

Computer Graphics: Belichting

Ruben Mennes
Robin Verschoren

Voeg ondersteuning voor belichting aan je engine toe. De pixels waarop een 3D-Lichaam door het Z-Buffering algoritme wordt afgebeeld hebben nu niet langer één en dezelfde kleur. In plaats daarvan wordt de kleur van elke pixel berekend op basis van de eigenschappen van de aanwezige lichtbronnen en de reflectieve eigenschappen en positie van het vlak waarin de pixel gelegen is. De opgave belichting is onderverdeeld in vier aparte deelopgaves. Elk van deze opgaves steunt op voorgaande opgaves. Als je bijvoorbeeld diffuus licht wilt implementeren ben je dus verplicht om eerst ambient licht te ondersteunen.

Belangrijk: In de cursus staat dat de maximale intensiteit van elk van de drie RGB kanalen nooit groter mag zijn dan 1. Voor deze opgave moet je engine echter ook kunnen omgaan met fouten in de specificatie waardoor deze maximale intensiteit toch groter wordt dan 1. Mocht voor een bepaalde pixel de intensiteit van een RGB kanaal te groot zijn, dan moet deze worden ingesteld op intensiteit 1. De intensiteit van de overige RGB kanalen blijft in dat geval ongewijzigd.

Ambient Licht (0.3 punten)

Bij ambient licht wordt de kleur van een pixel bepaald zuiver op basis van de ambiente kleurcomponenten van de aanwezige lichtbronnen en de reflectieve eigenschappen van het vlak. Met de positie van de lichtbronnen of het vlak wordt nog geen rekening gehouden.

Invoer formaat

Het invoer formaat voor deze opgave is gebaseerd op het input formaat voor Z-Buffering, maar heeft wel een aantal wijzigingen ondergaan. Deze wijzigingen worden hieronder per sectie besproken.

De General Sectie

De **General** sectie heeft de volgende wijzigingen ondergaan:

- **type** (string): Dit veld bevat voor deze opgave altijd de waarde ‘*LightedZbuffering*’.
- **nrLights** (integer): Geeft aan hoeveel lichtbronnen er in het .ini-bestand zijn gespecificeerd.

De Figure Secties

In de verschillende **Figure*** secties, waarin de 3D-Lichamen worden beschreven, werd het **color** veld vervangen door het volgende veld:

- **ambientReflection** (tuple van 3 doubles): Dit tuple bevat de reflectie coëfficiënten van het 3D-Lichaam voor ambient licht. Elk van de waarden in dit tuple ligt tussen 0 en 1 (inclusief). Een waarde van (0.5,0,1) geeft bijvoorbeeld aan dat rood licht voor de helft

wordt gereflecteerd, dat groen licht niet wordt gereflecteerd en dat blauw licht volledig wordt gereflecteerd.

Ter illustratie wordt hieronder een voorbeeld van een **Figure** sectie gegeven:

```
[Figure0]
type = "Cube"
center = (0, 0, 0)
scale = 1.0
rotateX = 0
rotateY = 0
rotateZ = 0
ambientReflection = (0, 1, 0)
```

Specificatie van Lichtbronnen

De verschillende lichtbronnen worden gespecificeerd in de secties **Light0** t.e.m. **Light<nrLights-1>**. In elk van deze **Light*** secties komt het volgende veld voor:

- **ambientLight** (tuple van 3 doubles): Dit tuple specificeert de RGB kleur van de ambiente lichtcomponent. Elk van de waarden in dit tuple ligt tussen 0 en 1 (inclusief).

Hieronder wordt een voorbeeld specificatie van een lichtbron weergegeven:

```
[Light0]
ambientLight = (0.25,0.25, 0.25)
```

Diffuus Licht: lichtbron op oneindig (0.4 punten)

Voeg ondersteuning voor diffuus licht (met de lichtbron op oneindig) toe aan je engine. In tegenstelling tot ambient licht wordt er nu voor de bepaling van de kleur van een pixel ook rekening gehouden met de positie van de lichtbronnen en het vlak waarin de pixel gelegen is. Voor dit deel van de opgave wordt er verondersteld dat de lichtbron op 'oneindig' is gelegen. Dit betekent dat de richting waarin het licht schijnt van tevoren is bepaald.

Invoer formaat

Het invoer formaat voor difuus licht is een uitbreiding van dat voor ambient licht. De **General** sectie blijft ongewijzigd maar aan de verschillende **Figure** secties werd het volgende veld toegevoegd:

- **diffuseReflection** (tuple van 3 doubles): Dit veld bevat de reflectie coëfficiënten van het 3D-Lichaam voor diffuus licht.

Ter illustratie wordt hieronder een voorbeeld van een **Figure** sectie gegeven:

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

```
rotateZ = 0
ambientReflection = (0, 1, 0)
diffuseReflection = (1, 0, 0)
```

Aan de verschillende **Light** secties werden de volgende velden toegevoegd:

- **infinity** (boolean): Dit veld bevat een boolean waarde die aangeeft of de lichtbron zich op oneindig bevindt of niet. Vermits voor dit deel van de opgave enkel lichtbronnen op oneindig moeten worden ondersteund hoeft je geen ondersteuning te bieden voor het geval dat *infinity* gelijk is aan **false**. Je moet dit veld echter wél controleren. Als *infinity* gelijk is aan **false** en je ondersteunt geen diffuus licht met puntbronnen dan moet je engine een lege image teruggeven.
- **direction** (tuple van 3 doubles): Dit tuple specificeert een vector die aangeeft in welke richting de diffuse lichtcomponent van de lichtbron schijnt.
- **diffuseLight** (tuple van 3 doubles): Dit tuple specificeert de RGB kleur van de diffuse lichtcomponent. Elk van de waarden in dit tuple ligt tussen 0 en 1 (inclusief).

Let op: als je diffuus licht implementeert moet je engine ook in staat zijn om images zonder diffuus licht te genereren. Je engine moet er dus mee om kunnen gaan dat bovenstaande velden niet in het ini-bestand aanwezig zijn.

Hieronder wordt een voorbeeld specificatie van een lichtbron weergegeven:

```
[Light0]
ambientLight = (0.25,0.25, 0.25)
diffuseLight = (0.50, 0.50, 0.50)
infinity = TRUE
direction = (10, -25, 54)
```

Diffuus Licht: puntbronnen (0.4 punten)

Voeg ondersteuning voor diffuus licht met puntbronnen toe aan je engine.

Invoer formaat

De **General** en **Figure*** secties zijn bij het gebruik van puntbronnen exact hetzelfde als bij gebruik van enkel lichtbronnen op oneindig. **Light*** secties die puntbronnen beschrijven ondergaan de volgende wijzigingen tov. lichtbronnen op oneindig.

- Het veld **infinity** heeft voor een puntbron de waarde **false**.
- Het **direction** veld wordt vervangen door een **location** veld. Dit veld heeft als waarde een tuple van 3 reële getallen dat de locatie van de lichtbron in de ruimte aangeeft.

Hieronder wordt een voorbeeld van de specificatie van een puntbron weergegeven:

```
[Light0]
infinity = false
location = (10, 0, 10)
ambientLight = (0, 0, 0);
diffuseLight = (0.50, 0.75, 0.25);
```

Hou er rekening mee dat er in één ini-bestand zowel puntbronnen als lichtbronnen op oneindig kunnen voorkomen.

Speculair (glanzend) licht (0.4 punten)

Voeg ondersteuning voor specular (glanzend) licht aan je engine toe. Speculair licht moet zowel voor lichtbronnen op oneindig als voor puntbronnen ondersteund worden.

Invoer formaat

In de **Figure*** secties wordt er bij het gebruik van specular licht de volgende velden toegevoegd:

- **specularReflection** (tuple van 3 doubles): Dit tuple bevat de reflectiecoëfficiënten voor specular licht.
- **reflectionCoefficient** (double): Deze waarde bevat de materiaal-coëfficiënt van de figuur die voor specular licht gebruikt wordt. In de cursus wordt deze waarde als m_s genoteerd.

Ter illustratie wordt hieronder een voorbeeld van een **Figure** sectie gegeven:

```
[Figure0]
type = "Cube"
center = (0, 0, 0)
scale = 1.0
rotateX = 0
rotateY = 0
rotateZ = 0
ambientReflection = (0, 1, 0)
diffuseReflection = (1, 0, 0)
specularReflection = (0, 0, 1)
reflectionCoefficient = 20
```

In de **Light*** secties wordt bij het gebruik van specular licht het volgende veld toegevoegd:

- **specularLight** (tuple van 3 doubles): Dit tuple specificeert de RGB kleur van de speculaire lichtcomponent. Elk van de waarden in dit tuple ligt tussen 0 en 1 (inclusief).

Hieronder wordt een voorbeeld van de specificatie van een puntbron weergegeven:

```
[Light0]
infinity = false
location = (10, 0, 10)
ambientLight = (0, 0, 0);
diffuseLight = (0.50, 0.75, 0.25);
specularLight = (0.25, 0.25, 0.25);
```

Let op: als je speculair licht implementeert moet je engine ook in staat zijn om images zónder speculair licht te genereren. Je engine moet er dus mee om kunnen gaan dat bovenstaande velden niet in het ini-bestand aanwezig zijn.

Tips

- Tijdens het implementeren van deze opdracht zal je waarschijnlijk een aantal ingrijpende wijzigingen moeten doorvoeren aan de datastructuren waarmee de 3D-ruimte wordt voorgesteld. Een 3D-Lichaam heeft nu immers niet meer één bepaalde kleur maar wel een aantal

reflectie coëfficiënten waarmee de kleur van het object wordt berekend. Je engine moet echter ook nog steeds *compatibel* zijn met de eerdere opgaves. Dat betekent dat je engine nog steeds 3D-Lichamen moet kunnen weergeven waarvan de kleur in het ini-bestand werd opgegeven. Het is hiervoor echter niet nodig om twee parallelle implementaties van je engine te voorzien.

Het geval waarbij de kleur van elk 3D-Lichaam van te voren wordt opgegeven is immers een speciaal geval van een ‘belichte’ ruimte waarbij er enkel ambient licht aanwezig is en de som van alle ambiente kleur componenten gelijk is aan de maximale lichtintensiteit (1.0, 1.0, 1.0).

Een 3D-Ruimte waarbij de kleur van alle objecten van te voren werd opgegeven kan dus worden weergegeven door de opgegeven kleur van een figuur als ambiente reflectie coëfficiënten te beschouwen, de diffuse en speculaire reflectie coëfficiënten op (0, 0, 0) in te stellen en één lichtpunt met maximale ambiente lichtintensiteit aan de ruimte toe te voegen.

- Op blackboard zijn er massa’s voorbeelden te vinden die je kunt gebruiken om de correctheid van je engine te controleren.