# Implementatieplan Opdracht 1

Jeroen Kok & Wilco Louwerse  
18-5-16

## Doel

Het doel van deze opdracht is om zelf een implementatie te geven aan de ImageShell voor RGB en IntesityImage. Gevolgt door een eigen implementatie voor de conversie van RGBImage naar IntensityImage

## Methoden

Voor de ImageShell was het nodig om pixel op te slaan in een storage, daarvoor konden wij gebruik maken van een aantal soorten storages, zoals: C-array, C++-array, map en een vector.  
De C en C++-array lijken heel erg op elkaar, maar een C++-array is veiliger in gebruik maar heeft wel enkele bytes overhead. Een vector is een C++-array dat kan groeien of krimpen op basis van de gewilde grote, terwijl je bij een C of C++-array van te voren een grote moet meegeven. Dit kan bij het gebruik van een vector echter wel lijden tot grote hoeveelheden aan overhead omdat het groeien niet altijd optimaal gebeurd. Als laatste optie is er nog het gebruiken van een map, of hashtable. Het gebruiken van een map zorgt echter voor nog grotere hoeveelheden aan overhead omdat naast de pixel ook de ‘positie’ van de pixel in de hashtable moet worden opgeslagen.

Voor de conversie van RGB naar Grayscale hadden we de keuzen uit het gemiddelde nemen van de RGB waarde en die te gebruiken als de nieuwe intensity. Ook is het mogelijk om een gewogen gemiddelde te nemen waarbij je de R, G of B waarde meer of minder laat meetellen dan de andere maar nog steeds wel een gemiddelde neemt van alle waarden.

## Keuze & Implementatie

Voor de opslag hebben we gekozen voor een 2D C-array, omdat een map te veel overheads zal gaan gebruiken en een C++-array en een vector niet wilde werken omdat je de grote voor de array at compile time moet meegeven en niet at run time kunt meegeven, wat voor de nodige problemen zorgden.

Voor de conversie van RGB naar Grayscale hebben we gekozen om gewoon het gemiddelde te nemen omdat we geen andere reden zagen om een ingewikkeldere methode te gebruiken en deze methode de meest eerlijke/duidelijke waardes geeft, wat dus zorgt voor de meest duidelijke zwart-wit afbeelding

## Evaluatie

Om deze implementatie te testen maken we gebruik van trial and error. Dit gaan we doen door de gui te runnen met de nieuwe door ons geïmplementeerde code en de resultaten te vergelijken met die van de default implementatie.