

```

12     public class ConsoleView
13     {
14         #region FIELDS
15
16         //
17         // declare a Universe and Traveler object for the ConsoleView object to
18     use
19         //
20         Universe _gameUniverse;
21         Traveler _gameTraveler;
22
23         #endregion
24
25         #region PROPERTIES
26
27         #endregion
28
29         #region CONSTRUCTORS
30
31         /// <summary>
32         /// default constructor to create the console view objects
33         /// </summary>
34         public ConsoleView(Traveler gameTraveler, Universe gameUniverse)
35         {
36             _gameTraveler = gameTraveler;
37             _gameUniverse = gameUniverse;
38
39             InitializeConsole();
40         }
41
42         #endregion
43
44         #region METHODS
45
46         /// <summary>
47         /// initialize all console settings
48         /// </summary>
49         private void InitializeConsole()
50         {
51             ConsoleUtil.WindowTitle = "The TARDIS Project";
52             ConsoleUtil.HeaderText = "The TARDIS Project";
53         }
54
55         /// <summary>
56         /// display the Continue prompt
57         /// </summary>
58         public void DisplayContinuePrompt()
59         {
60             Console.CursorVisible = false;
61
62             Console.WriteLine();
63
64             ConsoleUtil.DisplayMessage("Press any key to continue.");
65             ConsoleKeyInfo response = Console.ReadKey();
66
67             Console.WriteLine();
68
69             Console.CursorVisible = true;
70         }
71
72         /// <summary>
73         /// display the Exit prompt on a clean screen
74         /// </summary>

```

```
74     public void DisplayExitPrompt()
75     {
76         ConsoleUtil.HeaderText = "Exit";
77         ConsoleUtil.DisplayReset();
78
79         Console.CursorVisible = false;
80
81         Console.WriteLine();
82         ConsoleUtil.DisplayMessage(
83             "Thank you for playing The TARDIS Project. Press any key to Exit.");
84
85         Console.ReadKey();
86
87         System.Environment.Exit(1);
88     }
89
90     /// <summary>
91     /// display the welcome screen
92     /// </summary>
93     public void DisplayWelcomeScreen()
94     {
95         StringBuilder sb = new StringBuilder();
96
97         ConsoleUtil.DisplayReset();
98
99         ConsoleUtil.DisplayMessage("The TARDIS Project");
100        ConsoleUtil.DisplayMessage("");
101        ConsoleUtil.DisplayMessage("Written by John Velis");
102        ConsoleUtil.DisplayMessage("Northwestern Michigan College");
103        Console.WriteLine();
104
105        //
106        // TODO update opening screen
107        //
108
109        sb.Clear();
110        sb.AppendFormat(
111            "You have been hired by the Norlon Corporation to participate ");
112        sb.AppendFormat(
113            "in its latest endeavor, the TARDIS Project. Your mission is to ");
114        sb.AppendFormat(
115            "test the limits of the new TARDIS Machine and report back to ");
116        sb.AppendFormat("the Norlon Corporation.");
117        ConsoleUtil.DisplayMessage(sb.ToString());
118        Console.WriteLine();
119
120        sb.Clear();
121        sb.AppendFormat("Your first task will be to set up the initial para
122        meters of your mission.");
123        ConsoleUtil.DisplayMessage(sb.ToString());
124
125        DisplayContinuePrompt();
126    }
127
128    /// <summary>
129    /// setup the new Traveler object
130    /// </summary>
131    public void DisplayMissionSetupIntro()
132    {
133        //
134        // display header
135        //
136        ConsoleUtil.HeaderText = "Mission Setup";
```

```

132         ConsoleUtil.DisplayReset();
133
134         //
135         // display intro
136         //
137         ConsoleUtil.DisplayMessage("You will now be prompted to enter the s
starting parameters of your mission.");
138         DisplayContinuePrompt();
139     }
140
141     /// <summary>
142     /// display a message confirming mission setup
143     /// </summary>
144     public void DisplayMissionSetupConfirmation()
145     {
146         //
147         // display header
148         //
149         ConsoleUtil.HeaderText = "Mission Setup";
150         ConsoleUtil.DisplayReset();
151         ConsoleUtil.HeaderText = "Mission Setup";
152         ConsoleUtil.DisplayReset();
153
154         //
155         // display confirmation
156         //
157         ConsoleUtil.DisplayMessage("");
158         ConsoleUtil.DisplayMessage("Your mission setup is complete.");
159         ConsoleUtil.DisplayMessage("");
160         ConsoleUtil.DisplayMessage(
        "To view your TARDIS traveler information use the Main Menu.");
161
162         DisplayContinuePrompt();
163     }
164
165     /// <summary>
166     /// get player's name
167     /// </summary>
168     /// <returns>name as a string</returns>
169     public string DisplayGetTravelersName()
170     {
171         string travelersName;
172
173         //
174         // display header
175         //
176         ConsoleUtil.HeaderText = "Traveler's Name";
177         ConsoleUtil.DisplayReset();
178
179         ConsoleUtil.DisplayPromptMessage("Enter your name: ");
180         travelersName = Console.ReadLine();
181
182         ConsoleUtil.DisplayReset();
183         ConsoleUtil.DisplayMessage($
        "You have indicated {travelersName} as your name.");
184
185         DisplayContinuePrompt();
186
187         return travelersName;
188     }
189
190     /// <summary>
191     /// get and validate the player's race

```

```

192     /// </summary>
193     /// <returns>race as a RaceType</returns>
194     public Traveler.RaceType DisplayGetTravelersRace()
195     {
196         bool validResponse = false;
197         Traveler.RaceType travelersRace = Traveler.RaceType.None;
198
199         while (!validResponse)
200         {
201             //
202             // display header
203             //
204             ConsoleUtil.HeaderText = "Traveler's Race";
205             ConsoleUtil.DisplayReset();
206
207             //
208             // display all race types on a line
209             //
210             ConsoleUtil.DisplayMessage("Races");
211             StringBuilder sb = new StringBuilder();
212             foreach (Character.RaceType raceType in Enum.GetValues(typeof(
Character.RaceType)))
213             {
214                 if (raceType != Character.RaceType.None)
215                 {
216                     sb.Append($" [{raceType}] ");
217                 }
218             }
219             ConsoleUtil.DisplayMessage(sb.ToString());
220
221             ConsoleUtil.DisplayPromptMessage("Enter your race: ");
222
223             //
224             // validate user response for race
225             //
226             if (Enum.TryParse<Character.RaceType>(Console.ReadLine(), out
travelersRace))
227             {
228                 validResponse = true;
229                 ConsoleUtil.DisplayReset();
230                 ConsoleUtil.DisplayMessage($
231                     "You have indicated {travelersRace} as your race type.");
232             }
233             else
234             {
235                 ConsoleUtil.DisplayMessage(
236                     "You must limit your race to the list above.");
237                 ConsoleUtil.DisplayMessage("Please reenter your race.");
238             }
239             DisplayContinuePrompt();
240         }
241
242         return travelersRace;
243     }
244
245     /// <summary>
246     /// get and validate the player's TARDIS destination
247     /// </summary>
248     /// <returns>space-time location</returns>
249     public SpaceTimeLocation DisplayGetTravelersNewDestination()
250     {

```

```

251         bool validResponse = false;
252         int locationID;
253         SpaceTimeLocation nextSpaceTimeLocation = new SpaceTimeLocation();
254
255         while (!validResponse)
256         {
257             //
258             // display header
259             //
260             ConsoleUtil.HeaderText = "TARDIS Destination";
261             ConsoleUtil.DisplayReset();
262
263             //
264             // display a table of space-time locations
265             //
266             DisplayTARDISDestinationsTable();
267
268             //
269             // get and validate user's response for a space-time location
270             //
271             ConsoleUtil.DisplayPromptMessage(
272                 "Choose the TARDIS destination by entering the ID: ");
273
274             //
275             // user's response is an integer
276             //
277             if (int.TryParse(Console.ReadLine(), out locationID))
278             {
279                 ConsoleUtil.DisplayMessage("");
280
281                 try
282                 {
283                     nextSpaceTimeLocation = _gameUniverse.
284                         GetSpaceTimeLocationByID(locationID);
285
286                     ConsoleUtil.DisplayReset();
287                     ConsoleUtil.DisplayMessage($"You have indicated {nextSpaceTimeLocation.Name} as your TARDIS destination.");
288                     ConsoleUtil.DisplayMessage("");
289
290                     if (nextSpaceTimeLocation.Accessible == true)
291                     {
292                         validResponse = true;
293                         ConsoleUtil.DisplayMessage(
294                             "You will be transported immediately.");
295                     }
296                     else
297                     {
298                         ConsoleUtil.DisplayMessage("It appears this destination is not available to you at this time.");
299                         ConsoleUtil.DisplayMessage(
300                             "Please make another choice.");
301                     }
302                 }
303                 catch (ArgumentOutOfRangeException ex)
304                 {
305                     ConsoleUtil.DisplayMessage(
306                         "It appears you entered an invalid location ID.");
307                     ConsoleUtil.DisplayMessage(ex.Message);
308                     ConsoleUtil.DisplayMessage("Please try again.");
309                 }
310             }
311             //
312             // user's response was not in the correct range
313             //
314         }

```

```

307         }
308     }
309     //
310     // user's response was not an integer
311     //
312     else
313     {
314         ConsoleUtil.DisplayMessage(
            "It appears you did not enter a number for the location ID.");
        ConsoleUtil.DisplayMessage("Please try again.");
    }
315
316     DisplayContinuePrompt();
317 }
318
319 return nextSpaceTimeLocation;
320 }
321
322
323
324 /// <summary>
325 /// generate a table of space-time location names and ids
326 /// </summary>
327 public void DisplayTARDISDestinationsTable()
328 {
329     int locationNumber = 1;
330
331     //
332     // table headings
333     //
334     ConsoleUtil.DisplayMessage("ID".PadRight(10) + "Name".PadRight(20))
335     ;
336     ConsoleUtil.DisplayMessage("---".PadRight(10) + "-----".
337     PadRight(20));
338
339     //
340     // location name and id
341     //
342     foreach (SpaceTimeLocation location in _gameUniverse.
343     SpaceTimeLocations)
344     {
345         ConsoleUtil.DisplayMessage(location.SpaceTimeLocationID.
346         ToString().PadRight(10) + location.Name.PadRight(20));
347         locationNumber++;
348     }
349 }
350
351
352 /// <summary>
353 /// get the action choice from the user
354 /// </summary>
355 public TravelerAction DisplayGetTravelerActionChoice()
356 {
357     TravelerAction travelerActionChoice = TravelerAction.None;
358     bool usingMenu = true;
359
360     while (usingMenu)
361     {
362         //
363         // set up display area
364         //
365         ConsoleUtil.HeaderText = "Traveler Action Choice";
366         ConsoleUtil.DisplayReset();
367         Console.CursorVisible = false;

```

```

364
365 //
366 // display the menu
367 //
368 ConsoleUtil.DisplayMessage(
    "What would you like to do (Type Number).");
369 Console.WriteLine();
370 Console.WriteLine(
    "\t" + "1. Look Around" + Environment.NewLine +
    "\t" + "2. Travel" + Environment.NewLine +
    "\t" + "3. Display Traveler Info" + Environment.NewLine +
    "\t" + "4. Display Traveler Inventory" + Environment.
371 NewLine +
372 "\t" + "5. Display Traveler Treasure" + Environment.
373 NewLine +
374 "\t" + "6. Display All TARDIS Destinations" + Environment.
375 NewLine +
376 "\t" + "7. Display All Game Items" + Environment.NewLine +
377 "\t" + "8. Display All Game Treasures" + Environment.
378 NewLine +
379 "\t" + "E. Exit" + Environment.NewLine);
380
381 //
382 // get and process the user's response
383 // note: ReadKey argument set to "true" disables the echoing o
384 f the key press
385 //
386 ConsoleKeyInfo userResponse = Console.ReadKey(true);
387 switch (userResponse.KeyChar)
388 {
389     case '1':
390         travelerActionChoice = TravelerAction.LookAround;
391         usingMenu = false;
392         break;
393     case '2':
394         travelerActionChoice = TravelerAction.Travel;
395         usingMenu = false;
396         break;
397     case '3':
398         travelerActionChoice = TravelerAction.TravelerInfo;
399         usingMenu = false;
400         break;
401     case '4':
402         travelerActionChoice = TravelerAction.
403         TravelerInventory;
404         usingMenu = false;
405         break;
406     case '5':
407         travelerActionChoice = TravelerAction.
408         TravelerTreasure;
409         usingMenu = false;
410         break;
411     case '6':
412         travelerActionChoice = TravelerAction.
413         ListTARDISDestinations;
414         usingMenu = false;
415         break;
416     case '7':
417         travelerActionChoice = TravelerAction.ListItems;
418         usingMenu = false;
419         break;
420     case '8':

```

```

417         travelerActionChoice = TravelerAction.ListTreasures;
418         usingMenu = false;
419         break;
420     case 'E':
421     case 'e':
422         travelerActionChoice = TravelerAction.Exit;
423         usingMenu = false;
424         break;
425     default:
426         Console.WriteLine(
427
428             "It appears you have selected an incorrect choice." +
429             Environment.NewLine +
430             "Press any key to continue or the ESC key to quit
431             the application.");
432
433         userResponse = Console.ReadKey(true);
434         if (userResponse.Key == ConsoleKey.Escape)
435         {
436             usingMenu = false;
437         }
438         break;
439     }
440     Console.CursorVisible = true;
441     return travelerActionChoice;
442 }

```

```

443 /// <summary>
444 /// display information about the current space-time location
445 /// </summary>
446 public void DisplayLookAround()
447 {
448     ConsoleUtil.HeaderText = "Current Space-Time Location Info";
449     ConsoleUtil.DisplayReset();
450
451     ConsoleUtil.DisplayMessage(_gameUniverse.GetSpaceTimeLocationByID(
452         _gameTraveler.SpaceTimeLocationID).Description);
453
454     ConsoleUtil.DisplayMessage("");
455     ConsoleUtil.DisplayMessage("Items in current location.");
456     foreach (Item item in _gameUniverse.GetItemtsBySpaceTimeLocationID(
457         _gameTraveler.SpaceTimeLocationID))
458     {
459         ConsoleUtil.DisplayMessage(item.Name + " - " + item.
460             Description);
461     }
462
463     ConsoleUtil.DisplayMessage("");
464     ConsoleUtil.DisplayMessage("Treasures in current location.");
465     foreach (Treasure treasure in _gameUniverse.
466         GetTreasuresBySpaceTimeLocationID(_gameTraveler.SpaceTimeLocationID))
467     {
468         ConsoleUtil.DisplayMessage(treasure.Name + " - " + treasure.
469             Description);
470     }
471
472     DisplayContinuePrompt();
473 }

```

```

474 /// <summary>
475 /// display a list of all TARDIS destinations

```



```
472     /// <summary>
473     public void DisplayListAllTARDISDestinations()
474     {
475         ConsoleUtil.HeaderText = "Space-Time Locations";
476         ConsoleUtil.DisplayReset();
477
478         foreach (SpaceTimeLocation location in _gameUniverse.
SpaceTimeLocations)
479         {
480             ConsoleUtil.DisplayMessage("ID: " + location.
SpaceTimeLocationID);
481             ConsoleUtil.DisplayMessage("Name: " + location.Name);
482             ConsoleUtil.DisplayMessage("Description: " + location.
Description);
483             ConsoleUtil.DisplayMessage("Accessible: " + location.
Accessible);
484             ConsoleUtil.DisplayMessage("");
485         }
486
487         DisplayContinuePrompt();
488     }
489
490     /// <summary>
491     /// display a list of all game items
492     /// <summary>
493     public void DisplayListAllGameItems()
494     {
495         ConsoleUtil.HeaderText = "Game Items";
496         ConsoleUtil.DisplayReset();
497
498         foreach (Item item in _gameUniverse.Items)
499         {
500             ConsoleUtil.DisplayMessage("ID: " + item.GameObjectID);
501             ConsoleUtil.DisplayMessage("Name: " + item.Name);
502             ConsoleUtil.DisplayMessage("Description: " + item.Description)
;
503
504             //
505             // all treasure in the traveler's inventory have a SpaceTimeLo
cationID of 0
506             //
507             if (item.SpaceTimeLocationID != 0)
508             {
509                 ConsoleUtil.DisplayMessage("Location: " + _gameUniverse.
GetSpaceTimeLocationByID(item.SpaceTimeLocationID).Name);
510             }
511             else
512             {
513                 ConsoleUtil.DisplayMessage(
"Location: Traveler's Inventory");
514             }
515
516             ConsoleUtil.DisplayMessage("Value: " + item.Value);
517             ConsoleUtil.DisplayMessage("Can Add to Inventory: " + item.
CanAddToInventory.ToString().ToUpper());
518             ConsoleUtil.DisplayMessage("");
519         }
520
521         DisplayContinuePrompt();
522     }
523
524     /// <summary>
```

```

526     /// display a list of all game treasures
527     /// <summary>
528     public void DisplayListAllGameTreasures()
529     {
530         ConsoleUtil.HeaderText = "Game Treasures";
531         ConsoleUtil.DisplayReset();
532
533         foreach (Treasure treasure in _gameUniverse.Treasures)
534         {
535             ConsoleUtil.DisplayMessage("ID: " + treasure.GameObjectID);
536             ConsoleUtil.DisplayMessage("Name: " + treasure.Name);
537             ConsoleUtil.DisplayMessage("Description: " + treasure.
538                                     Description);
539
540             //
541             // all treasure in the traveler's inventory have a SpaceTimeLo
542             cationID of 0
543             //
544             if (treasure.SpaceTimeLocationID != 0)
545             {
546                 ConsoleUtil.DisplayMessage("Location: " + _gameUniverse.
547                                         GetSpaceTimeLocationByID(treasure.SpaceTimeLocationID).Name);
548
549             }
550             else
551             {
552                 ConsoleUtil.DisplayMessage(
553                     "Location: Traveler's Inventory");
554
555             }
556
557             ConsoleUtil.DisplayMessage("Value: " + treasure.Value);
558             ConsoleUtil.DisplayMessage("Can Add to Inventory: " + treasure
559                                     .CanAddToInventory.ToString().ToUpper());
560             ConsoleUtil.DisplayMessage("");
561         }
562
563         DisplayContinuePrompt();
564     }
565
566     /// <summary>
567     /// display the current traveler information
568     /// </summary>
569     public void DisplayTravelerInfo()
570     {
571         ConsoleUtil.HeaderText = "Traveler Info";
572         ConsoleUtil.DisplayReset();
573
574         ConsoleUtil.DisplayMessage($"Traveler's Name: {_gameTraveler.Name}"
575                                     );
576         ConsoleUtil.DisplayMessage("");
577         ConsoleUtil.DisplayMessage($"Traveler's Race: {_gameTraveler.Race}"
578                                     );
579         ConsoleUtil.DisplayMessage("");
580         string spaceTimeLocationName = _gameUniverse.
581             GetSpaceTimeLocationByID(_gameTraveler.SpaceTimeLocationID).Name;
582         ConsoleUtil.DisplayMessage($
583             "Traveler's Current Location: {spaceTimeLocationName}");
584
585         DisplayContinuePrompt();
586     }
587
588     /// <summary>
589     /// display the current traveler inventory

```

```
579     /// </summary>
580     public void DisplayTravelerItems()
581     {
582         ConsoleUtil.HeaderText = "Traveler Inventory";
583         ConsoleUtil.DisplayReset();
584
585         ConsoleUtil.DisplayMessage("Traveler Items");
586         ConsoleUtil.DisplayMessage("");
587
588         foreach (Item item in _gameTraveler.TravelersItems)
589         {
590             ConsoleUtil.DisplayMessage("ID: " + item.GameObjectID);
591             ConsoleUtil.DisplayMessage("Name: " + item.Name);
592             ConsoleUtil.DisplayMessage("Description: " + item.Description)
593
594             ;
595             ConsoleUtil.DisplayMessage("");
596         }
597
598         DisplayContinuePrompt();
599     }
600
601     /// <summary>
602     /// display the current traveler's treasure
603     /// </summary>
604     public void DisplayTravelerTreasure()
605     {
606         ConsoleUtil.HeaderText = "Traveler Inventory";
607         ConsoleUtil.DisplayReset();
608
609         ConsoleUtil.DisplayMessage("");
610         ConsoleUtil.DisplayMessage("Traveler Treasure");
611         ConsoleUtil.DisplayMessage("");
612
613         foreach (Treasure treasure in _gameTraveler.TravelersTreasures)
614         {
615             ConsoleUtil.DisplayMessage("ID: " + treasure.GameObjectID);
616             ConsoleUtil.DisplayMessage("Name: " + treasure.Name);
617             ConsoleUtil.DisplayMessage("Description: " + treasure.
618
619             Description);
620             ConsoleUtil.DisplayMessage("");
621         }
622
623         DisplayContinuePrompt();
624     }
625
626     #endregion
627 }
```