

# TDP005: Introduktion till SDL

Jonas Lindgren                      Torbjörn Lönnemark  
`jonas.lindgren@liu.se`      `torbjorn.lonnemark@liu.se`

Niklas Hayer

2014-11-16

## 1 Förberedande

Börja med att öppna upp en webbläsare med dokumentationen till SDL från <https://wiki.libsdl.org/CategoryAPI>. Den kommer att behövas under laborationens gång.

Ladda även ner den givna koden för denna labb från kurshemsidan. Kompilera den med make för att se till att det fungerar, och installera vid behov SDL2.

Tips: Detta kommando (där X är den identifierade bokstaven eller siffran från följande stycken) kan komma till nytta för att hitta omnämnda delar i den givna källkoden:

```
egrep -in "[*]L[*]" *
```

## 2 Grundläggande

### 2.1 “L” Laddning

Förklara hur bildinläsningskoden fungerar. Vad är skillnaden mellan en `SDL_Surface` och en `SDL_Texture`?

### 2.2 “B” Ritning

Förklara vad argumenten till `SDL_RenderCopy` gör. När skulle man vilja använda `SDL_RenderCopyEx` istället?

### 2.3 “T” FPS

Förklara vad den givna koden gör och hur den fungerar. Hur lång tid är ett tick? Hur många bildrutor per sekund (frames per second, FPS) siktar den givna koden på?

Varför används `int` och inte `Uint32` i den givna koden? Prova att ändra om `int` till `Uint32`, vad händer när ni kör koden?

### 2.4 “C” Genomskinlighet

Öppna png-filerna som kommer med projektet i GIMP eller liknande bildbehandlingsprogram. Hur kan de här bilderna, som av naturen är fyrkantiga, ändå se icke-fyrkantiga ut när de ritas ut?

## 3 Extrauppgifter

### 3.1 Scope

Undersök de olika metoder som används i exempelprojektet för att definiera scope i objekt.

#### 3.1.1 Area

Använder ingen metod, utan ser till att inga namnkollisioner sker genom att använda lite konstigare variabelnamn.

#### 3.1.2 AnimatedImage

Här används `::` för att referera till klassen, med `AnimatedImage::` innan varje klass- eller instansvariabel.

#### 3.1.3 CirclingImage

Här används `this`-pekaren, med `this->` innan varje instansvariabel.

Varför är `AnimatedImage::`-metoden önskad om man tänker ärva av klassen som använder den?

### 3.2 Kollision

Studera kollisionsdetekteringen i `area.cpp`.

- Vilken av de två metoderna (`isCollidingWithCircle` / `isCollidingWithRect`) anser ni vara bäst?
- Är `const Area&` ett bra val av indatatyp för en kollisionsdetekteringsfunktion? Varför, eller varför inte?
- Vad innebär det att det står `const` efter en funktionsdeklaration? (sök efter "1")

### 3.3 Omdöpning

Döp om `Image` till `Sprite` och lös alla fel som uppstår.

### 3.4 Minnesläcka

Varje gång `Image` allokerar minne för en bild så skriver den ut att den gör det i konsolen, och varje gång den frigör minnet så skrivs ytterligare ett meddelande ut. Om ni studerar utskriften så ser ni att det ibland allokeras minne som inte frigörs. Försök att hitta var minnet glöms bort och frigör det ordentligt.