

MINOR PROJECT

(Snake Game)

Student Name: KOMAL GUPTA

Branch: UIC -BCA

Semester: 5

Subject Name: WEB DEVELOPMENT

UID: 22BCA10262

Section/Group:22BCA 3-A

Date of Performance:25/10/2024

Subject Code:22CAH 301

1. AIM/OVERVIEW OF THE PRACTICAL:

The objective of this project is to develop a classic **Snake Game** using **C#** in a console-based environment. The game allows the player to **navigate a snake to eat** randomly placed food items within a specified grid, which **increases both the snake's length and the player's score**. The game **ends** when **the snake collides with the boundary walls** or itself.

2. TASK TO BE DONE:

- **Set Up Game Environment:** Define the game area dimensions and initialize variables for snake position, length, score, and food position.
- **Implement Controls:** Set up input detection for arrow keys, allowing the player to change the snake's direction without reversing it.
- **Food Generation:** Randomly place food items within the game area, ensuring they don't overlap with the snake's current position.
- **Collision Detection:** Implement logic to end the game upon collision with walls or the snake's own body.
- **Console Rendering:** Continuously update the console display to reflect the snake's movement, food position, and current score.

3. ALGORITHM/FLOWCHART:

1. Initialize Game:

- Set initial snake length and position at the centre of the game area.
- Generate the first piece of food at a random position.
- Set the initial score to zero and define the game speed.

2. Game Loop:

- Check for player input to update the snake's direction.
- Update the snake's position based on the current direction.
- Food Collision: If the snake reaches the food, increase its length, increment the score, and generate a new piece of food.
- Collision Detection: Check for wall or self-collisions; if detected, end the game.
- Render Game: Display the current game state in the console, including walls, snake, food, and score.
- Speed Control: Use a delay to control game speed and make it more challenging.

3. End Game:

- When a collision is detected, display "Game Over" and show the final score, giving the player an option to exit the game.

4. DATASET:

There is no predefined dataset for this project. The program dynamically collects data from the user, such as expense names and amounts, during execution.

5. CODE FOR EXPERIMENT/PRACTICAL:

using System;

namespace SnakeGame

{

internal class Program

{

static void Main(string[] args)

{

Random rand = new Random();

Console.CursorVisible = false;

bool ShouldExit = false;

// Snake

string snake = "-<===|:D-";

int snakeX = 0;

int snakeY = 0;

int frogEat = 0;

// Frog

string frog = "^(..)^";

int frogX = 0;

int frogY = 0;

// Canvas dimensions

int height = Console.WindowHeight;

```
int width = Console.WindowWidth - snake.Length;

// Start the game

StartGame();

while (!ShouldExit)
{
    if (TerminalResized())
    {
        Console.Clear();

        Console.WriteLine("Canvas size changed!!\nGAME
END!!");

        ShouldExit = true;
    }
    else
    {
        Move();

        if (GotFrog())
        {
            frogEat++;

            Console.SetCursorPosition(width - 2, 3);

            Console.Write($"FROG EAT:{frogEat}");

            ShowLevel();
```

```
        ShowFood();
    }
}
}

//GotFrog
bool GotFrog()
{
    return snakeX == frogX && snakeY == frogY;
}

// Function to handle snake movement
void Move()
{
    int lastSnakeX = snakeX;
    int lastSnakeY = snakeY;

    // Handle user input
    switch (Console.ReadKey(true).Key)
    {
        case ConsoleKey.UpArrow:
            snakeY--;
            break;

        case ConsoleKey.DownArrow:
```

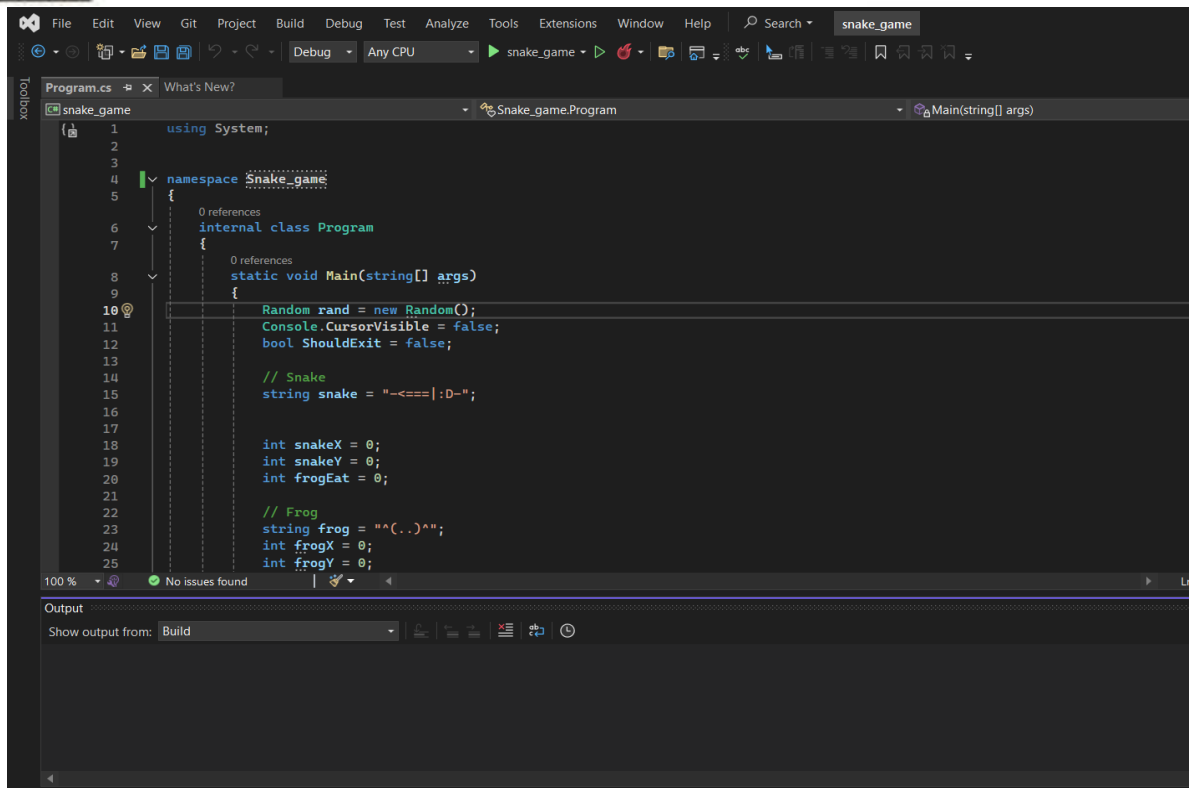
```
        snakeY++;  
  
        break;  
  
    case ConsoleKey.LeftArrow:  
  
        snakeX--;  
  
        break;  
  
    case ConsoleKey.RightArrow:  
  
        snake = "-<===|:D-";  
  
        snakeX++;  
  
        break;  
  
    default:  
  
        Console.Clear();  
  
        Console.SetCursorPosition(width / 2, height / 2);  
  
        Console.WriteLine("!! GAME END !!");  
  
        ShouldExit = true;  
  
        break;  
    }  
  
    // Clear the previous snake position  
  
    Console.SetCursorPosition(lastSnakeX, lastSnakeY);  
  
    for (int i = 0; i < snake.Length; i++)  
    {  
  
        Console.Write(" ");
```

```
}  
  
// Ensure snake stays within boundaries  
if (snakeX < 0 || snakeX > width || snakeY < 0 || snakeY > height)  
{  
    Console.Clear();  
    Console.SetCursorPosition(width / 2, height / 2);  
    Console.Write("GAME OVER!!");  
    ShouldExit = true;  
    return;  
}  
  
// Draw the snake at the new location  
Console.SetCursorPosition(snakeX, snakeY);  
Console.Write(snake);  
}  
  
// Returns true if the terminal was resized  
bool TerminalResized()  
{  
    return height != Console.WindowHeight || width !=  
Console.WindowWidth - snake.Length;  
}  
  
// Display the frog (food)  
void ShowFood()
```

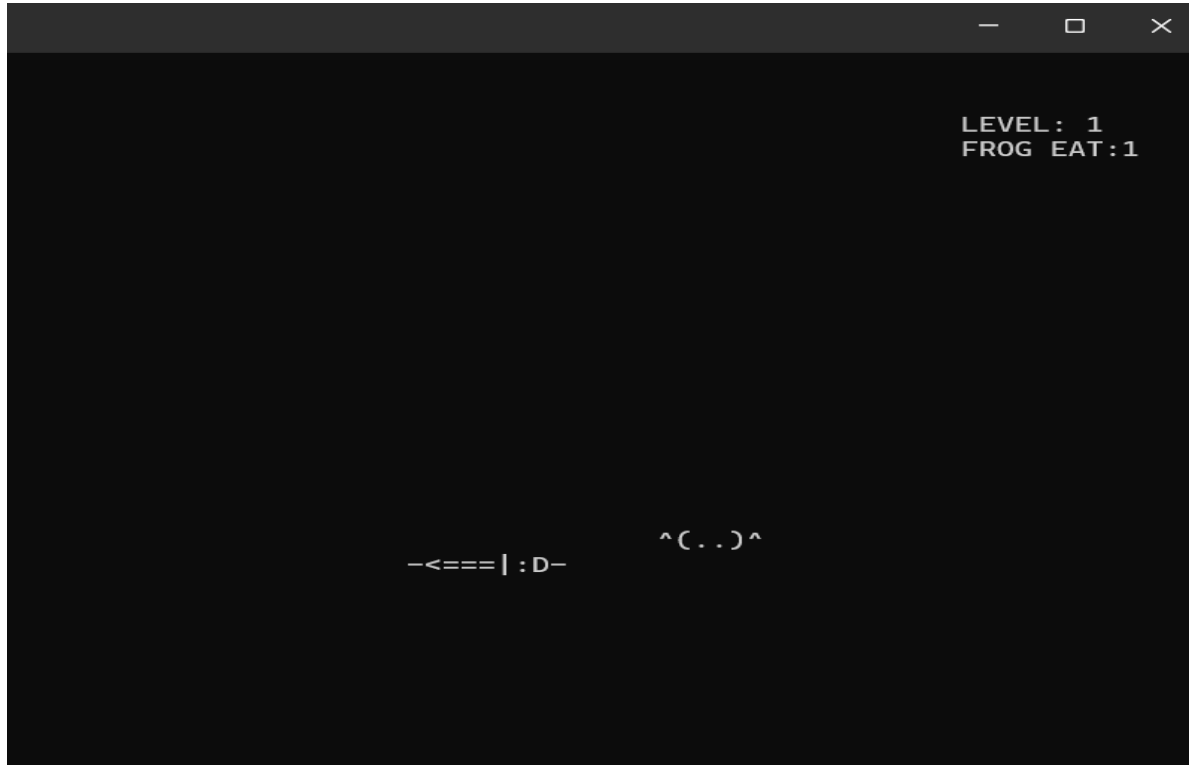
```
{  
  
    frogX = rand.Next(0, width);  
  
    frogY = rand.Next(0, height);  
  
  
    // Display the frog at a random position  
    Console.SetCursorPosition(frogX, frogY);  
    Console.Write(frog);  
}  
  
void ShowLevel()  
{  
    Console.SetCursorPosition(width - 2, 2);  
    if(frogEat >= 0 && frogEat < 5)  
    {  
        Console.Write($"LEVEL: 1");  
        Console.SetCursorPosition(width - 2, 2);  
  
    }  
    else if (frogEat >= 5 && frogEat < 10)  
    {  
        Console.Write($"LEVEL: 2");  
        Console.SetCursorPosition(width - 2, 2);  
    }  
}
```

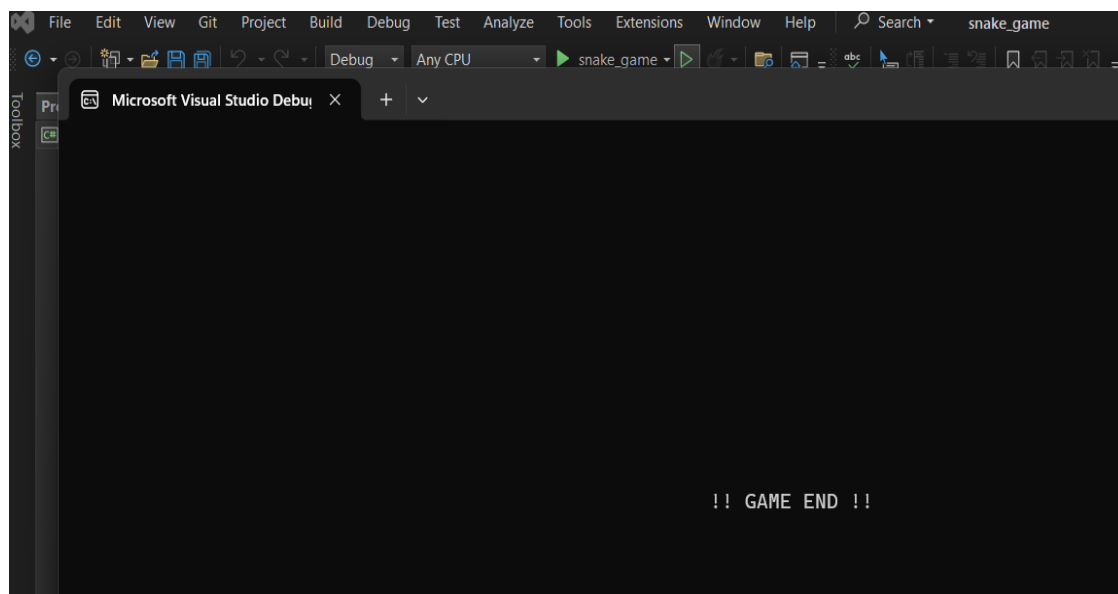
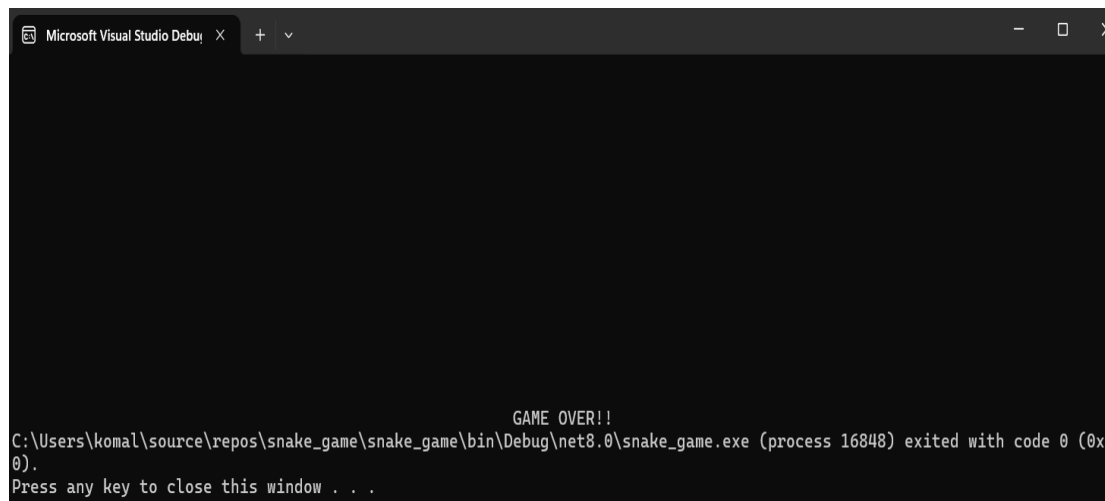
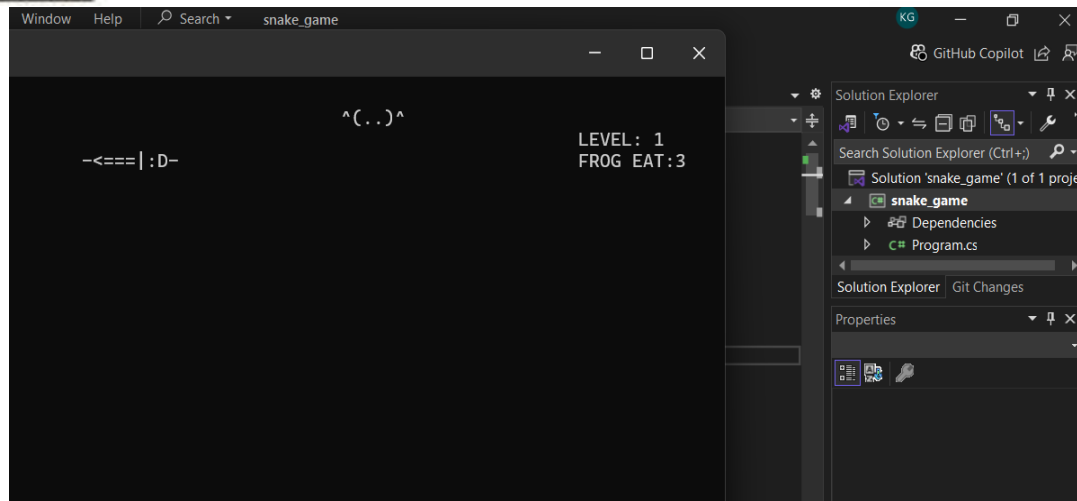


```
}  
  
else if (frogEat >= 10 && frogEat < 15)  
{  
    Console.Write($"LEVEL: 3");  
    Console.SetCursorPosition(width - 2, 2);  
}  
}  
  
// Start the game  
void StartGame()  
{  
    Console.Clear();  
    ShowFood();  
    ShowLevel();  
    Console.SetCursorPosition(0, 0);  
    Console.Write(snake);  
}  
}  
}  
}
```



```
1 using System;
2
3
4 namespace Snake_game
5 {
6     internal class Program
7     {
8         static void Main(string[] args)
9         {
10             Random rand = new Random();
11             Console.CursorVisible = false;
12             bool ShouldExit = false;
13
14             // Snake
15             string snake = "-<===|:D-";
16
17
18             int snakeX = 0;
19             int snakeY = 0;
20             int frogEat = 0;
21
22             // Frog
23             string frog = "^(. )^";
24             int frogX = 0;
25             int frogY = 0;
```





6. LEARNING OUTCOMES (WHAT I HAVE LEARNT):

1. **Advanced C# Fundamentals:** Gained hands-on experience with advanced C# concepts such as multi-threading (for game loop control), lists, and random number generation.
2. **Game Development Principles:** Developed an understanding of game loops, state management, and real-time user interaction, essential skills in game programming.
3. **Collision Detection and Boundary Handling:** Learned how to implement collision detection logic to ensure the snake doesn't pass beyond boundaries or overlap with itself.
4. **Console-based Graphics Rendering:** Mastered the art of rendering a visual representation of the game in the console, including walls, snake body, and food placement.
5. **Problem-solving and Logic Building:** Improved logical thinking and problem-solving skills through planning and structuring game mechanics.

7. CONCLUSION

This Snake Game project was a highly rewarding experience, combining elements of game logic, user interaction, and console-based graphics. This project not only improved my understanding of C# programming but also provided an introduction to game development principles. By implementing various mechanics like collision detection, input handling, and real-time score tracking, I gained valuable experience in logic building, debugging, and testing. This project has laid a strong foundation for more advanced game development and programming projects in the future.

Evaluation Grid:

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Demonstration and Performance (Pre Lab Quiz)		5
2.	Worksheet		10
3.	Post Lab Quiz		5