

# 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

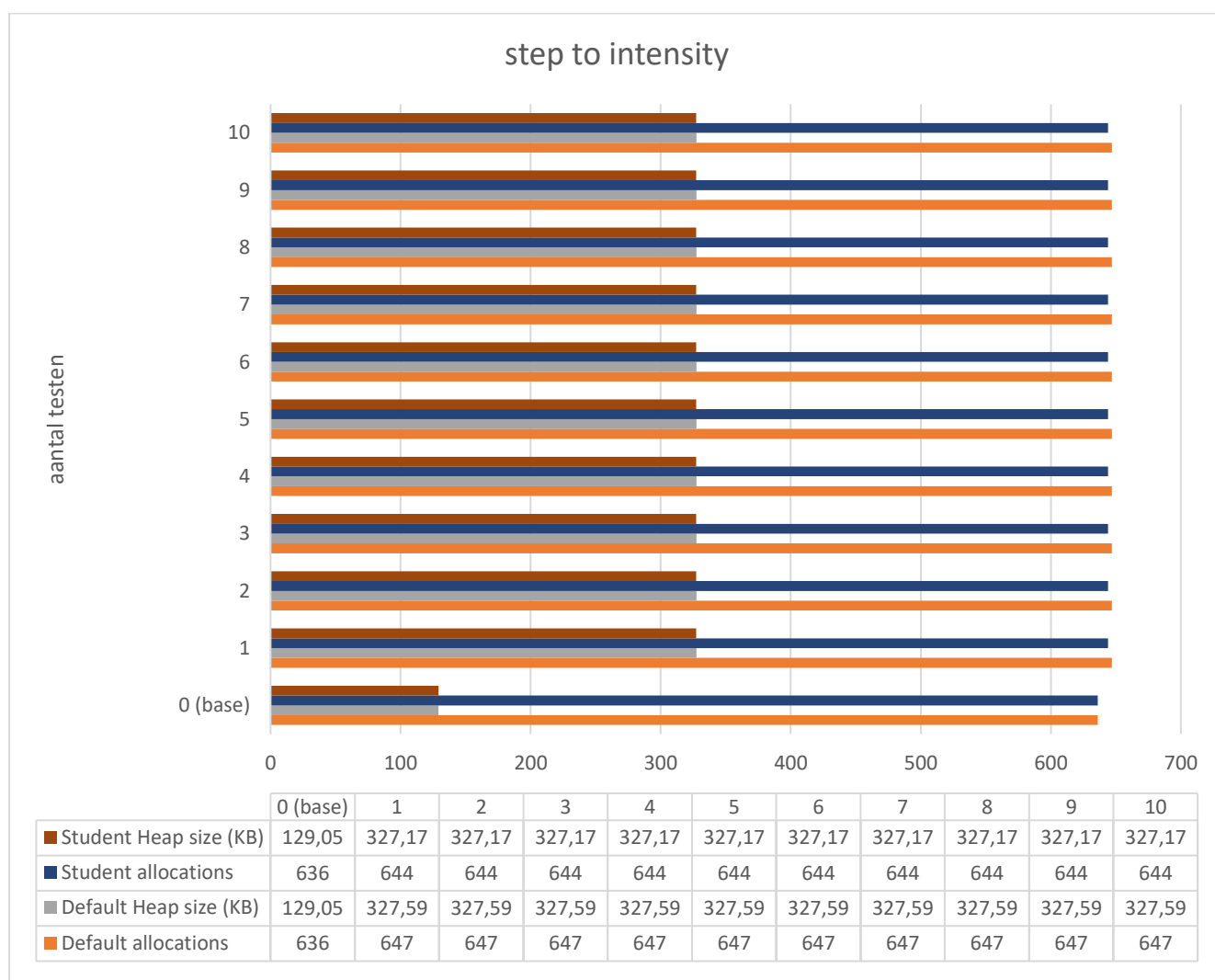
Wij verwachten tot onze implementatie een stuk minder memory in beslag neemt, omdat het een vrij simpele oplossing is.

## 1.4. Werkwijze

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

## 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 heap size – De student heap size

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