

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра САПР**

**ОТЧЕТ**  
**по курсовой работе**  
**по дисциплине «Программирование»**

Студент гр. 2301

\_\_\_\_\_

Мещеряков М.А.

Преподаватель

\_\_\_\_\_

Рыжов Н.Г.

Санкт-Петербург

2022

**Задание:** сделать экранную заставку "расходящиеся круги". Использовать несколько концентрических окружностей. Радиус каждой из окружностей отличается от ближайшей на равное количество пикселей. Движение создается последовательной сменой цветов всех окружностей, начиная с внутренней окружности и заканчивая внешней. Процесс смены цветов повторяется любое количество раз.

## **Схема алгоритма программы Концентрические окружности**

## Текст программы Концентрические окружности

```
#include <SFML/Graphics.hpp>           // для SFML
#include <iostream>                     // для cin
#include <windows.h>                   // для SetConsoleCP()
#define FRST_CIRCLE_R 55               // Радиус первого самого малого круга
#define RADIUS_DEFFERENCE 55          // Разница между радиусами
#define ANIMATION_SPEED 500           // Задержка проигрывания анимации
#define NUMBER_OF_CIRCLES 7           // Количество окружностей
#define LENGTH_WINDOW 1280            // Длина окна
#define WIGHT_WINDOW 960              // Ширина окна
#define FRST_COLOR 255,0,0            // Первый цвет
#define SCND_COLOR 255,102,0          // Второй цвет
#define THRD_COLOR 255,255,0          // Третий цвет
#define FRTH_COLOR 0,255,0            // Четвертый цвет
#define FVTH_COLOR 0,255,255          // Пятый цвет
#define SXTH_COLOR 0,0,255            // Шестой цвет
#define SVNTH_COLOR 255,0,255         // Седьмой цвет
using namespace sf;                   // пространство имен SF
int main()
{
    SetConsoleCP(1251);                // разрешить русский текст в потоке ввода
    SetConsoleOutputCP(1251);          // разрешить русский текст в потоке вывода
    std::cout << "Для завершения программы закройте окно 'Концентрические окружности' ...";
    int const colums = NUMBER_OF_CIRCLES; // количество столбцов в массиве цветов
    int color_array[colums][3] = { {FRST_COLOR},
        {SCND_COLOR},
        {THRD_COLOR},
        {FRTH_COLOR}, // Все цвета записаны в массив цветов
        {FVTH_COLOR},
        {SXTH_COLOR},
        {SVNTH_COLOR} };

    float circle_r = RADIUS_DEFFERENCE * (NUMBER_OF_CIRCLES + 1); // Радиус самого большого круга.
    Отрисовка кругов начинается с него по убыванию радиусов
    int i, j; // счетчики в циклах
    i = j = 0;
    RenderWindow window(VideoMode(LENGTH_WINDOW, WIGHT_WINDOW), L"Концентрические
окружности"); // создаем окно
    CircleShape circle_(circle_r); // Создаем класс круга
    while (window.isOpen()) //цикл пока графическое окно не закрыто
    {
        Event event;
        while (window.pollEvent(event)) //проверка событий
        {
            if (event.type == Event::Closed) //если поступила команда закрытия окна
                window.close(); // Закрыть окно
        }
        for (int i = NUMBER_OF_CIRCLES - 1; i >= 0; i--) // Цикл рисования кругов
        {
            circle_.setRadius(circle_r - RADIUS_DEFFERENCE); // Установка радиуса
            circle_r = circle_.getRadius(); // запись радиуса в переменную
            circle_.setPosition((LENGTH_WINDOW / 2) - circle_r, (WIGHT_WINDOW / 2) - circle_r); //
установка расположения круга
        }
    }
}
```

```

        circle_.setFillColor(Color(color_array[i][j], color_array[i][j + 1], color_array[i][j + 2])); // установка
цвета
        window.draw(circle_); // вывод круга в буфер
    }
    sleep(sf::milliseconds(ANIMATION_SPEED)); // задержка
    window.display(); // отображение буфера на экран
    i = NUMBER_OF_CIRCLES; // возвращаем значение i на изначальное, равное кол-ву
кругов
    circle_r = RADIUS_DEFFERENCE * (NUMBER_OF_CIRCLES + 1); // возвращаем изначальное значение
радиуса, т.е. радиус самого большого круга
    for (int i = NUMBER_OF_CIRCLES - 1; i > 0; i--) // цикл смены цветов
    {
        std::swap(color_array[i], color_array[i-1]); // функция swap меняет местами i и i+1 элемент в
массиве, т.е. цвета
    }

}
}

```

## **Описание программы Концентрические окружности**

Название программы – Концентрические окружности. Функциональным назначением является вывод экранной заставки смены цветов концентрических окружностей в окно размером 1280 на 960.

Все цвета окружностей записаны в массив, в котором в каждом столбце записан цвет. Установка каждой окружности своего цвета выполняется за счет перебора строк этого массива. Анимация происходит за счет последовательной смены цветов всех окружностей, начиная с внутренней окружности и заканчивая внешней. Отрисовка окружностей выполняется с самого большого и по убыванию до самого малого.

Программист имеет возможность настройки программы путем изменения следующих констант:

FRST\_CIRCLE\_R - Радиус первой самой малой окружности

RADIUS\_DEFFERENCE - Разница между радиусами

ANIMATION\_SPEED - Задержка проигрывания анимации

NUMBER\_OF\_CIRCLES - Количество окружностей

LENGTH\_WINDOW - Длина окна

WIGHT\_WINDOW - Ширина окна

FRST\_COLOR - Первый цвет

SCND\_COLOR - Второй цвет

THRD\_COLOR - Третий цвет

FRTH\_COLOR - Четвертый цвет

FVTH\_COLOR - Пятый цвет

SXTH\_COLOR - Шестой цвет

SVNTH\_COLOR - Седьмой цвет

## Руководство пользователя для программы Концентрические окружности

Название программы – Концентрические окружности. При запуске программы открывается окно размером 1280 на 960 с проигрывающейся анимацией (Рис. 1). Для завершения программы следует закрыть это окно, об этом сообщает текст в текстовом окне (Рис. 2).

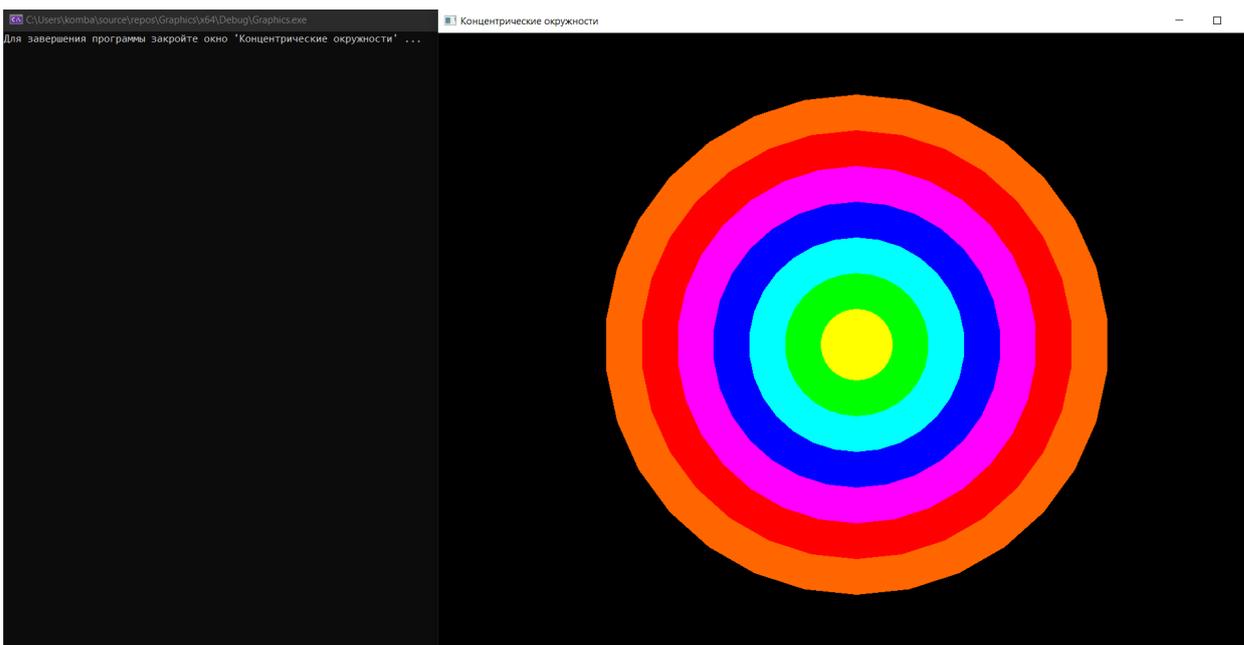


Рис. 1

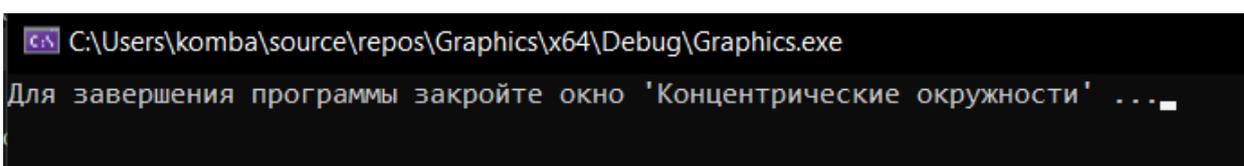


Рис. 2

Пример результата работы программы (Рис. 3).

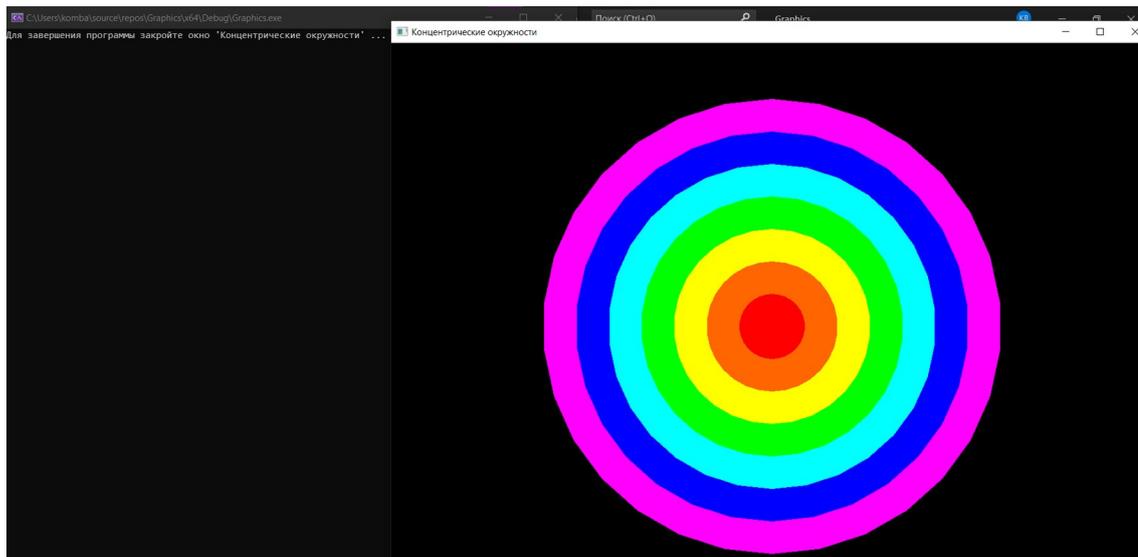


Рис. 3

Путей дальнейшего улучшения программы мной найдено не было.