

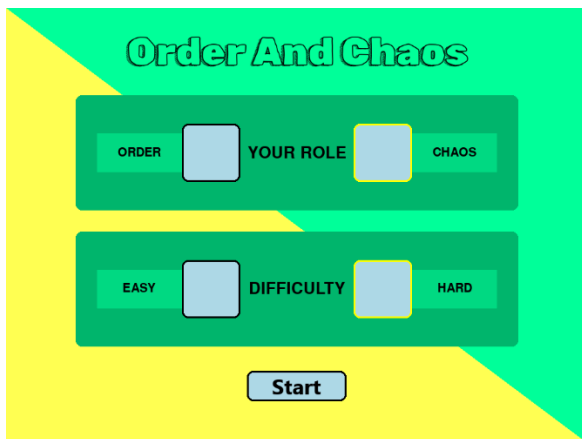
Order and Chaos

A python implementation of „Order and Chaos” tic tac toe game variation with GUI based on Pygame graphic library.

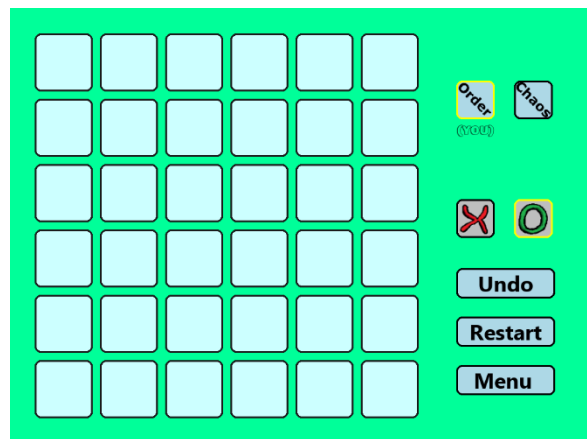
Cel i założenia projektu:

Stworzenie prostej aplikacji z interfejsem graficznym, umożliwiającej grę przeciwko bot’u jako ‘Order’ i ‘Chaos’ na różnych poziomach trudności – łatwym i trudnym.

Prosty ale przejrzysty i wygodny w użytku interfejs graficzny pozwala na łatwy wybór: roli, poziomu trudności, cofania poprzedniego ruchu, czy restartowania gry.



Menu



Gra

Opis najważniejszych klas:

Klasa GameEngine:

Klasa GameEngine odpowiedzialna za przetrzymywanie wszystkich obiektów innych klas. Stanowi główną klasę aplikacji, funkcja „run()”, w której znajduje się główna pętla aplikacji. Odpowiada za wyświetlanie, oraz obsługę okna aplikacji. Łączy interfejs graficzny (przyciski, tekst) z logiczną stroną aplikacji (bot, plansza). Posiada dedykowane metody do obsługi aplikacji takie jak:

- Wczytywanie ścieżek zdjęć i czcionek,
- Wyświetlanie Menu i Gry,
- Aktualizowanie Menu i Gry,
- Przetwarzanie danych wejściowych użytkownika.

Klasa GameBoard:

Klasa odpowiadająca za plansze do gry. Plansza jest przechowywana jako lista liczb całkowitych o wartościach [0, 1, 2], gdzie 0 – puste pole, 1 – kółko, 2 – krzyżyk. Główne funkcje klasy to:

- Metoda do wyświetlania planszy w oknie aplikacji jako 36 kwadratów,
- Kalkulacja pola planszy, nad którym znajduje się kursor użytkownika,
- Cofanie wykonanych ruchów.

Klasa i algorytmy Bot'a:

Bot 'Order' głównie kieruje się chęcią wygrania tych wierszy/kolumn/przekątnych (dalej określane jako **ciągi**), które są najbliższe zwycięstwa, tzn. są 'wygrywalne' i posiadają największą ilość jednakowego symbolu (kółko, krzyżyk). Przede wszystkim stara się najpierw uzupełnić środkowe wartości ciągu, gdyż brzegi są łatwo blokowane, ze względu na fakt, iż ciąg długości 6 tego samego symbolu, nie jest ciągiem wygrywającym. Dla zwiększenia różnorodności i ciekawości rozgrywki, pierwszy ruch bota jest losowy. Podobnie gdy bot ma do wyboru kilka pól środkowych (indeksy od 1 do 4), wybiera je losowo. Dopiero na samym końcu bot wybiera skrajne pola ciągu. Algorytm jest jednakowy do ciągów o długości 5 i 6.

Bot 'Chaos' dąży do blokowania ciągów najbliższych zwycięstwa, poprzez stawianie odwrotnego symbolu, do symbolu dominującego w danym ciągu, lub poprzez dodanie identycznego symbolu na początku i końcu ciągu, tak aby ciąg tych znaków był długości 6 (taktyka stosowana wyłącznie do ciągów o długości 6). Jeżeli ruch bota nie powoduje oznaczenia ciągu jako 'niewygrywalnego', pomija go i przechodzi do następnego możliwego, jeżeli ten ciąg jest jedynym ciągiem najbliższym do zwycięstwa, dokonuje on losowego ruchu (nie powodującego zwycięstwa), ponieważ w jego następny ruch sprawi, że dany ciąg będzie 'niewygrywalny', bądź też dany ciąg jest niemożliwy do zablokowania – wygrana 'Order'u'.

Do określania 'bliskości' zwycięstwa danego ciągu bot korzysta z funkcji „find_arrays_closest_to_win()”, która zwraca listę ciągów posiadającą jednakową ilość brakujących symboli do zwycięstwa, spośród tych, które są oznaczone jako 'wygrywalne'.

Natomiast określanie 'wygrywalności' jak i zwycięstwa ciągów następuje w funkcji „update_arrays()”, która za pomocą dwóch niezależnych funkcji „check_array_winnability()” oraz „check_array_win()”, sprawdza każdy ciąg o długości 5 i 6 z planszy. Funkcja aktualizowania ciągów usuwa dany ciąg z ciągów do sprawdzenia („indexes_arrays”), jeżeli ten jest oznaczony jako 'niewygrywalny' przez funkcję sprawdzania 'wygrywalności', oraz zwraca wartość **True**, gdy jeden z ciągów jest oznaczony jako 'zwycięski'.

Cały 'interfejs' bota spajają dwie główne publiczne funkcje „make_move(role)” i „check_winning()”, które za pomocą innych wbudowanych funkcji bota pozwalają na łatwe, proste, i głównie przejrzyste wykonywanie ruchów i sprawdzanie stanu gry.

Podsumowanie i wnioski z projektu:

Podsumowując, projekt aplikacji „Order and Chaos” pozwala na proste i wygodne w obsłudze granie w zmodyfikowaną wersję klasycznego „kółka i krzyżyk”. Bot na zaawansowanym poziomie trudności potrafi wygrać na niedoświadczonego użytkownika.

Dodatkową funkcją, która jest domyślnie wyłączona (wymaga ręcznej zmiany w kodzie), jest gra „bot vs bot”.

Autor:

Michał Eugeniusz Bloch

Indeks 331360