

Science Park 123  
1098XG Amsterdam

6 maart 2013

Jury ‘Elegante Algoritmes’  
SETUP  
Oudegracht 183  
3511 NE Utrecht

Beste Jury

Bij deze sturen wij u onze bijdrage aan de wedstrijd ‘Elegante Algoritmes’.

De ondertitel van de wedstrijd, ‘programmeren als ambacht’, brengt ons bij het volgende: als ambachtlieden gebruiken wij, zoals alle ambachtslieden door de eeuwen heen, gereedschap dat het beste past bij ons specifieke ambacht. Nu is ons ambacht niet programmeren op zich, maar wel numerieke analyse, met name aan dynamische en stochastische systemen. Ons favoriete gereedschap daarvoor is Matlab, een omgeving die in de ogen van ware programmeurs misschien alles behalve elegant is, maar ons juist toelaat eenvoudig, en dus elegant, te rekenen aan de problemen waar wij mee werken.

Een ander gevolg van ons ambacht is dat wij haast altijd werken met stochasten, en onze uitkomst van numerieke simulaties dus slechts één realisatie is uit een vaak oneindige ruimte van mogelijkheden. Zo ook dit werk, al is het aantal mogelijkheden in dit geval strikt gezien eindig. We nodigen u dan ook uit het algoritme een paar keer te proberen, overigens staat het u ook vrij om de parameters aan het begin van `main.m` naar eigen inzicht te variëren.

Tot slot willen wij u, en ook de organisatie, bedanken voor deze wedstrijd die twee geliefde kunstvormen verenigt.

Met vriendelijke groet

Jesse en Keith

Bijlage(n): Korte uitleg van het algoritme

## Korte uitleg van het algoritme

Het bestand `main.m` bevat de hoofdlijn van het algoritme, dat uit vier delen bestaat. Deze vier delen worden voorafgegaan door een lijst met parameters.

Het eerste deel verdeelt een matrix in vlakken met unieke nummers, en ‘linten’ met nummer 1. Het tweede deel kleurt de vlakken, het derde de linten. Tot slot verzorgt het vierde deel de uitkomst opdat het een mooi plaatje wordt. De delen worden hieronder in meer detail uitgelegd.

Het eerste deel draait om de recursieve functie `divide`. Deze functie werkt steeds met een stukje van de matrix en doet met dit stukje een van drie dingen:

1. hij deelt het vlak in tweeën,
2. i.d. met een ‘lint’ tussen beide delen of
3. hij voorziet het vlak van een uniek nummer,

bij de derde optie wordt eventueel een klein blokje toegevoegd. De recursie ontstaat doordat elke keer als een vlak in tweeën wordt gedeeld, de functie `divide` wordt aangeroepen om op beide delen te werken. De code voor de criteria voor bovengenoemde willekeurige keuzes wijst o.i. zichzelf.

Het inkleuren van de blokken gebeurt, helaas, op iets minder elegante wijze, in `colourblock.m`. Voor elk vlak wordt een willekeurige kleur gekozen volgens de opgegeven waarschijnlijkheden voor alle kleuren. Deze waarschijnlijkheden zijn afhankelijk van de afstand van het in te kleuren vlak tot het midden. Als de gekozen kleur gelijk is aan een kleur in een reeds ingekleurde buur, wordt de keuze verworpen en wordt een nieuwe kleur geprobeerd, etc.

De linten worden volgens andere kleurwaarschijnlijkheden ingekleurd. Hier wordt géén rekening gehouden met burens, in navolging van het origineel.

Tot nu toe werd steeds gewerkt met een vierkante matrix. Pas bij de vierde stap worden de hoeken eraf geknipt om zo tot een ruit<sup>1</sup> te komen. Tot slot wordt het geheel opgeschaald en van een dunne grijze rand voorzien, beiden uit puur esthetische overwegingen.

Overigens zijn wij ons er van bewust dat Matlab een commercieel programma is, dus hebben wij er voor gezorgd dat onze code ook werkt in opensource-variant Octave, zelfs op Android-smartphones.

---

<sup>1</sup>In de volksmond, althans.