NITTE
EDUCATION TRUST

N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India
(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC
☎: 08258 - 281039 – 281263, Fax: 08258 – 281265

Department of Computer Science and Engineering

B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

Report on Mini Project

SNAKE GAME

Course Code 18CS603

CourseName :Computer Graphics

Semester: 6 SEM Section: A

Submitted To:

Mr.Sannidhan M S

Assistant Professor
Department of Computer Science
and Engineering

Submitted By
Blen Joswin Saldanha-4NM18CS039
Dhanush Raj-4NM18CS048

Date of submission: 24-5-2021

Signature of Course Instructor

CERTIFICATE

“Snake Game ” is a simple work carried out by Blen Joswin Saldanha (4NM18CS039) and Dhanush Raj (4NM18CS048) in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering prescribed by N.M.A.M Institute Of Technology,Nitte during the year 2020-2021.

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The Mini project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

ACKNOWLEDGEMENT

We believe that our project will be complete only after we thank the people who have contributed to make this project successful.

We express our deep sense of gratitude and indebtedness to our guide Mr. Sannidhan MS, Assistant Professor , Department of Computer Science and Engineering, for his inspiring guidance, constant encouragement, support and suggestions for improvement during the course of our project.

We sincerely thank Dr.Joythi Shetty, Head of Department of Computer Science and Engineering, Nitte Mahalinga Adyantaya Memorial Institute of Technology, Nitte.

We sincerely thank our beloved principal, Dr. Niranjan N. Chiplunkar for giving us an opportunity to carry out our project work at our college and providing us with all the needed facilities.

We thank all the teaching and non-teaching staff members of the Computer Science and Engineering Department and our parents and friends for their honest opinions and suggestions throughout the course of our project.

Finally, we thank all those who have supported us directly or indirectly throughout the project and making it a grand success.

Blen Joswin Saldanha
(4NM18CS039)

Dhanush Raj
(4NM18CS048)

ABSTRACT

Snake game has been around since the earliest days of home computing and has re-emerged in recent years on mobile phones.

It isn’t the worlds greatest game but it does give an idea of what you can achieve with relatively simple programming,and perhaps the basis by which to extend the principles and create more interesting games of your own.

Snake is a semi-popular video game that originally came out in the late 1970s. In the game, the player controls a long, thin "snake" that continuously moves along the screen. The goal is to eat food items scattered around the field. Each time a food item is consumed, the snake's tail grows longer. The game ends if the snake collides with the boundary, an obstacle, or its own tail.

Table Of Contents

INTRODUCTION

Snake is the common name for the video game concept where the player maneuvers a line which grows in length.

The game called ‘Snake’ or ‘Snake Game’ typically involve the player controlling a line or a snake,there is no official version of the game,so gameplay varies. The version of our game involves the snake or line eating the items which makes it longer ,with the objective being to avoid running into the border or the snake itself for as long as possible.

The player loses when the snake either runs into a border or its own body.The speed of snake also increases as the game progresses.Because of these factors,the game becomes more difficult as it goes on,due to growth of the snake.

DESIGN AND IMPLEMENTATION

Pseudocode:

```
CLASS node
    DECLARE node for a linkedlist
CLASS S
    CONSTRUCTOR S
    DECLARE and INITIALIZE node
    FUNCTION erase
        Clear specified location on screen
     ENDFunction
    FUNCTION redraw(int m,int n)
         Redraws the screen
    ENDFunction
    FUNCTION checkStrangled
        IF snake crosses on itself
            return true
        ELSE
            return false
    ENDFunction

    FUNCTION snake
        temp=head
        WHILE temp!=NULL
            IF temp->next!=NULL
```

```
              Point temp->next to snake's next quad
          IF temp->next = NULL
              call: erase
              erase the snake body
          IF temp==head
          IF snake is moving in free area
              continue the game
              f=1
          ELSE IF snake hits the wall
              game over
              PRINT 'Game Over,Score:' score
          IF call:checkStrangled
              game over
              PRINT 'snake strangled'
              PRINT 'Your score:' score
      IF temp==head
          call:redraw
          draws the updated snake body
      IF snake eats the food
          ++score
          call:new_prey()
          draws new food
  IF f==1
      create new node
ENDfunction
```

```
ENDclass



FUNCTION DisplayText
    Pass In: string to display on the screen
        integer x and y to locate the position
            integer to set the size of font
    Displays the text on the screen
ENDFunction

FUNCTION Timer
    Pass In: integer to store time interval
    Registers a callback function in specified milliseconds
        IF game not over
            call: glutTimerFunc(t,Timer,0)
        ELSE
            IF game over
                Display score
            ELSE
                call: DisplayText(text,x,y,font)
                PRINT 'You won'
ENDFunction
FUNCTION boundary
    IF level=1
        Draw left,right,bottom,top wall/boundary
        Open game without maze
    IF level=2
        Draw left,right,bottom,top wall/boundary
        Open game with maze
ENDFunction
FUNCTION checkNext
    Pass In : Integer value ch representing key pressed
    Checks the direction in which snake should move next
    SWITCH(ch)
        case 1:Move snake upwards
        case 2:Move snake downwards
        case 3:Move snake left
        case 4:Move snake right
ENDFunction

FUNCTION new_prey
    b=true
    Generate and store random number in x
    Generate and store random number in y
    IF x,y out of boundary
        x=40 and y=40
    DO
        IF food plotted on snake body
            call:new_prey
            draws a new food
```

```
                return
    WHILE
    Pass In: b
    IF difficult game level
        Dont draw food in maze area
        x+=40
        y+=40
    Draw food on x,y point
ENDFunction
FUNCTION my_reshape
    Pass In: GLsizei x AND GLsizei y
    glViewport(0,0,x,y)
ENDFunction
FUNCTION my_keyboard
    Pass In: integer key,x,y
    Assign a number to state variable corresponding
    to the up,down,left,right arrow keys pressed to denote the key pressed
ENDFunction
FUNCTION menu
    Pass In:integer num
    call:boundary(),new_prey()
   glutTimerFunc(0, Timer, 0)
   glutPostRedisplay()
   glFlush()
ENDFunction
```

```
FUNCTION createmenu
        Creates the menu for game


        glutCreateMenu(menu)
        glutAddMenuEntry("EASY", 1)
      glutAddMenuEntry("HARD", 2)


        Select the difficulty level based on players choice
ENDFunction
FUNCTION myinit
   srand (time(NULL))
   glClearColor(0.0,0.0,0.0,0.0)
   glClear(GL_COLOR_BUFFER_BIT)
   glMatrixMode(GL_PROJECTION)
   glLoadIdentity()
   gluOrtho2D(0,600,0,600)
ENDFunction
FUNCTION main
        Pass In:Integer argc,character pointer argv
        PRINT 'After starting of the game, first right click using mouse and select the difficulty'
        PRINT 'Press Enter key to start the game'
        Glut functions to create window,set window size,window position
        glutDisplayFunc(my_display)
        glutReshapeFunc(my_reshape)
        glutSpecialFunc(my_keyboard)
        call:createmenu

        glutMainLoop()

        return 0

ENDFunction
```

Conclusion

This snake game is the single player version of the traditional snake game.The game has been successfully tested by us and is working fine.

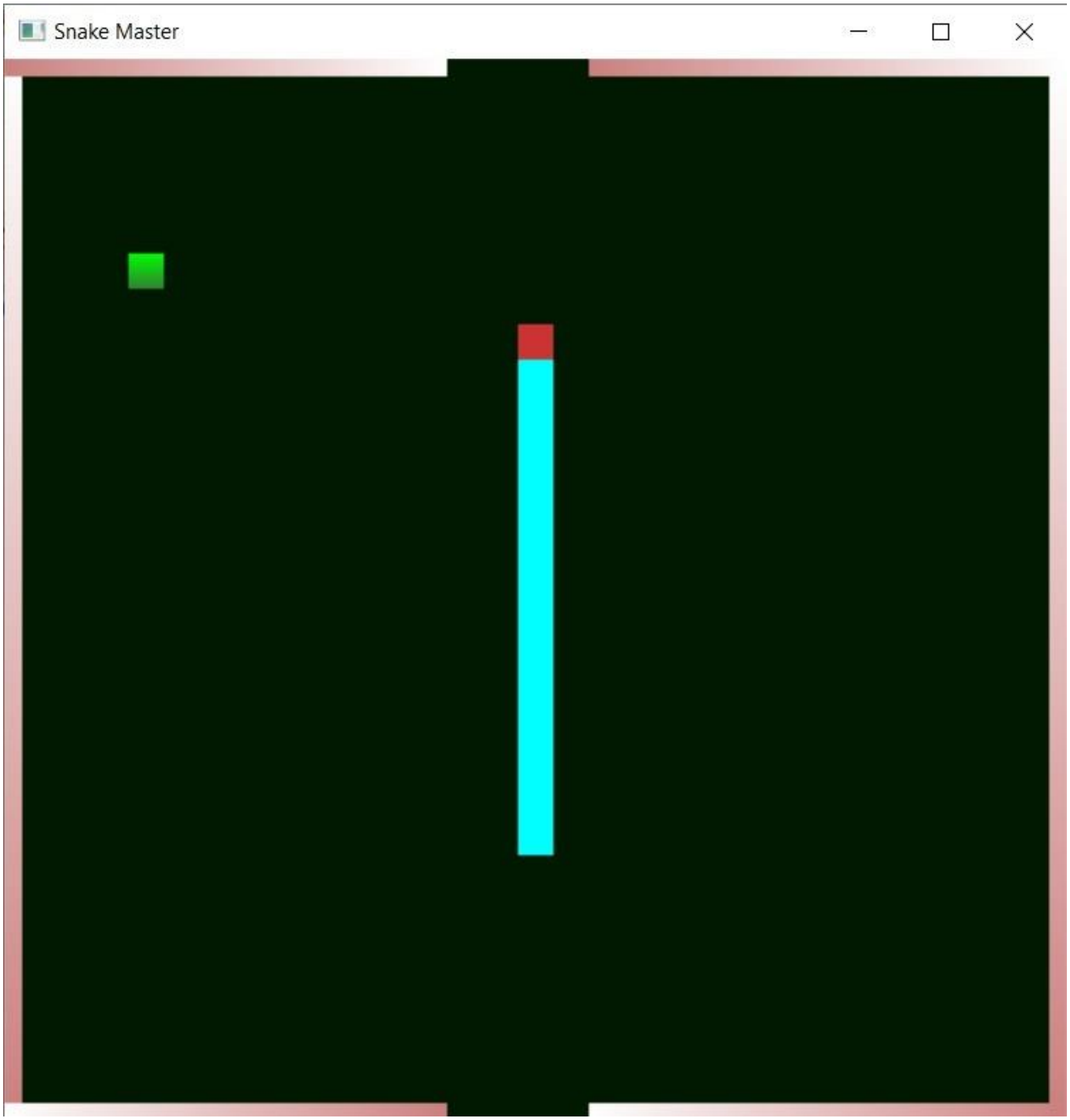
This project helped us in gaining knowledge in many sectors like planning,designing,developing,managing,programming skill,computer graphics and so on.

References

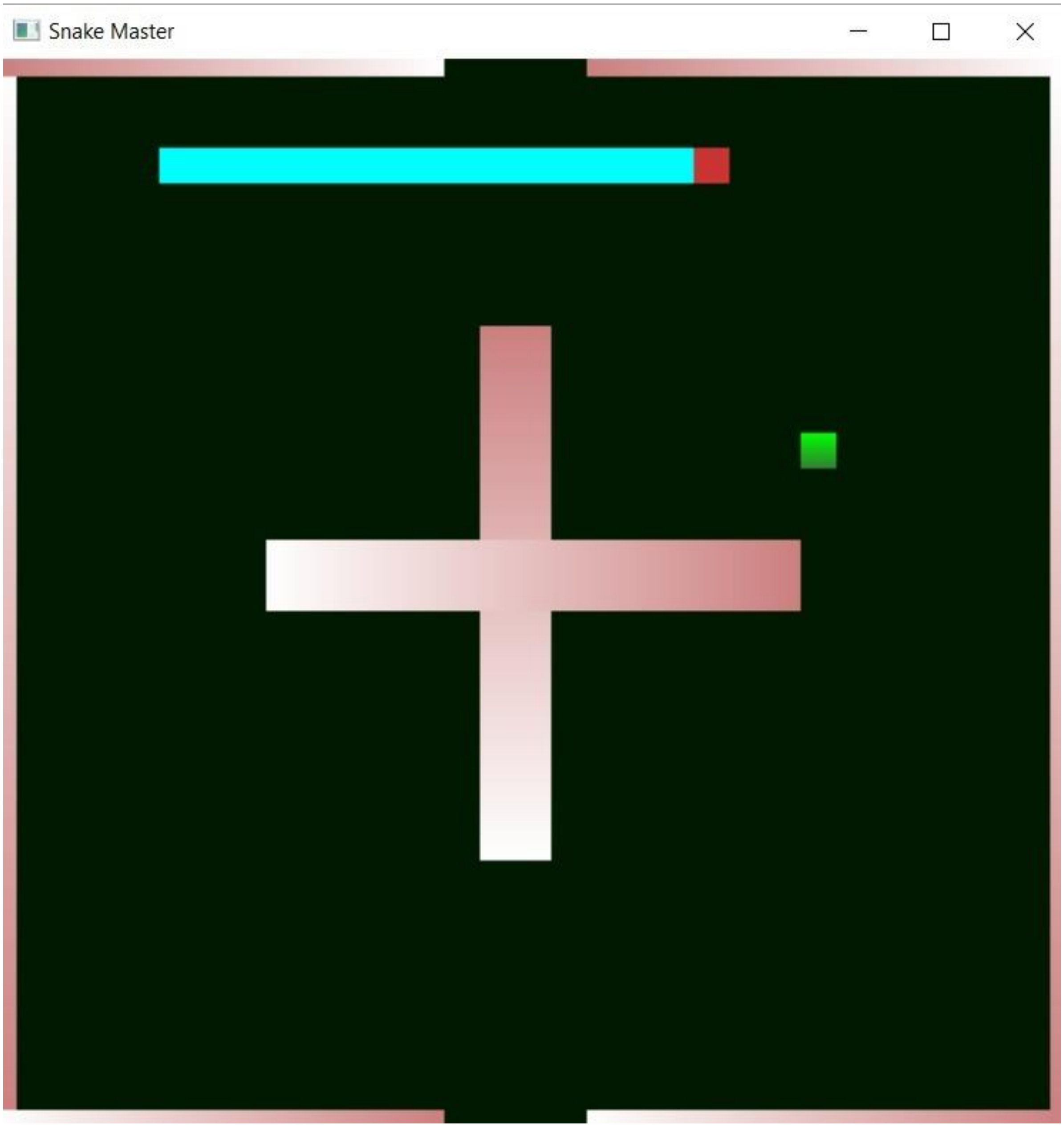
1.https://learnopengl.com/book/offline%20learnopengl.pdf
2.https://www.geeksforgeeks.org/create-a-snake-game-using-turtle-in-python/#:~:text=The%20snake%20in%20the%20Snake,the%20wall%20or%20hits%20itself.
3.https://www.youtube.com/watch?v=6Miai_t_ksw&list=PLWzp0Bbyy_3gXc0YBxiIR9Tb5KfmLSL_C

RESULT

Easy mode: The player gets out if the snake hits to the boundary or hits its own tail.



Difficult mode: The player gets out if the snake hits to the boundary or hits its own tail or the maze.


Snake Master