Visvesvaraya Technological University, Belagavi – 590010

CG MINI PROJECT REPORT

ON

Hungry Snake Game

Submitted by

Alstan Mascarenhas	4SO19CS015
Abhishek V Shetty	4SO19CS005

Under the guidance of

Ms Anusha S
(Assistant Professor, CSE Department)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ST JOSEPH ENGINEERING COLLEGE
Vamanjoor, Mangaluru -575028, Karnataka

2021-2022

Visvesvaraya Technological University, Belagavi – 590010

CG MINI PROJECT REPORT

ON

Hungry Snake Game

Submitted by

Alstan Mascarenhas	4SO19CS015
Abhishek V Shetty	4SO19CS005

Under the guidance of

Ms Anusha S
(Assistant Professor, CSE Department)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ST JOSEPH ENGINEERING COLLEGE
Vamanjoor, Mangaluru -575028, Karnataka

2021-2022

ST JOSEPH ENGINEERING COLLEGE
Vamanjoor, Mangaluru- 575 028

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

*This is to certify that the Mini project entitled **"Hungry Snake Game"** is a bonafide work carried out by*

Alstan Mascarenhas	**4SO19CS015**
Abhishek V Shetty	**4SO19CS005**

Students of sixth semester B.E. Computer Science & Engineering, and submitted as a part of the course Computer Graphics Laboratory with miniproject (18CSL67), during the academic year 2021-2022.

----------	----------
Ms Anusha S	**Dr Sridevi Saralaya**
Project Guide	**Head of the Department**

Name of the Examiners	**Signature with Date**
1. ----------	1. ----------
2. ----------	2. ----------

ABSTRACT

This project has used basic programming elements of OpenGL. HUNGRY SNAKE GAME is an OpenGL Computer Graphics Mini Project using OpenGL functions. The aim of the game is to collect the dots (food). As you collect food, the snake gets longer, so increasing your likelihood of crashing into yourself. When you have collected enough food, you progress onto the next level, where your snake gets longer, and the amount of food to collect to progress through the level gets larger.

To move the snake, 'up arrow' for up, 'down arrow' for down, 'left arrow' for left and 'right arrow' for right. Again, there are constants you can change if you want to alter these settings. Press 'ESC' to exit the game at any time.

To reinforce many of the C and programming concepts you have already met. To provide valuable experience of the design and implementation of a program. To provide a framework for a more challenging, and thus rewarding, laboratory exercise using OpenGL.

ACKNOWLEDGEMENT

We dedicate this page to acknowledge and thank those responsible for the shaping of the project. Without their guidance and help, the experience while constructing the dissertation would not have been so smooth and efficient.

We are extremely thankful to our Director, **Rev. Fr. Wilfred P D'Souza** and our Principal, **Dr. Rio D'Souza** for their support and encouragement.

We owe our profound gratitude to **Dr. Sridevi Saralaya**, Head of the Department, Computer Science and Engineering, whose kind consent and guidance helped us to complete this work successfully.

We sincerely thank **Ms. Anusha S**, Assistant Professor, Computer Science and Engineering for her guidance and valuable suggestions which helped us to fulfil the experiments prescribed by the university.

We would like to thank all our Computer Science and Engineering staff members who have always been with us extending their support, precious suggestions, guidance and encouragement through the project.

We also like to extend thanks to our friends and family members for their continuous support.

TABLE OF CONTENTS

CHAPTER 1 – INTRODUCTION

1.1 About The Project

This report contains the implementation of **'Hungry Snake Game'** using a set of OpenGL functions. The following is an example game written in C based on the game called 'snake game' which has been around since the earliest days of home computing and has re-emerged in recent years on mobile phones. It isn't the world's greatest game, but it does give you an idea of what you can achieve with a relatively simple C program, and perhaps the basis by which to extend the principles and create more interesting games of your own.

You get scored according to the length of the snake and the number of 'x' obstacles on the screen. The speed increases every 5 level. You get a bonus when you complete the level of 1000, increasing by 1000 each level. There is no concept of lives. Once you hit an obstacle, that's it, game over. Make sure you do not have the caps lock on, otherwise the keys will fail to respond

Computer Graphics is concerned with all aspects of producing pictures or images using a computer. A particular graphics software system called OpenGL, which has become a widely accepted standard for developing graphics applications.

The applications of computer graphics in some of the major areas are as follows

1. Display of information.
2. Design.
3. Simulation and Animation.
4. User interfaces.

OpenGL is a software interface to graphics hardware. This interface consists of about 150 distinct commands that you use to specify the objects and operations needed to produce interactive three-dimensional applications.

Pictorial synthesis of real/imaginary objects from computer-based models is Computer Graphics. Graphics provides one of the most natural means of communicating with a computer, since our highly developed 2D and 3D pattern-recognition abilities allow us to perceive and process pictorial data rapidly and efficiently. Interactive computer graphics is the most important means of producing pictures since the invention of photography and television. It has the added advantage that, with computer, we can make pictures not only of concrete real world objects but also of abstract, synthetic objects, such as mathematical surfaces and of data that have no inherent geometry, such as survey results.

1.2 Problem Definition

HUNGRY SNAKE GAME is an OpenGL Computer Graphics Mini Project using OpenGL functions. The aim of the game is to collect the dots (food). As you collect food, the snake gets longer, so increasing your likelihood of crashing into yourself. When you have collected enough food, you progress onto the next level, where your snake gets longer, and the amount of food to collect to progress through the level gets larger.

1.3 Scope & Importance

Snake is a classic game that requires players to assess their surroundings and find the quickest or safest route to a point. This is an excellent opportunity to learn about spatial awareness and plan ahead to your next move.

- The classic game is infamous for using your own success against you when you become so long that you get in your own way. Whilst many games and activities can teach your child about vital life skills, there are not many that would educate on long term strategic planning.
- Snake is a tool that can be used as an educational helping hand. One of the important parts of learning is that you will never get something right the first time. Snake teaches children that practice makes perfect when it comes to learning new skills.

CHAPTER 2 – ABOUT OPENGL

2.1 Computer Graphics

Graphics provides one of the most natural means of communicating with a computer, since our highly developed 2D and 3D pattern recognition abilities allow us to perceive and process pictorial data rapidly and efficiently. Interactive computer graphics is the most important means of producing pictures since the invention of photography and television. It has the added advantage that, with the computer, we can make pictures not only of concrete real world objects but also of abstract, synthetic objects, such as mathematical surfaces and of data that have no inherent geometry, such as survey results.

2.2 OpenGL

OpenGL (Open Graphics Library) is a standard specification defining a cross language cross platform API for writing applications that produce 2D and 3D computer graphics. The interface consists of over 250 different function calls which can be used to draw complex 3D scenes from simple primitives. OpenGL was developed by Silicon Graphics Inc. (SGI) in 1992 and is widely used in CAD, virtual reality, scientific visualization, information visualization and flight simulation. It is also used in video games, where it competes with direct 3D on Microsoft Windows Platforms. OpenGL is managed by the non profit technology consortium, the Khronos group Inc.

OpenGL serves two main purposes :

- To hide the complexities of interfacing with different 3D accelerators, by presenting programmer with a single, uniform API.
- To hide the differing capabilities of hardware platforms , by requiring that all implementations support the full OpenGL feature set.

OpenGL has historically been influential on the development of 3D accelerator, promoting a base level of functionality that is now common in consumer level hardware:

- Rasterized points, lines and polygons are basic primitives.
- A transform and lighting pipeline.
- Z buffering .
- Texture Mapping.
- Alpha Blending.

CHAPTER 3 – SOFTWARE REQUIREMENTS SPECIFICATIONS

A software requirement definition is an abstract description of the services which the system should provide, and the constraints under which the system must operate. It should only specify the external behaviour of the system. The requirements are specified below.

3.1 Functional Requirements

In Software Engineering , a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral reqirements describing all the cases where the system uses the functional requirements are captured in use cases.

OpenGL APIs:

If we want to have a control on the flow of program and if we want to interact with the window system then we use OpenGL API'S. Vertices are represented in the same manner internally, whether they are specified as two-dimensional or three-dimensional entities, everything that we do are here will be equally valid in three dimensions. Although OpenGL is easy to learn, compared with other APIs, it is nevertheless powerful. It supports the simple three dimensional programs and also supports the advanced rendering techniques.

GL/glut.h:

We use a readily available library called the OpenGL Utility Toolkit (GLUT), which provides the minimum functionality that should be expected in any modern windowing system. The application program uses only GLUT functions and can be recompiled with the GLUT library for other window system. OpenGL makes a heavy use of macros to increase code readability and avoid the use of magic numbers. In most implementation, one of the include lines.

myDisplay :

The module draws the output on the screen and the functions in it.

myIdle :

This module is used to modify the structure variables when the system is idle.

myReshape :

This module will reshape the window if the window is resized.

3.2 Software Requirements

Operating System: Windows 7 or higher

Language: C Programming

OpenGL API: freeglut

Software: Code Blocks

3.3 Hardware Requirements

Installed Memory (RAM): 2GB or higher

Processor: 1GHz or higher

Hard disk Space: 20 GB availability

Keyboard: Standard QWERTY serial

CD-ROM: Speed 48x and above

Cache memory: 256KB

Display: Standard display with resolution of 1280 x 720 or higher

CHAPTER 4- IMPLEMENTATION

4.1 Header files:

1. **#include<stdio.h>**

   Provides standard input/output functions.
2. **#include<stdlib.h>**

   Provides general utility functions.
3. **#include<GL/glut.h>**

   Provides simple windowing application programming interface for OpenGL.
4. **#include<math.h>**

   Provides mathematical functions.
5. **#include<string.h>**

   Provides string functions.
6. **#include<time.h>**

   Provides date and time functions.

4.2 OPENGL FUNCTIONS:

4.2.1 Build-in Functions

- **glutInit()**

  Used to initialize GLUT.
- **glutInitDisplayMode()**

  Display mode, normally the bitwise OR-ing of GLUT display mode bit masks.
- **glutInitWindowSize()**

  Requests future windows to open at a given width/height.
- **glutCreateWindow()**

  Create a new top-level window.
- **glutReshapeFunc()**

  Sets the reshape callback for the current window. .
- **glutDisplayFunc()**

  Sets the display callback for the current window.

- **glutIdleFunc()**

  Sets the global idle callback..
- **glutSpecialFunc()**

  Sets the Special callback for the current window.
- **glutKeyboardFunc()**

  Sets the Keyboard callback for the current window.
- **glutMainLoop()**

  The standard GLUT event loop entry point.
- **glLoadIdentity()**

  Replaces the current matrix with the identity matrix
- **glMatrixMode()**

  Specify which matrix is the current matrix
- **gluOtho2D()**

  This function sets up a two-dimensional orthographic viewing region.
- **glutPostRedisplay()**

  Mark the current window as needing a redisplay
- **glClearColor()**

  Specify clear values for the color buffers.
- **glClear()**

  Clear buffers to preset values.
- **glPointSize()**

  Specify the diameter of rasterized points
- **glRasterPos()**

  Specify the raster position for pixel operations.
- **glutBitmapCharacter()**

  Renders a bitmap character using OpenGL.

4.2.2 User Defined Functions

- **initLight()**

  This function will help to Configure the lightning.

- **Initialize()**

  This function will initialize the first configurations.

- **resize ()**

  This function will Configure window resize.

- **Write()**

  This function will Write string on the screen.

- **ManipulateViewAngle()**

  This Function will rotate the object according to the Angles.

- **Reset()**

  This Function will reset the snake size and location.

- **WelcomeScreen()**

  This function will display a welcome screen.

- **DrawSnake()**

  This function will drawing the plane ,head ,body of snake.

- **DrawFood()**

  This function will draw the Sphere representing the Food for the snake.

- **GameStatus()**

  This function print the status of the game on the screen.

- **RandomNumber()**

  This function generates Random Numbers for the location of the food that the snake will eat.

- **newFood()**

  Generate the New Food that the snake will eat.

- **collision()**

  This Function Will Check for Collision with body of the snake.

- **Run()**

  This Function will move the snake according to the directions.

- **display()**

  This contains Draw Function and check the value of the Flag.

SOURCE CODE:

```
void DrawSnake()
{
    int  i;
    glPushMatrix();
    ManipulateViewAngle();
    glPushMatrix();
    glColor3f(0.0, 0.6, 0.2);
    glTranslatef(75.0, -16.0, 75.0);
    glScalef(155,5.0,155);
    glutSolidCube(1);
    glPopMatrix();
    glColor3f(1,0,0);
    glTranslatef(_x,-10.0,_z);
    glScalef(0.5,0.5,0.5);
    glutSolidSphere(10,20,20);
    glRotatef(headRotation, 0.0, 1.0, 0.0);
    glColor3f(1,0,0);
    glTranslatef(0,0.0,6.0);
    glScalef(0.8,1.0,1.0);
    glutSolidCone(10,10,20,20);
    glColor3f(1,1,1);
    glTranslatef(-4.0,10.0,0.0);
    glScalef(0.3,0.3,0.3);
    glutSolidSphere(10,20,20);
    glTranslatef(26.0,0.0,0.0);
    glutSolidSphere(10,20,20);
    glPopMatrix();
    for(i=0; i<size; i++)
    {
        glPushMatrix();
        ManipulateViewAngle();
```

```
                glTranslatef(bodyPos[0][i],-10.0,bodyPos[1][i]);
                glColor3f(1,0,0);//Color Red
                glScalef(0.5,0.5,0.5);
                glutSolidSphere(7,20,20);
                glPopMatrix();
        }
}
```

This Functions helps to Drawing the head & the plane .This will draw the plane that the snake will run on. Here we will draw the Head of the snake. Draw the head as a sphere Drawing the body Loop throw the size and draw spheres representing the body and it will locate the spheres.

```
bool collision()
{
    int i;
    for(i=0; i<size; i++)
    {
        if((bodyPos[0][i] == _x && bodyPos[1][i] == _z)
        ||((bodyPos[0][i] >= _x) && (bodyPos[0][i] <= _x + 5)
        && (bodyPos[1][i] <= _z) && (bodyPos[1][i] >= _z - 5)))
        return true;
    }
    return false;
}
```

This Function help to Check for Collision. It detects when Snake body Sphere's coordinates collide to the head coordinates of snake.

CHAPTER 5 – SCREENSHOTS

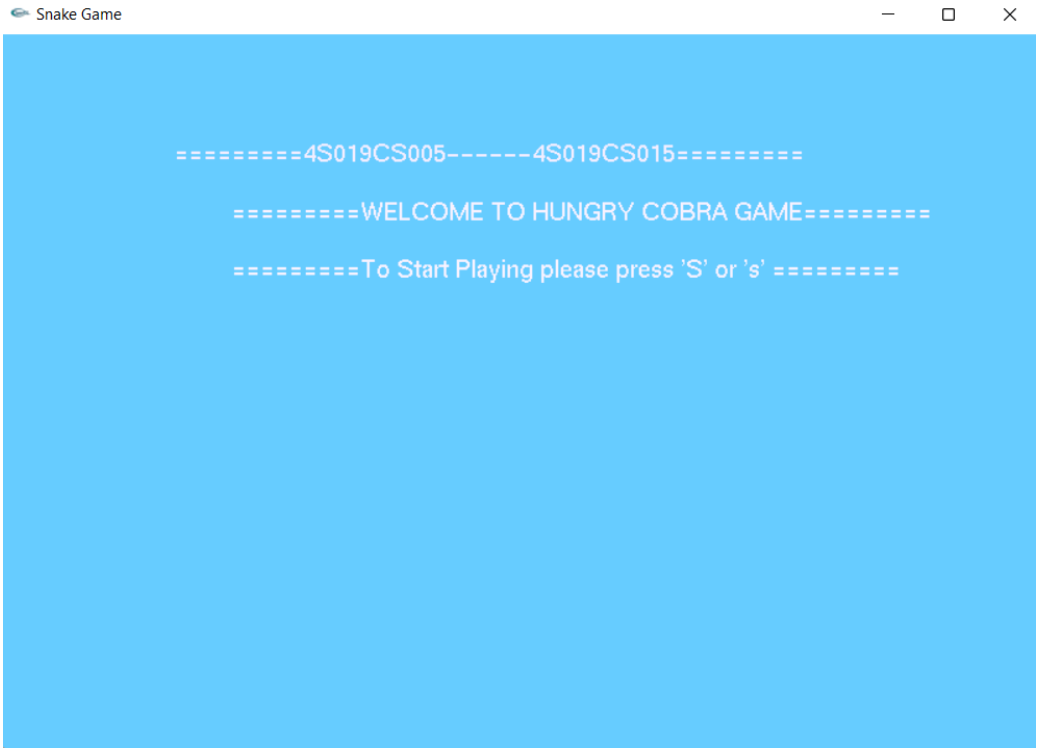

figure 5. 1 : Lauch Screen that display welcome with USN and Name.

figure 5. 2 : Game Start window that waits for user Input.

figure 5. 3 : Game Phase.

figure 5. 4 : Snake eating food

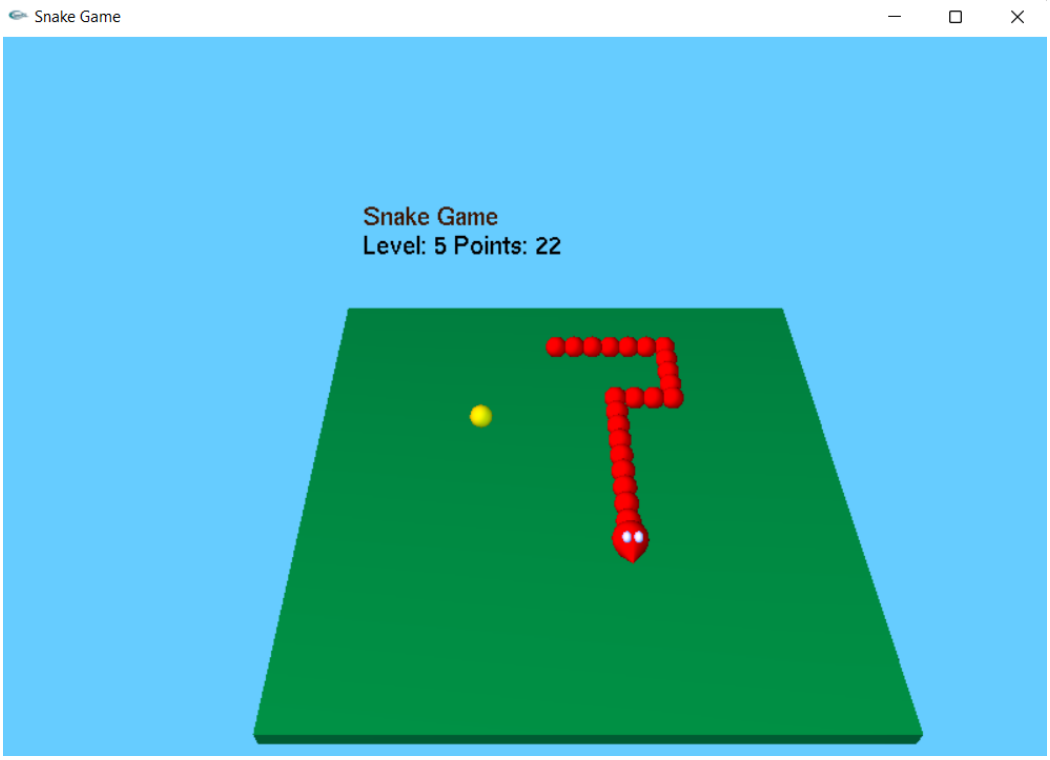

figure 5. 6 : Snake at Level 5 having Increased Size and speed of the snake.

figure 5. 5 : Collision Phase and game will exit.

CHAPTER 6 – CONCLUSION & FUTURE SCOPE

The classic game is infamous for using your own success against you when you become so long that you get in your own way. Whilst many games and activities can teach your child about vital life skills, there are not many that would educate on long term strategic planning.

As many parents will know, it can be extremely frustrating to reach such a high level and then lose as you crash into your own tail. The game requires patience in order to grow and a cool head once you inevitably lose. These are all valuable skills to learn early on in a child's life that will benefit them in later years.

Snake is a tool that can be used as an educational helping hand. One of the important parts of learning is that you will never get something right the first time. Snake game teaches children that practice makes perfect when it comes to learning new skills.

REFERENCES

1. Donald Hearn & Pauline Baker: Computer Graphics with OpenGL Version, 3rd/4th Edition, Pearson Education, 2011

2. Edward Angel: Interactive Computer Graphics- A Top Down approach with OpenGL, 5th Edition, Pearson Education, 2008

3. YouTube (*https://www.youtube.com*)

4. Google (*https://www.google.com*)

5. Stack Overflow (*https://stackoverflow.com*)

6. Geeksforgeeks (*https://www.geeksforgeeks.org*)