```
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://leanpub.com/mbg-memory
5   Affiliate Information:
    http://www.stephen-gose.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: play.js
20   * Live Demo:   http://makingbrowsergames.com/starterkits/memorymatch/
21   * Source Code: http://leanpub.com/mbg-memory
22   * Description: File for play phase state in game shell.
23   * Author:      Stephen Gose
24   * Version: 0.0.0.20101113-15 //semantic version: by (0.0.0.date-#)
25   * Phaser Version: 3.16+
26   * Author URL: http://www.stephen-gose.com/
27   * Support: support@pbmcube.com
28   *
29   * \u00A9 Copyright © 1974-2017 Stephen Gose LLC. All rights reserved.
30   *
31   * Do not sell! Do not distribute!
32   * This is a licensed file with EULA permission.
33   * Please refer to the Terms of Use
34   *   and End Users License Agreement (EULA).
35   *
36   * Search for [ //TODO ] to tailor this file for your own use;
37   *   doing so will void any support agreement.
38   *
39   * Redistribution of part or whole of this file and
40   * the accompanying files is strictly prohibited.
41   *
42   * Reference:
43   * http://www.adequatelygood.com/JavaScript-Module-Pattern-In-Depth.html
44   *
45   */
46
47  "use strict";
48  var pointer, scoreText,timeText,timer, clickTime, flipTimer, totalPairs;
49  var totalTiles = (GAMEAPP.numRows * GAMEAPP.numCols);
50  var noMatch = false;// game return  to back-side flip
51  var attempts = 0;   // count how many times tiles were chosen
52  var tileButton = {};
53  var target1 = -1;   // preserve 1st selected mismatched tile
```



```
54    var target2 = -2;   // preserve 2nd selected mismatched tile
55    var collectIO = []; // use array to collect "selected tiles"
56    var tileList = [];  // roster of all tiles created
57    var sfxAlbum = [];  // audio tones confirmations
58    var tileCount = 0;  // assigned tile index number
59    var playSound=true; // sound button toggle
60    var endGT = false;  // processing current game turn selections?
61
62    // ================================================
63    window.GAMEAPP.state.play = {
64        // ----------------------------------------
65        // Phaser Essential Functions
66        // ----------------------------------------
67        init: function () {
68            // not used in demo;
69            // licensed version: loads all 8 semesters
70        },
71        //
72        // ----------------------------------------
73        preload: function () {
74            console.log(" %c wcrpg101 rv_15 Game Play launched ",
              "color:white; background:red");
75            this.load.spritesheet("tiles",
              "assets/spriteSheets/colorTiles.png", {
76                frameWidth: GAMEAPP.tileSize,
77                frameHeight: GAMEAPP.tileSize
78            });
79
80            //individual tiles png
81            for (var i = 0; i < 9; i++) {
82                this.load.image("tiles" + i, "assets/images/colorTiles" + i +
                  ".png", {
83                    frameWidth: GAMEAPP.tileSize,
84                    frameHeight: GAMEAPP.tileSize
85                });
86            }
87            // or simply draw graphics with color array
88
89            // audio sound effects (sfx)
90            this.load.audio("first", ['assets/sfx/FirstCardSound.mp3',
              'assets/sfx/FirstCardSound.ogg']);
91            this.load.audio("match", ['assets/sfx/MatchSound.mp3',
              'assets/sfx/MatchSound.ogg']);
92            this.load.audio("miss", ['assets/sfx/MissSound.mp3',
              'assets/sfx/MissSound.ogg']);
93        },
94        //
95        // ----------------------------------------
96        create: function () {
97            // loading has finished - proceed to demo state
98            console.log("starting play phase");
99            // display theme & blackboard grid
100           var bg = this.add.image(config.width / 2, config.height / 2,
              'play');
101           var bb = this.add.image(config.width / 2, (config.height / 2) +
              30, 'blackboard');
102
103           // card flip timer
104           flipTimer = this.time.addEvent({
```

```
105             delay: 500,         // in ms
106             //callback: callback,
107             //args: [],
108             //callbackScope: thisArg,
109             loop: true
110         });
111
112         // add sound effects (sfx)
113         // https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API
114         sfxAlbum[0] = this.sound.add('first');
115         sfxAlbum[1] = this.sound.add('match');
116         sfxAlbum[2] = this.sound.add('miss');
117         //console.log("SFX Album: ")
118         //console.table(sfxAlbum);
119
120         // ==================
121         // Place Tiles
122         // - a separate function in v2.x.x
123         // - create list of Tile indices to shuffle
124         // - mimic mathdeck[] & spriteSheet color frames (cardlist[])
125         // ==================
126         totalTiles = (GAMEAPP.numRows * GAMEAPP.numCols);
127         totalPairs = totalTiles/2;  // if player finished before
            remaining time
128         // ------------------
129         // shuffle the tile roster (using Fisher-Yates method)
130         // hints use "color list" for pre-schoolers
131         //
            https://github.com/photonstorm/phaser/blob/v3.22.0/src/utils/array/
            Shuffle.js
132         for (var i = totalTiles; i > 0; i--) {
133             var a = _getRandom(0, totalTiles - 1);
134             var b = _getRandom(0, i);
135             var temp = mathdeck[a];
136             // frame colors mimic mathdeck
137             var color = cardlist[a];
138             mathdeck[a] = mathdeck[b];
139             cardlist[a] = cardlist[b];
140             mathdeck[b] = temp;
141             cardlist[b] = color;
142         }
143
144         // ------------------
145         // Design Decision: create grid, group or container for tiles
146         // - choose groups or containers?
147         // ------------------
148         // https://phaser.io/news/2018/10/ui-blocks
149         //
            https://phasergames.com/ui-blocks-a-lightweight-alternate-to-contai
            ners/
150         //
            https://photonstorm.github.io/phaser3-docs/Phaser.GameObjects.Conta
            iner.html
151         // https://phaser.io/phaser3/devlog/120
152         // https://github.com/photonstorm/phaser/issues/3852
153         // ------------------
154         //
            https://photonstorm.github.io/phaser3-docs/Phaser.GameObjects.Group
            .html
```



```
155             // - a group: A Group is a way for you to create, manipulate, or
                recycle similar Game Objects.
156             // Group membership is non-exclusive. A Game Object can belong to
                several groups, one group, or none.
157             // Groups themselves aren't displayable, and can't be positioned,
                rotated, scaled, or hidden.
158             // ------------------
159             //
                https://photonstorm.github.io/phaser3-docs/Phaser.GameObjects.Conta
                iner.html
160             // - a container: This method will only be available if the
                Container Game Object has been built into Phaser.
161             // pre3.14 = Containers can include other Containers for deeply
                nested transforms.
162             // Nested containers removed v3.14+
163             // ------------------
164             // standard double loop used
165             for (var j = 0; j < GAMEAPP.numRows; j++) { // (Y) vertical
166                 for (i = 0; i < GAMEAPP.numCols; i++) { // (X) horizontal
167                     // debug: exchange & uncomment to see tile colors
168                     // tileButton =
                        this.add.sprite(305+(i*(80*.8)),188+(j*(80*.8)),'tiles',car
                        dlist[tileCount]);
169                     // normal game play below
170                     // var cell = new Tile(tileCount,305 + (i * (80 *
                        .8)),188 + (j * (80 * .8)),{});
171                     tileButton = this.add.sprite(305 + (i * (80 * .8)), 188 +
                        (j * (80 * .8)), 'tiles',{frameWidth: 82,frameHeight: 82});
172                     // use Phaser Data Manager instead of attaching
                        properties to array?
173                     // -
                        https://phaser.io/examples/v3/view/components/data/change-d
                        ata-event
174                     // -
                        https://photonstorm.github.io/phaser3-docs/Phaser.Data.Data
                        Manager.html
175                     tileButton.index = tileCount;
176                     tileButton.front = mathdeck[tileCount]; // assign math
                        formula
177                     tileButton.color = cardlist[tileCount]; // assign random
                        color
178                     tileButton.col = i;
179                     tileButton.row = j;
180                     tileButton.name = "r" + j + "c" + i;
181                     tileButton.discovered = false; // for AI-bot opponent
182
183                     tileButton.setInteractive({ useHandCursor: true });
184                     tileButton.setFrame(0);
185                     tileButton.setScale(.8, .8);
186                     tileButton.enableBody = true;
187
188                     tileCount++;
189                     // -----------------------------------------
190                     // Design Decision:
191                     // - use containers originally since they have the
                        ability to
192                     // "getByName(name)". See
193                     //
                        https://github.com/photonstorm/phaser/blob/v3.22.0/src/game
```



```
                    objects/container/Container.js#L617
194                 //
195                 // However, further research shows that this
196                 // uses Phaser "ArrayUtils.GetFirst". See
197                 //
                    https://github.com/photonstorm/phaser/blob/v3.22.0/src/utils/array/GetFirst.js#L30
198                 //
199                 // Conclusion: write my own "seek" & find array
200                 // ----------------------------------------
201                 // insert tile into my own stylized "container array"
202                 tileList.push(tileButton);
203             } // end of X columns
204         } // end of Y rows
205         // ------------------
206         // ==================
207         // debug
208         // ------------------
209         //gameMechanics(cell," Alternate Design Option for Tiles ")
210         console.table(tileList);
211         console.table(mathdeck);
212         console.table(cardlist);
213         // ------------------
214         // ==================
215         // End Tile Placement
216         // ==================
217
218         // ==================
219         // Heads Up Display (HUD)
220         // ==================
221         // ------------------
222         // HUD - input Button
223         // ------------------
224         pointer = this.input.activePointer;
225         this.input.mouse.disableContextMenu();
226
227         // Phaser III Scene level
228         this.input.on('pointerdown', function (event, gameObjects) {
229             // debug:
230             console.info("Tile #" + gameObjects[0].index + " clicked.\n -
                    front: " + gameObjects[0].front);
231             _showTiles(gameObjects);
232         });
233         // ------------------
234         // HUD - Course subtitle
235         // ------------------
236         // Game Title embedded in background theme
237         this.subTitle = this.add.text(config.width / 2, 145,
            GAMEAPP.subTitle + " "+ GAMEAPP.gradeLevel+": 'Addition'",
            GAMEAPP.styleN);
238         this.subTitle.setOrigin(0.5, 0.5);
239         this.subTitle.setShadow(3, 3, 'rgba(0,0,0,0.5)', 5);
240
241         // ------------------
242         // HUD - score
243         // ------------------
244         // more points if "hints" are OFF
245         scoreText = this.add.text((config.width / 2), config.height - 60,
            "Score (hints off): 0", GAMEAPP.styleRA2);
```



```
246             scoreText.setOrigin(0.5);
247             scoreText.setShadow(3, 3, 'rgba(0,0,0,0.5)', 5);
248
249             // ------------------
250             // HUD - timer text
251             // ------------------
252             timeText = this.add.text(config.width / 2, config.height - 40,
                "Time left: " + GAMEAPP.gameTime, GAMEAPP.styleRA2);
253             timeText.setOrigin(0.5);
254             timeText.setShadow(3, 3, 'rgba(0,0,0,0.5)', 5);
255
256             // ------------------
257             // HUD - timer clocks; selectable game option
258             // - count down timer (decreaseTime) or
259             // - stopwatch timer?
260             // ------------------
261             // Time Events binding to Phaser III gaming framework;
262             // - https://labs.phaser.io/index.html?dir=events/&q=
263             // -
                https://labs.phaser.io/edit.html?src=src\events\listen%20to%20game%
                20object%20event.js
264             // -
                https://labs.phaser.io/edit.html?src=src\events\emit%20scene%20even
                t.js
265             // Repeat Timer:
                https://rexrainbow.github.io/phaser3-rex-notes/docs/site/timer/
266             // old Phaser v2.x.x: game.time.events.loop(Phaser.Timer.SECOND,
                setRandomNote, this);
267
268             /** new Phaser III timer event settings:
269             var timer = scene.time.addEvent({
270             delay:      500,        // in milliseconds
271             callback:   callback,   // what to do when event happens
272             args:       [],         // passed arguments to callback
273             callbackScope: thisArg,
274             loop:       fasle,      // continue?
275             repeat:     0,          // or limit repeativive actions
276             startAt:    0,          // starting timeframe
277             timeScale:  1,
278             paused:     false       // current action state
279             });
280              */
281
282             // new design strategy:
283             // - 60 seconds per game play session; mimics casual games!
284             // - only configurable option "GAMEAPP.gameTime"; default 60
                seconds
285             // - a stopwatch timer? (_stopWatchTimer)
286             //timer = this.time.addEvent({ delay: 1000, callback:
                _stopWatchTimer, callbackScope: this, repeat: GAMEAPP.gameTime});
287             // - OR count-down timer (_decreaseTime)
288
289             //works 100%; turned off during development
290             timer = this.time.addEvent({
291                     delay: 1000,
292                     callback: _decreaseTimer,
293                     callbackScope: this,
294                     repeat: GAMEAPP.gameTime
295                 });
```



```
296
297             // ------------------
298             // Copyright notice embedded into background theme
299             // Reason: ensures artwork exchange for bespoke solution
300             // ==================
301         },
302         //
303         // ----------------------------------------
304         update: function () {
305             if (GAMEAPP.hints) {
306                 // 25% penalty when color hints "ON"
307                 scoreText.setText("Score (hints on): " + (GAMEAPP.gameScore
                    *.75));
308             } else {
309                 // full score
310                 scoreText.setText("Score (hints off): " + GAMEAPP.gameScore);
311             }
312
313             if (noMatch) {  // see processSelection
314                 //if (flipTimer.getElapsedSeconds() - clickTime > 0.5) {
315                 if (clickTime > 50){    // use 60 fps as a timer
316                     tileList[target1].setFrame(0);  // turn on back-side
317                     tileList[target2].setFrame(0);  // turn on back-side
318                     //hiddenContent[firstTile].setVisible(false);   // turn
                        off front-side
319                     //hiddenContent[secondTile].setVisible(false);// turn off
                        front-side
320
321                     // reset sentinel flag for new game turn
322                     noMatch = false;    // turn off game turn flag
323                     target1 = null; target2 = null;
324                 }
325                 clickTime++;    // use 60 fps as the timer
326             }
327
328
329         }
330         //place " , " If using more supporting methods below
331
332     }; // End Phaser Essential Functions
333     // ================================================
334
335     // ================================================
336     // ----------------------------------------
337     // Supporting Game Function & Components
338     // (alphabetical listing)
339     // ----------------------------------------
340     // ================================================
341     // validate matching tiles
342     function _checkTiles() {
343         // ---------------
344         //debug
345         // ---------------
346         console.log("inside _checkTiles validation");
347         // ---------------
348         // and do their colors match?
349         if (GAMEAPP.firstCardColor == GAMEAPP.secondCardColor) {
350             // play successfully matched tone!
351             sfxAlbum[1].play(); // sfx "matched" alert
```

```
352
353             // if nothing remains; move to gameOver prior clock out.
354             totalPairs--;
355
356             // award discovery points
357             GAMEAPP.gameScore += GAMEAPP.pointsForMatch;
358
359             // remove selected tiles from game board
360             for (var j = 0; j < tileList.length; j++) {
361                 // search for "discovered" and tile by name
362                 if (GAMEAPP.firstCardNam == tileList[j].name) {
363                     console.log("found 1st tile!");
364                     tileList[j].setVisible(false);
365                     tileList[j].discovered = false; // removed from game
366                     //tileList[j].disableInteractive;
367                     console.log("found 1st match: "+tileList[j].name+" at
                        index #"+j+"!");
368                 }
369
370                 if (GAMEAPP.secondCardNam == tileList[j].name) {
371                     console.log("found 2nd tile!");
372                     tileList[j].setVisible(false);
373                     tileList[j].discovered = false; // removed from game
374                     //tileList[j].disableInteractive;
375                     console.log("found 2nd match: "+tileList[j].name+" at
                        index #"+j+"!");
376                 }
377
378             } // end of tileList
379
380         } else {
381             // oops! Try again!
382             sfxAlbum[2].play(); // sfx "missed" alert
383             console.log(">> mismatch 1) #"+GAMEAPP.firstCardNdx+"
                "+GAMEAPP.firstCardNam +"; - color: "+GAMEAPP.firstCardColor);
384             console.log(">> mismatch 2) #"+GAMEAPP.secondCardNdx+"
                "+GAMEAPP.secondCardNam +"; - color: "+GAMEAPP.secondCardColor);
385
386             // penalty points
387             GAMEAPP.gameScore -= GAMEAPP.pointsForMiss;
388
389             // not a match?
390             // turn on game turn flag for update()
391             noMatch = true;
392
393             // ----------------------------------------
394             // Design Decision:
395             // ----------------------------------------
396             // - use containers since they have the ability to
397             // "getByName(name)". See
398             //
                https://github.com/photonstorm/phaser/blob/v3.22.0/src/gameobjects/
                container/Container.js#L617
399             //
400             // However, further research shows that this
401             // uses Phaser "ArrayUtils.GetFirst". See
402             //
                https://github.com/photonstorm/phaser/blob/v3.22.0/src/utils/array/
                GetFirst.js#L30
```



```
403             //
404             // Conclusion: write my own "seek" & find array
405             // ------------------------------------------
406
407             // return selected tiles back to the "??" back-side
408             for (var j = 0; j < tileList.length; j++) {
409     /*          // Design Option 1) consolidated
410                 // search for by name
411                 if ((GAMEAPP.firstCardNam == tileList[j].name) ||
                    (GAMEAPP.secondCardNam == tileList[j].name)) {
412                     tileList[j].discovered = true; // available for AI-bot
413                     tileList[j].setFrame(0).setScale(.8, .8);
414                 }
415     */
416
417                 // Design Option 2) readable & exposed logic
418                 if (GAMEAPP.firstCardNam == tileList[j].name) {
419                     // flip to back-side
420                     tileList[j].discovered = true; // available for AI-bot
421                     target1 = j;    // preserve location
422                     console.log("found 1st mismatch: "+tileList[j].name+" at
                        index #"+j+"!");
423                 }
424
425                 if (GAMEAPP.secondCardNam == tileList[j].name) {
426                     // flip to back-side
427                     tileList[j].discovered = true; // available for AI-bot
428                     target2 = j;    // preserve location
429                     console.log("found 2nd mismatch: "+tileList[j].name+" at
                        index #"+j+"!");
430                 }
431             } // end of tileList
432         }
433
434         resetGT();
435
436         // if player finishes ahead of time remaining
437         // see stop-watch time below
438         // - use "short circuit" condition: checking totalPairs first
439         if(totalPairs == 0){
440             // set count-down timer to 0
441             GAMEAPP.gameTime = 0;
442         }
443     }
444
445     //
446     // ================================================
447     // count down timer: when it is zero; game session ends
448     function _decreaseTimer() {
449         GAMEAPP.gameTime--;
450         timeText.setText("Time left: " + GAMEAPP.gameTime);
451
452         if (GAMEAPP.gameTime < 0) {
453             this.scene.transition({
454
455                 target: "exitGame",
456                 duration: 250,
457                 sleep: false, // true = to sleep current scene;
458                 // false = to stop current scene
```

```
459                 moveAbove: true, // move the target Scene above this current
460                 //  scene before the transition starts
461                 moveBelow: false, // move the target Scene below this current
462                 //onUpdate: GameReturn(),
463                 onUpdateScope: "exitGame"
464             });
465         }
466     }
467 
468     //
469     // ================================================
470     // Randomly choose tiles
471     // The maximum is inclusive and the minimum is inclusive
472     function _getRandom(min, max) {
473         min = Math.ceil(min);
474         max = Math.floor(max);
475         var rndIndex = Math.floor(Math.random() * (max - min + 1)) + min;
476         return (rndIndex === undefined) ? max : rndIndex;
477     }
478 
479     //
480     // ================================================
481     // refactor common statements into function
482     function resetGT(){
483         // -
            https://photonstorm.github.io/phaser3-docs/Phaser.Time.TimerEvent.html#
            toc1__anchor
484         // - https://rexrainbow.github.io/phaser3-rex-notes/docs/site/timer/
485         // var elapsed = timer.getElapsed();   // ms
486         // var elapsed = timer.getElapsedSeconds(); // sec
487 
488         // Phaser III
489         clickTime = 0;          // use update 60fps as timer
490         clickTime = flipTimer.getElapsedSeconds();
491         console.log("2) time: "+clickTime);
492 
493         // update attempts
494         attempts++;
495 
496         // reset collection, tile sentinels, and game-turn flag
497         collectIO = [];
498         GAMEAPP.firstCard = false;
499         GAMEAPP.firstCardNdx = -1;      // forced mismatch to dirrerent index
500         GAMEAPP.firstCardColor = -1;    // forced mismatch to dirrerent color
501         GAMEAPP.firstCardNam = "";
502 
503         GAMEAPP.secondCard = false;
504         GAMEAPP.secondCardNdx = -2;     // forced mismatch to dirrerent index
505         GAMEAPP.secondCardColor = -2;   // forced mismatch to dirrerent color
506         GAMEAPP.secondCardNam = "";
507 
508         endGT = false;  // reset game turn flag
509         console.log("=== GT Reset ===");
510 
511     }
512 
513     //
514     // ================================================
515     // Show selected tiles
```



```
516   function _showTiles(target) {
517   //console.log("this tile has value = " + target.value);
518       target[0].setFrame(target[0].color).setScale(.8, .8);
519       target[0].discovered = true; // for AI-bot
520       console.log(target[0].name + " clicked");
521       // ---------------
522       // Design options:
523       // ---------------
524       // 1) use an "input collection" array
525       // - push 1st & 2nd selected tiles
526       // - if(collectIO.length < 2 && target[0].index > -1){
527       //
528       // 2) use just sentinel variables
529       // - if(GAMEAPP.firstCard == false && GAMEAPP.secondCard == false){
530       // - if(GAMEAPP.firstCard == true && GAMEAPP.secondCard == false){
531       // ---------------
532       // Design option 1)
533       if (collectIO.length < 2 && target[0].index > -1) {
534
535           if (GAMEAPP.firstCard == false) {
536               // toggle 1st sentinel flags
537               sfxAlbum[0].play(); // audio alert
538               // toggle sentinel
539               GAMEAPP.firstCard = true;
540               // preserve selected tile index
541               GAMEAPP.firstCardNdx = target[0].index;
542               GAMEAPP.firstCardColor = target[0].color;
543               GAMEAPP.firstCardNam = target[0].name;
544               console.log(">>1) #"+GAMEAPP.firstCardNdx+"
                  "+GAMEAPP.firstCardNam +"; - color: "+GAMEAPP.firstCardColor);
545           } else {
546               // toggle 2nd sentinel flags
547               GAMEAPP.secondCard = true;
548               // preserve selected tile index
549               GAMEAPP.secondCardNdx = target[0].index;
550               GAMEAPP.secondCardColor = target[0].color;
551               GAMEAPP.secondCardNam = target[0].name;
552               console.log(">>2) #"+GAMEAPP.secondCardNdx+"
                  "+GAMEAPP.secondCardNam +"; - color:
                  "+GAMEAPP.secondCardColor);
553           }
554
555           collectIO.push(target[0].index);
556       }
557
558       // two inputs collected yet?
559       // only validate tiles once we have 2
560       // are there 2 diffirent tiles?
561       if (collectIO.length == 2) {
562
563           console.table(collectIO);
564           // AND first tile NOT selected again
565           if(collectIO[0] != collectIO[1] ){
566               // end game turn and process these collections
567               endGT = true;
568               _checkTiles();  // validate match
569           } else {
570               // same tile clicked twice
571               console.log("same tile clicked twice: "+collectIO[0]+" |
```

```
                "+collectIO[1]);
572             // flip to back-side
573             target[0].discovered = false; // available for AI-bot
574             target[0].setFrame(target[0].color).setScale(.8, .8);
575
576             resetGT();
577         }
578
579     }
580
581 }
582
583 //
584 // ================================================
585 function _stopWatchTimer() {
586     GAMEAPP.gameStartTime++;
587     timeText.setText("Time left: " + GAMEAPP.gameStartTime);
588
589     // use "short circuit" condition: checking totalPairs first
590     if (totalPairs == 0 || GAMEAPP.gameStartTime > GAMEAPP.gameTime) {
591         this.scene.transition({
592
593             target: "exitGame",
594             duration: 250,
595             sleep: false, // true = to sleep current scene;
596             // false = to stop current scene
597             moveAbove: true, // move the target Scene above this current
598             //  scene before the transition starts
599             moveBelow: false, // move the target Scene below this current
600             //onUpdate: GameReturn(),
601             onUpdateScope: "exitGame"
602         });
603     }
604 }
605
606 //
607 // ================================================
608 // Alternate Design Option for Tiles as objects
609 function Tile(key, x,y, handler) {
610
611     this.key = key;                     // index from tileCount
612     this.front = mathdeck[tileCount];   // front content on tile
613     this.content = "";
614     this.color = cardlist[tileCount];   // randomly assigned color
615     this.col = x;
616     this.row = y;
617     this.name = "r" + x + "c" + y;
618     this.state = false;                 //sprite.state out of the game??
619     this.dicovered = false              // AI-bot can use?
620     this.handler = handler;
621
622     this._draw = function (x, y) {
623         //this.frontbg = this.add.sprite(x, y,
        'front.png').setInteractive();
624         //this.cover = this.add.sprite(x, y, 'back.png').setInteractive();
625         this.content = this.add.text(x, y, this.front);
626
627         //this.cover.on('pointerdown', this._onClickHandler.bind(this));
628         this.front.on('pointerdown', this._onClickHandler.bind(this));
```

```
629
630             this._faceDown();
631         };
632
633         this._readOnly = function () {
634             this.removed = true;
635         };
636
637         this._isVisible = function () {
638             return this.front.visible;
639         };
640
641         this._faceDown = function () {
642             if (!this.removed) {
643                 //this.frontbg.visible = false;
644                 this.front.visible = false;
645                 this.setFrame(0);
646                 //this.cover.visible = true;
647             }
648         };
649
650         this._faceUp = function () {
651             if (!this.removed) {
652                 //this.frontbg.visible = true;
653                 this.front.visible = true;
654                 this.setFrame(this.color);
655                 //this.cover.visible = false;
656             }
657         };
658
659         this._onClickHandler = function () {
660             this.handler(this);
661         };
662     }
663
664     //
665     // ================================================
666     // ------------------------------------------
667     /* End of file */
668     /* Location: ./js/state/play.js */
669
```