


# Проект по ВРПО

Авторы: Волковец Сергей, Папыльченко Ян  
и Полуянов Андрей

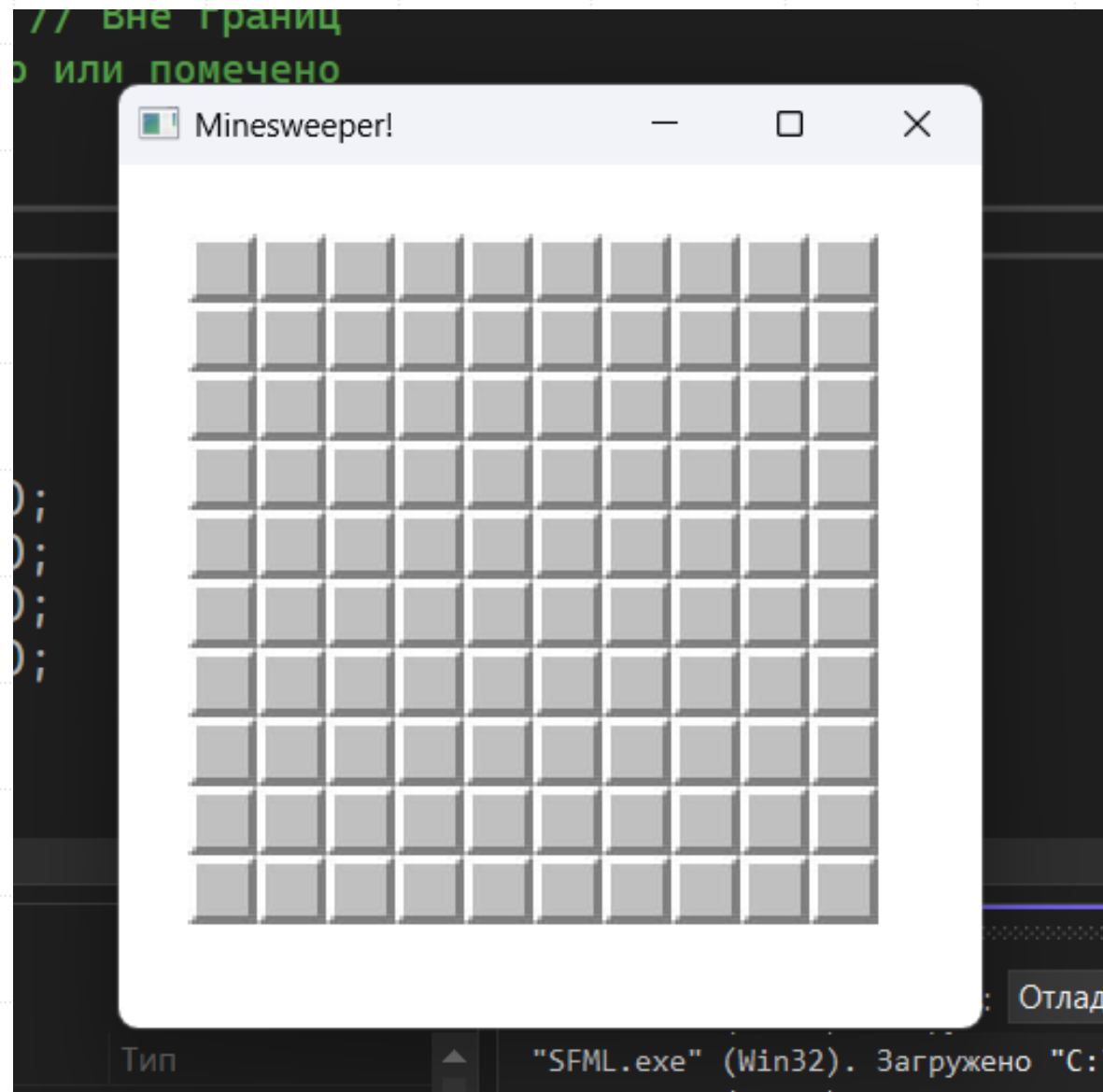
- Темой нашего проекта стала популярная компьютерная игра «Сапёр».
- «Сапёр» — это классическая компьютерная игра-головоломка, в которой игрок должен выявить расположение мин на игровом поле, используя числовые подсказки.

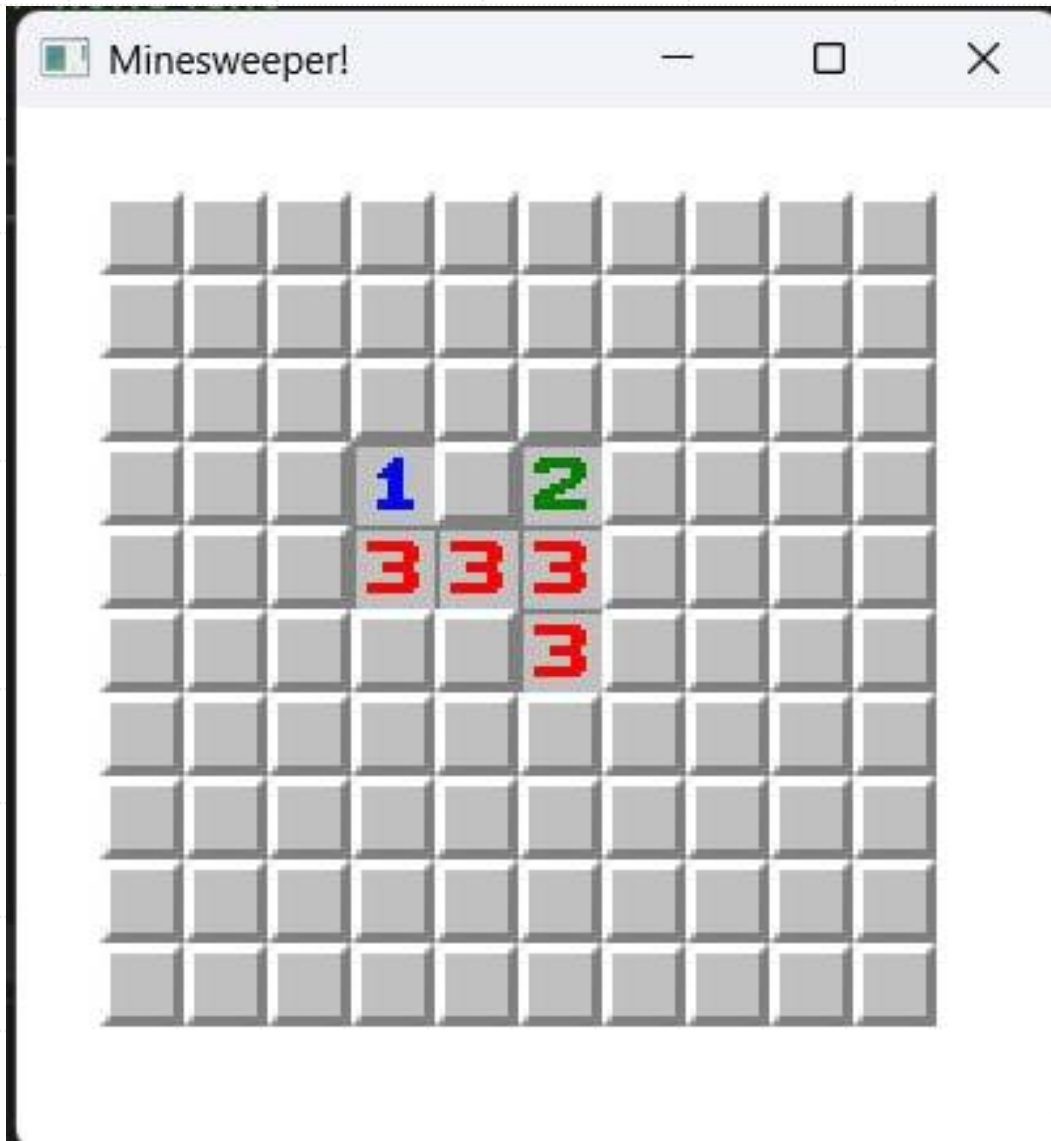


# Презентация будет разбита на 2 основные части:

- Внешний вид игры
- Код игры

Первым делом необходимо было решить, каким будет размер поля, ведь нельзя сделать его слишком большим или слишком маленьким. Мы остановились на варианте 10 на 10.





При первом нажатии по одной из клеток поля на ней самой и на нескольких ближайших клетках появляется цифра; эта цифра отображает количество мин вокруг этой клетки.

Клетки, в которых, по вашему мнению, находятся мины, можно пометить флажком; это обезопасит вас от случайного нажатия.

Игра заканчивается в тот момент, когда все клетки, в которых не находятся мины, открыты.

В случае нажатия на клетку с миной (не флажком) игра автоматически закроется.



Подключение библиотек SFML.

Подключаем библиотеки SFML для работы с графикой и аудио. Используем пространство имен `sf` для удобства.

```
#include <SFML/Graphics.hpp>
#include <SFML/Audio.hpp>
using namespace sf;
```

# Основные функции игры

- `openCells`: Рекурсивно открывает соседние ячейки, если они не содержат мины.
- `doubleClickOpen`: Открывает ячейки при двойном клике, если количество флажков совпадает с числом в ячейке.
- `checkWin`: Проверяет, выиграл ли игрок, открыв все ячейки без мин.

```
void openCells(int x, int y, int gridLogic[12][12], int gridView[12][12])  
    // Логика открытия ячеек  
{  
  
    void doubleClickOpen(int x, int y, int gridLogic[12][12], int gridView[12][12])  
        // Логика двойного клика  
    }  
  
    bool checkWin(int gridLogic[12][12], int gridView[12][12]) {  
        // Проверка на победу  
    }
```



# Основные функции кода

Инициализируем сетки. Расставляем мины случайным образом. Вычисляем числа в ячейках, которые показывают количество мин вокруг.

```
for (int i = 1; i <= 10; i++)
    for (int j = 1; j <= 10; j++) {
        gridView[i][j] = 10;
        if (rand() % 5 == 0) gridLogic[i][j] = 9; // Установка мин
        else gridLogic[i][j] = 0;
    }
// Установка чисел вокруг мин
```

Инициализируем окно приложения, текстуры и переменные игры. Создаем сетки gridLogic и gridView для логики и отображения игры.

```
int main() {
    RenderWindow app(VideoMode(400, 400), "Minesweeper!");
    srand(time(0));
    int w = 32;
    int gridLogic[12][12];
    int gridView[12][12];
    Texture t;
    t.loadFromFile("C:\\Users\\ASUS\\Desktop\\Дитя C++\\saper.jpg");
    Sprite s(t);
    // Инициализация игры
}
```

Победа и завершение.

Проверяем, выиграл ли игрок. Если да, открываем окно с поздравлением. Закрываем основное окно игры.

```
if (checkWin(gridLogic, gridView)) {  
    // Открытие окна поздравления  
}
```



**BCE!**