Tu-An Nguyen
tunhnguy@ucsc.edu
01/12/2021

CSE 13S Spring 2020
Assignment 1: The Garlic Game
Design Document

The Garlic Game is played with a circle of 2 to 10 vampires. All vampires start off with 3 lives and die when they run out. For every round, each vampire rolls a pair of 6 sided dice. The vampire with the first lowest roll during a round is forced to eat garlic and loses a life. If a vampire rolls 2 sixes, the vampires on the left and right of them resurrect if they're dead and sparkle if alive. In both cases, the vampire gains a life. Those who get resurrected do not roll during the current round. The last vampire standing is crowned the winner.

In this lab, I am simulating The Garlic Game in C using a list containing names of vampires and specific dice rolls. I use a pseudorandom number generator and prompt the user with a random seed to simulate the dice rolls. Each of the vampires' rolls are then recorded in the stdout with the round number, vampire's name, and the name of their dice roll. The program also records who eats garlic, dies, resurrects, and sparkles. The last line of the output states the winner.

Pseudocode

```
Main:
      num_vampires = user input
      random_seed = user input
      print num_vampires and random_seed

      vampires = array of names from names.h
      round_rolls = array of rolls, same size as vampires
      lives = array of lives

      while more than 1 vampire alive:
            print round number
            for i in vampires:
                  if alive:
                        first = roll_dice(random_seed)
                        second = roll_dice(random_seed)
                        print rolls[first, second] from names.h
                        round_rolls[i] = first + second
            vampire with lowest roll loses a life
            for every 2 sixes rolled:
```

```
                vampire on left and right gain a life
    print winner
```

Design Process

I was trying to figure out how I would store all of the information of each player: lives, names, rolls. At first, I was thinking I can create a Player object with name, lives, and rolls attributes, but then I figured I can just store each attribute in arrays because there is a fixed number of players for each game, and I can identify a player by their position in the array.

With this in mind, I drafted some pseudocode above and started implementing it from a top down perspective. I added a few functions to help clean up my code: `game_continue`, `dice_roll`, `left`, and `right`. I didn't add left and right until later on in my implementation. I actually stumbled across these functions by rereading the assignment specifications.

I implemented the assignment in this order:

1. Initialize `vampires` array and random seed integer and get user input
2. Store names into `vampires` array
3. Implement `game_continue` function
4. Add `dice_roll` function
5. Implement outputs for each round
6. Decrement lives of the player with the lowest roll each round
7. Implement midnight rolls
8. Error handling for user input
9. Documentation

I would say implementing the midnight rolls took the longest time as it was the most complex mechanism of the game and utilized the `left` and `right` functions.