BSCS ITC Project:

C SNAKE GAME BY:

1. **Murad Popattia (K17-3722)**
2. **Ali Ammar (K17-3713)**
3. **Osama Ali (K17-3716)**
4. **Hamza Ahmad (K17-3703)**

Supervisor:

MISS FARAH SADIA

FAST – National University of Computer & Emerging Sciences

Shah Latif Town, National Highway

Karachi

Project Description:

1. **Introduction**
   - DEV C++
   - Version: 5.11
   - Multiple Libraries have been used:

- #include<stdio.h> (to use normal c related functions)
- #include<stdlib.h> (to use system("cls") function)
- #include<windows.h> (to use the beep function for music)
- **#include<conio.h>** (to use kbhit function and getch())

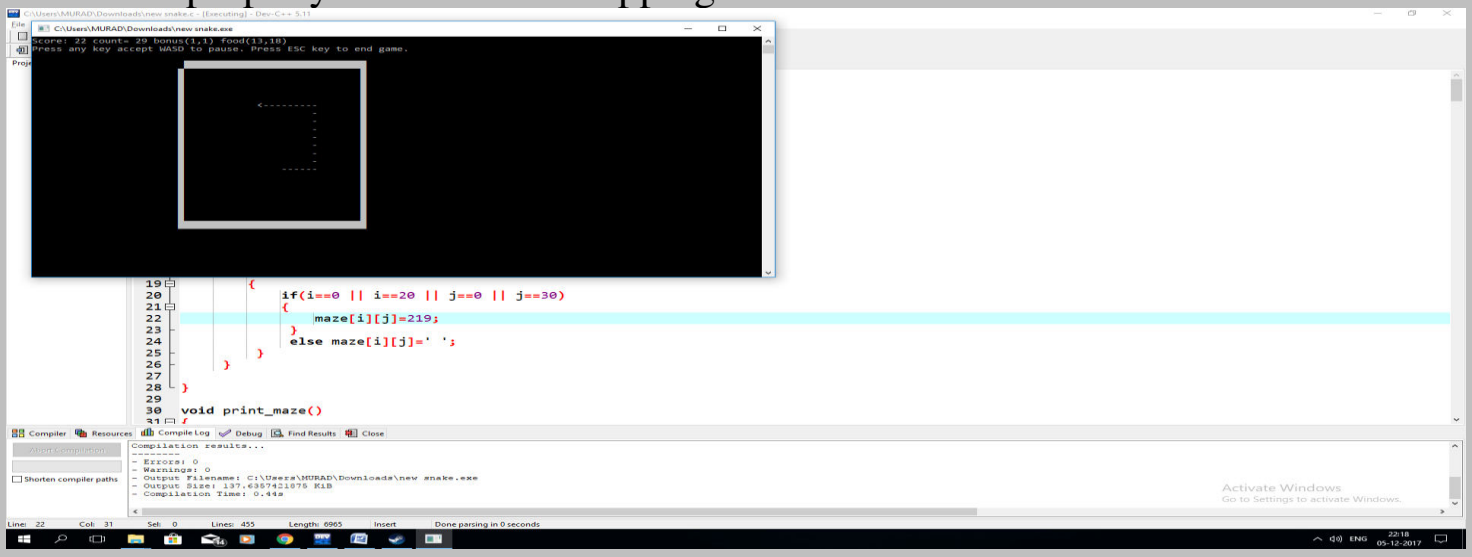
2. **Features Description:**
   - The snake game we have made is not made using **structures** instead it has been made using **2D arrays** without any sort of dynamic allocation.
   - **System("cls")** has been used **multiple times** in order to give it the game like effect due to the fact that this game is basically **assigns and prints the maze** and its assignment has been used to produce the game.
   - Maze has been rectangular fashion not in square **20*30** but it can be altered in the code. So snake **moves in y axis faster than in x axis**. Mentioned already to avoid any inconvenience.
   - The program consists of a declared maze which has been made into a function.
   - Consist of two co-ordinates **h1 and h2** which keep the track of the cords of snake's head.
   - **f1, f2** here are the coordinates of the food which is called through print_food function and is responsible for printing food at **random location** like in snake game.
   - **Two arrays to store tail's location** have been used to store the x cords and y cords of the tail to be printed.
   - The basic idea used here is to **print the tail** in location of the head and then use another function named **tail remover** to **remove the excess tails** that were printed depending upon the score. This is how tail moves.
   - The **movement functions** here basically work on a type of swapping here to move the head by printing it to the next location while system ("cls") does the work. **W, A,S,D** have been assigned using switch case for movement functions.

- **Beep function** have been used to indicate the score increment or game over for the user.
- As an added feature cords of all food and head and tail and count and bonus locations have been **displayed for function clarification.**
- The combination of **kbhit** and **sleep** has been used in order to produce a **minimal key input game**.
- There are **two files attached**. One is the game with bonus feature a that replicates the snake xenia game in which bonus was displayed for a **limited time** and gave users an increased amount of score. For the **bonus increment the length of snake increases as per the bonus score given.** But the bonus feature we have added has introduced **some problems** which I have addressed in problems faced section.

3. Problems Faced:

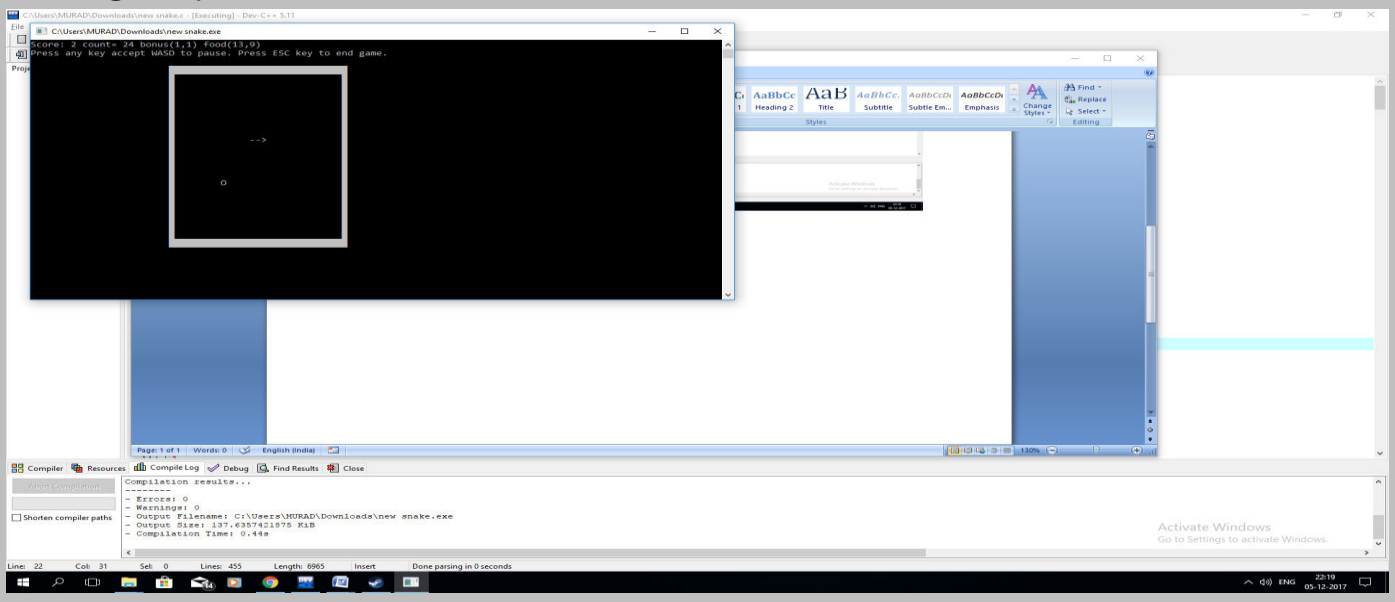
We have overall faced 2 problems only due to the added bonus feature which might be overcome in future with consultation with the respected supervisor.

1. The first problem is that the food character 'O' disappears sometimes about (8,..) or (17,..) or some xyz coordinates like that. The cords of the food are displayed meaning that it is still present on its respective location but is ' ' instead of 'O'.
   When snake placed their it eats food and then the program works properly evident that swapping has occurred.

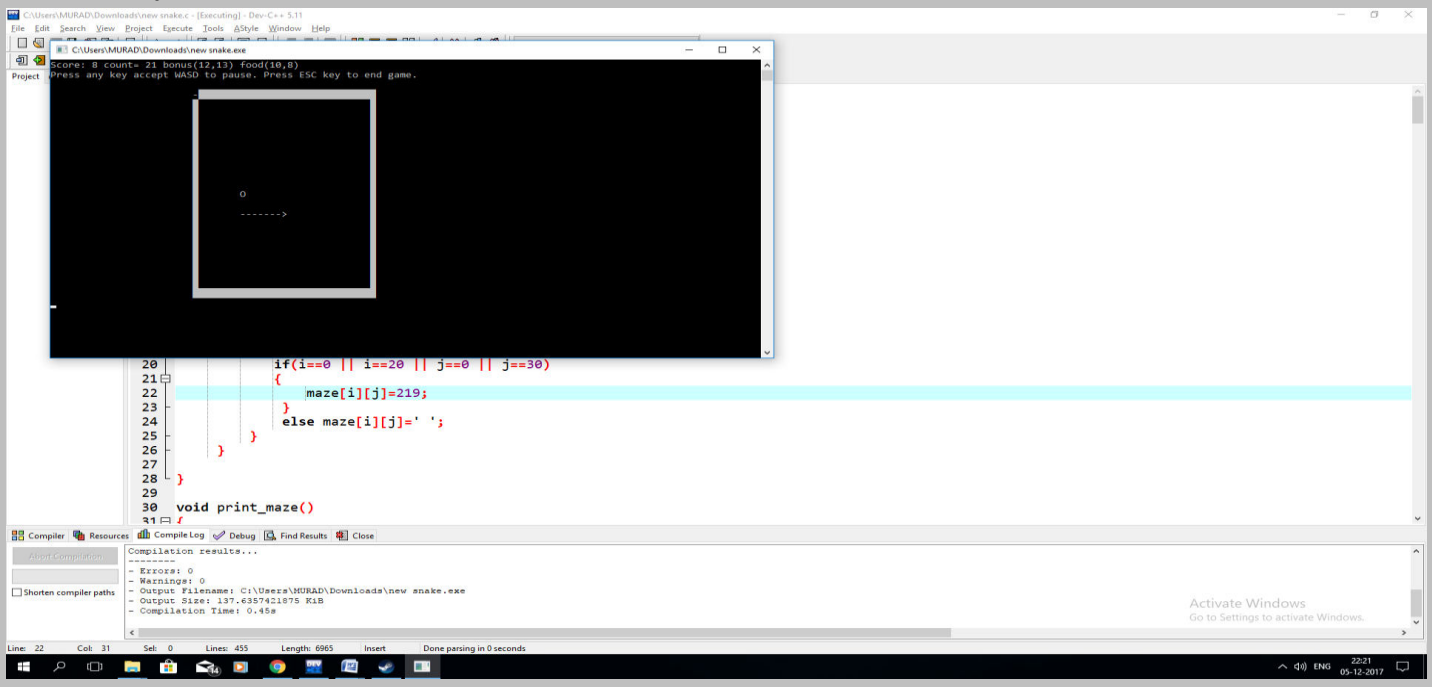


2. The second problem is once again due to this that one when the bonus is 0,0 the block at the 0,0 location disappears of the maze once the bonus is eaten. If b1 and b2 were to be zero that within the count of 30 the 0,0 block disappears so we initialized it with 1,1 but still in vain after eating the bonus.

BEFORE:



AFTER:



This is all due to the count==30 function added we suppose added by Murad

inorder to banish the bonus food after the counter reaches 30. This is a mistake yet to be overcome.

Code:

```
if(count==30)
        {
            maze[b1][b2]=' ';
            b1=1;
            b2=1;
        }
```

4. **Group member's contribution:**

Ali Ammar:

- Coded how the snake function should be moved.
- Also suggested the bonus printing feature.
- Random food location printing.

Osama Ali:

- Gave an idea on how the maze was to be assigned then printed and then game exiting options like snake hitting wall or its body will produce errors.
- Gave its rough code design.
- Suggested that intro screen and rules should be added.
- Helped us made code user interactive by asking user the option to play again.

Hamza Ahmad:

- He gave the unique idea to use the beep function in order to make the game more user interactive.
- And also that kbhit and sleep function should also be used to make it real like!

Murad Popattia:

- He suggested the tail controlling with consultation from seniors and juniors and ITC teachers.
- And the compiling the whole code into one form.
- Also told that how the bonus should be disappeared after some time.

5. Resources(If use any):

- Using of resources include net for the functions such as kbhit and sleep.
- The tail function and and its removing was consulted with our seniors and juniors and ITC teachers.
- No book was used in the creation of this game.
- Idea on display of how the game maze and other features was taken from google photos.