Programming Project Report

Name: Jan Bobda
Date: 10/19/2018

Academic Integrity Statement: I pledge that I have neither given nor received unauthorized help on this programming assignment.

Problem Statement:

The primary goal of this programming assignment was to get experience defining and calling functions. To do this, I created a text based adventure game that takes you from different Mountains until you reach the final Summit. The program is initialized with food and water inputs, and then repeatedly asks the user for directions, and what their reactions are to various events like finding water on the ground and food packages dropping from the skies. The program outputs the Mountain that the player ends up on and also tells the user their current food and water values. For error handling we just had to make sure that if the user entered a direction for a mountain that wasn't there, it would prompt them to enter a different direction.

Design:

The first design step was to design a map on paper for the rooms so it would be easier to implement programmatically. The program initializes the food and water variables, and then calls the first Mountain function to start. From each mountain, you are prompted for either food, water, or both. After this, you are asked to move into a direction, if there is a mountain in the direction you are moving, it will make a function call to that Mountain. This process repeats until you make it to the exit Mountain. From there, the program exits and drops from the call stack, without returning to main. A positive of this design approach was that it was very straightforward to implement and required little debugging. A negative to this approach was that if the user decided, or was lost between mountains, they could eventually fill up the call stack with different function calls until the program crashes due to stack overflow, while this is unlikely, the possibility is still there.

Implementation:

I implemented this program incrementally, I started with the code that was given to us and made sure that I was able to compile it properly on my machine. After that I read through the assignment and completed each of the individual tasks, first designing the maze of mountains on paper, then extending the existing mountain functions, creating new mountains, adding navigation, and finally playing the game. I started with source code given by Professor Gauch that essentially declared and defined utility functions like GainFood and FindWater,

and four Mountain functions. I extended this code by adding more Mountain functions, adding effects to the existing Mountain functions to make the adventure more exciting, and added a navigation system. I worked on this program throughout the week adding small parts whenever I could.

Testing:

To test the program I went through multiple playthroughs testing different inputs to see how the program responded. The normal input would be either a yes or no to the Food and Water events, and one of the four cardinal directions when prompted for directions. I tested the normal cases, and if users wanted to move back and forth between two different mountains. I also tested some special cases like if the user entered something that wasn't a cardinal direction or if they responded with something other than yes or no to a water or food event. Everything seemed to work as expected and the program ran without errors. A sample text file of the game play is included in the submission.

Conclusions:

Overall, the map that I designed on paper was fully implemented in the program and it seems to be running relatively bug free. The programming project was a success, I gained more experience using declaring and defining functions and building this program was fun along the way. Next time, I would start the project sooner so I can be less rushed when I work on it. The project, when you add up the time it took to complete the code that was given to us, and the time writing the documentation, it took roughly an hour and a half. This project, all in all, was a success in design, implementation, and testing.