**Student Declaration of Authorship**

| **Course code and name:** | B37VB - Praxis Programming - 2024-2025 |
| --- | --- |
| **Type of assessment:** | **Individual** |
| **Coursework Title:** | Game Project Dubai |
| **Student Name:** | Shteinberg Andrei |
| **Student ID Number:** | H00468125 |

| **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](https://www.hw.ac.uk/uk/students/studies/examinations/plagiarism.htm), 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](https://heriotwatt.sharepoint.com/sites/skillshub/SitePages/Academic-Integrity-and-Plagiarism.aspx)   **Student Signature** *(type your name): Shteinberg Andrei*  **Date**: *13/04/2025* |
| --- |

**Heriot Watt University**

School of engineering and physical sciences

Electrical and electronics engineering department

Robotics

Andrei Shteinberg

ID Number: H00468125

# Abstract

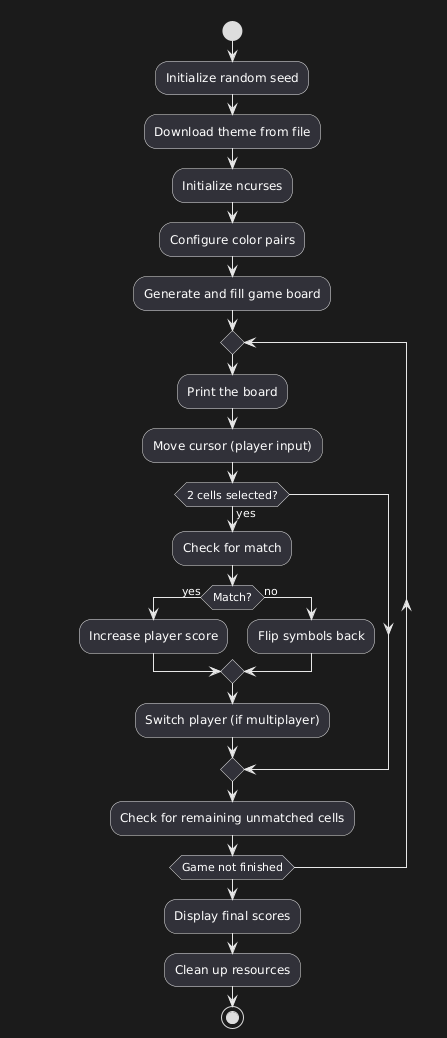
This is a real time memory based game written in C language and ncurses library. It can be played against a real player, bot or just practice mod. There is also a customization menu and settings are saved even after restarting the game.

# Introduction

This project was quite challenging for me in terms of connecting the library, setting up linux and its dependencies. The main objective of this program is to train short term memory and have fun.

# Methodology

All the code for this game was written in one file named “main.c”. The flow of the game is not very complex, logic is simple as you can see in appendix A.

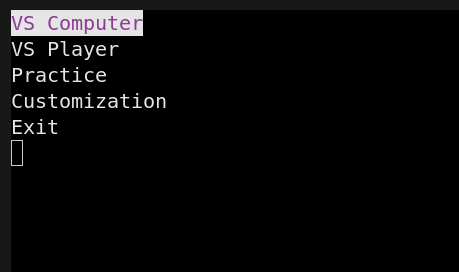


Appendix A

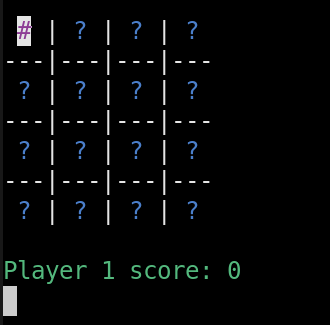
Some difficulties were faced with setting up general logic, how functions should interact with each other and what values should return. Dynamic memory allocation was new for me and I faced some difficulties of not tracking the length of the array which brought errors like memory leaks and out of bounds writings. Also during this project I learned a lot about reading and writing to files, opening them with different access levels like read only, replace, append etc.

In general the entire program looks very minimalistic. compared to the modern apps when their interface is very overloaded with various functionality, a simple console game feels as a breath of fresh air.

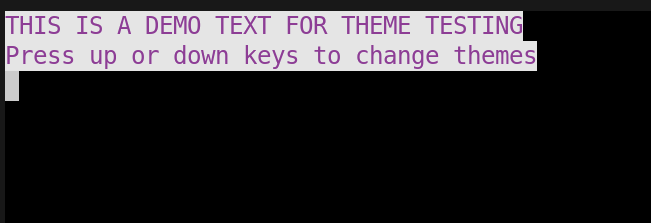
# Game overview



Main menu



Game interface



Customization menu

# Conclusion

The development of this terminal-based memory game highlights the practical application of C programming in creating interactive and visually engaging console applications.Overall, the project not only reinforces core programming concepts such as control structures, memory management, and user interaction but also demonstrates how these elements can be combined to build a fully functional and entertaining application.