Projekt PIPR 1

Gra Shannon Switching – typu Gale

wykonany przez: Igor Matynia

Spis treści

1 Cel i opis projektu

Celem projektu była nauka podstaw programowania i języka python. Do zrealizowania dostałem grę Shannon Switching – Gale. Realizację podzieliłem na pod-problemy zaimplementowane w osobnych modułach:

- ai → gracze komputerowi
- board → logika planszy gry
- FAU → graf „find and union” potrzebny do funkcjonowania logiki planszy i przechowywania innych informacji dla graczów komputerowych
- game → całościowa realizacja gry i metod potrzebnych do jej obsługi
- gui → interfejs użytkownika

2 Klasy programu

Szczegółowa dokumentacja znajduje się w kodzie.

2.1 Moduł ai

ExtendedMove

Przechowuje informacje o ruchu, który chce wykonać gracz komputerowy.

Agent

Bazowa klasa po której dziedziczą gracze komputerowi.

DumBot

Gracz komputerowy, który losuje swój następny ruch.

HeuriBot

Gracz komputerowy, który wybiera z dostępnych ruchów najbardziej optymalny wedle ustalonych heurystycznych kryteriów.

SwitchBot

Gracz komputerowy który wedle podanego współczynnika losuje czy ruch ma być wykonany przez gracza DumBot czy gracza HeuriBot.

2.2 Moduł board

Board

Trzyma stan planszy gry, operacje typu sprawdzanie wygranej czy stawianie połączenia.

BoardField

Bazowa klasa z której dziedziczą wszystkie pozycje na planszy. Jest elementem składowym Board.

Connector

Reprezentuje pole połączenia na planszy. Przetrzymuje informacje o połączeniu w danym miejscu.

StaticVertex

Reprezentuje statyczne pola danego gracza które się łączy połączeniami. Zawiera w sobie Vertex zawarty w grafie połączeń FAU.

2.3 Moduł FAU

Vertex

Reprezentuje wierzchołek w grafie FAU. Przechowuje różne informacje potrzebne do funkcjonowania HeuriBot’a.

FAUGraph

Klasa grafu typu find and union. Reprezentuje spójne wytworzone z połączeń na planszy gry. Obsługuje łączenie spójnych i zwracanie ich reprezentantów (w kodzie nazywanych master’ami).

2.4 Moduł game

GameConfig

Przechowuje informacje o konfiguracji gry. Zawiera w sobie wszystkie dane potrzebne do stworzenia obiektu gry.

GaleShannonGame

Klasa obsługująca wszystkie funkcje niezbędne do grania w grę.

2.5 Moduł gui

MainWindow

Klasa dziedzicząca po QmainWindow zajmująca się wyświetlaniem całego interfejsu użytkownika.

BoardFieldDisplay

Bazowa klasa z której dziedziczą pola planszy wyświetlane w interfejsie użytkownika. Dziedziczy po QLabel.

ConnectorDisplay

Widget QT odpowiedzialny za wyświetlanie pola połączenia planszy gry w interfejsie użytkownika. Dziedziczy po BoardFieldDisplay. Obsługuje kliknięcie oznaczające postawienie połączenia. Analogiczny do Connector z modułu Board.

StaticVDisplay

Widget QT odpowiedzialny za wyświetlanie statycznego pola gracza na planszy. Analogiczny do StaticVertex z modułu Board.

3 Instrukcja obsługi

3.1 Uruchamianie

Aby włączyć grę należy uruchomić interpreterem python'a plik main.py znajdujący się w najwyższym katalogu projektu.

3.2 Format pliku zapisu gry

Gra zapisana i odczytana może być z pliku json zawierającego te pola:

- "connections"→ lista zapisanych połączeń, gdzie każde połączenie zawiera informację o swojej pozycji „x", „y" oraz kierunku „direction". Może być pusta.
- "n" → wielkość planszy
- "opponents_last_move" i "players_last_move" → pary liczb opisujące pozycje ostatnich ruchów graczy
- "who_starts" → kod gracza którego jest kolej na ruch
- "opponent_type" → nazwa typu przeciwnika
- "fraction" → jeśli przeciwnik jest typu "SwitchBot", to tu zapisany jest jego współczynnik

3.3 Obsługa programu

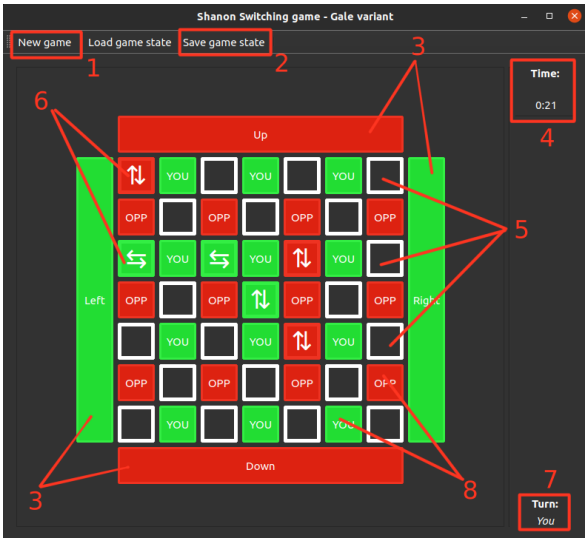
Menu główne

Po włączeniu gry otwiera się menu główne, tutaj użytkownik konfiguruje swoją grę:

1. Ładowanie stanu gry z pliku json
2. Wybór wielkości planszy
3. Wybór typu przeciwnika, liczba zmiennoprzecinkowa w nawiasie oznacza współczynnik SwitchBot'a, przesunięcie gałki skrajnie na prawo ustawia możliwość gry z innym graczem (bez gracza komputerowego)
4. Wybór która ze stron zaczyna rozgrywkę
5. Przycisk startu gry

Plansza

Po wciśnięciu przycisku startu gry lub załadowaniu pliku stanu gry otwiera się okno planszy:



1. Przekierowanie do menu głównego
2. Zapis stanu gry do pliku
3. Brzegi planszy
4. Licznik pokazujący czas gry
5. Nie zajęte pola połączeniowe planszy
6. Zajęte pola połączeniowe planszy
7. Tekst wskazujący na to kto ma teraz grać (You oznacza gracza zielonego, Opp oznacza gracza czerwonego)
8. Stałe pola graczy

W tym oknie rozgrywa się gra. Gracz, którego jest w danej chwili kolej, stawia swoje połączenie w danym polu, klikając na nie dowolnym przyciskiem myszy. Jeśli jeden z graczy zdoła połączyć jedną stronę planszy z drugą (dla gracza zielonego jest to lewa i prawa, dla czerwonego góra i dół) to ukazuje się tekst informujący o przegranej/wygranej gracza zielonego. Gracz komputerowy zawsze jest graczem czerwonym.

4 Podsumowanie

W moim projekcie zrealizowałem zadane mi polecenia. Użyłem do tego poznanych na zajęciach bibliotek i narzędzi. Na dodatek rozszerzyłem zakres funkcjonalności gry o zapis/ładowanie pliku zapisu gry, licznik czasu gry, typ gracza komputerowego „switchbot” oraz możliwość gry z innym graczem (bez gracza komputerowego).