

Projekt Chaos i Porządek – Adam Misiak

Chaos i **Porządek** to gra podobna do gry w kółko i krzyżyk, rozgrywana na planszy o rozmiarach 6 x 6.

W grze posługujemy się 2 zestawami znaków, X oraz O (od tej pory będę posługiwał się terminem pionek).

W grze występują 2 instancje, chaos i porządek:

Porządek ma za zadanie ustawić z rzędu 5 pionków tego samego typu.

Chaos ma zapobiec wykonaniu zadania przez Porządek.

Rozpoczęcie gry zaczyna się na pustej planszy. Oboje z graczy mogą wykorzystywać oba z pionków (zarówno X jak i O).

Pierwszy ruch wykonuje **Porządek**, następnie następuje zmiana co turę. Pionki można kłaść wyłącznie na puste pola, oraz raz postawionego pionka nie można już przełożyć.

Co warto zauważyć, wygrana jest osiągnięta jedynie przez posiadanie 5 pionków z rzędu, jeżeli **Porządek** posiada 6 pionków z rzędu, nie jest to uznawane jako wygrana.

Klasy:

- **Game:**
Nie przyjmuje parametrów, wywołuje ciąg sekwencyjny uszeregowujący powstanie rozgrywki, inicjalizuje obiekty innych klas
- **Board:**
Klasa planszy:
Generuje plansze do gry, posiada metody potrzebne do komunikacji między planszą do gry a użytkownikiem, np. wstawianie elementu pod wskazanym polem na planszy
- **Enemy:**
Klasa reprezentująca wroga, wróg może stawiać pionki na planszy w sposób uzależniony od trybu gry:
 - **Random**
Wróg stawia pionki na losowe puste pola

- AIChaos:
Wróg stawiający możliwie dobry ruch w zależności od poprzedniego ruchu użytkownika
- AIOrder:
Wróg stawiający możliwie dobry ruch w zależności od analizy środka planszy pod względem wyszukania najdłuższej nieprzerwanej sekwencji elementów
- Window:
Klasa reprezentująca okno interfejsu graficznego, umożliwia komunikację użytkownika z programem np. podczas wyboru trybu gry.

Instrukcja użytkownika:

Aby uruchomić program należy wywołać funkcję main z pliku main.py (funkcja ta jest uruchamiana wraz z uruchomieniem pliku)

Po uruchomieniu gracz ma możliwość dokonywania wyborów, dostępne wybory charakteryzują się podświetleniem na zielono przycisku po najejchaniu. Wybór jest zatwierdzany przez kliknięcie lewego przycisku myszy.

Po wybraniu ustawień następują gra na planszy:

- kliknięcie lewym przyciskiem myszy skutkuje postawieniem X we wskazanym miejscu
- kliknięcie prawym przyciskiem myszy skutkuje postawieniem O we wskazanym miejscu
- kliknięcie w zajęzione miejsce nie wywoła żadnego efektu

Gra jest zakończona gdy wykryta zostanie sekwencja 5 jednakowych znaków, lub skończą się puste miejsca na planszy.

Warstwa trwałości:

Jeżeli użytkownik opuści grę przed jej zakończeniem stan planszy zostaje zachowany do pliku oraz użytkownik ma możliwość powrotu do porzuconej wcześniej rozgrywki

Jeżeli plik z zapisem został uszkodzony (brakuje w nim danych lub dane te są nieprawidłowe), zostaje zgłoszony błąd, oraz zamiast załadowania mapy rozgrywka tworzona jest od nowa.

Plik ze stałymi:

W pliku constants.py przechowywane są stałe, które używane są do prawidłowej obsługi programu. Większości z nich nie należy zmieniać, ponieważ będzie to skutkować nieprawidłowym działaniem programu.

Istnieje możliwość zmiany zawartości stałej PATH, determinującą ścieżkę do pliku z zapisem danych gry.

Można także zmieniać stałe dotyczące kolorów, aczkolwiek nie jest to zalecane.

Refleksje:

Muszę przyznać, że tworzenie programu w interfejsie graficznym okazało się przyjemniejsze niż początkowo sądziłem.

Pierwotna wersja programu była wyświetlana w terminalu, aczkolwiek absolutnie nie żałuję zmiany na GUI.

Gra przeprowadzona za pomocą GUI jest po prostu o wiele przyjemniejsza.

Muszę jednak przyznać, że największą radość odczułem, gdy prawidłowo zaczął działać przeciwnik EnemyAIOrder. Zadowolony jestem z autorsko wymyślnego systemu przypisywania wag do różnych możliwości ruchu, w momencie gdy przegrywałem z Botem, były to słodkie porażki, które cieszyłem się że poniosłem.

Całość programu starałem się opatrzyć w docstringi oraz komentarze, dlatego mam nadzieję, że zrozumienie działania programu nie stanowiło problemu.