Blekinge Tekniska Högskola
Institutionen för datalogi och datorsystemteknik

Projektuppgift Bildgenerering

Inledande programmering i C
DV1550, DV1552

Författare: Carina Nilsson, Christian Nordahl, Cuong Phung

Innehåll

1 Introduktion

I det här momentet skall du göra en mer omfattande programmeringsuppgift. Examinationsmomentet är på 1,5 högskolepoäng och ska motsvara en veckas arbete på heltid, det vill säga ungefär 40 aktiva arbetstimmar. Projektuppgiften görs *individuellt*.
Uppgiften handlar om att göra program som manipulerar digitala bilder. Här tränas färdighet i att sätta sig in i redan befintlig kod och sätta in sin egen kod i ett större sammanhang.

1.1 RGB

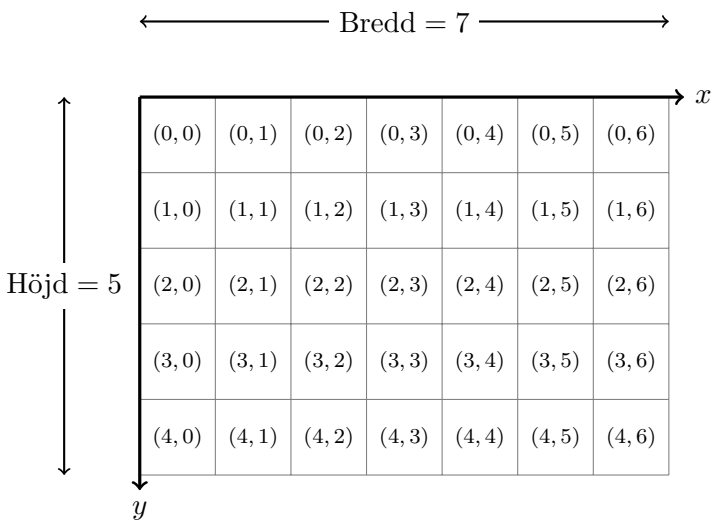
I projektuppgiften ska du jobba med bilder där färgen för varje bildpunkt (pixel) representeras i RGB. RGB (Red, Green, Blue) är en sorts additiv färgblandning, där man tar ljus av olika färger och blandar ihop dem till en ny färg. I en datorskärm består varje pixel av 3 dioder, en röd, en grön och en blå, som lyser. Dessa olika ljus blandas sedan ihop av människans ögon och vi uppfattar pixeln som enfärgad istället för tre individuella färger.

Beroende på hur starkt varje diod lyser så uppfattar vi färgen olika. Värdena på respektive diod är från 0 till 255. Nedan kommer ett par exempel på några olika färgkoder.

- Röd : {255,0,0}
- Grön: {0,255,0}
- Blå : {0,0,255}
- Gul : {255,255,0}
- Magenta: {255,0,255}
- Cyan: {0,255,255}
- Svart: {0,0,0}
- Brun: {150,74,0}
- Orange: {237,145,33}

1.2 Digital bildrepresentation

För att bygga upp en bild så behöver du använda dig utav en 2-dimensionell array av pixelelement. Access av ett element sker genom `pixels[heightIndex][widthIndex]`, uppifrån och ner. Antag att vår pixelarray har bredden `WIDTH` och höjden `HEIGHT`. Om pixeln som finns i det översta vänstra hörnet ska användas skriver man `pixels[0][0]`, pixeln i översta i högra hörnet `pixels[0][WIDTH-1]`, pixeln i nedre högra hörnet `pixels[HEIGHT-1][WIDTH-1]`, och pixeln i nedre vänstra hörnet `pixels[HEIGHT-1][0]`. Nästkommande bild ger en mer visuell förklaring.



Figur 1: Visar koordinatsystemet för bilderna i projektet.

2 Installation

Projektuppgiften kräver ett tredjepartsbiblioteket *libpng (och zlib)* för att skapa, manipulera, läsa, och spara bilder. Detta bibliotek finns förkompilerat för OS X och Linux (Ubuntu). För Visual Studio finns en zip-fil innehållande en mapp med en färdiginställd ”Solution” som innehåller diverse filer.

2.1 Windows — Visual Studio 2015

Det finns en mapp med en solutionfil på It’s som ska användas. Alla nödvändiga inställningar för att kompilering och länkning med biblioteken ska fungera är gjorda där. Testprogrammet som finns i `main.c` skriver en enkel bild till din disk.

Bilden som produceras när programmet körs heter `test.png` och bör se ut som följande bild.

Figur 2: test.png

2.2 Linux (Ubuntu) / OS X

libpng och *zlib* är oftast installerade från start i både Ubuntu och OS X. Om de inte redan finns kan de i Ubuntu installeras (och sedan eventuellt tas bort när projektet är klart).

För Ubuntu gör följande kommando: `sudo apt-get install zlib1g-dev libpng-dev`

För OS X gör följande steg:

1. Installera Homebrew. (Eller annan valfri paketinstallerare).
2. Installera libpng med kommandot `brew install libpng`

För att underlätta kompilering av filerna har vi gjort något som kallas för en `Makefile`. Den hanterar all kompilering när kommandot `make` används i terminalen. Kommandot körs från rotmappen där `Makefile`-filen ligger, alltså ett steg ovanför mappen `src`.

Vi har även gjort ett simpelt testprogram som testar att läsa och skriva en bild till din disk. Testprogrammet kompileras genom att skriva `make`. Kompilerar det utan några komplikationer så kan du köra den körbara filen `run`. Detta görs genom t.ex skriva `bin/run`. Kontakta din lärare om det inte fungerar. Bilden som produceras när `run` körs heter `test.png` och bör se ut som följande bild.

Figur 3: test.png

3 Given funktionalitet

Vi tillhandahåller vissa hjälpfunktioner för att minska svårigheten på uppgiften. De finns deklarerade i filen `functions.h`. I `functions.h` ligger även deklaration av två `structs`, `Pixel`, som skall användas för att hålla bildpixlarna under programkörningen, och `Image`, som innehåller en bilds pixlar, samt bildens bredd och höjd.

3.1 struct-typen Pixel

Deklarationen för `Pixel` ser ut så här:

```
typedef struct pixel {
    uint8_t r;
    uint8_t g;
    uint8_t b;
} Pixel;
```

Typen `uint8_t` finns specificerad i ett bibliotek och är egentligen bara ett nytt namn på `unsigned char`. Syftet med namnet är att tydliggöra att vi menar ett 8-bits teckenlöst tal vid användningen och inte ASCII-koden för ett tecken. Nyckelordet `typedef` i C gör att man inte behöver skriva `struct pixel` vid dekaration av variabler av den här `struct`-typen, utan det räcker med typnamnet `Pixel`.

3.2 struct-typen Image

Deklarationen för `Image` ser ut så här:

```
typedef struct image {
    Pixel **pixels;
    unsigned int height;
    unsigned int width;
} Image;
```

Syftet med denna `struct` är att underlätta arbetet med att skicka information mellan funktioner. Den innehåller en bilds bredd, höjd och pixlar (2-dimensionell array). Värt att notera är att allokering behövs göras till `pixels` när en bild görs. Om du inte läser in en bild m.h.a. funktionen `readImage` (se kapitel 3.3) får du göra detta själv. Avallokering överlåts helt till studenten.

3.3 Hjälpfunktionerna readImage() och writeImage()

Följande två funktioner finns färdiga att använda:

```
void writeImage(char* filename, Image *img);
```

Funktionen skriver en bild till en fil på disk, kräver filnamn och en pekare till en `Image` som parameter.

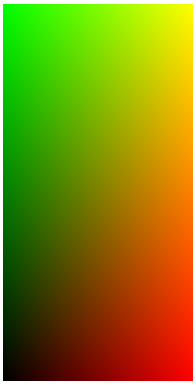
```
void readImage(char *filename, Image *img);
```

Funktionen läser in en bild från disk till en `Image`-post. Kräver filnamn och en pekare till en `Image`-pekare som parametrar. Funktionen kommer allokera minne till den 2-dimensionella `Pixel`-arrayen och sätter korrekta värden till variablerna `height` och `width` i posten.

4 Uppgifter

Du ska utveckla ett bildhanteringsprogram som är menystyrt, där tanken är att man via ett menyval läser in en bild, sedan via andra menyval manipulerar den och sedan sparar ner den till fil med ett annat menyval.
Menyvalen som ska kunna göras är:

1. Läsa in en bild från fil.
2. Spara en bild på fil.
3. Invertera en bild. (Alla RGB-värden i bilden ska bli (255 - orginalvärdet))
4. Färgskifta en bild. (Flytta värdet på röd till grön, grön till blå, blå till röd i alla pixlar i bilden.)
5. Vänd en bild på diagonalen. (Se Figur 4)
6. Förstora en bild. (Se Figur 5)
7. Zooma in en bild. (Se Figur 6)
8. Avsluta programmet.



Figur 4: Bilden i Figur 2 vänd på diagonalen.

Figur 5: Förstorad version av Figur 2.

Figur 6: Inzoomad version av Figur 2.

5 Krav & Kriterier

Uppgiften under avsnitt 4 i detta dokument ska implementeras i sin helhet.

- Respektive delproblem skall implementeras i en egen funktion som anropas från respektive menyval.
- Inga minnesläckor.
- Inga globala variabler.
- `#define` macron och konstanter ska vara globalt placerade.
- Kodstandarden som beskrivs i Kodstandardsdokumentet på It's Learning skall efterföljas.
- Inga kompileringsfel får finnas.

Tips! Organisera ditt programmeringsarbete så att du kan testa varje enskild del du gör i programmet. Vänta *inte* med att testa tills du tror att allt är klart.

6 Redovisning

Skriv koden för ditt huvudprogram i filen `main.c`. Dina funktioner läggs i en egen `.c`-fil `studentFunctions.c` och dina deklarationer i en egen `.h`-fil `studentFunctions.h`. Ladda upp dessa filer som separata textfiler (filtillägget `.txt`) i It's Learning.
Se också till att allt du lämnar in är skrivet på engelska.

VIKTIGT!!
Det kommer *inte* att vara möjligt att lämna in efter deadline. Om du missat den får du ingen ny chans förrän i läsperiod 3, och då får du en annan uppgift.

⚠ Observera att det inte är tillåtet att lämna in kod som någon annan än du själv har skrivit. Inlämningen kommer att plagiatkontrolleras.
Du får dock självklart använda och kopiera de filer vi tillhandahåller för att lösa uppgiften.

Projektuppgiften ska dessutom (eventuellt tillsammans med laborationsinlämningarna) redovisas muntligen för en lärare. Redovisningarna kommer att ske i tentamensperioden och anmälan till den redovisningen kommer att kunna göras i I'ts Learning. För att få redovisa projektet muntligt måste det vara inlämnat.