

```
1 package oregonTrail;
2
3 import java.awt.event.ActionEvent;
13
14 /**
15  * Handles all logic related to traveling. Contains Swing timer for traveling
16  * and updates TravelPanel accordingly.
17  */
18 public class Travel {
19     private OregonTrail oregonTrail;
20     private int milesTraveled;
21     private int milesNextLandmark;
22     private Random rand = new Random();
23     private Calendar date = new GregorianCalendar(1848, 8, 11); // Set to August 11,
1848
24     private Timer timer = new Timer(1000, new ActionListener() {
25         public void actionPerformed(ActionEvent arg0) {
26             travelCycle();
27         }
28     });
29     public static final int MILES_TO_FORT_STRONG = 150;
30     public static final int MILES_TO_FORT_OREGON = 400;
31
32     public Travel(OregonTrail oregonTrail) {
33         this.oregonTrail = oregonTrail;
34         this.milesNextLandmark = MILES_TO_FORT_STRONG;
35     }
36
37     /**
38      * Function that gets called every time the Swing timer runs
39      * @author Corbin Hibler
40      * @date 2024-04-09
41      */
42     private void travelCycle() {
43         // Use GregorianCalendar library as way of keeping track of time, and update
panel accordingly
44         date.add(GregorianCalendar.DAY_OF_MONTH, 1);
45         DateFormat dateFormat = new SimpleDateFormat("MMMM d, yyyy", Locale.US);
46         String formattedDate = dateFormat.format(date.getTime());
47         oregonTrail.TRAVEL_PANEL.setDateText(formattedDate);
48         oregonTrail.TRAIL_MENU_PANEL.setDateText(formattedDate);
49
50         // Generate miles generated and update label
51         int milesTraveledCycle = rand.nextInt(oregonTrail.WAGON.getTravelSpeed());
52         milesTraveled += milesTraveledCycle;
53         oregonTrail.TRAVEL_PANEL.setDistanceTraveledText(milesTraveled);
54
55         // Update miles until next landmark
56         milesNextLandmark -= milesTraveledCycle;
57         oregonTrail.TRAVEL_PANEL.setNextLandmarkMilesText(milesNextLandmark);
58
59         // Calculate new food weight and set accordingly based on the mathematical
models used in the original game [1]
60         // [1] R. P. Bouchard, "Chapter 16: Building the Mathematical Models," in R.
Philip Bouchard; 1st edition (January 28, 2016),
61         int totalFoodWeight = oregonTrail.WAGON.getTotalFoodWeight();
62         int newFoodWeight = (int) (totalFoodWeight -
(oregonTrail.WAGON.getFoodConsumptionRate() * 5));
63         oregonTrail.TRAVEL_PANEL.setFoodText(newFoodWeight);
```

```
64         oregonTrail.WAGON.setTotalFoodWeight(newFoodWeight);
65
66         // Check if we have reached fort
67         if (milesNextLandmark <= 0 && milesTraveled < MILES_TO_FORT_STRONG + 10) {
68             oregonTrail.openPanel(oregonTrail.FORT_STRONG_PANEL,
        oregonTrail.TRAVEL_PANEL);
69             milesNextLandmark = MILES_TO_FORT_OREGON;
70             oregonTrail.TRAVEL_PANEL.setNextLandmarkNameText("Fort Oregon");
71             travelToggle();
72         }
73
74         if (milesNextLandmark <= 0 && milesTraveled > MILES_TO_FORT_STRONG + 100) {
75             oregonTrail.openPanel(oregonTrail.FORT_OREGON_PANEL,
        oregonTrail.TRAVEL_PANEL);
76             milesNextLandmark = 99999;
77             travelToggle();
78         }
79     }
80
81     /**
82     * Function to handle starting and stopping of the travel timer
83     * @author Corbin Hibler
84     * @date 2024-04-09
85     */
86     public void travelToggle() {
87         this.oregonTrail.TRAVEL_PANEL.setNextLandmarkMilesText(milesNextLandmark);
88
89         if (!timer.isRunning()) {
90             timer.start();
91             oregonTrail.TRAVEL_PANEL.btnContinue.setText("Stop Traveling!");
92         }
93         else {
94             timer.stop();
95             oregonTrail.TRAVEL_PANEL.btnContinue.setText("Continue on Trail!");
96         }
97     }
98
99     /**
100     * @return the milesNextLandmark
101     */
102     public int getMilesNextLandmark() {
103         return milesNextLandmark;
104     }
105
106     /**
107     * @param milesNextLandmark the milesNextLandmark to set
108     */
109     public void setMilesNextLandmark(int milesNextLandmark) {
110         this.milesNextLandmark = milesNextLandmark;
111     }
112 }
113
```