

# Computer Graphics: 3D Lijntekeningen

Bart Sas  
Daniel van den Akker

## 3D Lijntekeningen (1 punt)

1. Implementeer functies die de transformatiematrices genereren voor de schaling, rotatie om de  $x$ -as, rotatie om de  $y$ -as, rotatie om de  $z$ -as en translatie op basis van respectievelijk de schaalfactor, de hoek waarover geroteerd moet worden en de translatievector.

De schaling is een transformatie die niet in de cursus wordt beschreven maar wél moet worden geïmplementeerd. Een schaling zal een 3D Figuur vergroten of verkleinen afhankelijk van een schaalfactor *scale*. Hierbij wordt voor de eenvoud aangenomen dat de 3D figuur rond de oorsprong van het assenstelsel gecentreerd is. De transformatiematrix voor deze transformatie wordt hieronder beschreven.

$$S(scale) = \begin{pmatrix} scale & 0 & 0 & 0 \\ 0 & scale & 0 & 0 \\ 0 & 0 & scale & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2. Implementeer een functie die de Eye-pointtransformatiematrix genereert op basis van de Eye-point coördinaten.
3. Implementeer een procedure die een punt projecteert volgens de perspectiefprojectie. Hou er rekening mee dat het punt eerst moet worden omgezet naar het Eye-coördinaatsysteem alvorens de projectie kan worden uitgevoerd.
4. Breid, op basis van deze functionaliteit, je engine uit zodat deze 3D lijntekeningen ondersteunt.

Voor deze opgave zul je gebruik moeten maken van matrices en vectoren om transformaties en punten voor te stellen. Het is echter niet nodig om deze zélf een Matrix Library te implementeren. Op Blackboard is in de **vector.tar.gz** archive een Matrix Library voorzien. Je bent echter niet verplicht deze te gebruiken, je mag ook gebruik maken van een zelfgeschreven library. Externe libraries mogen, zonder onze toestemming, *niet* worden gebruikt.

## Invoerformaat

De **General** sectie bevat in het geval van een 3D lijntekening de volgende entries:

- **type** (string): Dit veld (dat in alle configuratiebestanden aanwezig is) bevat in het geval van een 3D lijntekening de waarde 'Wireframe'.
- **size** (integer): De grootte van de afbeelding in de richting die de meeste pixels vereist.
- **eye** (tuple van 3 doubles): Dit veld bevat een tuple van 3 reële waarden die de cartesische coördinaten van het eye aangeven.

- **backgroundcolor** (tuple van 3 doubles): De RGB waarden van de achtergrond voorstellen. De waarden liggen tussen 0 en 1.0.
- **nrFigures** (integer): Het aantal figuren die in de .ini-file beschreven worden.

Naast de **General** sectie bevat het configuratiebestand van een 3D Lijntekening ook nog **nrFigures** secties met figuren, **Figure0** t.e.m. **Figure<nrFigures-1>** genaamd. Deze secties bevatten de volgende velden:

- **type** (string): Deze string specificeert het type van de 3D Figuur. Voor deze opgave moeten enkel 'LineDrawing' figuren ondersteund worden. In de toekomst moeten er ook andere 3D Figuren ondersteund worden en kan deze string dus een andere waarde aannemen.
- **scale** (double): De factor waarmee de figuur moet geschaald worden. Deze transformatie moet eerst worden uitgevoerd.
- **rotateX** (double): De hoek (in graden) waarover de figuur moet worden gedraaid om de *x*-as. Deze transformatie moet worden uitgevoerd na de schaling.
- **rotateY** (double): De hoek (in graden) waarover de figuur moet worden gedraaid om de *y*-as. Deze transformatie moet worden uitgevoerd na de rotatie om de *x*-as.
- **rotateZ** (double): De hoek (in graden) waarover de figuur moet worden gedraaid om de *z*-as. Deze transformatie moet worden uitgevoerd na de rotatie om de *y*-as.
- **center** (tuple van 3 doubles): De cartesische coördinaten van het punt waarheen de figuur moet worden verplaatst. Deze transformatie moet als laatste worden uitgevoerd.
- **color** (tuple van 3 doubles): De RGB waarden van de lijnen van de figuur. Elk van deze waarden ligt tussen 0 en 1.0 (inclusief).
- **nrPoints** (integer): Het aantal punten in de figuur.
- **point0** t.e.m. **point<nrPoints-1>** (tuples van 3 doubles): De cartesische coördinaten van de punten van de figuur.
- **nrLines** (integer): Het aantal lijnen in de figuur.
- **line0** t.e.m. **line<nrLines - 1>** (tuples van 2 integers): De indices van de begin- en eindpunten van de lijnen van de figuur. Een tuple (3,5) betekent dat er een lijn getekend moet worden van **point3** naar **point5**.

Ter verduidelijking wordt hieronder een voorbeeld gegeven:

```
[General]
type = "Wireframe"
size = 1000
eye = (100, 75, 75);
backgroundcolor = (0, 0, 0)
nrFigures = 1
```

```
[Figure0]
type = "LineDrawing"
rotateX = 0
rotateY = 0
rotateZ = 0
scale = 1.0
center = (0,0,0)
```

```

color = (0.0, 1.0, 0.0)

nrPoints = 4
point0 = (0.0, 0.0, 0.0)
point1 = (1.0, 0.0, 0.0)
point2 = (0.0, 1.0, 0.0)
point3 = (0.0, 0.0, 1.0)

nrLines = 3
line0 = (0, 1)
line1 = (0, 2)
line2 = (0, 3)

```

## Tips

- Stel éérs een datastructuur op waarmee je eenvoudig een ruimtelijke figuur kunt voorstellen. Hou hierbij rekening dat het, naar de volgende opdrachten toe, zéér interessant is om een figuur niet langer te beschouwen als een verzameling van punten en lijnen maar wel als een verzameling van punten en oppervlakken. Het grote verschil hierbij is dat een oppervlak (zoals een driehoek, vierkant of vijfhoek) wordt gedefiniëerd door méér dan twee punten. (Voor een voorbeeld datastructuur kun je de slides van het practicum raadplegen)  
Implementeer daarna éérs de eye-point transformatie, de perspectiefprojectie en de nodige functionaliteit om 3D Lijntekeningen uit een ini-bestand te lezen. Eens je er zeker van bent dat je engine wireframes van 3D Lijntekeningen correct kan weergeven kun je aan het implementeren van de transformaties beginnen.
- Om de Eye-transformatiematrix te genereren moet je de cartesische coördinaten  $(x, y, z)$  van het Eye omzetten naar de overeenkomende bolcoördinaten  $(r, \theta, \phi)$ . Hiervoor kun je de vergelijkingen gebruiken die onderaan pagina 20 van de cursus te vinden zijn. Voor het berekenen van de inversie tangens en inverse cosinus functie kun je gebruik maken van de `std::atan` `std::atan2` en `std::acos` functies. Hiervoor moet je wel de `cmath` header includen.
- Het configuratiebestand bevat geen waarde voor  $d$ . Gebruik *enkel* voor deze oefening een standaardwaarde voor  $d$  en gebruik de code voor de 2D lijntekening om de afbeelding te schalen zodat deze netjes op de afbeelding past.
- Op blackboard zijn massa's voorbeeld ini-bestanden samen met de verwachte output te vinden. Gebruik deze om je engine mee te controleren. Verder kun je altijd zélf ini-bestanden opstellen om ook alle randgevallen te testen.