Opgave 3i

Alexander Svanholm Bang

29. september 2022

Opgave $3i0$

$3i0a$)

I denne opgave skulle der laves fire funktioner som tog en liste af tupler af typen `(int*int)` og lavede det om til en liste af `Canvas.turtleCmd` kommandoer.

Nedefor er koden til de fire funktioner til opgaven. Det første er rekursiv og bruger ikke `List` pakken. De andre tre er ikke rekursive men bruger i stedet `List` pakken, hvor den anden bruger `List.map` funktionen, den tredje bruger `List.fold` og den sidste bruger `List.foldBack`.

```
/// <summary>Recursively turns list of
/// moves to list of turtleCmd commands</summary>
/// <param name="lst">List of moves</param>
/// <returns>List of turtleCmd commands</returns>
let rec fromMoveRec (lst: move list) : Canvas.turtleCmd list =
    match lst with
    | [] -> []
    | elm :: rst -> [ Turn(fst elm); Move(snd elm) ] @ fromMoveRec rst

/// <summary>Uses List.map function to turn list of
/// moves to list of turtleCmd commands</summary>
/// <param name="lst">List of moves</param>
/// <returns>List of turtleCmd commands</returns>
let fromMoveMap (lst: move list) : Canvas.turtleCmd list =
    List.concat (List.map (fun elm -> [ Turn(fst elm); Move(snd elm) ]) lst)

/// <summary>Uses List.fold function to turn list of
/// moves to list of turtleCmd commands</summary>
/// <param name="lst">List of moves</param>
/// <returns>List of turtleCmd commands</returns>
let fromMoveFold (lst: move list) : Canvas.turtleCmd list =
    List.rev (List.fold (fun l m -> Move(snd m) :: Turn(fst m) :: l) [] lst)

/// <summary>Uses List.foldBack function to turn
/// list of moves to list of turtleCmd commands</summary>
/// <param name="lst">List of moves</param>
/// <returns>List of turtleCmd commands</returns>
let fromMoveFoldBack (lst: move list) : Canvas.turtleCmd list =
    List.foldBack (fun m l -> Turn(fst m) :: Move(snd m) :: l) lst []
```

$3i0b$)

I denne opgave skulle alle fire funktioner oppefra testes om de giver samme resultet.

De blev testen med listen som var givet i opgaven, og som man kan se nedenfor gav de samme resultat.

```
fromMoveRex: [Turn 10; Move 30; Turn -5; Move 127; Turn 20; Move 90]
fromMoveMap: [Turn 10; Move 30; Turn -5; Move 127; Turn 20; Move 90]
fromMoveFold: [Turn 10; Move 30; Turn -5; Move 127; Turn 20; Move 90]
fromMoveFoldBack: [Turn 10; Move 30; Turn -5; Move 127; Turn 20; Move 90]
```

Figur 1: Resultat fra opgave 3i0b.

Opgave $3i1$

$3i1a)$

I denne opgave skulle man placere et fraktaltræ et tilfældigt sted på skærmen.

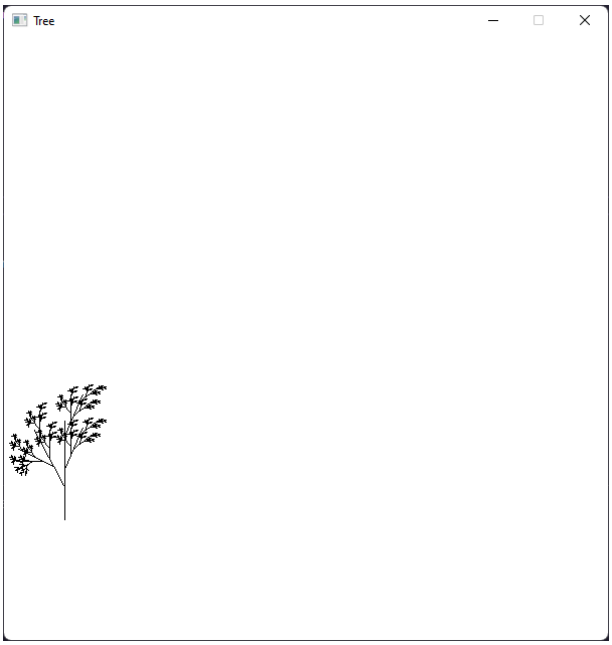
Her blev den i opgaven givede tree funktion brugt og der blev genereret et tal fra 0 til 10 som så blev skaleret højden og bredden af vinduet. Efter det bliver tree funktionen kaldt med pre og post listerne sat på foran og bag den. Det placeret turtleCursoren er tilfældigt sted. Placerer træet, og så går tilbage til centrum.

```
/// <summary>
/// Generates a turtle command tree, but places it randomly in the canvas
/// </summary>
/// <param name="sz">The size of the tree</param>
/// <returns>A list of turtle commands</returns>
let randomTree (sz: int) : Canvas.turtleCmd list =
    let rnd = System.Random()
    let woff = (w / 10) * (rnd.Next 10) - (w / 2)
    let hoff = (h / 10) * (rnd.Next 10) - (h / 2)

    let pre =
        [ PenUp
          Move hoff
          Turn 90
          Move woff
          Turn -90
          PenDown ]

    let post =
        [ PenUp
          Move -hoff
          Turn 90
          Move -woff
          Turn -90 ]

    pre @ tree sz @ post
```



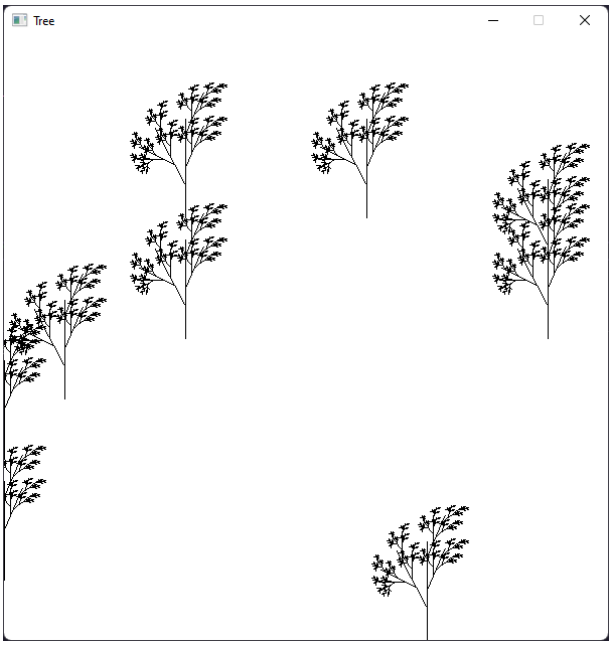
Figur 2: Resultat from opgave 3i1a.

3i1b)

Her skulle vi lave en masse tilfældig fordelte træer på en rekursiv måde.

Denne løsning bruger den forrige `randomTree` funktion og implementerer en rekursiv funktion der kalder `randomTree` n gange og laver en lang list til `turtleDraw` funktionen.

```
/// <summary>Recursively generates randomly
/// places fractal trees in the canvas</summary>
/// <param name="sz">The size of the tree</param>
/// <param name="n">Number of trees to place</param>
/// <returns>A list of turtle commands</returns>
let rec forest (sz: int) (n: int) : Canvas.turtleCmd list =
    if n > 0 then
        randomTree sz @ forest sz (n - 1)
    else
        []
```



Figur 3: Resultat from opgave 3i1b.