**Сопроводительный лист на издание в авторской редакции**

Название работы Простейшая анимация Методические указания к выполнению лабораторной работы № 21 для студентов очной формы обучения по направлениям подготовки 09.03.04 – «Программная инженерия», 09.03.01 «Информатика и вычислительная техника», 02.03.03 «Математическое обеспечение и администрирование информационных систем»

Актуальность и соответствующий   
научно-методический уровень подтверждаю \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись научного редактора)

Рукопись сверена и проверена автором \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись автора)

Рекомендуется к изданию \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись заведующего кафедрой)



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

Брянский государственный технический университет

Утверждаю

Ректор университета

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.Н. Федонин

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020

ПРОГРАММИРОВАНИЕ

ПРОСТЕЙШАЯ АНИМАЦИЯ

**Методические указания**

**к выполнению лабораторной работы № 21**

**для студентов очной формы обучения**

**по направлениям подготовки**

**09.03.04 – «Программная инженерия»,**

**09.03.01 «Информатика и вычислительная техника»,**

**02.03.03 «Математическое обеспечение и администрирование информационных систем»**

**Брянск 2020**

**УДК 004.**

Простейшая анимация [Текст] + [Электронный ресурс]: методические указания к выполнению лабораторной работы № 21 для студентов очной формы обучения по направлениям подготовки 09.03.04 – «Программная инженерия», 09.03.01 «Информатика и вычислительная техника», 02.03.03 «Математическое обеспечение и администрирование информационных систем». – Брянск, 2020. – 7 с.

Разработали:

Д. И. Булатицкий

канд. техн. наук, доц.,

Е..В. Коптенок

ассистент

Рекомендовано кафедрой «Информатика и программное обеспечение» БГТУ

**Методические издания публикуются в авторской редакции**

# Цель работы

Цель работы – научиться создавать простейшие анимированные изображения при помощи библиотеки SDL.

Продолжительность работы – 2 часа.

# ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

**Анимация** – это процесс создания иллюзии движения и изменения формы с помощью быстрой смены статичных изображений, которые обычно минимально отличаются друг от друга.

Каждый кадр должен быть нарисован отдельно и показывается на короткий промежуток времени, именно это и создаёт иллюзию движения.

Для отображения анимации следует внутри цикла выполнить следующие действия:

* Очистить кадр;
* Произвести рисование;
* Показать кадр;
* Произвести задержку экрана.

# ХОД РАБОТЫ

**Задание 1**. Воспроизвести движение прямоугольника сверху экрана вниз.

Для начала следует инициализировать графический режим и создать окно:

SDL\_Init(SDL\_INIT\_EVERYTHING);

SDL\_Window\* window = SDL\_CreateWindow("HI",

100, 100,

800, 600, SDL\_WINDOW\_SHOWN);

SDL\_Renderer\* renderer = SDL\_CreateRenderer(window, -1, 0);

Следует заметить, что после выполнения программы память должна быть очищена:

SDL\_DestroyRenderer(renderer);

SDL\_DestroyWindow(window);

SDL\_Quit();

Между двумя этими блоками будет отображаться цикл, реализующий анимацию. Нарисуем прямоугольник, размером 100 единиц.

Рисование будет производиться в цикле с параметром:

for (int i = -100; i <= 600; i += 10) { }

Такие параметры счетчика заданы, исходя из размеров окна с учетом того, что прямоугольник будет появляться из-за границы отображаемой области.

Очистим содержимое окна. Выполнив заливку черным цветом:

SDL\_SetRenderDrawColor(renderer, 0, 0, 0, 0);

SDL\_RenderClear(renderer);

Выполним рисование прямоугольника фиолетового цвета, изменяя его координату по вертикальной оси в соответствии со значением счетчика:

SDL\_SetRenderDrawColor(renderer, 128, 0, 128, 0);

SDL\_Rect target = { 300, i, 100, 100 };

SDL\_RenderFillRect(renderer, &target);

Выведем полученное изображение на экран и выполним задержку экрана.

SDL\_RenderPresent(renderer);

SDL\_Delay(20);

Чем большее значение будет передано функции SDL\_Delay(), тем на большее время будет осуществляться задержка экрана и, следовательно, тем медленнее будет происходить анимационное движение.

**Задание 2.** Реализовать движение прямоугольника по кругу.

Получить координаты на окружности для заданного угла можно следующим образом:

Пусть прямоугольник совершает четыре полных оборота. Зададим параметрами счетчика градусную меру угла от 0 до 1140 (360\*4=1140):

for (int i = 0; i <= 1440; i++) { }

Заливка экрана цветом и установка цвета рисования по сравнению с предыдущим примером не изменится. Для получения значений синуса и косинуса потребуется подключить библиотеку math.h.

Заметим, что эти функции на вход получают значения угла в радианах. Заведем вспомогательную переменную, в которой будет производится пересчет величины угла из градусов в радианы:

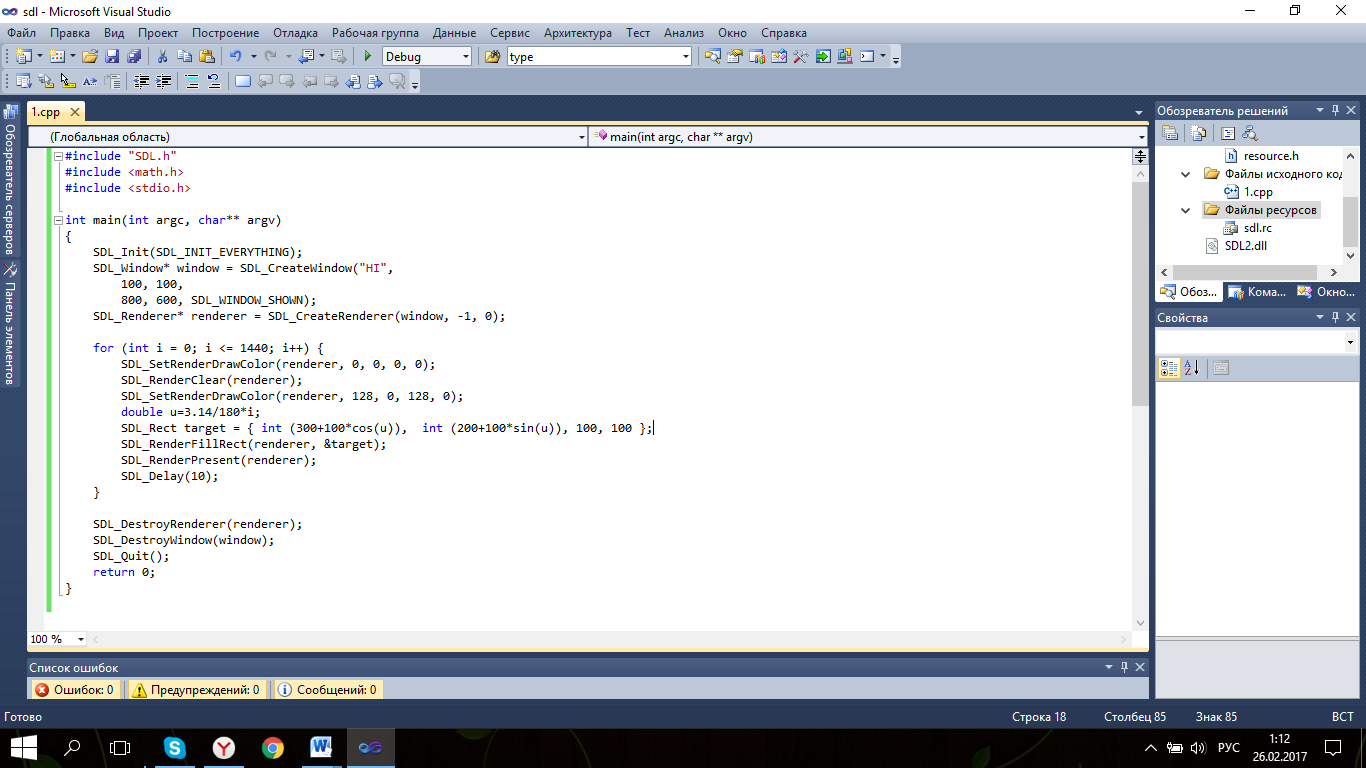
double u=3.14/180\*i;

Далее создадим структуру, описывающую прямоугольник, которую зададим следующим образом:

SDL\_Rect target = { int (300+100\*cos(u)), int (200+100\*sin(u)), 100, 100 };

Здесь радиус окружности равен 100, кроме того, задано смещение относительно начала координат 300 по горизонтали и 200 по вертикали для того, чтобы рисунок не выходил за границу окна. Стоит заметить, что результаты вычислений следует привести к целому типу.

Полученный код приведен на Рис.3.1.



Ри.3.1. Код программы

# ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОГО ВЫПОЛНЕНИЯ

Реализовать следующие анимационные сюжеты:

1. Падающий снег;
2. Ползущая змейка (змейка из 4 квадратов обходит внутреннюю границу экрана и возвращается в исходную точку);
3. Движение прямоугольника по диагонали;
4. Равноускоренное движение прямоугольника;
5. Падение от верхнего края экрана с ускорением;
6. Движение трех точек по круговой орбите, радиус задает пользователь.
7. Падающий снег.
8. Падающий снег.
9. Движение двух окружностей вокруг центра по эллиптической орбите. Параметры эллипса задает пользователь.
10. Падающий снег.
11. Равноускоренное движение прямоугольника;
12. Движение трех точек по круговой орбите, радиус задает пользователь.
13. Движение прямоугольника по диагонали;
14. Движение прямоугольника по диагонали;
15. Падающий снег.
16. Движение трех точек по круговой орбите, радиус задает пользователь.
17. Движение двух окружностей вокруг центра по эллиптической орбите. Параметры эллипса задает пользователь.
18. Равноускоренное движение прямоугольника;
19. Движение двух окружностей вокруг центра по эллиптической орбите. Параметры эллипса задает пользователь.
20. Движение прямоугольника по диагонали;
21. Движение двух окружностей вокруг центра по эллиптической орбите. Параметры эллипса задает пользователь.
22. Падающий снег.
23. Ползущая змейка (змейка из 4 квадратов обходит внутреннюю границу экрана и возвращается в исходную точку);
24. Равноускоренное движение прямоугольника;
25. Ползущая змейка (змейка из 4 квадратов обходит внутреннюю границу экрана и возвращается в исходную точку);
26. Движение двух окружностей вокруг центра по эллиптической орбите. Параметры эллипса задает пользователь.
27. Движение прямоугольника по диагонали;

# КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Для чего предназначена библиотека SDL?
2. Из каких этапов состоит создание простейшего приложения с использованием графики?
3. Из каких этапов состоит отображение простейшей анимации средствами SDL?

# Список рекомендуемой литературы

## Основная литература

1. Павловская Т.А. C#. Программирование на языке высокого уровня:учеб. Для вузов.-м. [и др.]:Питер,2014.-432
2. Давыдова Н.А.,боровская Е.В. Программирование: учеб. Пособие. М.:Бином. Лаб. Знаний,2011.-238 с.
3. Булатицкий Д.И. Объектно-ориентированное программирование: учеб. пособие.- Брянск: изд-во БГТУ,2008.-120 с.

## Дополнительная литература

1. Тюльпинова Н.В. Алгоритмизация и программирование: учеб. Пособие/Брян. Гос. Техн. Ун-т.- Брянск:изд-во БГТУ,2012.-199 с.-[+ электронная копия].
2. Иванова Г.С. Программирование: основы алгоритмизации и процедурное программирование, объектно ориентированное программирование: учеб. Для вузов.-2-е изд., стер.-м.:Кнорус, 2014.-425 с.

Простейшая анимация [Текст] + [Электронный ресурс]: методические указания к выполнению лабораторной работы № 21 для студентов очной формы обучения по направлениям подготовки 09.03.04 – «Программная инженерия», 09.03.01 «Информатика и вычислительная техника», 02.03.03 «Математическое обеспечение и администрирование информационных систем». – Брянск, 2020. – 7 с.

БУЛАТИЦКИЙ ДМИТРИЙ ИВАНОВИЧ

КОПТЕНОК ЕЛИЗАВЕТА ВИКТОРОВНА

Научный редактор В.Я. Израилев

Компьютерный набор Е.В. Коптенок

Иллюстрации Е.В. Коптенок

Подписано в печать 20.02.2017. Формат 60х84 1/15 Бумага офсетная. Офсетная печать. Усл.печ.л. \_\_\_ Уч.-изд.л. \_\_\_ Тираж 1 экз. Бесплатно

Брянский государственный технический университет

Кафедра «Информатика и программное обеспечение», тел. 56-09-84

241035, Брянск, бульвар 50 лет Октября, 7 БГТУ, тел. 58-82-49