**PROJECT REPORT**

**MEMBERS’ INTRODUCTION**

|  |  |
| --- | --- |
| Project Title: | **Snake Game (GUI)** |
| University Name: | FAST-NUCES, KARACHI. |
| Course Code | CL1002 |
| Course Teachers | Miss Ramsha Iqbal |
| Lead worker | Ali Akhter |
| Other participants | |  |  |  |  | | --- | --- | --- | --- | | S.No. | Name. | ROLL NO. | DISCIPLINE | | 1. | Rehan | 24K-0707 | BCS-1A | | 2. | Umer | 24K-0697 | BCS-1A | | 3. | Ali | 24K-0809 | BCS-1A | |

Date of Submission: 04/December/2024

**Motivation:**We chose to build this project as snake game is an important part of our childhood memories and building our childhood game must be the first step in starting our journey in development site especially game development. Furthermore, building a gui based project also enabled to break the cli barrier, gaining experience on both front and back-end development, exponentially benefitting us in our carrier.

**Introduction:**

Our project includes building a snake game based on graphical user interface (GUI). Our snake game is solely based on a static and structured programming language i.e C. Furthermore, the library/header files which we utilized in building this snake game GUI is also solely written in C. Thus, we overcame the limitations of C language, showcasing that such old and limited language could also be a part of something bigger.

**Teamwork Distribution:**

We distributed the building of this project in the following manner:

* Ali Akhter => Back-end logic and CLI implementation.
* Umer Jamal => Back-end logic and CLI implementation.
* Muhammad Rehan Siddiqui => GUI based implementation.

**Methodology:**

Following are the tools, frameworks, libraries, and languages that were used in the development of this project:

* Programming language: C language developed by Dennis Retchie in 1977.
* IDE: Visual Studio Code and NotePad++.
* Framework: Raylib library (written in C) and its cheatsheet for understanding its functions.
* Github: Github raylib community for seeking answers to queries.
* Google Meet: Online google platform to collectively work in development of project.

We developed our snake game project in the following manner:

* Ali Akhter builds the back-end logic of snake game and implemented it on CLI.
* Umer Jamal also builds his own back-end logic of snake game and implemented it on CLI.
* Both Ali and Umer then share their logics and CLI games effectively creating a perfect model from the interference of their logics.
* Rehan Siddiqui was meanwhile learning the GUI library and its functions. The logic was conveyed to him by both, who then implemented it in GUI.

**Features:**

Following are the features of our snake game project:

* After opening the game exe, it waits for the user to click, or type enter before starting the game. In short, it shows the opening screen of the game.
* After proceeding, from the opening screen the snake starts its motion from the center in the upward direction.
* A fruit will randomly spawn in the given space.
* User can control the movement of snake through arrow keys.
* Whenever the snake eats a fruit, length of snake increases by one block, speed of snake increases by 1millisecond, and fruit is again randomly spawned in the given space.
* If the snake hits itself or the boundaries of the window, the game ends.
* During the game, in the bottom score is present which keeps updating every time snake eats fruit. Furthermore, high score is present which keeps track of the highest score overall played by the user in a text file.
* If the player has exceeded the high score, a congrats msg will appear in the end and the high score would be overwritten in the file.
* Sound effects are added at every stage, input, and situation to make the game more interactive.

**Implementation:**

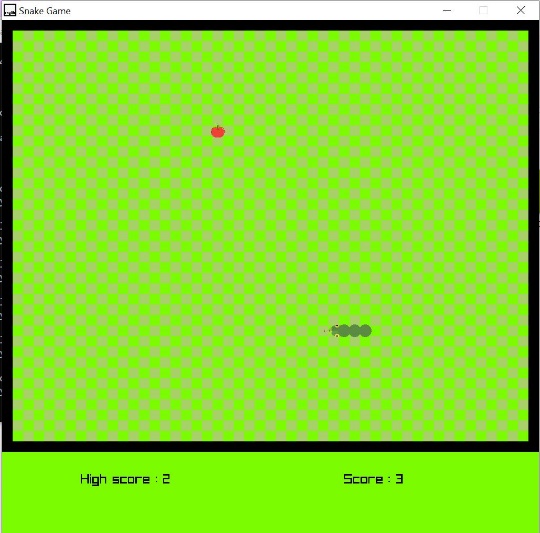
Our code logic implements as follows:

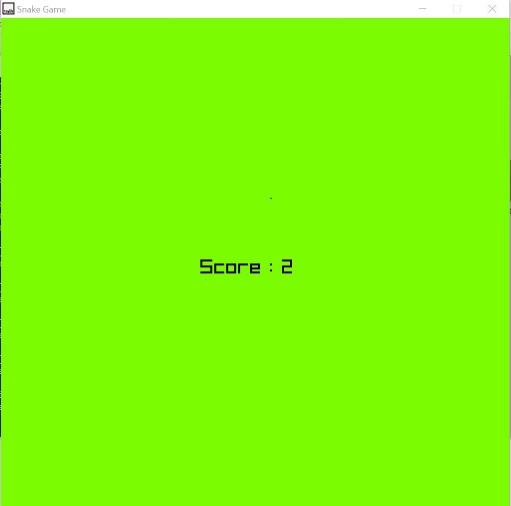
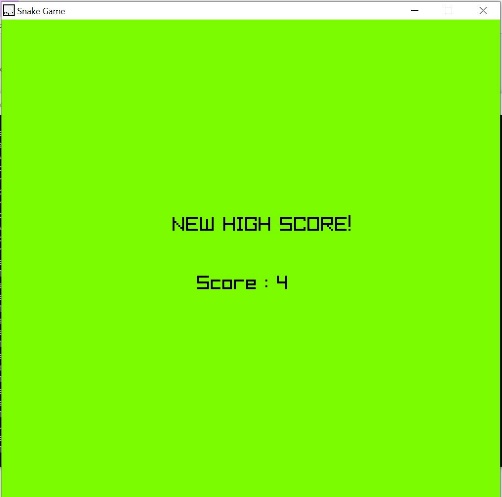
* Header Files: We included standard input output header file (<stdio.h>) for basic input/output and filing, time header file(<time.h>) for managing speed of snake, and raylib header file(raylib.h) for GUI.
* Defining a position struct which would contain x and y variables representing a 2D coordinate system. Similarly, a position struct for body was defined which would hold an array of x and y variable for representing snake body in a 2D system.
* Both structs were then declared for snake head, snake body, and body of the snake.
* Following are some of the variables that were initialized globally with their purpose written below:
* Char: inp for keeping track of the latest input, text[20] and textm[20] for storing a string containing the score and high score of the player.
* Int : len for storing the length of the snake, ex for checking if the game is lost or not, checkinp for handling the constricted motion of the snake, fx and fy for fruit coordinates, score for tracking the score of user.

Now comes the var datatypes that are defined under raylib files:

* Texture 2D: fruit=>fruit image, hu=>head image in upward direction, hd=> head image in downward direction, hl=>head image in left direction, hr=> head image in right direction.
* Color: Different colors were initialized for graphical representation of the game.
* Sound: Different sounds were defined for various purposes as per their name.
* Firstly, the head position is initialized to the center of the game with its direction being originally upward.
* Now we retrieve the high score from a text file by basic filing operation.
* Then we initialize the window, audio and textures using basic raylib functions.
* Then we call our own defined opening screen function which displays the opening image and prompt the user to continue.
* The loop for the game is run until ESC key is pressed or the window is closed manually.
* Inside the loop, begin drawing and end drawing (built-in raylib function) is called between our game designing circulates.
* Display function is used to draw all the textures necessary to graphically represent the snake game.
* Input function is used to take input and update inp variable accordingly.
* Update function is used to update position of snake.
* Logic function is used to check whether the game is over or wether the snake has eaten any fruit and updates variables accordingly.
* After the game over, if the score is greater than previous maximum score, update high score and congrats the player.
* Unload all the audio, video and file resources.

**Pictures:**

**Conclusion:**

Through the development of this project, first and foremost we learnt to work as a team, promoting a sense of companionship in coding which would help in our professional life. Secondly, we came to know the graphical or real-life implementation of our programming fundamental course, encouraging us to dive deep into the world of programming. Lastly the development of a snake game has motivated us to further enhance our development skills, paving a way in game development.