**Programming Project Report**

Name: Riley Nadwodny

Date: 11/11/19

**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 provide experience working with two dimensional arrays in C++. A partial implementation of the overall project was provided, and the rest had to be implemented. The program contains 3 inputs from the user; the number of rabbits in the simulation, the number of foxes in the simulation, and the number of steps to run the simulation for. The program outputs are the board itself that represents the forest in the simulation, the rabbits on the board, and the foxes on the board. Upon each step of the simulation, the number of rabbits and foxes on the board gets updated after running reproduction and battle simulations, and then the board is printed out again after the simulations are ran; if the user wants to run the simulation 10 times, the board and the rabbits and foxes on the board will be printed out 10 times as well. This programming project required checking array bounds in every rabbit reproduction, fox reproduction, and battle simulation that occurred in the program.

**Design:**

For this project, functions were used primarily to execute code, and then those functions were called in the main body of the program. The first function of the program is to initialize the board, an array, and add rabbits and foxes at random locations in the array. The whole array itself is first initialized to whitespace, e.g. ‘ ‘, and then the rabbits and foxes are added in. The second function draws the board on the screen, initializes a counter for rabbits and a counter for foxes, updates them according to how many R’s and F’s are on the board, and prints out the animal counts at the end. The third function chooses the moves that the rabbits and foxes will do next, and moves them to an adjacent location if the location is in bounds and that location is currently empty. The last provided function actually moves the animals on the board, and changes the letters on the board to upper-case if they are not an upper-case letter. There were 3 functions that needed to be defined for this project: simulate rabbit reproduction, simulate fox reproduction, and simulate fox-rabbit battles.

To simulate rabbit and fox reproduction, both functions had to check if the location where a new rabbit or fox was being placed was within the bounds of the array. If the location was within the array bounds, and the right conditions were met for rabbit or fox reproduction, reproduction would occur. For the rabbits, reproduction occurred 100% of the time. For the foxes, reproduction occurred 50% of the time; to achieve this, the random chance was divided by two, and the remainder determined whether or not reproduction occurred, based on if the remainder was a 1 or 0. To simulate the fox-rabbit battles, the function contains a loop with an if statement. The if statement executes if a rabbit is detected anywhere next to a fox on four sides- directly above, directly below, directly to the left, or directly to the right. If the statement is found to be true, the rabbit has a 10% chance to be eaten by the fox, and wiped off of the board. To do this, a random number is generated and modulo’d by 10, giving a remainder range from 0-9. If the remainder is equal to 1, the rabbit is eaten by the fox and removed from the board.