Projekt C: ‘Gra Quoridor’

Wojciech Fica, indeks: 280180

## Dokumentacja dla programisty

### Opis modółów

* *main.c:* tworzona jest tam plansza oraz funkcje do obsługi zdarzeń korzystające z GTK.
* *plansza\_nagl.h:* nagłówki struktur i definicje stałych
* *kolejka\_def.c:* implementacja kolejki z listy 6
* *kolejka\_nagl.h:* nagłówki kolejki z listy 6
* *win\_fifio\_projekt.c* oraz *fifo\_projekt.h:* z wykładu do komunikowania się między procesami

### Kompilacja i uruchamianie:

Należy utworzyć projekt ze wszystkimi modułami opisanymi powyżej. Opcje kompilatora:

* -std=c99
* `pkg-config gtk+-3.0 --cflags`
* `pkg-config gtk+-3.0 --libs`
* -Wall

Należy uruchomić program z argumentem A – gracz zielony, następnie z argumentem B – gracz czerwony.

### OPIS wybranych funckji

* bool IsReachable(int nr) – sprawdza czy pionek stojący na polu nr może wygrać przy danym ustawieniu planszy. Działa na zasadzie BFSa. Wykorzystywana do sprawdzania, czy ruch gracza jest dozwolony.
* void middle\_press( GtkWidget \*ebox, GdkEvent \*event, gpointer data) – funkcja obsługująca wstawienie ściany.
* void sqrt\_press ( GtkWidget \*ebox, GdkEvent \*event, gpointer data) – funkcja obsługująca ruchy pionkiem.
* void update\_sqrt (int nr\_pola, enum ORIEN orientacja) – funkcja uaktualniająca planszę po ruchu przeciwnika
* void update\_midd (int nr\_pola, enum ORIEN orientacja) – funkcja uaktualniająca planszę po ruchu przeciwnika

### OPIS wybranych struktur

* struct pole – pole planszy, na którym może być ściana albo pionek;
* struct to\_show\_walls – wykorzystywana do pokazania i pamiętania ile gracz ma jeszcze ścian;
* struct to\_show\_move – wykorzystywana do pokazania i pamiętania kogo jest ruch;
* struct to\_show\_option – umożliwia ustawienie orientacji ściany.