

Dokumentacja Projektu "Quoridor"

Mateusz Hazy

6 lutego 2017

Wstęp

Projekt "Quoridor" to prosta implementacja popularnej gry planszowej *quoridor*. Składa się z kilku modułów napisanych w C przy użyciu biblioteki GTK+ 3, folderów z grafiką oraz plików tekstowych.

Moduły

1. `stale_zmienne.h`

Moduł zawierający wszystkie zmienne globalne, struktury oraz tablice

- **SIZE_OF_BOARD** - rozmiar komórek planszy
- **SIZE_OF_GRID** - rozmiar planszy
- **INIT_WALLS** - liczba początkowych ścian
- **pole** - struktura łącząca dany button z jego współrzędnymi na planszy
- **gracz** - struktura przechowująca wszystkie informacje o danym graczu: nazwę, liczbę ścian, pozycję, przypisane do niego labelle, adres obrazka, stały numer oraz pozycje wygrywające
- **gracz A, B** - A to gracz wykonujący ruch, B- oczekujący.
- **grid_sizes, WALLS**- tablice z odpowiednimi wielkościami planszy i liczbą ścian dla danego trybu rozgrywki
- **plansza, sciany, used** - tablice z informacjami o stanie rozgrywki
- **polecenia_rozgrywki, scianyA, scianyB**- wskaźniki na labelle wyświetlane podczas rozgrywki
- **okienko_aktywne** - wskaźnik na aktualnie otwarte okienko

2. `rozgrywka.h`

Moduł z funkcjami dotyczącymi mechanizmu rozgrywki

- **init_players** - funkcja ustalająca początkowe pozycje i liczbę ścian graczy

- **umiesc_gracza** - ustawia gracza na danej pozycji
- **przesun_gracza** - przesuwa gracza na wybraną pozycję
- **miedzy_sciana** - czy istnieje ściana między danymi polami
- **pola_sasiadnie** - czy dane pola są sąsiadnie
- **clear_sciany, clear_used** - funkcje czyszczące tablice ścian oraz used
- **zwyciestwo** - czy gracz odniósł zwycięstwo
- **dfs, czy_blokuje** - funkcje sprawdzające, czy ewentualne postawienie ścian nie uniemożliwi zwycięstwa przeciwnikowi, bądź samemu sobie
- **dozwolony_(ruch, sciana pionowa, sciana pozioma)** - funkcje sprawdzające, czy wybrana przez gracza akcja jest dozwolona
- **(postav, podswietl, zgas)_sciane_(pionowa, pozioma)** - wykonują akcje jak w nazwie
- **zmiana_ruchu** - zamienia wskaźniki gracza wykonującego ruch z graczem oczekującym
- **ruch_(sciana pionowa, sciana pozioma, pole)** - funkcje ruchu wywoływane po kliknięciu na odpowiednie pola planszy przez gracza
- **(enter, leave)_sciana_(pionowa, pozioma)** - funkcje wywoływane po najechaniu na obszar ścian
- **new_button_(sciana pionowa, pozioma, pole)** - funkcje zwracające widget typu jak w nazwie

3. **app_manager.h**

Moduł z funkcjami związanymi z tworzeniem interfejsu i nawigacją wewnątrz aplikacji

- **generuj_(okienko settings, okienko instructions, okienko gry, strone glowna)** - funkcje tworzące odpowiednie okienka aplikacji
- **generuj_(pole , interfejs gry)** - funkcje pomocnicze do stworzenia okienka gry
- Funkcję wykonujące działania jak w nazwie: **powrot_do_glownej, restart_game, start_game, wczytaj_gre, zapisz_gre, go_to_settings, go_to_instructions**

Opis Działania

Rozgrywka polega na naprzemiennym wykonywaniu ruchów graczy. Każdy z nich polega na kliknięciu na odpowiednie pole (ściana lub zwykłe pole). Jeśli dany ruch jest dozwolony, nastąpi zmiana ruchu i wskaźników graczy. W momencie zwycięstwa, ruchy zostają uniemożliwione.

Użytkownik może zmienić nazwy graczy, wielkość planszy ($5x5$, $9x9$, $11x11$), zapisać oraz wczytać grę.

Do zapisu i wczytywania gry używany jest plik *zapis_gry.txt* przechowujący niezbędne informacje o ostatnim zapisanym stanie gry.

W pliku *instrukcje.txt* przechowywana jest treść wyświetlana w okienku *instructions*.

Foldery *images* oraz *css* zawierają pliki związane z grafiką