**Data structures and algorithms LAB**

**CSL-221**



Snake Game

**Group Members:**

fatima butt (02-134201-037)

M. Nawfal Burhan (02-134201-035)

Abdul Sameeh Qureshi (02-134201-104)

# Index

### Acknowledgement

### Introduction

### Background

### Problem Statement

### Proposed system

### Flow diagram of the system

### Screen shots of module with explanation

### Code snippet of algo applied

### Future work

### Conclusion

# **Acknowledgement:**

We both have taken serious efforts in this project. It would not have been possible without the support of our teachers “Miss Maham”, and “Miss Lubna”. We are grateful for the excellent quality of lectures they have provided us that made us achieve this project. We would also like to thank many YouTube channels who are sincerely passionate to educate the young minds of our generation without any cost. We would also like to express our special gratitude towards our parents for always supporting us from the beginning and until the end. Lastly, we are grateful to each other for becoming a great team and completing this project and it would not have been easy without each other’s help.

# **Introduction:**

The objective of this program is to create a game application which can hold the player's record and organize them in a sorted order. The game requires the players to assess their surroundings and find the quickest or safest route to a point. This is an excellent game to learn about the Two-Dimensional awareness and plan of your next move. When using the game application, the players will be prompted to enter their name after which game will proceed further. Upon losing, the player's score will be stored in a ".txt" file and will be sorted using a sorting algorithm. In the end a list of high scores will be printed on the screen.

# **Background:**

The Snake design dates to the arcade game Blockade, developed and published by Gremlin in 1976.[5] It was cloned as Bigfoot Bonkers the same year. In 1977, Atari released two Blockade-inspired titles: the arcade game Dominos and Atari VCS game Surround. Surround was one of the nine Atari VCS launch titles in the US. It was sold by Sears under the name Chase. That same year, a similar game was launched for the Bally Astrocade as Checkmate.

The first known home computer version, titled Worm, was programmed in 1978 by Peter Trefonas of the US on the TRS-80 and published by CLOAD magazine in the same year. This was followed shortly afterwards with versions from the same author for the Commodore PET and Apple II. A clone of the Hustle arcade game, itself a clone of Blockade, was written by Peter Trefonas in 1979 and published by CLOAD. An authorized version of Hustle was published by Milton Bradley for the TI-99/4A in 1980. In 1982's Snake for the BBC Micro, by Dave Bresnen, the snake is controlled using the left and right arrow keys relative to the direction it is heading in. The snake increases in speed as it gets longer, and there is only one life.

Nibbler (1982) is a single-player arcade game where the snake fits tightly into a maze, and the gameplay is faster than most snake designs. Another single-player version is part of the 1982 Tron arcade game, themed with light cycles. It reinvigorated the snake concept, and many subsequent games borrowed the light cycle theme.

Starting in 1991, Nibbles was included with MS-DOS for a period as a QBasic sample program. In 1992, Rattler Race was released as part of the second Microsoft Entertainment Pack. It adds enemy snakes to the familiar apple-eating gameplay.

# **Problem Statement:**

The game will have a snake, pellets to eat and a plane with side walls. People will play this game for removing frustration of mind. Children love this game. This game has no age restriction, every age of players will play it. The game is played automatically based on certain rules and ends when a player loses.

Rules:

• If snake hits itself game ends.

• If snake hits the wall game ends.

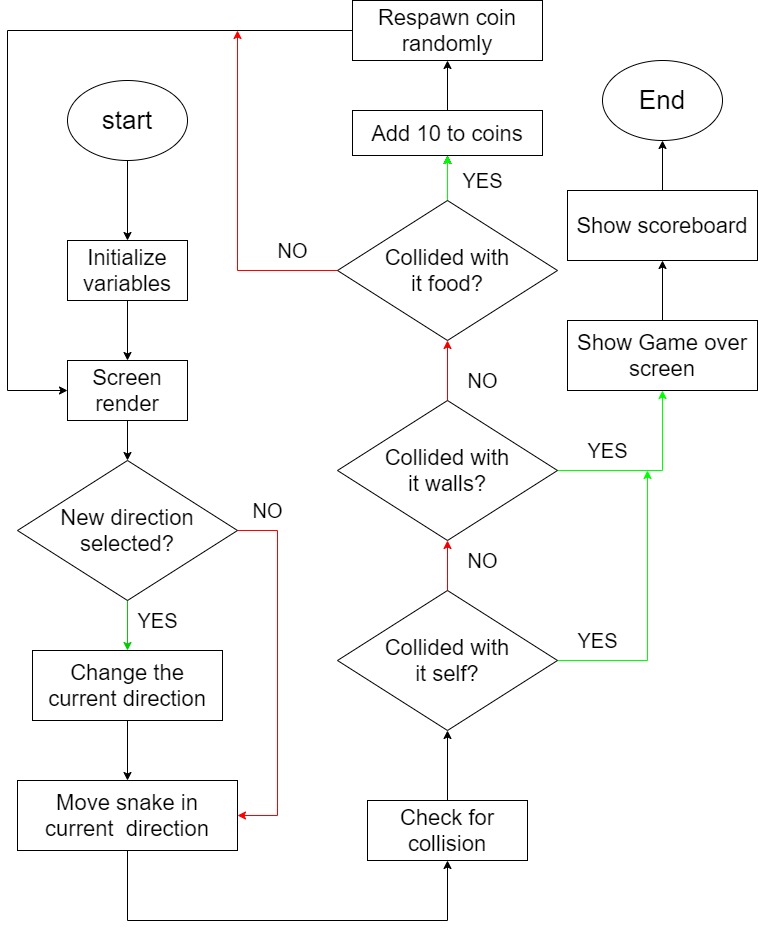
• Length of snake increase when snake hits or eat the pellet.

# **Proposed System:**

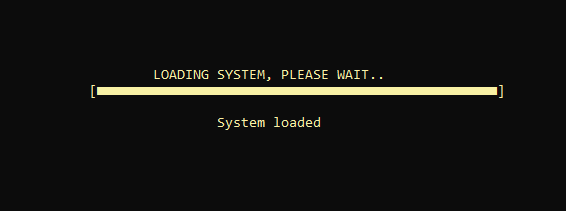
The sole purpose of the project is to provide entertainment to the users by providing a well optimized and user friendly low powered game and record their data by creating a fully automated system which will make it a lot easier to store and retrieve data of users. The system will firstly ask the users to insert their name to store their score. User will have to collect as many foods as they can to make their name reach the high score list. The players must avoid colliding to the walls upon which the game will be over. The project will also sort the high score record on the users according to the highest scores and then print the list at the end and store it in a “.txt” file.

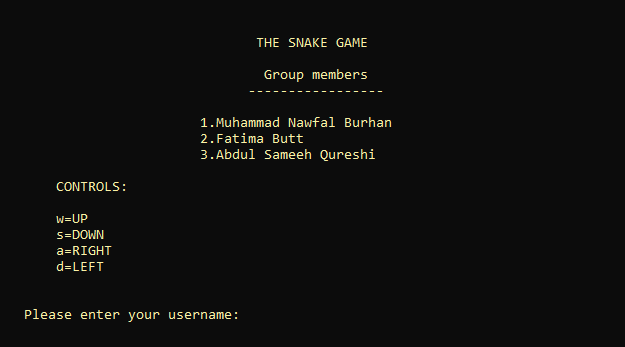
The engine which will run this code is compatible with Windows 8.1, 10 (x64, x86).

# **Flow diagram of system:**



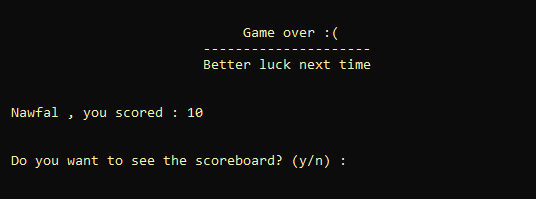
# **Screenshots of module:**

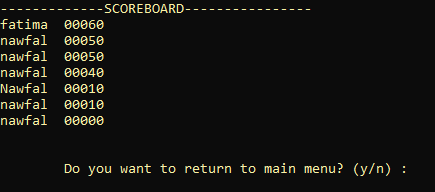












# **Code snippet of algorithms applied:**

void sorting()

{

for (int pas = 0; pas < arrNum + 1; pas++)

{

for (int j = 0; j < (arrNum - pas - 1); j++)

{

if (numArr[j] < numArr[j + 1])

{

int temp1 = numArr[j];

numArr[j] = numArr[j + 1];

numArr[j + 1] = temp1;

string temp2 = nameArr[j];

nameArr[j] = nameArr[j + 1];

nameArr[j + 1] = temp2;

}

}

}

}

void file\_handling()

{

string line;

int i = 1, len, num = 0;

string cpyScore;

system("cls");

nameArr[0] = name + "\t";

numArr[0] = score;

cout << "-------------SCOREBOARD----------------" << endl;

fstream MyFile("records.txt", ios::in); // reading from file

if (MyFile.is\_open())

{

while (MyFile)

{

arrNum++;

len = 0;

getline(MyFile, line);// to clear buffer

if (line == "")

break;

len = line.size();

len -= 5;

nameArr[i] = line.substr(0, len);

cpyScore = line.substr(len, 5);

len += 5;

numArr[i] = stoi(cpyScore); // converting score taken from file to int

i++;

}

sorting();

}

else

cout << "Error, could not read from file" << endl;

MyFile.close();

remove("records.txt"); // deletes the content of file // deleting the previous

// content so that new content can be inserted

MyFile.open("records.txt", ios::app);// writing data in the file/ ------------------ move thissss

if (MyFile.is\_open())

{

for (int i = 0; i < arrNum; i++)

{

len = numArr[i];

cpyScore = to\_string(numArr[i]);

len = to\_string(len).length();

if (len == 4)

cpyScore = "0" + cpyScore;

else if (len == 3)

cpyScore = "00" + cpyScore;

else if (len == 2)

cpyScore = "000" + cpyScore;

else if (len == 1)

cpyScore = "0000" + cpyScore;

MyFile << nameArr[i] << cpyScore << endl;

cout << nameArr[i] << cpyScore << endl;

}

}

else

cout << "Error! File could not be created";

MyFile.close();

}

# **Future work:**

Although this game does not have a bright future in this era anymore, due to the intense AAA titles made by gaming industries, there are some things which can be improved in this game. For example, sound libraries and 3-d rendered graphics can make this game much more pleasing to the eyes and ears.

# **Conclusion:**

This project gave us a thrilling, frustrating and a pleasurable experience. But most of all, it has expanded our thinking levels and we have learned yet again to overcome such an obstacle. The project enabled us to learn in many sectors like planning, designing, managing, developing, programming skills and so on.