

# triangles

Jon Sparring

October 18, 2018

## 1 Lærervejledningn

**Emne** Sumtyper

**Sværhedsgrad** Middel

## 2 Introduktion

Til de følgende opgaver udvider vi typen `figure` med en mulighed for at repræsentere trekanter:

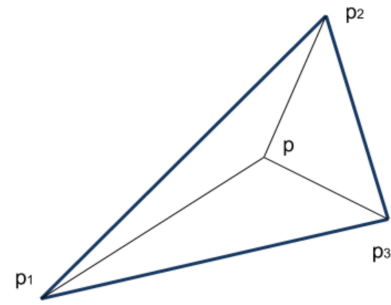
```
type figure =  
  | Circle of point * int * colour  
    // defined by center, radius, and colour  
  | Rectangle of point * point * colour  
    // defined by corners bottom-left, top-right, and colour  
  | Mix of figure * figure  
    // combine figures with mixed colour at overlap  
  | Triangle of point * point * point * colour  
    // defined by the three points and colour
```

Konstruktøren `Triangle` tager tre punkter og en farve som argument. Der er intet krav til hvordan de tre punkter er placeret i forhold til hinanden.

For at bestemme hvorvidt et punkt er placeret inde i en trekant benytter vi os af et trick der forudsætter at vi kan beregne arealet af en trekant ved at kende dens hjørnepunkter. Det viser sig at hvis en trekant er bestemt af punkterne  $p_1 = (x_1, y_1)$ ,  $p_2 = (x_2, y_2)$  og  $p_3 = (x_3, y_3)$  vil følgende relativt simple formel kunne benyttes til at udregne arealet:

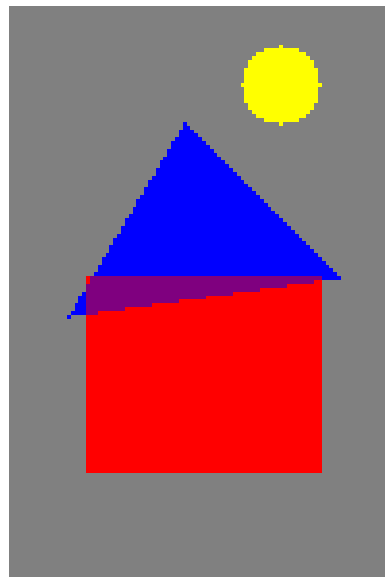
$$area = \left| \frac{x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)}{2} \right|$$

Tricket som vi nu skal benytte til at afgøre om et punkt  $p$  ligger inden i en trekant udspændt af hjørnerne  $p_1$ ,  $p_2$  og  $p_3$  forklares lettest ved at iagttage figuren til højre. Såfremt arealet af de tre trekanter  $(p_1, p_2, p)$ ,  $(p_2, p_3, p)$ , og  $(p_1, p_3, p)$  tilsammen er større end arealet af trekanten  $(p_1, p_2, p_3)$ , da ligger punktet  $p$  udenfor trekanten  $(p_1, p_2, p_3)$ ; ellers ligger punktet indenfor trekanten.



### 3 Opgave(r)

1. Lav en figur `figHouse` : `figure`, som består af en rød firkant udspændt af punkterne  $(20,70)$  og  $(80,120)$ , en blå trekant udspændt af punkterne  $(15,80)$ ,  $(45,30)$  og  $(85,70)$ , samt en gul cirkel med centrum  $(70,20)$  og radius 10.
2. Skriv en F# funktion `triarea2` der kan beregne den dobbelte værdi af arealet af en trekant ud fra dens tre hjørnepunkter ved at benytte formelen ovenfor. Funktionen skal tage hjørnepunkterne som argumenter og have typen `point -> point -> point -> int`. Test funktionen på et par simple trekanter.<sup>1</sup>
3. Udvid funktionen `colourAt` til at håndtere trekantsudvidelsen ved at implementere tricket nævnt ovenfor samt ved at benytte den implementerede funktion `triarea2`.
4. Lav en fil `figHouse.png`, der viser figuren `figHouse` i et  $100 \times 150$  bitmap. Resultatet skulle gerne ligne figuren nedenfor.



5. Udvid funktionerne `checkFigure` og `boundingBox` fra øvelsesopgaverne til at håndtere udvidelsen.  
`boundingBox houseFig` skulle gerne give `((15, 10), (85, 120))`.

<sup>1</sup>Det viser sig at være hensigtsmæssigt at undgå divisionen med 2, som kan forårsage uheldige afrundingsfejl.