

AI

---

## Kunstmatige Intelligentie (AI)



Universiteit  
Leiden

Walter Kosters

voorjaar 2024

[www.liacs.leidenuniv.nl/~kosterswa/AI/aieen.pdf](http://www.liacs.leidenuniv.nl/~kosterswa/AI/aieen.pdf)

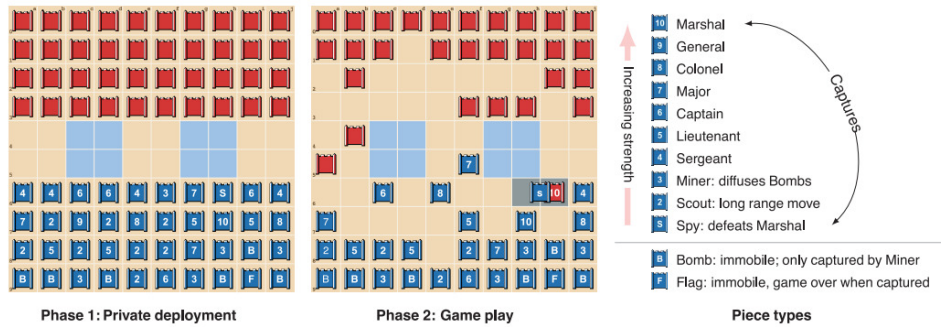
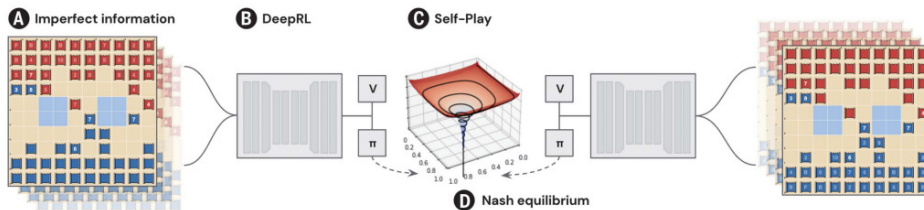


Fig. 1. Stratego is a two-player board game in which players try to capture the opponent's flag. Initially, the players secretly deploy 40 pieces of diverse strengths on the board. Then, they take turns moving pieces, possibly encountering an opponent piece that reveals both player identities, and then the weaker piece is removed. Two lakes (indicated in blue) cannot be crossed by any piece. The complete rules are defined by the International Stratego Federation.



Replicator dynamics:  $\frac{d}{dt} \pi_r^i(a^i) = \pi_r^i(a^i) [Q_{\pi_r}^i(a^i) - \sum_{b^i} \pi_r^i(b^i) Q_{\pi_r}^i(b^i)]$   
 Reward transformation:  $r^i(\pi^i, \pi^{-i}, a^i, a^{-i}) = r^i(a^i, a^{-i}) - \eta \log \left( \frac{\pi_r^i(a^i)}{\pi_{reg}^i(a^i)} \right) + \eta \log \left( \frac{\pi_r^{-i}(a^{-i})}{\pi_{reg}^{-i}(a^{-i})} \right)$

breaking news



[link](#)

RESEARCH

MACHINE LEARNING

**Mastering the game of Stratego with model-free multiagent reinforcement learning**

Julien Perolat<sup>\*†</sup>, Bart De Vylder<sup>\*†</sup>, Daniel Hennes, Eugene Tarassov, Florian Strub, Vincent de Boer<sup>‡</sup>, Paul Muller, Jerome T. Connor, Neil Burch, Thomas Anthony, Stephen McAleer, Romuald Elie, Sarah H. Cen, Zhe Wang, Audrunas Gruslys, Aleksandra Malysheva, Mina Khan, Sherjil Ozair, Finbarr Timbers, Toby Pohlen, Tom Eccles, Mark Rowland, Marc Lanctot, Jean-Baptiste Lespiau, Bilal Piot, Shayegan Omidshafiei, Edward Lockhart, Laurent Sifre, Nathalie Beauguerlange, Remi Munos, David Silver, Satinder Singh, Demis Hassabis, Karl Tuyls<sup>\*†</sup>

We introduce DeepNash, an autonomous agent that plays the imperfect information game Stratego at a human expert level. Stratego is one of the few iconic board games that artificial intelligence (AI) has not yet mastered. It is a game characterized by a twin challenge: It requires long-term strategic thinking as in chess, but it also requires dealing with imperfect information as in poker. The technique underpinning DeepNash uses a game-theoretic, model-free deep reinforcement learning method, without search, that learns to master Stratego through self-play from scratch. DeepNash beat existing state-of-the-art AI methods in Stratego and achieved a year-to-date (2022) and all-time top-three ranking on the Gravon games platform, competing with human expert players.



Perolat et al., Science 378 (2022) 990–996: Mastering the game of Stratego . . .

Deep neural nets, reinforcement learning, selfplay.

Kunnen computers denken? Diplomacy! En ChatGPT?

## Huishoudelijke mededelingen

Lectures are in DUTCH!



De **colleges** zijn op woensdagen, 7 februari tot en met 22 mei 2024 (niet op 27 maart), 13:15–15:00 uur, Gorlaeus zaal **C1**. De colleges zijn ook achteraf op video te bekijken, zie de website — ook voor oudere video's.

En de **werkcolleges**: donderdagen, 8 februari tot en met 23 mei 2024 (niet op 28 maart en 9 mei), 13:15–15:00 uur, in de computerzalen **302/306/307** van het Snellius.

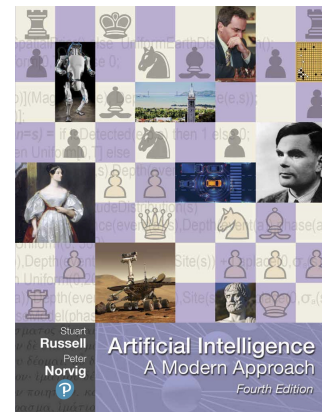
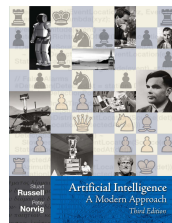
**Schriftelijk tentamen (tijdig aanmelden in MyStudyMap):**

- donderdag 13 juni 2024, 9:00–12:00 uur;  
Universitair Sportcentrum
- hertentamen: maandag 8 juli 2024, 9:00–12:00 uur;  
Universitair Sportcentrum

We maken gebruik van het volgende [boek](#):

Stuart J. Russell en Peter Norvig  
Artificial Intelligence, A Modern Approach  
fourth edition  
Pearson, 2020

Afkorting: [RN]. Er zijn vele andere boeken, zie de website.



Het **practicum** kent **vier** opgaven met **strikte deadlines**:

- 28 februari 2024, 13:15 uur: [Monte Carlo en Tetris](#)
- 20 maart, 13:15 uur: Agenten / Robots
- 17 april, 13:15 uur: A\*
- 13 mei, 13:15 uur: Neuraal netwerk

Assistentie: Raaf van Beusekom, Xander Lenstra. Justin de Rooij en Michael de Rooij.

We gebruiken de programmeertaal C++, via Linux of **Windows-WSL2** of Windows-Code::Blocks of de Mac, zie:

[www.liacs.leidenuniv.nl/~kosterswa/pm/](http://www.liacs.leidenuniv.nl/~kosterswa/pm/)

Het eindcijfer  $E$  is het cijfer  $T$  van het schriftelijk tentamen, *mits* de vier practicumopgaven alle als voldoende beoordeeld zijn; 6 EC. Als  $T \leq 5$  is dit het eindcijfer.

Als het gemiddelde practicumcijfer  $G$  groter is dan  $T$  en  $T \geq 5.5$ , wordt het eindcijfer  $E$  hun gemiddelde,  $(G + T)/2$ , waarbij dit maximaal 1 punt meer mag worden dan  $T$ .

En  $E$  wordt afgerond naar het dichtstbijzijnde getal uit de verzameling  $\{1, 2, 3, 4, 5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10\}$ .

Bij elke practicumopgave, te maken in  $\leq$ tweetallen, moet een circa zes pagina's tellend **verslag** in  $\text{\LaTeX}$  gemaakt worden (zie [voorbeeldfile](#) op de website). Globale opzet:

1. Inleiding
2. Uitleg probleem
3. Relevant werk (= Related work)
4. Aanpak
5. Implementatie (kort!)
6. Experimenten (tabel, grafiek)
7. Conclusie (inclusief verder onderzoek)

Referenties

Appendix: programma — eigen code



Het tekstverwerkingsysteem L<sup>A</sup>T<sub>E</sub>X is gebaseerd op Donald Knuth's T<sub>E</sub>X, en draait op “iedere” computer.

Je maakt met een editor naar keuze een ASCII-file `iets.tex` met je eigen tekst, “`latex-t`” deze met `pdflatex iets.tex` naar een printbare PDF-file `iets.pdf`.

Internet-versie: [www.overleaf.com](http://www.overleaf.com)

Handleiding:

[www.liacs.leidenuniv.nl/~kosterswa/stuva1/lshort.pdf](http://www.liacs.leidenuniv.nl/~kosterswa/stuva1/lshort.pdf)

met name Hoofdstuk 1, 2 en 3 (plaatjes: 4)

*of*: lees `sample2e.tex`

```
% commentaar: voorbeeldfile
\documentclass[12pt]{article}
\author{P.~Puk}
\title{Kabouters in de Tweede Kamer}
\frenchspacing

\begin{document}  %\maketitle
\section*{Verkiezingen}
In 2024--25 w\ 'e\ 'er \emph{ver-kie\-zingen}
--- geldt  $a^n - b_n = c \times n$ ?
\end{document}
```

---

## Verkiezingen

In 2024–25 wéér *ver-kiezingen* — geldt  $a^n - b_n = c \times n$ ?

#	Datum	Onderwerp	Deadline
1	7.2	<a href="#">Algemene introductie</a> [RN1]	
2	14.2	<a href="#">Intelligente agenten</a> [RN2]	
3	21.2	<a href="#">Logische agenten</a> [RN7;8]	
4	28.2	<a href="#">Probleemoplossen en zoeken</a> [RN3]	Opgave 1
5	6.3	<a href="#">Geinformeerd zoeken</a> [RN4]	
6	13.3	<a href="#">Spel(l)en</a> [RN5]	
7	20.3	idem, vervolg	Opgave 2
8	3.4	<a href="#">CSP's</a> [RN6]	
9	10.4	<a href="#">Leren</a> [RN19]	
10	17.4	<a href="#">Deep learning</a> [RN21]	Opgave 3
11	24.4	idem, vervolg: Reinforcement Learning [RN22]	
12	1.5	<a href="#">Locaal zoeken en Optimalisatie</a> [RN4.1]	
13	8.5	<a href="#">Bayesiaanse netwerken</a> [RN12;13] (geen werkcollege)	
14	15.5	Oude tentamens (geen (werk)college 22/23.5)	Opgave 4

Optioneel: Natural Language Processing [RN23,24], Robotica [RN26].

werkcollege "sommen op papier"

Idee:

- De werkcolleges zijn bedoeld om aan de programmeer-opgaven en de bijbehorende verslagen te werken.
- Vier werkcolleges bestaan deels uit het maken van sommen: 7 maart, 4 april, 2 mei en 16 mei.
- Tijdens de werkcolleges: vragen stellen, samenwerken, peer review, milestones, ...
- Er is **aanwezigheidscontrole**.

## Informele introductie



**Kunstmatige intelligentie** is een verzamelnaam voor een heel breed vakgebied, met onder andere:

- *robotica*: Hoe programmeer je een robot?
- *data mining*: Wat is verborgen in/op WikiLeaks?
- *rechtspraak*: Kan een machine rechter zijn? **NLP**  
↙
- *linguïstiek*: “the spirit is willing but the flesh is weak”  
→ ... → “the vodka is good but the meat is rotten”?
- *computer games*: Kan een computer Fortnite spelen?
- *neurale netwerken*: Kun je beurskoersen voorspellen?
- *cognitie*: Kunnen we het menselijk brein nadoen?

Je kunt op minstens **twee** manieren naar Kunstmatige intelligentie kijken:

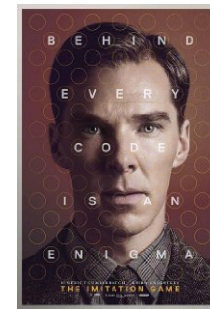
1. vanuit een meer *psychologische* of *filosofische* richting:  
Wat is het verschil tussen een mens en een computer?  
Kan een computer denken?
2. vanuit een meer *technische* richting:  
Hoe werkt een schaakprogramma?  
Hoe werkt een Marsrobot?

“Do androids dream of electric sheep?” →



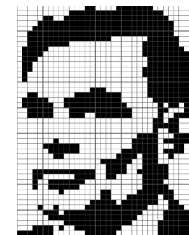
Kunstmatige intelligentie laat computers zich zo gedragen dat het **intelligent** zou heten als mensen het op die manier zouden doen.

De beroemde **Turing-test** uit 1950 vraagt (“the imitation game”):



In een afgesloten kamer bevindt zich een mens *of* een computer, waarmee we alleen via toetsenbord en beeldscherm contact hebben.

Is het een mens of juist een computer?



Het originele probleem was overigens met man ↔ vrouw.



IBM heeft in 2011 een computer “Jeopardy!” laten spelen:

1990 POP CULTURE	10' CANNA	KNOW YOUR BORDERS	WHAT'S YOUR SIGN	FLY LIKE AN EAGLE	NATIONAL PASTRIES
\$100	\$100	\$100	\$100	\$100	\$100
\$200	\$200	\$200	\$200	\$200	\$200
\$300	\$300	\$300	\$300	\$300	\$300
\$400	\$400	\$400	\$400	\$400	\$400
\$500	\$500	\$500	\$500	\$500	\$500

**IN 2013 ROB FORD,  
MAYOR OF THIS 4th-  
LARGEST CITY IN N.  
AMERICA, FIRST SAID  
HE SMOKED WEED,  
NOT CRACK...THEN  
YES, OK, CRACK, TOO**



What is  
**Toronto????**

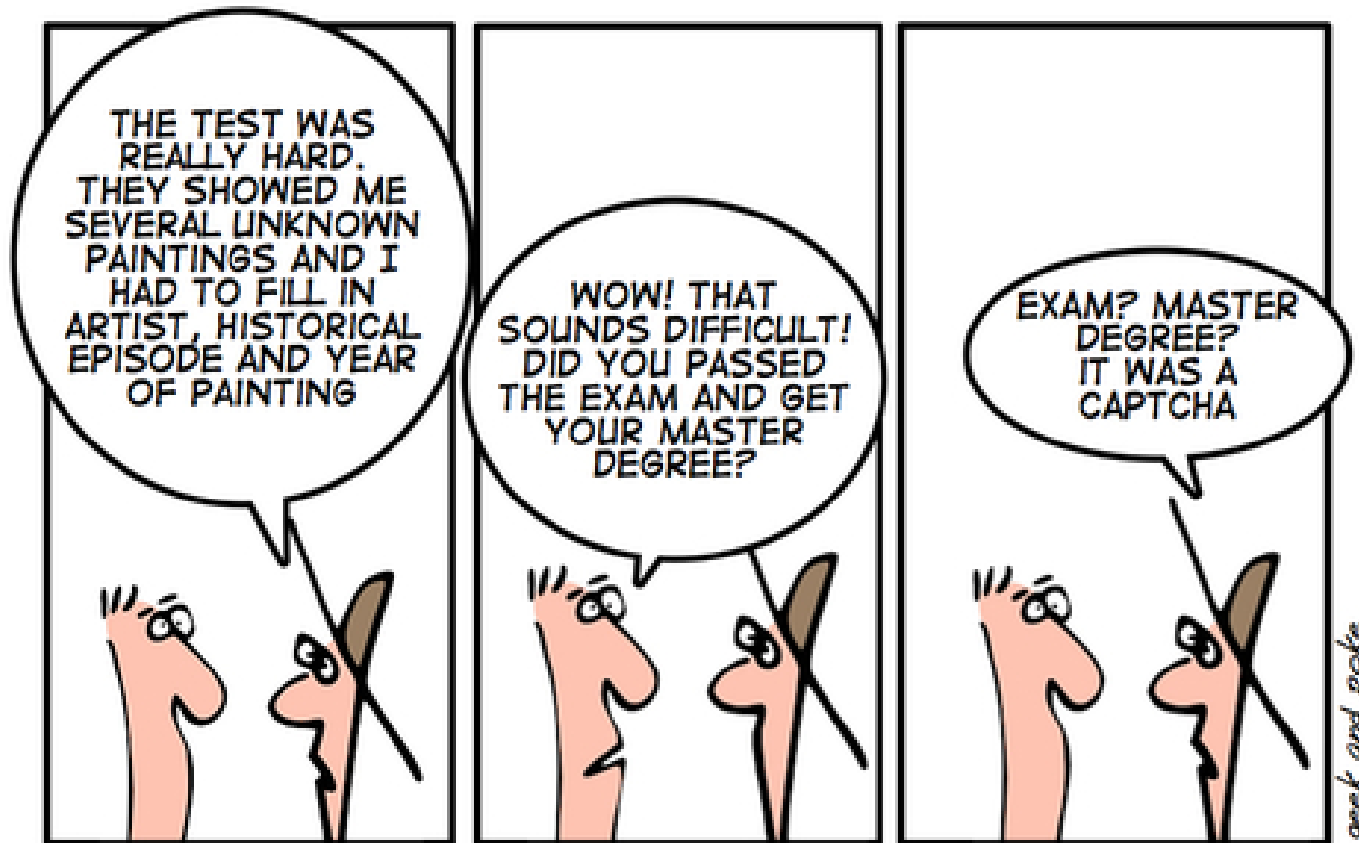


In 1964–66 schreef Joseph Weizenbaum van het MIT in Boston het beroemde programma **ELIZA/DOCTOR**, dat een psychotherapeut “speelt” — met veel succes.  
Een voorbeeldgesprek (Jason Hutchens’ HAL/HeX, 1997):

my name is bob  
Pleased to meet you.  
how are you doing, jas?  
I am just fine, thanks! And you?  
i am feeling great!  
If you are feeling great, it’s your business, mate!



Dit soort computerprogramma's doet mee in de jaarlijkse [Loebner Prize](#).



elzoufzi  
keyetr  
mathics  
nmespda  
ommarie  
ntednBus

IN THE FUTURE SOPHISTICATED CAPTCHAS WILL LOCK OUT ANY BOT

Een **robot** is een “actieve, kunstmatige agent wiens omgeving de fysieke wereld” is. Het woord stamt uit 1921 (of eerder), en is gemaakt door de Tsjechische broers Čapek. Nu kennen we ook **softbots**: internet programma’s.

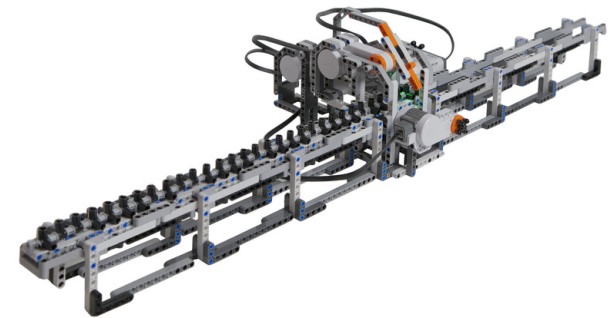
Van de science fiction schrijver Isaac Asimov (auteur van “I, Robot” ) zijn de drie wetten van de **robotica**:

1. Een robot mag een mens geen kwaad doen.
2. Een robot moet menselijke orders gehoorzamen (tenzij dat tegen 1. ingaat).
3. Een robot moet zichzelf beschermen (tenzij dat tegen 1. of 2. ingaat).

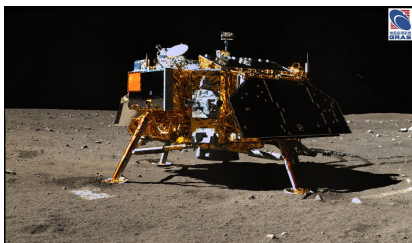


Een simpel programma voor een **Lego**-robot, dat deze random laat lopen, is:

```
task main ( ) {  
  while ( true ) {  
    OnFwd (OUT_A + OUT_C);  
    Wait (Random (100) + 40);  
    OnRev (OUT_A);  
    Wait (Random (85) + 30);  
  }  
}
```



Lego Turing machine



Chang'e-3 Moon lander



Pepper robot van Softbank (2016)

**Maxi** en **Mini** spelen het volgende eenvoudige spel: **Maxi** wijst eerst een (horizontale) rij aan, en daarna kiest **Mini** een (verticale) kolom:

	3	12	8
	2	4	6
①	14	5	2

②

Bijvoorbeeld: **Maxi** ① kiest rij 3, daarna kiest **Mini** ② kolom 2; dat levert einduitslag 5.

**Maxi** wil graag een zo groot mogelijk getal, **Mini** juist een zo klein mogelijk getal.

Hoe analyseren we dit?

Als **Maxi** rij 1 kiest, kiest **Mini** kolom 1 (levert 3); als **Maxi** rij 2 kiest, kiest **Mini** kolom 1 (levert 2); als **Maxi** rij 3 kiest, kiest **Mini** kolom 3 (levert 2). Dus kiest **Maxi** rij 1!

3	12	8
2	?	?
14	5	2

Nu merken we op dat de analyse hetzelfde verloopt als we niet eens weten wat onder de twee vraagtekens zit. Het  $\alpha$ - $\beta$ -algoritme onthoudt als het ware de beste en slechtste mogelijkheden, en kijkt niet verder als dat toch nergens meer toe kan leiden.

Ieder (?) schaakprogramma gebruikt deze methode.

Ook aan een spel als **Tetris** kleven allerlei vragen:

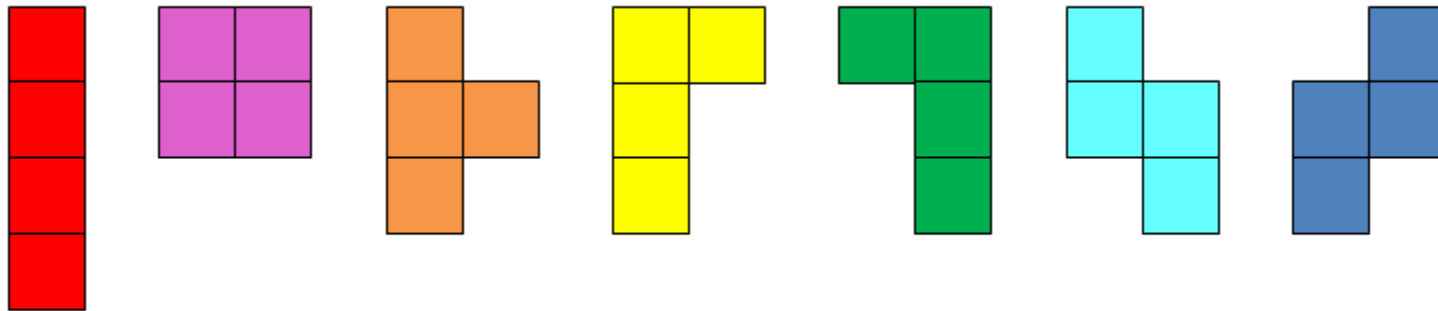
- Hoe speel je het zo goed mogelijk? (AI)
- Hoe moeilijk is het? (complexiteit)
- Wat kan er allemaal gebeuren?

Zo is bijvoorbeeld bewezen dat sommige Tetris-problemen **NP-volledig** zijn, dat je bijna alle configuraties kunt bereiken, maar dat niet alle problemen “beslisbaar” zijn, zie:

[www.liacs.leidenuniv.nl/~kosterswa/tetris/](http://www.liacs.leidenuniv.nl/~kosterswa/tetris/)



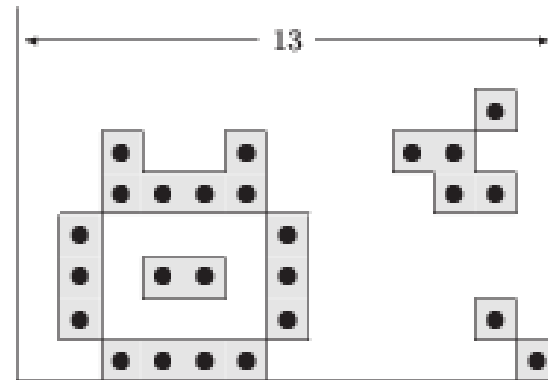
De 7 Tetris-stukken:



Stukken vallen random; volle regels worden verwijderd. De vraag “Kun je met een gegeven serie (inclusief volgorde) van deze stukken een bord helemaal leeg spelen?” is **NP-volledig**.

Als iemand het bord leeg speelt kun je dat eenvoudig controleren. Als het *niet* kan, kan men (tot nu toe) niks beters verzinnen dan alle mogelijkheden één voor één na te gaan!

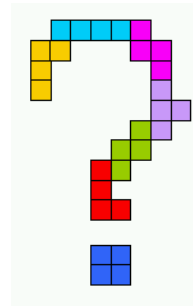
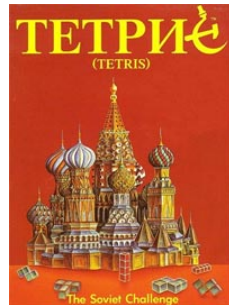
Een “willekeurige” configuratie:



Deze kan gemaakt worden door 276 geschikte Tetris-stukken op de juiste plaats te laten vallen, zie de website.  
Let op: alleen geheel gevulde regels verdwijnen, alles daarboven zakt *één rij*.

Claim: op een bord van oneven breedte kan elke configuratie bereikt worden!

De eerste programmeeropgave (deadline: 28 februari 2024, 13:15 uur; denk ook aan het verslag) gaat over Tetris.



[YouTube](#)

Schrijf een C++-programma dat redelijk Tetris speelt. Gebruik verschillende eenvoudige ‘strategieën’ (random, Monte Carlo, slim), en beschrijf experimenten.

Zie verder, ook voor voorbeeldcode:

[www.liacs.leidenuniv.nl/~kosterswa/AI/teet2024.html](http://www.liacs.leidenuniv.nl/~kosterswa/AI/teet2024.html)

Bij de Monte Carlo (MC) techniek laat je voor elk van de mogelijke zetten een aantal (1000 (?), de “payouts”) potjes *random* (?) tot het eind doorspelen. Je kiest dan de zet met de hoogste gemiddelde (?) uitkomst. Dit heet *Pure Monte Carlo search*.



Thema werkcollege 8.2: MC; 15.2: “slim”; 22.2: verslag.

1. Wat is het verschil tussen een mens en een computer?
2. Kan een computer denken?
3. Hoe werkt een Marsrobot?
4. Hoe werkt een vertaalprogramma?
5. Hoe bedenkt een computer een zet bij vier-op-een-rij?

En **schaken**? En **Diplomacy**? En **ChatGPT**? En **Tetris**?

## Series introduction



## Kunstmatige Intelligentie (AI)

Hoofdstuk 1 van Russell/Norvig = [RN]  
Introdactie

voorjaar 2024  
College 1, 7 februari 2024

[www.liacs.leidenuniv.nl/~koterswa/AI/](http://www.liacs.leidenuniv.nl/~koterswa/AI/)

Er zijn vele verschillende definities van AI. Ze vallen in vier categorieën uiteen:

Systemen die  
denken als mensen

Systemen die  
redelijk denken

Systemen die  
handelen als mensen

Systemen die  
redelijk handelen

Boven: gedachten-processen, redeneren; onder: gedrag.  
Links: meet succes af aan mens; rechts: aan ideaal beeld.

Voorbeeld: een autonome auto.

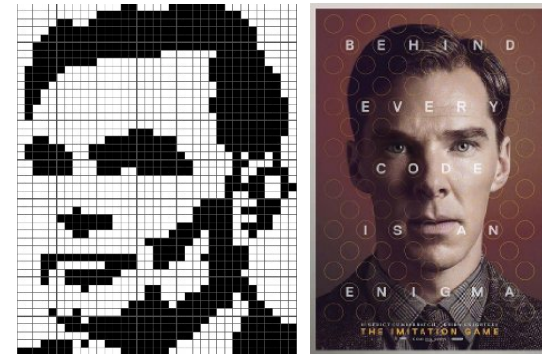




Als je de definitie “Systemen die handelen (**act**) als mensen” aanhangt, kom je bij de **Turing-test** — zie ook eerder.

Een computer slaagt voor deze test als hij/zij/het voor een menselijke ondervrager (met “typemachine-contact”) niet van een mens te onderscheiden is. Noodzakelijk:

- verwerking van natuurlijke taal
- kennis-representatie
- geautomatiseerd redeneren
- machine learning



**Alan Turing, 1912–1954**

Voor de totale Turing-test (met video, en doorgeven van fysieke voorwerpen) komen daar nog bij: computer-zien en robotica.

De Turing-test lijkt op het beroemde **Chinese kamer** gedachten-experiment van **John Searle** uit 1980.

Hierbij zit iemand in een kamer, krijgt geschreven Chinese opdrachten, zoekt in een (groot) boek het juiste “antwoord” — inclusief bijvoorbeeld manipulaties met stapeltjes en pennen —, en geeft dit door naar de buitenwereld.

Volgens Searle betekent het correct verwerken van het juiste programma niet noodzakelijk begrip, of bewustzijn.



En **ChatGPT**?

Een **agent** is iets wat handelt. Een **rationele agent** is iets wat handelt om de beste (verwachte) uitkomst te bereiken. Er is dus een doel.

AI “is” dan de studie van dit soort agenten.

We streven daarbij naar *perfecte* rationele agenten. In de praktijk nemen we — door bijvoorbeeld tijdsbeperking — genoegen met wat minder.

In de 21ste eeuw werd AI de studie van agenten die “het goede/juiste doen” (oftwel “do the right thing”).

De grondslagen van AI zijn te vinden in:

- Filosofie
- Wiskunde (algoritme, **Kurt Gödel**'s stelling, NP-volledigheid (Karp, Cook, Garey, Johnson))
- Economie
- Neuro-wetenschap
- Psychologie (behaviorisme van Watson, cognitieve psychologie)
- Informatica (**John von Neumann**, ...)
- Control theorie en Cybernetica (Wiener)
- Linguïstiek (**Noam Chomsky**)



De beginperiode: 1943–1956

1943 — McCulloch en Pitts — artificiële neuronen

1949 — leerregel van Hebb



schaakprogramma's van Shannon en Turing (en de test!)

1951 — Minsky en Edmonds — neurale netwerk computer

1956 — geboorte van de AI: Dartmouth

1956 — Newell en Simon — Logic Theorist (LT)

Groot enthousiasme: 1952–1969

John McCarthy noemde dit het “Look ma, no hands” tijdperk.

1952 — Samuel — checkers

1957 — Newell en Simon — General Problem Solver (GPS)

1958 — McCarthy (MIT, Stanford; 1927–2011) — de taal Lisp, time sharing, Advice Taker

1958 — Minsky (MIT) — microwerelden (Slagle’s SAINT (1963), Evans’ ANALOGY (1968), ...): blokkenwereld (Huffman, Waltz, Winston, Winograd’s SHRDLU, ...)

1960–62 — Widrow en Hoff — adalines; Rosenblatt — perceptrons

Een dosis realiteit: 1966–1973

Spoedig bleek dat alle optimistische claims niet werden waargemaakt. Een beroemd voorbeeld is de volgende vertaling van Engels via Russisch terug naar Engels: “the spirit is willing but the flesh is weak” naar “the vodka is good but the meat is rotten”. Alle subsidiëring in de VS werd in die tijd stop gezet.

Problemen: er werd geen “kennis” opgeslagen, maar louter syntactisch gemanipuleerd; en de echte problemen waren te groot. Zo experimenteerde men al in 1958 (weinig succesvol) met Genetische Algoritmen. Het boek van Minsky en Papert uit 1969 toonde ook zekere grenzen aan.

Daarna vonden/vinden allerlei ontwikkelingen plaats:

1969–1986 — kennis-gebaseerde systemen (zoals expert-systemen: MYCIN (Feigenbaum, . . . ))

1986–. . . — neurale netwerken zijn weer terug en in: . . . Support Vector Machines . . .

1987–. . . — redeneren, “machine learning” (Hidden Markov Modellen (HMM’s), data mining, Bayesiaanse netwerken)

2001–. . . — beschikbaarheid van grote datasets . . . big data . . . data science . . . grote bedrijven & successen

2011–. . . Deep learning (Geoff Hinton et al.), Reinforcement learning, GPT-3 van OpenAI, . . .



In Hoofdstuk 27 van [RN] (in de derde druk Hoofdstuk 26) staat van alles over de filosofische grondslagen van de AI.

John Searle maakt onderscheid tussen de zwakke AI-hypothese (machines kunnen zich wellicht intelligent gedragen) en de sterke (machines kunnen denken). Vergelijk: “Kan een machine vliegen?” en “Kan een machine zwemmen?”. En Alan Turing vond het een “polite convention that everyone thinks”.

Mensen als Ray Kurzweil (The Age of Spiritual Machines, 1999) voorspellen een versmelting van mens en computer: de “Singularity”.

Ethiek?

Positieve en negatieve verwachtingen!

Lees Hoofdstuk 1, p. 1–35 van [RN] (in de derde druk p. 1–30). En — voor de aardigheid — Hoofdstuk 27 (in de derde druk Hoofdstuk 26). En de teksten van Alan Turing en [Jason Hutchens](#), zie de website.

Het huiswerk voor de volgende keer (14 februari 2024): lees **Hoofdstuk 2**, p. 36–62 van [RN] (in de derde druk p. 34–59) door, en bekijk in het bijzonder Figure 2.6.

Denk na over  
[Monte Carlo en Tetris](#).

