

Подключение библиотеки OpenGL к Visual Studio

Андрюенко Иван Сергеевич

Приамурский государственный университет имени Шолом-Алейхема

студент

Вихляев Дмитрий Романович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

Для C++ написано множество различных библиотек для работы с графикой, данная статья содержит обзор на одну из популярных графических библиотек OpenGL. Исследование демонстрирует быстрое и простое подключение данной библиотеки, а также пример создания геометрической фигуры.

Ключевые слова: C ++, OpenGL, GLFW, Visual Studio.

Connecting the OpenGL library to Visual Studio

Andrienko Ivan Sergeevich

Sholom-Aleichem Priamursky State University

Student

Vikhlyaev Dmitry Romanovich

Sholom-Aleichem Priamursky State University

Student

Abstract

Many different libraries for working with graphics have been written for C++, this article contains an overview of one of the popular OpenGL graphics libraries. The study demonstrates the quick and easy connection of this library, as well as an example of creating a geometric shape.

Keywords: C ++, OpenGL, GLFW, Visual Studio.

1 Введение

1.1 Актуальность

OpenGL (Open Graphics Library) – спецификация, определяющая платформу-независимый программный интерфейс для написания приложений, использующих двумерную и трёхмерную компьютерную графику. Данная библиотека выделяется тем, что даёт хороший фундамент знаний о том, как создаются двухмерные игры, как создаётся физика только с помощью программного кода. Однако OpenGL абстрагируется от спецификаций операционных систем, что не предусматривает открытия окна

приложения для рисования. Это означает необходимость самостоятельного создания окна, определения контекста и обрабатывания пользовательского ввода. Существует много библиотек, которые обеспечивают необходимый нам функционал, одной из таких является библиотека GLFW. *Graphics Library FrameWork* – эта библиотека, специально предназначенная для работы с OpenGL. К тому же данная библиотека имеет предварительно скомпилированные бинарные и заголовочные файлы для Visual Studio 2010-2019, что освобождает от необходимости вручную компилировать библиотеку из исходного кода, вместо этого GLFW даёт нам возможность быстро и просто подключить все необходимые данные для работы с OpenGL.

1.2 Обзор исследований

Основной принцип построения графических объектов описала М.В. Смирнова [1]. Графическую среду разработки высокопроизводительных приложений исследовал И.Ю. Косов [2]. Рассмотрел графическую библиотеку SFML на C++ А. Е. Азаров [3]. Провёл исследования спецификации OpenGL Д.С. Ковтунов [4]. Функции и элементы OpenGL, используемые для построения основных форм в C# описали Ч.У. Хаятов, Г.И. Атаева, О.Р. Хайдаров [5]. Использование библиотек GLEW, GLFW и FREETYPE для разработки игры описали Д. С. Гурский, Н. Б. Осипенко [6]

1.3 Цель исследования

Целью данной статьи стоит рассмотреть возможность быстрого и простого подключения библиотеки GLFW к Visual Studio, и рассмотрение работы библиотеки OpenGL, на примере создания геометрической фигуры.

2 Материалы и методы

Для работы используем скомпилированный код библиотеки GLFW, который можно скачать с официального сайта, и подключим его к Visual Studio.

3 Результаты и обсуждения

Для начала необходимо зайти на официальный сайт GLFW в кладку DOWNLOAD и скачать скомпилированную библиотеку под операционную систему (рис. 1).

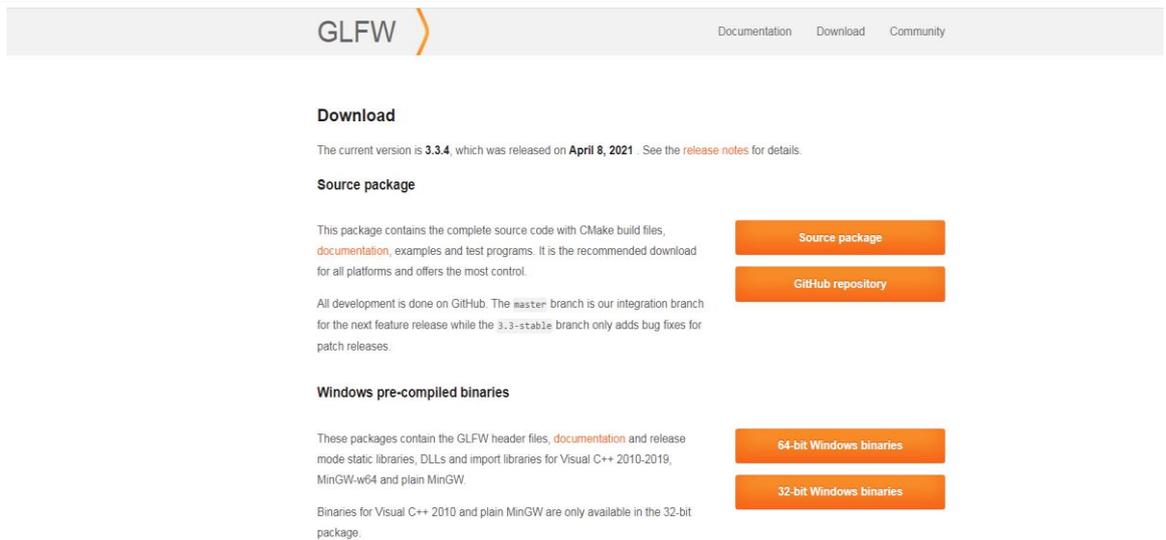


Рис. 1. Скачивание библиотеки GLFW с официального сайта.

Далее открываем пустой проект C++ в Visual Studio и создаём исходный файл main.cpp (рис. 2).

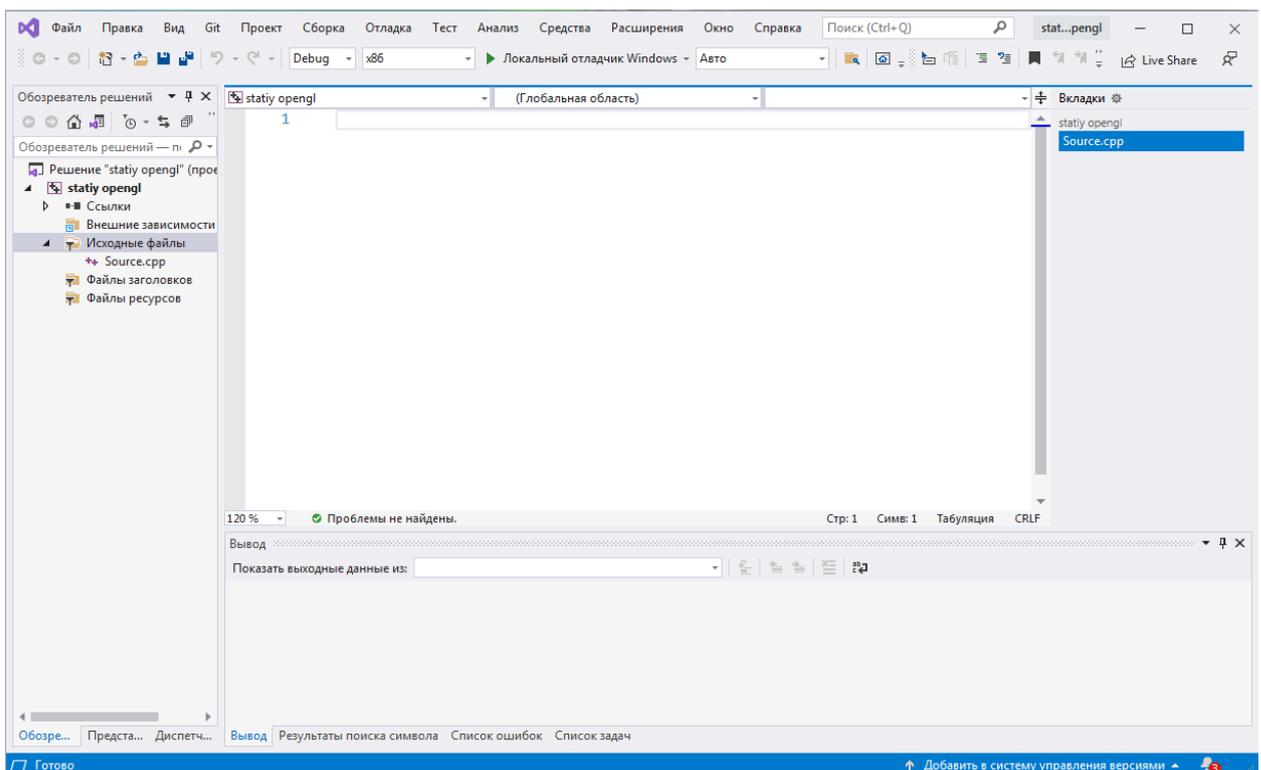


Рис. 2. Создание проекта в Visual Studio

Далее переходим в папку, где лежит наш проект, в ней создаём папку GLFW и перекидываем в неё файлы из скаченной библиотеки. Нам понадобится папка include и lib-vc2019 (рис.3).

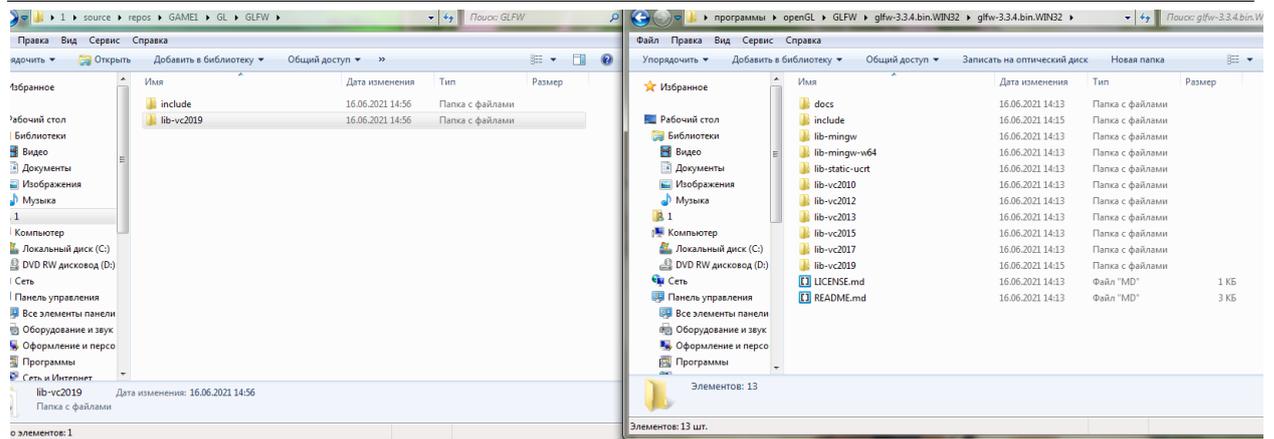


Рис. 3. Добавление библиотеки в репозиторий с проектом

Теперь заходим в Visual Studio → свойства проекта → C/C++ → Общие → Дополнительный каталог включаемых файлов → изменить в открывшемся окне прописываем путь к папке include, которую мы присоединили к нашему проекту (рис. 4).

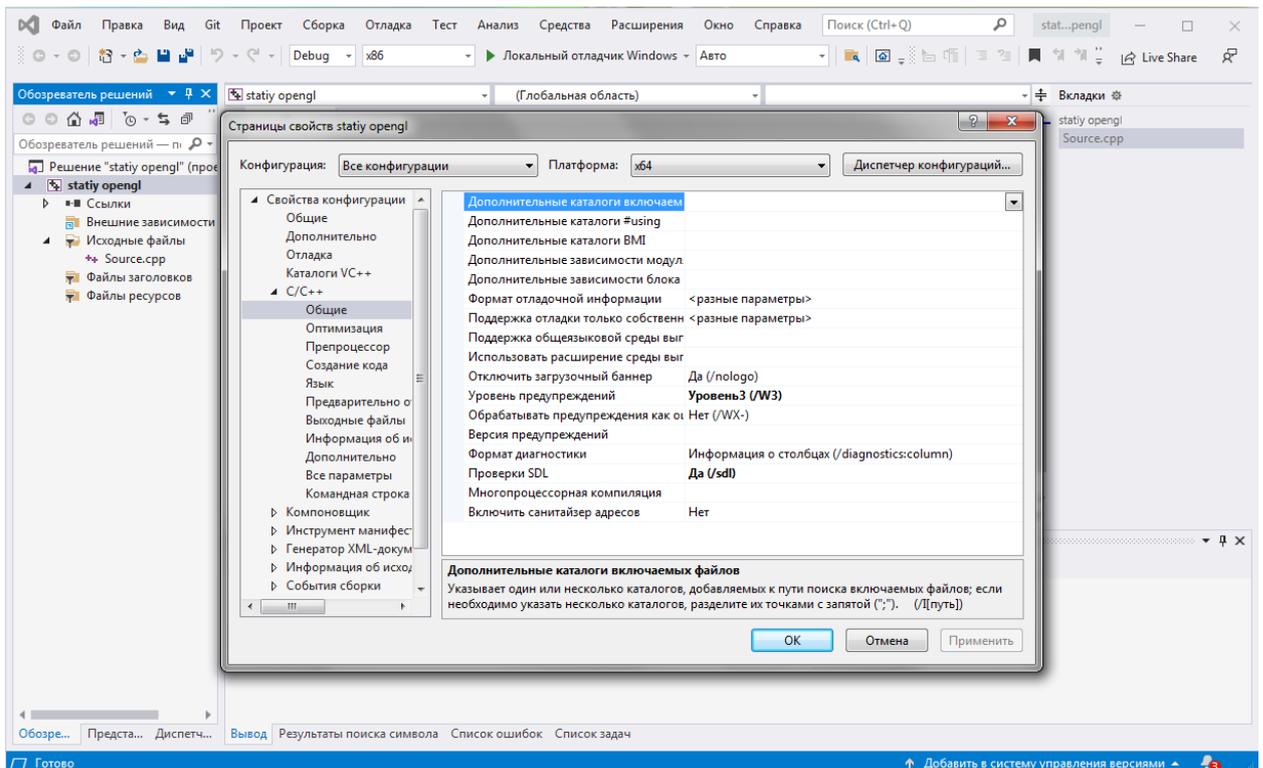


Рис. 4. Подключение библиотеки glfw3.h

Далее переходим во вкладку Компоновщик → Общие → Дополнительные каталоги библиотек → изменить. В открывшемся окне прописываем путь к папке lib-vc2019 (рис.5).

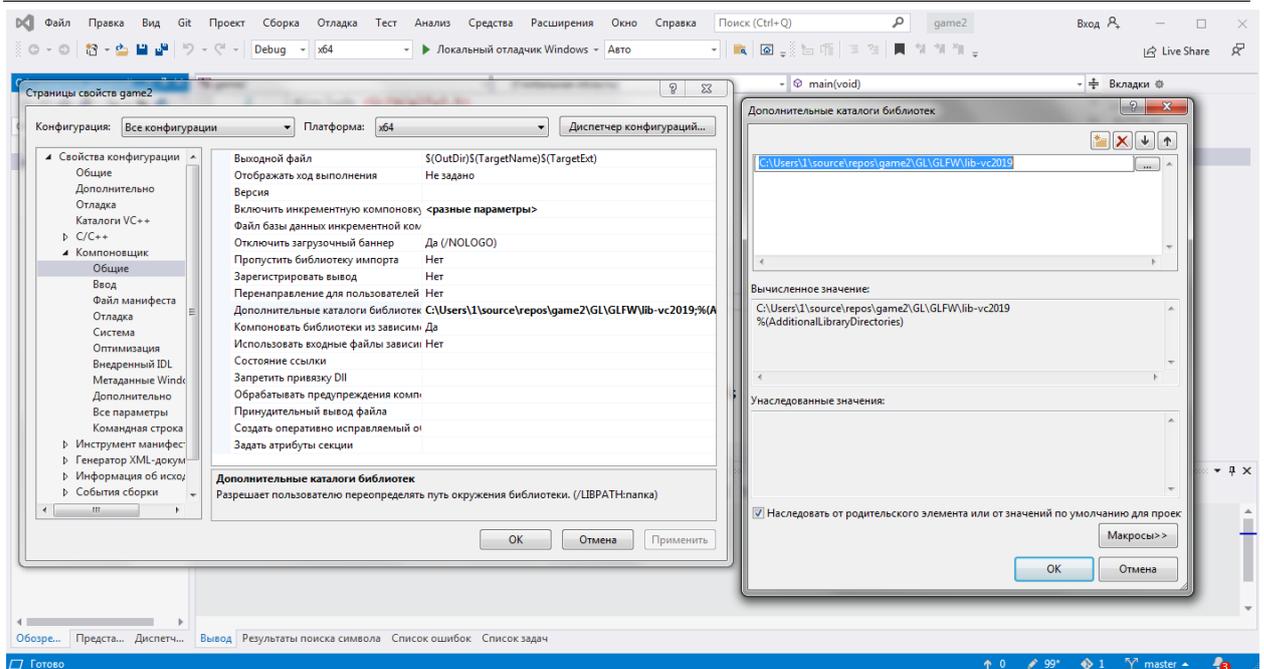


Рис. 5. Подключение библиотеки glfw3.lib

Не выходя из вкладки компоновщик, заходим в вкладку ввод → Дополнительные зависимости. В открывшемся окне записываем имена библиотек необходимых для полноценной работы проекта и открытия окна opengl32.lib и glfw3.lib, далее жмём ОК (рис.6).

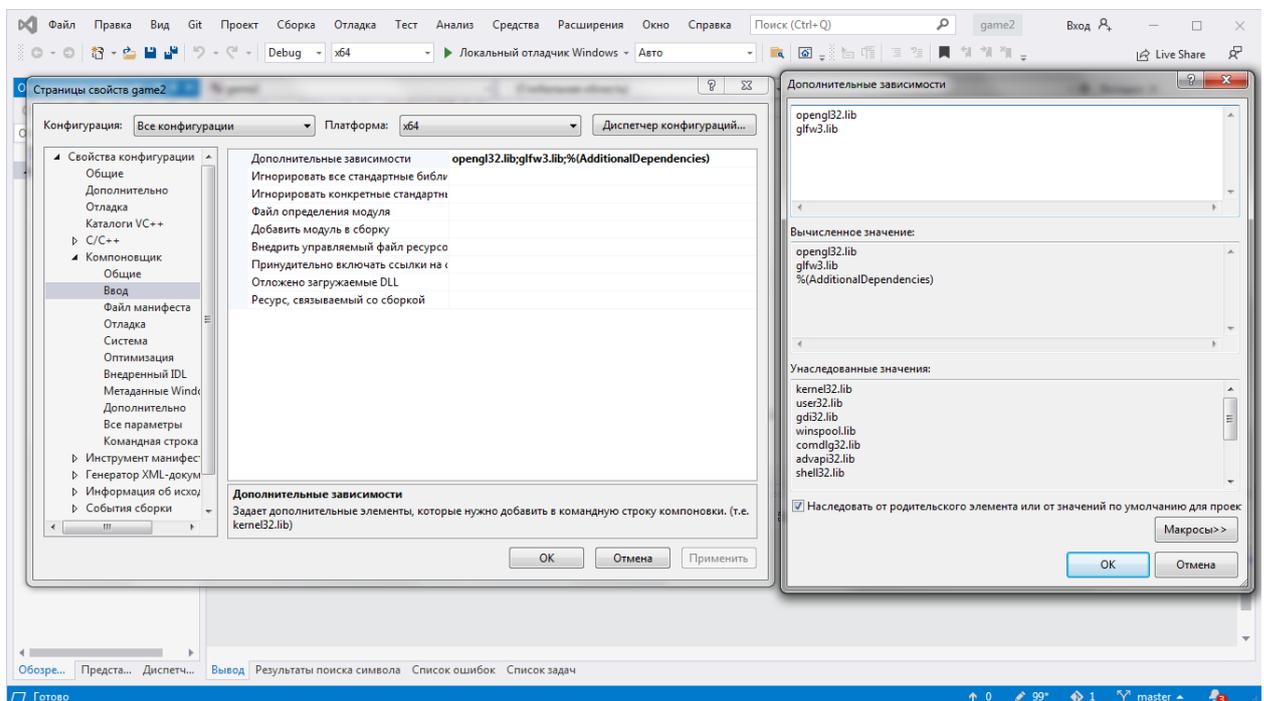


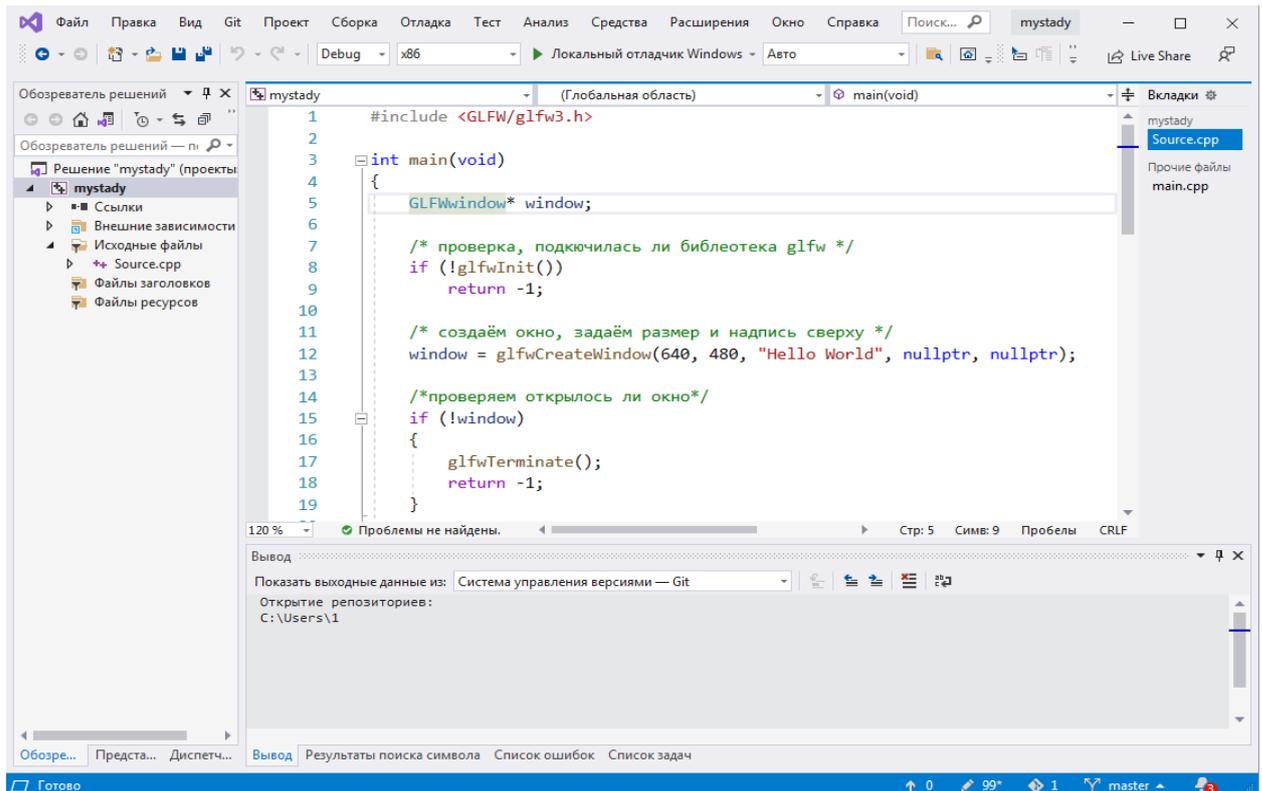
Рис. 6. Подключение библиотеки opengl32.lib и glfw3.lib

Для того чтобы проверить хорошо ли всё подключилось, нарисуем несколько геометрических фигур.

С помощью директивы `include` подключим библиотеку `GLFW/glfw3.h`. В функции `main` проверим, подключилась ли библиотека `glfw`, с помощью функции `glfwInit()`, в случае ошибки она выдаёт `false`.

Создаём переменную `window` и присваиваем ей значения окна с помощью функции `glfwCreateWindow()`. Первые два параметра функции – размеры окна в пикселях, затем название окна, остальные оставляем `NULL`.

Проверим, открылось ли окно, если окно не открылось, возвращаем ошибку и с помощью функции `glfwTerminate()`, уничтожаем все ресурсы, выделенные `glfw`. (рис. 7).



```
1 #include <GLFW/glfw3.h>
2
3
4 int main(void)
5 {
6     GLFWwindow* window;
7
8     /* проверка, подключилась ли библиотека glfw */
9     if (!glfwInit())
10         return -1;
11
12     /* создаём окно, задаём размер и надпись сверху */
13     window = glfwCreateWindow(640, 480, "Hello World", nullptr, nullptr);
14
15     /*проверяем открылось ли окно*/
16     if (!window)
17     {
18         glfwTerminate();
19         return -1;
20     }
21 }
```

Рис. 7. Создание окна

Для того чтобы окно не закрылось сразу после выполнения программы, используем цикл.

Все дальнейшие действия по обрисовке внутри окна будут использоваться внутри цикла.

Используем функцию `glfwSwapBuffers()`, которая меняет местами передний и задний буфер, в качестве параметра используем переменную `window`. А также функцию `glfwPollEvents()`, для обработки событий. В конце программы снова воспользуемся функцией `glfwTerminate()`, чтобы очистить данные. Запустим программу (рис. 8).

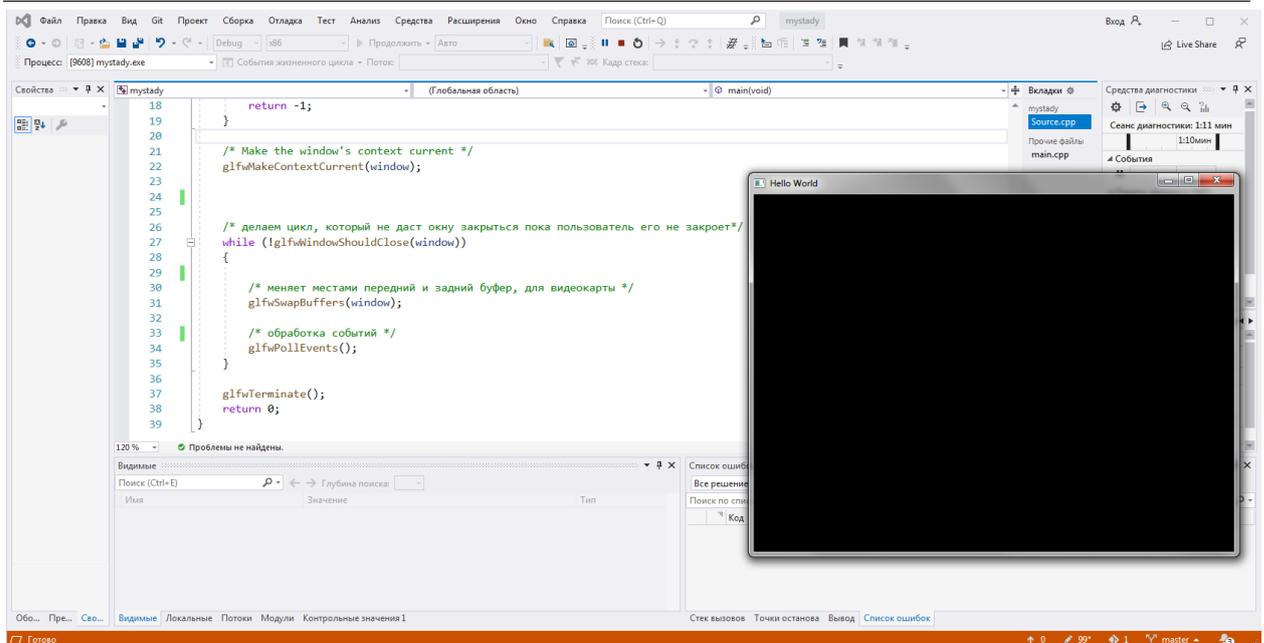


Рис. 8. Запуск программы

Запустилось окно. Для смены цвета окна, внутри цикла необходимо вызвать функцию `glColor()`, в неё передаются параметр, который указывает, как будет задаваться цвет.

В Функцию `glClearColor()` присваиваются параметры цвета в режиме `rgba`, её необходимо поставить перед функцией `glColor()`.

Создадим функцию, которая будет рисовать треугольник.

Процесс обрисовки в `OpenGL` начинается, когда вызывается функция `glBegin()`, и заканчивается при вызове `glEnd()`. В качестве параметров функции `glBegin()` мы присвоим `GL_TRIANGLES`, давая понять, что мы хотим нарисовать треугольник. Треугольник рисуется по точкам, чтобы задать точки используем функции `glColor3f()` и `glVertex2f()`, первая функция придаёт цвет точкам в режиме `rgb`, а вторая размещает точки по координатам на плоскости. Центр координат находится в центре окна, радиус координатной плоскости равен единице, т.е. всё окно имеет размер два с лева направо и снизу вверх, вне зависимости от размера экрана в пикселях. Нарисуем ещё три точки и вызовем функцию в главном цикле (рис. 9, 10).

```

4 void triangle()
5 {
6     glBegin(GL_TRIANGLES);
7     glColor3f(0, 0, 0); glVertex2f(-0.5, -0.3);
8     glColor3f(1, 0, 0); glVertex2f(0.5f, -0.3);
9     glColor3f(0, 0, 1); glVertex2f(0, 0.4f);
10
11     glEnd();
12 }

```

Рис. 9. Функция создания треугольника

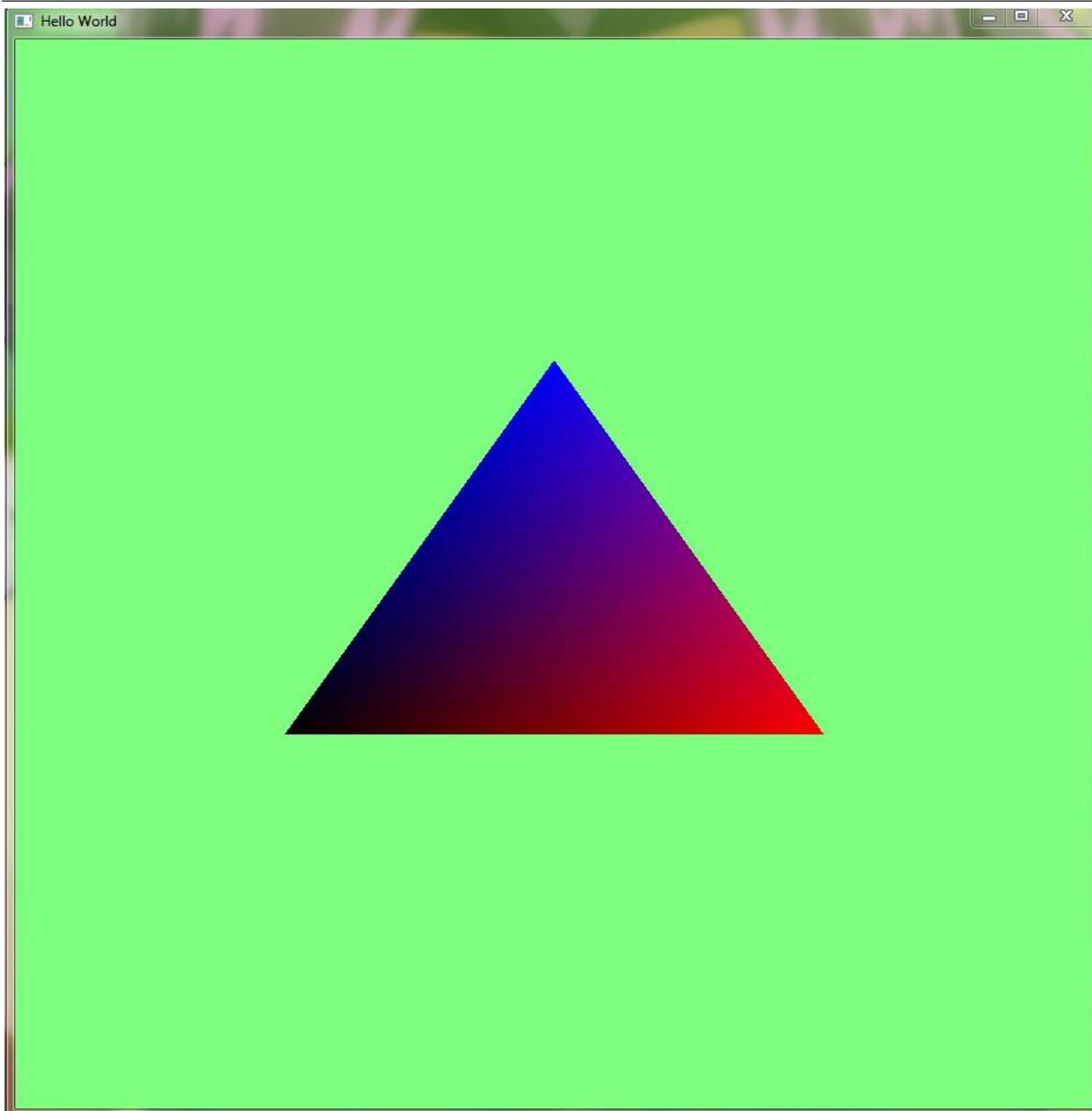


Рис. 10. Результат программы

Таким образом, мы подключили библиотеку OpenGL к Visual Studio, написали программу для открытия графического окна, задали фон и нарисовали геометрическую фигуру – треугольник.

Библиографический список

1. Смирнова М.В. Основной принцип построения графических объектов//Актуальные проблемы гуманитарных и естественных наук. 2016. № 2-2. С. 93-97.
2. Косов И.Ю. Графическая среда разработки высокопроизводительных приложений//Перспективные информационные технологии и интеллектуальные системы. 2006. № 4. С. 72-79.
3. Азаров А.Е. Обзор графической библиотеки SFML на C++ //Постулат.

2018 № 1 (27). С. 107.

4. Ковтунов Д.С. Обзор спецификации OpenGL //Вестник науки и образования. 2018. Т. 2. № 4 (40). С. 17-20.
5. Хаятов Ч.У., Атаева Г.И., Хайдаров О.Р. Функции и элементы OpenGL, используемые для построения основных форм в C# //Universum: технические науки. 2020. № 11-1 (80). С. 43-45.
6. Гурский Д. С., Осипенко Н. Б. Использование библиотек GLEW, GLFW и FREETYPE для разработки игры описали //Новые математические методы и компьютерные технологии в проектировании, производстве и научных исследованиях. Материалы XXII Республиканской научной конференции студентов и аспирантов. Гомель, 2019. С. 365-366.