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

**ЛАБОРАТОРНАЯ РАБОТА № 2**

По дисциплине «Информатика»

На тему «Сапёр»

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

Екатеринбург

2023

**Оглавление**

[1 Постановка задачи 3](#_Toc126352128)

[2 Правила игры сапёр 4](#_Toc126352129)

[2.1 Текстуры 5](#_Toc126352130)

[3 Код 6](#_Toc126352131)

[4 Результат работы 11](#_Toc126352132)

[5 Вывод 14](#_Toc126352133)

# **1 Постановка задачи**

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

# **2 Правила игры сапёр**

"Сапер" – это несложная игра, развивающая внимание, память и логику. Цель – открыть все ячейки на поле так, чтобы не "взорвать" ни одну мину. Для поиска взрывоопасных квадратиков можно использовать подсказки:

* Числа в ячейках показывают количество мин, скрытых в восьми находящихся рядом ячейках.
* Если возле открытой ячейки есть пустые клеточки, они откроются автоматически.
* Если вы понимаете, что в какой-то клеточке скрывается бомба, пометьте ее флажком.
* Когда вы откроете ячейку с миной, игра заканчивается проигрышем.
* Для победы в игре нужно открыть все ячейки на поле, пометив все клеточки с бомбой флажком.

## **2.1 Текстуры**

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

****

Рисунок 1 - Текстура

**C:\Users\koteg\Desktop\C++shka\Bomb (Saper)\smileE.jpg**

Рисунок 2 - Текстура

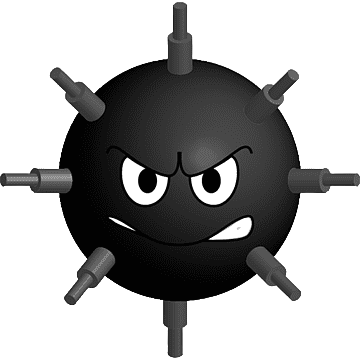


Рисунок 3 - Иконка

# **3 Код**

#include <SFML/Audio.hpp>

#include <SFML/Graphics.hpp>

#include <sstream>

#include "windows.h"

int main() {

const int box = 32;

sf::RenderWindow window(sf::VideoMode(box \* 13, box \* 14), L"Canep P-153", sf::Style::Close);

sf::Image icon;

icon.loadFromFile("icon.png");

window.setIcon(360, 360, icon.getPixelsPtr());

sf::Texture square;

square.loadFromFile("texture.jpg");

sf::Sprite part(square);

sf::Texture smile;

smile.loadFromFile("smile.jpg");

sf::Sprite face(smile);

sf::SoundBuffer bomb;

bomb.loadFromFile("bomb.ogg");

sf::Sound explosive(bomb);

sf::Font font;

font.loadFromFile("font.ttf");

sf::Text textBomb("", font, 40);

sf::Text textTime("", font, 40);

textBomb.setFillColor(sf::Color::Red);

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

int upper[13][14];

int down[13][14];

restart:

bool game = true;

srand(time(0));

int title[1][1] = { 0 };

int countBomb = 0;

int countTime = 0;

for (int i = 0; i < 12; i++)

for (int j = 1; j < 13; j++) {

if ((i > 0) && (j > 1)) {

upper[i][j] = 10;

if (rand() % 10 == 3) {

down[i][j] = 9;

countBomb++;

}

else

down[i][j] = 0;

}

}

int flag = countBomb;

for (int i = 0; i < 12; i++)

for (int j = 1; j < 13; j++) {

if ((i > 0) && (j > 1)) {

int n = 0;

if (down[i][j] == 9)

continue;

if (down[i - 1][j - 1] == 9)

n++;

if (down[i][j - 1] == 9)

n++;

if (down[i + 1][j - 1] == 9)

n++;

if (down[i - 1][j] == 9)

n++;

if (down[i + 1][j] == 9)

n++;

if (down[i - 1][j + 1] == 9)

n++;

if (down[i][j + 1] == 9)

n++;

if (down[i + 1][j + 1] == 9)

n++;

down[i][j] = n;

}

}

while (window.isOpen()) {

sf::Vector2i pos = sf::Mouse::getPosition(window);

int x = pos.x / box;

int y = pos.y / box;

sf::Event click;

while (window.pollEvent(click)) {

if (click.type == sf::Event::MouseButtonPressed) {

if ((click.key.code == sf::Mouse::Left) && (x < 12) && (y < 13)) {

if ((upper[x][y] != 11) && (game) && (x > 0) && (y > 1) && (flag > 0) ) {

upper[x][y] = down[x][y];

if ((down[x][y] == 0) && (x < 11) && (upper[x][y]!=11)) {

if (upper[x - 1][y - 1]!=11)

upper[x - 1][y - 1] = down[x - 1][y - 1];

if (upper[x][y - 1] != 11)

upper[x][y - 1] = down[x][y - 1];

if (upper[x + 1][y - 1] != 11)

upper[x + 1][y - 1] = down[x + 1][y - 1];

if (upper[x - 1][y] != 11)

upper[x - 1][y] = down[x - 1][y];

if (upper[x + 1][y] != 11)

upper[x + 1][y] = down[x + 1][y];

if (upper[x - 1][y + 1] != 11)

upper[x - 1][y + 1] = down[x - 1][y + 1];

if (upper[x][y + 1] != 11)

upper[x][y + 1] = down[x][y + 1];

if (upper[x + 1][y + 1] != 11)

upper[x + 1][y + 1] = down[x + 1][y + 1];

if ((down[x][y - 1] == 0) && (y != 1) && (upper[x][y-2] != 11) && (upper[x-1][y - 2] != 11) && (upper[x+1][y - 2] != 11)) {

y = y - 1;

upper[x - 1][y - 1] = down[x - 1][y - 1];

upper[x][y - 1] = down[x][y - 1];

upper[x + 1][y - 1] = down[x + 1][y - 1];

y = y + 1;

}

if ((down[x][y + 1] == 0) && (y != 13) && (upper[x][y + 2] != 11) && (upper[x - 1][y + 2] != 11) && (upper[x + 1][y + 2] != 11)) {

y = y + 1;

upper[x - 1][y + 1] = down[x - 1][y + 1];

upper[x][y + 1] = down[x][y + 1];

upper[x + 1][y + 1] = down[x + 1][y + 1];

y = y - 1;

}

if ((down[x + 1][y] == 0) && (x != 13) && (upper[x + 2][y - 1] != 11) && (upper[x + 2][y] != 11) && (upper[x + 2][y + 1] != 11)) {

x = x + 1;

upper[x + 1][y - 1] = down[x + 1][y - 1];

upper[x + 1][y] = down[x + 1][y];

upper[x + 1][y + 1] = down[x + 1][y + 1];

x = x - 1;

}

if ((down[x - 1][y] == 0) && (x != 1) && (upper[x - 2][y - 1] != 11) && (upper[x - 2][y] != 11) && (upper[x - 2][y + 1] != 11)) {

x = x - 1;

upper[x - 1][y - 1] = down[x - 1][y - 1];

upper[x - 1][y] = down[x - 1][y];

upper[x - 1][y + 1] = down[x - 1][y + 1];

x = x + 1;

}

}

else if ((down[x][y] != 9) && (x < 11)) {

if ((down[x - 1][y - 1] == 0) && (upper[x - 1][y - 1] !=11))

upper[x - 1][y - 1] = down[x - 1][y - 1];

if ((down[x][y - 1] == 0) && (upper[x][y - 1] != 11))

upper[x][y - 1] = down[x][y - 1];

if ((down[x + 1][y - 1] == 0) && (upper[x + 1][y - 1] != 11))

upper[x + 1][y - 1] = down[x + 1][y - 1];

if ((down[x - 1][y] == 0) && (upper[x - 1][y] != 11))

upper[x - 1][y] = down[x - 1][y];

if ((down[x + 1][y] == 0) && (upper[x + 1][y] != 11))

upper[x + 1][y] = down[x + 1][y];

if ((down[x - 1][y + 1] == 0) && (upper[x - 1][y + 1] != 11))

upper[x - 1][y + 1] = down[x - 1][y + 1];

if ((down[x][y + 1] == 0) && (upper[x][y + 1] != 11))

upper[x][y + 1] = down[x][y + 1];

if ((down[x + 1][y + 1] == 0) && (upper[x + 1][y + 1] != 11))

upper[x + 1][y + 1] = down[x + 1][y + 1];

}

}

if ((upper[x][y] == 9) && (game)) {

explosive.play();

down[x][y] = 12;

game = false;

}

if ((x == 6) && (y == 0))

goto restart;

}

if (click.key.code == sf::Mouse::Right) {

if ((x > 0) && (y > 1) && (flag > 0)) {

if (upper[x][y] == 11) {

upper[x][y] = 10;

countBomb++;

if ((upper[x][y] == 10) && (down[x][y] == 9))

flag++;

}

else if (upper[x][y] == 10 && (countBomb > 0)) {

upper[x][y] = 11;

countBomb--;

if ((upper[x][y] == 11) && (down[x][y] == 9))

flag--;

}

}

}

}

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

window.close();

}

window.clear(sf::Color::Color(192, 192, 192, 0));

for (int i = 0; i < 12; i++)

for (int j = 1; j < 13; j++) {

if ((i > 0) && (j > 1)) {

if (!game)

upper[i][j] = down[i][j];

part.setTextureRect(sf::IntRect(upper[i][j] \* box, 0, box, box));

part.setPosition(i \* box, j \* box);

window.draw(part);

}

}

for (int i = 0; i < 1; i++)

for (int j = 0; j < 1; j++) {

face.setTextureRect(sf::IntRect(title[i][j] \* box, 0, box, box));

face.setPosition(6 \* box, 0);

window.draw(face);

if (!game)

title[i][j] = 1;

if ((flag == 0) && (countBomb == 0))

title[i][j] = 2;

}

std::ostringstream streamBomb;

std::ostringstream streamTime;

streamBomb << countBomb;

streamTime << countTime;

textBomb.setString('0' + streamBomb.str());

textTime.setString("00" + streamTime.str());

textBomb.setPosition(33, 0);

textTime.setPosition(325, 0);

window.draw(textBomb);

window.draw(textTime);

window.display();

}

}

# **4 Результат работы**



Рисунок 2 - Проигрыш



Рисунок 3 - Победа

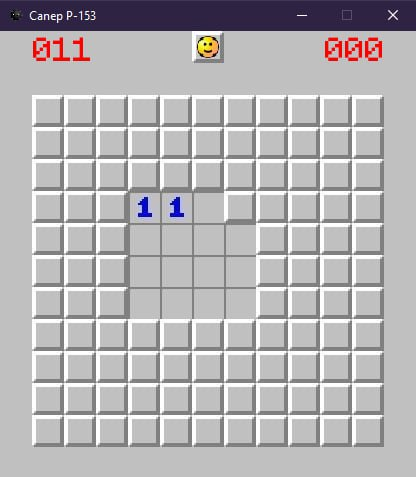


Рисунок 4 - Открытие пустых полей

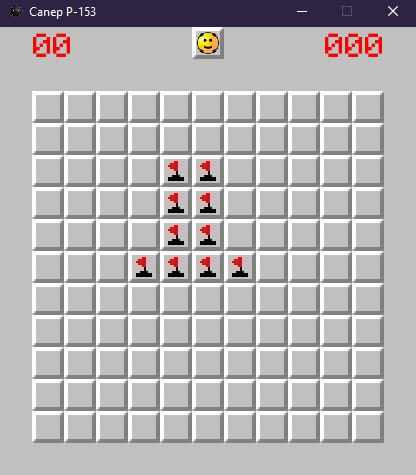


Рисунок 5 - Постановка флагов

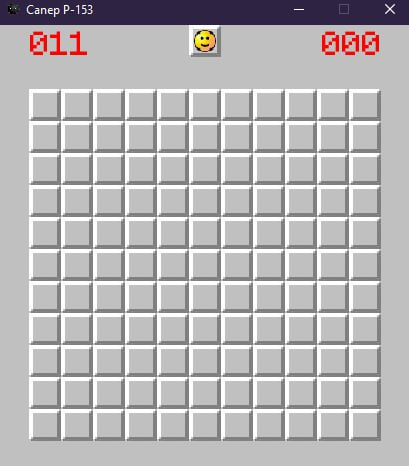


Рисунок 6 - Пустое поле

# **5 Вывод**

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