



IMPLEMENTATIEPLAN IMAGESHELL

Jasper Oosterbroek, Jens Bouman
22-2-2019

Inhoud

Doel	2
Methoden	2
Keuze.....	Fout! Bladwijzer niet gedefinieerd.
Implementatie.....	Fout! Bladwijzer niet gedefinieerd.
Evaluatie.....	3

Doel

In dit implementatieplan worden keuzes en methoden toegelicht voor het implementeren van een imageshell. Een imageshell dient als een huls voor een afbeelding, met extra functies (het veranderen van een pixel bijvoorbeeld). Deze shell moet voor zowel intensity als RGB gemaakt worden.

Methoden

Voor het opslaan van de pixel waarden zijn de volgende methoden gevonden.

	Toegankelijkheid	Opzoek snelheid	Ruimte gebruik	Schaalbaar
Één-dimensionale array van pixels	Direct	$O(1)$	N	Ja
Twee-dimensionale array van pixels	Direct	$O(1)$	$>N$	Ja
Linked list	Lineair	$O(N)$	$>N$	Ja
Vector van pixels	Direct	$O(1)$	$>N$	Ja

Keuze

Bij het implementeren is er gekozen voor een eendimensionale array van pixels. Dit is gekozen omdat het voordelen heeft ten opzichte van de andere opties:

- Een tweedimensionale array heeft twee array aanroepen nodig om data op te halen
- Een linked list heeft geen directe toegankelijkheid, waardoor het willekeurig aanpassen van pixels heel traag is
- Een vector van pixels heeft naast de grotere hoeveelheid ruimte ook geen consequent gealloceerd geheugen. De inhoud van de vector staat niet gegarandeerd in opeenvolgende stukken geheugen.

Implementatie

Omdat de eendimensionale array van omvang moet kunnen variëren kan hij niet op de stack toegeschreven worden. Standaard wordt bij arrays op compiletijd het geheugen gealloceerd. Om dit probleem te voorkomen wordt de array op de heap gealloceerd, waarbij de omvang kan veranderen wanneer nodig. Hierbij is het wel van belang dat bij elke nieuwe omvang, de vorige verwijderd wordt, anders krijg je floating pointers.

De ImageShell moet met een bestaande interface werken. Ik zal interessante opmerkingen per functie na gaan:

- `IntensityImageStudent(const IntensityImageStudent &other);`
 - o Moet de bestaande pixels kopiëren
- `~IntensityImageStudent();`
 - o Moet de floating pointer naar de heap deleten
- `void set(const IntensityImageStudent &other);`
 - o Moet de bestaande pixels kopiëren

- void setPixel(int x, int y, Intensity pixel);
 - o Moet i berekenen als volgt: $x + width * y$

Evaluatie

Om te testen of de implementatie voldoet zal de GUI gebruikt worden om te achterhalen of de stappen correct gevolgd worden. Hiernaast zal een test qua snelheid gemaakt worden.