

## Neutron

Neutron to gra dla dwóch graczy, na planszy o wymiarach 5 x 5. Każdy gracz ma 5 żołnierzy. Istnieje również neutralna figura, Neutron, która jest używana przez obu graczy.

### Zasady gry

Gra zaczyna się z pięcioma pionkami po jednej stronie, pięcioma po drugiej i Neutronem na środku. Wszystkie elementy poruszają się w linii prostej, pionowo, poziomo lub ukośnie, jednak muszą one poruszać się tak daleko, jak mogą iść w wybranym kierunku. Mogą poruszać się tylko po pustych polach, nie ma zbijania ani skoków.

Gra rozpoczyna się, gdy jeden gracz przesuwa pionek ze swojego rzędu. Następnie w każdym kolejnym ruchu gracz przesuwa najpierw neutron, a następnie jeden ze swoich pionków.

### Cel gry

Aby wygrać gracz musi doprowadzić neutron do swojego rzędu domowego. Rząd domowy to pierwszy rząd od strony gracza na polu. Gracz może podczas swojego ruchu doprowadzić neutron do swojego domowego rzędu lub zmusić do tego swojego przeciwnika. Innym sposobem na wygraną jest zagięcie swojego przeciwnika. Czyli nie pozwolić przeciwnikowi przesunąć neutronu i zakończyć ruch.

### Opis działania

Na początku gracz powinien wybrać rodzaj gry albo przeczytanie reguł. W tym celu wprowadza odpowiednią liczbę.

Aby móc przesunąć dowolną figurę na planszy, należy wprowadzić współrzędne pozycji początkowej i docelowej w postaci liczb X i Y. Na przykład dla żółtego pionka ze współrzędnymi wyjściowymi (2, 5) i końcowymi (2, 2) należy wpisać:

```
Write X and Y of pawn: two numbers from 1 to 5 with space

Player-1 - ● | Player-2 - ●
Y|X  1  2  3  4  5
1  ●  ●  ●  ●  ●
2  o  o  o  o  o
3  o  o  🐾  o  o
4  o  o  o  o  o
5  ●  ●  ●  ●  ●

Player-1 ●, move a piece:
From X Y: 2 5
HINTS
Your possible moves: [[2, 2], [1, 4], [5, 2]]
To X Y: 2 2

Y|X  1  2  3  4  5
1  ●  ●  ●  ●  ●
2  o  ●  o  o  o
3  o  o  🐾  o  o
4  o  o  o  o  o
5  ●  o  ●  ●  ●
```

## Różne rodzaje gier

Na początku rozgrywki jest możliwość wybrać jeden z następujących rodzajów gier:

- człowiek z człowiekiem:

podczas każdego ruchu gracz\_1 i gracz\_2 powinni wprowadzać dane do przesunięcia figur na żądanie;

- człowiek z komputerem, wersja łatwa:

komputer w wersji łatwej losowo wybiera swój ruch;

- człowiek z komputerem, wersja trudna

komputer w wersji trudnej wybiera swój najlepszy ruch według określonych kryteriów;

- wersja trudna komputera z wersją łatwą.

## Osobliwości

W przypadku trudniejszej wersji gry z komputerem, wprowadzono zmiany: pierwszy rozpoczyna grę komputer, dla bardziej skomplikowanej rozgrywki.

## Algorytm wyboru ruchu dla złożonego komputera

Dla ruchu neutrona, jeśli to możliwe, komputer:

- przemieszcza neutron do swojego rzędu domowego – zwycięstwo;
- nie przemieszcza neutronu na rząd przeciwnika – przegrana;
- przy wyborze punktu końcowego sprawdza możliwość przesunięcia neutronu na rząd przeciwnika z tego punktu (dopuszczalnego jako możliwy), czyli sprawdza możliwość zwycięstwa przeciwnika w kolejnym ruchu.

Dla ruchu pionka, jeśli to możliwe, komputer:

- przesuwa swojego pionka do rzędu przeciwnika, zamykając pustą kratkę - aby uniemożliwić przeciwnikowi możliwość umieszczenia neutronu w jego rzędzie domowym.

## Architektura projektu

main.py - ten plik jest przyznaczony do uruchamiania programu, zawiera obiekt klasy Game.

class\_game.py - ten plik zawiera klasę Game, która kontroluje całą grę, kolejność rozgrywek.

class\_board.py - zawiera klasę Board, w której wszystkie ruchy figurek na planszy są kontrolowane, wykonywane i wyświetlane.

class\_pawn.py - zawiera klasę Pawn, w której prezentowane są białe, czarne pionki i neutron; ogólnie dla każdej figurki na planszy, jest opisany ruch dla niej, ustalają się wszystkie możliwe ruchy;

class\_figure.py - zawiera klasę Figure, która wskazuje na określoną figurę na planszy - neutron, biały pionek czy czarny pionek

class\_empty.py - zawiera pustą klasę pola, pustą komórkę, która nie ma możliwych ruchów ani kolorów

errors.py - zawiera klasy do obsługi błędów: NotImplementedError, IncorrectNumberError, MoveEmptyError;

test\_class\_board.py, test\_class\_pawn.py, test\_class\_empty.py, test\_class\_figure.py – pliki z testami jednostkowymi do odpowiednich klas.

## **Uruchamianie programu**

Dla uruchamiania gry niezbędnym jest:

- 1) Pobranie tego repozytorium
- 2) Otwarcie folderu z plikami
- 3) Wpisanie w terminalu *python3 main.py*

## **Podsumowując:**

Dzięki bardzo intensywnemu kursowi i dużej ilości informacji, ja, jako człowiek, który nigdy wcześniej nie programował, potrafiłam napisać taki projekt.

Dzięki tej grze, można rozwijać pamięć, myślenie logiczne i przestrzenne. Ogólnie, wykonanie projektu było bardzo ciekawe i korzystne. Na tym etapie wszystkie ustalone przeze mnie cele zostały wykonane. Dodatkowo można byłoby zaimplementować jeszcze GUI za pomocą QT. Ale ponieważ to nie było głównym celem, całe skupienie zostało skierowane w stronę zaimplementowania algorytmu, według którego działa program. Następnym krokiem, który chcę wprowadzić, jest sztuczna inteligencja, żeby rozszerzyć funkcjonalność gry.