REPORT:

**PROJECT:**

SNAKE GAME.

**Snake Game**

# Objective:

Gaming industry is a multibillion-dollar industry and has maligned the youth to itself. Being fresh CS students and having interest in gaming, we built a snake game using the practical knowledge of C++ GAINED IN Programming Fundamentals Lab classes.

Following is the detail of the components of our code:

# Libraries:

We used iostream for regular input-output functions, windows,h for mode(), setup(), draw(), input(), logic() functions and conio.h is necessarily for \_kbhit((), \_getch() functions.

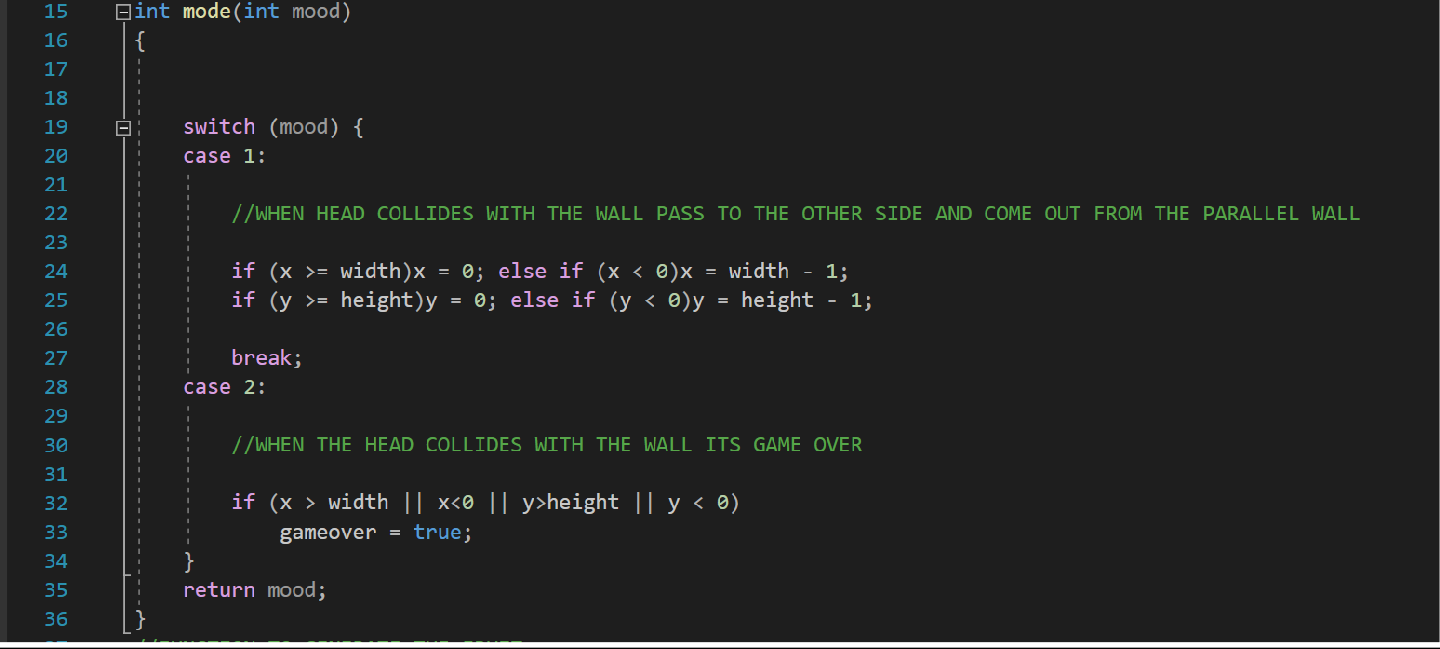
# Variable:

* **gameover** to keep a check on game.
* **width** and **height** are constants which are used to set the size of the walls.
* **choice** for choosing among menu options.
* **mood** for choosing the mode.
* **x** and **y** for snake’s head position.
* **fruitx** and **fruity** for positioning of fruit.
* **score** for showing the score.
* **tailx** and **taily** for positioning of tail.
* **ntail** for the length of tail.
* **eDirection** for control.

# Mood Function:

This function is for the wall mode off the game in which when the head of the snake collides with the wall, it is game over.

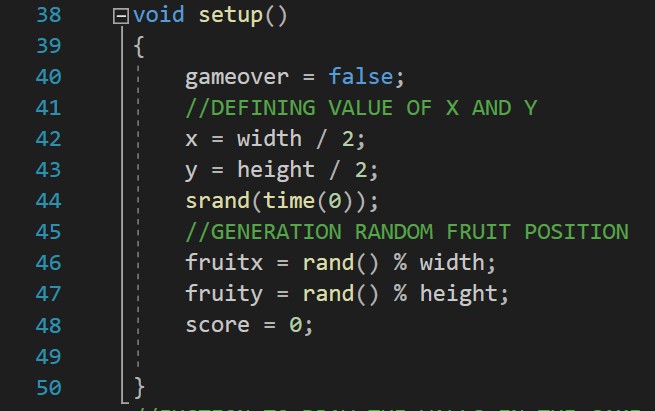
**Line of code:** 15

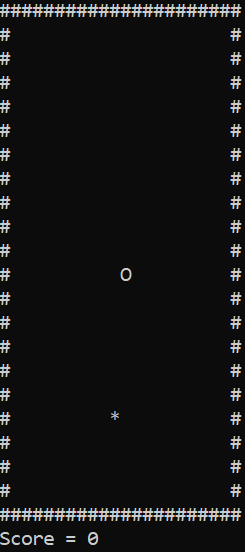


# Setup Function:

Here, we initialized the gameover to false, dir to stop. Though we can generate the snake anywhere, but we have generated it in the middle. And after generating the snake, we then generated the food at any random point but within the height and width of the wall.

**Line of code:** 38

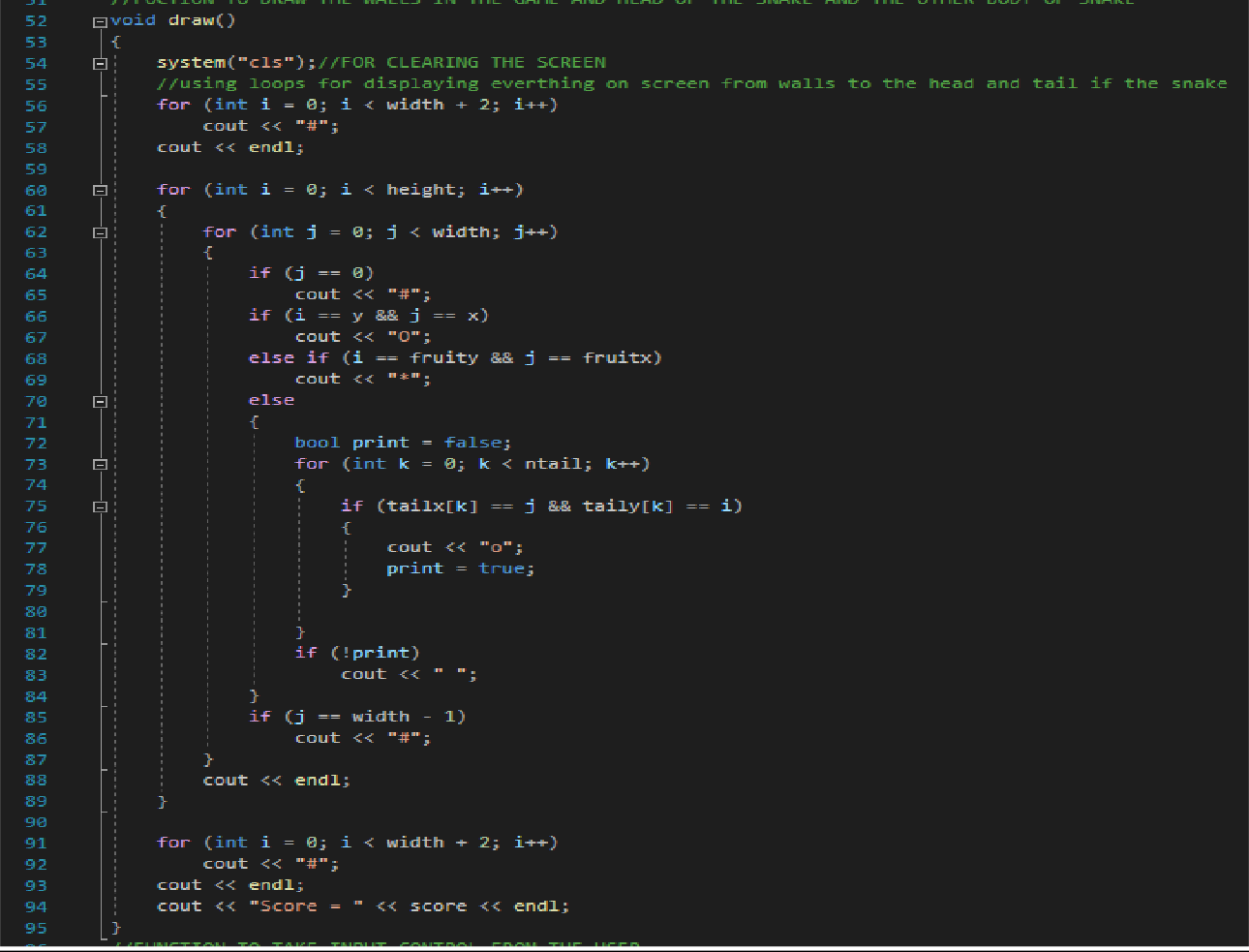




# Draw Function:

Here we just build up the wall boundary. Display the snake from head to tail and the fruit. The walls are shown by ‘#’ character, the snake’s body is shown by ‘o’ characters and the fruit by ‘\*’.

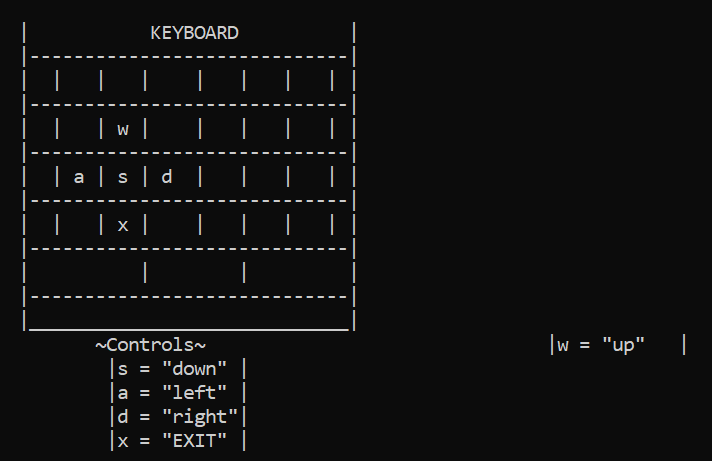
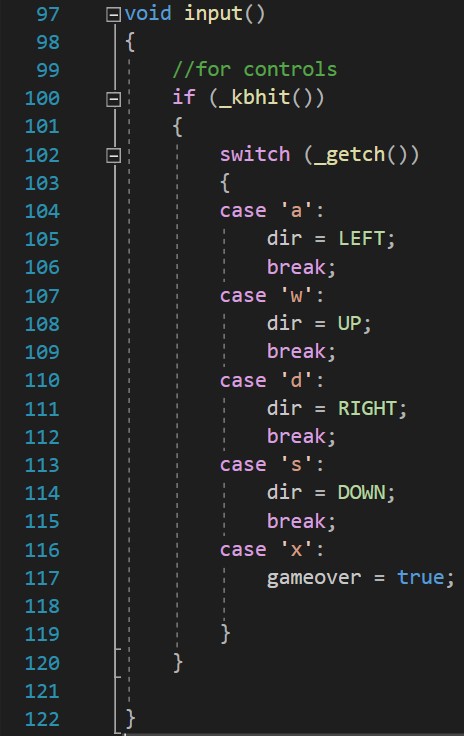
**Line of the code:** 52



# Input Function:

In the input function, we used switch statements and when the \_kbhit() function occurs we just maintained the switch cases (w, a, s, d) and change the direction respectively. The x key is for closing the game.

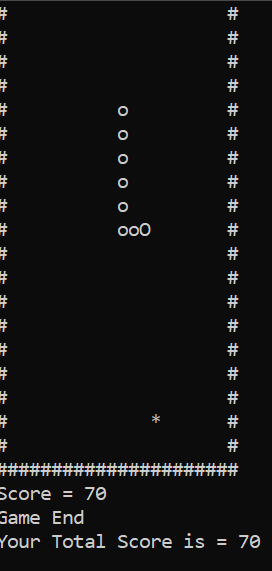
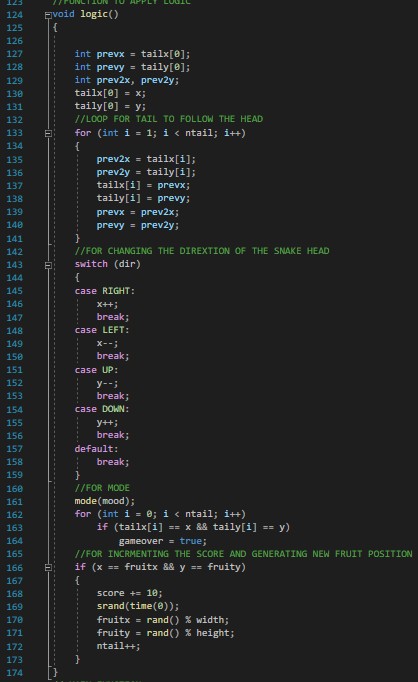
**Line of the code:** 97



# Logic Function:

In the logic function, we first initialized the tail. And after that, we switched the position of the snake’s body with its previous position. And after that, the program simply needs to implement the body according to the keyboard hit. Next, as this follows the concept of an open maze, so when it disappears at one side, it appears from the other side, so we kept in mind that once it touches the right wall it appears from the other left wall and vice-versa and same follows for the up and down walls. Next, the head touches the body, the game crashes. For the scoring system we added 10 points hen the head touches the food (their position becomes the same). And every touch increases the score.

**Line of code:** 124



# Main Function:

The main boy calls all the function and has the instructions to show the main menu and give a response according to the user input.

**Line of code:** 176

# Input:

The user uses the prescribed keyboard keys to play the game:

* **Keys:**

**[W] =** Up

**[A] =** Left

**[S] =** Down

**[D] =** Right

# Output:

Using the keys, the user can toggle through the menu and navigate the snake to get food.

# Program:

#include <iostream>

#include <conio.h>//FOR INPUT OUTPUT PURPOSE #include <windows.h>//FOR FUNCTIONS IN WINDOWS API using namespace std;

//GLOBALLY DECLARE VARIBALE

char again; bool gameover;

const int width = 20, height = 20;

int goback, choice, mood, x, y, fruitx, fruity, score; int tailx[100], taily[100];

int ntail;

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

//FUNCTION TO CHOOSE A MODE

int mode(int mood)

{

switch (mood) { case 1:

//WHEN HEAD COLLIDES WITH THE WALL PASS TO THE OTHER SIDE AND COME OUT FROM THE PARALLEL WALL

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

break; case 2:

//WHEN THE HEAD COLLIDES WITH THE WALL ITS GAME OVER

if (x > width || x<0 || y>height || y < 0) gameover = true;

}

return mood;

}

//FUNCTION TO GENERATE THE FRUIT

void setup()

{

gameover = false;

//DEFINING VALUE OF X AND Y

x = width / 2; y = height / 2; srand(time(0));

//GENERATION RANDOM FRUIT POSITION

fruitx = rand() % width; fruity = rand() % height; score = 0;

}

//FUCTION TO DRAW THE WALLS IN THE GAME AND HEAD OF THE SNAKE AND THE OTHER BODY OF SNAKE

void draw()

{

system("cls");//FOR CLEARING THE SCREEN

//using loops for displaying everything on screen from walls to the head and tail if the snake

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

cout << endl;

for (int i = 0; i < height; i++)

{

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 << "\*";

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;

}

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

cout << endl;

cout << "Score = " << score << endl;

}

//FUNCTION TO TAKE INPUT CONTROL FROM THE USER

void input()

{

//for controls if (\_kbhit())

{

switch (\_getch())

{

case 'a':

dir = LEFT; break;

case 'w':

dir = UP; break;

case 'd':

dir = RIGHT; break;

case 's':

dir = DOWN; break;

case 'x':

gameover = true;

}

}

}

//FUNCTION TO APPLY LOGIC

void logic()

{

int prevx = tailx[0]; int prevy = taily[0]; int prev2x, prev2y; tailx[0] = x; taily[0] = y;

//LOOP FOR TAIL TO FOLLOW THE HEAD

for (int i = 1; i < ntail; i++)

{

prev2x = tailx[i]; prev2y = taily[i]; tailx[i] = prevx; taily[i] = prevy; prevx = prev2x; prevy = prev2y;

}

//FOR CHANGING THE DIREXTION OF THE SNAKE HEAD

switch (dir)

{

case RIGHT:

x++;

break; case LEFT:

x--;

break; case UP:

y--;

break; case DOWN:

y++;

break; default:

break;

}

//FOR MODE

mode(mood);

for (int i = 0; i < ntail; i++)

if (tailx[i] == x && taily[i] == y) gameover = true;

//FOR INCRMENTING THE SCORE AND GENERATING NEW FRUIT POSITION

if (x == fruitx && y == fruity)

{

score += 10; srand(time(0));

fruitx = rand() % width; fruity = rand() % height; ntail++;

}

}

// MAIN FUNCTION

int main()

{

back:

//MAIN MENU

cout << "\t\t\t\t~WELCOME~ :)\n";

cout << "\t\t \n" "\t\t | MAIN MENU | \n"

"\t\t | ----------- | \n"

"\t\t | ~ SNAKE GAME ~ | \n"

"\t\t | | \n"

"\t\t | 1. PLAY (default mode) | \n"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| "\t\t  "\t\t | |  | | 2. MODE | |  | | \n"  \n" |
| "\t\t | | |  | | | \n" |
| "\t\t | | | 3. EXIT | | | \n" |
| "\t\t | | |  | | | \n" |
| "\t\t | | | 4. CONTROLS | | | \n" |
| "\t\t | | | | | | \n\n\n" |
| "\n"; |  | | |  |

cout << "\t\t\tENTER ->>"; cin >> choice;

// FOR SELECTING A MODE

if (choice == 2)

{

cout << "\t\tCHOOSE A MODE FOR THE GAME\n"

<< "\t\tAVAILABLE MODE ARE TWO\n"

<< "\t\tPRESS 1 FOR NO WALLS\n"

<< "\t\tPRESS 2 FOR WALLS\n";

cin >> mood; mode(mood); setup();

while (!gameover)

{

draw();

input();

logic(); Sleep(10);

}

cout << "\t\tGame End" << endl;

cout << "\t\tYour Total Score is = " << score << endl;

}

//FOR DEFAULT MODE AND START GAME GAME

else if (choice == 1)

{

mood = 1; mode(mood); setup();

while (!gameover)

{

draw();

input();

logic(); Sleep(10);

}

cout << "Game End" << endl;

cout << "Your Total Score is = " << score << endl;

}

else if (choice == 3)

{

score = 0;

cout << "\tGame End" << endl;

cout << "\tYour Total Score is = " << score << endl; exit;

}

else if (choice == 4)

{

//SHOWING CONTROLS ON KEYBOARD

cout << endl;

cout << "\t\t\t | KEYBOARD |\n" "\t\t\t | |\n" "\t\t\t | | | | | | | | |\n" "\t\t\t | |\n" "\t\t\t | | | w | | | | | |\n" "\t\t\t | |\n" "\t\t\t | | a | s | d | | | | |\n" "\t\t\t | |\n" "\t\t\t | | | x | | | | | |\n" "\t\t\t | |\n" "\t\t\t | | | |\n" "\t\t\t | |\n" "\t\t\t | |\n";

cout << "\t\t\t\t~Controls~";

cout << "\t\t\t\t |w = \"up\" |\n" "\t\t\t\t |s = \"down\" |\n" "\t\t\t\t |a = \"left\" |\n" "\t\t\t\t |d = \"right\"|\n" "\t\t\t\t |x = \"EXIT\" |\n";

cout << endl;

cout << "\t\t\tPRESS 1 to GoBack ->>"; cin >> goback; cout << endl;

if (goback == 1)

{

goto back;

}

}

}