

Programmering og Problemløsning

Datalogisk Institut, Københavns Universitet

Arbejdsseddel 4 - gruppeopgave

Jon Sparring

26. september - 3. oktober.
Afleveringsfrist: onsdag d. 3. oktober kl. 22:00

I denne periode skal I arbejde i grupper. Formålet er at arbejde med:

- Moduler og namespaces
- Afprøvning

Opgaverne er delt i øve- og afleveringsopgaver.

Øveopgaver

En farve repræsenteres ofte som en triple (rød, grøn, blå), hvor hver indgang kaldes en farvekanal eller blot en kanal, og hver kanal er typisk et heltal mellem 0 og 255:

$$c = (r, g, b) \tag{1}$$

Farver kan lægges sammen ved at addere deres kanaler,

$$c_1 + c_2 = (\text{trunc}(r_1 + r_2), \text{trunc}(g_1 + g_2), \text{trunc}(b_1 + b_2)), \tag{2}$$

$$c_i = (r_i, g_i, b_i) \tag{3}$$

$$\text{trunc}(v) = \begin{cases} 0, & v < 0 \\ 255, & v > 255 \\ v, & \text{ellers} \end{cases} \tag{4}$$

og farver kan skaleres ved at gange hver kanal med samme konstant.

$$ac = (\text{trunc}(ar), \text{trunc}(ag), \text{trunc}(ab)) \tag{5}$$

Farver, hvor alle kanaler har samme værdi, $v = r = g = b$, kaldes gråtoner, og man kan konvertere en farve til gråtone ved at udregne gråtoneværdien som gennemsnittet af de 3 kanaler,

$$v = \text{gray}(c) = \frac{r + g + b}{3} \tag{6}$$

- 4ø.0 Skriv en signaturfil for et modul, som indeholder funktionerne `trunc`, `add`, `scale`, og `gray` ud fra ovenstående matematiske definitioner og ved brug af tupler, hvor muligt.
- 4ø.1 Skriv en implementation af ovenstående signaturfil og kompilér begge filer til et bibliotek (dll-fil).
- 4ø.2 Skriv en applikation i 2 varianter: Een som benytter ovenstående bibliotek via `fsharp` og een som benytter det via `fsharpc`.
- 4ø.3 Lav en White-box afprøvning af jeres bibliotek og ved brug af `fsharpc`.
- 4ø.4 Udvid biblioteket (både signatur og implementationsfilen) med en funktion, som konverterer en farvetriplet til en gråtonetriplet. Udvid afprøvningen med en passende afprøvning af den nye funktion. Diskutér om bibliotek, applikation, og afprøvning er struktureret på en måde, så denne udvidelse har været let, eller om der er uhensigtsmæssige afhængigheder, som gør rettelse, vedligeholdelse og udvidelse besværlig og med risiko for fejl.

Afleveringsopgaver

Afleveringsopgaven omhandler to-dimensionelle vektorer. En to-dimensionel vektor (herefter blot vektor) er et geometrisk objekt som består af en retning og en længde. Typisk repræsenteres vektorer som par af tal $\vec{v} = (x, y)$, hvor længden og retning findes ved,

$$\text{len}(\vec{v}) = \sqrt{x^2 + y^2} \quad (7)$$

$$\text{ang}(\vec{v}) = \text{atan2}(y, x) \quad (8)$$

Vektorens ender kaldes hhv. hale og spids, og når halen placeres i $(0, 0)$, så vil spidsen have koordinat (x, y) . Vektorer har en række standardoperatorer,

$$\vec{v}_1 = (x_1, y_1) \quad (9)$$

$$\vec{v}_2 = (x_2, y_2) \quad (10)$$

$$a\vec{v}_1 = (ax_1, ay_1) \quad (11)$$

$$\vec{v}_1 + \vec{v}_2 = (x_1 + x_2, y_1 + y_2) \quad (12)$$

$$\vec{v}_1 \cdot \vec{v}_2 = x_1x_2 + y_1y_2 \quad (13)$$

Addition kan tegnes som vist i Figure 1.

4g.0 Skriv et bibliotek `vec2d.fs`, som implementerer følgende signatur fil:

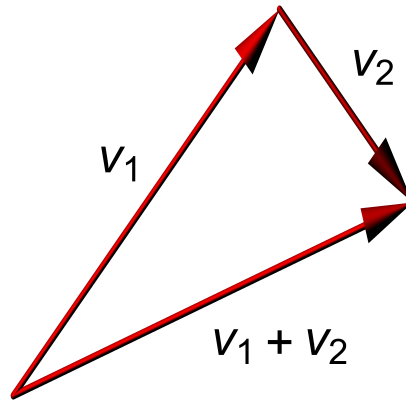


Figure 1: Illustration of vector addition in two dimensions.

Listing 1 vec2d.fsi:
A signature file.

```

1  /// A 2 dimensional vector library.
2  /// Vectors are represented as pairs of floats
3  module vec2d
4  /// The length of a vector
5  val len : float * float -> float
6  /// The angle of a vector
7  val ang : float * float -> float
8  /// Multiplication of a float and a vector
9  val scale : float -> float * float -> float * float
10 /// Addition of two vectors
11 val add : float * float -> float * float -> float * float
12 /// Dot product of two vectors
13 val dot : float * float -> float * float -> float

```

4g.1 Skriv en White-box afprøvning af biblioteket.

4g.2 Punkter på en cirkel med radius 1 kan beregnes som $(\cos \theta, \sin \theta)$, $\theta \in [0, 2\pi)$. Betragt det lukkede polygon, som består af $n > 1$ punkter på en cirkel, hvor $\theta_i = \frac{2\pi i}{n}$, $i = 0..(n-1)$.

(a) Skriv en applikation med en funktion,

`polyLen : n:int -> float`

som benytter ovenstående bibliotek til at udregne længden af polygonet. Længden udregnes som summen af længden af vektorerne mellem nabopunkter. Applikationen skal desuden udskrive en tabel af længder for et stigende antal værdier n , og resultaterne skal sammenlignes med omkredsen af cirklen med radius 1.

(b) Udform en hypotese ud fra tabellen for længden af polygonet når $n \rightarrow \infty$.

4g.3 Biblioteket `vec2d` tager udgangspunkt i en repræsentation af vektorer som par (2-tupler). Lav et udkast til en signaturfil for en variant af biblioteket, som ungår tupler helt. Diskutér eventuelle udfordringer og større ændringer, som varianten ville kræve både for implementationen og applikationen.

Afleveringen skal bestå af en zip-fil, som skal indeholde en mappe med fsharp koden. Der skal være en fsharp tekstfil per fsharp-opgave, og de skal navngives `4g0.fsx` osv. De skal kunne oversættes med `fsharpc` og køres med `mono`. Funktioner skal dokumenteres ifølge dokumentationsstandarden, og udover selve programteksten skal besvarelserne indtastes som kommentarer i de `fsx`-filer, de hører til.