Министерство науки и высшего образования

Российской Федерации

НГТУ

Лабораторная работа № 1

Рисование графических примитивов

|  |  |
| --- | --- |
| Выполнил:  Жуков Иван  Группа: АТ-34 | Проверил:  Смагин С. М. |

Новосибирск

2024

# Задание

1. Нарисовать сетку заданного размера ячейки в пикселях, с заданным количеством ячеек по осям X и Y.
2. В заданной ячейке сетки нарисовать окружность или квадрат, и заполнить их каким-либо цветом. Логическое начало координат у сетки - нижний левый угол.
3. Оформить процедуры отрисовки сетки и точки в виде отдельных подпрограмм.

#### ****Описание программы****

Программа запрашивает у пользователя:

1. **Количество квадратов** в сетке.
2. **Размер одного квадрата** (в пикселях).
3. **Координаты точки**, которую нужно отобразить.

На основе введённых данных создаётся окно SDL, в котором:

* Отображается координатная сетка.
* Отмечается точка на указанной позиции.

#### ****Основные функции****

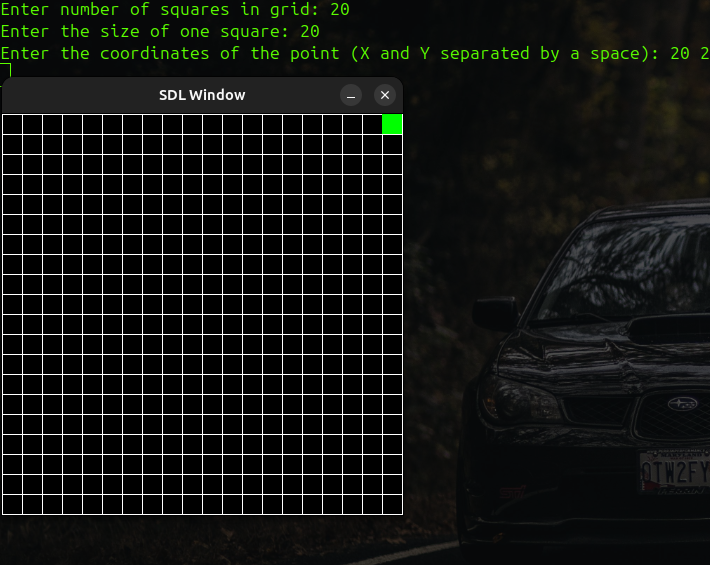
1. DrawGrid
   * Рисует координатную сетку.
   * Использует циклы для прорисовки вертикальных и горизонтальных линий, определяя их положение на основе количества квадратов и их размеров.
   * Устанавливает цвет линий в белый (255, 255, 255, 255).
2. DrawPoint
   * Рисует точку в виде закрашенного квадрата.
   * Выполняет преобразование координат из пользовательской системы (начало в левом нижнем углу) в систему SDL (начало в левом верхнем углу).
   * Отображает точку зелёным цветом (0, 255, 0, 255).
3. **Основной цикл**
   * Запрашивает входные данные.
   * Инициализирует SDL и создаёт окно.
   * Вызывает функции для рисования сетки и точки.
   * Завершает работу после ввода любого значения.

#### ****Алгоритм работы****

1. Пользователь вводит:
   * Количество квадратов numSquares.
   * Размер квадрата squareSize.
   * Координаты точки pointX, pointY.
2. Программа инициализирует SDL:
   * Создаёт окно размером (numSquares \* squareSize + 1) пикселей.
   * Устанавливает чёрный фон.
3. Сетка рисуется с помощью функции DrawGrid, которая:
   * Вычисляет позиции линий на основе размера квадратов.
   * Прорисовывает линии по осям X и Y.
4. Функция DrawPoint преобразует координаты точки в пиксели:
   * Смещает начало координат в нижний левый угол.
   * Рисует точку в соответствующем квадрате.
5. Результат отображается в окне, программа ждёт ввода значения для завершения.

Пример работы





Листинг программы

main.cpp:

#include <SDL2/SDL.h>  
#include <iostream>  
  
void DrawGrid(SDL\_Renderer \*renderer, int numSquares, int squareSize);  
  
void DrawPoint(SDL\_Renderer \*renderer, int x, int y, int squareSize,  
 int numSquares);  
  
int main() {  
  
 int numSquares; // Количество квадратов в сетке  
 int squareSize; // Размер одного квадрата  
 int pointX, pointY; // Координаты точки  
  
 std::cout << "Enter number of squares in grid: ";  
 std::cin >> numSquares;  
 std::cout << "Enter the size of one square: ";  
 std::cin >> squareSize;  
 std::cout  
 << "Enter the coordinates of the point (X and Y separated by a space): ";  
 std::cin >> pointX >> pointY;  
  
 if (SDL\_Init(SDL\_INIT\_VIDEO) != 0) {  
 std::cerr << "Ошибка инициализации SDL: " << SDL\_GetError() << std::endl;  
 return 1;  
 }  
  
 SDL\_Window \*window =  
 SDL\_CreateWindow("SDL Window", SDL\_WINDOWPOS\_CENTERED,  
 SDL\_WINDOWPOS\_CENTERED, numSquares\*squareSize+1, numSquares\*squareSize+1, *SDL\_WINDOW\_SHOWN*);  
 if (!window) {  
 std::cerr << "Ошибка создания окна: " << SDL\_GetError() << std::endl;  
 SDL\_Quit();  
 return 1;  
 }  
  
 SDL\_Renderer \*renderer =  
 SDL\_CreateRenderer(window, -1, *SDL\_RENDERER\_ACCELERATED*);  
 if (!renderer) {  
 std::cerr << "Ошибка создания рендерера: " << SDL\_GetError() << std::endl;  
 SDL\_DestroyWindow(window);  
 SDL\_Quit();  
 return 1;  
 }  
  
 SDL\_SetRenderDrawColor(renderer, 0, 0, 0, 255);  
 SDL\_RenderClear(renderer);  
 DrawGrid(renderer, numSquares, squareSize);  
 DrawPoint(renderer, pointX, pointY, squareSize, numSquares);  
 SDL\_RenderPresent(renderer);  
 int a;  
 std::cin >> a;  
 SDL\_DestroyRenderer(renderer);  
 SDL\_DestroyWindow(window);  
 SDL\_Quit();  
 return 0;  
}  
  
void DrawGrid(SDL\_Renderer \*renderer, int numSquares, int squareSize) {  
 SDL\_SetRenderDrawColor(renderer, 255, 255, 255, 255);  
 for (int i = 0; i <= numSquares; i++) {  
 SDL\_RenderDrawLine(renderer, i \* squareSize, 0, i \* squareSize,  
 numSquares \* squareSize);  
 SDL\_RenderDrawLine(renderer, 0, i \* squareSize, numSquares \* squareSize,  
 i \* squareSize);  
 }  
}  
  
void DrawPoint(SDL\_Renderer \*renderer, int x, int y, int squareSize,  
 int numSquares) {  
 x--;  
 y--;  
 SDL\_SetRenderDrawColor(renderer, 0, 255, 0, 255);  
 SDL\_Rect rect = {x \* squareSize,  
 numSquares \* squareSize - (y + 1) \* squareSize, squareSize,  
 squareSize};  
 SDL\_RenderFillRect(renderer, &rect);  
}