

Rapid and Reliable Adaptation of Video Game AI

Moderne videospellen zijn door de jaren heen steeds realistischer geworden, met name wat hun visuele en auditieve presentatie betreft. Daarentegen heeft de game AI nog geen hoge graad van realisme bereikt. Deze is typisch gebaseerd op niet-adaptieve technieken. Een groot nadeel hiervan is dat zodra de speler een zwakheid in het spel heeft ontdekt, hij deze oneindig uit kan buiten. En dat komt het speelplezier niet ten goede.

De *game AI* kan worden verbeterd door het in staat te stellen zich intelligent aan te passen aan de omstandigheden in het spel, typisch door het leren van adequaat gedrag tijdens het spelen. Dit noemen we *adaptive game AI*. In de praktijk wordt dit echter maar zelden geïmplementeerd. Dit komt doordat het een aanzienlijk aantal leermomenten nodig heeft om effectief gedrag te leren tijdens het spelen van een spel (dat wil zeggen, de game AI verandert niet snel genoeg met de omstandigheden in het spel). Ook zijn ontwikkelaars van spellen bezorgd dat het opnemen van *adaptive game AI* leidt tot oncontroleerbaar en onvoorspelbaar gedrag (dat wil zeggen, de *game AI* is niet betrouwbaar). Het is dus belangrijk om te onderzoeken, hoe *adaptive game AI* kan worden vormgegeven die zich snel en betrouwbaar aan kan passen aan de omstandigheden in een videospel.

Een goed startpunt is de gangbare benadering: *incremental adaptive game AI*. Deze benadering corrigeert het gedrag van artificiële spelkarakters op basis van

observaties die het maakt tijdens het spelen van het spel. Bijvoorbeeld: als het ziet dat bepaald gedrag niet goed blijkt te werken, dan past hij het aan naar gedrag dat hopelijk beter werkt. De vraag is nu, hoe weet de AI welke aanpassing goed is voor welke omstandigheden? Dit is een moeilijk probleem, met name omdat de omstandigheden in videospellen vaak plots veranderen; de tegenstander kan opeens van strategie wisselen, met als gevolg dat al het gedrag dat de *adaptive AI* heeft geleerd niet meer geschikt is voor de nieuwe omstandigheden. Dit zagen we in het spel 'QUAKE III Capture-The-Flag'². In deze omgeving hebben we incremental adaptive game AI geïmplementeerd, met als doel de computer effectief gedrag te laten leren tijdens het spelen van het spel. We lieten het strijden tegen de game AI zoals die is opgenomen door de ontwikkelaars. Deze AI is relatief sterk en kan 'intelligent' wisselen tussen twee strategieën; als de AI het spel dreigt te verliezen schakelt hij naar een sterkere strategie, en als hij teveel voorsprong oploopt gaat hij zwakker spe-

len. Zo blijft het spannend voor de speler. Onze *game AI* wist dominante strategieën te ontdekken (zogenaamde "rush" strategieën) waartegen de originele tegenstander niets in te brengen had. Zodra het deze strategie ontdekt had, was het over met de originele QUAKE AI. Met zijn statische programmering kon het niets inbrengen tegen onze adaptieve variant.

Een nadeel van onze incrementele benadering als een *online* leermechanisme (een mechanisme dat leert *tijdens* spelen van het spel) is dat de leermethoden incidenteel zeer lang zijn. Bij een complex spel met telkens nieuwe menselijke tegenstanders is dit ongewenst. Aan de ene kant moet het leermechanisme de schaarse spelobservaties gebruiken om gedrag voor het heden op te stellen (exploitatie), en aan de andere kant moet het van te voren gedrag verkennen dat in de toekomst of op de lange termijn goed zal uitpakken (exploratie). Om dit probleem op te lossen kun je domeinkennis toevoegen aan de *game AI*. Bij QUAKE is dit bijvoorbeeld kennis over de wapens en in welke situaties

ze gebruikt moeten worden. Het is echter moeilijk om domeinkennis toe te voegen van hoge kwaliteit, aangezien een complex spel veel verschillende spelomstandigheden presenteert waarvoor telkens specifiek gedrag is vereist. Aan de ene kant is het voor game-ontwikkelaars ondoenlijk deze specialistische kennis in te brengen (wat de *game AI* bovendien zeer gevoelig zou maken voor gebreken in de kennis), en aan de andere kant heeft *incrementele game AI* te weinig leermomenten beschikbaar om deze kennis op te doen tijdens spelen van het spel.

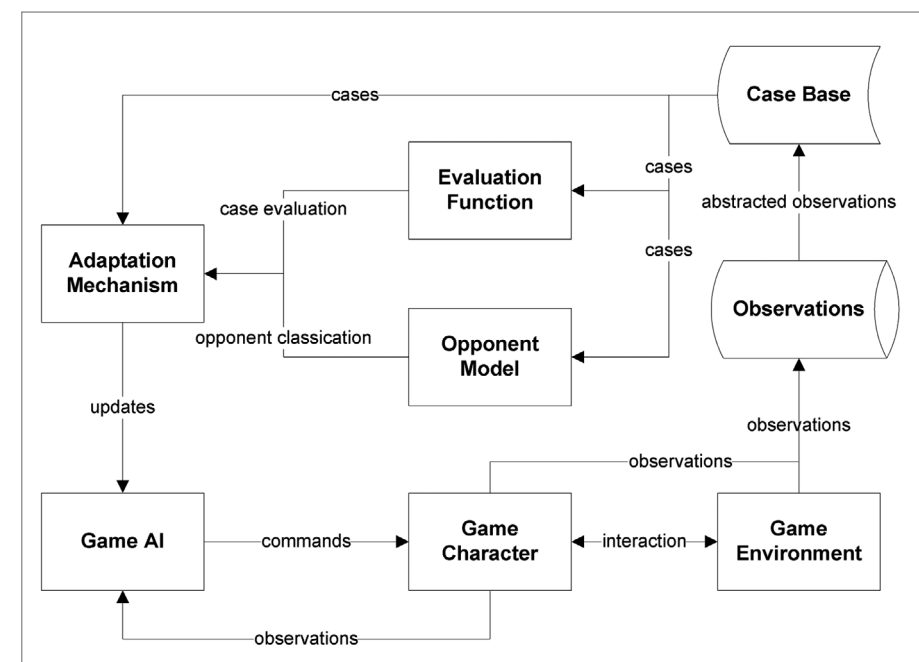
We stellen derhalve een alternatief voor de incrementele benadering voor. Deze benadering noemen we *case-based adaptive game AI*. Hierin zijn vooraf vastgestelde gedragspatronen opgeslagen in een "case base". Die worden gebruikt voor het *online* intelligent aanpassen van de *game AI*, al naar gelang de omstandigheden. De gedragspatronen bestaan uit observaties van eerder gespeelde spellen (tegen verschillende tegenstanders) die in abstracte vorm zijn opgeslagen in de case base. Tijdens het online spelen van het spel wordt domeinken-

nis automatisch vergaard door de AI, en direct (dat wil zeggen, zonder de benodigde leermomenten) geëxploiteerd om effectief gedrag op te stellen met behulp van de gedragspatronen in de case base.

De *case base* is dus feitelijk een vooraf vastgestelde voorraad gedragspatronen, te gebruiken in nieuwe situaties. Om het goed toe te passen zijn drie componenten nodig, namelijk (1) een *evaluatiefunctie* die nauwkeurig aangeeft hoe wenselijk het huidige gedrag van de AI is, (2) een *adaptatiemechanisme* wat de case base aan spelobservaties slim weet te gebruiken om het gedrag aan te passen, en (3) het opstellen en slim gebruiken van gedragsmodellen van de tegenstander (*opponent modelling*). Figuur 1 illustreert de samenhang van deze drie componenten.

Om de case-based adaptive game AI te testen, implementeren we het in het bestaande videospel SPRING. Dit is een open-source real-time strategy (RTS) spel met een strategische, gesimuleerde oorlogsomgeving³. Het spel is gebaseerd op de RTS klassieker Total Annihilation, wat door velen wordt beschouwd als een van de meest complexe RTS spellen ooit. In SPRING spelen typisch twee of meer spelers tegen elkaar. Iedere speler moet *in-game resources* verzamelen om hiermee eenheden en gebouwen te construeren. Het doel van het spel is de tegenstander te verslaan in een gesimuleerde, real-time veldslag, door de geconstrueerde eenheden en gebouwen goed te gebruiken. De speler die het eerst de zogenaamde *Commander* eenheid van de tegenstander weet te vernietigen, wint het spel.

Onze case based adaptive AI bestaat uit een evaluatiefunctie, adaptatiemechanis-



Figuur 1. Case-based adaptive game AI. De benadering slaat spelobservaties op in een case base. Deze observaties worden gebruikt om tegenstandermodellen en een evaluatiefunctie te leren, en de game AI direct aan te passen tijdens spelen van het spel. Onze game AI (linksonder) geeft commando's aan het spelkarakter. Deze interacteert met de omgeving en produceert observaties die in de case base worden gestopt. Het adaptatiemechanisme leert van de cases, en vernieuwt de game AI. De tegenstandermodellen en evaluatiefunctie worden gebruikt om het adaptatiemechanisme nauwkeurig te sturen, bijvoorbeeld wanneer de modellen erachter zijn gekomen welke speelstijl een tegenstander hanteert.

¹ Het onderzoek is uitgevoerd aan het Tilburg centre for Creative Computing (TiCC), van de Universiteit van Tilburg. Zie <http://sander.landofsand.com/index.php?publications> voor meer informatie over het onderzoek.

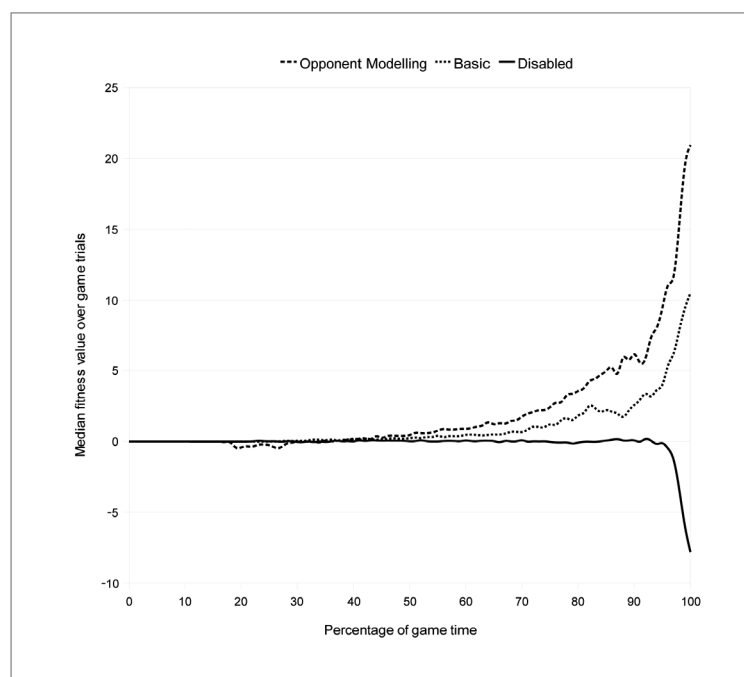
² Zie ook "Quake III Arena en Lerende Kunstmatige Intelligentie". De Connectie, Vol. 1, No. 5, pp. 22-23.

me en opponent modelling. *Machine learning* technieken vormen de **evaluatie-functie** op basis van spelobservaties uit de case base. In het bijzonder gebruiken we *Temporal Difference learning* om te leren hoe belangrijk verschillende eenheden zijn tijdens de gevechten. Van de in totaal 200 eenheden zijn sommige namelijk heel handig, en andere relatief verspilling van resources. Deze factoren zijn opgenomen in een functie die meet wiens leger sterker is, en hoe veilig de belangrijke Commander is op dat moment in het spel. Hiermee voorspelt de evaluatiefunctie de einduitslag. Experimenten met de evaluatiefunctie laten zien dat de functie vlak voor het einde van het spel de uitslag correct kan voorspellen met een nauwkeurigheid die de honderd procent benadert. Daarnaast doet de evaluatiefunctie tamelijk nauwkeurige voorspellingen voordat de helft van het spel is gespeeld. Aangezien SPRING plotseling gewonnen kan worden, en de uitkomst van het spel derhalve lastig is te voorspellen, is dit een bevredigend resultaat.

Het **adaptatiemechanisme** is erop gericht de *game AI* in staat te stellen zich snel en betrouwbaar aan te passen aan de omstandigheden in het spel. Het mechanisme bestudeert de case base en verandert op basis hiervan de *game AI*. Dit is nuttig voor het initialiseren van *game AI* met een naar verwachting effectieve spelstrategie, en voor online adaptatie van de *game AI* naar de omstandigheden in het spel. Bovendien is het bruikbaar voor offline generalisatie over spelobservaties. Oftewel, verzamelde observaties van eerder gespeelde spellen kunnen worden gebruikt om algemene uitspraken te

SmallDivide			
Adaptation mode	Trials	Goal achv.	Goal achv. (%)
Disabled	150	59	39%
Basic	150	115	77%
OM	150	135	90%
TheRing			
Adaptation mode	Trials	Goal achv.	Goal achv. (%)
Disabled	150	90	60%
Basic	150	122	81%
OM	150	127	85%
MetalHeckv2			
Adaptation mode	Trials	Goal achv.	Goal achv. (%)
Disabled	150	70	47%
Basic	150	124	83%
OM	150	130	87%

Figuur 3. Resultaten van experimenten op drie verschillende mappen van het spel. We observeren dat gebruik van case-based adaptive game AI (Basic) de speelsterkte van de game AI substantieel verbetert. Het gebruik van tegenstandermodellen (OM) doet de speelsterkte nog verder verbeteren.



Figuur 4. De Fitness van onze game-AI gemeten over de speeltijd. Een fitness waarde van 0 geeft een gelijkspel aan. Een positieve fitness waarde geeft aan dat de adaptieve game AI sterker gedrag vertoont dan de tegenstander. Te zien is dat de speelsterkte van onze game AI substantieel groeit met de speeltijd die het krijgt.

doen over de effectiviteit van strategieën. Geschikte strategieën kunnen dan later worden gebruikt in het spel ter initialisatie van de game AI of voor online adaptatie.

Het case-based adaptatiemechanisme is getest op drie verschillende spelomgevingen in SPRING. Het bleek goed stand te houden, zoals te zien in Figuur 3. Daarnaast was het telkens in staat een gelijkspel in stand te houden voor een onafgebroken tijdperiode. Dit is interessant, daar het gezien mag worden als een (simpele) manier om gebalanceerde gameplay op te stellen.

Bij **opponent modelling** worden modellen van de speelstijl en strategie van de tegenstander gemaakt op basis van casussen uit de case base. Deze casussen zijn observaties uit eerder gespeelde spellen, dus zelfs als een tegenstander nog nooit is geobserveerd is de kans groot dat het gedrag van de tegenstander te classificeren valt onder een generiek model. Mocht dat niet zo zijn, dan zal het gedrag van de tegenstander na het spelen van het spel automatisch worden opgenomen in een nieuw, automatisch gegenereerd model.

Waar zijn deze modellen nu goed voor? De modellen worden vooraf gebruikt om de adaptieve game AI nog nauwkeuriger te initialiseren met een effectieve strategie, en om tijdens het spelen van het spel nog nauwkeuriger de huidige speltoestand te vergelijken met wat reeds eerder is geobserveerd (zoals opgenomen in de case base). Hiermee richt *opponent modelling* zich erop om de speelsterkte, alsmede de adaptatiemogelijkheden, verder te verbeteren.

Dit zien we terug in onze experimenten met een adaptieve game AI waar-



Figuur 2. Screenshot van SPRING, een modern, complex real-time strategy (RTS) spel, waarin de effectiviteit van case-based adaptive game AI is aangetoond.

in de drie componenten van case-based adaptive game AI zijn geïntegreerd. Er was een toename te zien in de effectiviteit van de case-based adaptive game AI (figuur 3). Het was in staat zich snel en betrouwbaar aan te passen aan de spelomstandigheden, en vormt daarmee een sterke kandidaat voor inlijving in professionele game AI van de toekomst.

Deels wordt dit ook al gedaan. Het spel SPORE bijvoorbeeld maakt intensief gebruik van spelobservaties van vele spelers om de richting van het spel te sturen. Het verloop van het spel wordt aangepast aan de acties van de speler. Het

biedt een interessante richting voor AI-onderzoek; niet het gedrag van spelkarakters aanpassen, maar de spelomgeving zelf, bijvoorbeeld de levels, de spelregels, en de visuele stijl van weergave, aan de hand van modellen en actuele observaties van hoe de speler het spel speelt. ■

³ Zie ook <http://springrts.com/>