CS 230 Final Project: Zombie Trails

By: Maxine Hood and Sherry Xu

USER MANUAL

Introduction:
Zombie Trails is a spin off of the popular 90's game Oregon trail. It is a choose your own adventure kind of game, but with an element of strategy and randomness. The game is set during a zombie apocalypse in the US. The player has been bitten by a zombie, and has 30 days (720 hours) before they turn into a zombie. There is a cure located in Portland Oregon, but the player must reach Portland from Boston without dying or running out of time.

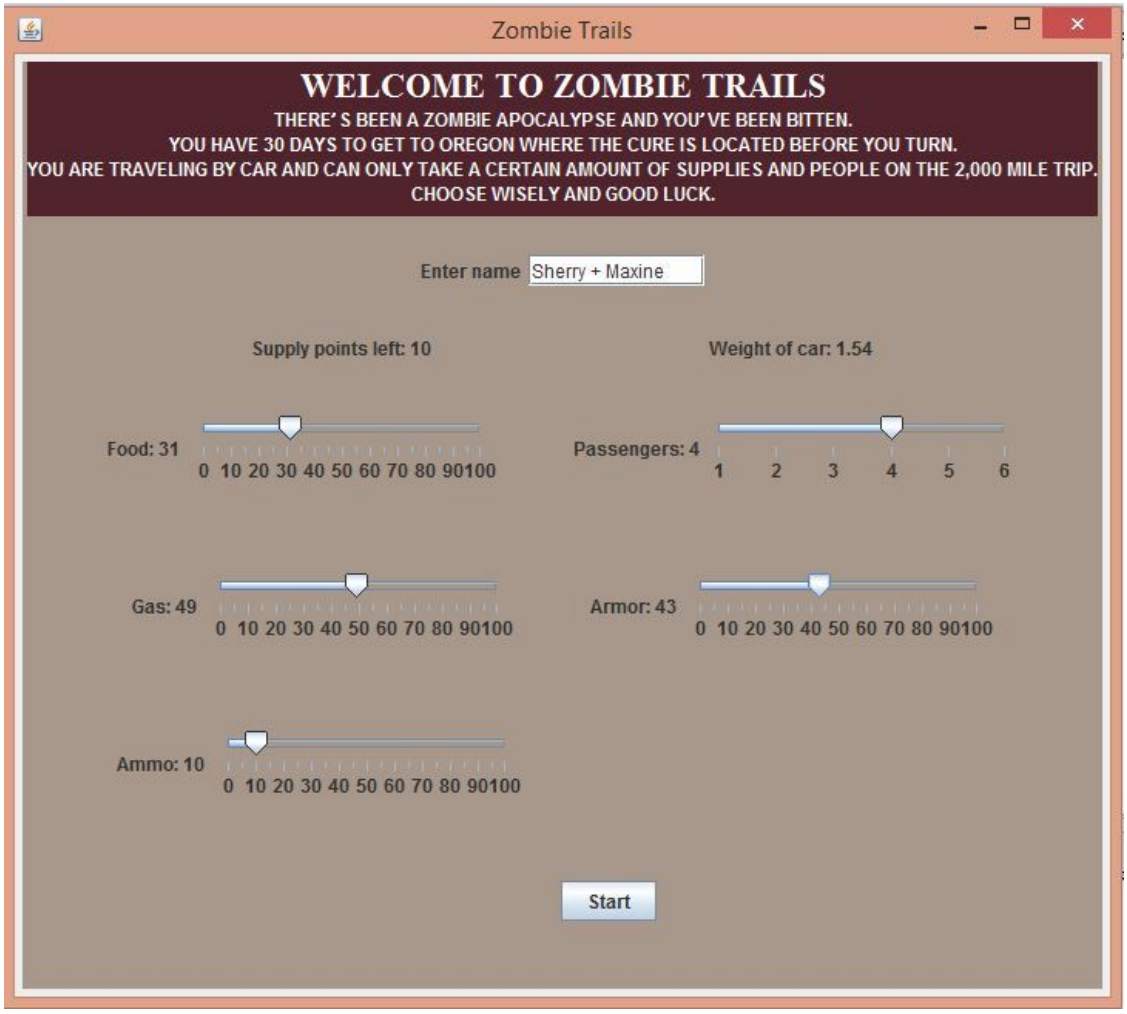
Objective:
Travel across the country and make it through the obstacles and events without having any of their resources reach 0, dying, or running out of time. There are a total of 25 events.

Elements:
Each element of the game is important in its own way. Here is a description of each element in game.

- Food: This stat is necessary to keep the members of the group well fed and healthy! At the end of each event, each passenger consumes some food to survive. If the player runs out of food, they will starve to death.
- Gas: This stat is necessary to keep the car running. In order to advance to the next event, the car needs to consume some gas to travel. If the player runs out of gas, they will be stranded and lose the game.
- Ammo: Ammo is necessary to defend against zombies and other dangers! Certain events may require the use of ammunition. If the player runs out of ammo, they will not be able to defend themselves from zombies and lose the game.
- Armor: Armor is another stat necessary to defend the car against zombies! It is also representative of the health of the car. Some events will damage the armor. Armor is a pretty heavy variable. If the player runs out of armor, their car will break down.
- Passengers: This is how many people the player wants in their group. Some events may kill off members of your group. The player cannot die until all other members of their group have perished. When the player has no other members in their group besides them and are presented with an life threatening event, the player will die.
- Weight: All resources add to the weight of the car. If a car is heavier, it will take more time to travel to the next "event". Measured in tons.

Start Screen:

At the beginning of the game, players are asked to decide which resources to bring and how to allocate them. Players start off with 100 supply points which they then have to distribute to three resources: ammo, food, and gas. The player is also given the option of adjusting the armor on the car and amount of people in the car. Both armor and amount of people are useful but have a large effect on the weight of the vehicle. All resources have an effect on weight. A heavier car moves slower, and may take longer to travel across the country.

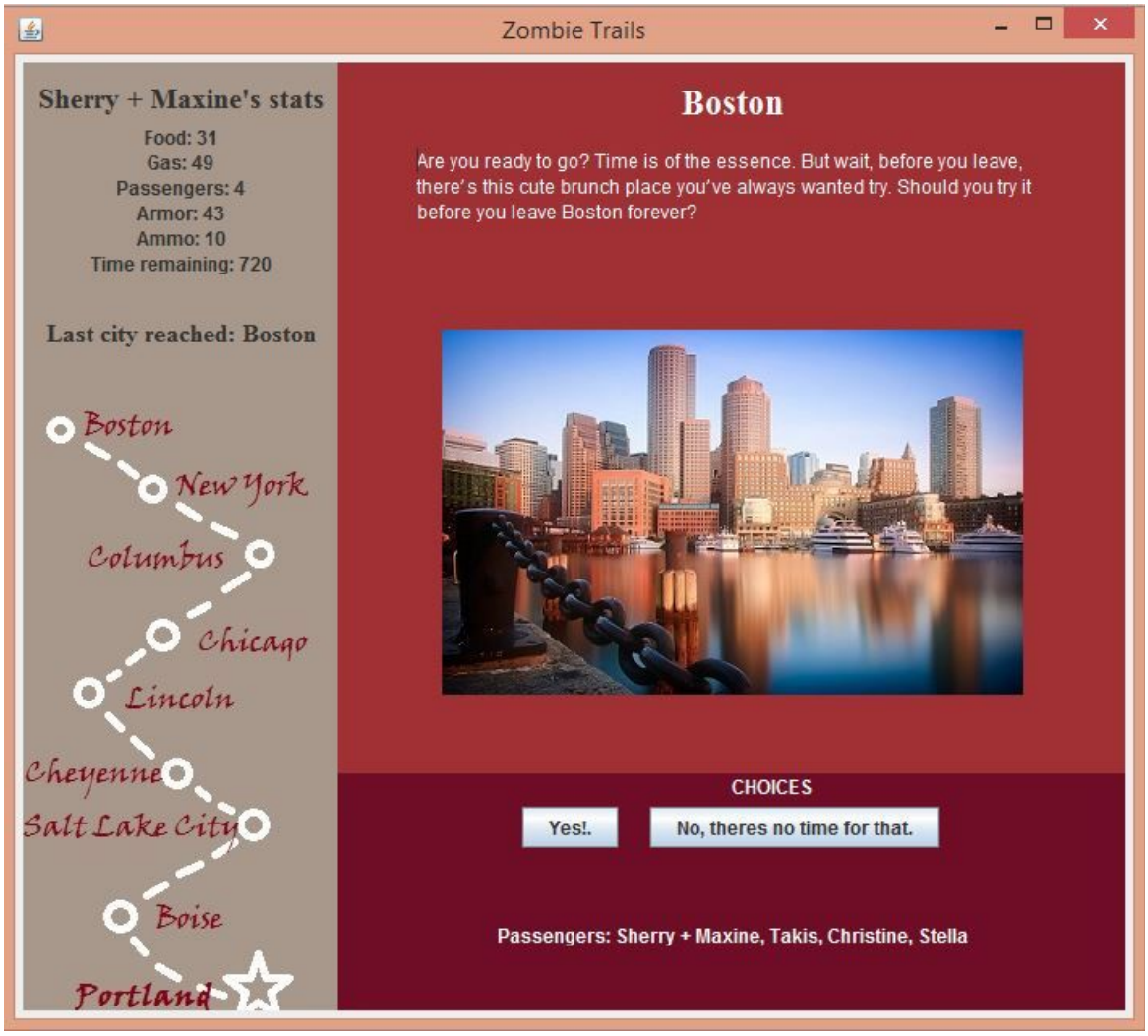


Gameplay:

Each event has decision the player needs to make, and two choices. Each choice has different consequences. There are two types of events in the game: City Events and Random Events.

City Events:

Each city event occurs when the player reaches the next city. These events come in a predetermined order. They also come with a nice picture.



Random Events:

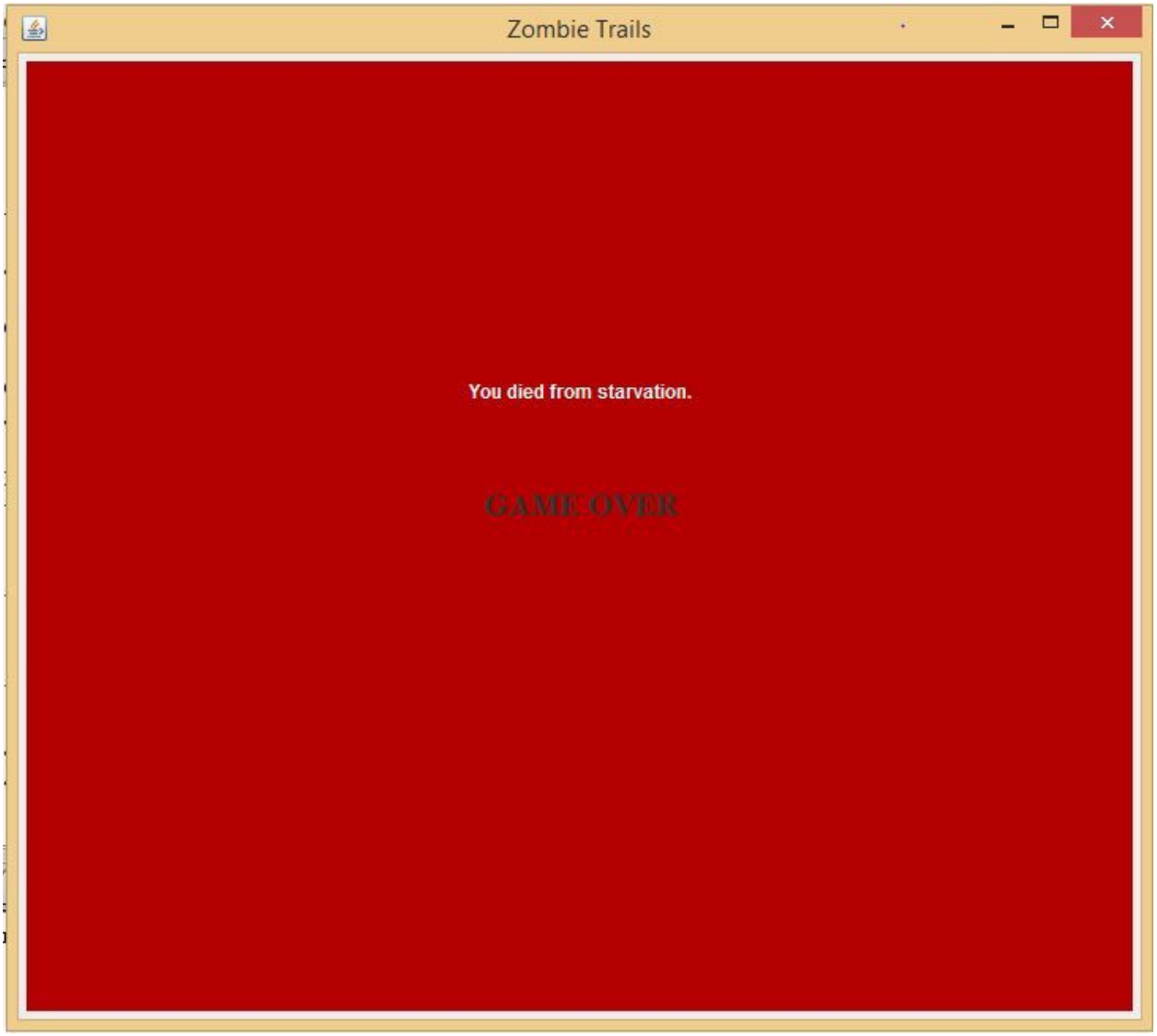
In between the city events, there are random events that occur. The order and which events occur are completely random and different from one gameplay to the next. Some random events have the same decision and choices, but the choices lead to different outcomes.

The game alternates between city events and random events. There are two random events in between each city event.

The panel on the side displays the player's current stats, the city the player most recently passed through, and a map that shows the cities the player needs to traverse before reaching Portland, Oregon. The bottom panel displays the possible choices for each event, and also the people currently traveling in the player's group.

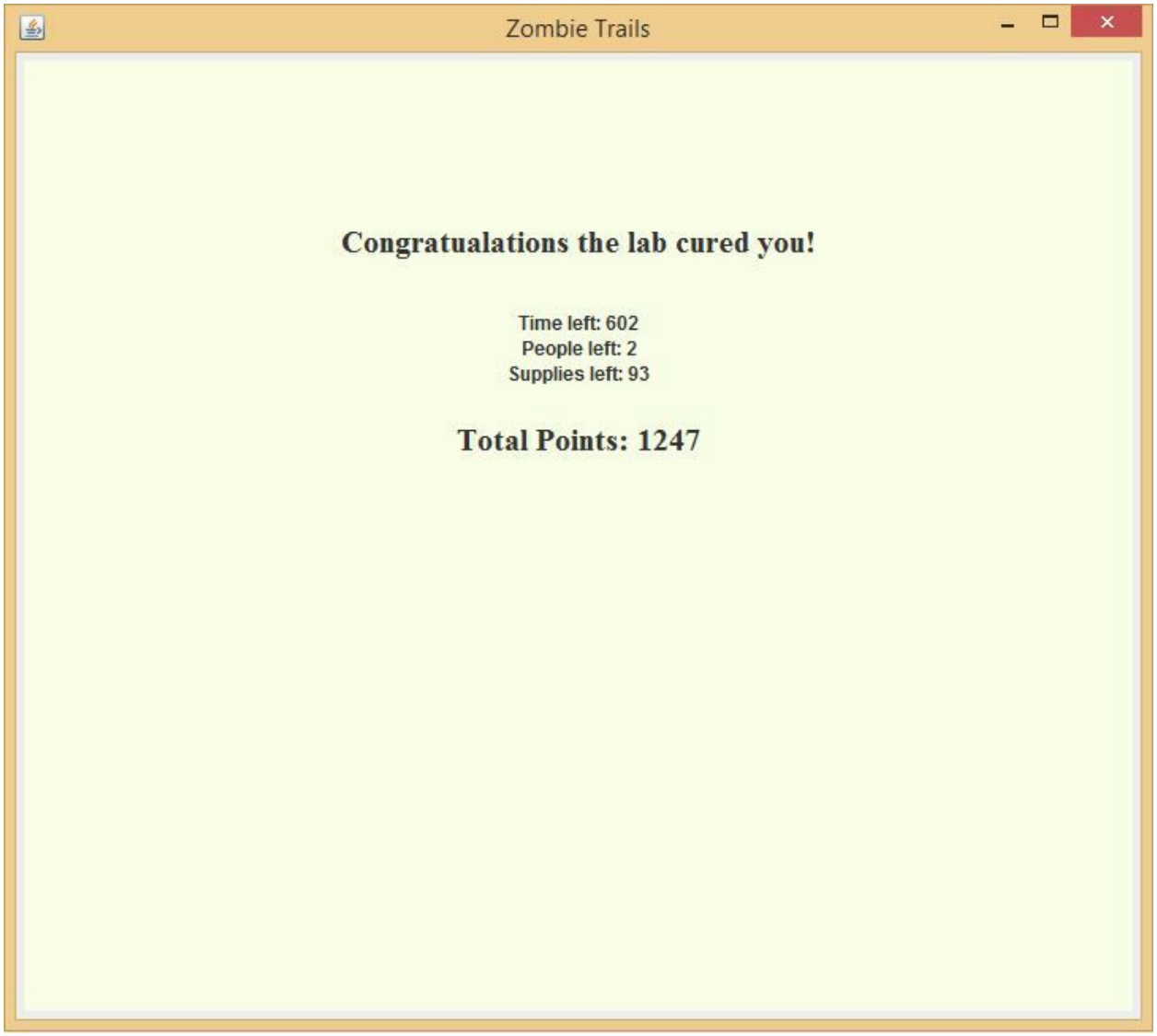
Death Screen:

If the player dies, the game displays the death screen, which lets the player know how they died



Win Screen:

If the player reaches Portland without running out of time or resources, then they win the game! The game displays the win screen, which takes the player's resources, passengers, and time remaining to tally up a score.



TECHNICAL REPORT

ADT choices

Priority Queue: This holds the possible events in the game. We decided to use a priority queue to implement the “random” event generator. As events are read from the file and added to the queue, they are assigned a random priority. This results in the events being lined up in a random order. When the player moves on to the next event, it will pull the event at the front of the queue. Also, since we have more types of random events than the number of random events encountered during the game, the player will play through a different set of random events each play through.

LinkedList: We used a linked list to store our story events (city events). We decided to use this data structure because the city events are traversed in a linear fashion. The player can only advance one city at a time, and once the player reaches a city, they cannot go back to the previous city.

Classes

Car: This class creates a car object which holds the supply and passenger variables. It executes events (updating variables and adding/removing passengers).

ZTEvent: This creates an event and stores event variables based on information given in it's constructor.
ZTStoryEvents: This class reads from a .txt file to create a linked list of a set order of story events.

ZTRandomEvents: This class reads from a .txt file to create a priority queue of random events.Two of these events happen between each city and each game has a different order of random events.

ZTGame: This program is the second page of the game. The player sees their stats and different events pop up which prompt them to make a decision. Their decision affects their stats and can cause them to win or lose the game. Winning or losing directs them to a different panel.

ZombieTrailsGUI This program is the GUI of the game Zombie Trails created for Wellesley's CS230 end project. It initiates the card layout panel which creates the visual game for the user.

> ZTCardLayout: This program initiates a card layout design which calls different panels to display the Zombie Trail game.

ZTStartPanel: This creates the start panel of the game, the first thing the user sees. It gives instructions, takes in their name, and uses sliders to get their supply number choices. A button starts the game, directing the code to PlayPanel.java

ZTPlayPanel: This program is the second page of the game. The player sees their stats and different events pop up which prompt them to make a decision. Their decision affects their stats and can cause them to win or lose the game. Winning or losing directs them to a different panel.

ZTWinPanel: This panel displays when the user has made it Oregon without dieing. A score is calculated based on remaining supplies, people, and time.

ZTDiePanel: This comes up when a supply got to 0 or below or they were killed by a random event. A message displays of how they were killed and a game over message.

Important Methods

Car:

- executeEvent(): given a ZTEvent, this method executes the event, meaning it updates the cars variables based on choice given in params.
- addPassenger(): Method adds a passenger by updating numPassenger var, expanding access to list of names. Also calls helper fcn to update weight of car
- removeRandom(): removes a random passenger that is not the player(unless only player left)
- calculatePoints(): Calculates total points a user has at the end of the game based on vars left
- updateWeight(): Sets weight (in tons) that is used to multiply with time.

ZTEvent

- setAll(): Takes a string array and choice and updates the proper effects var (A or B) with it's values
- printEffect(): Prints what was subtracted and added for user (used in GUI)

ZTStoryEvents

- Constructor: constructor reads from file and creates a linked list of story events.
- nextStory(): method returns next event in linked list.

ZTRandomEvents

- Constructor: constructor reads from file and creates a priority queue of events. Also assigns a random priority
- nextRand(): method returns next random event.

ZTGame

- nextEvent(): Method selects the next event based on a count that keeps track of whether the next event should be a city (linked list) or random event (priority queue). Also sets current City variable.
- checkSupplyPoints(): Returns a boolean based on whether the user has used too many supply points (max 100)
- runGame(): while loop runs the game through the console. Not used in the GUI. Used for backend testing or playing the game without the GUI.
- isGameOver(): Checks if game is over
- checkVars(): Checks if any supplies are or below 0.
- endGameString(): Checks supplies amount and based on whichever the first one is 0 or below reports cause of death. If the user was killed by a random event a generic statement is made.

ZombieTrailsGUI

- ZTCardLayout
- ZTStartPanel
- ZTPlayPanel
- ZTWinPanel
- ZTDiePanel