# Meetrapport titel

## Namen en datum

Alexander Streng 17-02-2014

## Doel

Bij het maken van een salt & pepper noise generator, kwam ik erachter dan rand() niet helemaal compleet random is. Graag wil ik deze vergelijken met de random implementatie van c++11 (gebruik makende van de Mersenne Twister 19937 generator) en kijken welke meer random is en hoe het met performance zit.

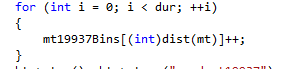
## Hypothese

Ik denk dat de c++ implementatie meer random is bij gebruik van dezelfde seed. In principe is dit voor het genereren van salt & pepper noise niet heel belangrijk, maar het speelt wel een rol. Ook denk ik dat de MT19937 sneller is. Omdat de rand altijd een nummer tussen de 0 en 32767 genereerd (en deze vervolgens module doet.)

## Werkwijze

Ik ga 2 histogrammen opstellen van de 2 random number methodes. Ook ga ik middels de bijgeleverde timer de tijden meten die de methodes er over doen.

Als seed zal ik de huidige tijd gebruiken. Ik doorloop eerst 1x een loop van 100,000,000 keer als controle. Vervolgens ga ik 20 keer een loop van 100, 000, 000, 000 keer doorlopen en genereer een random nummer tussen de 0 en 100. Deze zet ik in de bijbehorende bins.



De resultaten sla ik op in een .csv om te vergelijken. Op deze manier wordt de afwijking van de gegenereerde nummers duidelijk zichtbaar.

## Resultaten

De resultaten van de controle ( 100,000,000 keer een nummer genereren ) zijn als volgt;

De resultaten van 100,000,000,000 keer zijn iets interessanter. Zie de laatste pagina voor de tabel.

## Verwerking

Laat zien hoe je de meetresultaten verwerkt om een conclusie te kunnen trekken. Het is niet nodig om alle berekeningen op te schrijven.

## Conclusie

Geef aan welke conclusie kan worden getrokken uit de verwerking van de meetresultaten.

## Evaluatie

Leg een verband tussen de getrokken conclusie en het doel van het experiment (en de hypothese). Ga daarbij ook in op bijvoorbeeld de meetonzekerheid als gevolg van de gebruikte meetmethoden of eventuele meetfouten.

