

PRZEGLĄD ALGORYTMÓW SZTUCZNEJ INTELIGENCJI STOSOWANYCH W GRACH KOMPUTEROWYCH

1. Rola sztucznej inteligencji w grach komputerowych

Sztuczna inteligencja (dalej zastępczo określana skrótem AI, od angielskiej nazwy *Artificial Intelligence*) pełni w grach komputerowych różnorakie zadania, które można podzielić na trzy kategorie¹:

- urealnienie świata gry: stosowane przede wszystkim w grach typu cRPG (ang. *Computer Role-Playing Games*); AI ma za zadanie sterować poczynaniami agentów (postaci sterowane przez grę), z którymi zetknie się bohater gracza;
- wsparcie rozgrywania walki: najczęściej spotykana kategoria sztucznej inteligencji w grach komputerowych – głównie w grach strategicznych i tzw. shooterach; AI ma za zadanie sterować agentami podczas walki;
- komentowanie wydarzeń – najczęściej w grach sportowych; AI odpowiedzialne jest za komentowanie wydarzeń mających miejsce w świecie gry na podstawie bieżących działań gracza.

Działanie sztucznej inteligencji nie kończy się jednak tylko na symulowaniu inteligentnych zachowań, ale ma również nadać agentom cechy ludzkie. Dopiero zaimplementowanie do gry takiego założenia może sprawić, by zabawa przyciągnęła do siebie rzesze graczy. Uzyskuje się to poprzez próbę przeniesienia do gry ludzkich niedoskonałości, a także stworzenia bliskiego człowiekowi sposobu po-

¹ Por. T. Schreiner, *Artificial Intelligence in Game Design*, <<http://ai-depot.com/GameAI/Design.html>>, dostęp do wszystkich źródeł internetowych: 23 września 2007.

rozumiewania się (duży zasób słów, nieliniowe dialogi, indywidualizacja języka komputerowych postaci)².

2. Historia sztucznej inteligencji w grach komputerowych

Pierwszy algorytm AI dla gier komputerowych datuje się na rok 1951. Wtedy to matematyk Alan Turing stworzył pierwszą grę wykorzystującą sztuczną inteligencję. Gra Turinga była wzorowana na popularnej wówczas w Anglii zabawie towarzyskiej, która polegała na odgadnięciu płci dwóch osób ukrytych przed zgadującym. Starał się on odgadnąć płeć na podstawie uzyskanych odpowiedzi na zadane przez siebie pytania. Turing w swojej grze zamienił jedną z ukrywających się osób na swój algorytm, a odgadujący miał za zadanie wskazać, który z dwójki przepytanych jest żywym człowiekiem. Ponieważ Turing nie dysponował komputerem, symulował działanie algorytmu na papierze. Jak się okazało, algorytm ten przegrywał nawet z bardzo słabymi graczami, co sprawiło, że krytycy określili grę Turinga jako bezcelową³. Jednakże już rok po porażce Turinga profesor Arthur Samuel zatrudniony przez IBM stworzył algorytm bardzo dobrze grający w warcaby, który po wielu poprawkach pokonał w 1954 roku czwartego gracza rankingu USA. Kolejnym przełomem był napisany w 1967 roku przez Richarda Greenblatta program szachowy MacHack. Aplikacja ta zawierała bardzo rozbudowaną bazę (zbiór możliwych posunięć zależnych od sytuacji na planszy) i w starciach z żywymi przeciwnikami potrafiła zajmować wysokie pozycje. Po tych wydarzeniach nastąpił szybki rozwój AI w grach komputerowych. Jednak z powodu małych mocy obliczeniowych komputerów w latach osiemdziesiątych tworzenie sprawnie działającej sztucznej inteligencji dla innych typów gier niż planszowe było bardzo utrudnione. Podejmowane próby kończyły się bez większych sukcesów. Najlepszą w tym okresie grą nieplanszową wyposażoną w AI stała się gra *Hobbit* z 1983 roku (producent Beam Software). Występujący w niej agenci mieli za zadanie wspierać swoimi radami gracza, ale również często jak rady przydatne, zdarzały się wskazówki pozbawione sensu⁴.

Z końcem lat 80. na rynku pojawiły się gry wykorzystujące proste skrypty AI odpowiedzialne za zachowania agentów. Jednakże działanie takich przeciwników było wciąż łatwe do przewidzenia dla gracza. Dopiero gdy w 1994 roku wydana została gra *Warcraft* (producent Blizzard Entertainment) typu RTS (ang. *Real Time Strategy* – strategia czasu rzeczywistego), gracze otrzymali aplikację z sukcesem wykorzystującą AI. Po raz pierwszy na tak szeroką skalę zastosowano algorytm wyszukiwania drogi. W 1999 roku nastąpił kolejny przełom, gdy zaprezentowano grę *Unreal: Tournament* (producent Digital Extremes), w której

² J. Swacha, *Sztuczna inteligencja w grach komputerowych*, <<http://klubinformatyka.pl/arttykul.php?a=12>>.

³ P. Łupkowski, *Test Turinga*, <<http://www.kognitywistyka.net/encyklopedia/test-turinga.html>>.

⁴ M. Czapelski, *Gry na serio*, <<http://www.pcworld.pl/arttykuly/38243.html>>.

komputerowi agenci starali się naśladować zachowania graczy. Poruszali się oni po obszarach, na których toczyły się walki, tak samo jak gracz – i walczyli zgodnie z tymi samymi zasadami, które ograniczały człowieka. Ostatnim dużym przełomem, jaki można było zaobserwować w świecie gier komputerowych, było stworzenie w 2001 roku przez LionHead Studios gry *Black & White*. Po raz pierwszy wykorzystano w niej techniki uczenia się postaci komputerowych w trakcie trwania rozgrywki⁵.

3. Techniki symulacji sztucznej inteligencji

Jest wiele rodzajów algorytmów sztucznej inteligencji wykorzystywanych w grach komputerowych. W dalszej części tego rozdziału dokładnie omówione zostały tylko najpopularniejsze algorytmy, z jakimi można się spotkać wśród setek obecnych na rynku gier wykorzystujących sztuczną inteligencję. Ich popularność wynika z tego, że najprostsze techniki – takie jak automaty skończone, drzewa decyzyjne czy heurystyczne poszukiwanie drogi – okazują się najlepszymi z dostępnych metod projektowania sztucznej inteligencji na potrzeby gier.

3.1. Algorytm stadny

Algorytm stadny po raz pierwszy został przedstawiony w 1987 r. roku przez Craiga Reynoldsa. Reynolds zaproponował w swoim artykule trzy zasady, które po połączeniu umożliwiały grupie agentów realistyczne zbiorowe zachowanie przypominające zachowania ławicy ryb, roju pszczół albo stada ptaków. Te trzy zasady Reynolds nazwał sterowaniem zachowaniem, a są nimi:

- rozdzielczość – sterowanie zapobiegające tworzeniu tłumu w jednym miejscu, które polega na tym, że agenci muszą zachowywać pewną odległość od siebie,
- wyrównywanie – sterowanie dające możliwość agentowi zmiany kierunku i prędkości swego przemieszczania, dzięki czemu agent może dostosowywać te parametry do innych agentów przebywających w jego pobliżu,
- spójność – sterowanie odpowiedzialne za zbieranie się agentów przebywających blisko siebie w lokalne grupy.

Z czasem Reynolds dodał czwartą zasadę, którą określił mianem „unikania”. Jej zadaniem jest sterowanie zapobiegające zderzeniom agentów z przeszkodami⁶.

Między poszczególnymi cyklami (uaktualnieniami) – po których agenci za każdym razem sprawdzają środowisko, w jakim w danej chwili przebywają

⁵ J. Grzyb, *Sztuczna inteligencja w grach*, „Software Developer’s Journal”, nr 7, 2005, s. 26.

⁶ Por. M. DeLoura (red.), *Perleki programowania gier – Vademecum profesjonalisty*, t. 1, tłum. R. Jońca, Helion, Gliwice 2002.

– algorytm stadny nie przechowuje żadnych dodatkowych informacji. To powoduje zmniejszenie wymagań pamięciowych związanych ze sterowaniem wieloma agentami oraz pozwala na natychmiastową reakcję na zmianę sytuacji. Wynikiem działania tego algorytmu jest stado poruszające się z taką samą dynamiką ruchu jak jedno ciało, omijające wszelkie przeszkody i wrogie postacie.

3.2. Automaty stanów skończonych

Automaty stanów skończonych są często wykorzystywane w grach komputerowych. Już w grach z lat 90. technika ta była wykorzystywana do kontrolowania agentów, ale i w najnowszych produkcjach automaty stanów skończonych są używane do sterowania AI gry. Również w większości gier typu cRPG automaty skończone wykorzystuje się do sterowania dialogami gracza z agentami. Stanowią one także w wielu grach podstawę zarządzania światem, przechowują stan gry, przetwarzają polecenia od gracza lub zarządzają stanem obiektu⁷.

Automat stanów skończonych zbudowany jest z pewnej ściśle określonej liczby stanów znajdujących się w danej przestrzeni rozwiązań. Gdy pojawia się nowe zdarzenie, następuje przejście w inny stan. Prowadzi to do sytuacji pozwalającej podjąć jedno lub kilka różnych działań⁸.

3.3. Drzewa decyzyjne

Drzewa decyzyjne są popularną techniką sztucznej inteligencji stosowaną przy tworzeniu AI dla komputerowych odpowiedników gier planszowych takich jak szachy, warcaby, czy go. Zadaniem tego algorytmu AI może być rozwiązywanie problemów decyzyjnych, jak również tworzenie planu działania⁹.

Drzewo decyzyjne przedstawia się w postaci grafu decyzji i ich możliwych konsekwencji, którego węzły są stanem gry, a węzły potomne to położenia uzyskiwane po przeprowadzeniu jednego ruchu. Węzły decyzyjne i węzły konsekwencji występują naprzemiennie, a każda ze ścieżek kończy się węzłem końcowym. Agent analizuje drzewo decyzyjne tak daleko, jak potrafi lub uzna za konieczne, rozpatrując wszelkie możliwe posunięcia względem obecnej sytuacji – i wybiera to, które uzna za najlepsze¹⁰.

⁷ J. Grzyb, op. cit.

⁸ Por. N. Meyer, *Automaty skończone*, tłum. redakcja, „Software 2.0 Extra!”, nr 14, 2005.

⁹ Por. M. Lampiarski, *Metody sztucznej inteligencji w grach komputerowych wczoraj, dziś i jutro*, <<http://www.fizyka.umk.pl/~duch/Wyklady/AI/Prace06/Lamparski-siwgrach.pdf>>.

¹⁰ Por. M. DeLoura (red.), op. cit.

3.4. Heurystyczne poszukiwanie drogi – algorytm A*

Określenie najlepszej drogi poruszania z punktu A do B na mapie to problem pojawiający się w większości gier komputerowych. Na bazie techniki AI rozwiązującej to zagadnienie często tworzone są bardzo skomplikowane i złożone zachowania, takie jak planowanie strategiczne czy poruszanie się jednostek w formacjach. Sztandarowym rozwiązaniem dla tego problemu jest heurystyczny algorytm A*. Algorytm ten podczas określania trasy nie szuka drogi „na ślepo”, ale szacuje najlepszy kierunek eksploracji¹¹.

Działanie algorytmu A* polega na minimalizacji obszaru poszukiwań najlepszej trasy dzięki ukierunkowaniu poszukiwań na cel. Technika ta oblicza koszt dotarcia do punktu na mapie i dodaje do niego heurystykę określającą przewidywany koszt dotarcia do celu. Działanie to jest najczęściej odległością do celu mierzoną od obecnego punktu obszaru poszukiwań. Algorytm A* sprawdza po każdym ruchu agenta możliwe kierunki dalszej trasy i ponownie wybiera najlepszy kierunek ruchu. Gdy zbadane położenie okazuje się szukanym celem, algorytm kończy swoje działanie. W innym przypadku zapamiętuje przyległe położenia, by w przyszłości móc je jeszcze sprawdzić¹².

3.5. Logika rozmyta

Logika rozmyta jest powiązana z teorią zbiorów rozmytych i teorią prawdopodobieństwa. Pomiędzy stanem 0 (fałsz) a stanem 1 (prawda) rozciąga się wiele wartości pośrednich określających stopień przynależności elementu do zbioru. Technika ta pozwala na prace z koncepcjami, które wymagają podania odpowiedzi na pytania typu „w jakim stopniu?”, „ile?”, np.: „bardzo”, „mocno”, „dużo”, „mały”, „średni”, „trochę”. W grach komputerowych logika rozmyta wykorzystywana jest do symulowania emocji agentów (np. „miły”, „obojętny”, „nieuprzejmy”, „wrogi”, „zły”, „wściekły” itp.), dzięki czemu tworzona jest dużo głębsza i bardziej podobna do ludzkiej sfera uczuciowa komputerowych postaci. Innymi przykładami zastosowania logiki rozmytej jest sterowanie nieograniczonymi zachowaniami – przykładem mogą być poruszające się na niebie chmury czy drżenie liści na wietrze¹³.

3.6. Sztuczne sieci neuronowe

Sztuczne sieci neuronowe są wzorowane na sieciach neuronowych mózgu człowieka. Realizują obliczenia albo przetwarzają sygnały za pomocą elementów

¹¹ Por. J. Grzyb, op. cit.

¹² Por. M. DeLoura (red.), op. cit.

¹³ Ibidem.

wykonujących podstawową operację na wejściu, zwanych neuronami. Agent sterowany za pomocą sztucznej sieci neuronowej może się uczyć podczas gry wraz ze zdobywanym doświadczeniem: gdy natrafia na nowy rodzaj sytuacji, dostosowuje się do niej, korzystając z tego, co „pamięta”¹⁴.

O możliwościach zaadaptowania sztucznych sieci neuronowych do tworzenia AI w grach komputerowych mówiło się od wielu lat, jednak dopiero w grudniu 2000 roku pojawiła się gra *Collin MaRae Rally 2.0* (producent Codemasters Software), będąca symulatorem wyścigów samochodowych wykorzystującym tę technikę. Na bazie danych wejściowych, którymi były takie parametry jak krzywizna łuku drogi, odległość od zakrętu, rodzaj gruntu, prędkość i parametry pojazdu, sieć neuronowa miała za zadanie wygenerować dane wyjściowe. Dane te musiały być tak dobrane, by samochód kierowany przez program bez kłopotów pokonywał wszystkie przeszkody i zakręty¹⁵.

4. Dalszy rozwój AI w grach komputerowych

Rola sztucznej inteligencji w grach komputerowych zwiększyła się o czwartą kategorię: analizowanie przebiegu gry i jej dostosowanie do gracza. Polega ona na tym, iż wraz ze wzrostem doświadczenia gracza, nabytego podczas grania, cały świat gry również się rozwija.

Liquid Narrative Group działająca na Uniwersytecie Północnej Karoliny pracuje nad stworzeniem gry o nieliniowej fabule na bazie gry *Unreal Tournament 2003*. Założeniem jest, aby świat gry, z jakim zetknie się gracz, zmieniał się wraz z jego działaniami. Zdaniem szefa tego projektu Michaela Younga „Użytkownicy będą wchodzić do gry tak, jak do realnej rzeczywistości. Faktycznie będą grali w świecie kontrolowanym przez tworzone przez nas mechanizmy sztucznej inteligencji, które sterują jak marionetką wszystkim, co się dzieje”¹⁶. Young wraz ze swoją grupą stworzył system opracowywania interaktywnych gier o nazwie Mimesis. Składa się on z tzw. kontrolera oraz serwera Mimesis, będącego zmodyfikowanym jądrem gry *Unreal: Tournament*. Kontroler odpowiedzialny jest za planowanie zadań, które gracz ma wykonać, aby ukończyć grę. Odpowiada również za komunikację z serwerem Mimesis poprzez XML (ang. *Extensible Markup Language*), a zadaniem serwera jest tworzenie odpowiedniego świata oraz historii i udostępnienia jej graczowi na jego komputerze¹⁷.

Autorowi niniejszego artykułu wydaje się jednak, że ewolucja AI w grach komputerowych nie zakończy się tylko na przedstawionym powyżej czwartym typie

¹⁴ Por. M. Lampiarski, op. cit.

¹⁵ Por. J. Grzyb, op. cit.

¹⁶ Por. M. Czapelski, op. cit.

¹⁷ Ibidem.

sztucznej inteligencji, ale będzie zmierzać dalej, rozszerzając jej funkcje. Sztuczna inteligencja nie tylko będzie miała za zadanie kreowanie i kontrolowanie świata wokół postaci gracza, ale również sterowanie fizjologią i psychologią samej postaci. Dzięki temu wymusi na graczu dużo większe zaangażowanie w grę. O wniosek ten można się pokusić, gdyż wyraźnie widać, że twórcy gier komputerowych rozwijają sztuczną inteligencję tak, by każdy aspekt wirtualnego świata wydawał się jak najbardziej realny.

Literatura

- de Back W., Veldkamp R., 2005, *Walcz o swe sztuczne życie*, tłum. redakcja, „Software 2.0 Extra!”, nr 14.
- Czapelski M., *Gry na serio*, <<http://www.pcworld.pl/artykuly/38243.html>>.
- DeLoura M., 2002, *Perłki programowania gier – Vademecum profesjonalisty*, 2002, red. M. DeLoura, t. 1 i 2, tłum. R. Jońca, Helion, Gliwice.
- Dickheiser M. i in., 2007, *Perłki programowania gier – Vademecum profesjonalisty*, t. 6, tłum. A. Grażyński, M. Szczepaniak, Helion, Gliwice.
- Grzyb J., 2005, *Sztuczna inteligencja w grach*, „Software Developer’s Journal”, nr 7.
- Kasperski M.J., 2003, *Sztuczna Inteligencja*, Helion, Gliwice.
- Lampiarowski M., *Metody sztucznej inteligencji w grach komputerowych wczoraj, dziś i jutro*, <<http://www.fizyka.umk.pl/~duch/Wyklady/AI/Prace06/Lamparski-siwgrach.pdf>>.
- Łupkowski P., 2007, *Test Turinga*, <<http://www.kognitywistyka.net/encyklopedia/test-turinga.html>>.
- Meyer N., 2005, *Automaty skończone*, tłum. Redakcja, „Software 2.0 Extra!”, nr 14.
- Nareyek A., 2004, *AI In Computer Games*, <<http://www.acmqueue.com/modules.php?name=Content&pa=showpage&pid=117>>, Schreiner T., *Artificial Intelligence in Game Design*, <<http://ai-depot.com/GameAI/Design.html>>.
- Swacha J., *Sztuczna inteligencja w grach komputerowych*, <<http://klubinformatyka.pl/artykul.php?a=12>>.
- Teixeira de Sousa B. M., 2003, *Programowanie gier – Kompendium*, tłum. A. Bochenek, J. Dobrzański, S. Dzieniszewski, Helion, Gliwice.
- Treglia D. (red.), 2003, *Perłki programowania gier – Vademecum profesjonalisty*, 2003, red. D. Treglia, t. 3, tłum. R. Jońca, Helion, Gliwice.