

Dokumentacja projektu

Przedmiot: Podstawy Informatyki i Programowania

Nazwa Projektu: Gra w GO

Autorka: Patrycja Wysocka

Nr albumy: 306 022

1) Cel i opis projektu

Celem projektu, było stworzenie gry w GO dla planszy 19x19, która umożliwi rozgrywkę dwóm graczom.

W projekcie są zawarte następujące:

Pliki: Board.py, Group.py, Player.py, Game.py, BoardDisplay.py, global_var.py, test_board.py, test_player.py, test_group.py, test_game.py, goban-19x19.png

Biblioteki: numpy, pygame, math, argparse

2) Podział programu na klasy i opis klas

- Board – klasa, która trzyma planszę gry z wykorzystaniem biblioteki numpy. Jest ona odniesieniem dla wielu z pozostałych klas, ponieważ jest pewnego rodzaju bazą, trzymającą potrzebne informacje. Ma w sobie metody umożliwiające jej modyfikację oraz zliczające terytoria.
- Group – klasa, która przechowuje wszystkie grupy danego gracza/koloru. Zawiera w sobie listę przechowującą wszystkie grupy. Każda grupa to zbiór, w którym każdy kamień jest wyrażony poprzez krotkę z jego koordynatami.
- Player – klasa zawierająca wszystkie informacje o graczu, tj. imię, kolor, którym gra, ilość zdobytych niewolników, terytoria jakie posiada oraz zdobyte punkty. Ma w sobie metody umożliwiające zliczenie i ustawienie punktów, terytoriów oraz niewolników.
- Game – klasa łącząca wszystkie poprzednie. Ma metody sprawdzające możliwość ruchu, wykonujące jeden kompletny ruch gry, zliczające punkty, ustanawiające terytoria, zmieniające kolej graczy i zwracające zwycięzcę.
- BoardDisplay – klasa, która w oparciu o bibliotekę pygame wyświetla goban (planszę do gry w GO) o wymiarach 19x19 oraz umieszcza na niej kamienie, w zależności od tego jak wygląda sytuacja na planszy w klasie Board. Ta klasa czyta, w którym miejscu kliknął użytkownik i w najbliższym miejscu, gdzie przecinają się linie, umieszcza jego kamień. Przy każdym ruchu, plansza się na nowo aktualizuje – jest na nowo zaczytywana z planszy, którą trzyma Board (numpy array). Na koniec wyświetla informacje o zwycięzcy.

3) Instrukcja użytkownika

Celem gry jest zdobycie jak największej ilości kamieni przeciwnika oraz uzyskanie jak największego terytorium. Kamienie przeciwnika zdobywane są poprzez otaczanie jego grupy kamieni poprzez swoje kamienie. Grupa kamieni to kamienie sąsiadujące ze sobą po lewej/prawej stronie bądź góra/dół (nie po skos). Grupa jest otoczona, gdy nie przylega do niej żadne wolne pole.

Gra jest uruchamiana za pomocą argumentów wiersza poleceń. Jako pierwszy argument należy podać nazwę gracza, który gra czarnymi kamieniami, jako drugi gracza, którego kolorem jest biały. Dodatkowo opcjonalnie można wywołać flagę '-komi', która spowoduje dodanie punktów Komi białemu graczowi, zgodnie z zasadami GO. Przykładowe wywołanie programu wygląda następująco:

```
'python3 ./main.py Roch Walery -komi'
```

Następnie wyświetli się plansza, na której, aby postawić swój kamień, należy kliknąć w odpowiednie pole. Jeżeli gracz będzie chciał postawić kamień w miejscu, które od razu spowoduje samobójstwo (zabicie kamienia), gra mu na to nie pozwoli i będzie czekała, aż jego ruch będzie prawidłowy. Rozgrywka trwa, dopóki obaj gracze kolejno po sobie nie spasuują. Mogę to zrobić w dowolnym momencie poprzez kliknięcie spacji.

4) Część refleksyjna

Zanim zaczęłam tworzyć projekt gruntownie przemyślałam jakie chcę go podzielić, cały 'szkielet' rozpisałam sobie na kartce oraz dokładnie starałam się przeanalizować każdą możliwą sytuację, dzięki czemu podczas pisania programu nie napotkałam żadnych niespodziewanych trudności. Na pewno jednak ilość planowanych rzeczy w stosunku do wykonanych znacząco się zmieniła. Początkowo nie planowałam tworzenia jakiegokolwiek interfejsu graficznego do gry, tylko chciałam wyświetlać wszystko w terminalu. Niestety przy wymiarach 19x19 (typowych dla GO) widok planszy był naprawdę bardzo nieprzejrzysty. Decyzja o podjęciu próby 'zaprzyjaźnienia' się z pygmem była dość spontaniczna, początki toporne, ale satysfakcja z efektów bezcenna. Oczywiście 'apetyt rośnie w miarę jedzenia', więc pojawił się pomysł opcji ustawiania handicapów, czy stworzenia ekranu startowego, lecz niestety na te elementy zwyczajnie nie starczyło czasu. Niemniej jednak, sądzę, że program jest precyzyjny, przejrzysty, opatrzone odpowiednimi komentarzami, dokładnie przetestowany oraz przyjazny dla użytkownika.