Notulen

4-11-15

Standaardvariabele:  
statesArchive: tree: het archief waar de verschillende states worden opgeslagen.  
moveOptions: Stack: de verschillende mogelijkheden.  
path: list of list: geeft de nog mogelijke kinderen van de huidige states. Elk element in de lijst stelt een level voor in de tree. Dus een lijst met een state en het aantal mogelijke kinderen.  
Solutionpaths: list of list: een lijst met alle mogelijke oplossingen. De oplossingen bestaan uit een rij aan paths.

De volgende dingen moeten gemaakt worden:

Een functie waarbij er een lijst wordt gegeven met alle mogelijke states waarbij elke auto maximaal 1 stap is verzet.

De tree moet gemaakt worden met een functie add to three and is in three.

Het hele depth-search algoritme.

Stack voor de stappen maken. Als de state al geweest is dan pop je de optie.

Creëer een array? Voor de zetten die je maakt.   
De opties slaan we op in de **archive**(van de coördinaten maken we tegels) we checken hier altijd of dit het antwoord is. Alle opties slaan we op in de **Stack**.  
We kiezen 1 optie en daar gaan we in door. De opties slaan we op in de **Stack.**De nieuwe stappen slaan we op in de stack(bovenop).

30-10-15

Taakverdeling besproken voor het weekend eerstvolgende vergadering is **03-11-2015**.

Pim: De visualisatie afronden. De output van Alex implementeren. De visualisatie updaten.

Alex: Implement can make move maken. Er voor zorgen dat er bewogen kan worden. De restricties om de functie maken. Kijken naar de mogelijke zetten.

Daan: Het depth-first algoritme implementeren in python.

Notitie: oorsprong is linker onderhoek!. De coordinaten worden bepaald door het kortste stuk naar de oorsprong