Computer Graphics: Introductie

Jakob Struye
Tim Apers

Inleiding

Het doel van deze oefeningensessie is om vertrouwd te raken met enerzijds de vereisten die we aan jullie grafische engines stellen en anderzijds met de klasses die we daarvoor tot jullie beschikking stellen. De grafische engine moet in staat zijn om, op basis van een geparsed INI-bestand, de overeenkomstige image te genereren. Hiervoor moet de functie `generate_image` in de source-file `engine.cc` worden geïmplementeerd. Hou er rekening mee dat deze functie om moet kunnen gaan met de verschillende INI-bestandsformaten van de verschillende opdrachten. Om onderscheid te maken tussen INI-bestanden van verschillende opdrachten, bevat het INI-bestand *altijd* een `General`-sectie met daarin het `type`-veld. De waarde van dit veld verschilt naargelang de opdracht. Ter verduidelijking wordt hieronder een voorbeeld gegeven:

```
[General]
type = "<type van de opdracht>"
```

Om de INI-bestanden te interpreteren en de bijbehorende images te kunnen genereren worden de klasses `ini::Configuration` en `img::EasyImage` ter beschikking gesteld. Deze klasses zijn, samen met het bestand `engine.cc` en een voorbeeld `CMakeLists.txt` te vinden in de archive `utils.tar.gz` op BlackBoard. Meer informatie over deze klasses is in de slides te vinden.

In de rest van dit document worden drie opdrachten gegeven. Elk van deze opdrachten wordt beschreven op dezelfde manier als de opdrachten van de komende weken. Let op: De opdrachten van deze sessie zijn *niet* verplicht en worden op het einde van het jaar ook *niet* gequoteerd. Het is echter wél aangeraden om deze oefeningen te maken om vertrouwd te raken met de manier van werken bij het practicum. Net zoals bij voor alle volgende opdrachten, zijn er voor deze opdrachten op BlackBoard een aantal voorbeeld ini-bestanden samen met de verwachte output beschikbaar. Deze zijn te vinden in de archive `intro.tar.gz`

Opdracht 1: ColorRectangle

Beschrijving

Implementeer code in staat is om een 'ColorRectangle' van variabele breedte en hoogte te genereren. De stand-alone code om een 'ColorRectangle' van 256 bij 256 pixels te genereren wordt hieronder weergegeven:

```
#include <iostream>
#include <fstream>
#include "easy_image.h"

int main()
{
```

```
    img::EasyImage image(256,256);
    for(unsigned int i = 0; i < 256; i++)
    {
        for(unsigned int j = 0; j < 256; j++)
        {
            image(i,j).red = i;
            image(i,j).green = j;
            image(i,j).blue = (i+j)%256;
        }
    }
    std::ofstream fout("out.bmp", std::ios::binary);
    fout << image;
    fout.close();
    return 0;
}
```

Begin met bovenstaande code te integreren in `engine.cc` door de code die de 'ColorRectangle' genereert in een aparte functie te zetten. Implementeer vervolgens in de functie `generate_image` de nodige functionaliteit om de benodigde input parameters uit een geparsed INI-bestand in te lezen en de functie die de 'ColorRectangle' genereert aan te roepen.

Hou er rekening mee dat deze code *enkel* het het correcte resultaat geeft voor een vierkant dat precies 256 bij 256 pixels groot is. Jullie implementatie moet eveneens in staat zijn om een 'ColorRectangle' van willekeurige grootte te genereren. Hierbij kunnen de breedte en de hoogte van de rechthoek van elkaar verschillen. Je hoeft er echter niet voor te zorgen dat de kleuren correct geschaald worden. Na 256 pixels mag de 'ColorRectangle' herhaald worden.

Invoerformaat

De input parameters van de opdracht worden via een INI-bestandsformaat opgegeven. Het INI-bestand bevat altijd een `General` sectie. Deze bevat, in het geval van deze opdracht, het volgende veld:

- **type** (string): Dit veld (dat in alle configuratiebestanden aanwezig is) bevat in het geval van de 'ColorRectangle' opdracht de waarde '*IntroColorRectangle*'.

Naast de `General` sectie bevat het INI-bestand ook nog een `ImageProperties` sectie. Deze sectie bevat de volgende velden:

- **width** (int): De breedte van de rechthoek (en de image) in pixels.
- **height** (int): De hoogte van de rechthoek (en de image) in pixels.

Ter verduidelijking wordt hieronder een voorbeeld gegeven:

```
[General]
type = "IntroColorRectangle"

[ImageProperties]
width = 256
height = 256
```

Opdracht 2: Blocks

Beschrijving

Implementeer een functie waarmee een image met dambord-motief gegenereerd kan worden. Deze functie krijgt als input parameters W_i en H_i: de breedte en hoogte van de te genereren image, N_x en N_y: het aantal vakjes dat het dambord in de X- en Y-richting bevat. Verder worden ook de kleuren, die voor de 'witte' en voor de 'zwarte vakjes' moeten worden gebruikt opgegeven. Voor de eenvoud wordt er verondersteld dat de breedte en de hoogte van de image altijd een volkomen veelvoud zijn van het aantal vakjes in X- en de Y-richting van het dambord. Zoals te zien in

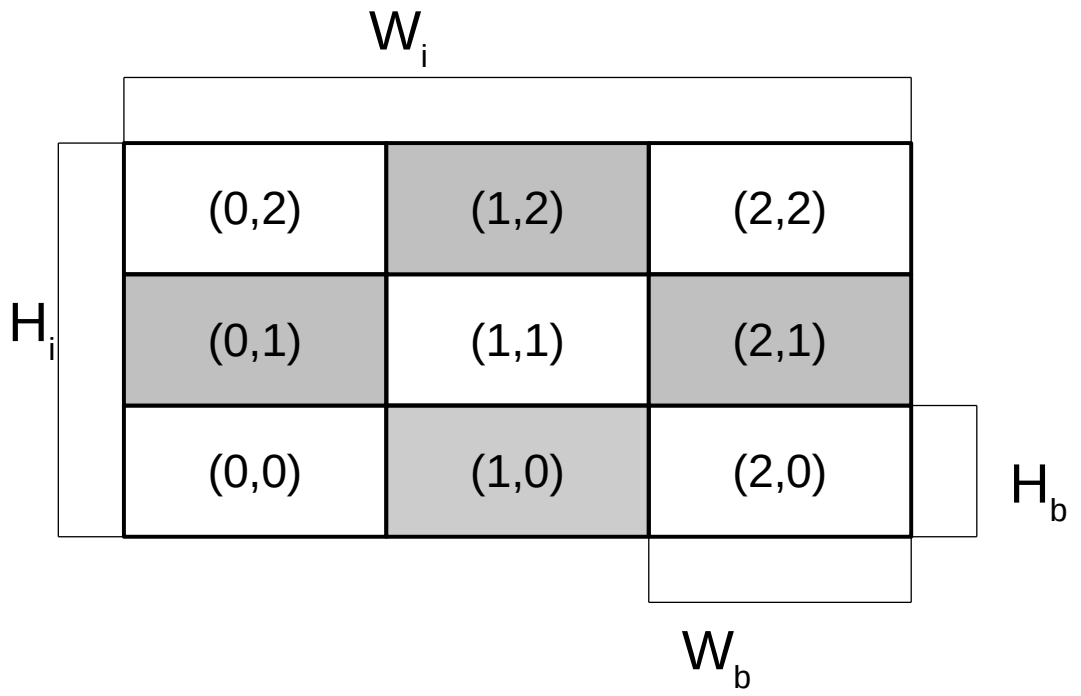

Figure 1: Een voorbeeld dambord.

figuur 1, bestaat een dambord-image uit een aantal vakjes. Elk van deze vakjes heeft een X- en een Y-coördinaat (B_x, B_y). Als de som van deze coördinaten even is dan wordt het vakje wit gekleurd. Als de som van deze coördinaten oneven is wordt het vakje zwart gekleurd.

Om een dambord-image te genereren moeten eerst W_b en H_b, de breedte en de hoogte van deze vakje, berekend worden. Deze zijn gelijk aan de hoogte en de breedte van de image gedeeld door respectievelijk het aantal vakjes in de X- en het aantal vakjes in de Y-richting:

- $W_b = \frac{W_i}{N_x}$
- $H_b = \frac{H_i}{N_y}$

Vervolgens kan het dambord gegenereerd worden door, net zoals bij de 'ColorRectangle' over alle pixels van de image te itereren en voor elke pixel de kleur te berekenen op basis van zijn x- en y-coördinaat. De kleur van een pixel met coördinaten (p_x, p_y) kan als volgt berekend worden:

1. Bepaal eerst de coördinaten van het vakje waartoe de pixel behoort. Deze coördinaten kunnen berekend worden door de x en y- coördinaat van de pixel te delen door de breedte en de hoogte van één vakje. *Let Op*: gebruik hierbij de *integer*deling!

- $B_x = \lfloor \frac{p_x}{W_b} \rfloor$
- $B_y = \lfloor \frac{p_y}{H_b} \rfloor$

2. Bereken de som van de coördinaten van het vakje waartoe de pixel behoort. Als deze som even is kleur dan het vakje 'Wit'. Kleur anders het vakje 'Zwart'

Invoerformaat

Het INI-bestandsformaat voor deze opdracht bevat drie verschillende secties:

De `General` sectie bevat het volgende veld:

- **type** (string): Dit veld bevat voor deze opdracht de waarde '*IntroBlocks*'.

De `ImageProperties` sectie bevat de volgende velden:

- **width** (int): De breedte van de te genereren afbeelding.
- **height** (int): De hoogte van de te genereren afbeelding.

De `BlockProperties` sectie bevat de volgende velden:

- **nrXBlocks** (int): Het aantal vakjes van het dambord in de X-richting.
- **nrYBlocks** (int): Het aantal vakjes van het dambord in de Y-richting.
- **colorWhite** (tuple van 3 doubles): De RGB waarden van de kleur van de 'Witte' vlakjes. Elke waarde ligt tussen 0 en 1 (inclusief). Je moet deze waarden zélf schalen naar de waarden die de `EasyImage` klasse verwacht.
- **colorBlack** (tuple van 3 doubles): De RGB waarden van de kleur van de 'Zwarte' vlakjes. Elk van deze waarden ligt tussen 0 en 1 (inclusief). Ook deze waarden moeten geschaald worden.
- **invertColors** (boolean): Dit is een *optioneel veld*. Als dit veld aanwezig is en de waarde TRUE heeft dan moeten de kleuren voor de zwarte en de witte vlakjes omgewisseld worden.

Ter verduidelijking wordt er hieronder een voorbeeld gegeven:

```
[General]
type = "IntroBlocks"

[ImageProperties]
width = 534
height = 534

[BlockProperties]
colorWhite = (0.5, 0.0, 0.0)
colorBlack = (0.0, 0.75, 0.0)
nrXBlocks = 5
nrYBlocks = 5
invertColors = FALSE
```

Opdracht 3: Lines

Beschrijving

Deze opdracht bestaat uit drie delen:

1. Implementeer een functie die de 'QuarterCircle' lijn structuur die in figuur 2 wordt weergegeven op een image kan tekenen. Deze functie heeft als input parameters H_i en W_i, de hoogte en de breedte van de te genereren image en N, het aantal te tekenen lijnen. Verder wordt ook de kleur van de lijnen en de achtergrondkleur van de image gespecifiëerd. H_s en W_s de verticale en horizontale afstand tussen twee opeenvolgende lijnen wordt hierbij als volgt berekend:
   - $H_s = \frac{H_i}{N-1}$
   - $W_s = \frac{W_i}{N-1}$

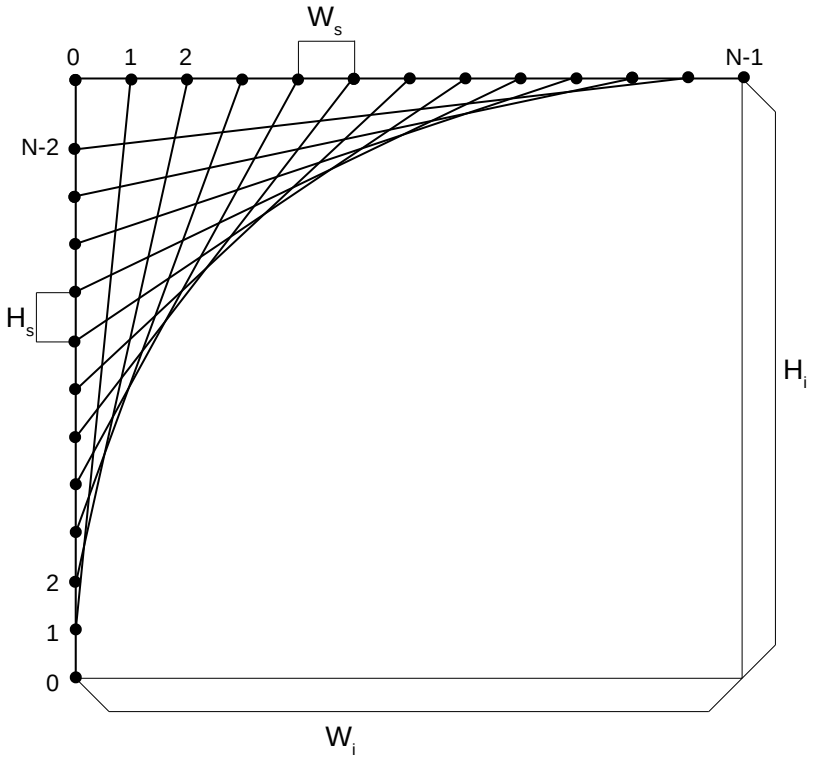

Figure 2: De 'QuarterCircle' draden structuur.

2. Implementeer een functie die gebruikmaakt van de vorige functie om de 'Eye' lijn structuur op een image te tekenen. Een voorbeeld van de eye lijn structuur is te vinden in figuur 3. Deze functie heeft dezelfde input parameters als de vorige functie.

3. Implementeer een functie die gebruikmaakt van de 'QuarterCircle'-Functie om een 'Diamond' lijn structuur op een image te tekenen. In figuur 4 is een voorbeeld van de 'Diamond' lijn structuur te vinden. Ook deze functie heeft dezelfde input parameters als de eerste functie.

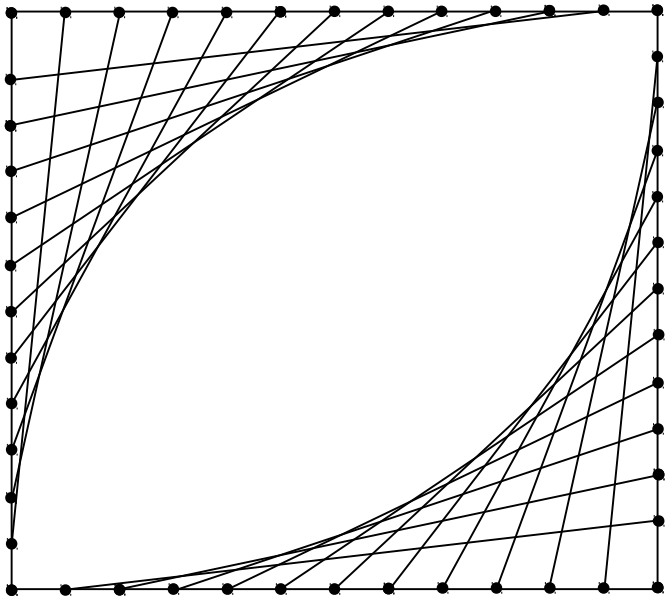

Figure 3: De ‘Eye’ draden structuur.

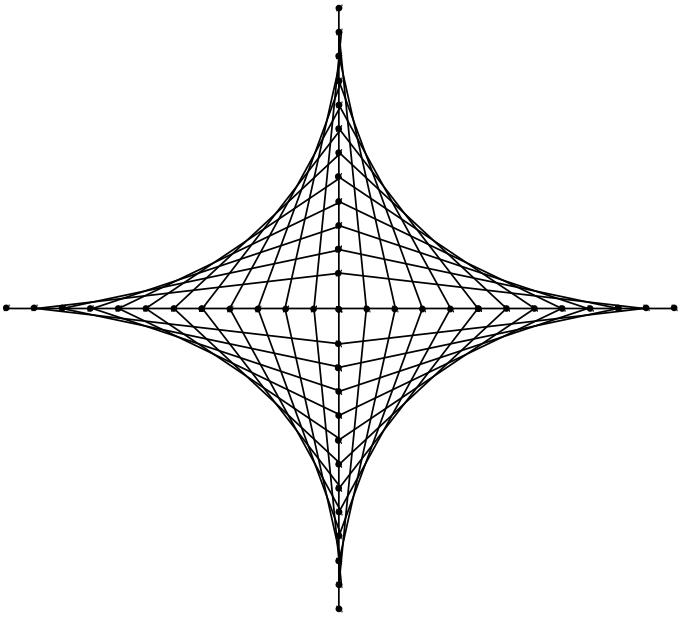

Figure 4: De ‘Diamond’ draden structuur.

Invoerformaat

Het INI-bestandsformaat voor deze opdracht bevat drie verschillende secties:
De `General` sectie bevat het volgende veld:

- **type** (string): Dit veld bevat voor deze opdracht de waarde ‘*IntroLines*’

De `ImageProperties` sectie bevat de volgende velden:

- **width** (int): De breedte van de te genereren afbeelding.
- **height** (int): De hoogte van de te genereren afbeelding.

De `LineProperties` sectie bevat de volgende velden:

- **figure** (string): Dit veld bevat de waarde '*QuarterCircle*', '*Eye*' of '*Diamond*' afhankelijk van de lijnfiguur die getekend moet worden.
- **nrLines** (int): Het aantal lijnen dat per 'QuarterCircle' getekend moet worden. Als er voor het genereren van een bepaalde figuur meerdere QuarterCircles getekend moeten worden, dan moet voor elk van deze 'QuarterCircles' *numLines* lijnen worden gebruikt.
- **backgroundcolor** (tuple van 3 doubles): De RGB waarden van de achtergrondkleur van de image. Elk van deze waarden ligt tussen 0 en 1 (inclusief).
- **lineColor** (tuple van 3 doubles): De RGB waarden van kleur waarin de lijnen moeten worden getekend. Elk van deze waarden ligt tussen 0 en 1 (inclusief).

Hieronder wordt er een voorbeeld gegeven:

```
[General]
type = "IntroLines"

[ImageProperties]
width = 500
height = 500

[LineProperties]
figure = "QuarterCircle"
bgColor = (0.0, 0.0, 0.0);
lineColor = (1.0, 0.0, 0.0)
nrLines = 20
```