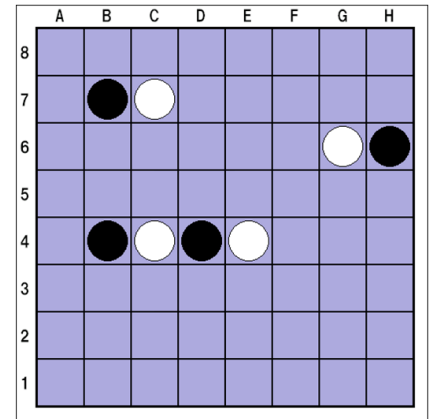
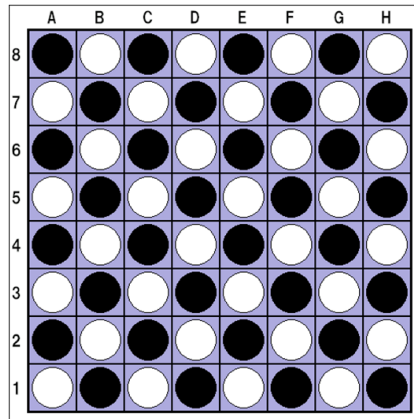


Kunstmatige intelligentie

Programmeeropgave 1 van 2023 — Clobber

De eerste programmeeropgave (in het voorjaar van 2023) behorende bij het vak [Kunstmatige intelligentie](#) gaat over het spel *Clobber*. Het is de bedoeling een eenvoudig programma te schrijven dat dit spel in de twee, drie- of meerpersoons versie zo goed mogelijk speelt.



Op een rechthoekig m bij n bord bevatten de niet-lege vakjes een witte of zwarte schijf, en in de versie met meer spelers een "schijf" met 0, 1, 2, ... De beginstand kan willekeurig zijn, of regelmatig. Als een speler aan de beurt is (ze spelen om en om: wit-zwart-wit... of 0-1-2-3-0-1-2...) moet deze met een steen van de eigen kleur een horizontaal of verticaal direct aangrenzende steen van de/een andere kleur pakken (slaan). Als een speler niet kan zetten moet deze passen. Bij de tweepersoons versie: zodra een speler past, heeft de ander gewonnen. Bij de meerpersoons versie: als alle spelers moeten passen heeft de laatste die een zet deed gewonnen. (Overigens: als hier een speler moet passen, kan deze nooit meer.)

Een leuke publicatie over Clobber: J. Griebel en J. Uiterwijk, Combining Combinatorial Game Theory with an alpha-beta Solver for Clobber, Proceedings BNAIC 2016, pp. 48-55; [link](#).

Gebruik het [Clobber-voorbeeldprogramma](#) van de website.

Dit programma is geschreven in eenvoudig C++, zoals behandeld bij het college [Programmeermethoden](#). Daar staan ook korte video's, bijvoorbeeld over het installeren van een C++-compiler: [Code::Blocks](#) OF [Mac](#) (zie ook [hier](#)) OF [Linux](#). En een [stoomcursus C++](#) met [bijbehorende sheets](#).

Bestudeer nu AI-spelers voor twee, drie of meer spelers:

- EEN: speelt volledig random, zie de [voorbeeld-code](#). (Tip: kies optie 66 voor MC-speler, dan speelt iedereen random.)
- TWEE: een Monte Carlo speler.
Speel voor alle mogelijke zetten een aantal random spelletjes (de "playouts", zeg 1000; met gebruik van de random-speler), en kies daaruit de "beste".
- DRIE: bedenk zelf een (eenvoudige) strategie.
Een idee is bijvoorbeeld het volgende. Loop alle mogelijkheden af, speel per mogelijkheid in een kopie van het bord de betreffende zet (via `clobber kopie = *this;`), en eventueel alle mogelijke zwarte tegenzetten (bij twee spelers), beoordeel de stand, en selecteer tot slot de "beste" zet. Denk niet verder vooruit! Gebruik geen recursie! Kijk bijvoorbeeld naar het aantal mogelijke zetten voor de andere spelers. Of: hoeveel "actieve" stenen heeft de huidige speler nog, al dan niet "gedekt"?

Probeer dit zo simpel mogelijk te programmeren. Noem in het verslag ook andere aanpakken, bijvoorbeeld een eindspel-database, met referenties.

Laat dus de spelers spelen, en rapporteer de bevindingen (gemiddeldes, standaard-deviaties, grafieken).

Kortom: maak diverse simulaties; varieer onder meer de grootte van het bord, en het aantal spelers, en hun strategieën. Er mag vrijelijk —met bronvermelding— gebruik gemaakt worden van de [voorbeeld-code](#).

Er wordt niet gevraagd om een programma dat aan alle software engineering standaards voldoet. Houd het programma dus eenvoudig — maar wel netjes!

Gebruik [gnuplot](#) om voor het verslag enkele grafieken te plotten.

Tips: maak een plot-file `plot.gpi` met daarin bijvoorbeeld

```
# plot-file voor gnuplot
set xrange [0:10]
set yrange [17:42]
set xlabel "iets"
set ylabel "iets anders"
plot "iets.uit" using 1:2 with lines notitle
```

en roep deze binnen gnuplot aan met `load "plot.gpi"`. Gebruik de export-mogelijkheden om pdf-uitvoer te genereren.

Als je een file met resultaten hebt (zeg `iets.uit`; twee getallen per regel: de x- en y-coördinaten van een punt):

```
1 24
2 31
3 56
4 7
```

kun je deze eenvoudig met gnuplot in een grafiek printen met `plot "iets.uit" with lines`.

Op internet staat alles en meer over gnuplot, bijvoorbeeld [hier](#) of [daar](#).

Deadline: woensdag 1 maart 2023, 13:15 uur.

In te leveren: een *geprint* exemplaar van het verslag (alleen *eigen* C++-code in de appendix) tijdens het college, en de C++-code van het programma via Brightspace => Assignments. Het verslag moet aan verschillende [eisen](#) voldoen.

Er is een klein skelet voor de verslagen: [verslag.tex](#), dat deze [PDF-versie](#) op kan leveren. Zie ook [diverse opmerkingen](#).

Vragen en/of opmerkingen kunnen worden gestuurd naar: aicollege@liacs.leidenuniv.nl.

6 februari 2023 — <http://www.liacs.leidenuniv.nl/~kosterswa/AI/clobber2023.html>