|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

|  |  |
| --- | --- |
| **ФАКУЛЬТЕТ** | **ИУК «Информатика и управление»** |
| **КАФЕДРА** | **ИУК4 «Программное обеспечение ЭВМ,** |
| **информационные технологии»** | |

**ДОМАШНЯЯ РАБОТА №1**

**«БИБЛИОТЕКА SFML»**

**ДИСЦИПЛИНА: «Высокоуровневое программирование»**



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил: студент гр. ИУК4-21Б | |  |  | ( | Суриков Н.С | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |
| Проверил: | |  |  | ( | Пчелинцева Н. И. | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |

|  |  |
| --- | --- |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | |
|  | - Балльная оценка:  - Оценка: |

**Цель:** формирование практических навыков реализации графических программ с использованием библиотеки SFML

**Задачи:**

1. Познакомиться с разработкой графических программ на языке программирования C++
2. Изучить основные процедуры и функции библиотеки SFML

**Условие задачи:**

**Вариант 2**

*Задача 1*

Напишите программу, которая строит по периметру экрана семейство разноцветных квадратов, а в середине – закрашенную окружность.

*Задача 2*

Окружность должен исчезать и появляться в различных местах экрана

**Листинг программы:**

1 *#include* *<SFML/Graphics.hpp>*

2 *#include* *<random>*

3

4 *// Функция для генерации случайных координат окружности*

5 std::pair<int, int> generateRandomCirclePosition(const sf::RenderWindow &window, int circleRadius)

6 {

7 std::random\_device rd;

8 std::mt19937 gen(rd());

9 std::uniform\_int\_distribution<> distX(circleRadius, window.getSize().x - 3 \* circleRadius);

10 std::uniform\_int\_distribution<> distY(circleRadius, window.getSize().y - 3 \* circleRadius);

11

12 return std::make\_pair(distX(gen), distY(gen));

13 }

14

15 *// Функция для отрисовки квадратов по периметру экрана*

16 void drawSquares(sf::RenderWindow &window, int squareSize)

17 {

18 for (int i = 0; i < (window.getSize().x - squareSize) / squareSize; ++i)

19 {

20 sf::RectangleShape square(sf::Vector2f(squareSize, squareSize));

21 square.setPosition(i \* squareSize, 0);

22 square.setFillColor(sf::Color(i \* 255 / ((window.getSize().x - squareSize) / squareSize), 0, 0));

23 window.draw(square);

24

25 square.setPosition(i \* squareSize, window.getSize().y - squareSize);

26 square.setFillColor(sf::Color(i \* 255 / ((window.getSize().x - squareSize) / squareSize), i \* 255 / ((window.getSize().x - squareSize) / squareSize), 0));

27 window.draw(square);

28 }

29

30 for (int j = 0; j < (window.getSize().y - squareSize) / squareSize; ++j)

31 {

32 sf::RectangleShape square(sf::Vector2f(squareSize, squareSize));

33 square.setPosition(0, j \* squareSize);

34 square.setFillColor(sf::Color(0, j \* 255 / ((window.getSize().y - squareSize) / squareSize), 0));

35 window.draw(square);

36

37 square.setPosition(window.getSize().x - squareSize, j \* squareSize);

38 square.setFillColor(sf::Color(0, j \* 255 / ((window.getSize().y - squareSize) / squareSize), j \* 255 / ((window.getSize().y - squareSize) / squareSize)));

39 window.draw(square);

40 }

41 }

42

43 *// Функция для отрисовки окружности*

44 void drawCircle(sf::RenderWindow &window, int circleX, int circleY, int circleRadius)

45 {

46 sf::CircleShape circle(circleRadius);

47 circle.setPosition(circleX, circleY);

48 circle.setFillColor(sf::Color::Red);

49 window.draw(circle);

50 }

51

52 int main()

53 {

54 sf::RenderWindow window(sf::VideoMode(800, 600), "SFML Window");

55

56 const int squareSize = 50; *// Размеры квадратов*

57 const int circleRadius = 50; *// Радиус окружности*

58 const sf::Time visibilityTime(sf::seconds(0.1)); *// Время видимости окружности*

59 int circleX{50}, circleY{50}; *// Координаты окружности*

60 sf::Clock visibilityClock; *// Время последнего изменения видимости*

61 bool isCircleVisible = true; *// Флаг видимости окружности*

62

63 while (window.isOpen())

64 {

65 sf::Event event;

66 while (window.pollEvent(event))

67 {

68 if (event.type == sf::Event::Closed)

69 {

70 window.close();

71 }

72 }

73

74 window.clear();

75 drawSquares(window, squareSize);

76 *// Изменение видимости окружности*

77 if (visibilityClock.getElapsedTime() >= visibilityTime)

78 {

79 isCircleVisible = !isCircleVisible;

80 visibilityClock.restart();

81

82 *// Изменение положения окружности при появлении*

83 if (isCircleVisible)

84 {

85 std::tie(circleX, circleY) = generateRandomCirclePosition(window, circleRadius);

86 }

87 }

88 if (isCircleVisible)

89 {

90 drawCircle(window, circleX, circleY, circleRadius);

91 }

92 window.display();

93 }

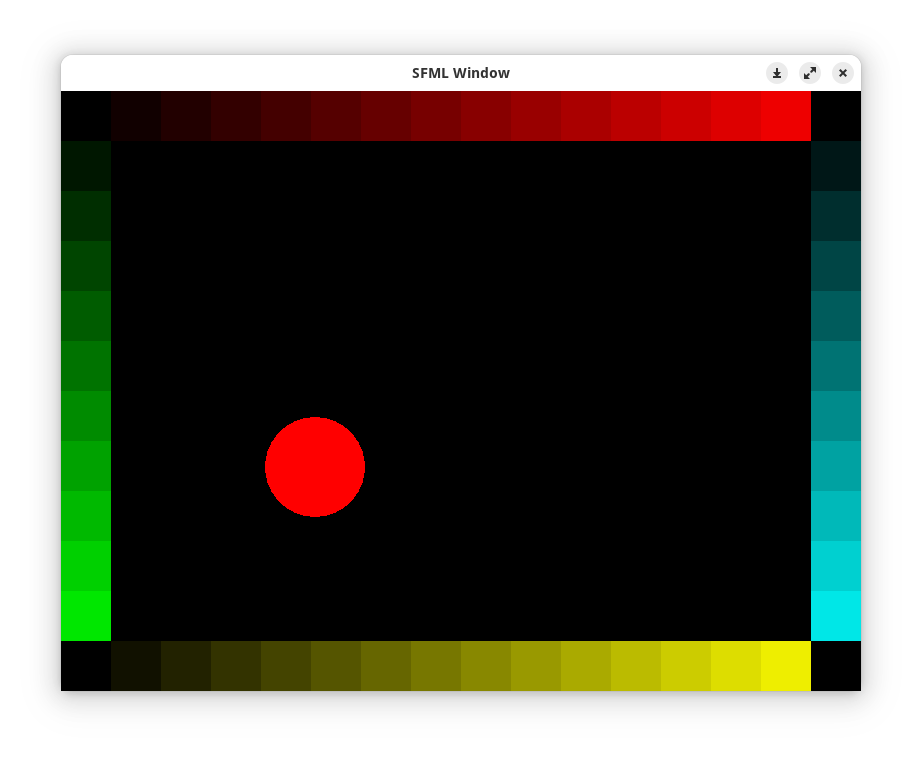
94

95 return 0;

96 }

97

**Демонстрация работы:**



**Вывод:** в ходе выполнения лабораторной работы были получены практические навыки реализации графических программ с использованием библиотеки SFML.