

Reversi - dokumentacja dla programisty

Jan Mazur

1. Moduł main

Moduł ten tworzy GUI i podpiną sygnały do widgetów.

Odpowiada także za cykliczne wywoływanie funkcji `get_signal()` z modułu "reversi".

2. Moduł pipes

Odpowiada za komunikację międzyprocesową. Przyporządkowuje odpowiednio każdemu z graczy jednostronne kanały (kolejki) komunikacyjne.

Najważniejsze funkcje:

- PipesPtr initPipes(int argc, char *argv[])
Funkcja tworzy kanały komunikacyjne i przyporządkowuje je odpowiednim graczom. Zwraca wskaźnik na strukturę przechowującą uchwyty do kanałów.
- void sendStringToPipe(PipesPtr pipes, const char *data)
Funkcja wysyła łańcuch znaków do przeciwnika.
- bool getStringFromPipe(PipesPtr pipes, char *buffer, size_t size)
Funkcja pobiera z kanału ciąg bitów (znaków) długości nie większej niż `size`. Zwraca TRUE gdy pobierze chociaż jeden znak. FALSE w p. p.

3. Moduł reversi

Zarządza całym przebiegiem gry i wszystkimi akcjami z nią związanymi, oraz czatem.

Ważne zmienne:

- int ruch_gracza
Kontroluje który gracz obecnie ma ruch. Gdy dzieli się przez 2 to biały a gdy nie to czarny. Po gracza jest inkrementowana.
- int numer_tury
- char id
Przechowuje id gracza. 'B' lub 'C'.
- int flag_new_game
Kontroluje czy można nacisnąć przycisk nowej gry.
- int flag_cofnij
Kontroluje czy można nacisnąć przycisk cofnij.
- int przeciwnik_cofnal
1 - jeśli przeciwnik w dopiero co cofnął ruch. 0 - w p. p.
- State last_state[9][9]
Tablica przechowująca stan planszy w poprzednim ruchu.

- `typedef struct button {
 GtkWidget *btn;
 State stan;
 char id[3];
 GtkImage *obrazek;
} Button;`

Struktura która udaje obiekt, czyli jeden przycisk na planszy.

Zawiera wskaźnik na obecny obrazek, id pola, stan i wskaźnik na EventBox.

Najważniejsze funkcje:

- `void new_game_process(GtkWidget *widget, gpointer data)`
Zdarzenie przycisku nowej gry. Wysyła zapytanie do przeciwnika o nową grę.
- `void new_game(GtkWidget *widget, gpointer data)`
Funkcja uruchamia nową grę. Przygotowuje plansze etc.
- `void open_dialog_start()`
Funkcja otwiera okno dialogowe w którym gracz może się zgodzić na rozpoczęcie nowej gry lub nie.
- `void on_cofnij_clicked(GtkWidget *widget, gpointer data)`
Zdarzenie przycisku cofnij. Wysyła do przeciwnika zapytanie o możliwość cofnięcia ruchu.
- `void open_dialog_cofnij()`
Funkcja otwiera okno dialogowe w którym gracz może się zgodzić na cofnięcie ruchu przeciwnika bądź nie.
- `void open_dialog_wait()`
Funkcja wyświetla okno oczekiwania na odpowiedź przeciwnika.
- `void open_dialog_exit()`
Funkcja wyświetla okno dialogowe gdy przeciwnik wyjdzie z gry.
Wtedy gracz też wychodzi z gry.
- `void open_dialog_finish()`
Funkcja wyświetla okno dialogowe pod koniec gry w którym informuje kto wygrał i daje możliwość rozpoczęcia nowej gry lub zrezygnowania z owej.
- `void pobierz_tekst(char *data)`
Funkcja pobierająca tekst z czatu od przeciwnika i wyświetla go w oknie czatu.
- `void przekaz_tekst(GtkWidget *widget, GtkWidget *text)`
Funkcja przekazująca tekst z okna czatu do czatu przeciwnika.
- `int get_signal()`
Funkcja pobiera sygnał od przeciwnika i na jego podstawie wykonuje akcje.

Sygnały:

- "!XX" gdzie XX to współrzędne pola planszy. Przeciwnik wykonał ruch XX.
- ".NM" przeciwnik nie mógł wykonać poprawnego ruchu.
- "@EX" przeciwnik wyszedł z gry.
- "\$GO" sygnał końca gry.
- "#SP" pozytywna odpowiedź do nowej gry.
- "&SN" negatywna odpowiedź do nowej gry.
- "%SR" pytanie o nową grę.

- "^CR" pytanie o cofnięcie.
- "(CP" pozytywna odpowiedź do cofnięcia.
- ")CN" negatywna odpowiedź do cofnięcia.
- "*" sygnał do wysyłania wiadomości na czacie.

- void send_signal(char *data)
Funkcja wysyła sygnał do przeciwnika.
- void update_history(char *data)
Funkcja aktualizuje okno historii rozgrywki o odpowiedni ruch.
- void dump_cofnij()
Funkcja zapisuje stan planszy do tablicy last_state.
- void on_field_clicked(GtkWidget *btn_clicked, GdkEventButton *event, char *data)
Zdarzenie przycisku planszy. Wysyła sygnał ruchu do przeciwnika.
Aktualizuje historię rozgrywki. Zapisuje stan planszy do cofania.
Zmienia odpowiednio do ruchu. Inkrementuje ruch_gracza.
Blokuję i odblokowuję plansze.
- void cofnij()
Cofa ruch gracza. Aktualizuje historię rozgrywki, plansze etc.
- int blokuj()
Dezaktywuje pola których kliknięcie jest niezgodne z zasadami gry.
- void odblokuj()
Aktywuje wszystkie pola.
- int zmien_plansze(char *id, int ruch_gracza)
Zmienia plansze w zależności od tego który gracz miał ruch i jaki ruch wykonał.
- int czy_zmiana(char *id)
Funkcja sprawdza czy stawiając pionka na danym polu zmieni się plansza. Czyli czy ruch w danym miejscu jest zgodny z zasadami gry.

4. Funkcjonalności aplikacji

- Po kliknięciu przycisku "Nowa gra" przeciwnik dostaje zapytanie czy chce rozpocząć nową grę. Może zaakceptować albo odrzucić propozycję. W tym czasie gracz który wnioskował o nową grę czeka na odpowiedź przeciwnika. W przypadku odpowiedzi pozytywnej gra rozpoczyna się.
- Po kliknięciu przycisku "Cofnij" przeciwnik dostaje zapytanie czy pozwala swojemu przeciwnikowi na cofnięcie jednego ruchu. W tym czasie gracz który wnioskował o cofnięcie ruchu czeka na odpowiedź przeciwnika. W przypadku odpowiedzi pozytywnej ruch jest cofany.
Można wnioskować tylko o cofnięcie jednego swojego ruchu.
- Gracz nie może nacisnąć żadnego pola dla którego jego ruch byłby niepoprawny.
- Pola oznaczone małym szarym kółkiem to pola na które gracz może nacisnąć i wykonać poprawny ruch.
- Jeśli poprawny ruch na gracza nie istnieje to automatycznie przeciwnik ma 2 ruch.
- Jeśli gracz wyjdzie z gry to jego przeciwnik otrzymują taką informację i może wyjść z gry.

- Wszystkie niedozwolone akcje są na bieżąco blokowane.
- Gracze podczas rozgrywki mogą ze sobą konwersować.
- Po zakończonej rozgrywce obu graczom wyświetla się informacja który z nich wygrał. Mają możliwość albo od razu wnioskować o następną nową grę albo zrezygnować z tej przyjemności i wejść do okna z zakończoną grą i znowu ewentualnie zacząć nową grę lub wyjść.