

# Opgave 1i

Alexander Svanholm Bang

13. september 2022

## Opgave 1i0

I denne opgave skulle vi tage et eksisterende program og ændre det således at programmet ville spørge useren om at skrive navnet på et programmeringssprog. Den skulle så printe "Fsharp is cool" hvis den fik input "fsharp". Ellers skulle den skrive "I don't know <input>", hvor input var det man skrev. Et ekstra tilfælde var at hvis man skrev "quit" vil programmet slukke.

Det jeg fandt på som løsning var at ændre det oprindelige pattern matching til at matche "quit" og "fsharp" og i alle andre tilfælde vil den matche med den sidste.

```
(* Fsharp is cool *)
let rec readNonZeroValue () =
    let a = string (System.Console.ReadLine())
    match a with
    | "fsharp" ->
        printfn "Fsharp is cool"
        readNonZeroValue ()
    | "quit" -> printfn "Quitting"
    | _ ->
        printfn "I don't know %A" a
        readNonZeroValue ()

printfn "Please enter the name of a programming language:"
let b = readNonZeroValue ()
```

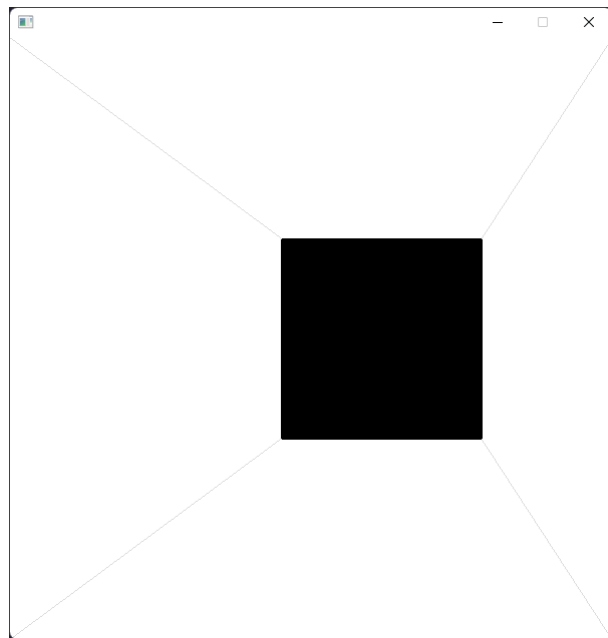
## Opgave 1i1

I denne opgave skulle vi lave en sort boks som kunne rykke sig til højre og til venstre ved at bruge piltasterne på tastaturet. Dette fik jeg gjort ved at ændre på programmet vi fik gevet. Det jeg gjorde var at sætte state til at være en integer (som var offset fra midten) og bare rykke koordinaterne af selve boksen og de grå linjer.

```
type state = int
let draw w h (s: state) =
  let C = create w h
  setFillBox C black (200 + s, 200) (400 + s, 400)
  setLine C lightgrey (0, 0) (200 + s, 200)
  setLine C lightgrey (w, 0) (400 + s, 200)
  setLine C lightgrey (0, h) (200 + s, 400)
  setLine C lightgrey (w, h) (400 + s, 400)
  C
```

I react funktionen fik jeg den bare til at reagere på venstre og højre piltaster og ændrede state til at addere eller trække fra alt efter hvilken taste blev trykket.

```
let react (s: state) (k: key) : state option =
  match getKey k with
  | LeftArrow -> Some(s - 10)
  | RightArrow -> Some(s + 10)
  | _ -> None
```



Figur 1: Screenshot fra programmet 1i1.fsx