Final Project – Hunt the Wumpus

My final project is a game. The game is called “Hunt the Wumpus”, it was originally made by Gregory Yob in 1973 using BASIC. This game stood out among others like it because it used an unorthodox mapping system: the map was a system of rooms arranged like the vertices of a dodecahedron (each one having three other nodes to connect to). The game has been remade in various languages and with augmented features. The original is a text based game and that is how I will make it, though mine will vary somewhat from the original.

For the purposes of this project, the game entities will be static, not randomized. A map with locations of the entities will be attached.

About the Game:

Your mission, should you chose to accept it, is to hunt the Wumpus. The Wumpus lives in a cave system with 20 rooms. Each room has three tunnels connecting to other rooms.

HAZARDS:

- Bottomless Pits - Two rooms have bottomless pits in them.
    - If you enter, you fall into the pit (and lose!)
- Giant Bats - Two other rooms have giant bats.
    - If you enter, the bat grabs you and Takes you back to the start.

WUMPUS:

The Wumpus is not bothered by the hazards (he has sucker feet and is too big for a bat to lift). Usually he is asleep. Two things wake him up: entering his room and shooting an arrow.

If you wake the Wumpus, he will eat you (and you lose!)

YOU:

- Each turn you will have the option to move or to shoot a crooked arrow
- MOVING: you may take any tunnel connecting to your current room
    - (be warned, not every room is safe)
- ARROWS: You have ONE arrow (MAKE IT COUNT).
    - Your arrow can be shot into any adjacent room.

- If the arrow hits the Wumpus, you win.
- If you miss, you will wake the Wumpus and he will eat you!

WARNINGS:

- You will be given warning messages when you approach hazards. The console will output the following when a hazard is in an adjacent room:
  - Wumpus: "You smell the unmistakable stench of the Wumpus"
  - Bat: "You hear the flapping of large wings"
  - Pit: "You feel an ominous breeze"

Implementation:

Whole program will be while loop: after completing each menu selection, user will be returned to menu (clear screen and reprint title and menu). Title and menu will be string arrays and printed with same function.

The program will use a switch case menu to offer users the following options:

1. View Instructions
2. View Statistics
3. Play Game
4. Exit Program

Instructions

- Read in instructions from file if needed (first time only)
- File can be found in directory with source, named "instructions.txt"
- Print text to console

Statistics

- Read in statistics from file if needed (first time only)
- File can be found in directory with source, named "stats.txt"
- Print stats to console

Play

- Warn that exiting program early will not save statistics, and inform user where stats may be viewed

- Read in statistics from file if needed (if user did not load in main menu)
- Load map if needed (first time only)
- Start game loop:
    - Check if player is in room with Wumpus, Bat or Pit
        - Order is important: Wumpus trumps all, but Bat can save player from Pit.
    - Check for nearby hazards and print warnings
    - Print current room and valid exits
    - Prompt for game action: ([M]ove Player or [S]hoot arrow
    - Get valid target (which room to move or shoot into)
    - Move player or check win/lose if arrow is shot
    - Game returns:
        - -1 = Eaten by Wumpus (includes missing with arrow)
        - 0 = Fell in Pit
        - 1 = Killed Wumpus
    - If no win/lose, take next turn
- Update statistics (calculate new stats and output to file)

Exit: Simply prints thank you message and exits program

Input / Output files:

instructions.txt – text file with game info from “About the Game” above. Read in as a string array length 37.

stats.txt – text file with the game statistics. The format is flexible. Input algorithm ignores text and only reads in the numbers in order:

1. (INT) Number of games played
2. (INT) Number of times the Wumpus won
3. (INT) Number of times the Player fell in Pit
4. (INT) Number of times the Player won
5. (DOUBLE) Wumpus win rate (%)
6. (DOUBLE) Pit win rate (%)
7. (DOUBLE) Player win rate (%)

8. (INT) Number of moves in shortest game
    - Read in and stored as double, but type casted and printed as integer
9. (INT) Number of moves in longest game
    - Read in and stored as double, but type casted and printed as integer
10. (DOUBLE) Average number of moves per game

First input stored in int variable; 2^{nd} – 4^{th} inputs stored in an int array; 5^{th} – 7^{th} inputs stored in a double array; 8^{th} – 10^{th} stored in double array.

I will store them this way because of the logical grouping in the input/output file. Each array is 3 elements and is displayed in a table with text labels and values. Also, it's more convenient to pass 4 stats variables around instead of 10. Even though the last three numbers are not of the same type, logically they are grouped together for output, I will read them in as doubles and type cast the first two back to ints when outputted.

For output, I will print out a one line heading for the file; print out a label and the first stat on the same line; print out the next two arrays in a 3 x 3 table (with text headings and a title); the last array will be printed in a 2 x 3 table (title, headings and values). Empty new lines should be printed to separate each section of stats.