

TDP005: Introduktion till SDL

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

Niklas Hayer

2013-11-10

1 Förberedande

Börja med att öppna upp en webbläsare med dokumentationen till SDL från <http://www.libsdl.org/release/SDL-1.2.15/docs/>. 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 SDL 1.2.

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 ett tick? Hur många bildrutor per sekund siktar den givna koden med?

Varför används `int` och inte `Uint32` i den givna koden? Prova 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

`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.

4.1.1 Area

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

4.1.2 AnimatedImage

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

4.1.3 CirclingImage

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

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

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 `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 du studerar utskriften så ser du att det ibland allokeras minne som inte frigörs. Försök att hitta var minnet glöms bort och frigör det ordentligt.