

Gatunki gier opartych na sztucznej inteligencji

Większość gier jest dość skromna pod względem stosowanych technik sztucznej inteligencji i zachowań, które chcą osiągnąć. Wyróżniająca się i rzucająca się w oczy sztuczna inteligencja nie jest zwykle dobrą sztuczną inteligencją. Od czasu do czasu pojawiają się jednak gry, które zawierają określone techniki sztucznej inteligencji jako mechanikę gry. Wyzwanie polega na manipulowaniu umysłem postaci w grze. Jako programista AI byłoby wspaniale widzieć więcej tego rodzaju tytułów, ale jak dotąd było stosunkowo niewiele przykładów opartych na ograniczonej liczbie stylów gry. Przyjrzymy się dwóm opcjom rozgrywki skoncentrowanej na sztucznej inteligencji. Opisane gatunki reprezentuje tylko kilka tytułów, które odniosły sukces komercyjny. Nie jest jasne, czy powstanie więcej gier wykorzystujących dokładnie to samo podejście, chociaż każda z nich może zostać wydobyta w celu uzyskania ciekawego zachowania i rozgrywki, które można zastosować w bardziej popularnych gatunkach. Dla każdego typu gry opiszę zestaw technologii, które wspierałyby odpowiednią rozgrywkę. Chociaż niektóre szczegóły konkretnych gier z każdego gatunku są dostępne w domenie publicznej, wiele szczegółów algorytmicznych jest poufnych. Nawet jeśli informacje są dostępne, ograniczona liczba tytułów oznacza, że trudno jest ogólnie określić, co działa, a co nie. Jest zatem nieuniknione, że ta dyskusja będzie miała charakter spekulacyjny. Tutaj postaram się wskazać alternatywy.

CHARAKTERY NAUCZANIA

Nauczanie nieudolnej postaci działania zgodnie z twoją wolą pojawiło się w wielu grach. Oryginalna gra tego rodzaju, *Creatures*, została wydana w 1996 roku. Obecnie gatunek ten jest najbardziej znany z *Black & White* z 2001 roku. Niewielka liczba postaci (tylko jedna w *Black & White*) ma mechanizm uczenia się, który uczy się wykonywać czynności, które ma widziane, pod nadzorem gracza. Obserwacyjny mechanizm uczenia obserwuje poczynania innych postaci oraz gracza i stara się je naśladować. Kiedy odtwarza czynność, gracz może dać pozytywną lub negatywną informację zwrotną (zwykle uderzenie i łaskotanie), aby zachęcić lub zniechęcić do ponownego wykonania tego samego działania.

REPREZENTOWANIE DZIAŁAŃ

Podstawowym wymogiem uczenia się obserwacyjnego jest umiejętność reprezentowania działań w grze za pomocą dyskretnej kombinacji danych. Postać może wtedy nauczyć się naśladować te działania, prawdopodobnie z niewielkimi zmianami. Zazwyczaj akcje są reprezentowane przez trzy elementy danych: samą akcję, opcjonalny obiekt akcji i opcjonalny obiekt pośredni. Na przykład akcją może być „walka”, „rzut” lub „sen”; tematem może być „wróg” lub „kamień”; a pośrednim przedmiotem może być „miecz”. Nie każda akcja wymaga obiektu (na przykład sen), a nie każda akcja, która ma podmiot, ma również przedmiot pośredni (na przykład rzut). Niektóre działania mogą przybierać różne formy. Możliwe jest np. rzucenie kamieniem lub rzucenie kamieniem w konkretną osobę. Dlatego akcja rzut zawsze przyjmuje obiekt, ale opcjonalnie może również przyjmować obiekt pośredni. W trakcie realizacji dostępna jest baza działań. Dla każdego rodzaju akcji gra rejestruje, czy wymaga obiektu, czy obiektu pośredniego. Kiedy postać coś robi, można stworzyć strukturę akcji, aby ją reprezentować. Struktura akcji składa się z rodzaju akcji i szczegółów rzeczy w grze, które mają działać jako obiekt i pośredni obiekt, jeśli jest to wymagane.

REPREZENTUJĄCY ŚWIAT

Oprócz akcji, postaci muszą być w stanie zbudować obraz świata. To pozwala im powiązać działania z kontekstem. Na przykład nauka jedzenia jest dobra, ale nie wtedy, gdy jesteś atakowany przez wroga. To właściwy czas na ucieczkę lub walkę. Prezentowane informacje kontekstowe są zazwyczaj dość wąskie. Duże ilości informacji kontekstowych mogą poprawić wydajność, ale radykalnie zmniejszyć

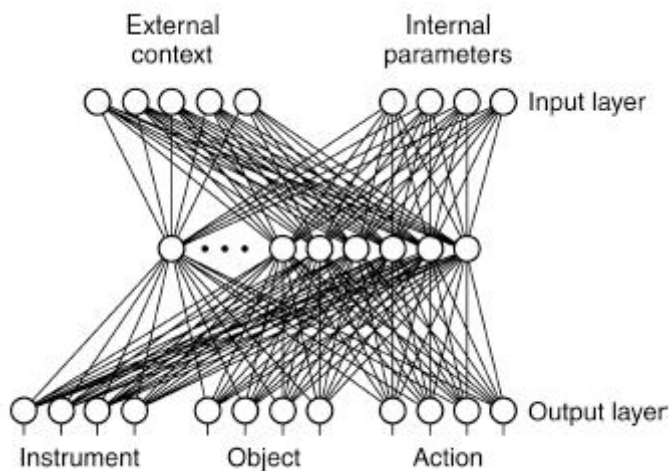
szybkość uczenia się. Ponieważ gracz jest odpowiedzialny za nauczanie postaci, gracz chce zobaczyć jakąś oczywistą poprawę w stosunkowo krótkim czasie. Oznacza to, że uczenie się musi odbywać się tak szybko, jak to możliwe, nie prowadząc do głupiego zachowania. Zazwyczaj wewnętrzny stan postaci jest zawarty w kontekście wraz z kilkoma ważnymi danymi zewnętrznymi. Może to obejmować odległość do najbliższego wroga, odległość do bezpieczeństwa (domu lub innych postaci), porę dnia, liczbę osób obserwujących lub inne wartości zależne od gry. Ogólnie rzecz biorąc, jeśli postać nie otrzyma jakiejś informacji, to skutecznie ją zignoruje przy podejmowaniu decyzji. Oznacza to, że jeśli decyzja byłaby niewłaściwa w pewnych warunkach, warunki te muszą być przedstawione postaci. Informacje kontekstowe mogą być prezentowane postaci w postaci serii wartości parametrów (bardzo powszechna technika) lub w postaci zestawu dyskretnych faktów (podobnie jak reprezentacja akcji).

MECHANIZM NAUKI

Dla postaci możliwe są różne mechanizmy uczenia się. Opublikowane gry z tego gatunku wykorzystywały sieci neuronowe i uczenie drzew decyzyjnych, Naive Bayes i uczenie się ze wzmocnieniem mogą być również interesującymi podejściami do wypróbowania. Jako obszerny przykład pracy przyjrzyjmy się szczegółowo wykorzystaniu sieci neuronowej w tej sekcji. W przypadku algorytmu uczenia sieci neuronowych istnieje połączenie dwóch rodzajów nadzoru: silnego nadzoru z obserwacji i słabego nadzoru z informacji zwrotnej od gracza.

Architektura sieci neuronowych

Chociaż w tego typu grach można używać wielu różnych architektur sieciowych, zakładam, że używana jest wielowarstwowa sieć perceptronowa, jak pokazano na rysunku.



Zostało to zaimplementowane wcześniej i można je zastosować z minimalną modyfikacją. Warstwa wejściowa dla sieci neuronowej pobiera informacje kontekstowe ze świata gry (w tym wewnętrzne parametry postaci). Warstwa wyjściowa dla sieci neuronowej składa się z węzłów kontrolujących rodzaj akcji oraz obiekt i obiekt pośredni akcji (plus wszelkie inne informacje wymagane do utworzenia akcji).

Niezależnie od uczenia się, sieć może być używana do podejmowania decyzji dotyczących postaci, podając bieżący kontekst jako dane wejściowe i odczytując akcję z danych wyjściowych. W sposób nieunikniony większość działań wyjściowych będzie nielegalna (może nie być w tym czasie możliwe lub nie ma takiego obiektu lub obiektu pośredniego), ale te, które są legalne, są przeprowadzane. Można próbować zniechęcić do nielegalnych działań, przechodząc przez słabo nadzorowany etap uczenia się za każdym razem, gdy zostanie to zasugerowane. W praktyce może to poprawić wydajność w krótkim okresie, ale może prowadzić do problemów ze stanami patologicznymi w dłuższej perspektywie. Nauka

obserwacyjna Aby uczyć się przez obserwację, postać rejestruje działania innych postaci lub gracza. Dopóki te działania są w jego wizji, wykorzystuje je do uczenia się. Po pierwsze, postać musi znaleźć reprezentację dla akcji, którą widziała, oraz reprezentację dla aktualnego kontekstu. Następnie może trenować sieć neuronową za pomocą tego wzorca wejścia-wyjścia, jednorazowo lub wielokrotnie, aż sieć nauczy się prawidłowego wyjścia dla danych wejściowych. Wykonanie tylko jednego przejścia przez algorytm uczenia prawdopodobnie spowoduje bardzo niewielką różnicę w zachowaniu postaci. Z drugiej strony wykonywanie wielu iteracji może spowodować, że sieć zapomni o przydatnych zachowaniach, których już się nauczyła. Ważne jest, aby znaleźć rozsądną równowagę między szybkością uczenia się a szybkością zapominania. Gracze będą tak samo sfrustrowani koniecznością ponownego uczenia swojego stwora, jak oni, jeśli nauka jest bardzo powolna.

Czytanie w myślach

Jedną z istotnych kwestii związanych z uczeniem się przez obserwację jest określenie informacji kontekstowych, które pasują do obserwowanego działania. Jeśli postać, która nie jest głodna, zauważy jedzenie głodnej postaci, może nauczyć się kojarzyć jedzenie z brakiem głodu. Innymi słowy, twoje własne informacje kontekstowe nie mogą być dopasowane do czyichś działań. W grach, w których gracz sam uczy, ten problem nie pojawia się. Zazwyczaj gracz stara się pokazać postaci, co ma dalej robić. Można wykorzystać informacje kontekstowe postaci. W przypadkach, gdy postać obserwuje inne postacie, jej własna informacja kontekstowa jest nieistotna. W realnym świecie niemożliwe jest zrozumienie wszystkich motywów i wewnętrznych procesów kogoś innego, gdy widzimy jego działanie. Próbowalibyśmy zgadywać lub czytać w myślach, co muszą myśleć, aby wykonać to działanie. W sytuacji w grze jesteśmy w stanie wykorzystać niezmienione informacje kontekstowe obserwowanych postaci. Chociaż możliwe jest dodanie pewnej niepewności, aby przedstawić trudności w poznaniu myśli innych, w praktyce nie sprawia to, że postać wygląda bardziej wiarygodnie i może dramatycznie spowolnić tempo uczenia się.

Nauka informacji zwrotnych

Aby uczyć się przez informacje zwrotne, postać zapisuje listę wyjść, które stworzyła dla każdego z ostatnich wejść. Ta lista musi rozciągać się co najmniej o kilka sekund. Gdy gracz otrzyma informację zwrotną (na przykład uderzenie lub łaskotanie), nie ma sposobu, aby dokładnie wiedzieć, z której akcji gracz był zadowolony lub zły. Jest to klasyczny „problem przypisywania kredytów” w AI: w serii działań, jak możemy stwierdzić, które działania pomogły, a które nie? Prowadząc listę kilkusekundowych par wejścia-wyjścia, zakładamy, że informacja zwrotna od użytkownika jest związana z całą serią działań. Po nadejściu informacji zwrotnej sieć neuronowa jest szkolona (przy użyciu metody słabo nadzorowanej) w celu wzmocnienia lub osłabienia wszystkich par wejścia-wyjścia w tym czasie. Często przydatne jest stopniowe zmniejszanie ilości sprzężenia zwrotnego w miarę cofania się par wejścia-wyjścia w czasie. Jeśli postać otrzyma informację zwrotną, najprawdopodobniej chodzi o akcję wykonaną sekundę lub dwie temu (jeszcze mniej czasu, a użytkownik nadal przeciągałby kursor w odpowiednie miejsce, aby go uderzyć lub połaskotać).

PRZEWIDYWALNE MODELE UMYSŁOWE

W przypadku tego rodzaju gier AI ma wspólny problem: trudno jest zrozumieć, jaki wpływ na postać będą miały działania gracza. W pewnym momencie gry wydaje się, że postać bardzo łatwo się uczy, podczas gdy w innych momentach wydaje się całkowicie ignorować gracza. Sieć neuronowa, na której działa postać, jest zbyt skomplikowana, by mogła być właściwie zrozumiana przez jakiegokolwiek gracza i często wydaje się, że robi coś złego. Oczekiwania graczy są istotną częścią tworzenia dobrej sztucznej inteligencji. Jak omówiono, postać może robić coś bardzo inteligentnego, ale jeśli nie jest to to, czego oczekiwał gracz, często będzie wyglądać głupio. W powyższych algorytmach informacja

zwrotna od gracza jest rozdzielana na szereg działań typu input - output. Jest to powszechne źródło nieoczekiwanej wiedzy. Kiedy gracze przekazują informację zwrotną, nie są w stanie powiedzieć, którą konkretną akcję lub część akcji oceniają. Jeśli na przykład postać podnosi kamień i próbuje go zjeść, gracz uderza go, aby nauczyć go, że kamienie nie nadają się do jedzenia. Kilka chwil później bohater próbuje zjeść trujący muchomor. Gracz ponownie go uderza. Wydaje się logiczne dla gracza, że uczył postać, co jest dobre, a co złe do jedzenia. Bohater jednak rozumie tylko, że „jedzenie kamieni” jest złe, a „jedzenie muchomorów” jest złe. Ponieważ sieci neuronowe w dużej mierze uczą się przez uogólnianie, gracz po prostu nauczył postać, że jedzenie jest złe. Stworzenie powoli umiera z głodu, nigdy nie próbując jeść niczego zdrowego. Nigdy nie dostaje szansy na łaskotanie przez gracza za zjedzenie właściwej rzeczy. Te mieszane komunikaty są często źródłem nagłego i dramatycznego pogorszenia zachowań postaci. Podczas gdy gracz spodziewałby się, że postać będzie coraz lepiej zachowywać się we właściwy sposób, często szybko osiąga poziom plateau i czasami może się pogarszać. Nie ma ogólnej procedury rozwiązywania tych problemów. W pewnym stopniu wydaje się to być słabością tego podejścia. Można to jednak do pewnego stopnia złagodzić, używając „instynktów” (tj. ustalonych domyślnych zachowań, które działają dość dobrze) wraz z uczącą się częścią mózgu. Instynkt to wbudowane zachowanie, które może być przydatne w świecie gry. Postać może na przykład otrzymać instynkt do jedzenia lub spania. Są to skutecznie przepisane pary wejścia-wyjścia, których nigdy nie można całkowicie zapomnieć. Mogą być wzmacniane w regularnych odstępach czasu poprzez nadzorowany proces uczenia się lub mogą być niezależne od sieci neuronowej i wykorzystywane do generowania okazjonalnych zachowań. W obu przypadkach, jeśli instynkt zostanie wzmocniony przez gracza, stanie się on częścią wyuczonych zachowań postaci i będzie wykonywany znacznie częściej.

Mózg Śmierć Postaci

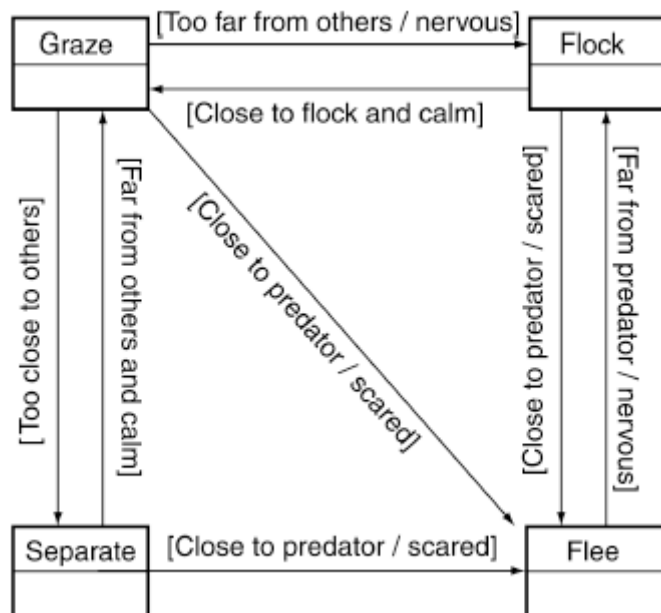
Istnieją kombinacje uczenia się, które pozostawiają sieć neuronową w dużej mierze niezdolną do robienia niczego sensownego. Zarówno w *Creatures*, jak i *Black & White* możliwe jest uczynienie wyuczonej postaci bezsilną. Chociaż możliwe jest uratowanie takiej postaci, rozgrywka jest nieprzewidywalna (ponieważ gracz nie zna rzeczywistego wpływu ich opinii) i żmudna. Ponieważ wydaje się to nieuniknioną konsekwencją zastosowanej sztucznej inteligencji, warto uwzględnić ten wynik w projektowaniu gry.

GRY W ZWIERZĘCIE I WYPADANIE

Proste symulatory pasterskie istnieją od lat 80. XX wieku, ale na początku XXI wieku pojawiło się kilka bardziej udanych gier, aby poprawić stan wiedzy. Gry te polegają na przenoszeniu grupy postaci przez (zwykle wrogi) świat gry. *Herdy Gerdy* jest najbardziej rozwinięty, choć komercyjnie nie radził sobie dobrze. *Pikmin*, *Pikmin 2* i niektóre poziomy *Oddworld: Munch's Oddysee* używają podobnych technik. Stosunkowo duża liczba postaci ma proste indywidualne zachowania, które powodują pojawienie się na większą skalę. Postać będzie gromadzić się z innymi tego rodzaju, zwłaszcza gdy jest narażona na niebezpieczeństwo, i reagować w pewien sposób na graczy (albo uciekając przed nimi, jakby byli drapieżnikami, albo podążając za nimi). Postacie będą reagować i uciekać przed wrogami oraz wykonywać podstawowe sterowanie i unikanie przeszkód. Różne typy postaci są często tworzone w łańcuchu pokarmowym lub ekosystemie, a gracz stara się chronić jeden lub więcej gatunków zdobyczy.

TWORZENIE STWORZEŃ

Każda indywidualna postać lub istota składa się z prostych ram podejmowania decyzji kontrolujących portfolio zachowań sterujących. Proces decyzyjny musi reagować na świat gry w bardzo prosty sposób: może być zaimplementowany jako maszyna skończona (FSM) lub nawet drzewo decyzyjne. Maszyna skończonych stanów dla prostej istoty podobnej do owcy jest pokazana na rysunku.



Podobnie zachowania związane z kierowaniem mogą być stosunkowo proste. Ponieważ tego rodzaju gry są zwykle rozgrywane na zewnątrz w obszarach o niewielkich ograniczeniach, zachowania związane z kierowaniem mogą działać lokalnie i można je łączyć bez skomplikowanego arbitrażu. Rysunek pokazuje zachowania związane z kierowaniem uruchamiane jako nazwa każdego stanu w FSM (pasanie może być zaimplementowane jako powolna wędrówka, od czasu do czasu zatrzymując się na jedzenie). Oprócz wypasu, każde zachowanie kierujące jest albo jednym z podstawowych zachowań dążących do celu (na przykład ucieczka) albo prostą sumą zachowań dążących do celu (takich jak stado). Zobacz rozdział 3 na temat ruchu, aby uzyskać więcej informacji. Rzadko potrzeba wyrafinowanej sztucznej inteligencji dla stworzeń w grze pasterskiej, nawet dla drapieżników. Gdy stworzenie jest w stanie autonomicznie poruszać się po świecie gry, zazwyczaj jest zbyt inteligentne, aby gracz mógł nim łatwo manipulować, a punkt gry jest zagrożony.

STROJENIE STEROWANIA DLA INTERAKTYWNOŚCI

W symulacjach animacji lub efektów tła w grze płynny ruch kierowniczy zwiększa wiarygodność. Jednak w kontekście interaktywnym gracz często nie może wystarczająco szybko zareagować na ruch grupy. Kiedy na przykład stado zaczyna się rozdzielać, trudno jest je okrążyć z wystarczającą prędkością, aby z powrotem je połączyć. Zapewnienie postaci takiej zdolności ruchowej naraziłoby na szwank inne aspekty projektowania gry. Aby uniknąć tego problemu, zachowania układu kierowniczego są zazwyczaj sparametryzowane tak, aby były mniej płynne. Postacie poruszają się małymi zrywaniami, a ich chęć tworzenia spójnych grup wzrasta. Dodanie przerw w ruchu postaci spowalnia ich ogólny postęp i pozwala graczowi okrążyć je i manipulować ich akcjami. Można to osiągnąć, zmniejszając tempo ich ruchu, ale często wygląda to sztucznie i nie pozwala na ciągły ruch z pełną prędkością, gdy są bezpośrednio ścigani. Poruszanie się w zrywach daje również stworzeniu atmosferę ukradkową i nerwową, co może być korzystne. Zarówno pod względem szybkości, jak i spójności, ważne jest zmniejszenie bezwładności poruszających się postaci. Podczas gdy ptaki w symulacjach stada zazwyczaj mają dużą bezwładność (zmiana prędkości lub kierunku wymaga dużo wysiłku), stworzenia, którymi manipuluje gracz, muszą mieć możliwość nagłego zatrzymania się i odejścia w nowym kierunku. Przy dużej bezwładności decyzja, która prowadzi istoty do zmiany kierunku, będzie miała konsekwencje dla wielu klatek i może wpłynąć na ruch całej grupy. Przy małej bezwładności ta sama decyzja jest łatwo odwracana, a konsekwencje są mniejsze. Może to dawać mniej wiarygodne zachowanie, ale jest łatwiejsze (a przez to mniej frustrujące) dla gracza do kontrolowania. Warto zauważyć, że istnieją

prawdziwe międzynarodowe zawody pasterskie, które wymagają lat szkolenia. Trudno zagonić garstkę prawdziwych owiec. Gra prawdopodobnie nie powinna wymagać tego samego poziomu umiejętności, aby była grywalna.

STABILNOŚĆ ZACHOWANIA STEROWANIA

Ponieważ podejmowanie decyzji i kierowanie zachowaniami grupy stworzeń staje się bardziej wyrafinowane, często pojawia się punkt, w którym grupa wydaje się nie być w stanie samodzielnie działać rozsądnie. Często charakteryzuje się to nagłymi zmianami w zachowaniu i pojawieniem się niestabilnego tłumu. Te niestabilności są spowodowane propagacją decyzji przez grupę, często wzmacnianą na każdym kroku. Na przykład grupa owiec może się spokojnie paść. Jeden z nich zbliża się zbyt blisko swojego sąsiada, który odsuwa się, powodując ruch drugiego i tak dalej. Jak we wszystkich procesach decyzyjnych, wymagany jest pewien stopień histerezy, aby uniknąć niestabilności. Owca może być całkiem zadowolona z tego, że inni są bardzo blisko niej, ale zbliży się do nich (tj. utworzy stado), jeśli odejdą daleko. Zapewnia to zakres odległości, w których owca w ogóle nie zareaguje na sąsiada. Istnieje jednak pewien rodzaj niestabilności, która pojawia się w grupie różnych stworzeń, której nie można rozwiązać po prostu za pomocą histerezy w indywidualnych zachowaniach. Grupa stworzeń może wykazywać drgania, ponieważ każda z nich powoduje zmianę zachowania innej grupy. Drapieżnik może na przykład gonić stado ofiar, dopóki nie znajdą się poza zasięgiem. Ofiara przestaje się poruszać, jest bezpieczna i następuje opóźnienie, zanim drapieżnik się zatrzyma. Drapieżnik jest teraz bliżej, a ofiara znowu zaczyna się poruszać. Ten rodzaj oscylacji może łatwo wymknąć się spod kontroli i wyglądać sztucznie. Cykle, które obejmują tylko dwa gatunki, można łatwo dostosować, ale cykle, które pojawiają się tylko wtedy, gdy kilka gatunków jest razem, są trudne do debugowania. Większość programistów umieszcza różne stworzenia w pewnej odległości od siebie na poziomie gry lub używa tylko kilku gatunków na raz, aby uniknąć nieprzewidywalności, gdy wiele gatunków spotyka się jednocześnie.

PROJEKTOWANIE EKOSYSTEMU

Zazwyczaj w grze pasterskiej występuje więcej niż jeden gatunek stworzeń i to właśnie interakcje wszystkich gatunków sprawiają, że świat gry jest interesujący dla gracza. Jako gatunek zapewnia dużo miejsca na interesujące strategie: jeden gatunek może być wykorzystany do wpływania na inny, co może prowadzić do nieoczekiwanych rozwiązań zagadek w grze. W najbardziej podstawowym gatunku gatunek można ułożyć w łańcuch pokarmowy, w którym gracz często ma za zadanie chronić wrażliwą grupę stworzeń. Projektując łańcuch pokarmowy lub ekosystem gry, można wprowadzić efekty niepożądane, ale także pozytywne, ale nieoczekiwane. Aby uniknąć załamania na poziomie gry, gdzie wszystkie stworzenia są szybko zjadane, należy przestrzegać kilku podstawowych wskazówek.

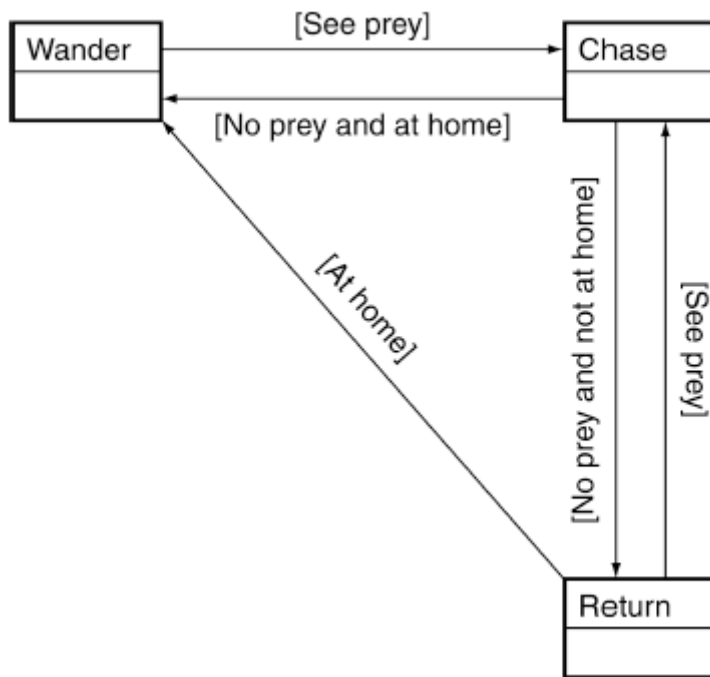
Rozmiar łańcucha pokarmowego

Łańcuch pokarmowy powinien mieć dwa poziomy powyżej twoich podstawowych stworzeń i prawdopodobnie jeden poziom poniżej. Tutaj „główne stworzenia” odnoszą się do stworzeń, którymi gracz zwykle zajmuje się wypasem. Posiadanie dwóch poziomów nad stworzeniami pozwala na skontrolowanie drapieżników przez inne drapieżniki (podobnie jak mysz Jerry używa buldoga Spike'a, aby wydostać się z kłopotów z kotem Tomem). Więcej poziomów i istnieje ryzyko, że „pomocny drapieżnik” nie będzie w pobliżu, aby pomóc.

Złożoność zachowania

Stworzenia znajdujące się wyżej w łańcuchu pokarmowym powinny zachowywać się prościej. Ponieważ gracz pośrednio wpływa na zachowanie innych stworzeń, staje się trudniejsze do kontrolowania wraz

ze wzrostem liczby pośrednich. Przemieszczanie stada stworzeń jest wystarczająco trudne. Używanie tego stada do kontrolowania zachowania innego stworza zwiększa trudności, a następnie używanie tego stworza do wpływania na jeszcze innego – to naprawdę trudne zadanie. Zanim dotrzesz na szczyt łańcucha pokarmowego, stworzenia muszą wykazywać bardzo proste zachowania. Rysunek pokazuje przykładowe zachowanie na wysokim poziomie pojedynczego drapieżnika.



Stworzenia znajdujące się wyżej w łańcuchu pokarmowym nie powinny pracować w grupach. Wynika to z poprzedniej wytycznej: grupy stworzeń pracujących razem prawie zawsze będą miały bardziej skomplikowane zachowanie (nawet jeśli pojedynczo są dość proste). Chociaż wiele drapieżników na przykład w Pikminie pojawia się w grupach, ich zachowanie rzadko jest skoordynowane. Działają po prostu jako jednostki.

Granice sensoryczne

Wszystkie stworzenia powinny mieć dobrze zdefiniowane promienie do zauważania rzeczy. Ustalenie limitu zdolności spostrzegania istoty pozwala graczowi lepiej przewidywać jego działania. Ograniczenie pola widzenia drapieżnika do 10 metrów pozwala graczowi na pokonanie stada na odległość 11 metrów. Ta przewidywalność jest ważna w złożonych ekosystemach, ponieważ umiejętność przewidzenia, które stworzenia zareagują w jakim czasie, jest ważna dla strategii. Wynika z tego, że realistyczna symulacja zmysłów zwykle nie jest odpowiednia dla tego rodzaju gry.

Zakres ruchu

Stworzenia nie powinny z własnej woli oddalać się zbyt daleko. Im mniejsze zaplecze stwora, tym lepiej projektant poziomów może ułożyć poziom. Jeśli stworzenie może wędrować losowo, możliwe jest, że znajdzie się obok drapieżnika, zanim przybędzie gracz. Gracz nie doceni przybycia w miejsce, gdzie okaże się, że stado zostało już zjedzone. Ograniczenie zasięgu stworzeń (przynajmniej do czasu, gdy gracz na nie wpłynie) można również osiągnąć poprzez nałożenie granic świata gry (takich jak płoty, drzwi lub bramy). Zazwyczaj jednak stworzenia po prostu śpią lub stoją, gdy gracza nie ma w pobliżu.

Kładąc wszystko razem

Podobnie jak w przypadku wszystkich AI, najważniejszą częścią uzyskania grywalnej gry jest budowanie i ulepszanie postaci. Pojawiający się charakter gier pasterskich oznacza, że nie da się dokładnie przewidzieć zachowania, dopóki nie będzie można go zbudować i przetestować. Zapewnienie wspaniałych wrażeń z gry na ogół wymaga zdecydowanych ograniczeń w zachowaniu stworzeń w grze, poświęcając pewną wiarygodność dla grywalności.