

# Computer Graphics: 2D L-Systemen

Ruben Mennes  
Robin Verschoren

## 2D L-Systemen (1 punt)

1. Implementeer code die een lijst van lijnen gecentreerd kan weergeven op een (Easy)Image. De **EasyImage**-klasse is te vinden op Blackboard in de Archive **utils.tar.gz**. Hierbij moet je er rekening mee houden dat de lijnen zich bevinden in het reële vlak. Dat betekent dat de coördinaten van de begin- en eindpunten van het lijnstuk reële getallen zijn. Verder mag de langste zijde van de image maximaal **size** pixels groot zijn. Voor het implementeren van deze functionaliteit kun je je baseren op de algoritmes die worden beschreven vanaf pagina 7 in de cursus.
2. Implementeer code die in staat is om de 7 componenten van een 2D L-Systeem (Alphabet, Draw functie, Angle, Initiator, Replacement Rules, Initial angle en Iterations) uit een L2D-bestand in te lezen en een lijst met reële lijnen te genereren. Voor de primaire opdracht moet er nog geen ondersteuning worden ingebouwd voor haakjes en stochastische replacement rules. Voor het inlezen van de L2D-bestanden kun je gebruikmaken van de **LSystem2D**-klasse die op Blackboard is voorzien. Deze is te vinden in de Archive **lparser.tar.gz**. Voor het omzetten van een **LSystem2D**-Object naar een lijst met reële lijnen kun je je baseren op de algoritmes die in de cursus vanaf pagina 9 worden besproken.
3. Schrijf op basis van deze functionaliteit, en de code die in **utils.tar.gz** te vinden is een eerste versie van je Grafische Engine. Deze moet in staat zijn om een L-Systeem uit een L2D-bestand uit te lezen en vervolgens weer te geven op een image met maximale grootte **size**.

## Invoerformaat

De L-Systemen worden gespecificeerd in het L2D-bestandsformaat. Dit bestandsformaat laat toe om 2D L-Systemen op een eenvoudige manier te beschrijven. L2D-bestanden kunnen worden geparset met de **LSystem2D**-klasse in de **lparser.tar.gz** Archive.

De rest van de input parameters worden volgens via een INI-bestandsformaat opgegeven. De Parser hiervoor is te vinden op Blackboard in de Archive **utils.tar.gz**.

De .ini-bestanden bevatten altijd een **General** sectie. Deze bevat, in het geval van deze opdracht, de volgende velden:

- **type** (string): Dit veld (dat in alle configuratiebestanden aanwezig is) bevat in het geval van een 2D lijntekening de waarde '*2DLSystem*'.
- **size** (integer): De grootte van de afbeelding in de richting die de meeste pixels vereist.
- **backgroundcolor** (tuple van 3 reële getallen): De RGB waarden van de achtergrondkleur van de image. Elk van deze waarden ligt tussen 0 en 1.0 (inclusief).

Naast de **General** sectie bevat het .ini-bestand ook nog een **2DLSystem** sectie. Deze bevat de volgende velden:

- **inputfile** (string): De naam van het bestand waarin het L-Systeem beschreven wordt.
- **color** (tuple van 3 reële getallen): De RGB waarden van de lijnen van de figuur. Elk van deze waarden ligt tussen 0 en 1.0 (inclusief).

Ter verduidelijking wordt hieronder een voorbeeld gegeven:

```
[General]
type = "2DLSystem"
size = 1000
backgroundcolor = (0.0, 0.0, 0.0)

[2DLSystem]
inputfile = "Koch.L2D"
color = (0.5, 1, 0)
```

## L-Systemen met haakjes (0.125 punten)

Voorzie ondersteuning voor het renderen van L-Systemen met haakjes. De klasse **LSystem2D** kan reeds overweg met L-Systemen met haakjes dus voor deze opdracht moet deze klasse niet worden aangepast.

## L-Systemen met stochastische replacement rules (0.125 punten)

Voorzie ondersteuning voor het renderen van L-Systemen met stochastische replacement rules. De klasse **LSystem2D** heeft nog geen ondersteuning voor stochastische replacement rules dus voor deze opdracht zul je zélf het L2D bestandsformaat moeten uitbreiden en de Parser moeten aanpassen. Dit betekent uiteraard ook dat je bij het indienen van je engine zélf enkele ini- en L2D bestanden zult moeten voorzien waarmee deze functionaliteit wordt gebruikt. Je engine moet echter ook nog overweg kunnen met gewone L2D bestanden dus zorg ervoor dat je gekozen bestandsformaten *backwards compatible* zijn met die voor gewone L-Systemen.

## Tips

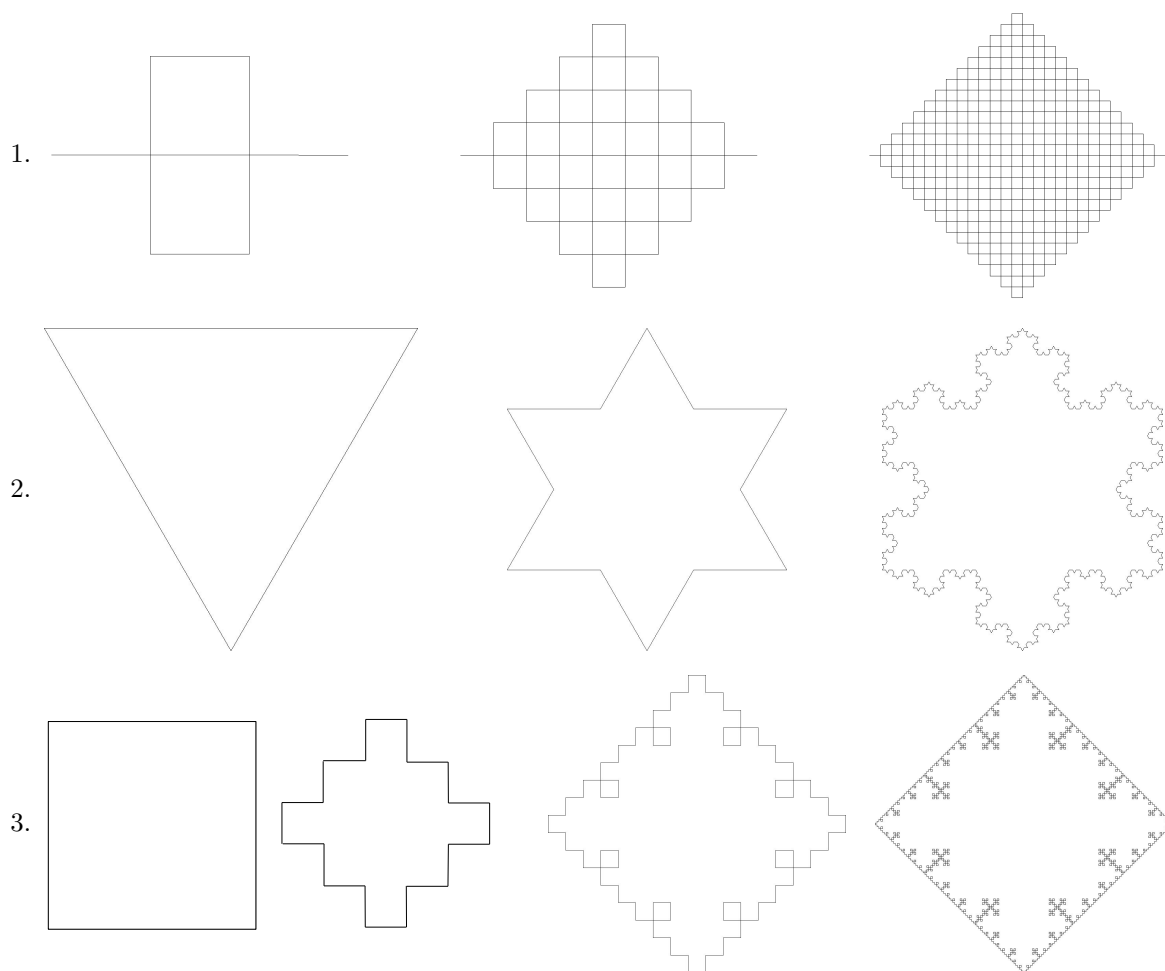
1. Werk vóór je begint met implementeren eerst een *zinnige* datastructuur uit om de lijnen van het L-Systeem in bij te houden. Hou hierbij rekening met het feit dat je niet van te voren weet hoeveel lijnen je zult moeten opslaan en dat je deze lijnen bij het tekenen vooral lineair zult moeten overlopen. Als je geen duidelijk idee hebt van hoe deze datastructuur eruit zou moeten zien, kun je altijd de datastructuur die in de slides wordt besproken als vertrekpunt gebruiken.
2. Door gebruik te maken van recursie hoeft je niet expliciet de replacements toe te passen. Hierdoor voorkom je dat je een grote string moet opstellen wat een aanzienlijke snelheidswinst kan opleveren.

3. Voor gewone L-Systemen en L-Systemen met haakjes zijn er op blackboard voorbeeld in- en L2D-bestanden, samen met de verwachte output voorzien. Aan de hand van deze bestanden kun je controleren of je Engine naar behoren werkt. Daarnaast kun je hiervoor altijd gebruik maken van de L-Systemen die in de cursus beschreven staan. Voor L-Systemen met stochastische replacement rules zijn er geen voorbeeld bestanden voorzien vermits het bestandsformaat voor deze L-Systemen vrij te kiezen is.

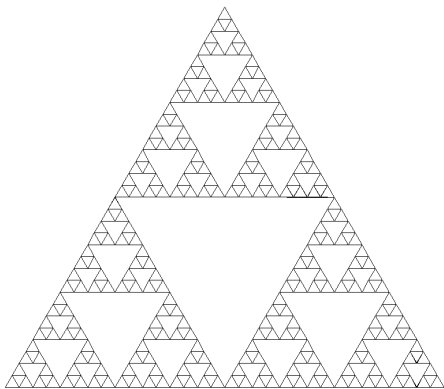
## Extra oefeningen

De volgende oefeningen zijn *niet* verplicht en de oplossingen worden *niet* gequoteerd. Het is wel belangrijk om ze op te lossen aangezien er op het theorie examen gelijkaardige opgaves zullen zijn voor hoofdstuk 1.

Hieronder worden een aantal L-Systemen grafisch weergegeven in verschillende stadia van ontwikkeling. Stel, voor elke serie tekeningen, het overeenkomende L-Systeem op.



4.



5.

