

## Komme i gang med SFML på Visual Studio Express 2013

### Opprett en solution og et prosjekt i Visual studio.

- Eksempelvis kan du kalle solution for "Oving 8" og prosjektet for "Minesveiper".
- Kopier kildekodefilene utlevert sammen med øvingen over i Minesveiper prosjektkatalogen (eller hva du nå har kalt prosjektet ditt).
- Legg kildekodefilene til i prosjektet ditt ved å høyreklikke på prosjektet og velge Add -> Existing Items

### Last ned SFML fra <http://www.sfml-dev.org>

- Gå til download-siden og velg en versjon som stemmer med din Visual Studio
- Har du Visual Studio Express 2013 velger du SFML versjonen Visual C++ 12 (2013)
- Velg 32-bits utgaven av SFML (men de fleste kan også trygt velge 64-bits)

### Pakk ut SFML og lagre på et egnet sted

- Et godt råd er å legge SFML i prosjektkatalogen du laget for Minesveiper.
- ...men det viktigste er at du klarer å finne igjen SFML-katalogen siden du skal konfigurere prosjektet ditt med stien til noen SFML-underkataloger

### Legg til SFML include-stien i prosjektet ditt

- For at kompilatoren skal finne header filene til SFML må prosjektet konfigureres til å lete i riktig katalog.
- Høyreklikk på prosjektet og velg Properties.
- Det er mulig å konfigurere forskjellig for debug og release utgaven av programmet, men du kan basere deg på "All Configurations" i høyre venstre dropdown-meny.
- Velg Configuration Properties->C/C++->General->Additional Include Directories og legg til stien "C:\fulle\stien\til\SFML-2.2\include"

### Legg til SFML lib-stien i prosjektet ditt

- Velg Configuration Properties->Linker->General->Additional Library Directories og legg til stien C:\fulle\stien\til\SFML-2.2\lib"

### Legge til bibliotekene du trenger

- Velg Configuration Properties->Linker->Input->Additional Dependencies
- I SFML/lib katalogen finnes det to sett av biblioteker. Et sett som er compilert for debugging (har navn som slutter med "-d.lib" og et sett som ikke er compilert uten det ekstra som kreves for å kunne kjøre trinnvis gjennom koden i biblioteket.
- Du bør liste opp debugging-bibliotekene under debug og de som ikke er debug under release.
- List opp biblioteksnavnene du bruker (en linje for hvert bibl.). Det skader ikke å ta med alle. Under release lister du sfml-graphics.lib, sfml-window.lib, sfml-system.lib, sfml-audio.lib, sfml-network.lib. Under debug lister du sfml-graphics-d.lib, sfml-window-d.lib, sfml-system-d.lib, sfml-audio-d.lib, sfml-network-d.lib
- PS! Egentlig holder det å liste for eksempel bare release-bibliotekene, men da kan du bare kjøre starte release-versjonene (velges i verktøy-linja på VS).

### Legg til SFML bin-stien i prosjektet ditt

- Programmet som kompileres er avhengig av å finne dll-filene (bibliotek som lastes runtime). Det er fullt mulig å kompilere programmet hvis du har utført trinnene over, men du får ikke kjørt programmet hvis ikke programmet finner dll-filene.
- Legg til følgende i Configuration Properties->Debugging->Environment:  
PATH=C:\fulle\stien\til\SFML-2.2\bin;\$(PATH)
- Da finner programmet ditt bibliotekene når det kjøres med debugging (F5)
- Et annet alternativ er å kopiere dll-filene til samme katalog som den kjørbare fila i prosjektet ditt.

### Nå kan programmet kjøres fra Visual Studio

- Men bruk den grønne knappen for å kjøre programmet, alternativt menyvalget "Debug->Start with debug", eller rett og slett bare trykk F5.

### Hvorfor har vi både lib og dll biblioteker?

- dll-filer (dynamic link libraries) er compilert bibliotekskode som programmet kan linke til dynamisk mens det kjører. Fordelen er at samme dll-fil kan brukes av mange programmer. Alternativet er å statisk lenke sammen bibliotek og ditt program, men da får du store kjørbare filer og mye duplisering av kode mellom programmene.
- lib-filene er objektkode som blir inkludert i programmet ditt ved kompilering. Denne koden er nødvendig for at programmet ditt skal finne og kunne bruke funksjonene i dll-fila.