

# Project Con-tac-tix

## Inleiding:

In dit verslag zullen we mijn implementatie van een AI voor het spel con-tac-tix bespreken, dit is verre van de ideale AI voor dit spel, maar ik heb toch een goede poging ondernomen. Het spel zelf ziet er mij persoonlijk zeer interessant uit en ik zou dit graag eens spelen. Door mijn onbestaande ervaring met dit spel was het te moeilijk voor mij om een heuristiek te vinden voor dit spel, desondanks heb ik toch mijn best gedaan propere en performante AI te maken.

## Interne bord voorstelling

We houden volgende data bij voor een bord:

- de grootte in beide richtingen
- wie aan de beurt is
- wat de oriëntatie is van de 2 spelers
- de huidige status van het bord (dit is een getal, waar -100 en 100 een winst aanduiden en de andere waarden ertussen voor een heuristiek zijn)

## Algoritme

We maken gebruik van min-max bomen die tot een diepte van 4 zoeken naar een winnend spel voor de huidige speler.

## Memoization

Het concept van memoizatie is niks nieuws, in plaats van een functie met dezelfde argumenten steeds te herberekenen sla je het resultaat op om dit later te kunnen teruggeven. Men moet goed opletten op welke functies men memoizatie toepast, net zoals bij multithreading is het mogelijk hierdoor een overhead te creëren die ervoor zorgt dat het algemeen effect van memoizatie een vertraging is van het geheel en geen versnelling.

Prolog maakt memoizatie mogelijk binnen de taal zelf (vroeger was dit in de library tabling, maar deze zit nu in prolog zelf), dit door gebruik te maken van tabling, voor een simpele memoizatie volstaat het `:- table <predicaat>.` uit te voeren.

Om het effect van memoization te zien voeren we `tests/run_tests.sh` uit en bekijken we hoe lang dit duurt met of zonder memoization per propositie

Propositie	Tijd zonder	Tijd met	Percentage met/zonder
floodfill/2	228s	67s	29.4%

We zien dat floodfill goed gebruik kan maken van de memoizatie, dit is al-

licht omdat floodfill zichzelf vele malen gebruikt, dit valt binnen de lijn van verwachtingen omdat floodfill recursief is.

## **Conclusie**

- Een AI maken met bomen is niet altijd even makkelijk, spelbomen kunnen heel snel heel groot worden en dit kan een probleem vormen, zeker als je geen heuristiek vind.
- Memoizatie kan bepaalde soorten functies significant versnellen