

Reversi – dokumentacja projektu

Jakub Froń 2019 WdPC

1. Opis interakcji:

Program umożliwia rozgrywkę dwuosobową na jednym komputerze. Uruchomienie polega na otwarciu dwóch programów z odpowiednimi danymi wejściowymi, to znaczy "A" lub "B", które oznaczają odpowiednio pierwszego oraz drugiego gracza. Pierwszym oknem, które ujrzą użytkownicy jest Menu gry, składa się ono z przycisków: Nowa Gra, Wczytaj Gre, Zasady oraz Wyjdź z gry.

2. Moduły:

Main:

Odpowiada za inicjalizację niektórych przycisków i ułożenie widgetów.

W grze występują dwa różniące się okna, po uruchomieniu rozgrywki tworzony jest `gtk_box` menu, który zawiera przyciski:

Nowa Gra – daje opcje wyboru wielkości mapy, po czym wysyła do drugiego gracza zapytanie o rozpoczęcie nowej rozgrywki, w zapytaniu znajduje się informacja o wybranej wielkości mapy.

//wywołuje funkcję `rozp_nowa_gre_menu()`;

Wczytaj Gre – przycisk ma dwa stany, jeśli nie istnieje plik zapisu, bądź istnieje ale jest pusty przycisk będzie zablokowany (nie da się go wcisnąć), w p.p wysyłane zostaje zapytanie do drugiego gracza o wczytanie rozgrywki z zapisu. //wywołuje funkcję `wczytajGre()`;

Zasady – widget typu przycisk z odnośnikiem do strony internetowej, po naciśnięciu odsyła nas do strony dokładnie tłumaczącej zasady rozgrywki.

Wyjdź z gry - wysyła do drugiego gracza informacje o zakończeniu rozgrywki // wywołuje funkcję `quit()`;

Po rozpoczęciu rozgrywki widok zmieniany jest na główny widok rozgrywki, ujawniany jest box główne, który składa się z trzech mniejszych boxów:

Lewe – box zawiera grid, na którym umieszczona jest historia

Srodek – widgety powiązane z planszą, grid na przyciski, label na numerację pól

Prawe – czat i powiązane z nim label'e

Komunikator (-fifo.):

Zadaniem tego modułu jest stworzenie i utrzymanie kanałów typu kolejka fifo pomiędzy graczami, składać się będzie z funkcji:

PipesPtr initPipes(int argc, char *argv[]) - funkcja tworzy kanały pomiędzy graczami i przypisuje je do nich (AtoB, BtoA). Zwraca wskaźnik na strukturę przechowującą "uchwyty" kanałów.

void sendStringToPipe(PipesPtr pipes, const char *data) - Wysyła łańcuch znaków do drugiego gracza.

bool getStringFromPipe(PipesPtr pipes, char *buffer, size_t size) - odbiera łańcuch znaków, zwraca TRUE jeśli pobierze przynajmniej jeden znak, FALSE w p.p.

Reversi:

Moduł będzie zarządzać całym przebiegiem rozgrywki, zdarzeniami oraz czatem. Moduł będzie między innymi: kontrolować czyja jest tura, przechowywać stan planszy, przechowywać numer tury.

Ważne zmienne:

char id; - informacja o id (A, B) gracza

int czyjruch; - informacja o ruchu przeciwnika (czyjruch%2==0 - biały, czyjruch%2==1 - czarny)

int numer_tury; - licznik tur

int wielkosc_planszy;

typedef struct button {

```

GtkWidget *btn;

int rodzaj;      //0-bialy 1-czarny 2-nic

char numer[4];   //numer pola

GtkImage *obrazek;

int isblocked;

} Buttons;

```

Struktura zawierająca wskaźnik na przycisk na planszy, informacje o rodzaju, numerze na planszy, przechowująca obrazek i stan (czy jest zablokowany).

Ważne funkcje:

```
int aktualizuj_plansze(char *numer, int czyjruch);
```

Po kliknięciu na odblokowany przycisk, który został ustalony jako poprawny przez funkcję podswietl(); aktualizowana jest plansza zgodnie z zasadami gry Reversi.

```
int podswietl();
```

Po wykonaniu ruchu i na początku rozgrywki uruchamiana jest ta funkcja w celu odnalezienia pól na które dany gracz może wykonać ruch, następnie te pola są podświetlane.

```
void clicked(GtkWidget *btn_clicked,GdkEventButton
*event,char *data);
```

Funkcja uruchamiana po naciśnięciu podświetlanego przycisku uruchamia szereg funkcji: aktualizuj_plansze, aktualizuj_historie, podswietl.

```
void wyslij_sygnal(char *data);
```

Funkcja wysyłająca wcześniej podany jako argument ciąg znaków, który jest wykorzystywany do komunikacji pomiędzy programami.

```
int odbierz_sygnal();
```

Funkcja odbierająca z kolejki fifo ciągi znaków, zależnie od pierwszych znaków uruchamiane są inne, kolejne funkcje.

```
void przekaz_tekst( GtkWidget *widget, GtkWidget *text);
```

```
void pobierz_tekst(char *data);
```

Funkcje odpowiedzialne za komunikacje pomiędzy graczami za pomocą chat'u.

```
void rozp_nowa_gre(GtkWidget *widget, gpointer data);
```

Funkcja uruchamiana po naciśnięciu przycisku Nowa Gra, podobnie jak funkcja rozp_nowa_gre_menu(); funkcja wysyła sygnał do drugiego gracza z zapytaniem o nową rozgrywkę.

```
void nowa_gra();
```

Funkcja ustawia początkowe rozstawienie przycisków planszy, podłącza sygnały i uruchamia funkcje podswietl();

Grupa funkcji ze słowem "dialog" w nazwie odpowiada za tworzenie interaktywnych okien, dialog_wczytaj wyświetla zapytanie o wczytanie rozgrywki, dialog_koniec wyświetla informacje o opuszczeniu rozgrywki przez przeciwnika.