Computer Graphics: Z-Buffering met Lijnen

Jakob Struye
Tim Leys

Z-Buffering met lijnen (1 punt)

Breid je engine uit zodat deze Z-Buffering met lijnen ondersteunt.

Invoerformaat

Het invoerformaat voor Z-Buffering met lijnen is nagenoeg hetzelfde als het invoerformaat als voor platonische lichamen. Het enige verschil is dat het `type`-veld uit de `General`-sectie de waarde '*ZBufferedWireframe*' heeft ipv. '*Wireframe*'.

Integratie

Vanaf volgende opdracht tekenen we vlakken op een andere manier. Hoewel het concept van Z-buffering blijft, wordt waarschijnlijk het overgrote deel van de code voor deze opdracht niet hergebruikt.

Tips

- Aangezien het invoerformaat van Z-Buffering met lijnen hetzelfde is als dat voor platonische lichamen kan je dezelfde invoerbestanden (mits een kleine aanpassing) gebruiken voor het testen van z-buffering met lijnen. De resulterende afbeeldingen zouden eveneens nagenoeg hetzelfde moeten zijn behalve op de plaatsen waar 2 lijnen elkaar kruisen.
- Je kan, voor het tekenen van de lijnen, jezelf baseren op het algoritme uit `EasyImage`.
- Het algoritme voor 1/z in de cursus gaat uit van *a+1* pixels geïndexeerd van *i==a* bij punt A tot en met *i==0* bij punt B. De variabelen in je code zullen mogelijk anders gestructureerd zijn. De formule zal er dus waarschijnlijk wat anders uitzien in je code.
- Je kan voorbeeldbestanden vinden op Blackboard. Merk op dat sommige lijnen die elkaar snijden in de projectie, elkaar erg dichtbij kruisen in 3D-space. Daarom is het al eens mogelijk dat een snijpunt er bij twee engines anderes uitziet door subtiele implementatieverschillen, zonder dat een van de beide fouten is. Staar je niet blind op enkele snijpunten die afwijken van het voorbeeld; implementatiefouten zullen erg duidelijk worden uit afbeeldingen met vele lijnen.