

# Computer Graphics

Benny Van Houdt

Dept. Mathematics and Computer Science  
University of Antwerp

February 14, 2013



Universiteit  
Antwerpen

# Computer Graphics Onderdelen

## Theorielessen: 12x2 uur op rooster

- DI 10h45-12h45: week 1-9 en 11-13 (G010)
- Uitgewerkte cursus beschikbaar (70 pagina's)

## Practica: 2 Groepen, elk 12x2 uur

- Group A: DI 13h45-15h45 (G025)
- Group B: VR 13h45-15h45 (G025)
- Op rooster tot 17h, zodat je kan verder werken



Universiteit  
Antwerpen

# Computer Graphics Examenvorm

## Theorielessen

- Schriftelijk examen: 10 punten (of minder)
- Gesloten boek
- Eén of meer vragen per hoofdstuk

## Practica

- 10-tal opdrachten
- 6 punten t/m Z-buffering met driehoeken = MUST DO
- totaal 11 punten voor zaken uit de cursus
- 1 punt extra als meer doet dan in de theorie
- Geen verplichte tussentijdse evaluatie (wel aangeraden)



Universiteit  
Antwerpen

# Computer Graphics Best Student Award

Wat brengt het op?



- Certificaat
- ACM student membership gedurende een jaar (bevat ACM communications magazine)
- Eeuwige vermelding op website:

<http://win.ua.ac.be/vanhoudt/>



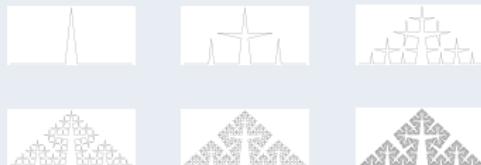
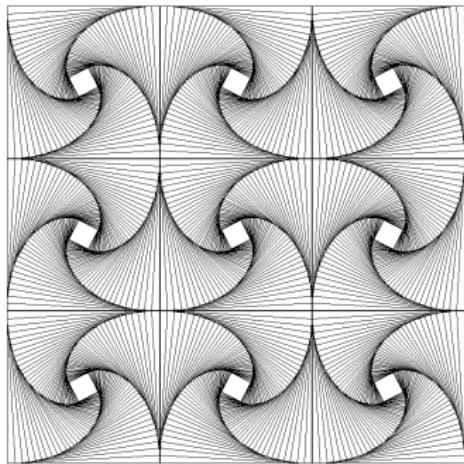
Universiteit  
Antwerpen

## Hoofdstukken en opdrachten

- 2D L-Systemen (1,25)
- 3D-lijntekening + L-systemen (1,25)
- Platonische lichamen (0,75)
- Z-Buffering met lijnen (1)
- Z-Buffering met driehoeken (1,75)
- 3D-fractalen (1)
- Belichting (1,5)
- Schaduw (1)
- Texture mapping (0,75)
- 3D-lijntekeningen met cylinders (0,75)



## Hoofdstuk 1: 2D-Lijntekeningen



Universiteit  
Antwerpen

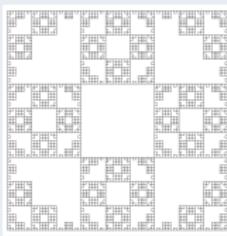
# Computer Graphics Inhoud

## Hoofdstuk 1: L-systemen

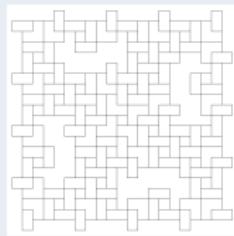
$n=4, F \rightarrow FF-F-F-F-F-F+F$



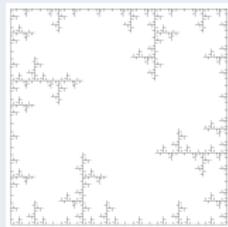
$n=4, F \rightarrow FF-F-F-F-F-FF$



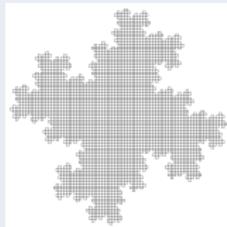
$n=3, F \rightarrow FF-F+F-F-FF$



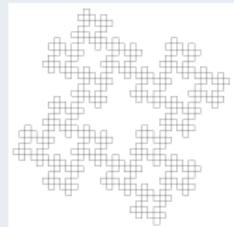
$n=4, F \rightarrow FF-F--F-F$



$n=5, F \rightarrow F-FF--F-F$



$n=4, F \rightarrow F-F+F-F-F$



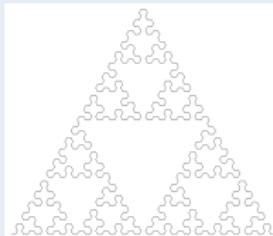
Universiteit  
Antwerpen

## Hoofdstuk 1: L-systemen

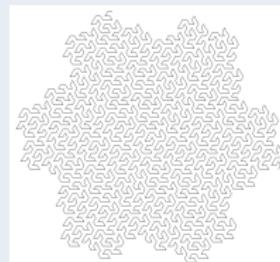
$n=10, I=F, \delta=\pi/2,$   
 $F \rightarrow F+G+, G \rightarrow -F-G$



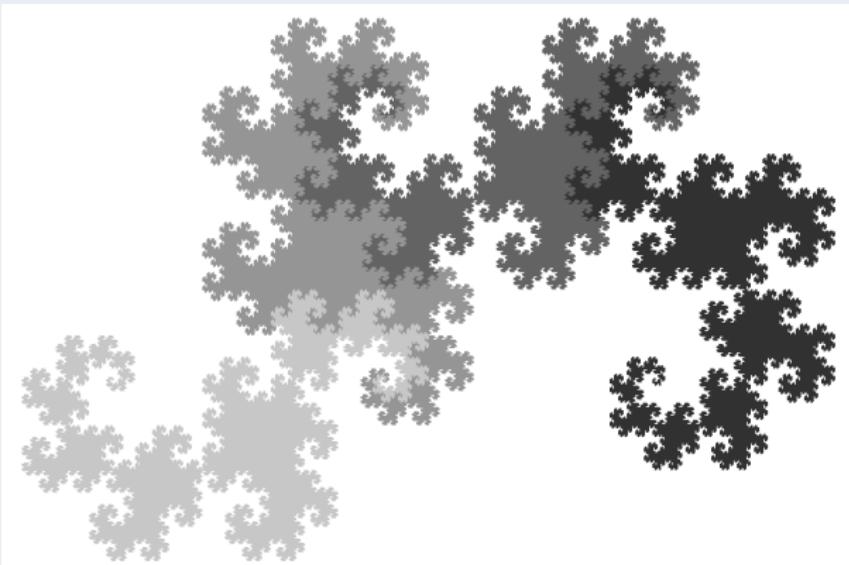
$n=6, I=F, \delta=\pi/3,$   
 $F \rightarrow G-F-G, G \rightarrow F+G+F$



$n=4, I=F, \delta=\pi/3,$   
 $F \rightarrow F+G++G-F--FF-G+,$   
 $G \rightarrow -F+GG++G+F--F-G$



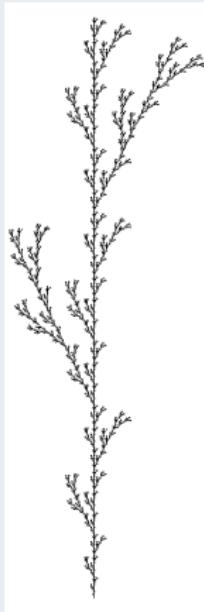
## Hoofdstuk 1: L-systemen



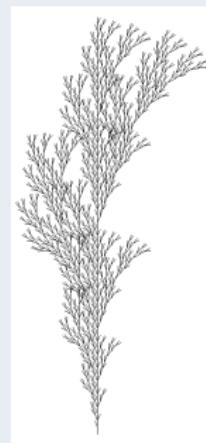
Universiteit  
Antwerpen

## Hoofdstuk 1: L-systemen met haakjes

$n=5, \delta=2\pi (25.7/360),$   
 $F \rightarrow F[+F]F[-F]F$



$n=5, \delta=2\pi (20/360),$   
 $F \rightarrow F[+F]F[-F][F]$



$n=4, \delta=2\pi (22.5/360),$   
 $F \rightarrow FF[-F+F+F]+[+F-F-F]$



$n=6, \delta=2\pi (25/360), l=+X,$   
 $F \rightarrow FF, X \rightarrow F-[lX]+X+F+[FX]-X$



## Hoofdstuk 2: Wireframe tekeningen in 3D (zwart-wit)



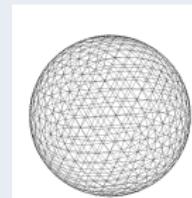
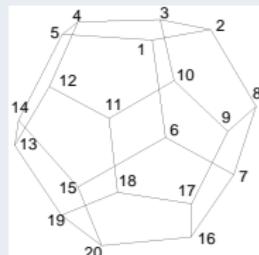
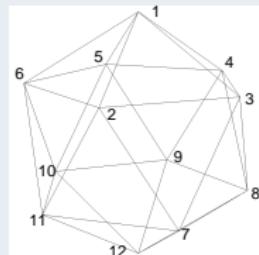
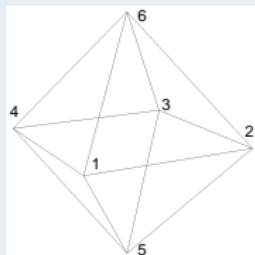
Nodige theorie:

- 2D- en 3D-rotatie matrices
- Transformatie naar Eye-coördinaatsysteem
- Perspectief projectie



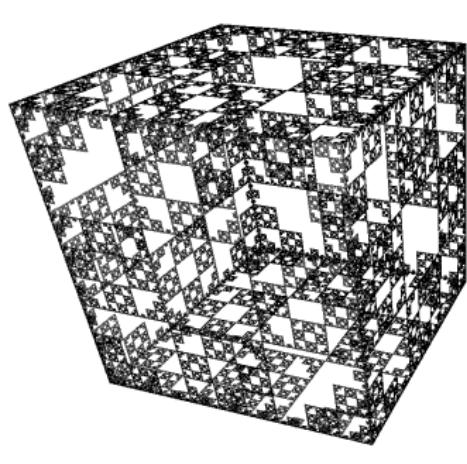
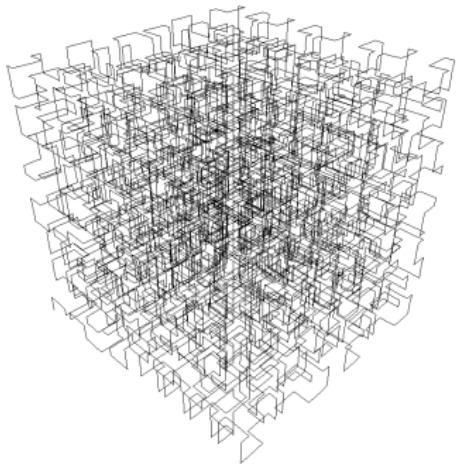
Universiteit  
Antwerpen

## Hoofdstuk 2: Platonische Lichamen en de Bol



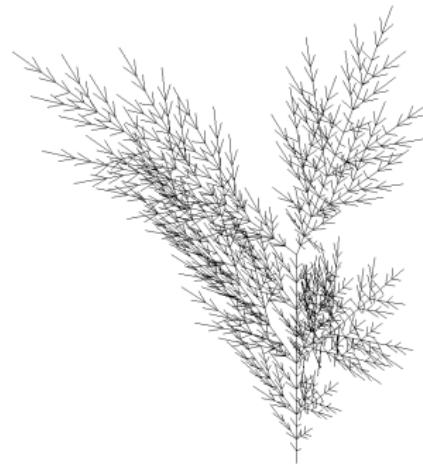
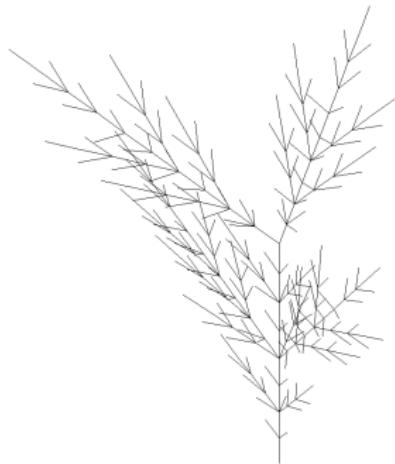
Universiteit  
Antwerpen

## Hoofdstuk 2: 3D L-systemen



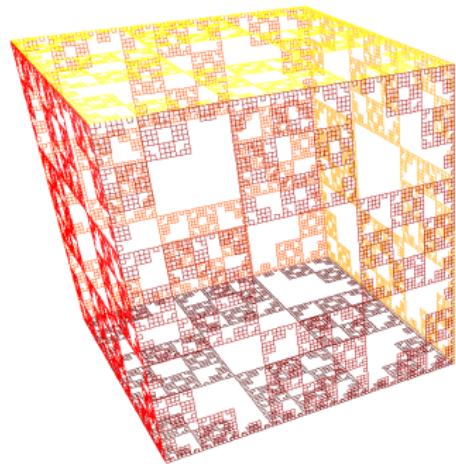
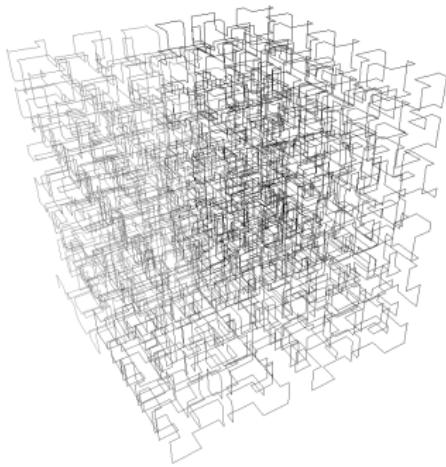
Universiteit  
Antwerpen

## Hoofdstuk 2: 3D L-systemen met haakjes



Universiteit  
Antwerpen

## Hoofdstuk 3: Z-buffering, Wireframes in kleur

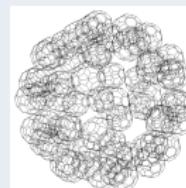
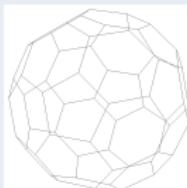
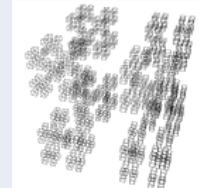
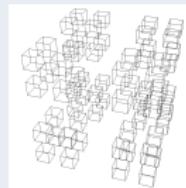
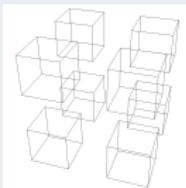
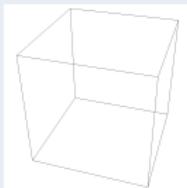


Nodige theorie: Z-buffering met lijnen en 1/z-interpolatie



Universiteit  
Antwerpen

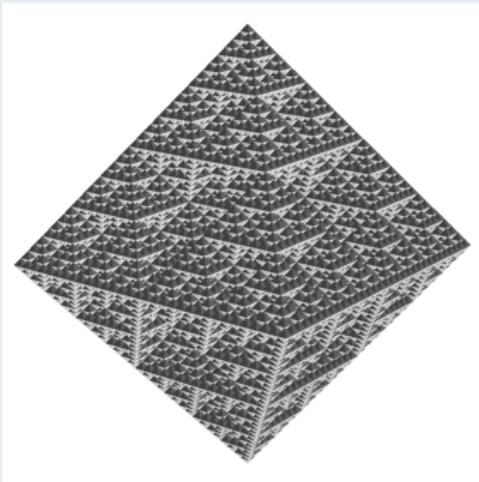
## Hoofdstuk 3: Z-buffering, 3D-fractalen



Nodige theorie: Inkleuren van een driehoek en Z-buffering met driehoeken

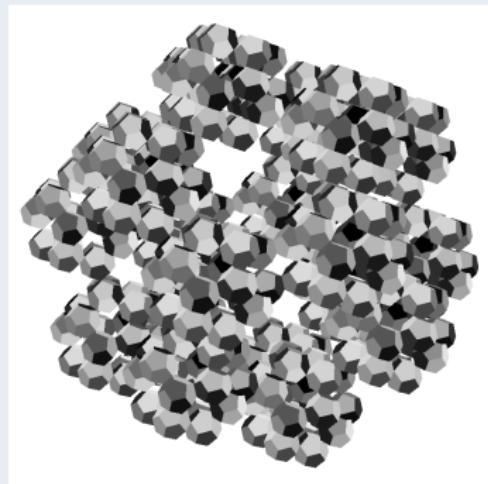
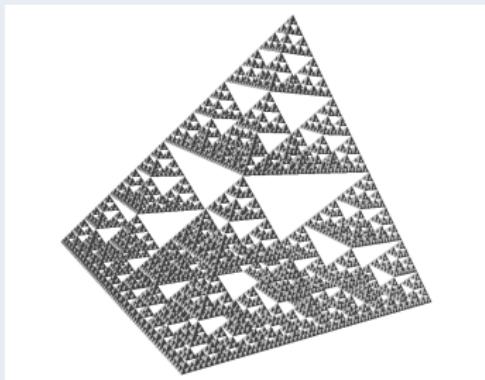


## Hoofdstuk 3: Z-buffering, 3D-fractalen



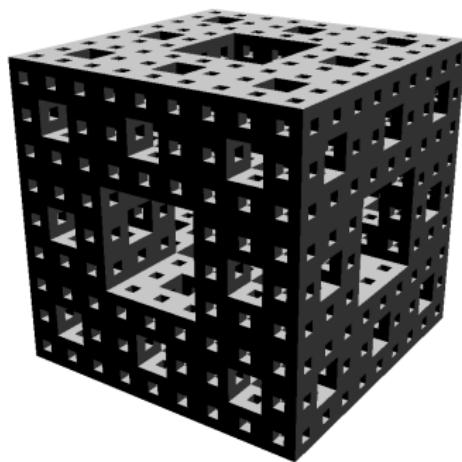
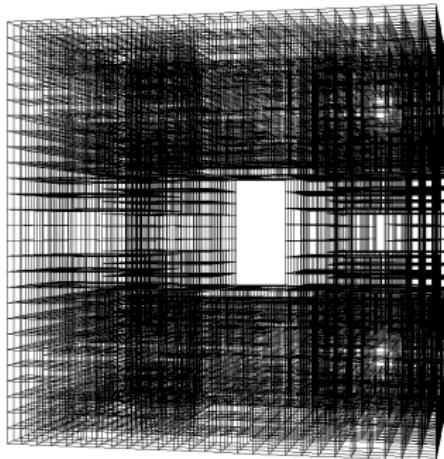
Universiteit  
Antwerpen

## Hoofdstuk 3: Z-buffering, 3D-fractalen



Universiteit  
Antwerpen

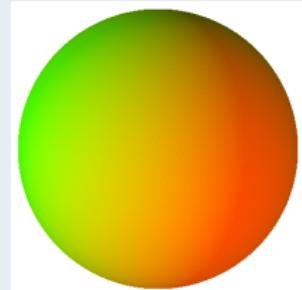
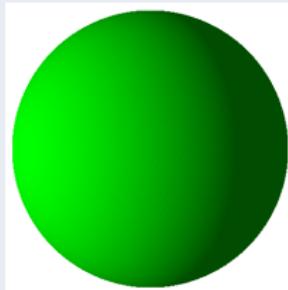
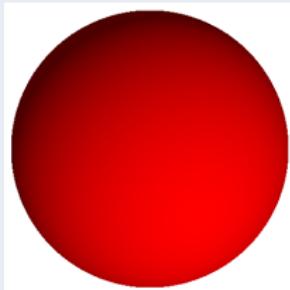
## Hoofdstuk 3: Z-buffering, 3D-fractalen



Universiteit  
Antwerpen

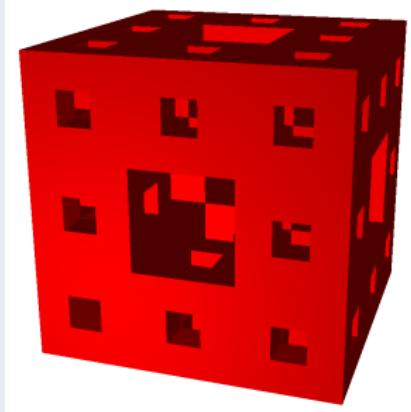
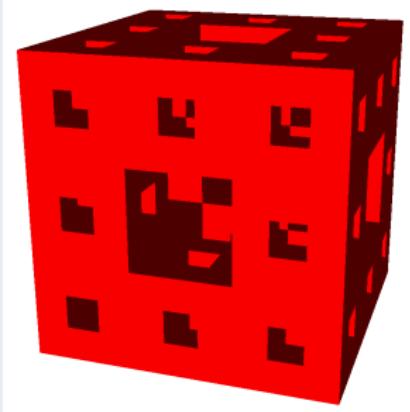
## Hoofdstuk 4: Ambient en Diffuus licht (op oneindig)

Nodige theorie: RGB model



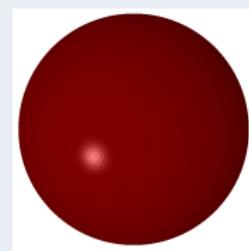
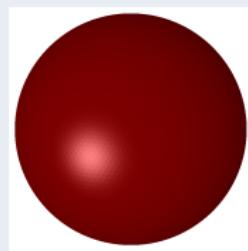
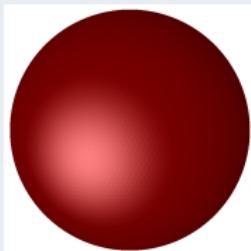
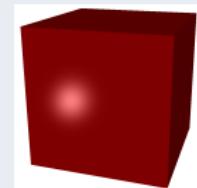
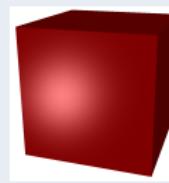
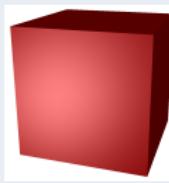
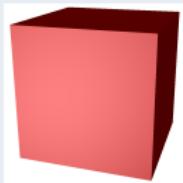
Universiteit  
Antwerpen

## Hoofdstuk 4: Diffuus licht (op oneindig vs. puntbron)



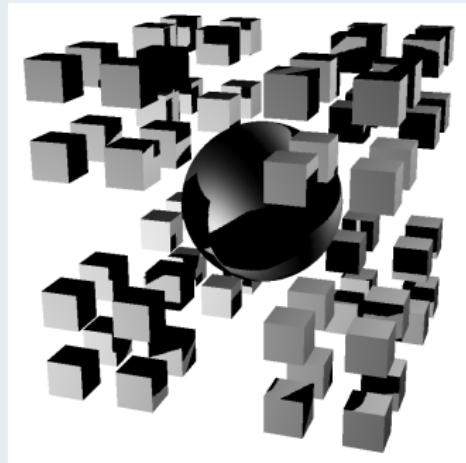
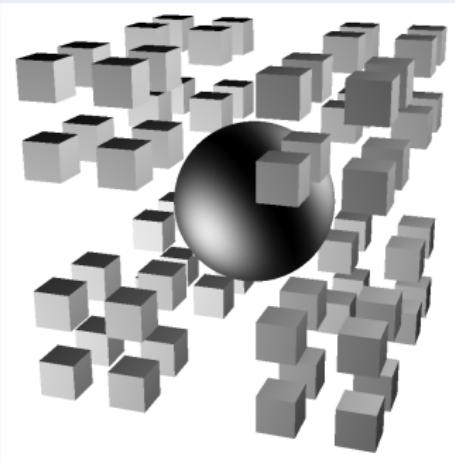
Universiteit  
Antwerpen

## Hoofdstuk 4: Spiegelend/Specular licht



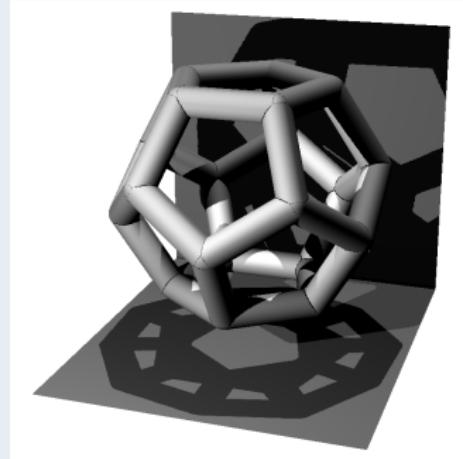
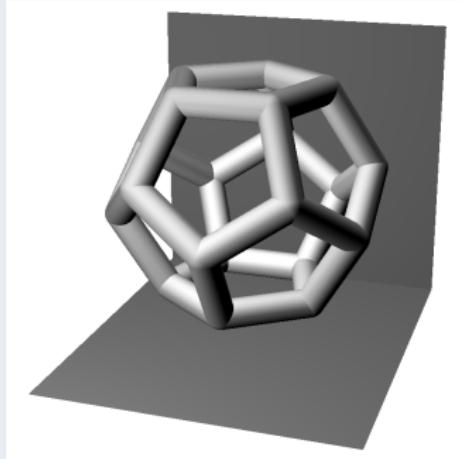
Universiteit  
Antwerpen

## Hoofdstuk 4: Schaduw en Z-buffering



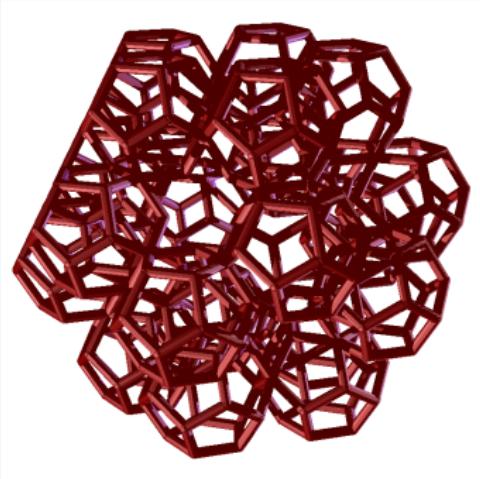
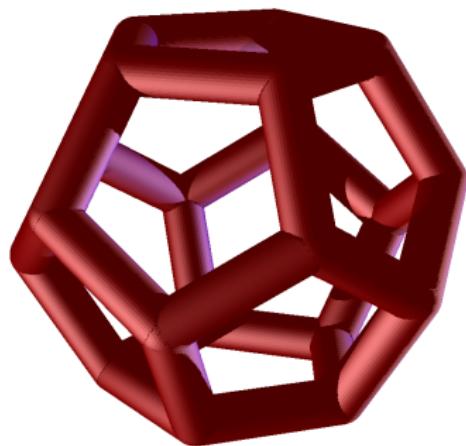
Universiteit  
Antwerpen

## Hoofdstuk 4: Schaduw en Z-buffering



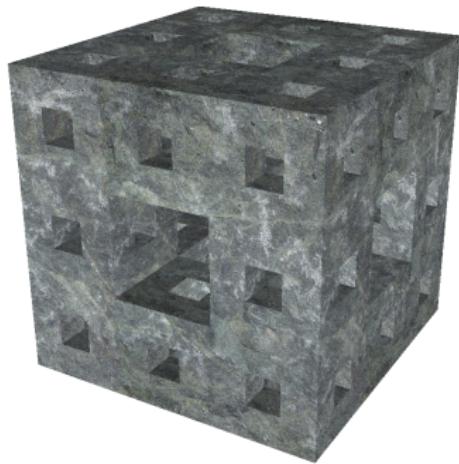
Universiteit  
Antwerpen

## Hoofdstuk 4: Schaduw en Z-buffering



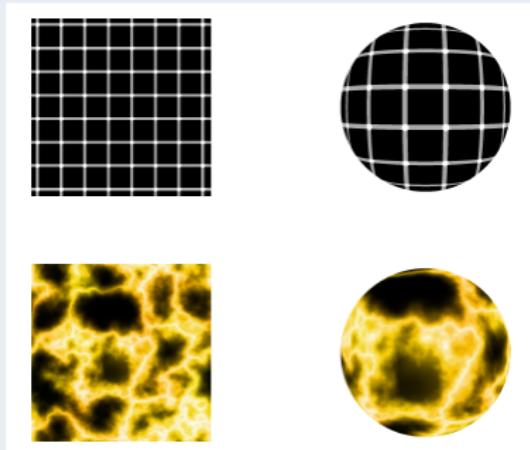
Universiteit  
Antwerpen

## Hoofdstuk 4: Texture Mapping op een vlak



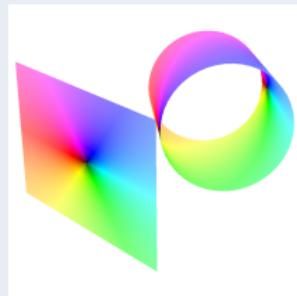
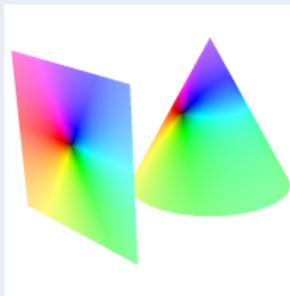
Universiteit  
Antwerpen

## Hoofdstuk 4: Texture Mapping op een bol



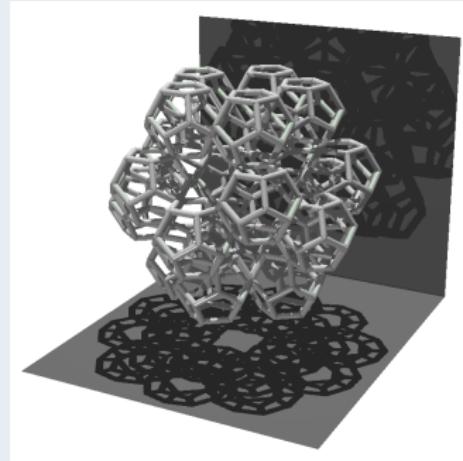
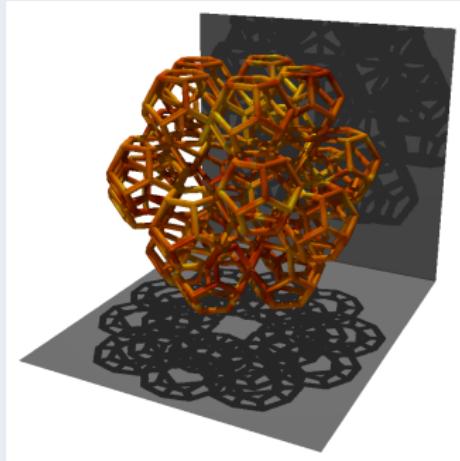
Universiteit  
Antwerpen

## Hoofdstuk 4: Texture Mapping op willekeurig oppervlak



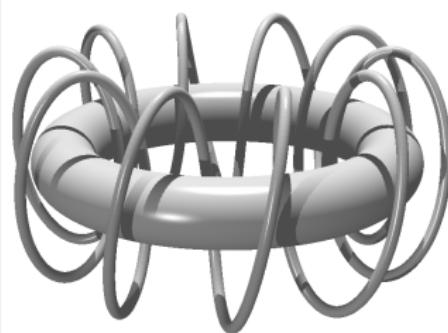
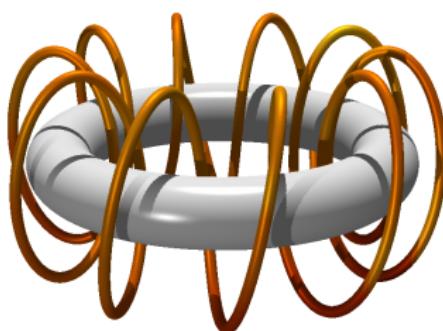
Universiteit  
Antwerpen

## Alles samen



Universiteit  
Antwerpen

## Alles samen



Universiteit  
Antwerpen

## Hoofdstuk 1

- 3 Gegeven enkele 2D-afbeeldingen (waarvan sommige stapsgewijs worden opgebouwd), geef aan hoe je deze kan genereren met behulp van een L-systeem. Geef ook een woordje uitleg bij je oplossing.
- 1 Hoeveel pixels gebruik je om een punt  $(x_A, y_A)$  te verbinden met  $(x_B, y_B)$ ?
- 1 Hoe werkt een stochastic L-systeem en wat is hiervan het nut?
- 1 Wat is het idee van een L-systeem met haakjes en hoe kan je dit efficient implementeren?



## Hoofdstuk 2

- 3 Stel de rotatiematrix in 2D op gebruik makend van een tekening (Sectie 6.2).
- 1 Geef aan hoe je de 3D rotatiematrices kan uitdrukken mbv de 2D rotatiematrix.
- 1 Wat is een translatie? Hoe kan je deze in matrixvorm schrijven, waarom is dit laatste nuttig?
- 2 Leg uit wat we bedoelen met bolcoördinaten, dat is, hoe drukken we de Cartesische coördinaten uit in bolcoördinaten? Hoe kan je Cartesische coördinaten omzetten in bolcoördinaten?
- 1 Waarom stappen we over op het Eye-coördinaatsysteem? Kunnen we niet onmiddellijk een projectie uitvoeren op ons scherm?



Universiteit  
Antwerpen

## Hoofdstuk 2

- 3 Wat bedoelen we met het Eye-coördinaatsysteem? Leg uit hoe je de  $V$  matrix bekomt die we gebruiken om de coördinaten van een punt  $(x, y, z)$  om te zetten in zijn Eye-coördinaten  $(x_E, y_E, z_E)$  (hoe doe je dit laatste?).
- 2 Leg uit (met behulp van een tekening) hoe je de perspectiefprojectie kan uitvoeren (Sectie 7.2).
- 2 Hoe kan je de input file voor een bol aanmaken? Uit hoeveel hoekpunten en driehoeken bestaat je bol als je deze procedure  $n$  keer uitvoert?
- 2 Hoe werkt een 3D L-systeem (leg de nadruk op de verschillen met een 2D systeem)?
- 2 Leg uit (met behulp van een tekening) hoe de  $H$ ,  $L$  en  $U$  vectoren moet aanpassen als gevolg van een  $+$ ,  $\wedge$  of  $\backslash$ ?

## Hoofdstuk 3

- 2 Leg de algemene werking van Z-buffering uit. Werk dit uit voor een 3D-lijntekening en geef aan hoe je de  $1/z$ -interpolatie hierbij gebruikt.
- 3 Maak gebruik van een tekening om de  $1/z$ -interpolatie uit te leggen (Sectie 13).
- 3 Bespreek een efficiënt algoritme om alle pixels behorende tot een driehoek te bepalen (Sectie 14).
- 1 Hoeveel pixels hanteren we indien de drie hoekpunten van een driehoek op een enkele lijn worden geprojecteerd? Hoe herkennen we deze situatie?



Universiteit  
Antwerpen

## Hoofdstuk 3

- 2 Hoe werkt het Z-buffer algoritme op basis van driehoeken?  
Bespreek de formule voor het bepalen van de diepte in detail.
- 2 Leg uit (op basis van een tekening) hoe je  $dz/dx$  en  $dz/dy$  kan uitdrukken op basis van de pixel coörindaten  $(x', y')$  en de vergelijking van een (welk?) vlak (deel 1, Sectie 17).
- 2 Hoe bereken je  $dz/dx$  en  $dz/dy$  precies (deel 2, Sectie 17)?
- 1 Geef aan hoe je eenvoudig 3D fractalen kan aanmaken op basis van de Platonische lichamen. Kan dit ook voor andere lichamen?



Universiteit  
Antwerpen

## Hoofdstuk 4

- 1 Bespreek kort het RGB model (RGB + ambient licht Sectie).
- 2 Welke vormen van licht hebben we allemaal bekeken en wat is kenmerkend voor elk type (ivm positie van lichtbron, object, camera, snelheid van berekening).
- 1 Waarom is de volgorde van de hoekpunten van belang bij het opsommen hiervan?
- 2 Hoe bepalen we de diffuse lichtcomponent van een object, indien we diffuus licht op oneindig hanteren?
- 2 Hoe bepalen we de diffuse lichtcomponent van een object, indien we diffuse puntbron hanteren? Waarom gebruiken we niet steeds een puntbron (is het resultaat niet gelijk voor een bron die erg ver staat)?



## Hoofdstuk 4

- 2 Hoe bepalen we de speculaire lichtcomponent van een object (werk de berekening van de reflectie richting  $r$  eveneens uit)?
- 2 Hoe kunnen we schaduw toevoegen indien we Z-buffering hanteren? Bespreek de algemene werking. Welke problemen kunnen zich voordoen (en wanneer treden deze meestal op) en hoe kunnen we deze aanpakken?
- 2 Bespreek de verschillende stappen die je moet ondernemen op de geprojecteerde coördinaten  $(x'_L, y'_L)$  te berekenen wanneer we schaduw wensen toe te voegen (waarom is  $V$  inverteerbaar?).
- 2 Waarom gebruiken we een interpolatie om de diepte te bepalen waarmee we  $1/z_L$  wensen te vergelijken? Leg in detail uit hoe je deze berekent.



## Hoofdstuk 4

- 2 Leg het algemene idee achter texture mapping uit, wat weet je te melden ivm de resolutie van de gebruikte afbeelding.
- 2 Werk texture mapping uit voor een vlak oppervlak.
- 2 Geef een mogelijke oplossing voor het mappen van een textuur op een bol, waarom is deze beter dan de methode voor een willekeurig oppervlak. Kunnen we deze ook gebruiken voor bewegende beelden?
- 2 Hoe kunnen we de methode van het mappen van texturen op een vlak oppervlak gebruiken om willekeurige objecten te bekleden. Bespreek de berekening voor het bekomen van de texel nummer in detail.



Universiteit  
Antwerpen