**Mini-Project Report**

****

**DOCUMENTATION**

**Project Name: Snake Game**

**Development Language: C**

**Name: Sejal Patil**

**Table of Content**:

**No.** …………………. **Content**

1 ………………… Introduction

2 ………………… Requirement

3 ………………… Objective

4 .....………… Flow chart

5 ………………… Structure Diagram

6 …………………… Input and output Display

7 …………………… Future scope

8 …………………… Reference

**Introduction**

The game called “snake” or “Snake Game” typically involves the player controlling a line or snake, there is no official version of the game, so gameplay varies. The most common version of the game involves the snake or line eating items which make it longer, with the objective being to avoid running into a border or the snake itself for as long as possible.

The player loses when the snake either runs into a border or its own body.

Because of this, the game becomes more difficult as it goes on, due to the growth of the snake.

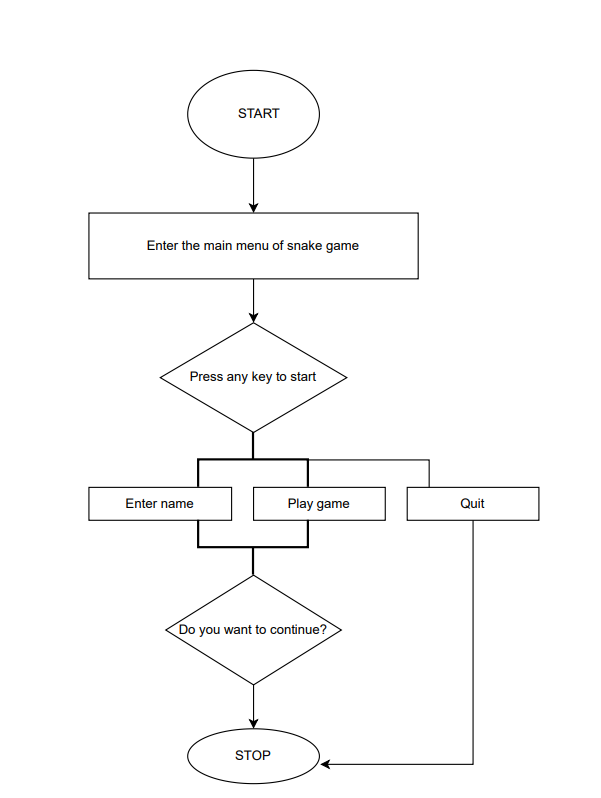
Nokia has installed the “Snake Game” on many of its phones. The game is also available on several websites, including YouTube, allowing viewers to play the game while a video load.

**Requirements**

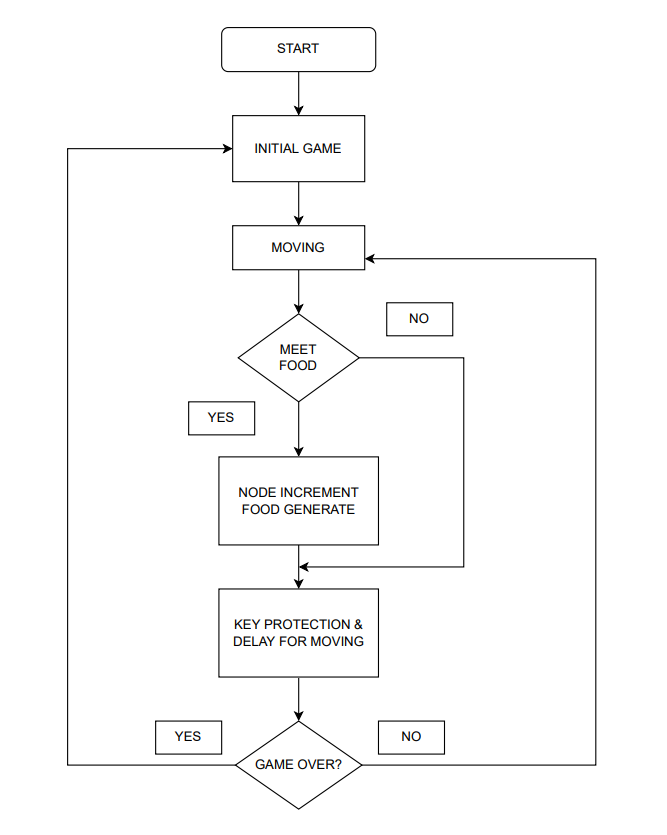
* Software Requirement
* Operating system: WINDOWS
* Application Software: Code-Blocks
* Language: C-Programming

**Objective**

* This Project in the c language of snake Game is a simple console application with very simple graphics. In this project, you can play the popular” Snake Game” just like you played it elsewhere, You have to use the up, down, right, or left arrows to move the snake.
* Food is provided at the several co-ordinates of the screen for the snake to eat. Every time the snake eats the food, its length will be increased by one element along with the score.
* It isn’t the world’s greatest game, but it does give you an idea of what you can achieve with a relatively simple c program, and perhaps the basis by which to extend the principles and create more interesting games of your own.

**Flow Chart**

**Structure Diagram**

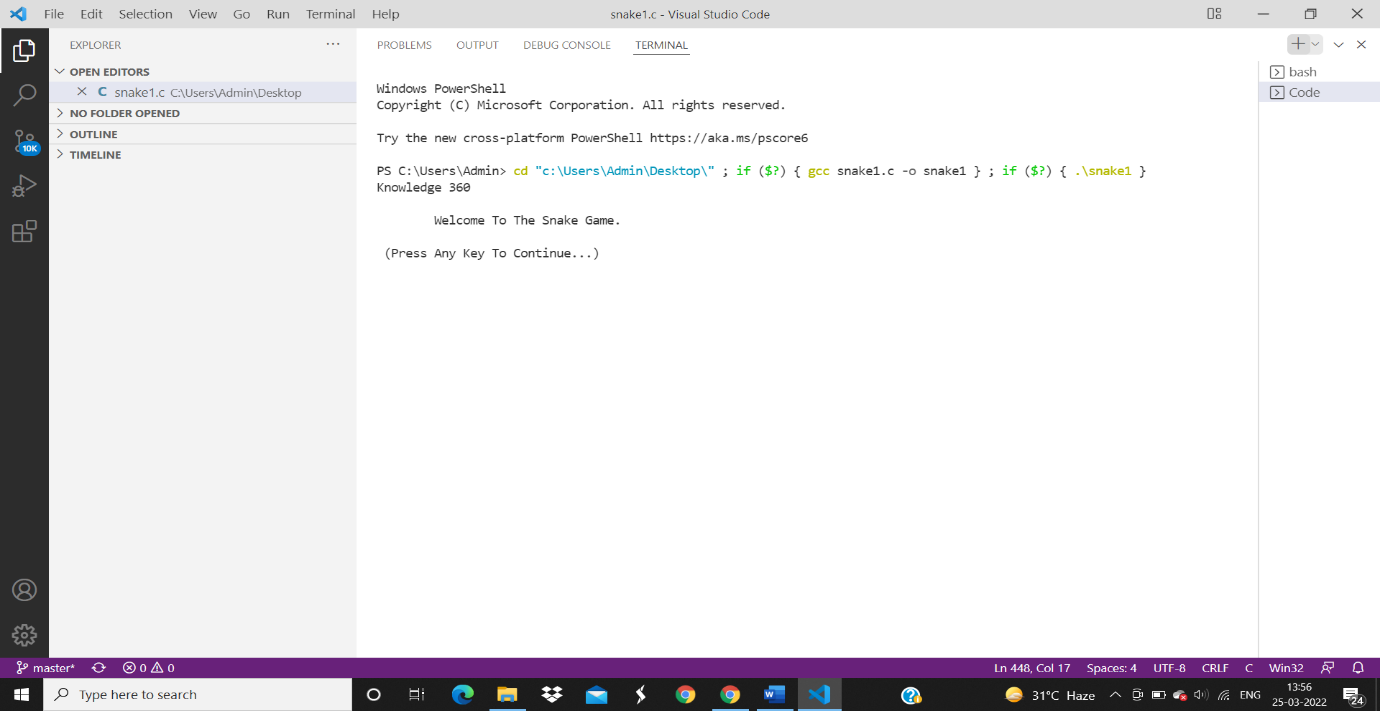
****

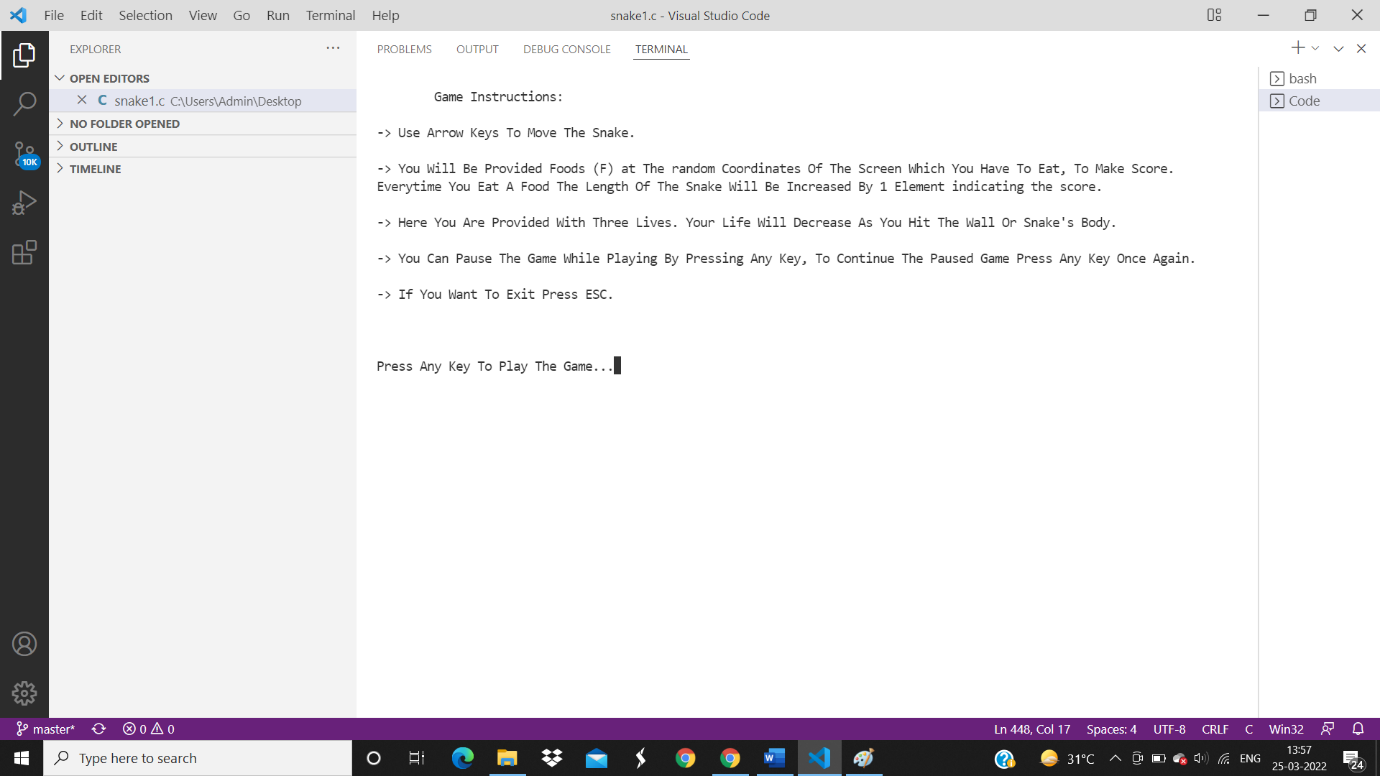
System

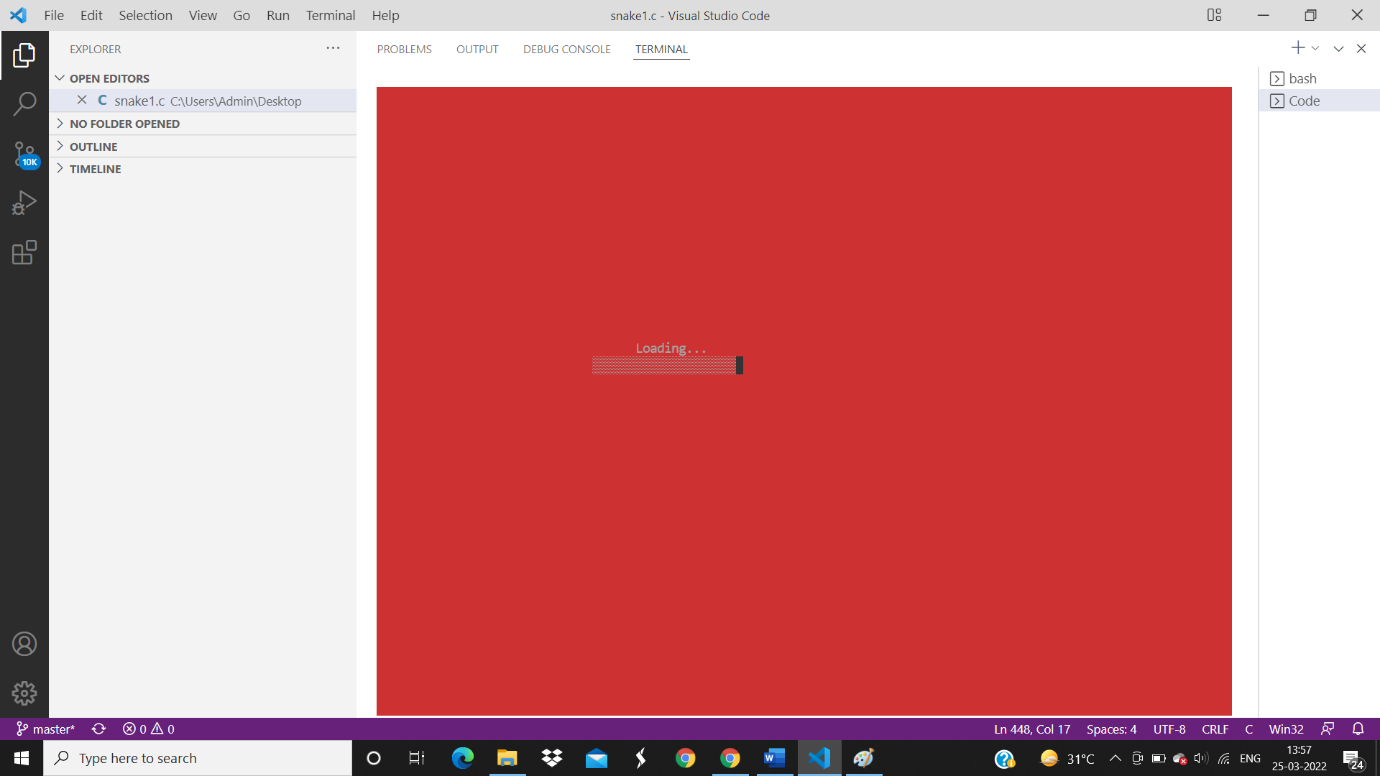
The system is the skeleton view behind the GUI part of a game. The system defines the working methodology of the game and shows the components, their relationships, and how they evolve to make the game work.

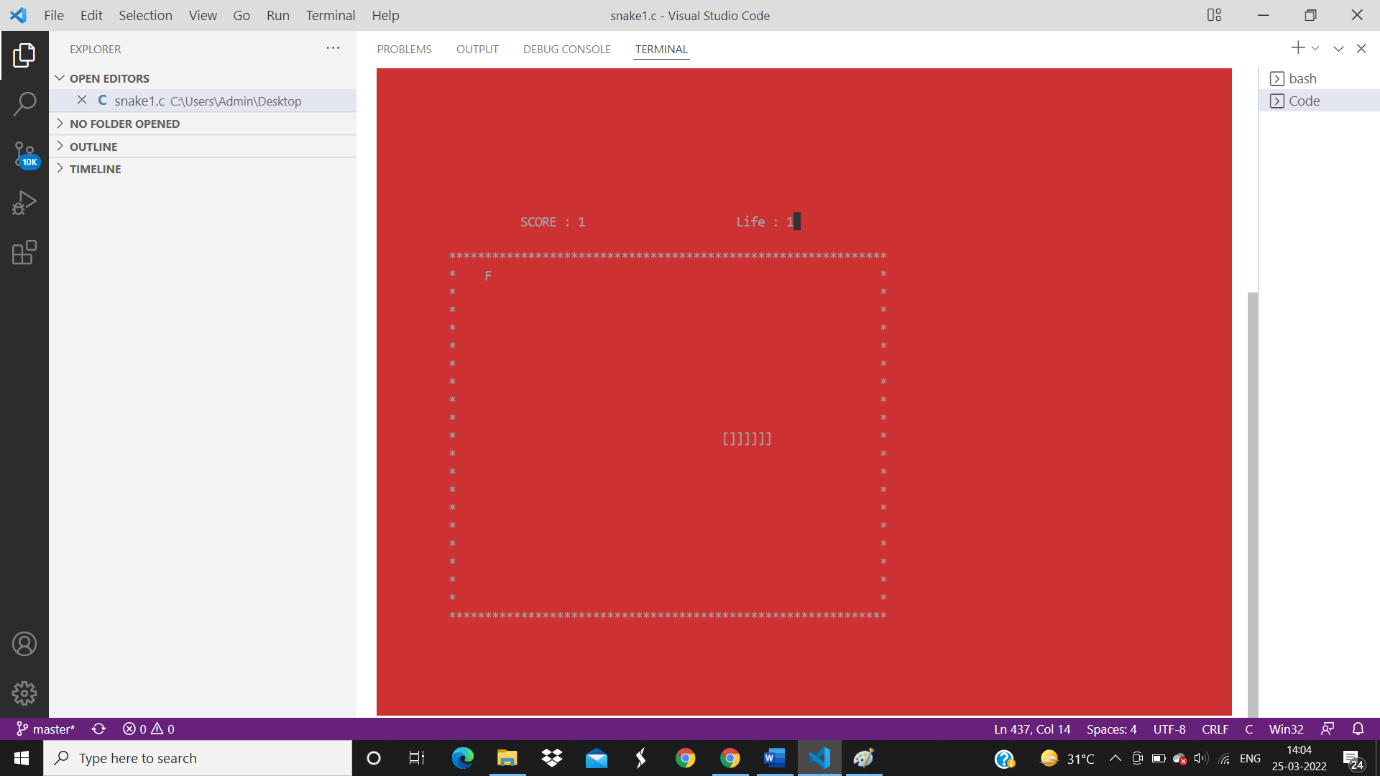
* The snake is represented with an [](asterisk) symbol.
* The fruit is represented with an F (at sign) symbol.
* The snake can move in any direction according to the user with the help of the keyboard (⬇️, ⬆️, ⬅️, ➡️ keys).
* When the snake eats a fruit, the score will increase by 1 point ✔️.
* The fruit will generate automatically within the boundaries 🍓 🍒 🍇.
* Whenever the snake will touch the boundary, the game is over ❌.

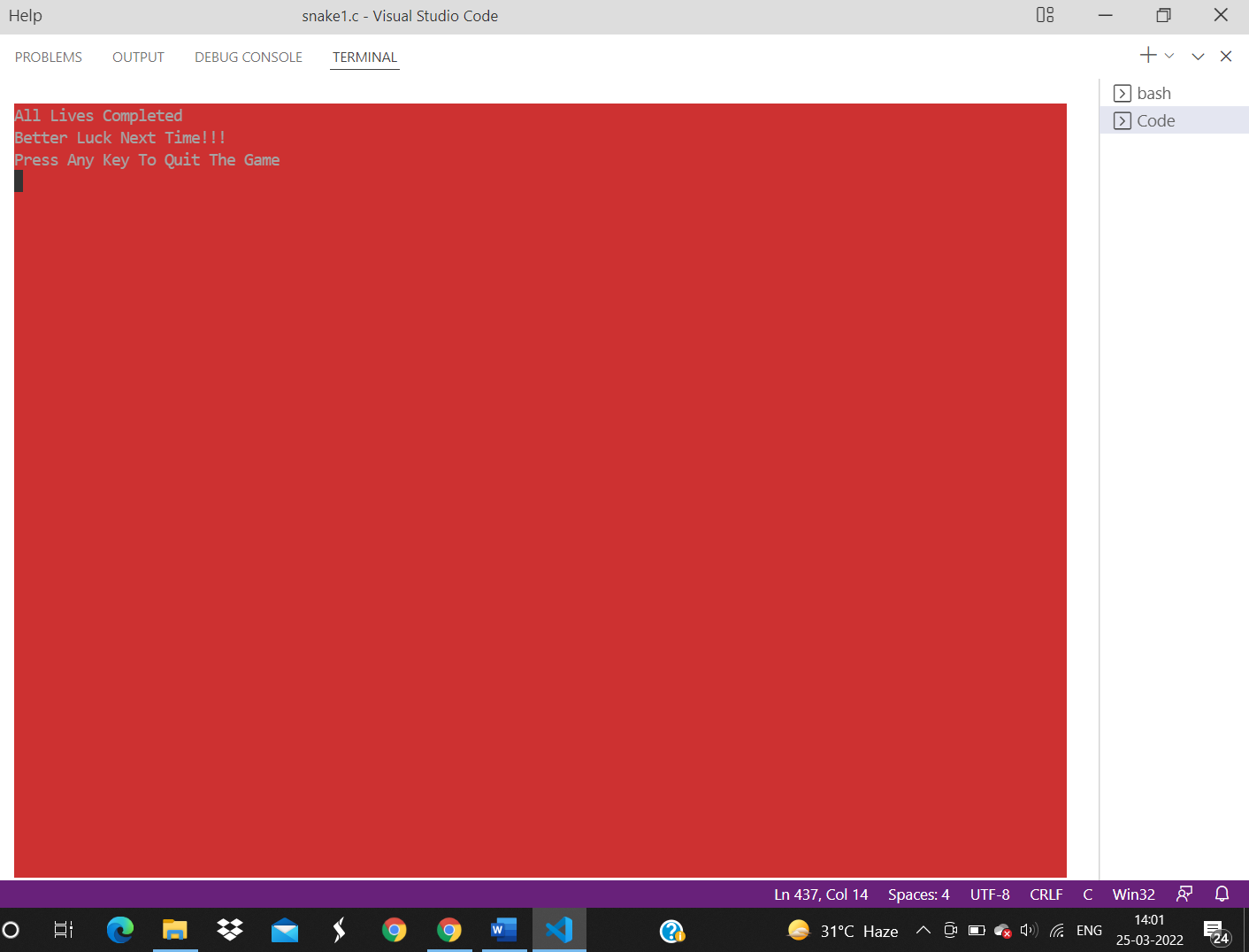
**Input and output Display**

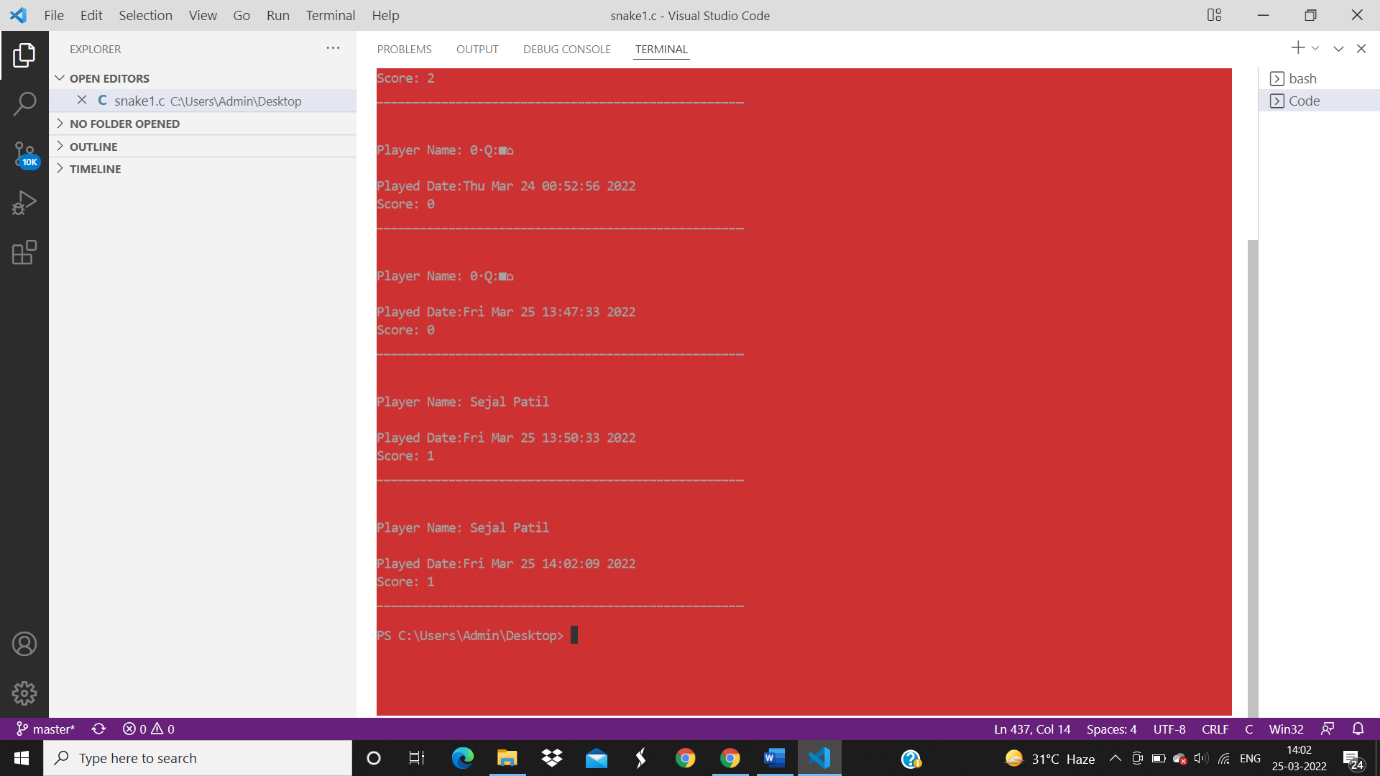
****

****

****

****

****

****

**Functional Requirements:**

1. WASD keys for Up, Left, Down, Right change the snake's direction.
2. The sake has a velocity in the direction it is moving. That velocity increases with time. (Provide function of time for detailed requirement.)
3. Randomly place power-ups:
   1. Slow snake velocity.
   2. Food makes the snake longer.
   3. Bonus targets gain bonus points.
   4. etc.
4. If the sake collides with the snake the round ends and a new round starts.
5. At the end of rounds, the score is registered as the player’s highest if it is better than any recorded before. The highest registered score is tracked between all players on a leaderboard.
6. Log in to identify the player. (Details should include username/password requirements, is two-factor auth needed, should obscene words be allowed as usernames? etc..?)
7. Options menu to set music and sound effects volumes. Color scheme?

**Non-Functional Requirements:**

1. On cutting-edge computers, as of 2010, it should get a 40 FPS refresh rate.
2. The leaderboard needs to support 9 million concurrent users.
3. The leaderboard must be updated within 2 min after a new high score is registered by a play.

This is the idea. For real game development documentation, you would also want some discussion about monetization and promotion. You would want some details about the look and feel. (Is it neon colors, does it use a disco beat to the music/effects, should be it slower and more casual or fast and intense? Etc…)

A good game design document is more than requirements. But, the requirements are a major part of it. Functional requirements are about what it should do. Non-Functional requirements are about how well it should perform on which platforms.

**Future scope of the project**

* Our project will be able to implement in the future after making some changes and modifications as we make our project at a very low level. so the modification that can be done in our project are:
  + It can be made with good graphics.
  + We can add more options like Top score and player profile
  + We can add a multiplayer option.

**Reference**

<https://www.codewithc.com/mini-project-in-c-snake-game>.

<https://www.niit.com/india/knowledge-centre/Game-development-in-C><https://www.youtube.com/watch?v=LYOGym_9K_E>

<https://joombig.com/sqlc/Mini-project-Snake-game-using-C-tutorials-advance>

<https://www.geeksforgeeks.org/snake-game-in-c/>