

# 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

<b>1</b>	<b>Wstęp</b>	<b>9</b>
1.1	Cel i zakres pracy . . . . .	9
<b>2</b>	<b>Technologie</b>	<b>10</b>
2.1	Rodzaje uczenia maszynowego . . . . .	10
2.2	Silniki oraz pakiety do uczenia maszynowego . . . . .	11
2.3	Algorytmy do uczenia maszynowego . . . . .	12
2.4	Systemy walki w turowych grach RPG . . . . .	12
<b>3</b>	<b>Bibliografia</b>	<b>13</b>





# 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 RPG<sup>1</sup> 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.

---

<sup>1</sup>RPG (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. 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 oraz pakiety do uczenia maszynowego

Wybór silnika 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:

- Unity z pakietem ML-Agents - Wśród wymienionych gotowych pakietów, ML-Agents jest najstarszym rozwiązaniem (udostępnione w 2017r.<sup>1</sup>). Dostępne w pakiecie są 2 algorytmy uczenia przez wzmacnianie - PPO oraz SAC<sup>2</sup>. 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].
- 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 GDExtension 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 pakietach<sup>3</sup> 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].
- Unreal Engine z pakietem Learning Agents - Pod względem innych pakietów Learning Agents jest najmłodszy (powstał na początku 2023r.) oraz najsłabiej udokumentowany. Pakiet operuje algorytmami SAC oraz Q-Learning<sup>4</sup>. Mimo to sam silnik Unreal Engine jest potężnym narzędziem z bardzo dużym wsparciem i możliwościami. Posiada wsparcie dla języka C++ oraz funkcji Blueprints umożliwiającej programowanie wizualne (bez pisania kodu). Dokumentacja zarówno silnika Unreal Engine jak i pakietu Learning Agents znajduje się w [8].

---

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

<sup>2</sup>Więcej informacji o tych algorytmach w rozdziale "Algorytmy do uczenia maszynowego"

<sup>3</sup>Pakiety na których bazuje RL Agents - StableBaselines3, SampleFactory, CleanRL oraz Ray rllib (stan na 06.01.2024r.)

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

- 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 PyTorch<sup>5</sup> lub też jeden z 4 pakietów wykorzystywanych przez RL Agents.

## **2.3 Algorytmy do uczenia maszynowego**

## **2.4 Systemy walki w turowych grach RPG**

---

<sup>5</sup>Używany w ML-Agents oraz Learning Agents

### 3 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](https://github.com/edbeeching/godot_rl_agents) (dostęp 06.01.2024r)
- [8] *Unreal Engine Documentation* <https://docs.unrealengine.com> (dostęp 06.01.2024r)



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

.....  
(czytelny podpis autora pracy)

