

# TDP005: SDL

Niklas Hayer <nikha354@student.liu.se>  
Torbjörn Lönnemark <torlo515@student.liu.se>

15 november 2009

## 1 Förberedande

Börja med att ladda ner Dokumentationen till SDL från <http://www.libsdl.org/archives/sdl-doc-html.zip> och packa upp den på lämpligt ställe.

Den kommer att behövas under laborationens gång.

Ladda även ner det projekt vi kommer att använda under laborationen. Packa upp det och kompilera det med make för att se till att SDL är installerat och fungerande.

De av er som har glömt STONE kommer vilja notera detta kommando: (där X är en bokstav eller siffra)

```
egrep -in "/*X*/" *.cpp *.h
```

som kan användas för att hitta den kod som refereras till i labbhandledningen.

## 2 Grundläggande

### 2.1 /\*L\*/ Laddning

Förklara hur bildinläsningskoden fungerar. Varför används `SDL_DisplayFormat`?

### 2.2 /\*B\*/ Ritning

Förklara vad argumenten till `SDL_BlitSurface` gör.

## 2.3 /\*T\*/ FPS

Förklara vad den givna koden gör och hur den fungerar. Hur lång tid är en tick? Hur många bildrutor per sekund siktar den givna koden på? Varför används int och inte Uint32 i den givna koden? (Pröva att ändra om int till Uint32, vad händer när du kör koden?)

## 2.4 /\*C\*/ Genomskinlighet

Förklara skillnaderna mellan att använda en färgnyckel (Chroma key) eller en alfakanal (Alpha compositing) för genomskinlighet.

# 3 Utökning

## 3.1 /\*F\*/ Förbättra FadeOutScreen

Den FadeOutScreen som just nu används i exemplet är ett fult hack, men för en bättre lösning så behöver vi installera ett utökningsbibliotek till SDL.

- Installera SDL\_GFX
- Ändra i makefilen så att den länkar med SDL\_GFX biblioteket
- Inkludera SDL\_gfxPrimitives.h i FadeOutScreen.cpp
- Ersätt raderna markerade med /\*F\*/ med ett anrop till boxRGBA

# 4 Extrauppgifter

## 4.1 Scope

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

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

**AnimatedImage** Använder sig av att referera till klassen. (AnimatedImage:: innan varje klass- eller instansvariabel)

**CirclingImage** Använder sig av this-pekaren. (this-> innan varje instansvariabel)

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

## 4.2 Kollision

Studera kollisionsdetekteringen i `area.cpp`

- Vilken av de två metoderna (`isCollidingWithCircle`/`isCollidingWithRect`) anser du vara bäst?
- Är `"const Area&"` ett bra val av indatatyp för en kollisionsdetekteringsfunktion? Om inte, vilken variabeltyp vore bättre?
- Vad innebär det att det står `const` efter en funktionsdeklaration? (se `/*1*/`)

## 4.3 Omdöpning

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

## 4.4 Minnesläcka

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