Game-Project Report

By: Yaseen Moawia
HW ID: H00462816

Introduction

The assignment called for the design of a game or the implementation of an existing game using the C programming language. This report discusses the design and the rules of a text-based version of the well-known snake game. The goals of this project were to:

- Create a snake-game that functions similar to the original version.
- Include new features such as different difficulty levels and a score counter.
- Make sure game code is clear and well-commented.
- Make sure the game is easy to run and play.

Design

Including libraries, declaring variables, initializing terminal screen and enabling keyboard input

Four libraries were required for the design of this game: stdio.h, ncurses.h, stdlib.h and unistd.h. To start, the terminal screen where the game was to be displayed was initialized and a function was called to get the maximum height and width of the screen needed as different users will run the game on different sized terminals followed by the declaration of the game variables as integers. Keyboard input was enabled using the keypad() function.

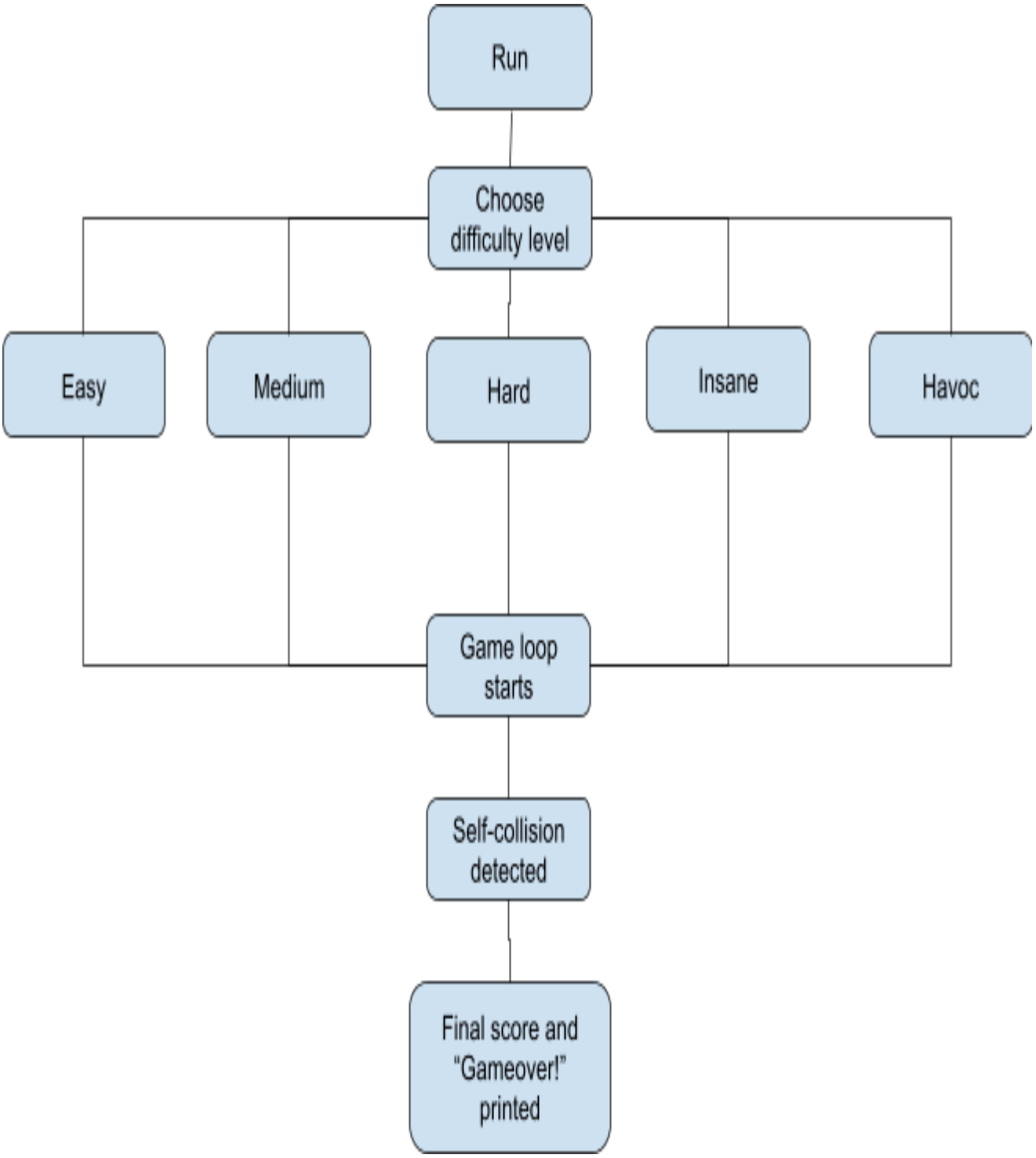
Starting screen

The game starts by displaying the difficulty selection screen printed using the mvprintw () function.

Infinite loops

Two infinite loops were made by using the While(True) function, one for the difficulty selection mechanism and one for the main playing mechanism. Inside the first loop a simple if function was used to to get users difficulty selection followed by if and else if functions to vary the snake's speed thus making it easier or harder based on the user's input. The second loop fetches user input and sets snake direction based on users input using the arrow keys referred in the code as (KEY_UP, KEY_DOWN, KEY_LEFT and KEY_RIGHT) by using wgetch(), if() and for() functions.

Game logic flow-chart



How to play

After the game is run the difficulty level is chosen using the number keys 1 for Easy, 2 for Medium, 3 for Hard, 4 for Insane, 5 for Havoc!. Afterwards, the main game loop begins allowing the snake to start moving, the food to spawn and the score counter to begin. In order to control snake direction the four arrow keys are used.

The Rules of the game are pretty straightforward
1- Each time food is consumed the snake grows by one segment and the score increases by 10.
2- The only way to lose is for the snakehead to collide with any other part of the snake.

Conclusion

In conclusion, the finished game proved to be functional, quite enjoyable and met the previously set goals & expectations. Overall the task was challenging, educational and provided a lot of experience in the programming space.

References

Ahmed Abuharthieh (2024). *Snake Game in C (Less than 100 lines of code!)*. [online] YouTube. Available at: https://www.youtube.com/watch?v=XTPAllcEL4s [Accessed 16 Apr. 2025].

Nir Lichtman (2023). *Making Minimalist Snake Game in C on Linux*. [online] YouTube. Available at: https://www.youtube.com/watch?v=SFtWBVMdAOA [Accessed 16 Apr. 2025].

Opengroup.org. (2025a). . [online] Available at: https://pubs.opengroup.org/onlinepubs/7908799/xcurses/curses.h.html [Accessed 16 Apr. 2025].

Opengroup.org. (2025b). *getch*. [online] Available at: https://pubs.opengroup.org/onlinepubs/7908799/xcurses/getch.html [Accessed 16 Apr. 2025].

Github copilot. [online] Available at: https://github.com/Keyboard-Warrior-05/Game-Project-H00462816/blob/main/snake.c