```
#include <SFML/Graphics.hpp>
#include <SFML/Audio.hpp>
#include <SFML/Window.hpp>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <ctime>
#include <iostream>

/* Tutorial: https://www.sfml-
dev.org/tutorials/2.5/ */
using namespace sf; using namespace
std;
//Snake's Map
int Space = 4;
int N = 75, M = 45;
//The Snake's Map that contain Height
= 45 blocks --- Width = 75 blocks
int size = 16;
//Square Pixel: Width x Height (images
included in nearby folder) ---
Constant data
int Width = size * N;
//Width (w) shows the number of pixel
available (= size x N)
int Height = size * (M + Space);
//Height (h) shows the number of pixel
available (= size x M)

//Obstacle
float ObstacleNumber = 10; int
ObstacleCount = 1;
float ObstacleDuration,
ObstacleRealTime;

//Player
int PlayerNumber = 0; int
scoreTemporary;       //Number of
player who played game (initial)
float delay; int Structable = 0;

//Snake
int dir = 0; int score = 0;
//dir = Direction - Left = 1; Right =
2; Up = 3; Down = 0, intitialize score
= 0
int Snake_Length = 1;
//Snake_Length = Length of the Snake

//Structing class
struct snake {int x, y;}  Snake[100],
Snake_2[100];
//Make object named Snake
with two data shows position (x, y)
and set it as array
struct Fruit {int x, y;}  Food; int
FoodChoice;
//Make object named Food
with two data shows position (x, y)
struct obstacle {int x, y;}
Obstacle[100];
//Make object named obstacle
with two data shows position (x, y)
and set it as array
struct achievements {char name[100] ;
int score;} Player[1000],
Player_2[1000]; int n;   //Make object
named achievements      with two data
shows name and score and set it as
array

//Compulsory
Window window0, window, window1; Font
font;
String playerInput; Text playerText;
Texture t12; int RestartStatus = 0;
float RealTime;

//Music
Music music;
float MusicDuration, EndingMusicTime;
int MusicChoice, MusicStatus;

//Function
void GUI();
void SingleEasy(); void SingleHard();
void SingleBrutal();
void DoubleEasy(); void DoubleHard();
void DoubleBrutal();
```

```
// Snake #2
int dir_2 = 0; int Snake_Length_2 = 1;
int score_2 = 0;

void EasyMode ()
{
        //Snake Movement Algorithm for
the whole body -- Starting from its
tails
        for (int i = Snake_Length; i >
0 ; --i)
    {
                Snake[i].x = Snake[i-
1].x;
                Snake[i].y = Snake[i-
1].y;
        }
        // Snake Movement Algorithm
for the head movement
    if (dir == 0) Snake[0].y += 1;
if (dir == 1) Snake[0].x -= 1;
        if (dir == 2) Snake[0].x += 1;
if (dir == 3) Snake[0].y -= 1;
        // If Snake eats Food
        if ((Snake[0].x == Food.x) &&
(Snake[0].y == Food.y))
    {
                if (FoodChoice == 1)
{score = score + 2;}
//Apple
                if (FoodChoice == 2)
{score = score + 5;}
//Melon
                if (FoodChoice == 3)
{score = score + 10;}
//Strawberry

                Snake_Length =
Snake_Length + 1;
//Increment the Snake's Length by 1
                Food.x = rand() % N;
//Set up new position for food at x-
axis
                Food.y = rand() % M +
Space;                  //Set up new
position for food at y-axis
                // Setting the next
food
                FoodChoice = rand() %
3 + 1;
                if (FoodChoice == 1)
{t12.loadFromFile("Images/Fruit/Apple.
png");          Sprite sprite12(t12);}
//Apple
                if (FoodChoice == 2)
{t12.loadFromFile("Images/Fruit/Melon.
png");          Sprite sprite12(t12);}
//Melon
                if (FoodChoice == 3)
{t12.loadFromFile("Images/Fruit/Strawb
erry.png");     Sprite sprite12(t12);}
//Strawberry
        }
        //If Snake Passed Wall -- Give
him available at the other size (for
Easy Mode)
    if (Snake[0].x > N - 1)
Snake[0].x = 0;
        if (Snake[0].x < 0)
Snake[0].x = N - 1;
    if (Snake[0].y > M - 1 + Space)
Snake[0].y = Space;
        if (Snake[0].y < Space)
Snake[0].y = M - 1 + Space;
}

void ObstacleDelete()
{
        for (int i = 0; i <
ObstacleNumber; i++)
        {
                Obstacle[i].x = - 1;
                Obstacle[i].y = - 1;
        }
}

void Restart()
{
        Snake[0].x = 0; Snake[0].y =
Space; score = 0; dir = 0;
        Snake_2[0].x = 1; Snake_2[0].y
```

```
= Space; score_2 = 0; dir_2 = 0;
        Food.x = (N - 1) / 2; Food.y =
(M - 1 + Space) / 2;
        ObstacleDelete();
ObstacleNumber = 10;
        MusicChoice = rand() % 6 + 1;
MusicPlay();
}

void GUI()
{
        RenderWindow
window0(VideoMode(1200, 784), "Snake
Game!", Style::Default);

window.setVerticalSyncEnabled(true);
        while (window0.isOpen())
        {
                Event e;
                while
(window0.pollEvent(e))
                {
                        if (e.type ==
Event::TextEntered) {playerInput
+=e.text.unicode;
playerText.setString(playerInput);
}
                        if (e.type ==
Event::Closed) {window0.close();}
                        if (e.type ==
Event::KeyPressed)
                        {
                                if
(Keyboard::isKeyPressed(Keyboard::Esca
pe)) {window0.close(); GUI();
}
                                if
(Keyboard::isKeyPressed(Keyboard::S)
and
Keyboard::isKeyPressed(Keyboard::E))
                                {

//Name Conversion

int n = playerInput.getSize() - 1;

for (int i = 0; i < n; i++)
{Player[PlayerNumber].name[i] =
playerInput[i];}

Player[PlayerNumber].name[n] = '_';
Player[PlayerNumber].name[n+1] = 'S';
Player[PlayerNumber].name[n+2] = 'E';

Restart(); window0.close();
SingleEasy(); RestartStatus = 0;
Restart();
                                }
                        }
                }

                Texture t0, t1;

t0.loadFromFile("Images/Screen/Opening
#1.jpg"); Sprite sprite0(t0);
//Source:
https://apkpure.com/vn/snake-rivals-
new-snake-games-in-
3d/com.supersolid.snake

t1.loadFromFile("Images/Screen/Textbox
.png"); Sprite sprite1(t1);

font.loadFromFile("Images/Times-New-
Romans.ttf");
                //--- Drawing GUI ---
                window0.clear();
                sprite0.setPosition(0,
0); window0.draw(sprite0);

sprite1.setPosition(-15, 500);
                Text text1, text2,
text3, text4;

                text1.setFont(font);
text1.setString("How to play");

text1.setCharacterSize(36);
text1.setFillColor(Color::Red);

text1.setStyle(Text::Bold |
```

```
Text::Underlined);
text1.setPosition(15, 505);
                window0.draw(text1);

                text2.setFont(font);
text2.setString("Player #1: Use Arrows
to control. \nPlayer #2: W(Up),
S(Down), A(Left), D(Right). \n---------
--------------------------------------
-------");

text2.setCharacterSize(24);
text2.setFillColor(Color::White);

text2.setStyle(Text::Bold);
text2.setPosition(15, 555);
                window0.draw(text2);

                text3.setFont(font);
text3.setString("\n\nEsc: Return to
GUI - R: Replay \nG + A: Game Paused -
G + L: Game Played \nM + A: Music
Paused - M + L: Music Played \nS + E =
Single Easy Mode");

text3.setCharacterSize(20);
text3.setFillColor(Color::White);

text3.setStyle(Text::Bold);
text3.setPosition(15, 590);
                window0.draw(text3);

                text4.setFont(font);
text4.setString("SNAKE GAME");

text4.setCharacterSize(100);
text4.setFillColor(Color::Blue);

text4.setStyle(Text::Bold);
text4.setPosition(270, 100);
                window0.draw(text4);

                window0.display();
        }
}
```