

# Aflevering 1

Peter Asp Hansen

23. februar 2023

## Delopgave 1

I delopgave 1 laves et program som tager et input som er en tekststreng fra brugeren vha. kommandoen `System.Console.ReadLine` og printer "fsharp is cool", hvis fsharp er indtastet. Hvis der indtastes noget andet, f.eks. et andet kodesprog såsom python, printes der "I don't know python". Yderligere kan man afslutte programmet ved at skrive quit. Koden er vist nedenfor.

```
let rec readCode () =  
    let (a:string) = System.Console.ReadLine ()  
    match a with  
        "fsharp" ->  
            printfn "fsharp_is_cool"  
            printfn "Please_enter_a_programming_language:"  
            readCode ()  
        | "quit" ->  
            ()  
        | _ ->  
            printfn "I_don't_know_%A" a  
            printfn "Please_enter_a_programming_language:"  
            readCode ()  
    printfn "Please_enter_a_programming_language:"  
    let b = readCode ()  
    b
```

I dette program benytter vi en rekursiv funktion, kaldet `readCode`. Vi benytter denne type funktion fordi, at vi har behov for at funktionen kan kalde sig selv indtil vi beder den om at slutte vha. indtastningen quit. Derefter benytter vi en hjælpefunktion hvor vores input bliver defineret vha. `System.Console.ReadLine`. Dette input tildeler vi variabellet `a`. Dernæst parrer vi `a` med det ønskede output vha. `match-with` udtrykket. Her inddeler vi alle muligheder for `a` i tre kategorier, fsharp, quit og alt andet. Alt andet defineres med et wildcard som skrive `"_"`. Bemærk også der benyttes `"|"` for at indikere eller. Til sidst printes teksten "Please enter a programming language" og funktionen kaldes vha variabellet `b`. Her er en lille demo:

```
Please enter a programming language:  
c  
I don't know "c"  
Please enter a programming language:  
fsharp  
fsharp is cool  
Please enter a programming language:  
quit
```

Figur 1: Demo

## Delopgave 2

I delopgave 2 laves et program som bevæger en firkant horisontalt på et Canvas. Yderligere forbindes det øvre venstre hjørne af vores canvas til det øvre venstre hjørne af firkanten med en linje. Dette gøres for alle hjørner af Canvas og det tilsvarende hjørne af firkanten. Koden er vist nedenfor.

```
#r "nuget:DIKU.Canvas, 1.0"
open Canvas

type state = int

let draw w h (s:state)=
    let C = create w h
    let l = w / 4
    let r = 3*l
    do setFillBox C black (l+s, l) (r+s, r)
    do setLine C lightgrey (0,0)(l+s,l)
    do setLine C lightgrey (w,0)(r+s,l)
    do setLine C lightgrey (0,w)(l+s,r)
    do setLine C lightgrey (w,w)(r+s,r)
    C

let react (s:state)(k:key) : state option =
    match getKey k with
    | LeftArrow -> Some (s-5)
    | RightArrow -> Some (s+5)
    | _ -> None

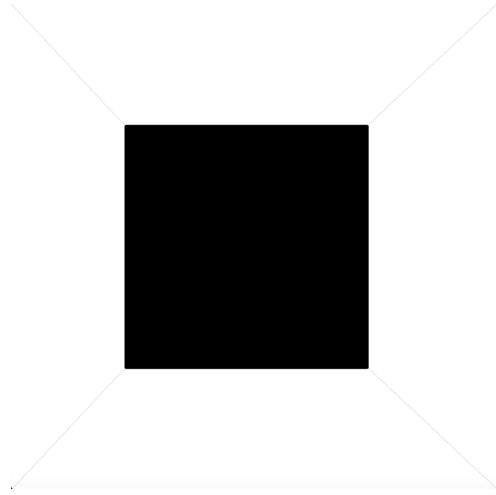
do runApp "ColorBoxes" 600 600 draw react 0
```

Først åbnes canvas og vi definerer en type som vi kalder state, og definerer den som et heltal.

Derefter danner vi en funktion som kaldes draw. Denne funktion laver et canvas med bredden w og højden h. Derudover defineres også et par andre variabler l og r som henholdsvis bliver defineret som en 1/4 bredden og 3/4 bredden. Vi benytter disse variabler til at beskrive boxen og de fire linjer. Bemærk at vi også benytter variablen s som vi bruger til at ændre på deres placering.

Den næste funktion kalder vi react. Denne funktion bruger vi til at ændre på værdien af s vha a piletasterne. Dette gøres vha en option type, hvor man blandt andet kan se at når venstre piletast benyttes returneres Some(s-5) som ændrer s værdien med -5. Derudover bruges der også et wildcard for alle andre taster og der returneres none, altså der sker ingenting.

Til sidst køres programmet med kommandoen do runApp "MovingBox"600 600 draw react 0. Her er w og h sat til 600 pixels. Yderligere tager den de to funktioner som input og start værdien af s er sat til 0. Køres programmet er det her det første man ser:



Figur 2: Canvas