

## Autor: Pola Witkowska

### Opis gry

Gra Plaster Miodu dla dwóch graczy (czerwony R i niebieski B) na grafie z 6 wierzchołkami. Gracze na zmianę łączą wierzchołki krawędziami. Przegrywa gracz, który utworzy trójkąt swoim kolorem.

### Schemat komunikacji

#### Komunikaty serwer → klient:

- YOU\_ARE {R|B} - informuje gracza o przydzielonym kolorze
- YOUR\_TURN - sygnał, że gracz może wykonać ruch
- WAIT - gracz czeka na ruch przeciwnika
- OK - ruch został zaakceptowany
- ILLEGAL - ruch niepoprawny, należy powtórzyć
- WIN - gracz wygrał (przeciwnik utworzył trójkąt)
- LOSE - gracz przegrał (utworzył trójkąt)

#### Komunikaty klient → serwer:

- MOVE {a} {b} - wykonanie ruchu łączącego wierzchołki a i b (gdzie a, b ∈ {0,1,2,3,4,5})

### Użytkowanie programu

#### Uruchomienie:

1. Uruchom serwer: python server.py
2. Uruchom pierwszego klienta: python client.py
3. Uruchom drugiego klienta: python client.py

#### Wykonywanie ruchów:

- Gdy pojawi się komunikat "Twoja tura", wpisz ruch w formacie: a b (np. 1 3)
- Wierzchołki numerowane są od 0 do 5
- Gracz wykonuje ruch, łącząc dwa różne wierzchołki krawędzią

#### Obsługiwane sytuacje błędne:

- **Niepoprawny format ruchu** - program poprosi o ponowne wprowadzenie w prawidłowym formacie
- **Połączenie tego samego wierzchołka ze sobą** - komunikat ILLEGAL, należy wykonać inny ruch
- **Krawędź już istnieje** - komunikat ILLEGAL, należy wybrać inne wierzchołki
- **Wierzchołki poza zakresem 0-5** - komunikat ILLEGAL
- **Ruch poza kolejką** - serwer automatycznie wysyła WAIT
- **Utworzenie trójkąta** - natychmiastowa przegrana, gra się kończy

#### Przykładowa rozgrywka:

Gracz R: 0 1 → OK

Gracz B: 2 3 → OK

Gracz R: 1 2 → OK

Gracz B: 3 4 → OK

Gracz R: 0 2 → LOSE (utworzył trójkąt 0-1-2)

### **Struktura programu**

- server.py - serwer gry obsługujący połączenia i logikę rozgrywki
- client.py - klient obsługujący interakcję z graczem
- game\_logic.py - logika gry (sprawdzanie poprawności ruchów, wykrywanie trójkątów)