

1. Meetrapport Imageshell efficiency

1.1. Namen en datum

Stefan van der Ham & Bas van Rossem, 7 april 2019.

1.2. Doel

We willen met dit experiment kijken welke van de twee implementaties efficiënter is. De default implementatie van de imageshell tegenover onze implementatie van de imageshell.

1.3. Hypothese

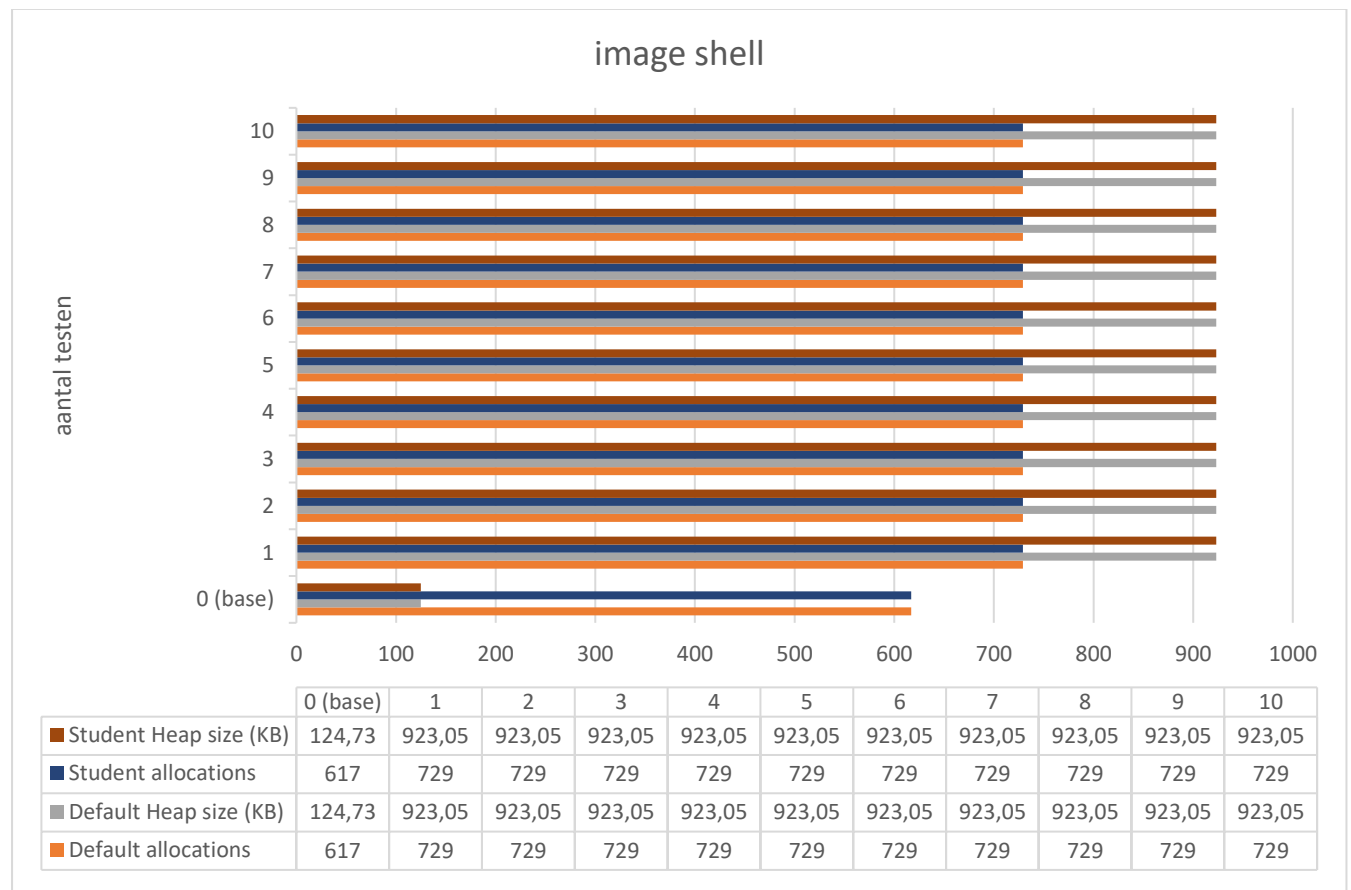
Wij verwachten tot onze implementatie een stuk minder memory in beslag neemt, omdat het een vrij simpele oplossing is.

1.4. Werkwijze

wij gaan het programma 10 keer uitvoeren en maken dan tussendoor een snapshot met de Memory Snapshot tool van Visual studio.

1.5. Resultaten

TIMES	DEFAULT ALLOCATIONS	DEFAULT HEAP SIZE (KB)	STUDENT ALLOCATIONS	STUDENT HEAP SIZE (KB)
0 (BASE)	617	124,73	617	124,73
1	729	923,05	729	923,05
2	729	923,05	729	923,05
3	729	923,05	729	923,05
4	729	923,05	729	923,05
5	729	923,05	729	923,05
6	729	923,05	729	923,05
7	729	923,05	729	923,05
8	729	923,05	729	923,05
9	729	923,05	729	923,05
10	729	923,05	729	923,05



1.6. Verwerking

Difference = De default heap size – De student heap size

Difference = 923,05 – 923,05

Difference = 0

Difference = De default heap size – De student heap size

Difference = 729 – 729

Difference = 0

1.7. Conclusie

Onze implementatie heeft nul verschil met de default implementatie. Het blijkt dat het niet uitmaakt hoe vaak we de code uitvoeren aangezien de resultaten altijd hetzelfde waren. Dit zal waarschijnlijk komen door het goed vrijgeven van de resources op de computer.

1.8. Evaluatie

Het doel van dit experiment was om te kijken hoeveel efficiënter onze implementatie van de image shell was in vergelijking tot de default implementatie. Onze hypothese zat er helemaal naast.. We hebben heel wat metingen gedaan, maar de uitkomst was altijd hetzelfde