1. Image shell & intensity

1.1. Namen en Datum

Jasper van Poelgeest en Youri Okkerse, 03-04-19

1.2. Doel

Het doel van dit experiment is om aan te tonen dat onze manier ((r + b + g) / 3) net zo snel, dan wel sneller is dan de oorspronkelijke implementatie (DEFAULT).

Onderzoeksvraag:
Is het mogelijk een snellere, dan wel niet even snelle implementatie van de image shell en grayscaling te maken, dan de default implementatie?

1.3. Hypothese

Wij verwachten niet dat het mogelijk is een snellere versie te maken, maar een even snelle versie zou moeten lukken. grayscaling is niet reken heavy, in de zin van het is maar een enkele berekening in de meeste gevallen. Dit zal niet veel efficiënter kunnen. Het kan misschien “mooier” en preciezer, maar het gaat er in dit geval om, dat het functioneert en dat het net zo snel of sneller is dan de default.

1.4. Werkwijze

Na de implementatie van de code om de kleur image naar een gray image om te zetten hebben we met een for-loop 100x de functie gerund. Hierbij hebben we de gemiddelde tijd genomen die het heeft gekost per loop om de image om te zetten.

1.5. Resultaten

Tijd STUDENT implementatie (ms)		**Tijd DEFAULT implementatie (ms)**	
100x	**avg**	**100x**	**avg**
38367	384	38525	385

1.6. Verwerking

Met std:clock is de totale tijd gemeten, vervolgens is deze door 100 gedeeld. Om het gemiddelde per cycle te krijgen.

1.7. Conclusie

De twee implementaties zijn ongeveer even snel, 1 ms verschil op 100 cycles is bijna niets en al helemaal niet merkbaar als mens.

1.8. Evaluatie

Doordat er meerdere programma's op de computer aan het runnen zijn kan het zijn dat als je de ene implementatie runt die vaker door het OS op halt wordt gezet dan de andere implementatie. Dit hangt er maar net vanaf wat die programma's op dat moment willen doen. Wij hebben dit zo veel mogelijk proberen te voorkomen, door niet op de computer te doen terwijl hij aan het runnen was.