

Artificial Intelligence in games

Door Ghanny Alladien, Nils Peters, Wijnand Potthoff en Tim Stolp

Introductie

Een spel heeft vastgestelde regels en een simpel doel, namelijk winnen, en de benodigde stappen zijn vaak eenvoudig weer te geven. Dit zorgt ervoor dat spellen, ongeacht de vorm, een goede omgeving zijn voor het ontwikkelen van een Artificiële Intelligentie. en van de oudste spellen in de wereld, schaken, is dan ook compleet beheerst door de AI Deep Blue, die in 1996 wereldkampioen Garry Kasparov versloeg. Ook in andere spellen als dammen en othello zijn kampioenen verslagen door AI's. Een videospel heeft ook een bepaalde set regels, en zorgt voor een surrogaat voor de echte wereld waarin de AI eindelijk getest kan worden door de consument. Deze levert feedback via reviews, forums en natuurlijk verkoopcijfers. Dit soort enorme testgroepen zijn moeilijk op een andere manier te verkrijgen, en leveren zowel informatie op als financiering voor verdere ontwikkelingen.

Fundamenten

Deep blue, versloeg in 1997 de huidige schaak wereldkampioen Kasparov. Deep blue werkt door met brute kracht alle mogelijkheden te checken. Tegenwoordig heb je schaakprogramma's zoals Rybka, Deep Fritz of Deep Junior die werken met uitgebreidere software.

Watson, kan door middel van een verzameling van databasen een in spreektaal gestelde vraag goed beantwoorden en berekend daarbij ook de betrouwbaarheid van het gevonden antwoord. Als de betrouwbaarheid niet hoog genoeg is dan geeft Watson geen antwoord.

AlphaGo, een AI ontwikkeld door Google, won in 2017 van Ke Jie in het spel Go. Go wordt gezien als het meest geavanceerde bordspel wat de mens heeft bedacht. Het verslaan van de wereldkampioen was daarom ook een van de grootste doorbraken in de AI van 2017.

Complicaties

Het implementeren van AI in Games levert verscheidene problemen op, een aantal erg belangrijke en lastig compenseerbare verhinderingen in dit onderzoeksgebied zijn:

Computationele limitaties

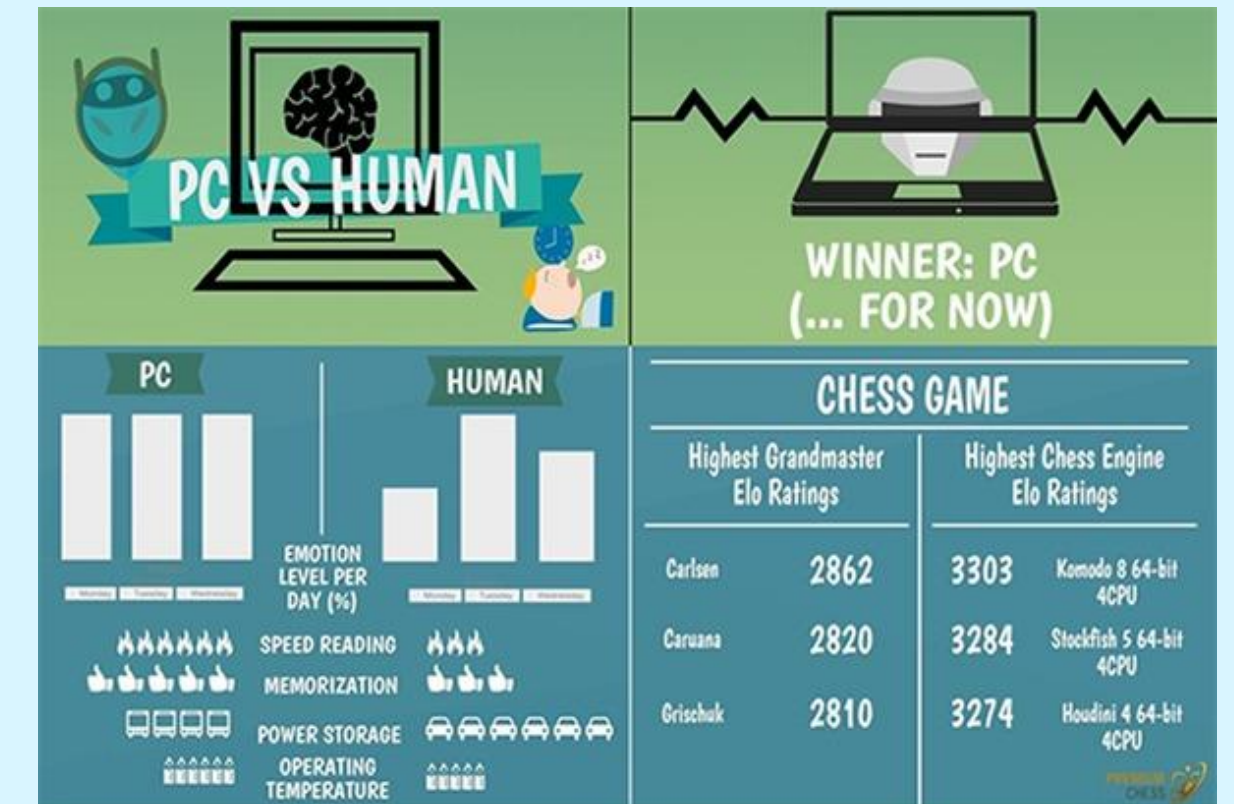
In vele situaties zijn de mogelijkheden op het gebied van Intelligent Gaming gelimiteerd door de hedendaagse computationele limieten. Hedendaagse games maken vaak gebruik van *complex decision spaces*, situaties waarbij uit zeer veel opties kan worden gekozen of waarbij zeer veel factoren van invloed zijn. Hierbij is het dus ook vaak niet gunstig om computationeel zware AI in deze games te implementeren. Vaak wordt er dus, als er überhaupt AI wordt geïmplementeerd, slechts gebruik gemaakt van lichte implementaties.

Knowledge Engineering

Het implementeren van AI in games is op zichzelf al een erg zware taak voor game developers; Alle beschikbare kennis over het domein moet beschikbaar zijn voor de AI, dit moet dus in zijn geheel worden geprogrammeerd. Hiernaast wordt deze kennis geprogrammeerd door mensen, *bugs* en errors zullen zich dus voordoen.

Onvoorspelbare situaties

AI is zeer sterk in het leveren van variatie aan een game, daar de *events* niet gescript zijn. Het is echter zo dat voor gebruik van AI, alle mogelijke kennis over een situatie beschikbaar moet zijn in de software. Op deze wijze kan de AI beslissingen maken en acties ondernemen. In het geval van een onvoorspelbare situatie is er echter geen beschikbare informatie (de situatie was immers onvoorspelbaar). Dit zal dus resulteren in ongewenst gedrag van de AI.



Is het mogelijk meerdere Artificial Intelligent agents met elkaar te laten samenwerken naar een doel?

Op het moment is het nog nergens in het vakgebied gelukt om meerdere agents te laten samenwerken. Als het mogelijk is om dit toch wel uit te kunnen voeren zullen de games beter worden en het hele vakgebied AI erop vooruitgaan.

Door veel verschillende agents te maken en daar ervaring mee op te doen, zal het mogelijk worden om deze beter te begrijpen en daarmee meer inzichten in het programmeren van de agents krijgen. Dus met behulp van het empirisch cyclus kunnen we meer te weten komen over agents en uiteindelijk meer testen tussen onderlinge verbanden en eventuele relaties tussen meerdere en zelfs verschillende agents.

Conclusie