

Unit 3 Project, Option 1:

Oregon Trail

This assignment is an opportunity to apply your new skills in creating custom functions to implement the Oregon Trail, a classic game! This is the standard project that most students should work on.

1. Overview

We will be recreating Oregon Trail! The goal is to travel from Independence, Missouri to Oregon City, Oregon (2000 miles) by Dec 31st. However, the trail is arduous. Each day costs you food. You can hunt and rest, but you have to get there before winter!

You can try the original Oregon Trail game for free online:

<https://www.visitoregon.com/the-oregon-trail-game-online/>

(continued...)

2. Example Program Output

Welcome to the Oregon Trail!
You are in Independence, Missouri. It is March 1.
You must travel 2,000 miles to Oregon by December 31.
You have 500 lbs of food and 5 health to start.
You eat 5 lbs of food per day and lose 2 hearts per month.
Actions: travel, rest, hunt, status, help, or quit
Good luck!

[...]

What would you like to do? travel
You travelled for 52 miles
You travelled for 3 days
You ate 15 lbs of food

What would you like to do? status
It is day 24 (March 24)
You have travelled 285 miles
You have 3 health
You have 150 lbs of food

What would you like to do? rest
You rested for 2 days
You gained 1 health
You ate 10 lbs of food

What would you like to do? hunt
You hunted for 5 days
You found 100 lbs of food
You ate 50 lbs of food

[...]

Unfortunately, on day 203, you ran out of health and lost the game.
Try again?

(continued...)

3. Basic Gameplay

Player starts in Independence, Missouri on March 1 (day = 0) with 2,000 miles to go, 500lbs of food, and 5 health. They player must get to Oregon by December 31 (day = 305). Implement these functions:

- select_action
 - Prompt the user to take an action (travel, rest, status, help, or quit) and call the appropriate function
 - This function should be called repeatedly using a while loop so that the player can be repeatedly prompted to take an action
- add_day
 - Implement a function which executes all the logic required for a day to pass. Specifically:
 - Player consumes 5lbs of food per day
 - Player health decreases by 1 twice per month
 - One random event happens per month (see section 5)
- travel
 - Implement a function which executes all the logic required to travel along the trail. Specifically:
 - The player should be moved between 30-60 miles(randomly)
 - 3-7 days (randomly) should pass
- rest
 - Implement a function which executes all the logic required to rest. Specifically:
 - Increases health by 1 (up to 5 maximum)
 - 2-5 days (randomly) should pass
- hunt
 - Implement a function which executes all the logic required to hunt. Specifically:
 - Adds 100 lbs of food
 - 2-5 days (randomly) should pass
- status
 - Implement a function that lists the player's food, health, distanced travel, and day. Day can be represented either as a date ("March 2") or as a counter ("Day 2")
- help
 - Implement a function that lists and describes all commands
- quit
 - Exit the game

4. Basic End States

There are three basic end states that must be accounted for in your game. For each end state, please print an appropriate victory or defeat message.

- Victory: If the player reaches Oregon, they win!
- Health Depleted: If the player runs out of health, they lose and the game is over.
- Timer: If the player does not reach Oregon by December 31, they lose and the game is over.

5. Special Events

Implement at least two appropriately creative, original special events the player may encounter. Each of these special events should be incorporated as a function. The player should encounter a special event at a random time every month.

You can implement any special events you can brainstorm. Here are some ideas for inspiration:

- Players get sick and lose a random amount of health
- Players get stuck and a random amount of days pass
- Players encounter a character and gain a random amount of food

6. Style

For each of your functions, please include a comment in this format to make your code easier to read:

```
# Name:  
# Purpose:  
# Inputs:  
# Outputs:
```

7. Resources

In addition to the lab assignments you've completed and class notes you've taken, here are some resources for relevant course topics

- 3.1 – 3.4 – readings: <https://tealsk12.github.io/2nd-semester-introduction-to-computer-science/readings.md.html#associatedreadings/unit3-functions>

8. Challenge

These challenges are opportunities to independently develop your computer science skills by extending your program with interesting, advanced functionality. Note that these challenges are optional and are not eligible for a grade. Please complete the rest of the assignment thoroughly before working on the challenges.

- Your special events can be as creative as you'd like. You might consider having different special events depending on season, location (i.e. mountainous regions, desert regions), or other factors. You might consider special events that involve the user making choices.
- Food consumption could be adjusted to reflect the activity the player is taking: the player might consume more food per day while hunting, but less food per day while resting.
- You can adjust the setting to suit your interests: maybe space travel or other adventures appeal to you!
- You can add or adjust gameplay mechanics as long as all the core gameplay mechanics are present.

(continued...)

9. Rubric

Please submit this assignment on Repl. If you have questions or issues, please don't hesitate to ask any member of the teaching team, or email help@teals.dev

The rubric is a helpful tool to make sure you've completed all parts of the assignment.

Rubric Item	Points
3. select_action	2
3. add_day	5
3. travel	5
3. rest	5
3. status	2
3. help	2
3. quit	1
4. victory ending	1
4. health depleted ending	1
4. time out ending	1
5. special event 1	5
5. special event 2	5
6. style	5
Assignment Total	40