

PROJEKT PIPR – „Statki”

AUTOR: Jerzy Muszyński, Informatyka gr 202.

Cel Projektu:

Głównym celem projektu było stworzenie interaktywnej aplikacji do gry w statki, pozwalającej na walkę z komputerem. Użytkownik miał zmierzyć się z „wirtualnym” przeciwnikiem, który naśladuje człowieka i jest w stanie wykonywać logiczne ruchy, w celu zatopienia wszystkich statków.

Opis Projektu:

Projekt został wykonany w języku *Python* a rozgrywka odbywa się poprzez okno konsoli bądź terminalu. Do dyspozycji są plansze w rozmiarze od 1x1 do 26x26, na których ulokowanych może być maksymalnie 5 statków, które są 1,2,3,4 i 5 masztowcami. Użytkownik na zmianę z komputerem „strzelają” w wybrane współrzędne do momentu zatopienia przez któregoś z nich wszystkich statków przeciwnika.

Użyte klasy:

- *Warship* – klasa opisująca pojedynczy statek na planszy, każdy z nich wyróżnia się w szczególności rozmiarem i „blokami”, które reprezentują poszczególne współrzędne tworzące ten statek. Statek może być maksymalnie 5 masztowy i przewidziano jedynie „jednowymiarowe” warianty, tzn. zawsze jeden z wymiarów statku będzie wynosił 1 (długość lub szerokość)
- *Board* – klasa opisująca planszę. Plansza jest zawsze macierzą kwadratową a jej możliwe wymiary należą do przedziału $<1 \times 1, 26 \times 26>$. Plansze $N \times N$, gdzie $N \leq 4$ zawierają N statków, natomiast plansze $N \times N$, gdzie $N > 4$, zawierają zawsze 5 statków, (statki są obiektami klasy *Warship*)
- *Player* – klasa opisująca gracza. Każdy gracz charakteryzuje się swoją własną planszą (obiekt klasy *Board*), na której dobrowolnie alokuje przypisane statki.
- *Ai* – klasa opisująca „wirtualnego” gracza, dziedziczy po klasie *Player*. Dla jej planszy ułożenie statków jest wybierane w pełni losowo. Wykonuje logiczne ruchy, na podstawie rezultatów poprzednich strzałów. Dokonuje stałego zestawienia wszystkich uprzednio wykonanych strzałów oraz strzałów trafionych aby następnie wybrać możliwe pozostałe „bloki” jeszcze nie zatopionych statków.
- *Game* – klasa opisująca główną logikę rozgrywki. Zarządza kolejnością ruchów, pilnuje stanu gry i ustala rezultat na podstawie przebiegu gry.

Instrukcja Obsługi:

Obecnie program należy w środowisku wirtualnym, którego konfiguracja znajduje się w pliku requirements.txt

Należy zatem:

1. Utworzyć środowisko wirtualne `python3 -m venv .venv`
2. Uruchomić środowisko wirtualne `source .venv/bin/activate`
3. Zainstalować wymagane biblioteki `pip install -r requirements.txt`
4. Uruchomić plik z główną pętlą gry `python3 warships/play.py`