

Praca Dyplomowa Inżynierska

Rafał Kuligowski
205835

Zastosowanie metod uczenia maszynowego do stworzenia sztucznej inteligencji w turowych grach RPG

The use of machine learning methods to create artificial intelligence in
turn-based RPG games

Praca dyplomowa na kierunku:
Informatyka

Praca wykonana pod kierunkiem
dra Marka Karwańskiego
Instytut Informatyki Technicznej
Katedra Zastosowań Matematyki

Warszawa, rok 2023



SZKOŁA GŁÓWNA
GOSPODARSTWA
WIEJSKIEGO

Wydział Zastosowań
Informatyki
i Matematyki

Oświadczenie Promotora pracy

Oświadczam, że niniejsza praca została przygotowana pod moim kierunkiem i stwierdzam, że spełnia ona warunki do przedstawienia tej pracy w postępowaniu o nadanie tytułu zawodowego.

Data

Podpis promotora

Oświadczenie autora pracy

Świadom/a odpowiedzialności prawnej, w tym odpowiedzialności karnej za złożenie fałszywego oświadczenia, oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami prawa, w szczególności z ustawą z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (Dz. U. 2019 poz. 1231 z późn. zm.)

Oświadczam, że przedstawiona praca nie była wcześniej podstawą żadnej procedury związanej z nadaniem dyplomu lub uzyskaniem tytułu zawodowego.

Oświadczam, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną. Przyjmuję do wiadomości, że praca dyplomowa poddana zostanie procedurze antyplagiatowej.

Data

Podpis autora pracy

Streszczenie

Zastosowanie metod uczenia maszynowego do stworzenia sztucznej inteligencji w turowych grach RPG

Tematem pracy było implementowanie sztucznej inteligencji uczącej się za pomocą metod uczenia maszynowego w turowej grze RPG. Praca składa się z czterech głównych części. Pierwsza część omawia technologie wykorzystane w pracy oraz ich alternatywy. Druga część skupia się na implementacji wcześniej wspomnianych technologii. Trzecia część zawiera instrukcję użytkownika gotowej aplikacji. W czwartej części znajdują się wnioski dotyczące implementacji tematycznego rozwiązania w szerszym kontekście.

Słowa kluczowe – uczenie maszynowe, sztuczna inteligencja, gry RPG, uczenie przez wzmocnianie, turowe systemy walki

Summary

The use of machine learning methods to create artificial intelligence in turn-based RPG games

The topic of the paper was the implementation of artificial intelligence, which learns through machine learning methods, in a turn-based RPG game. The paper consists of four main parts. The first part discusses the technologies used in the work and their alternatives. The second part focuses on the implementation of the aforementioned technologies. The third part contains user instructions for the finished application. The fourth part includes conclusions regarding the implementation of the thematic solution in a broader context.

Keywords – machine learning, artificial intelligence, RPG games, reinforcement learning, turn-based battle system

Spis treści

1	Wstęp	9
1.1	Cel i zakres pracy	9
2	Technologie	10
2.1	Rodzaje uczenia maszynowego	10
2.2	Silniki graficzne oraz pakiety do uczenia	11
2.2.1	Unity z pakietem ML-Agents	11
2.2.2	Godot z pakietem RL Agents	11
2.2.3	Unreal Engine z pakietem Learning Agents	11
2.2.4	Własny silnik z wykorzystaniem różnych pakietów do uczenia maszynowego	12
2.3	Algorytmy do uczenia maszynowego	12
2.4	Systemy walki w turowych grach RPG	13
2.4.1	Standart Turn Battle	13
2.4.2	Active Turn Battle	14
2.4.3	Press Turn System	14
3	Implementacja	16
4	Instalacja i instrukcja korzystania	17
5	Wnioski	18
6	Bibliografia	19

1 Wstęp

Sztuczna Inteligencja w przeciągu ostatnich lat objęła wiele dziedzin życia i jest ciągle dynamicznie rozwijana. Jej zastosowanie jest bardzo rozległe, a jedną z branż o dużych perspektywach rozwoju tej technologii jest przemysł rozrywkowy w który wchodzi gry wideo. W grach można zastosować tą technologię do chociażby generowania misji i zadań dla graczy aby nie były one powtarzalne, stworzenia mądrego systemu automatycznej walki dzięki któremu gracz nie lubiący tego elementu w grze może go pominąć, czy też do stworzenia sztucznej inteligencji dla przeciwników gracza. Ostatnie z tych przykładowych zastosowań zostanie zgodnie z tematem pracy omówione pod względem teoretycznym i praktycznym.

1.1 Cel i zakres pracy

Praca ma na celu zbadanie możliwości implementacji sztucznej inteligencji dla przeciwników w grach wideo z gatunku RPG¹ z turowym systemem walki. Wynikowo otrzymamy 2 aplikacje: trenującą model oraz testującą model w postaci symulatora walki. Treść pracy będą stanowić opisy technologii użytych w pracy wraz z alternatywami, opis implementacji rozwiązania na podstawie wybranych wcześniej technologii, instrukcję obsługi napisanej aplikacji oraz wnioski na temat implementacji takiego rozwiązania w szerszym zakresie.

¹RPG (ang. Role Playing Games) - gry w których gracz wciela się w rolę postaci występujących w fikcyjnym świecie. Gracze ponoszą wszelkie konsekwencje swoich akcji jako postać w świecie gry.

2 Technologie

W tym rozdziale zostaną szczegółowo omówione komponenty potrzebne do implementacji takowej sztucznej inteligencji. Według kolejności wyboru są to:

- Rodzaj uczenia maszynowego
- Silnik do stworzenia gry oraz pakiet do uczenia maszynowego kompatybilny z silnikiem
- Wykorzystywany algorytm do uczenia maszynowego
- System walki (w przypadku tej pracy - system oparty o tury)

2.1 Rodzaje uczenia maszynowego

W uczeniu maszynowym można rozgraniczyć 3 jego główne rodzaje (w oparciu o drugi rozdział z [1]):

- Nadzorowane (ang. Supervised) - model jest trenowany na danych wejściowych oraz odpowiednich dla nich wartościach wynikowych. Na ich podstawie uczy się przewidywania wyniku dla nowych, nieznanych danych wejściowych.
- Nienadzorowane (ang. Unsupervised) - model jest trenowany na samych danych wejściowych nie dostając oczekiwanych wyników. Model sam musi odkryć sens jaki stoi za danymi wejściowymi.
- Przez wzmacnianie (ang. Reinforcement) - model trenowany za pomocą interakcji trenowanego agenta z wykreowanym otoczeniem. Agent wykonując akcje w otoczeniu dostaje nagrody lub kary w zależności od wyników jego akcji.

W przypadku tej pracy użyty zostanie rodzaj uczenia przez wzmacnianie przez charakterystyczną dla niego metodę uczenia przez interakcje. Metoda ta pasuje do gier wideo, gdyż z założenia polegają one na interakcji gracza ze światem gry. Model będzie mógł, tak samo jak gracz grający w grę, poprzez ingerowanie za pomocą akcji w otoczenie uczyć się wykonywania najlepszych możliwych akcji w danym momencie. Więcej wiedzy o tym jak działa uczenie przez wzmacnianie jest zawarte w [2].

2.2 Silniki graficzne oraz pakiety do uczenia

Wybór silnika graficznego oraz pakietu do uczenia maszynowego jest ze sobą głęboko powiązany. Ponadto jest on mocno subiektywny, gdyż jest uzależniony od umiejętności autora projektu. Lista popularnych wyborów w zakresie silnika i pakietu do uczenia maszynowego:

2.2.1 Unity z pakietem ML-Agents

Wśród wymienionych gotowych pakietów, ML-Agents jest najstarszym rozwiązaniem (udostępnione w 2017r.¹). Dostępne w pakiecie są 2 algorytmy uczenia przez wzmacnianie - PPO oraz SAC². Pakiet posiada własną dokumentację[3]. Sam silnik Unity umożliwia tworzenie aplikacji z grafiką 2D oraz 3D, operuje na łatwym w nauce języku C# oraz ma rozbudowaną dokumentację[4].

2.2.2 Godot z pakietem RL Agents

Godot jest silnikiem OpenSource o coraz mocniejszej pozycji na rynku. Według dokumentacji[5] silnik wspiera oficjalnie 2 języki programowania: autorski język GDScript oraz C#. Twórcy za pośrednictwem technologii GDEExtension dodali możliwość dodania wsparcia dla innych języków przez społeczność. Dzięki temu, można na tym silniku programować również w innych językach programowania takimi jak Rust, C++ czy Swift. Pakiet RL Agents jest najbardziej rozbudowanym pakietem wśród wymienionych. Bazuje na 4 innych pakietach³ do uczenia przez wzmacnianie oraz wspiera ponad 12 algorytmów uczenia. Więcej informacji o RL Agents można znaleźć w artykule naukowym poświęconym tej technologii [6] oraz w dokumentacji dostępnej w repozytorium projektu [7].

2.2.3 Unreal Engine z pakietem Learning Agents

Pod względem innych pakietów Learning Agents jest najmłodszy (powstał na początku 2023r.) oraz najslabiej udokumentowany. Pakiet operuje algorytmami PPO oraz BC (ang. Behavioral Cloning)⁴. Mimo to sam silnik Unreal Engine jest potężnym narzędziem z

¹Repozytorium powstało 8.09.2017r, pierwsza wersja pakietu pochodzi z 16.09.2017r - link do repozytorium <https://github.com/Unity-Technologies/ml-agents>

²Więcej informacji o tych algorytmach w rozdziale "Algorytmy do uczenia maszynowego"

³Pakiety na których bazuje RL Agents - StableBaseLines3, SampleFactory, CleanRL oraz Ray rllib (stan na 06.01.2024r.)

⁴Informacja pochodzi ze źródła (stan na 06.01.2024r.): <https://dev.epicgames.com/community/learning/tutorials/80WY/unreal-engine-learning-agents-introduction>

bardzo dużym wsparciem i możliwościami. Posiada wsparcie dla języka C++ oraz funkcji Blueprints umożliwiającą programowanie wizualne (bez pisanie kodu). Dokumentacja zarówno silnika Unreal Engine jak i pakietu Learning Agents znajduje się w [8].

2.2.4 Własny silnik z wykorzystaniem różnych pakietów do uczenia maszynowego

Rozwiązanie oferujące największą swobodę, ale też najtrudniejsze w implementacji. Do stworzenia środowiska gry potrzebna jest biblioteka graficzna pomagająca stworzyć silnik np. OpenGL lub pakiet do stworzenia aplikacji okienkowej np. PyGame. Wybierając pakiet do uczenia maszynowego warto szukać wśród pakietów Pythona gdyż ma on ich bardzo dużo. Dla uczenia przez wzmocnienie można zastosować pakiet PyTorch⁵ lub też jeden z 4 pakietów wykorzystywanych przez RL Agents.

Wybrany silnik i pakiet do uczenia maszynowego

Do pracy wybrałem Unity z pakietem ML-Agents przez moje wcześniejsze doświadczenie z Unity oraz zaawansowanie w języku C#. Uważam jednak, że dużo bardziej rozbudowany RL Agents na silniku Godot jest lepszym wyborem do tego typu projektu, gdyż daje dużo większe pole do dostosowania uczenia dla osiągnięcia lepszych rezultatów.

2.3 Algorytmy do uczenia maszynowego

Unity ML-Agents posiada 2 algorytmy uczące PPO (ang. Proximal Policy Optimization) oraz SAC (ang. Soft Actor-Critic). Pierwszy z nich jest podstawowym algorytmem używanym w pakiecie. Według [3] algorytm PPO jest algorytmem ogólnego przeznaczenia i jest bardziej stabilny w porównaniu z innymi algorytmami uczenia przez wzmacnianie. To samo źródło opisuje również różnicę między algorytmem SAC i PPO. SAC jest algorytmem "off-policy" kiedy PPO jest algorytmem "on-policy". Oznacza to że SAC uczy się z doświadczeń zebranych w dowolnym momencie w przeszłości, a podczas uczenia są losowo dobierane do treningu, gdzie w PPO występuje "policy", które reguluje podejmowanie decyzji i jest ulepszane z każdą iteracją uczenia. Dodatkowe informacje o algorytmie PPO znajdują się w [9], a o SAC w [10]. Do projektu wybrany został algorytm PPO z uwagi na wcześniej podaną stabilność oraz fakt, że jest on algorytmem będącym ponad

⁵Używany w ML-Agents oraz Learning Agents

6 lat w pakiecie ML-Agents co sugeruje, że jest on lepiej rozwiniętym rozwiązaniem w pakiecie.

2.4 Systemy walki w turowych grach RPG

Systemów walki turowej w grach RPG jest bardzo dużo, ale wszystkie charakteryzują się paroma wspólnymi cechami:

- Rozgrywka polega na wymianie tur postaci gracza z przeciwnikami
- Podczas tury gracz oraz przeciwnicy wykorzystują swoje tury jako zasób do wykonywania akcji
- Postacie gracza oraz przeciwnicy posiadają statystyki, które określają ich cechy. Do takich statystyk może należeć np. Obrona zmniejszająca obrażenia otrzymywane przez postać, czy też Atak zwiększający zadawane obrażenia
- Postacie gracza oraz przeciwnicy posiadają umiejętności, których mogą używać zużywając tury do wykonywania akcji w środowisku gry
- Walka kończy się gdy wszystkie postacie gracza lub wszyscy przeciwnicy umrą

Poniżej przedstawię parę podejść do turowego systemu walki wraz z przykładową grą w której został taki system walki użyty:

2.4.1 Standart Turn Battle

Najbardziej podstawowy, najpowszechniejszy i zarazem najprostszy system walki turowej. Na początku walki jest tworzona kolejka tur złożona ze wszystkich postaci na scenie na podstawie statystyki Szybkości. Gdy postać wykorzysta swoją turę jest przenoszona na koniec kolejki i następna postać w kolejce dostaje możliwość wykorzystania tury. Posiada on 2 główne warianty:

1. Postać posiadająca turę od razu po wybraniu ruchu go wykonuje i kolejka przechodzi do następnej osoby. Jest to system wykorzystywany np. w grach z serii Pokemon, czy też jako podstawowy system walki w kreatorze gier RPG o nazwie RPG Maker MV.
2. Wszystkie postacie na początku wybierają ruch jakiego będą używać podczas aktualnego przejścia przez kolejkę. Po wybraniu walka trwa jedno całe przejście kolejki, aby po ruchu ostatniej postaci w kolejce znów poprosić o wybranie ruchu. Jest to rozwiązanie znane np. z gier z serii Etrian Odyssey.

2.4.2 Active Turn Battle

System walki charakteryzuje się tym, że każda postać na początku zaczyna z pustym paskiem tury ładującym się w równych interwałach o wartość zależną od statystyki Szybkości. Im większa owa statystyka, tym szybciej tura jest ładowana. Gdy tura jakiejś postaci zostanie w pełni naładowana, walka zatrzymuje się, a postaci z naładowaną turą wybierają akcję jaką mają wykonać. Następnie pasek tury jest ponownie ładowany. Gdy dobiegnie do końca postać wykona wybraną wcześniej akcję i resetuje ładowanie tury do wartości początkowej. System ten jest znany z gier z serii Final Fantasy.

2.4.3 Press Turn System

Został stworzony i w głównej mierze jest używany w grach firmy Atlus począwszy od gry [11]. System opiera się o zwiększenie znaczenia tury jako zasobu gdzie na początku gra oddziela fazę ruchu gracza i przeciwnika. Następnie tworzy kolejkę ruchów dla uczestników danej fazy na podstawie statystyki Szybkości. Następne kroki jakie podejmuje system są zależne od wariantu tego systemu.

Wariant klasyczny

Występujący w grach z serii Shin Megami Tensei począwszy od [11]. Ilość tur dla strony wykonującej ruchy jest liczona na podstawie ilości postaci po tej stronie. Każda akcja wykonana przez postać jest oceniana przez system i na podstawie oceny zmienia się ilość tur. W przypadku tego modelu musi występować system Elementów, które są przypisane do każdej umiejętności ofensywnej, oraz relacji postaci do Elementu. Elementem mogą być np. żywioły takie jak Ogień, Lód czy Wiatr, natomiast relacjami są skutki Elementu na postać, które potem edytują ilość tur. W grach z tej serii występuje 5 relacji do Elementów:

1. Normalna
2. Słabość
3. Niewrażliwość
4. Odbicie
5. Absorpcja

Jeśli postać trafi w słaby punkt przeciwnika nie traci całej tury, a tura jest "wciskana" (ang. Pressed). "Wciśnięta" tura zachowuje się jak zwykła tura, ale nie może być wciśnięta ponownie. Tury można "wciskać" trafiając w słaby punkt przeciwnika aż do momentu,

w którym nie będziemy mieli dostępnych zwykłych tur. Jeśli posiadamy zarówno zwykłą turę, jak i "wciśniętą" turę to akcja z normalną relacją do Elementu spowoduje zużycie "wciśniętej" tury. Inne typy relacji zmniejszają ilość tur o określoną ilość np. trafienie w Element z relacją Niewrażliwości może obniżyć ilość tur atakującego o 2 priorytetyzując tury "wciśnięte". Jeśli postać postanowi nic nie robić podczas tury, jest ona "wciskana". Gdy stronie wykonującej akcje skończą się tury następuje zmiana stron dla które takie same zasady są zachowane. W przypadku gdy walka jest z silniejszą postacią (tzw. Bossem), która występuje sama system daje tej postaci dodatkową turę na start.

Wariant uproszczony

Występujący w grach z serii Persona począwszy od części trzeciej. W tym przypadku również musi występować system Elementów i relacji opisany jak w wersji klasycznej. Postacie mogą wykonać jedną akcję według kolejki określonej na początku fazy. Jeśli po użyciu umiejętności ofensywnej postać trafi w słaby punkt przeciwnika nie traci ruchu i może użyć czegoś jeszcze raz, a przeciwnik dostaje status "Powalenia". Jeśli trafi się przeciwnika o tym statusie w słaby punkt nie dostaje się dodatkowego ruchu i rusza się kolejna osoba z drużyny. Status "Powalenia" znika gdy postać ma się ruszyć. W każdym innym przypadku ruch jest tracony i rusza się kolejna osoba. W grze Persona 5 dodano dodatkowo mechanikę Baton Pass, która umożliwia oddanie swojej tury po trafieniu w słaby punkt innej osobie z drużyny. Jeśli kolejna osoba trafi w słaby punkt może oddać swoją turę kolejnej osobie z drużyny wyłączając każdą osobę która korzystała z tej mechaniki wcześniej. Po wykonaniu akcji nie dającej dodatkowego ruchu, ruch dostaje następna postać.

Wybrany system walki

Do implementacji został wybrany klasyczny wariant systemu Press Turn. Mimo swojego skomplikowania (co może spowodować dłuższy czas uczenia modelu) posiada on bardzo dużo ciekawych interakcji związanych z turami, które sztuczna inteligencja będzie mogła wykorzystać na swoją korzyść.

3 Implementacja

4 Instalacja i instrukcja korzystania

5 Wnioski

6 Bibliografia

- [1] Shagan Sah, *Machine Learning: A Review of Learning Types*, (2020 r.)
- [2] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G. Bellemare, Joelle Pineau, *An Introduction to Deep Reinforcement Learning*, (2018 r.)
- [3] *Unity ML-Agents Toolkit Documentation* <https://unity-technologies.github.io/ml-agents/ML-Agents-Toolkit-Documentation/> (dostęp 06.01.2024r)
- [4] *Unity Documentation* <https://docs.unity.com> (dostęp 06.01.2024r)
- [5] *Godot Documentation* <https://docs.godotengine.org/en/stable> (dostęp 06.01.2024r)
- [6] Edward Beeching, Jilles Debangoye, Olivier Simonin, Christian Wolf, *Godot Reinforcement Learning Agents*, (2021 r.)
- [7] *Godot RL Agent Repository* https://github.com/edbeeching/godot_rl_agents (dostęp 06.01.2024r)
- [8] *Unreal Engine Documentation* <https://docs.unrealengine.com> (dostęp 06.01.2024r)
- [9] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov, *Proximal Policy Optimization Algorithms*, (2017 r.)
- [10] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, Sergey Levine, *Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor*, (2018 r.)
- [11] Atlus, *Shin Megami Tensei III* (2003 r.)

Wyrażam zgodę na udostępnienie mojej pracy w czytelniach Biblioteki SGGW w tym w Archiwum Prac Dyplomowych SGGW.

.....
(czytelny podpis autora pracy)

