

INDU: ASCII Art Studio

Lars Arvestad

Innehåll

Introduktion	1
Hjälpmedel	1
Uppgift 1 (E-C)	2
Exempelkörning	3
Krav på uppgift 1	4
Uppgift 2 (E-A)	4
Utökad funktionalitet	4
Kommandospråket	5
Exempelkörning	5
Krav på uppgift 2	7
Betygsättning	7

Introduktion

Konstformen *ASCII art* är från datorernas barndom och den går ut på att uttrycka sig visuellt med textfiler. Dessa kunde visas på textterminaler och skickas till enkla skrivare utan grafikmöjligheter. Se figur 2 för ett exempel. *ASCII* är den teckenkodning som lanserades på 60-talet och kan återfinnas än idag. Principen med *ASCII art* är att helt vita bildpunkter representeras av ett mellanslag, svart av något tecken med ”mycket trycksvärta”, till exempel M, och låter grånivåer ges av andra tecken. Figur 1 visar hur en gråskala kan representeras med *ASCII art*.

I det här projektet ska du skriva ett program, *ASCII Art Studio*, där man kan experimentera med interaktivt med konstformen, men med automatisk rendering av bildfiler till *ASCII*. Ditt program ska kunna läsa till exempel JPG- och PNG-bilder och med ett enkelt kommandospråk ska användaren kunna hantera flera bilder samtidigt, justera kontrast och exponering av bilderna, och experimentera med olika storlekar på utdata.

Projektet har två uppgifter på olika nivå. Den första är enklare och kan som högst ge dig betyg C. Den andra nivån kan ge betyg A, men kräver mer implementerad funktionalitet och att du dessutom använder objektorientering för implementationen. Om du bestämt dig för A-nivån behöver du inte lämna in en lösning för uppgift 1.

Hjälpmedel

För att hantera bildfiler rekommenderar vi modulen *pillow* (*Python Image Library*). Det finns bra med [dokumentation för pillow](#), både teknisk referensdokumentation och allmän handledning med bra exempel. Som komplement listar vi några operationer som är bra att använda.

För uppgift 2 är det bra att använda en modul för serialisering (dvs, att transformera en datastruktur eller klass till ett textrepresentation på fil). Vi rekommenderar *json*, men det finns andra som också får användas, som till exempel *pickle*.

För kommandospråket får du använda modulen *re* om du vill. Observera att den antagligen är jobbig att använda om man inte lärt sig om *regex* sen tidigare.

Du får också använda *math*.



I exemplet nedan är AAS programmets prompt. Instruktionerna är:

- ```
Welcome to ASCII Art Studio!
AAS: load image slalom.jpg
AAS: render
```

```
AAS: load grayscale.jpg
AAS: render
.....:ijjjjxxxxooo0hhhhhXXXX%@@@#####
```

AAS: quit  
Bye!

## Kommandospråket

De kommandon som ska stödjas är:

**load image *filename*** Läs in filen i *filename* och spara den i sessionen under dess filnamn. Senast inlästa bild är också *current*.

**load image *filename* as *alias*** Som ovan, men spara bilden under namnet *alias*.

**info** Lista inlästa bilder och deras egenskaper, samt vilken bild som *current* pekar på.

**render** Skapa ASCII art för aktuell bild. När kommandot ges ska bilden ges rätt storlek, kontrast, och ljusstyrka. Bredden på renderad bild ska vara 50, om annat ej angetts (se nedan). Höjden bör väljas så att bildens proportioner bevaras och med en justering för att de rektanglar som fonter använder tenderar att vara smala (och fontberoende).

**render *img*** Som *render*, men för den bild som är sparad som med namnet *img*. Det kan vara filnamnet, dess *alias*, eller *current*.

**render *img* to *filename*** Som ovan, men utdata sparas på fil.

**set *img* width *num*** Bestäm att bredden på bilden *img* (*alias* eller filnamn) ska vara *num*. Bildens höjd ska justeras så att proportionen höjd/bredd bibehålls. Det är denna storlek som ska användas när man renderar bilden.

**set *img* height *num*** Som ovan, men för bildens höjd.

**set *img* brightness *num*** Ange hur ljusstyrkan för bilden *img* ska ändras jämfört med originalet innan renderas. Om *num* är 1.1 så ska bilden bli 10% ljusare, och om *num* är 0.8 så ska bilden bli 20% mörkare.

**set *img* contrast *num*** Som ovan, men för kontrast.

**save session as *filename*** De inlästa bilderna sparas som filnamn, storlek, ljusstyrka, och kontrast. Bildens pixlar sparas inte. Aktuell bild, *current*, sparas också.

**load session *filename*** Läs in den session som är sparad i *filename*. De bilder vars filnamn är sparade ska läsas in på nytt och sparade parametrar ska sättas. Aktuell bild ska vara den som anges i den sparade sessionen.

**quit** Avslutar sessionen.

## Exempelkörning

I följande exempel läser vi in bilderna som är givna i *lasse.jpg* och *stadshuset.jpg*

```
Welcome to ASCII Art Studio!
AAS: load image lasse.jpg as lasse
AAS: load image stadshuset.jpg as hus
AAS: info
=== Current session ===
Images:
lasse
 filename: lasse.jpg
 size: (2458, 3565)
 target size: (50, 44)
 brightness: 1.0
 contrast: 1.0

hus
 filename: stadshuset.jpg
 size: (640, 426)
 target size: (50, 20)
```

```
Current image: hus
AAS: save session as s1
AAS: quit
Bye!
```

Welcome to ASCII Art Studio!

AAS: render

```
AAS: set hus width 80
```

AAS: render

6

