

Game-Project Report

By: Yaseen Moawia
HW ID: H00462816

Course code and name: B37VB	
Type of assessment: Game project	Individual
Coursework Title: Game project Dubai	
Student Name: Yaseen Moawia Mohamed Ahmed	
Student ID Number: H00462816	

Declaration of authorship. By signing this form:

- I declare that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.
- I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the [University's website](#), and that I am aware of the penalties that I will face should I not adhere to the University Regulations.
- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism

Student Signature (*type your name*): Yaseen Moawia

Date: 18/4/2025

Copy this page and insert it into your coursework file in front of your title page.
For group assessment each group member must sign a separate form and all forms must be included with the group submission.

Your work will not be marked if a signed copy of this form is not included with your submission.

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.

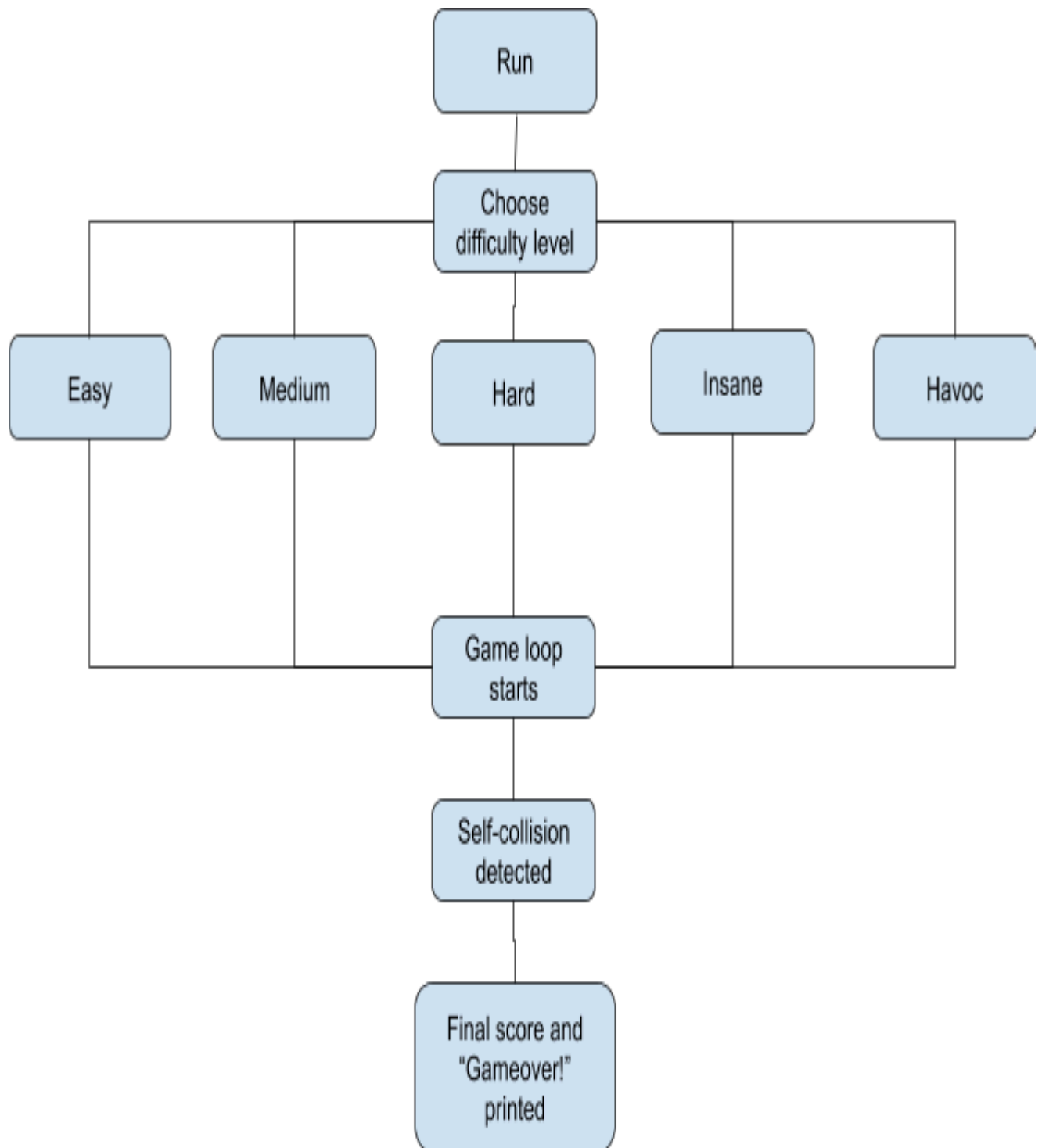
Starting screen

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 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>