**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра САПР**

**ОТЧЁТ**

**По практической работе №2**

**По дисциплине «Программирование»**

Студент гр. 3351 \_\_\_\_\_\_\_\_\_\_\_\_ Морозов А.А.

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_ Рыжов Н.Г.

Санкт-Петербург

2024

цель работы

Напишите программу, организующую вывод на экран нескольких прямоугольников. Программа должна распознавать следующие команды пользователя: нажатие левой кнопки мыши (ЛКМ) - в месте, определенном положением курсора, рисуется новый прямоугольник стандартного размера; нажатие правой кнопки мыши (ПКМ) изменяется размер текущего прямоугольника (положение курсора задает нижний правый угол прямоугольника), нажатие клавиш «стрелка влево», «стрелка вправо», «стрелка вверх», «стрелка вниз» - последний нарисованный прямоугольник перемещается в указанную сторону, но при этом не должен выходить за пределы окна; Esc – завершение работы программы. Для сохранения информации о положении и размере всех прямоугольников используйте список.

**ТЕКСТ ПРОГРАММЫ**

#include <SFML/Graphics.hpp>

#include <vector>

#include <iostream>

using namespace std;

class coordinateofrectangles {

public:

int x\_position;

int y\_position;

int x\_side;

int y\_side;

coordinateofrectangles(int rectXPosition, int rectYPosition, int rectxSize, int rectYSize) {

x\_position = rectXPosition;

y\_position = rectYPosition;

x\_side = rectxSize;

y\_side = rectYSize;

}

};

int winxSize = 640, winYSize = 480;

int rectXSize = 120, rectYSize = 50;

int rectXPosition = 100;

int rectYPosition = 200;

int number\_of\_rectangles = 0;

void keyPressed(const sf::Event& event);

void mouseButtonPressed(const sf::Event& event);

vector <coordinateofrectangles> rectangles;

sf::RenderWindow window(sf::VideoMode(winxSize, winYSize), "Rectangle Moving");

sf::RectangleShape rectangle;

int main() {

while (window.isOpen()) {

sf::Event event;

while (window.pollEvent(event)) {

switch (event.type) {

case sf::Event::Closed:

window.close();

break;

case sf::Event::KeyPressed:

keyPressed(event);

break;

case sf::Event::MouseButton Pressed :

mouseButtonPressed(event);

break;

default:

break;

}

break;

}

window.clear();

for (int i = 0; i < number\_of\_rectangles; i++) {

rectangle.setSize(sf::Vector2f(rectangles[i].x\_side, rectangles[i].y\_side));

rectangle.setPosition(rectangles[i].x\_position, rectangles[i].y\_position);

window.draw(rectangles);

}

window.display();

}

return 0;

}

void keyPressed(const sf::Event& event) {

switch (event.key.code)

{

case sf::Keyboard::Left:

// analyze pressed key

// if left key is pressed

if (rectangles[number\_of\_rectangles].x\_position - rectXSize < 0)

{

// check if going out of window bounds

break;

}

rectangles[number\_of\_rectangles].x\_position -= rectXSize;

break;

case sf::Keyboard::Right:

if (rectangles[number\_of\_rectangles][1].x\_position > 0)

{

// if right key is pressed

rectangles[number\_of\_rectangles][1].x\_position--;

if (rectangles[number\_of\_rectangles][1].x\_position + rectangles[number\_of\_rectangles][1].x\_side < winxSize)

{

// check if going out of window bounds

break;

}

}

rectangles[number\_of\_rectangles][1].x\_position++;

break;

case sf::Keyboard::Down:

if (rectangles[number\_of\_rectangles][1].y\_position + rectangles[number\_of\_rectangles][1].y\_side < winYSize)

{

// Если кнопка нажата

rectangles[number\_of\_rectangles].y\_position += rectYSize;

}

break;

case sf::Keyboard::Up:

if (rectangles[number\_of\_rectangles][1].y\_position > 0)

{

// Если нажата

rectangles[number\_of\_rectangles][1].y\_position--;

}

break;

case sf::Keyboard::Escape:

if (rectangles[number\_of\_rectangles][1].y\_position > 0)

{

rectangles[number\_of\_rectangles][1].y\_position--;

window.close();

}

break;

default:

break;

}

}

void mouseButtonPressed(const sf::Event& event) { //обработка нажатия кнопок мыши

switch (event.mouseButton.button) { //анализ нажатой клавиши

case sf::Mouse::Button::Left:

// если нажата левая кнопка

rectangles.push\_back(coordinateofrectangles(event.mouseButton.x,

event.mouseButton.y, rectXSize, rectYSize));

rectangle.setPosition(event.mouseButton.x, event.mouseButton.y);

number\_of\_rectangles++;

break;

case sf::Mouse::Button::Right:

if ((rectangles[number\_of\_rectangles - 1].x\_position) > event.mouseButton.x) {

rectangles[number\_of\_rectangles - 1].x\_side = -1;

rectangles[number\_of\_rectangles - 1].x\_position = event.mouseButton.x;

}

else {

rectangles[number\_of\_rectangles - 1].x\_side = 1;

rectangles[number\_of\_rectangles - 1].x\_position = event.mouseButton.x;

}

if ((rectangles[number\_of\_rectangles - 1].y\_position) > event.mouseButton.y) {

rectangles[number\_of\_rectangles - 1].y\_side = -1;

rectangles[number\_of\_rectangles - 1].y\_position = event.mouseButton.y;

}

else {

rectangles[number\_of\_rectangles - 1].y\_side = 1;

rectangles[number\_of\_rectangles - 1].y\_position = event.mouseButton.y;

}

break;

default:

break;

}

}

**Результаты выполнения программы**

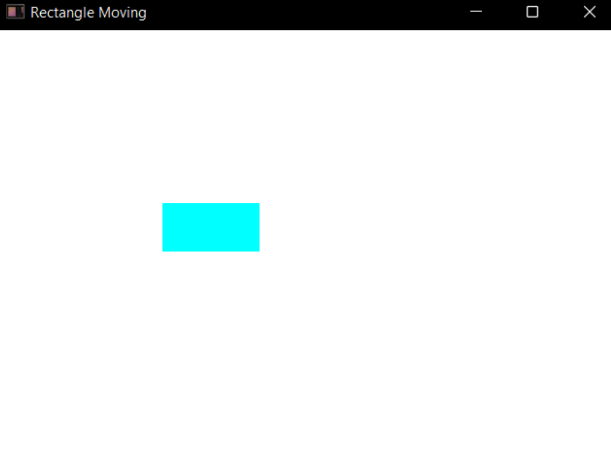


Рис. 1



Рис. 2

При нажатии клавиш вниз, вверх, влево, вправо, последний созданный прямоугольник начнет двигаться в указанном направлении, пока не достигнет края окна.

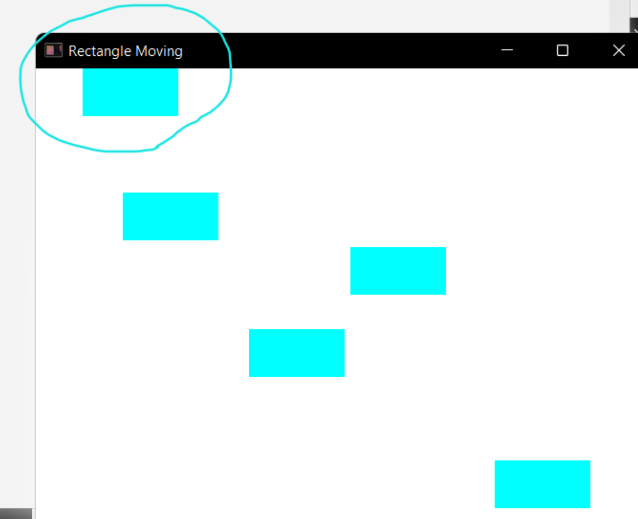


Рис. 3

При наведении курсора на какой-либо прямоугольник и нажатии клавиши ПКМ, прямоугольник, в границах которого находится курсор начнет изменяться: он будет уменьшаться в заданное количество раз, пока не достигнет лимита, который также задан в программе. После достижения лимита фигура вернет свой первоначальный размер.

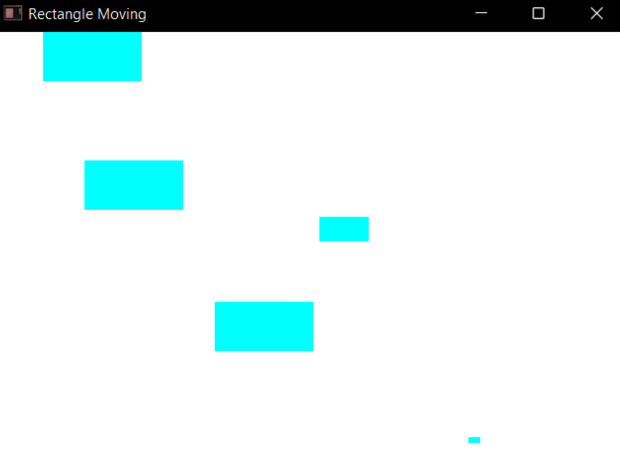


Рис. 4

**Пути дальнейшего улучшения программы**

1. Улучшение визуальной части программы
2. Повышение