Verslag sorting en boggle opgave

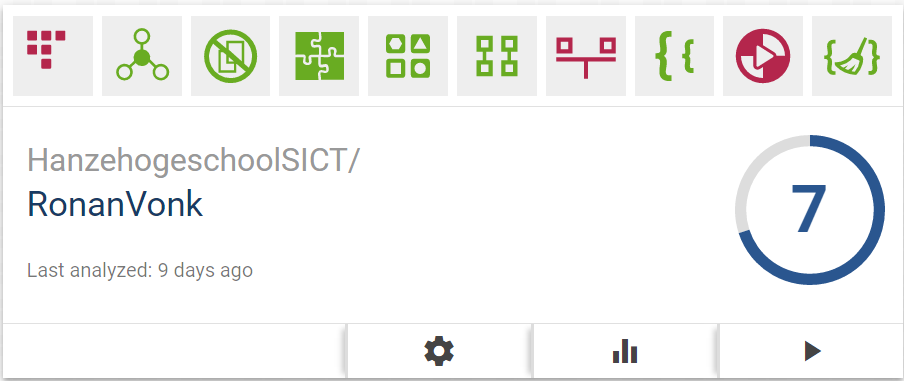
Door: Ronan Vonk

# Inleiding

In dit verslag staat reflectie omtrent de sorting en boggle opgave. Deze opgave vereiste de studenten een applicatie te schrijven die een drietal sorteeralgoritmen inzichtelijk maakt. De tweede applicatie moest in staat zijn boggle automatisch te spelen door middel van een algoritme. Dit verslag behandelt specifiek de resultaten van BetterCodeHub.

# BetterCodeHub sorting opgave

BetterCodeHub is gebruikt om de code automatisch te analyseren en een cijfer te geven aan de gegeven code. Het resultaat voor de sorting opgave was als volgt:



De onderdelen waarvoor de code een onvoldoende ontving waren:

* Schrijf korte eenheden code
* Houd componenten gebalanceerd
* Automatiseer tests

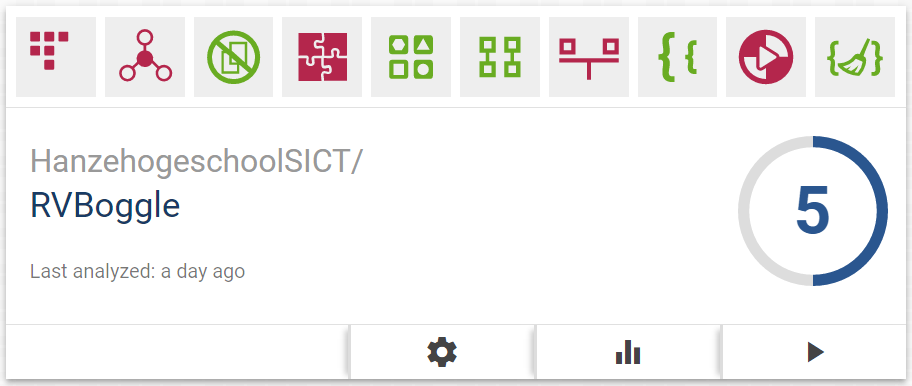
Ten eerste het schrijven van korte eenheden van code is voor het grootste deel gelukt, echter zijn de sorteermethoden te lang om een voldoende aan deze beoordeling te halen. Dit zou opgelost kunnen worden door gedeelten van de code, zoals lange vergelijkingen, in andere methoden te zetten.

Ten tweede het balanceren van de componenten, in dit geval staat alle code in een enkele src package. In dit geval gaat het om twee klassen dus het is lastig dit te vermijden, een optie zou zijn om het FXML bestand in een res directory te plaatsen in plaats van bij de code.

Tot slot het automatiseren van tests. Dit punt is onvoldoende in allebei de opgaven omdat het automatiseren van tests overbodig leek voor een tweetal opgaven. Daarom zijn er geen automatische tests aanwezig in deze opgave, en de tweede opgave.

# BetterCodeHub Boggle opgave

De beoordeling voor deze opgave was als volgt:



In dit geval waren de volgende aspecten van de analyse onvoldoende:

* Schrijf korte eenheden code
* Schrijf simpele code eenheden
* Houd eenheid interfaces klein
* Houd componenten gebalanceerd
* Automatiseer tests

Ten eerste, het eerste punt heeft hetzelfde probleem als die van de eerste opgave. In dit geval is er een grote klasse die in principe alle relevante functies uitvoert voor deze opgave. Nogmaals had de FXML verplaatst kunnen worden om dit te balanceren. Daarbij had de grote klasse opgedeeld kunnen worden. Dit is ook voor een grote reden waarom het tweede punt onvoldoende is beoordeeld. De grote klasse in kwestie is te complex en heeft te veel lange methden. Hierdoor word een onvoldoende behaalt op het schrijven van simpele code eenheden en het klein houden van de interfaces. In dit geval had deze applicatie een groot voordeel gehad aan een MVC structuur. Dit zou waarschijnlijk een groot deel van de problemen oplossen.