

A MINI PROJECT
on
SNAKE GAME

(Submitted for practical fulfillment of the requirements for the award of the degree)
of

BE 2/4(CSE) MINI PROJECT

by

ABRAR ATHAR HASHMI

(160115733082)

ABDUL WAHED

(160115733079)

under the guidance of

KALPANA

Department of CSE

CBIT, Hyderabad



Department of computer science and Engineering

Chaitanya Bharathi Institute of Technology

Gandipet, Hyderabad-500075

CERTIFICATE

This is to certify that the project work entitled “**SNAKE GAME**” is bonafide work carried out by **ABRAR ATHAR HASHMI (160115733082)** and **ABDUL WAHED (160115733079)** in fulfillment of the award of the degree of **BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING** by the **OSMANIA UNIVERSITY**, Hyderabad, under our guidance and supervision.

The results enclosed in this report are not submitted to any other university or organization for the award of any degree or diploma.

Internal guide
Kalpana

Associate professor

Department of CSE

CBIT, Hyderabad

Head of the department
Dr.Y.Rama Devi

Professor and Head

Department of CSE

CBIT, Hyderabad

DECLARATION

This is to certify that the work reported in the present project entitled "**SNAKE GAME**" is a record of work done by us in the Department of Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, Osmania University. The reports are based on the project work done entirely by us and not copied from any source.

ABRAR
(160115733082)

ABDUL WAHED
(160115733079)

ACKNOWLEDGEMENTS

I would like to express my deep felt appreciation and gratitude to **Ms Kalpana**, Department of CSE, my project guide, for his skillful guidance , constant supervision, timely suggestion, keen interest and encouragement in completing the individual seminar within the stipulated time.

I am honored to express my profound sense of gratitude to **Dr.Y.Rama Devi**, Head of Department, CSE, who has served as a host of valuable corrections and for providing me time and amenities to complete this project.

I gratefully express my thanks to **Dr.Chenna Keshava Rao**, Principal of our college and the management of "**CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY**" for providing excellent academic and learning environment in the college.

I wish to express my heartfelt gratitude to the **Members of staff** and all others who helped me in bringing up my project. I would also like to thanks the Lab **assistants and programmers** for helping me through my project.

ABSTRACT

The player controls the direction of movement of the snake. Fruits are present at random points inside the barrier. The goal is to eat the fruit. When the snake hits the fruit, size of its tail increases. The score is updated by +10 each time the snake eats the fruit.

The user can play two levels.

In Level 1, the snake can touch the boundary and comes from the opposite side of the boundary. When you have attained a particular score, you will enter the next level.

In Level 2, the game is over if the snake touches the barrier.

TABLE OF CONTENTS

Name of the topic	page no:	
Certificate	2	
Declaration	3	
Acknowledgement	iii	4
Abstract	5	
1.INTRODUCTION		
1.1 Introduction to Snake Game	1	
2.LITERATURE SURVEY	2	
C++		
3.SYSTEM REQUIREMENTS	4	
Functional Requirements		
4.DESIGN OF SNAKE GAME	5	
Module description		
Flow chart		
5.IMPLEMENTATION	8	
6.RESULTS & DISCUSSIONS	25	
7.CONCLUSION & FUTURE ENHANCEMENT	31	
* REFERENCES		

INTRODUCTION

1.1

The concept of snake game originated in the 1976 arcade game Blockade, and the ease of implementing Snake has led to thousands of versions for many platforms like Android, iOS, Symbian etc.

This project is my work in an attempt to study how games are created on different platforms and consoles.

The game is user friendly and completely menu driven so that the users shall have no problem in using all options. It is efficient and fast in response. It is a game which can be played by anyone

NOTE: Make sure you do not have the caps lock on, otherwise the keys will fail to respond.

2

Chapter 2

LITERATURE SURVEY

C++

C++ is a middle-level programming language developed by Bjarne Stroustrup starting in 1979 at Bell Labs. C++ runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX. C++ is object oriented programming language.

Object Oriented programming is a programming style that is associated with the concept of Class, Objects and various other concepts revolving around these two, like Inheritance, Polymorphism, Abstraction, Encapsulation etc.

Class: It is similar to structures in C language. Class can also be defined as user defined data type but it also contains functions in it. So, class is basically a blueprint for object. It declares & defines what data variables the object will have and what operations can be performed on the class's object.

Objects: Objects are the basic unit of OOP. They are instances of class, which have data members and use various member functions to perform tasks.

Encapsulation and Data abstraction: Wrapping up (combing) of data and functions into a single unit is known as encapsulation. The data is not accessible to the outside world and only those functions which are wrapping in the class can access it. This insulation of the data from direct access by the program is called data hiding or information hiding.

3

Data abstraction refers to, providing only needed information to the outside world and hiding implementation details. In C++, classes provide methods to the outside world to access & use the data variables, but the variables are hidden from direct access. This can be done by using access specifiers.

Inheritance: Inheritance is a way to reuse once written code again and again. The class which is inherited is called base class & the class which inherits is called derived class. So when, a derived class inherits a base class, the derived class can use all the functions which are defined in base class, hence making code reusable.

Polymorphism: It is a feature, which lets us create functions with same name but different arguments, which will perform differently. That is function with same name, functioning in different way. Or, it also allows us to redefine a function to provide its new definition. You will learn how to do this in details soon in coming lessons.

Overloading: The concept of overloading is also a branch of polymorphism. When the existing operator or function is made to operate on new data type, it is said to be overloaded.

4

Chapter 3

SYSTEM REQUIREMENTS

Functional Requirements

3.1.1 Hardware Requirements

- * processor : Intel Core i3
- * RAM : 512 MB or more
- * Hard disk : 10GB

3.1.2 Software Requirements

- * Language : C++
- * Operating system : Any OS

5

Chapter 4

DESIGN OF SEARCH ENGINE SYSTEM

4.1 MODULE DESCRIPTION

The project has four modules.

(A)Module One:

Setup Function

A function which is called first to set up the variables which includes

- Default position of the snake
- To choose random position of the fruit

(B)Module Two:

Draw Function

A function which sets up the layout of the game every time the function is called.

(C)Module Three:

Input Function

A function that takes input of left, right, up and down from the keyboard to move the snake.

6

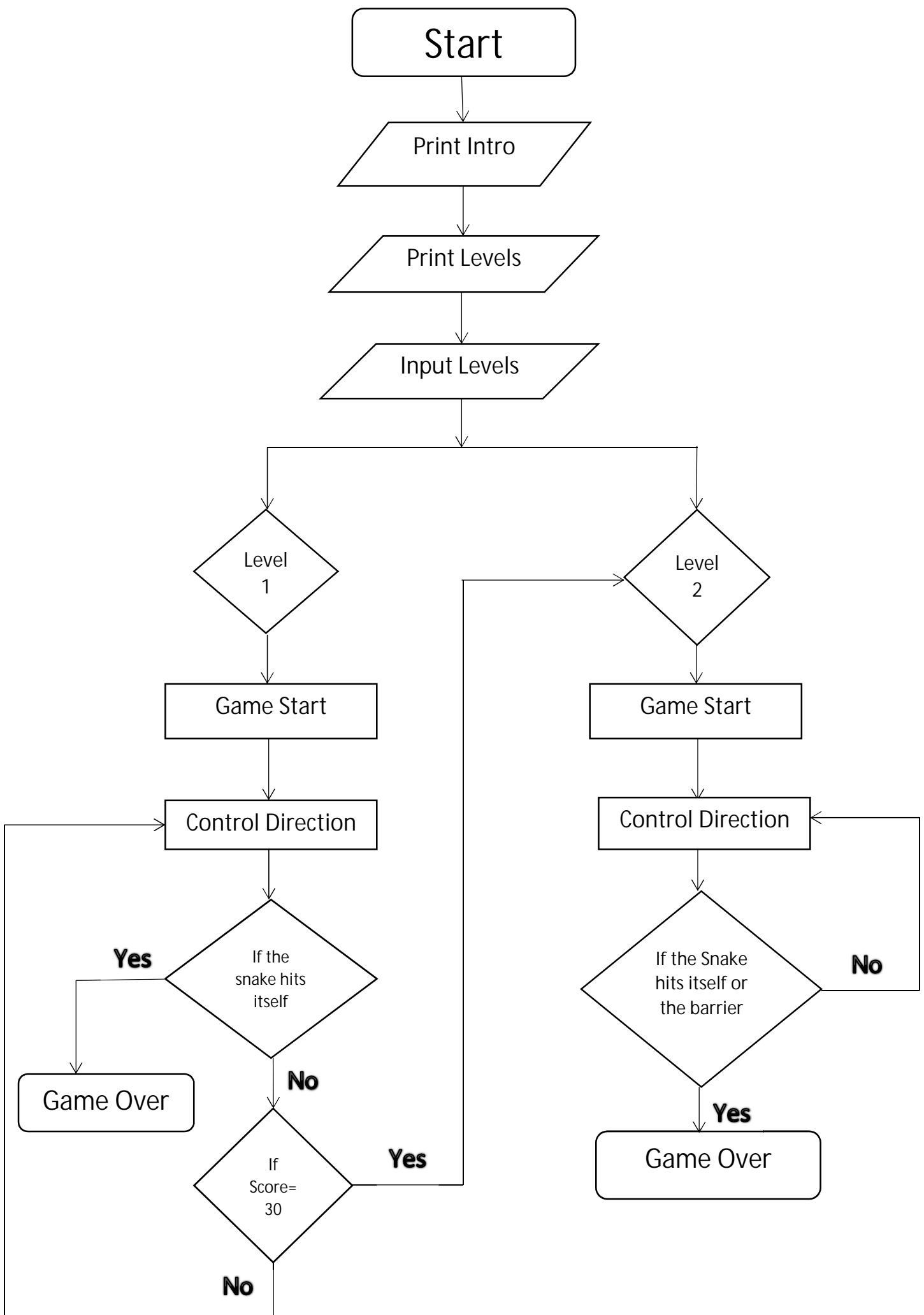
(D)Module Four:

Logic Function

- Implements logic of the array like moving the snake when user inputs key. Changes fruit's coordinates when snake eats the fruit.
- As it moves forward, it leaves a trail behind, resembling a moving snake.

FLOWCHART

Here we use Flowchart to supplement the working of the new system. The system thus made should be reliable, durable and above all should have least possible maintenance costs. It should overcome all the drawbacks of the old existing system and most important of all meet the user requirements.



8

Chapter 5 IMPLEMENTATION

```
#include <iostream>

#include <conio.h>

#include <windows.h>

#include <dos.h>

using namespace std;

bool gameOver;

const int width = 20;

const int height = 20;

int x, y, fruitX, fruitY, score;

int tailX[50], tailY[50];

int nTail;

enum eDirecton { STOP = 0, LEFT, RIGHT, UP, DOWN};

eDirecton dir;


int ch;// for levels


void Setup()

{

    gameOver = false;
```

```

dir = STOP;

x = width/2;

y = height/2;

fruitX = rand() % width;

fruitY = rand() % height;

score = 0;

nTail=0;
}

void Draw()
{
    system("cls"); //system("clear");

    cout<<"-----LEVEL "<<ch<<"-----"
\n\n\n\n";

    cout<<"\t\t\t\t";//change****

    for (int i = 0; i < width+2; i++)

        cout << "#";

    cout << endl;

    for (int i = 0; i < height; i++)
    {
        cout<<"\t\t\t\t";//change****

        for (int j = 0; j < width; j++)
        {
            if (j == 0)

```



```

        cout << "#";
    if (i == y && j == x)
        cout << "O";
    else if (i == fruitY && j == fruitX)
        cout << "F";
    else
    {
        bool print = false;
        for (int k = 0; k < nTail; k++)
        {
            if (tailX[k] == j && tailY[k] == i)
            {
                cout << "O";
                print = true;
            }
        }
        if (!print)
            cout << " ";
    }

    if (j == width - 1)
        cout << "#";
}

cout << endl;

```

```
}
```

```
cout<<"\t\t\t\t"; //change****
```

```
for (int i = 0; i < width+2; i++)
```

```
    cout << "#";
```

```
cout << endl;
```

```
cout<<"\t\t\t\t"; //change****
```

```
cout << "Score:" << score << endl;
```

```
if(ch==1 && score==30)
```

```
    gameOver=true;
```

```
}
```

```
void Input()
```

```
{
```

```
    if (_kbhit())
```

```
    {
```

```
        switch (_getch())
```

```
        {
```

```
            case 'a':
```

```
                dir = LEFT;
```

```
                break;
```

```
    case 'd':  
        dir = RIGHT;  
        break;  
    case 'w':  
        dir = UP;  
        break;  
    case 's':  
        dir = DOWN;  
        break;  
    case 'x':  
        gameOver = true;  
        break;  
    }  
}  
}  
  
void Logic()  
{  
    int prevX = tailX[0];  
    int prevY = tailY[0];  
    int prev2X, prev2Y;  
    tailX[0] = x;  
    tailY[0] = y;  
    for (int i = 1; i < nTail; i++)  
    {
```

```
    prev2X = tailX[i];
    prev2Y = tailY[i];
    tailX[i] = prevX;
    tailY[i] = prevY;
    prevX = prev2X;
    prevY = prev2Y;
}
switch (dir)
{
case LEFT:
    x--;
    break;
case RIGHT:
    x++;
    break;
case UP:
    y--;
    break;
case DOWN:
    y++;
    break;
default:
    break;
}
```

```
if(ch==1) //in level 1 snake touches boundary comes from other side
```

```
{ if (x >= width) x = 0; else if (x < 0) x = width - 1;  
  if (y >= height) y = 0; else if (y < 0) y = height - 1;  
}
```

```
else //in level 2 if snake touches boundary GAMEOVER
```

```
{ if (x > width || x < 0 || y > height || y < 0)  
  gameOver =true;  
}
```

```
for (int i = 0; i < nTail; i++) //if tail touches head
```

```
if (tailX[i] == x && tailY[i] == y)
```

```
gameOver =true;
```

```
if (x == fruitX && y == fruitY) //eats fruit length increases
```

```
{  
  score += 10;  
  fruitX = rand() % width;  
  fruitY = rand() % height;  
  nTail++;  
}
```

```
}
```

```
int main()
```

```

{

    for(int i=0;i<59;i++)
    {cout<<"--";}

    cout<<"\t\t\t\t\tSNAKE GAME"<<endl;


    for(int i=0;i<59;i++)
    {cout<<"--";}

    cout<<endl<<endl;


    cout<<"\n\n\n";
    for(int i=0;i<16;i++)
    {cout<<" ";}

    for(int i=0;i<20;i++)
    {cout<<"--";}

    cout<<"\n\t\t\t\t\tBY ABDUL WAHED AND ABRAR ATHAR HASHMI\n";
    for(int i=0;i<16;i++)
    {cout<<" ";}

    for(int i=0;i<20;i++)
    {cout<<"--";}


    system("color 0a");

    cout<<"\n\n\n\t\t\t\t\tPlease wait while loading\n\n";

    char a=177, b=219;

```



```
for(int i=0;i<16;i++)  
  
    {cout<<" ";}  
  
for(int i=0;i<20;i++)  
  
    {cout<<"--";}  
  
  
cout<<"\n\t\t\t\t\t LEVEL 1"<<endl<<endl;  
  
cout<<"\n\t\t\t\t\t LEVEL 2"<<endl;  
  
  
cout<<"\n\t\t\t\t\t ENTER YOUR CHOICE:";  
  
cin>>ch;  
  
  
if(ch==1)  
  
cout<<"*****INSTRUCTION : snake CAN touch the barrier ";  
  
else  
  
    cout<<"*****INSTRUCTION : snake CANNOT touch the barrier ";  
  
  
  
system("cls");  
  
cout<<"\n\n\t\t\t\t\tGAME LOADING";  
  
Sleep(1000);  
  
cout<<".";  
  
Sleep(1000);  
  
cout<<".";
```



```

switch(ch)
{
    case 1:
        Setup();
        while (!gameOver)
        {
            Draw();
            Input();
            Logic();
            Sleep(10); //sleep(10);
        }

        int t;

        if(score!=30)
        {
            system("cls");

            cout<<"\n\n\t\t\t\t*****GAME OVER*****";

            cout <<"\tScore:" <<score<< endl;

            cin>>t;

            break;
        }
        else

```

```
{  
    cout<<"\n\t\t\t***congratulations LEVEL 1 completed";  
    Sleep(1000);  
    cout<<".";   
    Sleep(1000);  
    cout<<".";   
  
    system("cls");  
    cout<<"\n\t\t\tENTERING LEVEL 2";  
    Sleep(1000);  
    cout<<".";   
    Sleep(1000);  
    cout<<".";   
  
    system("cls");  
  
    ch=2;  
}
```

case 2:

```
    ch=2;  
gameOver =false;
```

```
Setup();

while (!gameOver)

{

    Draw();

    Input();

    Logic();

    Sleep(10); //sleep(10);

}


Sleep(1000);

cout<<".";


system("cls");


cout<<" \n\n\n\n\n\t\t\t*****GAME OVER*****";

cout << "\n\n\t\t\tScore:" << score << endl;


break;


}

Sleep(1000);

cout<<".";
```

```
system("cls");

cout<<"\n\n\n\t\t\tTHANK YOU FOR PLAYING";

Sleep(1000);

cout<<".";

Sleep(1000);

cout<<".";

return 0;

}

//End of the Program
```

Chapter 6

RESULTS & DISCUSSIONS



```

LEVELS
-----

SELECT DIFFICULTY
-----
LEVEL 1
LEVEL 2
ENTER YOUR CHOICE:

```

SELECT DIFFICULTY

LEVEL 1

LEVEL 2

ENTER YOUR CHOICE:

[illegible][illegible]

Chapter 7

CONCLUSION & FURTHER ENHANCEMENT

The package was designed in such a way that future modifications can be done easily. The following conclusions can be deduced from the development of the project.

- Developing the game helped us to understand the concepts of looping functions to draw the layout of the game.
- We learned several project management techniques used by professionals to develop large scale project. The experience of working in team and integration of modules developed independently, with just requirement specifications, is a very important achievement for game developers.
- It helps us in many sectors like planning, designing, developing, managing, programming skills and so on.
- It can be further enhanced by adding multiplayer modes and storing score of different users.

REFERENCES

1. Basics of C by Cengage Publications

The complete reference C++ by BalaguruSwamy.