

# Computer Graphics: 3D Fractalen

Ruben Mennes  
Robin Verschoren

## 3D Fractalen (0.75 punten)

Voeg ondersteuning aan je grafische engine toe voor het genereren van 3D Fractalen op basis van platonische lichamen. Meer bepaald moeten er 3D Fractalen kunnen worden gegenereerd op basis van de volgende ruimtelijke figuren:

- Kubus (Cube)
- Dodecahedron
- Icosahedron
- Octahedron
- Tetrahedron

Merk op dat het genereren van deze 3D Fractalen onafhankelijk is van de manier waarop de uiteindelijke ruimtelijke figuur wordt weergegeven. Je engine moet dus in staat zijn om deze 3D Fractalen weer te geven volgens alle manieren die door je engine ondersteund worden. (Wireframe, Z-Buffering met lijnen, Z-Buffering met driehoeken, ...)

## Invoerformaat

Het invoer formaat voor deze opdracht is een uitbreiding van het invoer formaat dat voor de vorige drie opdrachten werd gespecificeerd. **type** veld in de **General** sectie kan de waarden ‘*Wireframe*’, ‘*ZBufferedWireframe*’ en ‘*ZBuffering*’ aannemen afhankelijk van de manier waarop de fractaal moet worden weergegeven.

Elke 3D Fractaal wordt als een nieuw type figuur beschouwd. Het **type** veld van deze figuren is afhankelijk van het platonische lichaam dat moet worden gebruikt om de fractaal te genereren. De mogelijke waarden zijn ‘*FractalCube*’, ‘*FractalDodecahedron*’, ‘*FractalIcosahedron*’, ‘*FractalOctahedron*’ en ‘*FractalTetrahedron*’. De **Figure\*** sectie om een 3D Fractaal te beschrijven bevat, naast de standaard velden die in de vorige opgaves werden gespecificeerd, ook de onderstaande velden:

- **fractalScale** (double): De schaal factor die moet worden gebruikt bij het vervangen van de figuur door de kleinere figuren. Een schaal factor van 3, bijvoorbeeld, betekent dat bij elke stap de nieuwe figuren 3 keer kleiner moeten zijn dan het origineel.
- **nrIterations** (integer): Het aantal iteraties dat moet worden uitgevoerd bij het genereren van de 3D-Fractaal. Als dit veld gelijk is aan 0 moet de oorspronkelijke figuur worden weergegeven. Als dit veld gelijk is aan 1 moeten alle punten één keer worden vervangen, bij een waarde van 2 twee keer, etc.

Ter verduidelijking wordt hieronder een voorbeeld gegeven:

```
[General]
type = "Wireframe"
size = 1000
eye = (100, 50.0, 75.0);
backgroundcolor = (0.0, 0.0, 0.0)
color = (0.0, 1.0, 0.0)
nrFigures = 1
```

```
[Figure0]
type = "FractalCube"
scale = 1
rotateX = 0
rotateY = 0
rotateZ = 0
center = (0, 0, 0)
color = (1, 0, 0)
fractalScale = 3
nrIterations = 1
```

Hou er rekening mee dat er in één input bestand zowel 3D-Fractalen als gewone figuren kunnen voorkomen.

## BuckyBall (0.125 punten)

Implementeer een procedure om een Buckyball te genereren. zorg er daarna voor dat je engine ook een 3D Fractaal op basis van deze ruimtelijke figuur kan genereren. De procedure om een Buckyball te genereren op basis van een Icosahedron staat beschreven op pagina 46 van de cursus. Voor het genereren van de 3D Fractaal op basis van een Bucky ball kan dezelfde procedure worden gevolgd als voor de platonische lichamen.

### Invoerformaat

Voor de BuckyBall is het **type** veld in de **Figure** sectie gelijk aan *'BuckyBall'*. Voor de rest worden alle velden overgenomen die voor de platonische lichamen gedefinieerd zijn. Een voorbeeld wordt hieronder gegeven:

```
[Figure0]
type = "BuckyBall"
scale = 1
rotateX = 0
rotateY = 0
rotateZ = 0
center = (0, 0, 0)
color = (1, 1, 1)
```

Voor de *'BuckyBall Fractaal'* is het **type** veld in de **Figure** sectie gelijk aan *'FractalBuckyBall'*. De overige velden voor de *'FractalBuckyBall'* zijn dezelfde als voor alle andere 3D Fractalen. Een voorbeeld wordt hieronder gegeven:

```
[Figure0]
type = "FractalBuckyBall"
scale = 1
```

```

rotateX = 0
rotateY = 0
rotateZ = 0
center = (0, 0, 0)
color = (1, 0, 0)
fractalScale = 3
nrIterations = 1

```

## Menger Spons (0.125 punten)

Implementeer een procedure om een Menger spons te genereren op basis van een Kubus (Cube). Het aantal keer dat elke kubus door kleinere kubussen wordt vervangen moet instelbaar zijn en de Menger spons moet (net zoals de oorspronkelijke kubus) rond de oorsprong gegenereerd worden. Een 'Fractaal versie' van de Menger spons moet *niet* worden ondersteund vermits de Menger spons op zich al een fractaal is. Een mogelijke extra optimalisatie is het weghalen van de ribben van de binnenin gelegen kleinere kubussen. Meer informatie over het genereren van de Menger spons kun je vinden op volgende site: [http://en.wikipedia.org/wiki/Menger\\_sponge](http://en.wikipedia.org/wiki/Menger_sponge)

### Invoerformaat

Het invoerformaat voor de Menger spons is gebaseerd op het formaat voor 3D Fractalen. Het **type** veld van de figuur is voor de Menger Spons altijd gelijk aan '*MengerSponge*'. Daarnaast worden ook de **scale**, **rotateX-Z**, **center**, **color**, en **nrIterations** velden overgenomen. Het **fractalScale** veld komt *niet* voor!! Een voorbeeld wordt hieronder gegeven:

```

[Figure0]
type = "MengerSponge"
nrIterations = 1
scale = 1
rotateX = 0
rotateY = 0
rotateZ = 0
center = (0, 0, 0)
color = (1, 1, 0)

```

### Tips

- Het is niet nodig om het genereren van een 3D-fractaal voor elk platonisch lichaam apart te implementeren. Voor het genereren van een 3D-fractaal worden er namelijk enkel schalingen en verschuivingen uitgevoerd op een aantal kopieën van de oorspronkelijke figuur. Als je engine in staat is om een gegenereerd lichaam te kopiëren (als de copy-constructor is geïmplementeerd) kan de fractaal dus berekend worden onafhankelijk van de oorspronkelijke figuur.
- Op blackboard zijn er een aantal voorbeeld configuratiebestanden te vinden samen met de verwachte output. Deze kun je gebruiken om je engine om de correctheid van je engine te verifiëren. Daarnaast kun je ook de voorbeelden die in de cursus worden gegeven implementeren.