

## 1 Codeer Koch's Curve

Vandaag gaan we door middel van **recursieve functies** fractals tekenen! In de vorige *assignment* heb je de functie `kochCurve` geschreven die de figuur ?? tekent. Als je iedere lijn vervangt door deze zelfde figuur krijg je de situatie van figuur ??. Als je dit oneindig vaak herhaald krijg je een **fractal** (zie figuur ??).

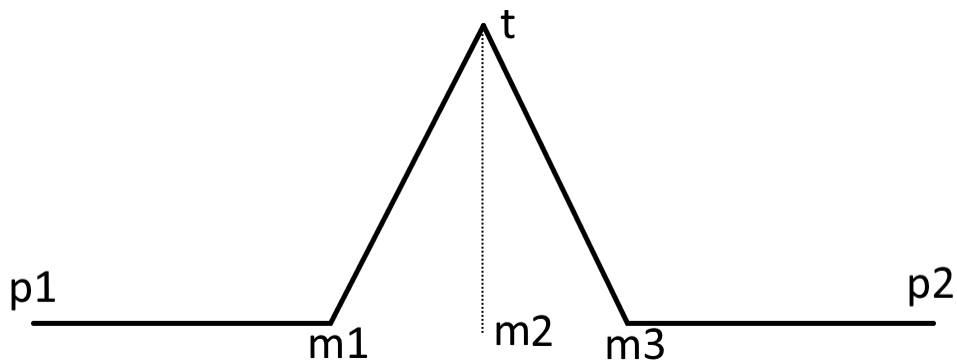


Figure 1: Koch's Curve generatie 1

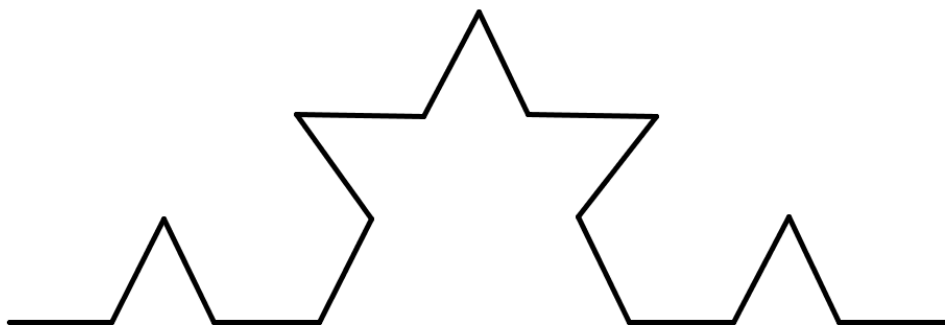


Figure 2: Koch's Curve generatie 2

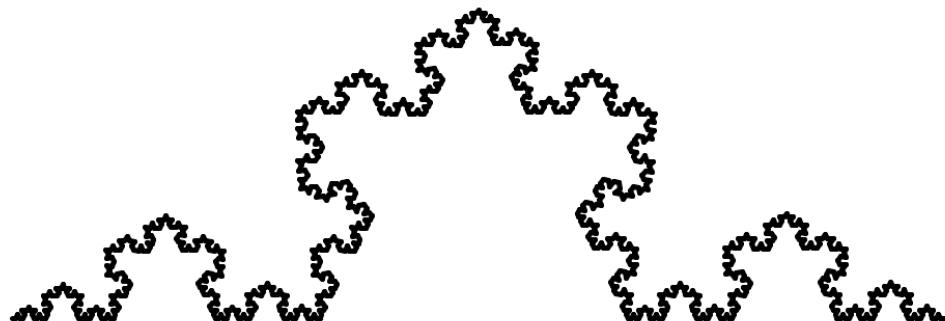


Figure 3: Koch's Curve Fractal (generatie  $\infty$ )

In deze opdracht ga je een functie schrijven die dit proces **n** keer herhaald. Maak gebruik van de al geschreven functie **kochCurve** van de vorige assignment.

```
void kochCurve(int n, PVector p1, PVector p2) {
    if (n == 0) {
        line(p1.x, p1.y, p2.x, p2.y);
    } else {
        //TODO
    }
}
```

## 2 [Bonus] Koch's Snowflake

Door Koch's curve een aantal keer te draaien ontstaat **Koch's Snowflake**.

```
void kochSnowflake(int n, int sides, int r, PVector center) {
}
```

De parameters zijn:

**n** het aantal iteraties van Koch's curve.

**center** het midden van de snowflake.

**sides** het aantal zijden van de snowflake.

**r** de radius van de snowflake.

### Tip

Gebruik de code die je hebt geschreven voor opdracht 2 polygon!

De volgende code geeft als resultaat figuur ??.

```
snowflake(2,5,50, new PVector(100,200));
snowflake(5,6,100,new PVector(300,200));
snowflake(1,6,30, new PVector(500,200));
snowflake(3,3,100,new PVector(700,200));
```

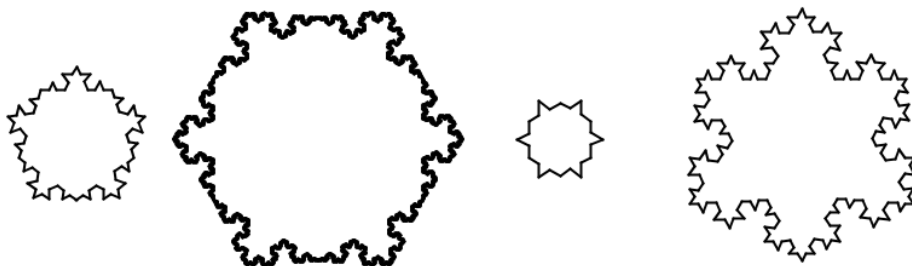


Figure 4: Koch snowflakes!