

1. Meetrapport StepToIntensity 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. Wij denken met dit experiment 30 minuten bezig te zijn.

1.4. Werkwijze

Voor deze tests maken wij gebruik van de originele Main.cpp, omdat deze onaangepast is (Code referenties wijzen naar OG_Main.cpp)

Stap 1

Regel 18-19 zet de image factory op ImageFactory:: STUDENT.

Stap 2

Plaats een for loop (size 10) om *regel 26 en voor 50*.

Stap 3

Zet een breakpoint neer op *regel 18 en 49*.

Stap 4

Start het programma en wacht tot de eerste breakpoint wordt geraakt.

Stap 5

Maak een snapshot (dit is je base line) verwijder daarna het breakpoint bij *18* en vervolg het programma (continue), wacht tot het volgende breakpoint wordt geraakt.

Stap 6

Maak een snapshot, noteer resultaat en vervolg het programma.

Stap 7

Bij breakpoint Ga naar Stap 6.

Stap 8

Zet executor->executePreProcessingStep1(false) naar true op regel 66

Stap 9

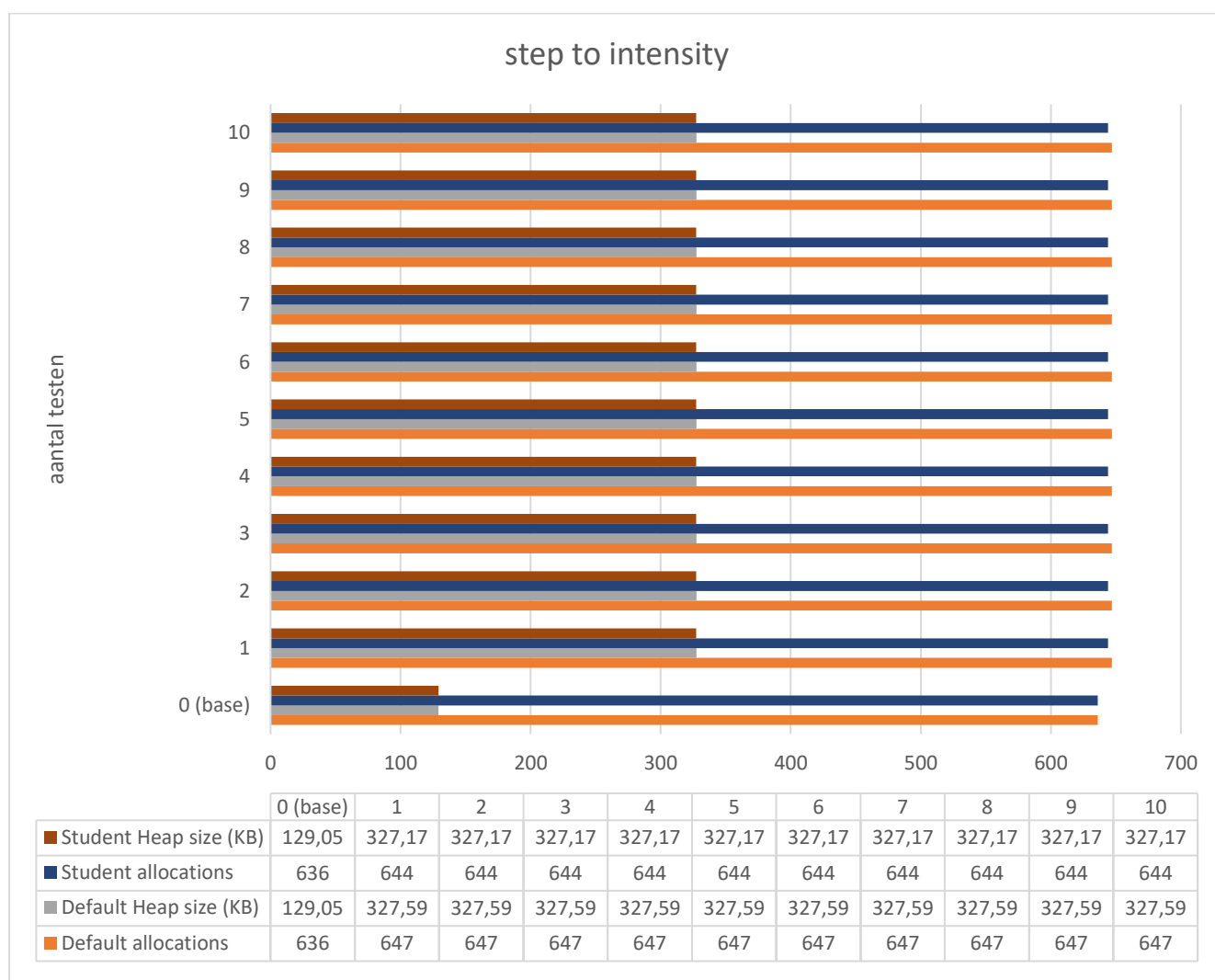
voer de stappen 2 tot en met 7 uit.

Stap 10

Zet resultaat om in een diagram.

1.5. Resultaten

<i>times</i>	<i>Default allocations</i>	<i>Default Heap size (KB)</i>	<i>Student allocations</i>	<i>Student Heap size (KB)</i>
<i>0 (base)</i>	636	129,05	636	129,05
<i>1</i>	647	327,59	644	327,17
<i>2</i>	647	327,59	644	327,17
<i>3</i>	647	327,59	644	327,17
<i>4</i>	647	327,59	644	327,17
<i>5</i>	647	327,59	644	327,17
<i>6</i>	647	327,59	644	327,17
<i>7</i>	647	327,59	644	327,17
<i>8</i>	647	327,59	644	327,17
<i>9</i>	647	327,59	644	327,17
<i>10</i>	647	327,59	644	327,17



1.6. Verwerking

Difference = De default heap size – De student heap size

Difference = 327,59 – 327,17

Difference = 0,42KB

Difference = De default allocations – De student allocations

Difference = 647 – 644

Difference = 3

1.7. Conclusie

Onze implementatie heeft weinig 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. Desondanks was onze implementatie toch zuiniger

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 niet helemaal naast. We hebben heel wat metingen gedaan, maar de uitkomst was altijd hetzelfde