



@HighMem

Пользователь

10 марта 2016 в 16:08

1.1 SFML и Visual Studio из песочницы tutorial

Разработка игр*, Visual Studio*, C++*



От переводчика: данная статья является первой в цикле переводов официального руководства по библиотеке SFML. Данный цикл статей ставит своей целью предоставить людям, не знающим язык оригинала, возможность ознакомиться с этой библиотекой. SFML — это простая и кроссплатформенная мультимедиа библиотека. SFML обеспечивает простой интерфейс для разработки игр и прочих мультимедийных приложений. Оригинальную статью можно найти [тут](#). Начнем.

Библиотека SFML предоставляет простой интерфейс для различных компонентов вашего компьютера, чтобы облегчить разработку игр и мультимедийных приложений. Она состоит из пяти модулей: system, window, graphics, audio и network.

Используя SFML, ваше приложение может быть скомпилировано и запущено на наиболее распространенных платформах: Windows, Linux, Mac X(планируется поддержка Android и IOS).

Предварительно скомпилированные SDK для вашей ОС доступны на [странице загрузки](#).

SFML официально поддерживает C и .NET. Также, благодаря своему активному сообществу, она доступна на многих других языках, таких как Ruby, Python, Go и др.

[Оглавление:](#)

Вступление

Эта статья — первая, которую вам следует прочитать, если вы используете среду разработки Visual Studio (Visual C++ compiler). В ней будет рассказано, как настроить ваш проект.

Установка SFML

Для начала вам необходимо скачать SFML SDK со [страницы загрузки](#).

Скаченный вами пакет должен соответствовать вашей версии Visual C++. Например, библиотека, скомпилированная с помощью VC++ 10 (Visual Studio 2010) не будет совместима с VC++ 12 (Visual Studio 2013). Если вы не найдете на странице загрузки пакет SFML, скомпилированный для вашей версии Visual C++, вам придется собрать SFML самостоятельно.

Далее вы должны распаковать архив с SFML в любую удобную для вас директорию. Копировать заголовочные файлы и библиотеки в вашу Visual Studio не рекомендуется. Лучше держать библиотеки в отдельном месте, особенно если вы намереваетесь использовать несколько версий одной библиотеки или несколько компиляторов.

Создание и конфигурирование проекта SFML

Первое, что вам необходимо сделать — это выбрать тип создаваемого проекта: вы должны выбрать «Win32 application». Мастер предложит вам несколько опций для настройки проекта: выберите «Console application» в том случае, если вам нужна консоль, либо «Windows application» в обратном случае. Выберите «Empty project», если вам не нужен автоматически сгенерированный код.

Создайте файл main.cpp и добавьте его в проект. Этим вы примените настройки C++ (в противном случае Visual Studio не будет знать, какой мы будем использовать для данного проекта). Содержимое файла main.cpp будет приведено ниже.

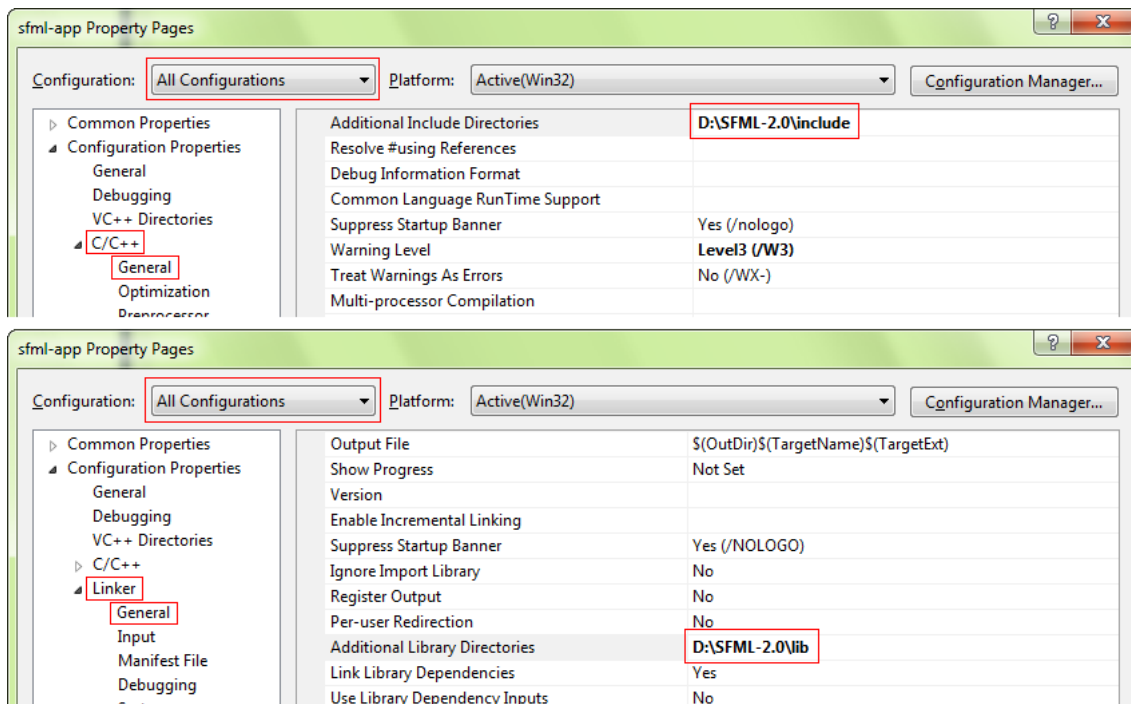
Теперь вам необходимо указать компилятору, где искать заголовочные файлы (файлы с расширением .hpp) и компоновщику, где искать библи

SFML (файлы с расширением .lib).

Добавьте в свойства проекта следующее:

- Путь до заголовочных файлов SFML (<путь-к-установке-SFML>/include) в C/C++ » General » Additional Include Directories
- Путь до библиотек SFML (<путь-к-установке-SFML>/lib) в Linker » General » Additional Library Directories

Эти пути совпадают для конфигураций Debug и Release, так что вы можете установить их глобально для вашего проекта («All configurations»)

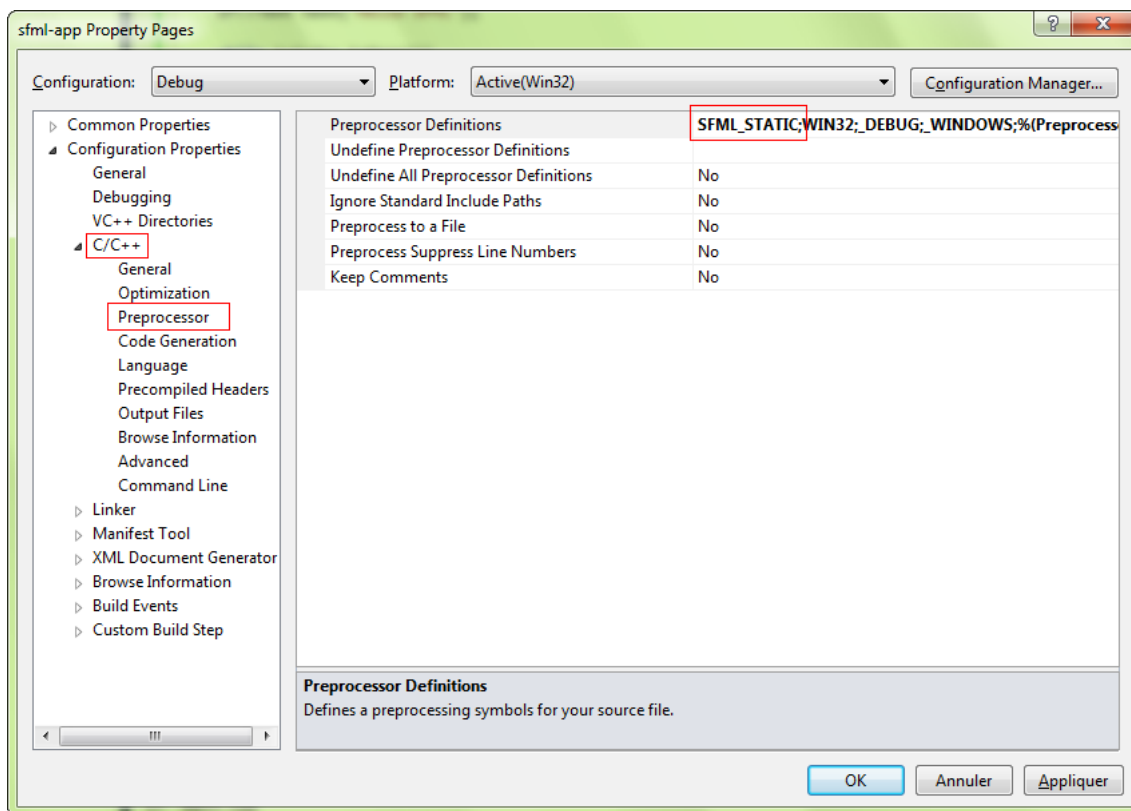


Следующий шаг — компоновка вашего приложения с библиотеками SFML (файлы с расширением .lib). SFML состоит из пяти модулей (system, window, graphics, network и audio) и библиотек для каждого из них. Библиотеки должны быть добавлены в свойства проекта в Linker » Input » Additional Dependencies. Добавьте те библиотеки SFML, в которых вы нуждаетесь, например «sfml-graphics.lib», «sfml-window.lib» и «sfml-system.lib».

Важно указать библиотеки, соответствующие конфигурации: «sfml-xxx-d.lib» для Debug и «sfml-xxx.lib» для Release, иначе могут возникнуть ошибки.

Настройки, приведенные выше, позволят вам скомпоновать ваш проект с динамической версией SFML, для которой требуются DLL файлы. Если хотите напрямую интегрировать SFML в ваш исполняемый файл, а не использовать компоновку с динамической библиотекой, вы должны скомпоновать статическую версию библиотеки. Статические библиотеки SFML имеют суффикс "-s": «sfml-xxx-s-d.lib» для конфигурации Debug и «sfml-xxx-s.lib» для Release.

Также вам необходимо определить макрос SFML_STATIC в опциях препроцессора вашего проекта.



Начиная с SFML 2.2 при статической компоновке вам также необходимо компоновать все зависимости SFML. Это означает, что если, к пример скомпонуете `sfml-window-s.lib` или `sfml-window-s-d.lib`, вам также придется скомпоновать `opengl32.lib`, `winmm.lib` и `gdi32.lib`. Некоторые из эти библиотек, возможно, уже перечислены в разделе «Inherited values», но добавление их не должно вызвать каких либо проблем.

В таблице ниже представлены зависимости для каждого модуля, добавьте `-d` если вы хотите скомпоновать Debug библиотеки SFML:

Модуль	Зависимости
<code>sfml-graphics-s.lib</code>	<ul style="list-style-type: none"> <code>sfml-window-s.lib</code> <code>sfml-system-s.lib</code> <code>opengl32.lib</code> <code>freetype.lib</code> <code>jpeg.lib</code>
<code>sfml-window-s.lib</code>	<ul style="list-style-type: none"> <code>sfml-system-s.lib</code> <code>opengl32.lib</code> <code>winmm.lib</code> <code>gdi32.lib</code>
<code>sfml-audio-s.lib</code>	<ul style="list-style-type: none"> <code>sfml-system-s.lib</code> <code>openal32.lib</code> <code>flac.lib</code> <code>vorbisenc.lib</code> <code>vorbisfile.lib</code> <code>vorbis.lib</code> <code>ogg.lib</code>
<code>sfml-network-s.lib</code>	<ul style="list-style-type: none"> <code>sfml-system-s.lib</code> <code>ws2_32.lib</code>
<code>sfml-system-s.lib</code>	<ul style="list-style-type: none"> <code>winmm.lib</code>

Из таблицы вы могли заметить, что одни модули SFML могут зависеть от других, например, `sfml-graphics-s.lib` зависит от `sfml-window-s.lib` и `sfml-system-s.lib`. Если вы производите статическую компоновку библиотеки SFML, будьте уверены, что все зависимости из цепочки зависимостей будут удовлетворены. Если одна из зависимостей отсутствует, вы получите ошибку компоновщика.

Если вы немного запутались, не волнуйтесь, для начинающего совершенно нормально быть перегруженным всей этой информацией о статическом связывании. Если у вас что-то не получится с первого раза, вы можете попробовать еще раз имея в виду все сказанное выше. Если у вас все же возникнут трудности со статическим связыванием, вы можете попробовать поискать решение в разделе [FAQ](#) или на [форуме](#).

Если вы не знаете чем отличаются динамические (также называемые общими) и статические библиотеки и какой тип библиотек использовать, можете найти больше информации в интернете. На данную тему есть множество хороших статей/блогов/постов.

Ваш проект готов, давайте напишем немного кода, что бы проверить, что все работает правильно. Поместите следующий код в файл `main.cpp`

```
#include <SFML/Graphics.hpp>

int main()
```

```

{
    sf::RenderWindow window(sf::VideoMode(200, 200), "SFML works!");
    sf::CircleShape shape(100.f);
    shape.setFillColor(sf::Color::Green);

    while (window.isOpen())
    {
        sf::Event event;
        while (window.pollEvent(event))
        {
            if (event.type == sf::Event::Closed)
                window.close();
        }

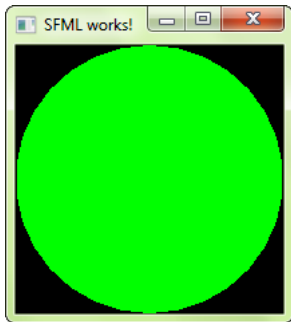
        window.clear();
        window.draw(shape);
        window.display();
    }

    return 0;
}

```

Если при создании проекта вы выбрали опцию «Windows application», то точкой входа в вашу программу должна быть функция «WinMain» в «main». Так как это специфика Windows, ваш код не будет компилироваться на Linux или Mac OS X. SFML предоставляет способ сохранить стандартной точкой входа «main» в том случае, если вы скомпилируете свой проект с модулем sfml-main («sfml-main-d.lib» для Debug, «sfml-main.lib» для Release) таким же способом, которым вы скомпилировали sfml-graphics, sfml-window и sfml-system.

Теперь скомпилируйте проект, и, если вы скомпилировали вашу программу с динамической версией SFML, не забудьте скопировать файлы с расширением .DLL (они расположены в <путь-к-установке-SFML>/bin) в директорию, в которой расположен исполняемый файл вашей программы. Запустите программу, и, если все было сделано правильно, вы должны увидеть это:



Если вы используете модуль sfml-audio (независимо от того, статическую или динамическую его версию), вы также должны скопировать внешнюю библиотеку OpenAL32.dll. Этот файл также может быть найден в директории <путь-к-установке-SFML>/bin.

Следующая статья: SFML и Code::Blocks (MinGW).

🔖 SFML, установка, перевод, документация, c++, visual studio



👁 13,7k ★ 73



@HighMem

карма рейтинг
4,0 0,0

ПОХОЖИЕ ПУБЛИКАЦИИ

23 марта 2016 в 11:30

Настройка редактора Visual Studio Code для разработки на PHP

↑ +14 👁 30,8k ★ 127 💬 16

17 марта 2016 в 10:42

Что есть в Visual Studio 2015 для разработчика на Python

↑ +9 👁 14k ★ 68 💬 7

2 декабря 2015 в 12:08

4 декабря смотрите Visual Studio Connect 2015 в России онлайн

САМОЕ ЧИТАЕМОЕРазра

- Сейчас
- Сутки
- Неделя
- Месяц

Как стать веб-разработчиком в 2017 году — план действий

↑ +29

👁 19k

★ 352

💬 56

Опасайтесь прозрачных пикселей

↑ +102

👁 13,2k

★ 88

💬 39

Трагедия стопроцентного покрытия кода

↑ +20

👁 2k

★ 11

💬 2

Твоя идея — ерунда

↑ +55

👁 16,4k

★ 53

💬 50

Борьба с перехватом HTTPS-трафика. Опыт Яндекс.Браузера


↑ +11

👁 1,9k

★ 12


💬 6

Комментарии (11)

-  VioletGiraffe

10 марта 2016 в 16:55

#


Неплохо было бы начать с того, что за библиотека SFML и для чего она может понадобиться.
-  HighMem

11 марта 2016 в 08:27

#

h

↑


Добавил вступление от переводчика. Там основная информация об SFML.
-  lexxpavlov

11 марта 2016 в 14:01

#

h

↑


Лучше добавлять информацию ДО хабраката, а то непонятно, что за библиотека.
-  HighMem

11 марта 2016 в 15:35 (комментарий был изменён)

#

h

↑

Поправил. Спасибо.
-  Nikobraz


16 марта 2016 в 05:21

#

h


↑

И всё-таки... Полистал статьи, вроде понял что представляет из себя эта библиотека. Но не понял каковы её возможности, преимущества и недосравнении с конкурентами.

Сделайте обзорную статью.
-  maaGames


10 марта 2016 в 16:59

#

Осталось записать статью про «Hello world!». Только не как обычно, а такую, как в книгах «C++ за 21 день».
-  Chaos_Optima

10 марта 2016 в 17:34

#

Не понял, это статья о том как прописать инклюд и либы в проекте? Скоро похоже и правда будут появляться статьи, как набирать текст на клавиатуре ((
-  HighMem


11 марта 2016 в 05:30 (комментарий был изменён)

#

h

↑

в начале написано, что эта статья-перевод официальной документации по SFML. Первые разделы документации-установка и конфигурирование SFML (G started если точнее).




stepanp

11 марта 2016 в 15:44

h ↑

Не будет таких статей. будут видеоуроки



iroln

11 марта 2016 в 17:23

#

Это же CMake-based проект? Так давайте использовать возможности CMake!


Разве нельзя сделать как-то вот так?

```
project(MyProject)

find_package(SFML COMPONENTS system window graphics REQUIRED)
include_directories(${SFML_INCLUDE_DIRS})

add_executable(MyProgram main.cpp)
target_link_libraries(MyProgram ${SFML_LIBRARIES})
```

Разработчики не позаботились об этом? Они предлагают настраивать проект, использующий их библиотеку, вручную? Seriously?



HighMem

11 марта 2016 в 18:21

h ↑

Я с вами полностью согласен, однако в официальной документации рассматривается лишь настройка проектов под определенными IDE.

Только зарегистрированные пользователи могут оставлять комментарии. Войдите, пожалуйста.

ИНТЕРЕСНЫЕ ПУБЛИКАЦИИ

Какой map быстрее, и есть ли альтернатива Judy

👁 284 ⭐ 2 💬 6

Реверс-инжиниринг «Казаков», часть последняя: второе дыхание

👁 841 ⭐ 10 💬 2

«Дальше не придумали, импровизируй» или Agile в информационной безопасности



👁 337 ⭐ 5 💬 1

Использование модулей C++ в Visual Studio 2017

👁 522 ⭐ 8 💬 2

Трагедия стопроцентного покрытия кода

👁 1,9k ⭐ 11 💬 1

Аккаунт	Разделы	Информация	Услуги	Приложения
Войти	Публикации	О сайте	Реклама	<div><div> Загрузите в App Store</div><div> доступно на Google</div></div>
Регистрация	Хабы	Правила	Тарифы	
	Компании	Помощь	Контент	
	Пользователи	Соглашение	Семинары	
	Песочница	Помощь стартапам		
	© 2006 – 2017 «ТМ»	Служба поддержки	Мобильная версия	