

# Opis projektu gry Reversi

Konrad Woźniak 322945

Projekt używa biblioteki GTK, oraz do jego kompilacji potrzebny jest make. Program działa tylko na Linuxie. Aby rozpocząć grę należy uruchomić program dwa razy z dwoma różnymi parametrami:

- `./reversi A & ./reversi B &`

Po wywołaniu obu komend pojawią nam się dwa okienka jedno dla gracza A, który będzie miał kolor czarny, oraz drugie dla gracza B z kolorem białym. Na początku gracze będą widzieli standardowe ułożenie planszy 8 na 8, na której są dwa piony czarne i dwa białe, gracz czarny będzie mógł wykonać ruch, a gracz biały będzie czekać na wykonanie ruchu gracza czarnego i tak na zmianę. Każdy gracz może wykonać tylko taki ruch, który przejmie jakikolwiek pion przeciwnika. Piony przeciwnika zdobywa się, otaczając je w jednej linii. Otoczone piony zmieniają kolor i przynależność. Plansza będzie podświetlała pola, na których gracz może umieścić piona. W przypadku braku możliwego ruchu gracz traci kolejkę. Gra zakończy się w momencie gdy:

- Plansza będzie w pełni wypełniona,
- Żaden z graczy nie może wykonać ruchu,
- Jeden z graczy straci wszystkie swoje piony,

Wygrywa ten, który posiada więcej pionów swojego koloru na planszy, w przypadku równej ilości u obu graczy jest remis. W okienku programu liczba pionów poszczególnych kolorów będzie wyświetlana na bieżąco.

Stan planszy będzie przechowywany przy pomocy tablicy typu char posiadającej 64 elementy każdy indeks będzie odpowiadał za poszczególne pole i jego zawartość oznaczać będzie co na tym polu się znajduje. Na przykład indeks 0 to pole a8 indeks 1 to b8 indeks 2 to c8 itd. Dwa programy będą porozumiewać się ze sobą przy pomocy zmiennej typu int, która będzie oznaczała indeks, na którym przeciwnik umieścił ostatni pion. W razie braku możliwego ruchu zawartość zmiennej będzie wynosić -1 co oznaczać będzie, że nic na planszy się nie zmieniło. Po wykonaniu ruchu przez gracza jego plansza zaktualizuje się oraz zablokuje aż do momentu wykonania ruchu przez przeciwnika. Po otrzymaniu informacji przez przeciwnika jego plansza się zaktualizuje oraz będzie mógł wykonać ruch.

Program jest podzielony na 3 moduły:

- logic.h moduł obsługujący wyświetlanie planszy oraz wykonywanie ruchów
- connection.h moduł obsługujący przesyłanie i odbieranie danych pomiędzy dwoma programami
- fifo.h moduł obsługujące pliki kolejkowe