**Project Report**

*Project: Snake Game*

Group members:

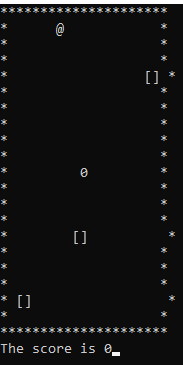
Zayan Ahmed (23k – 0891)

Muhammad Uzair Uddin Khan (23k – 0665)

Muhammad Saud Imran (23k – 0836)

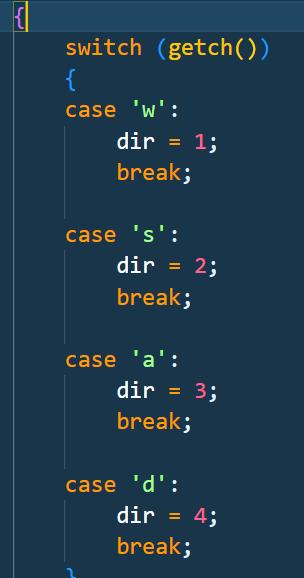
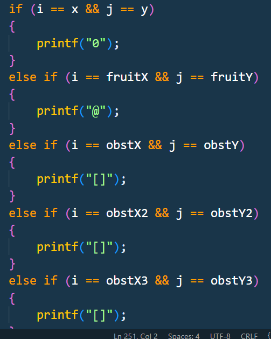
Report:

**Initialization**:

* The game starts with a snake positioned at the center of a grid.
* A fruit and three obstacles are placed randomly on the grid.
* 

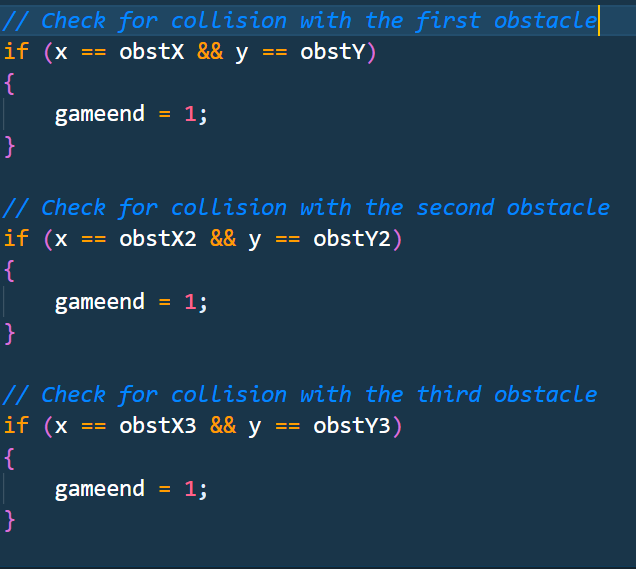
**Game Display:**

* The console displays a grid where the snake ('0'), fruit ('@'), and obstacle ('[ ]') are represented by characters.
* The current score is displayed at the bottom of the console.
* The user can control the snake's direction using the 'w', 's', 'a', and 'd' keys for up, down, left, and right, respectively.
* The snake moves continuously in the current direction.



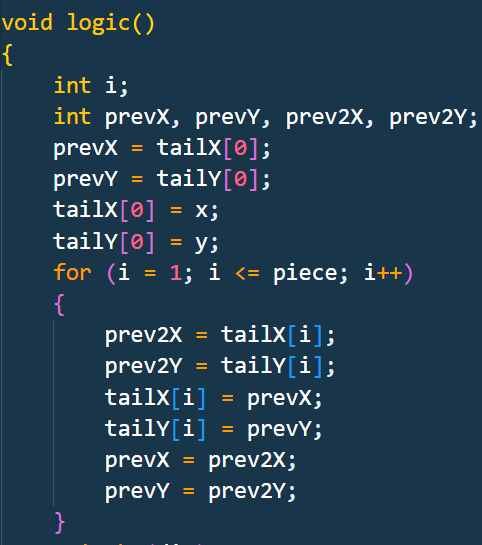
**Collision Handling:**

* The game checks for collisions with the walls. If the snake collides with the walls, it appears on the opposite side.
* If the snake collides with the obstacle, the game ends.



**Fruit Consumption and Snake Movement:**

* If the snake's head coordinates match the fruit's coordinates, the snake grows longer, and a new fruit and obstacle are placed randomly.
* The score increases with each fruit consumed.



**Modular**:

1. **generateObstacle()**

* + Generates the first obstacle's coordinates.

1. **generateSecondObstacle()**
   * Generates the second obstacle's coordinates, avoiding overlap with the first obstacle.
2. **generateThirdObstacle()**
   * Generates the third obstacle's coordinates, avoiding overlap with the first two obstacles.
3. **logic()**
   * Handles the logic of the game, including updating the snake's position, checking for collisions with obstacles and fruit, and updating the score.
4. **input()**
   * Takes user input to change the direction of the snake. It uses **kbhit()** and **getch()** functions for keyboard input.
5. **setup()**
   * Initializes the game by setting up the initial positions of the snake, fruit, and obstacles.
6. **draw()**
   * Renders the game on the console, displaying the current state of the snake, fruit, and obstacles.
7. **main()**
   * The main function where the program execution starts. It contains the game loop, calling functions to set up, draw, and update the game until the game ends.

Additionally, you have several standard C library functions such as **rand()**, **printf()**, **goto**, **Sleep()**, **system("cls")**, and others that are used for various purposes in your program.

**Uses of Libraries**:

1. **#include <stdio.h>**
   * Purpose: Provides input and output functions like **printf** and **scanf**. It's a standard input/output library.
2. **#include <stdlib.h>**
   * Purpose: Provides general-purpose functions involving memory allocation, random number generation (**rand()**), and other utility functions.
3. **#include <conio.h>**
   * Purpose: This header is not part of the standard C library and is often considered non-portable. It provides functions like **kbhit()** and **getch()** for handling keyboard input. Note that its usage might not be supported on all compilers or platforms.
4. **#include <windows.h>**
   * Purpose: This header is specific to Windows systems. It provides functions like **Sleep()** for introducing delays in the program and interacting with the Windows API. Note that it makes your code less portable to other operating systems.