

מכללת הדסה, החוג למדעי המחשב

מבוא לתכנות מונחה עצמים והנדסת תוכנה

סמסטר א', תשפ"ד

הוראות התקנת ספריית SFML והשימוש בה

הגרסה העדכנית של הספרייה היא גרסת 2.6.0 ואיתה נעבוד. חשוב להקפיד להתקין את הספרייה על פי ההוראות כאן כדי לוודא שההגשה שלכם תתקמפל גם אצל הבודק.

תוכן העניינים

[שלבי ההתקנה](#)

[שימוש בספרייה](#)

[התבססות על קובצי הדוגמה](#)

[שינויים בקובץ ה-CMakeLists.txt](#)

[הסבר](#)

[שימוש באודיו או ברשת](#)

[כתיבת הקוד](#)

[הרצה ללא טרמינל](#)

[דוגמה](#)

[למידע נוסף](#)

שלבי ההתקנה

1. נוריד מאתר SFML, מהדף <https://www.sfml-dev.org/download/sfml/2.6.0>, את הקובץ המיועד ל-64-bit - Visual C++ 17 (2022). הקישור הישיר לקובץ הוא: [64bit](#).

2. את הקובץ שהורדנו, נחליץ לתוך תיקייה בשם C:\SFML, כלומר אחרי שהחילוץ יסתיים, אנחנו מצפים לקבל תיקייה בשם SFML בכוון C:, בתוכה תיקייה בשם SFML-2.6.0, ובתוכה מספר תיקיות:

← → ▼ ▲			📁 > This PC > Local Disk (C:) > SFML > SFML-2.6.0		
^					
<input type="checkbox"/>	Name		Date modified	Type	
📁	bin		8/9/2023 12:04 AM	File folder	
📁	doc		8/9/2023 12:04 AM	File folder	
📁	examples		8/9/2023 12:04 AM	File folder	
📁	include		8/9/2023 12:04 AM	File folder	
📁	lib		8/9/2023 12:04 AM	File folder	
📄	license.md		8/9/2023 12:04 AM	Markdown Source...	
📄	readme.md		8/9/2023 12:04 AM	Markdown Source...	

○ שימו לב לוודא שזה המצב שהתקבל, ולא מצב שבו תיקיית SFML-2.6.0 מופיעה ישירות תחת C:\, למשל.

3. זה הכול. הנקודה החשובה פה היא לחלץ את הקבצים למקום הנכון, כדי שההפניות שנעשה מהפרויקט שלנו לשם תעבודנה באופן זהה במחשב שלנו, במחשבי המעבדות במכללה וכמובן (הכי חשוב :)) במחשב של הבודק.

שימוש בספרייה

התבססות על קובצי הדוגמה

כרגיל, גם כאן מומלץ להשתמש בקובצי הדוגמה מהתרגול כבסיס לתרגיל שלכם. בפרט, מאוד לא מומלץ לנסות להעתיק בהעתק-הדבק את הפקודות מקובץ ה-PDF הזה, בגלל תווים נסתרים שנוספים בהעתקה כזו. להלן נסביר את השינויים מול הקבצים הרגילים רק כדי להבין את המשמעות. בנוסף, בסוף הסעיף מוסברים שינויים נוספים הנדרשים לשימוש באודיו או ברשת.

שינויים בקובץ ה-CMakeLists.txt

כדי להשתמש בספרייה, הוספנו לקובץ ה-CMakeLists.txt שלנו את השורות הבאות (שימו לב שהסדר שלהן חשוב!):

```
set(SFML_LOCATION "C:/SFML/SFML-2.6.0")
set(SFML_DIR "${SFML_LOCATION}/lib/cmake/SFML")

find_package(SFML 2.5 COMPONENTS graphics REQUIRED)
```

```
set_target_properties(sfml-main sfml-system sfml-window sfml-graphics sfml-audio  
sfml-network PROPERTIES MAP_IMPORTED_CONFIG_RELWITHDEBINFO RELEASE)
```

```
target_link_libraries(${CMAKE_PROJECT_NAME} sfml-graphics sfml-main)
```

```
include(cmake/SFML.cmake)
```

הסבר

הפקודה הראשונה מגדירה את המיקום של ספריית SFML, כך שיהיה יותר נוח לשנות אותה בעת הצורך. אנחנו מפנים אותה לנתיב שאליו חילצנו קודם את הקבצים. שימו לב לשימוש בקו נטוי קדמי (forward slash) במקום קו נטוי אחורי (backslash) המקובל בדרך כלל ב-Windows.

הפקודה השנייה גורמת לפקודת find_package שבשורה הבאה למצוא את קובצי ה-CMake של SFML שנמצאים כחלק מההתקנה.

הפקודה השלישית, find_package, גורמת ל-CMake למצוא בפועל את ספריית SFML ולאפשר את השימוש בה. (ברשימת הרכיבים (COMPONENTS) ניתן להוסיף לפי הצורך גם את audio ו-network, בנוסף ל-graphics שתמיד נצטרך. כרגע אין לנו בהן צורך.)

הפקודה הרביעית (set_target_properties, שמתפרשת כאן על פני שתי שורות) מאפשרת את הקומפילציה כראוי גם לקונפיגורציית Release.

הפקודה החמישית, target_link_libraries, מורה ל-CMake להשתמש בספריית SFML עבור התוכנית שלנו. היא מחברת אותה לקובץ ההרצה שלנו (שהוגדר בפקודת add_executable להתאים לשם של הפרויקט שלנו). גם פה, נוסיף לרשימת הספריות את sfml-audio ו-sfml-network אם נרצה להשתמש בהן. נשים לב שעבור sfml-audio חובה להוסיף את המילה audio גם לרשימת הרכיבים בפקודת find_package (הפקודה השלישית, כפי שמוזכר לעיל).

לסיום, הפקודה השישית מפעילה קובץ CMake נוסף שצירפנו לקבצים לדוגמה, שמכיל פקודות להעתקה אוטומטית של מספר קבצים (DLL-ים) של SFML אל תיקיית התוכנית שלנו, כדי לאפשר הפעלה תקינה של התוכנית.

שימוש באודיו או ברשת

כמו שהוסבר לעיל, הפקודות השלישית והחמישית הן אלה שנצטרך לשנות כשנרצה להוסיף שימוש ברכיבים נוספים של SFML. בפקודת find_package להוסיף את המילה audio ובפקודת target_link_libraries להוסיף את sfml-audio או sfml-network לפי הצורך.

כתיבת הקוד

כעת נוכל לחזור ולכתוב את הקוד שלנו תוך שימוש ב-SFML, ונוכל להוסיף את ה-include-ים הנדרשים, למשל:
#include <SFML/Graphics.hpp>

הרצה ללא טרמינל

כשהתוכנית עובדת כראוי, ואין לנו עוד צורך בטרמינל (שבו אולי השתמשנו להדפסות לצורך debugging), ניתן להוסיף בפקודת `add_executable` אחרי שם קובץ ההרצה את המילה `WIN32` כדי לקבל קובץ הרצה שאינו פותח טרמינל אלא מיד פותח את החלון של SFML שהגדרנו בקוד. לדוגמה, השורה:

```
add_executable (${CMAKE_PROJECT_NAME})
```

תהפוך להיות:

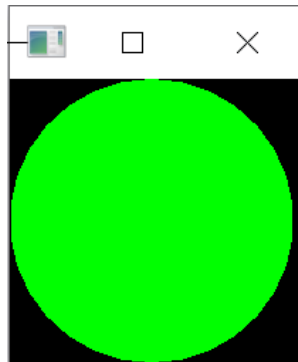
```
add_executable (${CMAKE_PROJECT_NAME} WIN32)
```

דוגמה

בעמוד הקבצים מהתרגול מצורף קובץ `SFMLTest.zip` המכיל דוגמה קצרה לשימוש ב-SFML. (הדוגמה מבוססת על הדוגמה שבקובצי ההדרכה של הספרייה, בכתובת

<https://www.sfml-dev.org/tutorials/2.6/#getting-started>).

הורידו את קובץ ה-`zip`, פתחו אותו למקום לבחירתכם, פתחו את הפרויקט כרגיל (בעזרת `Open a local folder`), כמוסבר בקובץ ההנחיות לשימוש ב-VS) ונסו לקמפל ולהריץ אותו. אם הכול מתנהל כשורה, יופיע חלון ובו עיגול ירוק גדול:



למידע נוסף

Official tutorials – <https://www.sfml-dev.org/tutorials/2.6/>

Online API documentation – <https://www.sfml-dev.org/documentation/2.6.0/>

Community wiki – <https://github.com/SFML/SFML/wiki>