



UNIWERSYTET MARII CURIE-SKŁODOWSKIEJ  
W LUBLINIE

Wydział Matematyki, Fizyki i Informatyki

Kierunek: **Informatyka**

Specjalność: -

**Wiktor Wajszczuk**

nr albumu: 296534

## **Zastosowanie uczenia przez wzmacnianie w środowisku agentowym**

**Application of reinforcement learning in an agent-based  
environment**

Praca licencjacka  
napisana w Katedrze Oprogramowania Systemów Informatycznych  
pod kierunkiem dr Marcina Denkowskiego

**Lublin, rok 2022**

# Spis treści

<b>Wstęp</b>	<b>3</b>
<b>1 Gra</b>	<b>4</b>
1.1 Zasady gry . . . . .	4
1.2 Wykorzystane technologie . . . . .	5
1.3 Implementacja . . . . .	5
<b>2 Uczenie przez wzmacnianie</b>	<b>8</b>
2.1 Q-learning . . . . .	8
<b>3 Implementacja Q-learningu</b>	<b>9</b>
3.1 Podejście pierwsze grid 10 na 10 . . . . .	9
3.2 Podejście drugie grid 10 na 10 i flaga za śmierć od wysokiej rury . . . . .	9
3.3 Podejście trzecie grid 5 na 5 z flagą . . . . .	9
<b>4 Podsumowanie</b>	<b>10</b>
4.1 Co można by ulepszyć . . . . .	10
<b>Bibliografia</b>	<b>10</b>

# Wstęp

Jedną z większych dziedzin uczenia maszynowego jest uczenie przez wzmacnianie (ang. *reinforcement learning*). W odróżnieniu od zarówno uczenia nadzorowanego i nienadzorowanego nie potrzebujemy w tym przypadku żadnych gotowych danych wejściowych i wyjściowych. Zamiast tego, algorytm pozyskuje dane na bieżąco ze środowiska do, którego jest zastosowany. Dzięki temu, że algorytmy uczenia przez wzmacnianie nie mają tego ograniczenia możemy zastosować je do problemów takich jak gra na giełdzie [1], czy nauka grania w gry, w swojej pracy skupię się na tym drugim.

Celem pracy jest zaimplementowanie algorytmu uczenia przez wzmacnianie Q-Learningu. Następnie zoptymalizowaniu go tak by po zastosowaniu go do własnoręcznie zaimplementowanej gry był w stanie osiągnąć w niej możliwie najwyższy wynik. Do zrealizowania tego zostanie użyty język python oraz moduł pygame.

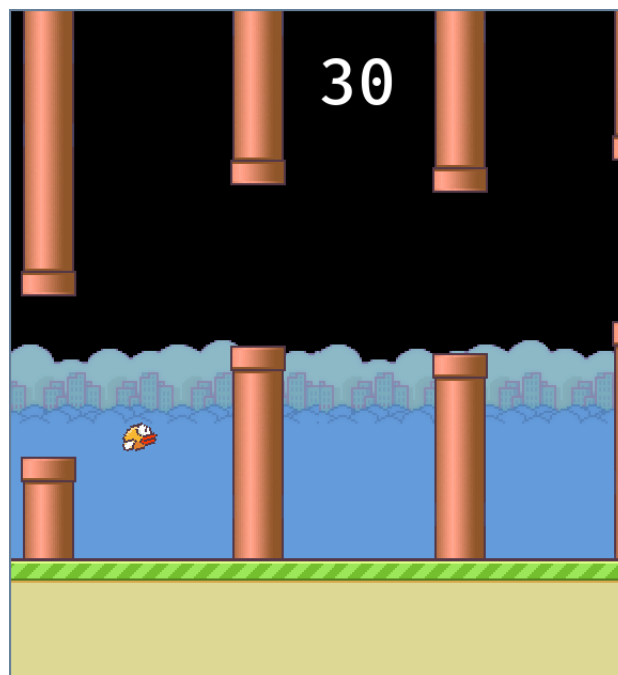
Na początku przedstawię jaką grę wybrałem, opiszę jej zasady oraz zaprezentuję szczegóły jej implementacji. Następnie przybliżę zagadnienie uczenia przez wzmacnianie oraz algorytmu Q-learning. Pod koniec pokażę jak zaimplementowałem wspomniany algorytm, oraz testy po optymalizacjach algorytmu, które zastosowałem.

# Rozdział 1

## Gra

Problem, który postawiłem przed Q-learningiem to nauczenie się grania w “Flappy Bird” – grę z 2013 autorstwa wietnamskiego developera Dong Nguyena [2]. Zdecydowałem się właśnie na tę grę, gdyż sterowanie w niej jest bardzo proste – jest tylko jeden przycisk do kontrolowania toteż jedna akcja do podjęcia przez algorytm.

### 1.1 Zasady gry



Rysunek 1.1: Zrzut ekranu z gry

Gracz kontroluje obiekt reprezentowany w grze jako żółty ptak, jego zadaniem jest tak nim sterować by omijać obiekty reprezentowane przez czerwone rury oraz nie pozwolić na to by siła grawitacji doprowadziła do kolizji obiektu ptaka z obiektem ziemi (na rysunku

1.1 zielona część na dole). Na obiekt ptaka działa jedynie siła grawitacji a obiekty rur przesuwane są w lewo w jego stronę ze stałą prędkością. Gracz ma możliwość kontrolować jedynie czy obiekt, którym steruje zwiększy swoją wysokość o stałą, zakodowaną ilość jednostek czy tego nie robi i obiekt reprezentowany przez ptaka zmniejszy swoją wysokość. Gra kończy się gdy dojdzie do kolizji obiektu ptaka z, którymkolwiek obiektem rur – dolnej bądź górnej albo z obiektem ziemi. Punkty przyznawane są jeżeli graczowi uda się ominąć nadchodzące przeszkody przelatując przez szczelinę między nimi. Za każde ominięcie przyznawany jest jeden punkt. Na górze rysunku 1.1 białymi cyframi wypisana jest ilość osiągniętych przez gracza punktów. Jak widać gracz ominął już trzydzieści par rur. Podsumowując – gra sprowadza się do kontrolowania wysokości ptaka tak by ten przelatywał między nadchodzącymi rurami.

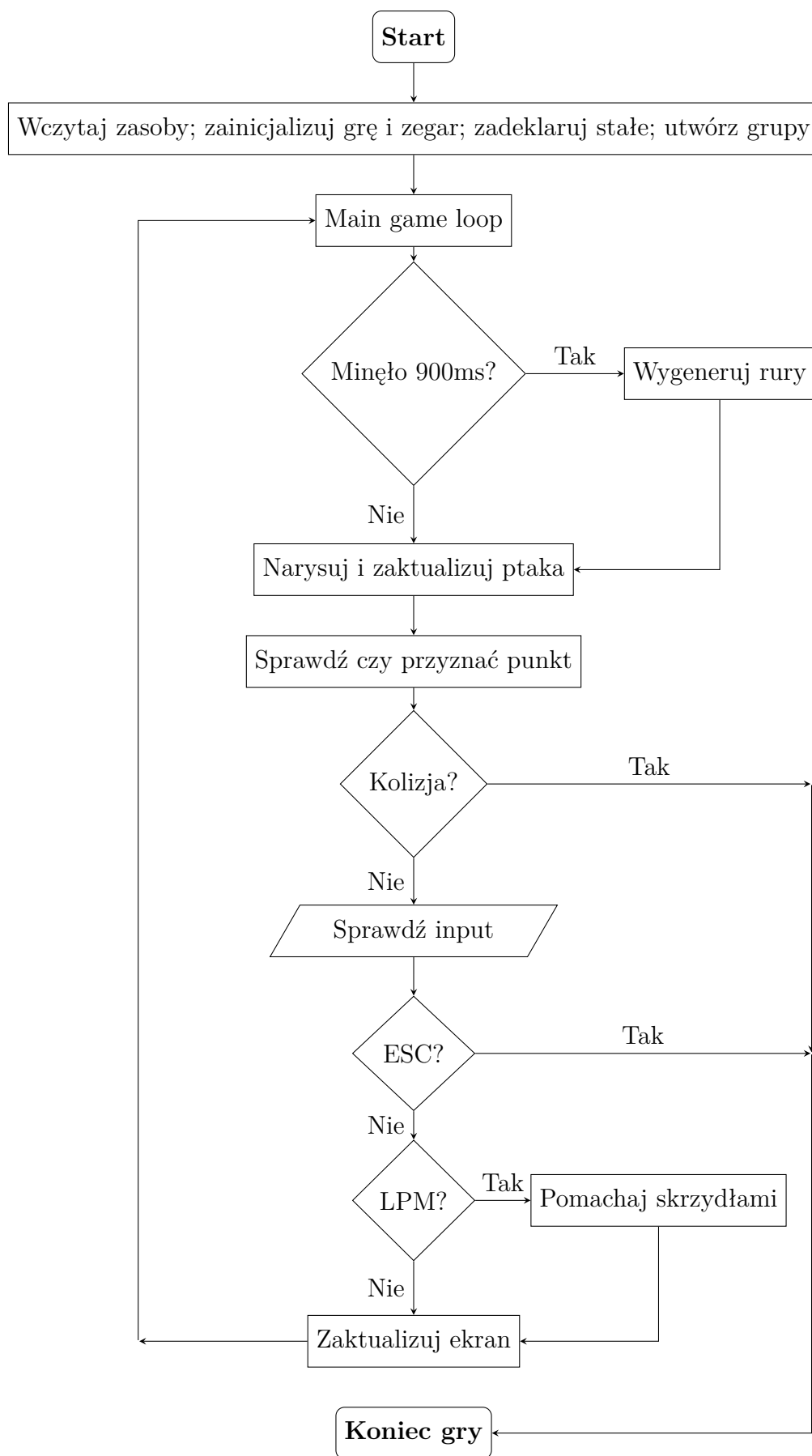
## 1.2 Wykorzystane technologie

Do zaimplementowania tej gry zdecydowałem się na użycie skryptowego języka **python** w wersji 3.8. Python jest dostępny na większości współczesnych platform oraz jest to język open-source. Dodatkowo wykorzystałem moduł **pygame**.

Pygame dodaje funkcjonalność do biblioteki SDL. Simple DirectMedia Layer(SDL) jest to wieloplatformowa biblioteka zapewniająca nisko poziomowy dostęp do audio, urządzeń peryferyjnych takich jak mysz, klawiatura, joystick oraz sprzętu graficznego (za pomocą OpenGL i Direct3D). Technologia open source zaimplementowana w C. Używana w grach firm takich jak *Valve Software* czy *SuperTuxCart*[3]. To za pomocą tej biblioteki pygame może odczytywać wejście czy rysować wyjście na ekranie.

Pygame pozwala tworzyć w pełni funkcjonalne gry oraz programy multimedialne w pythonie. Zaletami pygame są między innymi : podstawowe funkcje używają zoptymalizowanego kodu w C oraz Assembly, dostępny podobnie jak python na większości współczesnych platform. Ponadto jest modularny co pozwala programiście używać tylko tych komponentów, których naprawdę potrzebuje. Gier stworzonych z użyciem pygame na samej stronie projektu jest ponad 660 [4].

## 1.3 Implementacja



Rysunek 1.2: Uproszczony diagram przepływu gry

Tutaj będą opisane po kolei nody co robią na diagramie

# Rozdział 2

## Uczenie przez wzmocnianie

Tutaj do napisania o tym czym jest uczenie przez wzmocnianie pare slow jakie są algorytmy jak działa, schemat środowisko - agent - akcja - reward

### 2.1 Q-learning

Tutaj wszystko związane z qlearningiem, wzory, itp :)



# Rozdział 3

## Implementacja Q-learningu

Tutaj fragmenty kodu, jakie zmiany dokonałem względem książkowego algorytmu

### 3.1 Podejście pierwsze grid 10 na 10

Po kolei wyjaśnienie co to grid 10 na 10 jak to jest implementowane itp

### 3.2 Podejście drugie grid 10 na 10 i flaga za śmierć od wysokiej rury

Jak wyżej

### 3.3 Podejście trzecie grid 5 na 5 z flagą

Jak wyżej

# Rozdział 4

## Podsumowanie

Wyniki osiągane są w miarę ok ale mogło by być lepiej bo są na internecie lepsze podejścia do tego problemu – być może wynika to z tego że sam implementowałem grę i nie jest ona tak dobrze przetestowana jak ta dostępna na GitHub?

### 4.1 Co można by ulepszyć

Tutaj będę pisał co mogłem zrobić gdybym poświęcił więcej czasu np:

- uczenie bez wizualizacji pygamowej
- zrównoleglenie by paru agentów na raz mogło się uczyć
- doszlifowanie algorytmu??

# Bibliografia

- [1] K. Dabérius, E. Granat, and P. Karlsson, “Deep execution - value and policy based reinforcement learning for trading and beating market benchmarks,” 2019. [Online]. Available: <http://dx.doi.org/10.2139/ssrn.3374766>
- [2] “Strona internetowa dotgears.” [Online]. Available: <https://dotgears.com/games/flappybird>
- [3] “Strona internetowa sdl.” [Online]. Available: <https://www.libsdl.org/>
- [4] “Strona about projektu pygame.” [Online]. Available: <https://www.pygame.org/wiki/about>