

## 1.i

Jonas Hansen, Metin Wesselhoff, Alexander Bang

24. september 2022

### Opgave 2g0

Vi har i denne opgave skulle benytte 8 steps guiden til at skrive 4 funktioner der lavede forskellige regneoperationer på en vektor

- a I denne opgave lavede vi bare en funktion som tog to vektorer som argumenter og adderede dem sammen.

```
(* Addition af to vektorer *)
/// <summary>
///   Given two vectors , add them together
/// </summary>
/// <param name="vec1">First Vector</param>
/// <param name="vec2">Second Vector</param>
/// <returns>New Vector vec1 + vec2</returns>
let add ((a1, b1): vec) ((a2, b2): vec) : vec = a1 + a2, b1 + b2
```

- b Her tog vi en vektor og gangede dem med en skalarværdi.

```
(* Gang vektor med konstant *)
/// <summary>
///   Given a vector and a scalar value , multiply vector with scalar
/// </summary>
/// <param name="vec1">First Vector</param>
/// <param name="c">Constant</param>
/// <returns>New Vector vec1 * c</returns>
let mul ((a1, b1): vec) (c: float) : vec = a1 * c, b1 * c
```

- c Denne opgave benytter de matematiske funktioner som var givet på forhånd til at rotere vektoren

```
(* Roter vektor *)
/// <summary>
///   Given a vector , rotate it by "r" radians
/// </summary>
/// <param name="vec">Vector</param>
/// <param name="r">Rotation value in Radians</param>
/// <returns>New Vector Rotation(vec)</returns>
let rot ((a, b): vec) (r: float) : vec =
    (a * cos (r) - b * sin (r), a * sin (r) + b * cos (r))
```

Funktionaliteten i programmet er baseret på at brugeren indtaster længden og retningen på en vektor, hvor programmet så kan rotere, multiplicere, eller plusse to vektorer med hinanden. Alt efter hvilken funktion der vil blive kaldt, hvor programmet så retunerer den nye vektorer der så er blevet beregnet

### Opgave 2g1

Vi har i denne her opgave skulle benytte canvas funktionerne til at tegne en række vektorer som skal kunne roteres

I denne her opgave skulle man benytte sig af meget basal matematik igen så her er optionaliteten heller ikke så stor.

Denne opgave tager en vektor af floats ændrer til heltal(integers)

- a (\* Skift fra float til int \*)

```

/// <summary>
///   Given a vector, og two float values, convert to integer
/// </summary>
/// <param name="vec">Vector</param>
/// <returns>New Vector in integers</returns>
let toInt ((a, b): vec) : int * int = int (a), int (b)

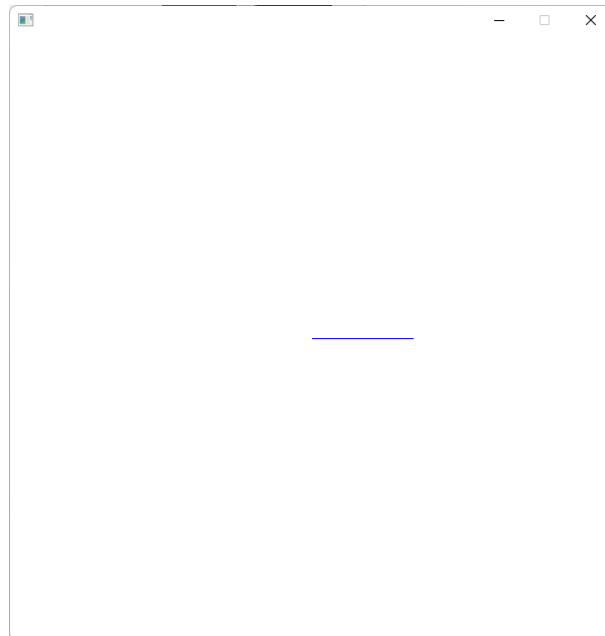
```

- b Til denne opgave har vi lavet en funktion der tegner en vektor, med en vilkårlig, farve der har hale i centrum.

```

/// <summary>
///   Given a point and a vector, draw a line from the point to the vector
/// </summary>
/// <param name="c">Canvas</param>
/// <param name="color">Color og the line</param>
/// <param name="v">Vector</param>
/// <param name="p">Point</param>
/// <returns>Unit</returns>
let setVector (c: canvas) (color: color) (v: vec) (p: vec) : unit =
    setLine c color (toInt p) (toInt (add v p))

```



- c Til denne opgave skal man benytte en løkke. Her kunne man både vælge at benytte sig af at while loop med en mutable index værdi eller et for loop. Vi synes at et for loop gav mest mening at benytte da det holder styr på indexet selv

```

/// <summary>
///   Draw function for the show function to display
/// </summary>
/// <param name="w">Width of window</param>
/// <param name="h">Height of window</param>
/// <returns>Canvas</returns>
let draw w h (s: float) =
    let c = create w h
    let centerx = float (w / 2)
    let centery = float (h / 2)

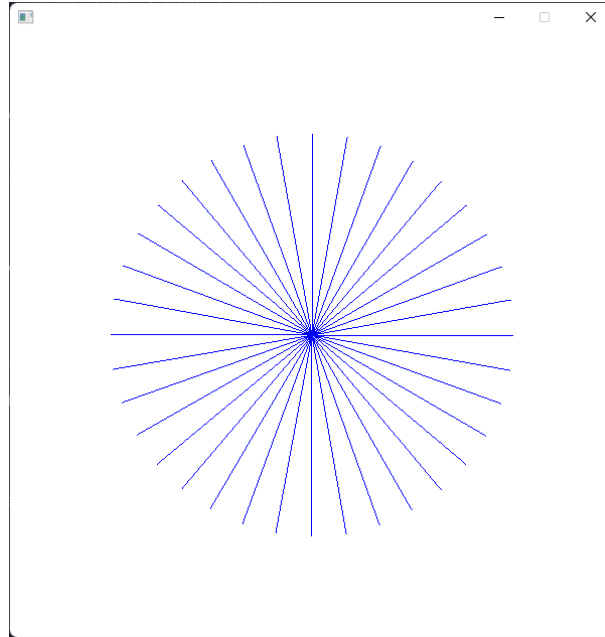
```

```

    for i = 1 to 36 do
        setVector c blue (rot (200, 0) ((float (i * 2) * System.Math.PI) / float (36) +

```

c



d Til denne opgave skulle vi lave en react funktion ligesom i sidste uge. Vi benytter match og Some keywordsne så vi kan differentiere mellem hvilket stykke kode der skal køres efter hvad brugerens input er

Funktionaliteten i programmet er baseret på et for loop der skal tegne 36 vektorer som alle har haler i centrum. Disse vektorer kan så blive flyttet ved at klikke på højre- og venstre piletast