
672     this.fireButton.x = this.world.centerX+800;
673     if(Person[6].ModStmn <= 0){
674         this.enemy.tint = 0xFF0000;
675         //GAMEAPP.RoomAlert = "Victory over the " +
676         M[MT[GAMEAPP.LastRoom].Mnstr].Race + "!"
677         this.attackButton.x = this.world.centerX+800;
678         this.fireButton.x = this.world.centerX+800;
679     }
680 }

```

```
679         if(Person[1].ModStmn <= 0){
680             this._GameReturn;
681         }
682     };
683     //
684     // =====
685
686     /* End of file */
687     /* Location: ./js/state/combat-ifStatements-Original.js */
688
689
690
```