
Meetrapport container

Florian Humblot - 1720570
Vera Schoonderwoerd - 1721202

Datum: 21 februari 2019

Inhoudsopgave

Doel	3
Hypothese	3
Werkwijze.....	3
Resultaten	3
Verwerking	3
Conclusie	4
Evaluatie	4

Doel

Het doel van dit experiment is het meten van de snelheid van de imageShell implementatie. De meting gaat over de algemene execution time van het volledige programma waarbij de enige verandering de implementatie van de studentRGBImage en studentIntensityImage is.

Hypothese

De hypothese is als volgt: de student implementatie zal 10% sneller zijn dan de implementatie die aangeleverd is. Dit omdat wij proberen om eventuele overhead weg te halen door zo dicht mogelijk op het geheugen te werken.

Werkwijze

Om de snelheid van de implementaties te meten zonder invloed van andere programma's wordt elk plaatje uit de testset 100x door het programma heen gehaald voor de default implementatie en vervolgens 100x door de student implementatie. Op die manier hopen wij externe invloeden uit te kunnen filteren. De totale execution time wordt bijgehouden voor elk van de twee implementaties waarna wij een gemiddelde executietijd kunnen berekenen. Totaal worden er 7 plaatjes 1000x per implementatie getest. Dit zorgt dus voor 1400 uitvoeringen van het programma. De gemiddelde tijd van elke executie wordt in een csv bestand geschreven dat vervolgens met excel geanalyseerd kan worden. De metingen worden in release mode uitgevoerd met optimalisatie flag O2 (Maximum optimization for speed).

Resultaten

Average of executiontime Image	default	student
Set A/TestSet Images/child-1.jpg	212.37	192.28
Set A/TestSet Images/female-1.png	127.83	138.54
Set A/TestSet Images/female-2.png	102.31	93.54
Set A/TestSet Images/female-3.png	123.51	145.84
Set A/TestSet Images/male-1.png	127.58	129.99
Set A/TestSet Images/male-2.png	138.77	133.1
Set A/TestSet Images/male-3.png	170.02	139.99
Grand Total	143.1985714	139.04

Verwerking

	default	student	% faster
	212.37	192.28	9.46%
	127.83	138.54	-8.38%
	102.31	93.54	8.57%
	123.51	145.84	-18.08%
	127.58	129.99	-1.89%
	138.77	133.1	4.09
	170.02	139.99	17.66
	143.1985714	139.04	2.90

Voor de verwerking van de resultaten hebben wij de baseline gelegd op de default implementatie. Dus de gemiddelde tijd van een default verwerking van plaatje x is gelijk aan 100%. De formule was dus: $(100 - \text{student_snelheid}/(\text{default_snelheid}/100))$

Conclusie

Gemiddeld was de student container niet 10% sneller zoals verwacht, los van de uitschieter op 17% bleek uiteindelijk de gemiddelde snelheidswinst maar 2.9% te zijn geworden.

Evaluatie

Het is ons nog niet helemaal duidelijk waarom de studentimplementatie bij sommige plaatjes trager is dan de default implementatie, ook met het gereedschap dat visual studio biedt is het ons niet gelukt erachter te komen waar dit aan ligt.