

# Warsztaty z technik uczenia maszynowego - Halite

Wojciech Drochomirecki

Maciej Ejduk

Aleksandra Gajda

Marzec 2022

## 1 Temat

Celem projektu będzie stworzenie algorytmu, który będzie znajdował optymalną strategię ruchów dla gry Halite. Opis gry na Kaggle pod linkiem: [Kaggle](#). Opis Halite na oficjalnej stronie twórcy: [Halite](#). Jest to projekt, który dzięki dużemu wachlarzowi problemów które zawiera pozwoli na zastosowanie różnych algorytmów i wykorzystanie jak największej ilości informacji zdobytych na przedmiocie. Jednocześnie dzięki swojej postaci pozwala na stworzenie rozwiązania zarówno banalnie prostego jak i wysoce wysublimowanego dzięki czemu będziemy mogli dostosować poziom zaawansowania do możliwości w trakcie semestru.

## 2 Zadania w zespole

Sama gra udostępnia dosyć rozbudowany interfejs pozwalający na komunikację z jej środowiskiem, wizualizację rezultatów jak i zbieranie danych z wielu symulacji. Aby efektywnie rozpocząć pracę związaną z implementacją algorytmów ML w naszym rozwiązaniu potrzebujemy dobrze zapoznać się z tym jak on działa oraz poznać jego funkcje. W tym celu każdy wykona swoją wersję podstawowego algorytmu do prowadzenia rozgrywki wykorzystującą proste algorytmiczne podejście do problemu. Następnie rozwiązania będą połączone, lub wybrane zostanie najlepsze z nich które później stanie się podstawą do dalszych prac nad projektem.

Następnym krokiem będzie wdrożenie różnych algorytmów ML w konkretnych decyzjach podejmowanych przez bota. Gra udostępnia rozmaite przestrzenie, w których mogą one zostać wykorzystane. Zśród już rozpoznanych przez nas dziedzin można wyróżnić takie jak sterowanie statkami, interakcje z nieprzyjacielem oraz makrostrategie takie jak tworzenie nowych statków czy gospodarowanie zasobami. W każdej z tych dziedzin planujemy przetestować różne podejścia oparte na wstępnym reaserchu dostępnych rozwiązań. Dodatkowo podczas przeprowadzonych eksperymen-

tów, planujemy zbierać przykładowe rozgrywki ilustrujące typowe dla danego problemu i rozwiązania cechy, które pozwolą na końcu projektu wysnuć wnioski na temat poszczególnych algorytmów.

W tym etapie planujemy przydzielić każdemu z członków zespołu jedną z dziedzin w której rozwinie algorytm ML spełniający którąś z wymienionych wyżej funkcji, lub inną która okaże się bardziej interesująca dla wyników eksperymentu. Dzięki temu członowie zespołu otrzymają pewnego rodzaju niezależność tworząc wspólnie ostateczne rozwiązanie.

Dodatkowo wspólnie będziemy tworzyć dokumentację projektu związaną z odpowiadającymi nam częściami projektu.

## 3 Narzędzia

Narzędzia których będziemy używać w rozwoju naszej aplikacji będą głównie bibliotekami Pythona. Najprawdopodobniej będą to rozwiązania należące do podstawowego kanonu narzędzi wykorzystywanych w zadaniach z zakresu Data Science, takie jak pyTorch, numPy, matplotlib, tensorflow. Dzięki ich powszechnej dostępności i szerokim zastosowaniom będziemy mieli dostęp do rozbudowanych dokumentacji oraz tutoriali pomagających w zrozumieniu sposobu ich działania jak i metod implementacji konkretnych algorytmów. Są to też doskonałe narzędzie pomagające w wizualizacji otrzymanych wyników która pomoże w wyciągnięciu konstruktywnych wniosków z eksperymentu. Do tworzenia wizualnych reprezentacji naszych rozgrywek jak i do interakcji ze światem gry wykorzystamy środowisko udostępnione przez twórców gry. Dzięki temu zredukujemy ilość pracy niezwiązanej bezpośrednio z przedmiotem polegającej na samodzielnej implementacji środowiska gry.

### 3.1 Język programowania

Projekt planujemy wykonać przy użyciu języka *Python*. Jest on, obok *R*, narzędziem powszechnie używanym we wszystkich dziedzinach Data Science i najpopularniejszym do zastosowań związanych z ML oraz AI. Sposób w który jest zaprojektowany pozwala na łatwe prototypowanie rozwiązań w celu przeprowadzenia doświadczeń. Dzięki wykorzystaniu narzędzi takich jak Jupyter Notebook pozwala na bardzo dynamiczną pracę z konkretnymi danymi. Dodatkowo posiada wiele bibliotek oraz rozszerzeń zaprojektowanych do wydajnego implementowania algorytmów ML. Ostatecznym argumentem dla użycia tego języka jest jego pełna integracja ze środowiskiem gry Halite, i jest on wyróżniony jako polecany przez twórców gry.

### 3.2 Przetwarzanie danych

Ze względu na to że zaproponowany przez nas projekt ma dość nietypowy charakter, wszystkie przetwarzane przez nas dane będą wygenerowane na podstawie eksperymentów. Algorytmy które

będziemy stosować będą w większości należały do rodziny Unsupervised Learning takich jak Reinforced Learning czy Evolutionary Algorithms. Dodatkowo dzięki popularności konkursu będziemy mogli przetestować nasze rozwiązania w rywalizacji z botami zaprojektowanymi przez inne zespoły, często profesjonalistów. Pozwoli to na uwypuklenie ich cech w różnych środowiskach.

Dodatkowo wykorzystując dane z poprzednich rozgrywek innych graczy możemy wykorzystać rozwiązania takie jak regresja liniowa do ekstrakcji pewnych zwyciężkich strategii.

### **3.3 Wstępne interakcje z danymi**

Pierwszym krokiem do uzyskania danych potrzebnych w naszym eksperymencie jest podstawowa integracja ze środowiskiem. W tym celu przed pierwszym checkpointem projektu skupiliśmy się na zapoznaniu ze środowiskiem gry. Rozpoczeliliśmy od pobrania i skonfigurowania plików gry na maszynie lokalnej wykorzystując instrukcje twórców konkursu. Następnie uruchomiliśmy podstawowego bota dostarczonego w pakiecie startowym. Zaznajomiliśmy się z API udostępnianym przez środowisko gry, a także poznaliśmy możliwości interakcji z nią poprzez inspekcję plików gry. Ostatecznie podjeliśmy się prostych modyfikacji podstawowego bota, modyfikując sposób sterowania statków, a także próbując zapobiec kolizjom. Dzięki tym operacjom mogliśmy zapoznać się z plikami `zeplay` oraz `errorlog` które są podstawowymi źródłami informacji o przebiegu rozgrywki. Mogliśmy też przetestować różne sposoby produkcji logów, dzięki czemu w przyszłości będziemy mogli szybko zbierać dane z dużej ilości symulacji.