```java
package oregonTrail;

import java.awt.*;

/**
 * Main class containing all game logic for switching between panels and instantiating panels
 * @authors Corbin Hibler, Lukas Dunbar, Ray Otto, Ethan Vaughn
 * @date 2024-04-09
 */
public class OregonTrail {

    private JFrame frame;
    private Travel travelState;
    public final Wagon WAGON;
    public final StartupPanel STARTUP_PANEL;
    public final TravelPanel TRAVEL_PANEL;
    public final TrailMenuPanel TRAIL_MENU_PANEL;
    public final FortPanel FORT_STRONG_PANEL;
    public final SecondFortPanel FORT_OREGON_PANEL;
    public HuntingPanel huntingPanel;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    OregonTrail window = new OregonTrail();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the application.
     */
    public OregonTrail() {
        // Create wagon and preload items
        WAGON = new Wagon();
        WAGON.addItem(new Item("Apple Vinegar"));
        WAGON.addItem(new Item("Bacon"));
        WAGON.addItem(new Item("Water"));
        WAGON.addItem(new Item("Bullets"));
        WAGON.setTotalFoodWeight(350);
        WAGON.addItem(new Item("Cast Iron Stove"));

        STARTUP_PANEL = new StartupPanel();
        TRAVEL_PANEL = new TravelPanel(this);
        travelState = new Travel(this);
        TRAIL_MENU_PANEL = new TrailMenuPanel(this);
        FORT_STRONG_PANEL = new FortPanel(this, new ImageIcon(this.getClass().getResource("/images/FortStrong.jpg")));
        FORT_OREGON_PANEL = new SecondFortPanel(this, new ImageIcon(this.getClass().getResource("/images/FortOregon.jpg")));
        initialize();
    }

    /**
     * Opens a new panel and closes previous panel
     * @param panelOpen The panel to be opened
     * @param panelClose The panel to be closed
     * @author Corbin Hibler
     * @date 2024-04-08
     */
    public void openPanel(JPanel panelOpen, JPanel panelClose) {
        frame.getContentPane().remove(panelClose);
        frame.getContentPane().add(panelOpen);
        frame.getContentPane().validate();
        frame.getContentPane().repaint();
    }

    /**
     * Getter method for instantation of Travel class
     * @return the travelState
     */
    public Travel getTravelState() {
        return this.travelState;
    }

    /**
     * Initialize the contents of the frame.
     */
    private void initialize() {
        frame = new JFrame();
        frame.setBackground(Color.GRAY);
        frame.setTitle("Oregon Trail");
        frame.setBounds(100, 100, 839, 544);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.getContentPane().setLayout(new BorderLayout());
        // Always start program maximized!
        frame.setExtendedState(frame.getExtendedState() | JFrame.MAXIMIZED_BOTH);

        // TODO: Separate image logic into its own class with images stored
        // TODO (optional): Make automatic parsing of image files in directory and
        //                  automatically create JLabels from them. Perhaps we could
        //                  also have a CSV file which contains the appropriate scaling for each image.

        // Show logo and team name
        frame.getContentPane().add(STARTUP_PANEL, BorderLayout.CENTER);

        Timer startupTimer = new Timer(StartupPanel.STARTUP_TIME, new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                // After startup screen is finished, display first panel
                frame.getContentPane().remove(STARTUP_PANEL);
                frame.getContentPane().add(TRAVEL_PANEL);
                // The image must be added separate from the panel because LoadWagonPanel uses
                // a Grid Layout, and the image won't fit nicely
                // Update frame
                frame.getContentPane().validate();
                frame.getContentPane().repaint();

            }
        });
        startupTimer.setRepeats(false);
        startupTimer.start();
    }

}

```