

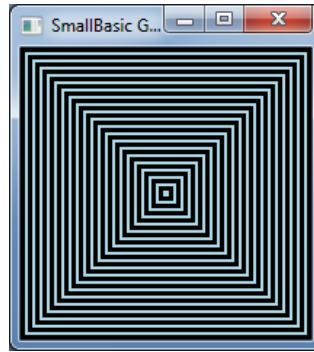
Plezier met vormen

In dit hoofdstuk gaan we plezier beleven aan alles wat we tot nu toe hebben geleerd. Dit hoofdstuk bevat voorbeelden waarin alles wat we tot nu toe hebben geleerd wordt gecombineerd voor het maken van enkele leuke programma's.

Rechthoeken in overvloed

Hier tekenen we in een lus meerdere rechthoeken met toenemende afmetingen.

```
GraphicsWindow.BackgroundColor = "Black"  
GraphicsWindow.PenColor = "LightBlue"  
GraphicsWindow.Width = 200  
GraphicsWindow.Height = 200  
  
For i = 1 To 100 Step 5  
    GraphicsWindow.DrawRectangle(100 - i, 100 - i, i * 2, i * 2)  
EndFor
```

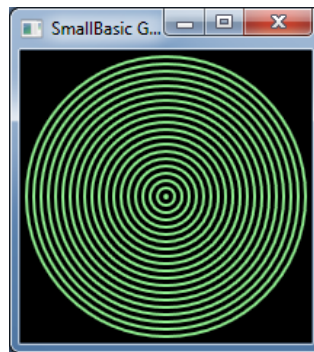


Afbeelding 33 – Rechthoeken in overvloed

Spectaculaire cirkels

Met een variant op het vorige programma teken we cirkels in plaats van vierkanten.

```
GraphicsWindow.BackgroundColor = "Black"  
GraphicsWindow.PenColor = "LightGreen"  
GraphicsWindow.Width = 200  
GraphicsWindow.Height = 200  
  
For i = 1 To 100 Step 5  
    GraphicsWindow.DrawEllipse(100 - i, 100 - i, i * 2, i * 2)  
EndFor
```

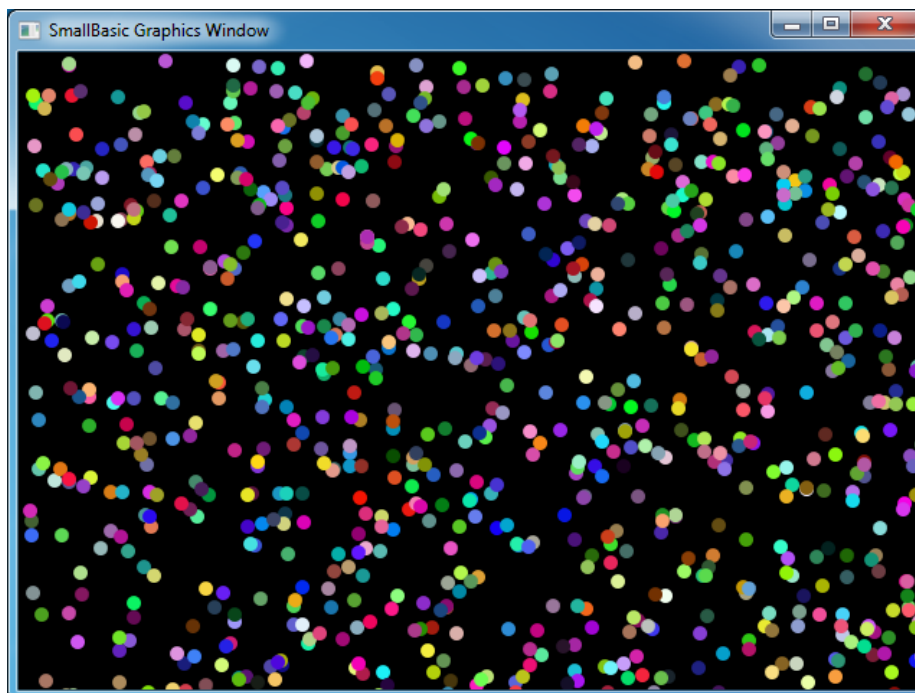


Afbeelding 34 – Spectaculaire cirkels

Willekeurig maken

Met dit programma gebruiken we de bewerking `GraphicsWindow.GetRandomColor` voor het instellen van willekeurige kleuren voor de kwast en vervolgens stellen we de x- en y-coördinaten voor de cirkels in met `Math.GetRandomNumber`. Deze twee bewerkingen kunnen op interessante manieren worden gecombineerd om interessante programma's te maken met verschillende resultaten telkens als ze worden uitgevoerd.

```
GraphicsWindow.BackgroundColor = "Black"  
For i = 1 To 1000  
    GraphicsWindow.BrushColor = GraphicsWindow.GetRandomColor()  
    x = Math.GetRandomNumber(640)  
    y = Math.GetRandomNumber(480)  
    GraphicsWindow.FillEllipse(x, y, 10, 10)  
EndFor
```



Afbeelding 35 – Willekeurig maken

Fractalen

In het volgende programma tekenen we een eenvoudige driehoekfractal met willekeurige getallen. Een fractal is een geometrische vorm die kan worden opgedeeld in delen, waarbij elk deel precies lijkt op de oorspronkelijke vorm. In dit geval worden er honderden driehoeken getekend die lijken op de oorspronkelijke driehoek. En aangezien dit programma een aantal seconden duurt, kun je zien hoe de driehoeken langzaam worden gevormd van puntjes. De logica hier achter is enigszins moeilijk te beschrijven en daarom laat ik je dit ontdekken in een oefening.

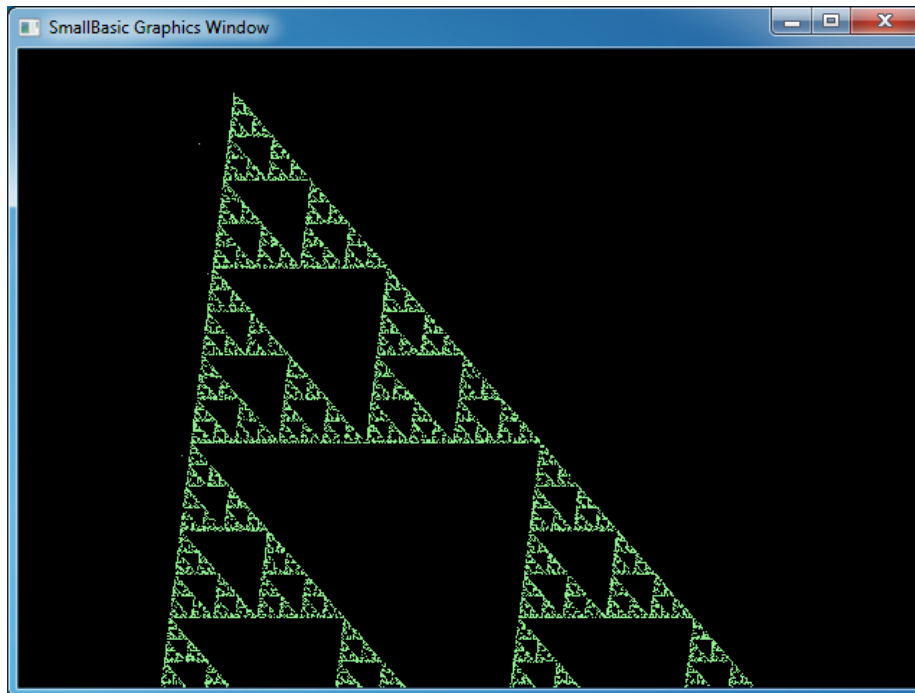
```
GraphicsWindow.BackgroundColor = "Black"
x = 100
y = 100

For i = 1 To 100000
  r = Math.GetRandomNumber(3)
  ux = 150
  uy = 30
  If (r = 1) then
    ux = 30
    uy = 1000
  EndIf

  If (r = 2) Then
    ux = 1000
    uy = 1000
  EndIf

  x = (x + ux) / 2
  y = (y + uy) / 2

  GraphicsWindow.SetPixel(x, y, "LightGreen")
EndFor
```



Afbeelding 36 – Een driehoekfractal

Als je goed wilt zien hoe de fractal langzaam uit puntjes wordt gevormd, kun je de lus vertragen met de bewerking **Program.Delay**. Voor deze bewerking moet je een getal opgeven dat in milliseconden specificeert hoelang de vertraging is. Hier volgt het aangepaste programma, met de gewijzigde regel vetgedrukt.

```
GraphicsWindow.BackgroundColor = "Black"  
x = 100  
y = 100  
  
For i = 1 To 100000  
    r = Math.GetRandomNumber(3)  
    ux = 150  
    uy = 30  
    If (r = 1) then  
        ux = 30  
        uy = 1000  
    EndIf  
  
    If (r = 2) Then  
        ux = 1000  
        uy = 1000  
    EndIf
```

```
x = (x + ux) / 2
y = (y + uy) / 2

GraphicsWindow.SetPixel(x, y, "LightGreen")
Program.Delay(2)
EndFor
```

Het programma wordt langzamer uitgevoerd naarmate je de vertraging vergroot. Experimenteer met de getallen om erachter te komen wat de beste vertraging voor jou is.

Een andere wijziging aan dit programma is de vervanging van de volgende regel:

```
GraphicsWindow.SetPixel(x, y, "LightGreen")
```

met

```
kleur = GraphicsWindow.GetRandomColor()
GraphicsWindow.SetPixel(x, y, kleur)
```

Met deze wijziging worden de pixels van de driehoek met willekeurige kleuren getekend.