Universiteit van Amsterdam

Opdracht 2

Programmeerkeuzes bij Check

20 februari 2015

Docent:
Andy Pimentel
Floris Kroon

Student:
Tijmen Zwaan
10208917

Cursus:
Datastructuren

Vakcode:
5062DATA6Y

1 Introductie

De opgave die in dit verslag besproken wordt, is relatief simpel: Lees een bepaalde opstelling van een schaakbord in, en kijk of één van de twee koningen schaak staat.

1.1 Definities

De belangrijkste definities zijn hier de regels van het spel schaak. Daarvoor zou ik u graag hierheen verwijzen.

2 Methode

Om dit voor elkaar te krijgen moet er voor meerdere stukken worden gekeken of ze een van de twee koningen schaak zetten.

Wat ik me als eerste bedacht was dat het checken van alle stukken apart geen efficiente manier zou zijn, dus ik heb ervoor gekozen om alle checks vanuit de koning te doen. Dit geeft uiteindelijk hetzelfde resultaat.

Daarnaast heb ik ervoor gekozen om het bord te representeren met een simpele 2-dimensionale array van characters.

2.1 Algoritme

Het algoritme zoekt eerst de witte koning. Waarna vanuit die koning één voor één de verschillende stukken worden gezocht in hun bijbehorende richtingen.

Voor de loper wordt bijvoorbeeld schuin gekeken, en voor de toren juist recht. Als een overeenkomend stuk wordt gevonden, dan staat de koning schaak, en eindigt het algoritme.
Wordt er een ander of geen stuk gevonden, dan gaat het algoritme door naar de volgende check. Wanneer alle checks gedaan zijn en geen of alleen andere stukken zijn gevonden, dan gaat het algoritme door met de zwarte koning.

Ik heb ervoor gekozen om voor ieder stuk een bijbehorende functie te schrijven die speciaal voor dat stuk checkt. Deze worden één voor één aangeroepen.

3 Resultaten

Het algoritme werkt zoals verwacht. Verder zijn er geen bijzonderheden te noemen.

4 Discussie

Het kwam wat later in mijn hoofd op dat ik nu bepaalde richtingen tweemaal afga. Zoals schuin om te checken of er een loper of een koningin staat. Het zou waarschijnlijk sneller zijn geweest als ik die checks tegelijkertijd had gedaan. Hetzelfde geldt voor de kongingin en de toren.