

Cel i opis projektu :

Celem projektu było umożliwienie zagrania w grę planszową Neutron. Grę dwuosobową, gdzie na przeciwległych krawędziach planszy (standardowo o rozmiarach 5x5) rozmieszczone są pionki gracza i jego przeciwnika, oraz pionek neutronu w centrum. Każdy z graczy po kolei wykonuje ruch składający z przesunięcia się swojego pionka po linii prostej aż do natrafienia na przeszkodę, a następnie wykonanie ruchu pionkiem neutronu w ten sam sposób. Gra kończy się gdy jednemu z graczy uda się przesunąć neutron na kraniec planszy na którym zaczynały pionki przeciwnika, lub neutron zostanie unieruchomiony, w którym przypadku gracz który nie ma możliwości poruszenia neutronem w swojej turze przegrywa.

Instrukcja obsługi:

Należy uruchomić plik Main.py, Pojawi się okienko w którym należy wybrać wymiary planszy, oraz rodzaje graczy. Wymiary planszy muszą być liczbami całkowitymi, nieparzystymi, większymi od 2 i mniejszymi od 20. Możliwe są 3 rodzaje graczy.

- Player. Interaktywny rodzaj gracza, którym można grać używając myszki. Gracz najeżdża i klika na pionek którym zamierza się poruszyć po czym przesuw kursor w kierunku w którym zamierza go przesunąć i klika lewy przycisk myszy. Zmiana pionka przed wykonaniem ruchu jest możliwa.
- Random. Gracz który wybiera swój następny ruch czysto losowo.
- AI. Gracz który, jeśli nie może wygrać w jednym ruchu, wykonuje ruch losowo, uważając żeby nie skutkował w natychmiastowej porażce

Po wybraniu odpowiednich danych okno się zamknie i otworzy się nowe prezentujące planszę. Napis na środku planszy przekaże która strona zaczyna i po zniknięciu komunikatu gra się rozpoczyna. Po zakończeniu gry Pojawi się komunikat stwierdzający kto wygrał.

Podział programu na klasy i ich opis:

Funkcja Main() – Tworzy główną pętlę gry, łączy grafikę z samą grą

MainGui – tworzy okno pobierające dane na początku rozgrywki

Window – jest odpowiedzialne za tworzenie i aktualizacje okna wyświetlającego planszę

Game – tworzy pionki, graczy i plansze i następnie łączy je ze sobą.

Board – przechowuje i przekazuje informacje jak są rozmieszczone na planszy pionki względem siebie.

Pawn – klasa rodzic wszystkich klas pionków

NeutronPawn, TopPawn, BottomPawn – podklasy Pawn. Posiadają informacje o indywidualnych pionkach. Na przykład do kogo należą, albo jakiego mają być koloru

Type – podklasa enum, przechowuje możliwe typy graczy

Set – podklasa enum, przechowuje informacje o początkowym umiejscowieniu

Colors – podklasa enum, zawiera kody wszystkich użytych kolorów

Player – klasa rodzic wszystkich rodzajów graczy, porusza pionkami

PlayablePlayer – klasa gracza interaktywnego

PlayablePlayerGui – klasa obsługująca część graficzną klasy PlayablePlayer

AIPlayer – klasa gracza sztucznej inteligencji, decyduje o najlepszym możliwym ruchu

RandPlayer – klasa gracza poruszającego się losowo

Test_Game.py – zawiera testy

Refleksja:

Trzymanie lokalizacji pionków zarówno w klasie pionków jak i na tablicy nie wydaje się być najlepszym rozwiązaniem. W przypadku, gdy z jakiegoś powodu, dane nie będą ze sobą zgodne, cała gra może przestać działać poprawnie.

Nie byłem w stanie wpaść na pomysł jak zaimplementować w lepszy sposób rejestrowanie lokalizacji pionków w określonym przez siebie czasie na zaprojektowanie szkieletu programu, a ponieważ nie byłem w stanie określić jak długo zajmie mi napisanie samego projektu, uznałem że sposób jest wystarczająco dobry, upewniłem się tylko, żeby lokalizacja była nadpisywana w minimalnej ilości miejsc, by jak najłatwiej wykryć i naprawić możliwe problemy.

Projekt dał mi możliwość poeksperymentowania z graficznymi bibliotekami, co było ciekawym doświadczeniem. Możliwość stworzenia programu, którego rzeczywiście mogę użyć była bardzo satysfakcjonująca.

Sama gra wydaje się być w pełni odporna, na nieprawidłowo wprowadzone dane. Udało mi się w wielu miejscach uogólnić program, by nie przyjmował tylko jednej wartości, a zamiast tego umożliwiał wprowadzenie różnych zmiennych.