**COURSE BASED PROJECT REPORT ON PROGRAM ON SNAKE GAME**

**I SEMESTER**

**B.E CSE (INTERNET OF THINGS WITH CYBER SECURITY INCLUDING BLOCKCHAIN TECHNOLOGY)**

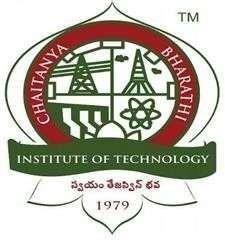
BY

SYED FAIZ AHMED BASHA 160124749056 MUBASHIR SHAHWEZ 160124749052

PADE RACHIT KISHORE 160124749054 NIKHIL JUTTUKONDA 160124749053

UNDER THE GUIDENCE OF MRS. CH. SRILAKSHMI, ASST. PROFESSOR,

DEPT. OF COMPUTER ENGINEERING & TECHNOLOGY



SUBMITTED TO

DEPARTMENT OF COMPUTER ENGINEERING & TECHNOLOGY CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY (A)

(AFFILIATED TO OSMANIA UNIVERSITY) GANDIPET, HYDERABAD – 500075 2024-25



**CERTIFICATE**

This is to certify that the report entitled ‘**Program on Snake Game’** submitted to Chaitanya Bharathi Institute of Technology in I Sem of

B.E. in CSE (IoT with Cybersecurity including Blockchain Technology) during the Academic Year 2024-25, is a record of original work done by MUBASHIR SHAHWEZ (160124749052), SYED FAIZ AHMED BASHA (160124749056), PADE RACHIT KISHORE (160124749054) and NIKHIL JUTTUKONDA (160124749053), during the

period of study in the Department of Computer Engineering & Technology, CBIT, Hyderabad.

Course Coordinator Mrs. Ch. Srilakshmi,

Asst Professor, Dept. of CET.

**DECLARATION**

I declare that the project entitled **“Program on Snake Game”** is being submitted by me in the Department of Computer Engineering Technology, Chaitanya Bharathi Institute of Technology (A), Osmania University.

This is a record of Bonafide work carried out by me under the guidance and supervision of **Mrs. Ch. Srilakshmi ma’am, Assistant Professor, Dept. of CET,**

**C.B.I.T.** The content of this project is based on the knowledge and practical work I gained during the study of Database Management Systems course.

MUBASHIR SHAHWEZ

160124749052

SYED FAIZ AHMED BASHA

160124749056

NIKHIL JUTTUKONDA 160124749053

PADE RACHIT KISHORE

160124749054



**INDEX CONTENTS**

1.Introduction 2.Objective

1. Source Code
2. Output
3. Conclusion
4. Refernces

# INTRODUCTION

## Problem Definition

Design and implement a Snake game in C programming language. The Snake game is a classic arcade-style game where you control a snake moving around the screen, eating food to grow longer. The goal is to avoid hitting the walls or your own tail while trying to achieve the highest score possible.

# Objective

## The main purpose of this problem is to create a user friendly and convenient game to play snake game.

* **Goal:** Eat food, grow longer, get points.
* **Control:** Use arrow keys to change direction.
* **Eat:** Guide snake's head to the food.
* **Avoid:** Don't hit walls or your own tail.
* **Score:** Points increase with each food eaten.
* **End:** Game over if you crash.

# SOURCE CODE

#include <stdio.h> #include <stdlib.h> #include <conio.h> #include <time.h>

// Game settings #define WIDTH 20

#define HEIGHT 10

// Snake direction

int direction = 0; // 0: Right, 1: Up, 2: Left, 3: Down

// Snake coordinates

int snakeX[100], snakeY[100]; int snakeLength = 3;

// Fruit coordinates int fruitX, fruitY;

// Score

int score = 0;

// Function to generate a random number within a range int randRange(int min, int max) {

return rand() % (max - min + 1) + min;

}

// Function to place the fruit randomly void placeFruit() {

fruitX = randRange(1, WIDTH - 2); fruitY = randRange(1, HEIGHT - 2);

}

// Function to initialize the game void initializeGame() {

// Initialize snake position snakeX[0] = WIDTH / 2; snakeY[0] = HEIGHT / 2; snakeX[1] = WIDTH / 2 - 1; snakeY[1] = HEIGHT / 2; snakeX[2] = WIDTH / 2 - 2; snakeY[2] = HEIGHT / 2;

// Place the initial fruit placeFruit();

// Initialize score score = 0;

}

// Function to update the game state void updateGame() {

// Move the snake

for (int i = snakeLength - 1; i > 0; i--) { snakeX[i] = snakeX[i - 1]; snakeY[i] = snakeY[i - 1];

}

// Move the head based on the direction switch (direction) {

case 0: snakeX[0]++; break; // Right case 1: snakeY[0]--; break; // Up case 2: snakeX[0]--; break; // Left case 3: snakeY[0]++; break; // Down

}

// Check for collisions with walls

if (snakeX[0] < 0 || snakeX[0] >= WIDTH || snakeY[0] < 0 || snakeY[0] >= HEIGHT) { printf("Game Over!\n");

exit(0);

}

// Check for collisions with itself for (int i = 1; i < snakeLength; i++) {

if (snakeX[0] == snakeX[i] CC snakeY[0] == snakeY[i]) { printf("Game Over!\n");

exit(0);

}

}

// Check if the snake ate the fruit

if (snakeX[0] == fruitX CC snakeY[0] == fruitY) { score++;

snakeLength++; placeFruit();

}

}

// Function to draw the game void drawGame() {

system("cls"); // Clear the console

// Draw the top border

for (int i = 0; i < WIDTH + 2; i++) { printf("#");

}

printf("\n");

// Draw the game area

for (int y = 0; y < HEIGHT; y++) {

printf("#"); // Left border

for (int x = 0; x < WIDTH; x++) {

if (x == snakeX[0] CC y == snakeY[0]) { printf("O"); // Snake head

} else {

int isSnakePart = 0;

for (int i = 1; i < snakeLength; i++) {

if (x == snakeX[i] CC y == snakeY[i]) { printf("o"); // Snake body isSnakePart = 1;

break;

}

}

if (!isSnakePart) {

if (x == fruitX CC y == fruitY) { printf("\*"); // Fruit

} else {

printf(" "); // Empty space

}

}

}

}

printf("#\n"); // Right border

}

// Draw the bottom border

for (int i = 0; i < WIDTH + 2; i++) { printf("#");

}

printf("\n");

// Print the score printf("Score: %d\n", score);

}

int main() {

// Initialize random number generator srand(time(NULL));

// Initialize the game initializeGame();

// Game loop while (1) {

// Draw the game drawGame();

// Handle user input if (\_kbhit()) {

switch (\_getch()) {

case 'w': if (direction != 3) direction = 1; break; // Up (avoid going opposite) case 's': if (direction != 1) direction = 3; break; // Down

case 'a': if (direction != 0) direction = 2; break; // Left case 'd': if (direction != 2) direction = 0; break; // Right case 'x': exit(0); // Exit the game

}

}

// Update the game state updateGame();

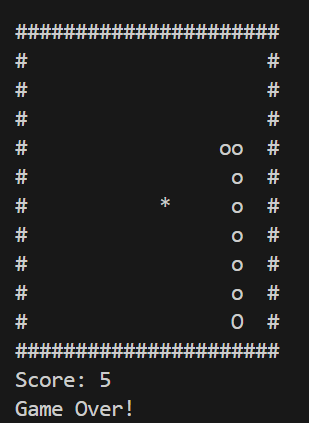
// Add a small delay to control game speed usleep(100000); // 100 milliseconds

}

return 0;

}

# OUTPUT

****

* 1. **CONCLUSION**

This project successfully implemented a classic Snake game using the C programming language. The game includes core features such as snake movement, fruit consumption, collision detection, and a scoring system. While developing the game, challenges such as optimizing game speed and ensuring smooth controls were addressed. Future enhancements could include adding obstacles, power-ups, and levels to increase complexity and player engagement. Overall, this project provided valuable experience in game development and programming, and the resulting Snake game offers an enjoyable and nostalgic gaming experience.

* 1. REFERENCES

1. Snake Game in C without using Graphics - Sanfoundry
2. Snake Game in C - GeeksforGeeks
3. Snake Game with C - Stack Overflow