

SFML - Uvod

Objektno programiranje - 3. vježbe (1. dio)

Marko Živković

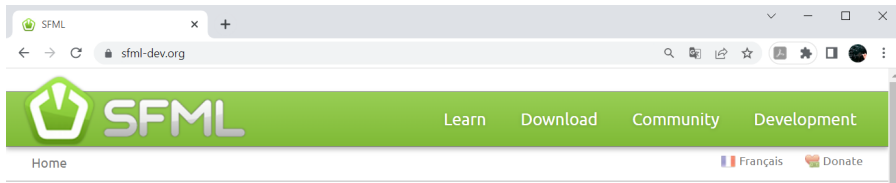
Prirodoslovno-matematički fakultet,
Sveučilište u Zagrebu

20. ožujka 2026. godine

SFML (*Simple and Fast Multimedia Library*)

- biblioteka za razvoj softvera
- daje jednostavno sučelje za programiranje aplikacija (API) koje se oslanjaju na multimedijski sadržaj (video, slika, zvuk i dr.)
- prva verzija (v1.0) objavljena 9. kolovoza 2007. godine

Web-stranica: www.sfml-dev.org



SFML je višejezičan i višeplatformski

- iako pisan u jeziku **C++**, može se koristiti i u jezicima: Ada, C, Crystal, D, Euphoria, Go, Java, Julia, .NET, Nim, OCaml, Python, Ruby, Rust i drugi (popis se može vidjeti [ovdje](#))
- može se koristiti u aplikacijama za većinu operacijskih sustava (**Windows**, Linux, macOS - u eksperimentalnoj fazi su i za Android i iOS)



Video igre u kojima je korišten SFML

- [Atom Zombie Smasher](#)
- [Crea](#)
- [Extreme Tux Racer](#)
- [HolySpirit](#)
- [I Can Transform](#)
- [KeeperRL](#)
- [Kroniax](#)
- [M.A.R.S.](#)
- [Open Hexagon](#)
- [Postmortem: one must die](#)
- [Project Black Sun](#)
- [The Duke](#)
- [The Shooting of Isaac](#)

Još primjera možete pronaći na sljedećoj web-stranici:

sfmlprojects.org/games

Pet modula od kojih se sastoji SFML

- nije ih nužno uključiti sve: uključujemo ih po potrebi
- **system**
 - osnovne strukture podataka, omogućuje pristup dretvama, satovima i dr.
- **window**
 - omogućuje upravljanje prozorom, sakupljanjem događaja i ulaznih podataka od korisnika
- **graphics**
 - sve što se tiče dvodimenzionalnog [renderiranja](#) (primjeri: [How do games render their scenes?](#))
- **audio**
 - sve vezano uz reproduciranje glazbe, zvukova, audio *streamova*, snimanje zvuka
- **network**
 - za prijenos podataka između računala i rad s nekoliko mrežnih protokola

Svaki od modula kompajliran u zasebnoj datoteci (.lib)

Posebni postfiksi označavaju je li biblioteka:

- linkana statički (**-s**) ili dinamički,
- izgrađena (*built*) u *debug* (**-d**) ili *release* načinu.

 sfml-system.lib

 sfml-system-d.lib

 sfml-system-s.lib

 sfml-system-s-d.lib

Linkanje:

- statičko - biblioteka uključena u izvršivu datoteku
- **dinamičko** - za izvršavanje aplikacije potrebno prisustvo **.dll** datoteka

 sfml-system-2.dll

 sfml-system-d-2.dll

 sfml-window-2.dll

 sfml-window-d-2.dll

Prednosti dinamičkog linkanja: ☒ manja veličina aplikacije
☒ moguća nadogradnja biblioteke bez potrebe mijenjanja aplikacije

debug vs. release

debug

- dodatne informacije korisne pri uklanjanju grešaka
- koristi se samo za testiranje (sporije izvršavanje aplikacije)

release

- hrpa različitih optimizacija $\Rightarrow$ manji (i brži) izvršivi kod

Zavisnosti

- svaka SFML aplikacija zahtijeva da je biblioteka `system` linkana
- različite biblioteke ipak ovise jedna o drugoj

Biblioteka	Biblioteke koje zahtijeva
<code>system</code>	-
<code>window</code>	<code>system</code>
<code>graphics</code>	<code>system</code> , <code>window</code>
<code>audio</code>	<code>system</code>
<code>network</code>	<code>system</code>

Primjer. Ako aplikacija nešto crta na ekran, treba linkati `system`, `window` i `graphics`.

Primjer. Uključivanje zaglavlja *Graphics*:

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

Umjesto cijelog modula, moguće je uključiti samo stvarno potrebno zaglavlje: `#include <SFML/Graphics/Color.hpp>`

Napomena: Sadržaj datoteke SFML/Graphics.hpp

- prikazan je dio sadržaja datoteke Graphics.hpp iz mape SFML (nalazi se u mapi SFML-2.6.1/include)

```
#ifndef SFML_GRAPHICS_HPP
#define SFML_GRAPHICS_HPP

#include <SFML/Window.hpp>
#include <SFML/Graphics/BlendMode.hpp>
#include <SFML/Graphics/CircleShape.hpp>
#include <SFML/Graphics/Color.hpp>
#include <SFML/Graphics/ConvexShape.hpp>
#include <SFML/Graphics/Drawable.hpp>
#include <SFML/Graphics/Font.hpp>
... da ne navodimo sad sve 😊 ...
#include <SFML/Graphics/View.hpp>

#endif
```

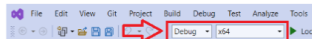
SMFL - preuzimanje i pokretanje

- pogledati šalabahter sa službene stranice kolegija

SFML - stvaranje novog projekta (Visual Studio, Praktikum)

(1.) Stvaranje novog projekta u Visual Studiu

- Pokrenuti **Visual Studio** (trenutno instaliran 2019)
- *Create a new project* → *Empty Project* (mora biti označen jezik C++)
- Upisati naziv projekta i odabrati lokaciju spremanja (zapamtiti tu lokaciju).
- *Solution Configurations* je **Debug** i *Solution Platforms* je **x64** - slika:

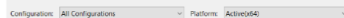


(2.) Preuzimanje SFML-a

- sa www.sfml-dev.org/download/sfml/2.6.1 preuzeti **Visual C++17(2022)-64-bit**
- raspakirati preuzetu .zip mapu i dobivenu mapu "SFML-2.6.1" spremiti (zapamtiti adresu) - npr. u disk C tako da je adresa C:\SFML-2.6.1
- u toj mapi "SFML-2.6.1" nalazi se mapa "bin" - datoteke iz nje iskopirati u mapu gdje je naš projekt (tamo gdje se nalazi .vcxproj datoteka)

(3.) Postavke VS projekta

[Nakon] dodavanja/stvaranja nove .cpp datoteke (desni klik u *Solution Explorer* prozoru na *Source Files* i odabir nove/postojeće datoteke u izborniku *Add*) u projektu:
[*Project* → *Properties*] (ili *Ime_Projekta Properties*) - u prozoru koji se otvori treba biti označeno kao ovdje:



Potrebne postavke:

- *C/C++* → *General* u polje *Additional Include Directories* upisati [gdje_je_mapa_SFML-a]\SFML-2.6.1\include - npr. C:\SFML-2.6.1\include
- *Linker* → *General* u polje *Additional Library Directories* upisati [gdje_je_mapa_SFML-a]\SFML-2.6.1\lib - npr. C:\SFML-2.6.1\lib
- *Linker* → *Input* u polje *Additional Dependencies* dodati (desno se javi strelica dolje pa odabrati *Edit*) sve potrebne module - primjerice, najčešće ćemo trebati sljedeće:

```
sfml-system-d.lib  
sfml-window-d.lib  
sfml-graphics-d.lib
```

- koristit ćemo verziju **2.6.1**

Kako dobiti statičku verziju (VS2022)

- omogućuje da se „riješimo” .dll datoteka
- treba definirati `SFML_STATIC` makro u opcijama preprocesora (u opcijama projekta)

PrviProjekt Property Pages

Configuration: All Configurations Platform: Active(x64)

Configuration Properties

- General
- Advanced
- Debugging
- VC++ Directories
- C/C++**
 - General
 - Optimization
 - Preprocessor**
 - Code Generation

Preprocessor Definitions **SFML_STATIC; <different options>**

Undefine Preprocessor Definitions	
Undefine All Preprocessor Definitions	No
Ignore Standard Include Paths	No
Preprocess to a File	No
Preprocess Suppress Line Numbers	No
Keep Comments	No
Use Standard Conforming Preprocessor	

Kako dobiti statičku verziju (VS2022) (nastavak)

- od verzije SFML 2.2, treba linkati sve `.lib` o kojima pojedini modul zavisi (uočite sufiks `-s` za module)

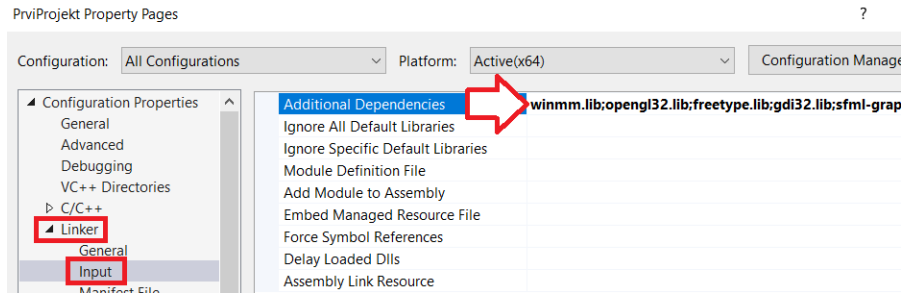
Modul	Zavisnosti
sfml-graphics- <code>s</code> .lib	sfml-window- <code>s</code> .lib sfml-system- <code>s</code> .lib opengl32.lib freetype.lib
sfml-window- <code>s</code> .lib	sfml-system- <code>s</code> .lib winmm.lib opengl32.lib gdi32.lib
sfml-audio- <code>s</code> .lib	sfml-system- <code>s</code> .lib flac.lib vorbisfile.lib ogg.lib openal32.lib vorbisenc.lib vorbis.lib
sfml-network- <code>s</code> .lib	sfml-system- <code>s</code> .lib ws2_32.lib
sfml-system- <code>s</code> .lib	winmm.lib

Napomena. Audio modul svejedno zahtijeva kopiranje datoteke `openal32.dll` iz `<gdje-je-sfml/bin>`, neovisno o tome želimo li statičko ili dinamičko linkanje!

Kako dobiti statičku verziju (VS2022) (nastavak)

Primjer. Ako želimo statičko linkanje (i *debug* način - zato sufiks **-d**), a trebamo module `System`, `Window` i `Graphics`, prema prethodnom slajdu treba u postavkama projekta postaviti linkanje sljedećih datoteka:

```
winmm.lib; opengl32.lib; freetype.lib; gdi32.lib;  
sfml-graphics-s-d.lib; sfml-window-s-d.lib;  
sfml-system-s-d.lib
```



Kako dobiti statičku verziju (VS2022) (nastavak)

- sad nema potrebe za .dll-ovima:

