Государственное автономное профессиональное образовательное учреждение Свердловской области «Уральский радиотехнический колледж им. А. С. Попова»

**ЛАБОРАТОРНАЯ РАБОТА № 3**

По дисциплине «Информатика»

На тему «Змейка»

|  |  |
| --- | --- |
| Студент: | Катаев Е. А., Веремчук М. Р. |
| Группа: | Р-153 |
| Преподаватель: | Зырянов А. В. |

Екатеринбург

2023

**Оглавление**

[1 Постановка задачи 3](#_Toc128516652)

[2 Правила игры змейки 4](#_Toc128516653)

[2.1 Текстуры 5](#_Toc128516654)

[3 Код 7](#_Toc128516655)

[4 Результат работы 10](#_Toc128516656)

[5 Вывод 13](#_Toc128516657)

# **1 Постановка задачи**

Изучение библиотеки SFML – Simple and Fast Multimedia Library, написание программы с использованием данной библиотеки, а именно: реализация видео-игры "Змейка" на языке программирования C++, который является объектно-ориентированным языком программирования.

# **2 Правила игры змейки**

"Змейка" – это несложная игра, развивающая реакцию. Цель – ползая по всему полю съедать фрукты и заполнить всё поле. При съедании обычной еды, змейка увеличивается на единицу длинны тела, увеличивается скорость и количество очков увеличивается на 10 единиц. При съедании хлебушка так же идёт увеличение длины тела, увеличивается скорость и количество очков увеличивается на 50 единиц. А при съедании баклажана вычитается 10 очков и не увеличивается тело. Так же каждые 5 секунд еда пропадает и появляется в случайном месте.

## **2.1 Текстуры**

Ниже представлены текстуры, использованные для реализации объектов (спрайтов) в игре:

**C:\Users\koteg\Desktop\C++shka\Tetris\Multimedia\texture.jpg**

Рисунок 1 - Текстура

**C:\Users\koteg\Desktop\C++shka\Tetris\Multimedia\snake.png**

Рисунок 2 - Текстура

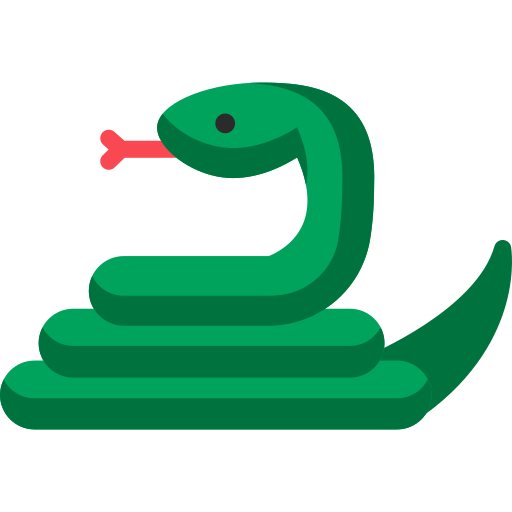
****

Рисунок 3 - Иконка

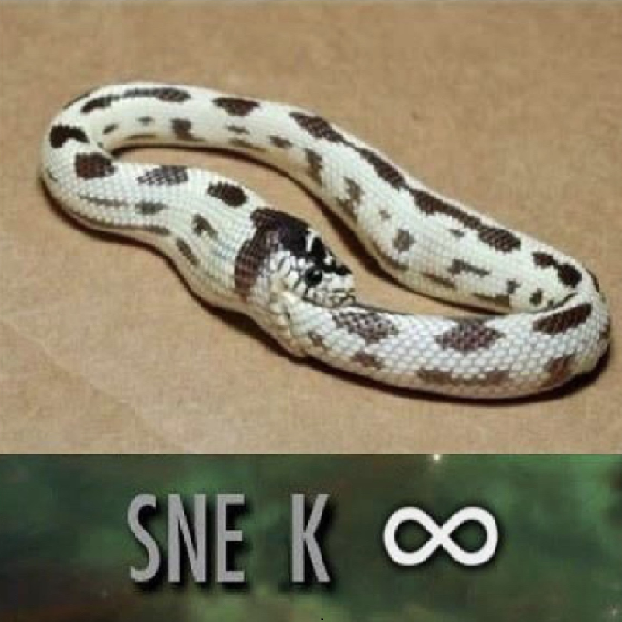
****

Рисунок 4 - Текстура

****

Рисунок 5 - Текстура

# **3 Код**

#include <SFML/Graphics.hpp>

#include <SFML/Audio.hpp>

#include <sstream>

#include "windows.h"

int main() {

srand(time(0));

const int square = 20;

const int size = 32;

sf::RenderWindow window(sf::VideoMode(size \* square, size \* square), L"Змейка", sf::Style::Close);

sf::Image icon;

icon.loadFromFile("Multimedia/icon.png");

window.setIcon(512, 512, icon.getPixelsPtr());

sf::Texture textureSnake;

textureSnake.loadFromFile("Multimedia/snake.png");

sf::Sprite python(textureSnake);

sf::Texture textureFruit;

textureFruit.loadFromFile("Multimedia/texture.jpg");

sf::Sprite fruit(textureFruit);

sf::Texture textureGO;

textureGO.loadFromFile("Multimedia/gameOver.bmp");

sf::Sprite gameOver(textureGO);

sf::Texture textureWin;

textureWin.loadFromFile("Multimedia/win.jpg");

sf::Sprite winner(textureWin);

sf::SoundBuffer sound;

sound.loadFromFile("Multimedia/bite.wav");

sf::Sound bite(sound);

sf::Music music;

music.openFromFile("Multimedia/shakotan.wav");

music.play();

music.setLoop(true);

sf::Font font;

font.loadFromFile("Multimedia/font.ttf");

sf::Text textScore("", font, size);

textScore.setFillColor(sf::Color::Green);

sf::Text textTime("", font, size);

textTime.setFillColor(sf::Color::Green);

sf::Clock clock;

int time = 0;

int score = 0;

struct {

int x;

int y;

} snake[square \* square];

snake[0].x = square / 2;

snake[0].y = square / 2;

int delay = 250;

char go = 's';

int fruitX = rand() % square;

int fruitY = rand() % square;

int food[1] = { rand() % 7 + 1 };

int lenght = 1;

bool game = true;

bool win = false;

while (window.isOpen())

{

sf::Event press;

while (window.pollEvent(press))

{

if (press.type == sf::Event::Closed)

window.close();

if ((press.type == sf::Event::KeyPressed) && (game) && (!win))

{

if ((press.type == sf::Keyboard::W) || (press.key.code == sf::Keyboard::Up) && (go!='d'))

go = 'u';

else if ((press.type == sf::Keyboard::D) || (press.key.code == sf::Keyboard::Right) && (go != 'l'))

go = 'r';

else if ((press.type == sf::Keyboard::A) || (press.key.code == sf::Keyboard::Left) && (go != 'r'))

go = 'l';

else if ((press.type == sf::Keyboard::S) || (press.key.code == sf::Keyboard::Down) && (go != 'u'))

go = 'd';

}

}

if ((game) && (!win))

{

sf::Time timer = clock.getElapsedTime();

time = timer.asSeconds();

for (int i = lenght - 1; i > 0; i--)

{

snake[i].x = snake[i - 1].x;

snake[i].y = snake[i - 1].y;

}

switch (go)

{

case 'u':

python.setRotation(270);

snake[0].y--;

Sleep(delay);

break;

case 'l':

python.setRotation(180);

snake[0].x--;

Sleep(delay);

break;

case 'd':

python.setRotation(90);

snake[0].y++;

Sleep(delay);

break;

case 'r':

python.setRotation(0);

snake[0].x++;

Sleep(delay);

break;

}

}

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

for (int j = 0; j < square; j++)

{

fruit.setTextureRect(sf::IntRect(0, 0, size, size));

fruit.setPosition(i \* size, j \* size);

window.draw(fruit);

}

fruit.setTextureRect(sf::IntRect(food[0] \* size, 0, size, size));

fruit.setPosition(fruitX \* size, fruitY \* size);

window.draw(fruit);

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

{

if (snake[0].x > (square - 1))

snake[0].x = 0;

else if (snake[0].x < 0)

snake[0].x = square - 1;

else if (snake[0].y > (square - 1))

snake[0].y = 0;

else if (snake[0].y < 0)

snake[0].y = square - 1;

if (i == 0)

python.setTextureRect(sf::IntRect(size, 0, size, size));

else if (i == lenght - 1)

python.setTextureRect(sf::IntRect(size \* 2, 0, size, size));

else

python.setTextureRect(sf::IntRect(0, 0, size, size));

python.setOrigin(size / 2, size / 2);

python.setPosition((snake[i].x \* size) + (size / 2), (snake[i].y \* size) + (size / 2));

window.draw(python);

if ((snake[0].x == fruitX) && (snake[0].y == fruitY))

{

if (food[0] == 7)

score -= 10;

else if (food[0] == 6)

{

score += 50;

lenght++;

}

else

{

score += 10;

lenght++;

}

clock.restart();

bite.play();

food[0] = rand() % 7 + 1;

fruitX = rand() % square;

fruitY = rand() % square;

fruit.setPosition(fruitX \* size, fruitY \* size);

if (delay >= 50)

delay -= 10;

}

if (i > 0)

if ((snake[0].x == snake[i].x) && (snake[0].y == snake[i].y))

game = false;

if ((fruitX == snake[i + 1].x) && (fruitY == snake[i + 1].y))

{

food[0] = rand() % 7 + 1;

fruitX = rand() % square;

fruitY = rand() % square;

fruit.setPosition(fruitX \* size, fruitY \* size);

}

if (lenght == ((square \* square) - 1))

win = true;

}

if (win)

{

music.stop();

winner.setScale(0.54, 0.73);

window.draw(winner);

}

if (!game)

{

music.stop();

gameOver.setScale(1.03, 1.03);

window.draw(gameOver);

}

if (time == 5)

{

clock.restart();

food[0] = rand() % 7 + 1;

fruitX = rand() % square;

fruitY = rand() % square;

fruit.setPosition(fruitX \* size, fruitY \* size);

}

std::ostringstream streamScore;

streamScore << score;

textScore.setString("Score: " + streamScore.str());

textScore.setPosition(1, 0);

window.draw(textScore);

std::ostringstream streamTime;

streamTime << time;

textTime.setString("Time: " + streamTime.str());

textTime.setPosition(490, 0);

window.draw(textTime);

window.display();

}

# **4 Результат работы**

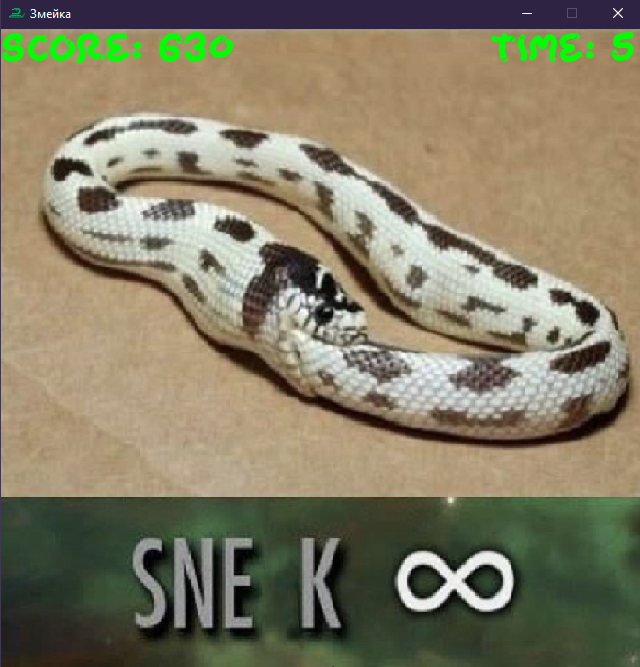
****

Рисунок 6 - Проигрыш



Рисунок 7 - Победа

****

Рисунок 8 - Съедение обычной еды



Рисунок 9 - Съедение хлебушка

****

Рисунок 10 - Съедение баклажана

****

Рисунок 11 - Начало игры

# **5 Вывод**

В ходе лабораторной работы мы научились пользоваться библиотекой SFML, и реализовали игру "Змейка", на основе данной библиотеки в объектно-ориентированном программировании, в среде разработки "Visual studio" версии "community" 2021 года, на языке "C++".