

Dokumentacja dla programisty

Robert Pelka

29 stycznia 2018

Dwuosobowa gra „Pentago” zaimplementowana z użyciem interfejsu GTK+

Program napisany w zgodności z GTK+ w wersji 3.18

Na program składają się cztery moduły:

1. `pentago.c` - zawiera funkcje modyfikujące wygląd i działanie przycisków
2. `gamestatus.c` - zawiera funkcje sprawdzające i modyfikujące stan gry
3. `linfifo.c` - zawiera funkcje obsługujące pliki kolejkowe
4. `main.c` - zawiera głównie szkielet interfejsu graficznego programu

1 `pentago.c`:

- `refresh()` - Wywołuje funkcje `get_text()` i `set_colors()`, aby pobrać najnowsze ustawienie przycisków, a następnie zwraca TRUE.
- `lock_buttons()` - Blokuje przyciski planszy i odblokowuje przyciski obracania.
- `lock_rotatebuttons()` - Blokuje przyciski obracania.
- `unlock_buttons()` - Odblokowuje przyciski planszy i blokuje przyciski obracania.
- `alert()` - Wyświetla okienko z zadany jako argument komunikatem.
- `CSS()` - Inicjuje obsługę kaskadowych arkuszy stylów.
- `clicked_button()` - Funkcja wywoływana kliknięciem jednego z 36 przycisków planszy. Zmienia ona wartość odpowiedniej komórki w tablicy `matrix` na '1' (kolor czarny) lub '2' (kolor biały), następnie wywołuje funkcję `push_text()`, by wysłać zmiany i `lock_buttons()`, by zablokować możliwość kolejnego ruchu.
- `set_colors()` - Na podstawie tablicy `matrix` ustawia kolory przycisków i wywołuje funkcję `is_win()`.

- `new_game()` - Rozpoczyna nową grę, tj. wypełnia tablicę `matrix` zerami i wywołuje funkcje `set_colors()`, `lock_buttons()` oraz `lock_rotatebuttons()`, by wyczyścić przyciski z kolorów, a następnie je zablokować.
- `rotate_right()` - Na podstawie współrzędnych zadanych jako argument funkcji, obraca jedną z czterech części planszy o 90 stopni zgodnie z ruchem wskazówek zegara. Następnie wywołuje funkcję `lock_rotatebuttons()`, by zablokować przyciski służące do obracania i funkcję `push_text()`, by wysłać zmiany.
- `rotate_left()` - Na podstawie współrzędnych zadanych jako argument funkcji, obraca jedną z czterech części planszy o 90 stopni przeciwnie do ruchu wskazówek zegara. Następnie wywołuje funkcję `lock_rotatebuttons()`, by zablokować przyciski służące do obracania i funkcję `push_text()`, by wysłać zmiany.
- `show_error()` - Pokazuje okienko z błędem, którego treść jest przekazana jako argument tej funkcji.

2 gamestatus.c:

- `start()` - Funkcja wywoływana przez kliknięcie przycisku *"I'll start"*, ustawia wartość zmiennej `player` na `'2'`, co powoduje, że gracz który kliknął przycisk, gra kolorem białym. Następnie wywołuje funkcje `unlock_buttons()`, by odblokować przyciski planszy i `push_text()`, aby przesłać zmiany.
- `give_up()` - Funkcja wywoływana przez kliknięcie przycisku *"I give up"*, ustawia wartość zmiennej `giveup` na `'1'`, następnie wywołuje funkcję `push_text()` i ustawia wartość zmiennej `giveup` na `'0'`.
- `maybe_restart()` - Wyświetla komunikat z propozycją restartu gry, po kliknięciu przycisku *"yes"* ustawia wartość zmiennej `giveup` na `'2'`, wywołuje funkcję `push_text()`, ustawia wartość zmiennej `giveup` na `'0'` i wywołuje funkcję `new_game()`.
- `vertical()` - Zwraca ilość sąsiadujących w pionie przycisków jednakowego koloru.
- `horizontal()` - Zwraca ilość sąsiadujących w poziomie przycisków jednakowego koloru.
- `diagonal()` - Zwraca większą z dwóch sum sąsiadujących po jednym z ukosów przycisków jednakowego koloru.
- `is_draw()` - Zwraca `0` w przypadku remisu i `1` w przeciwnym wypadku.
- `is_win()` - Sprawdza dla każdej komórki w tablicy `matrix`, czy któraś z funkcji: `vertical()`, `horizontal()` i `diagonal()` zwraca wartość większą lub równą pięć, w takim wypadku wywołuje funkcję `alert()` z odpowiednim komunikatem.

3 linfifo.c

- `push_text()` - Ustawia wartość zmiennej `unlock` na `0`, następnie tworzy tablicę `tekst[38]` typu `char`, której zerowa komórka przechowuje wartość zmiennej `player` dla drugiego okna, 37. komórka przechowuje wartość zmiennej `giveup`, a pozostałe komórki przechowują wartości poszczególnych komórek tablicy `matrix`, by w końcu za pomocą funkcji `sendStringToPipe()` przesłać wartości tablicy `text`.
- `get_text()` - Wywołuje funkcję `getStringFromPipe()`, jeżeli pobierze ona tekst do tablicy `tmp`, to wartość zmiennej `player` jest ustawiana na wartość pierwszej komórki tablicy `tmp`. Z komórek 1-36 tej tablicy następuje przepisanie wartości do tablicy `matrix`. Jeżeli wartość 37. komórki tablicy `tmp` wynosi `1`, to znaczy, że przeciwnik poprosił o restart gry i wywoływana jest funkcja `maybe_restart()`, jeżeli natomiast wartość wynosi `2`, to oznacza, że drugi gracz zgodził się na restart gry i wywoływana jest funkcja `new_game()`. Jeżeli wartość zmiennej `unlock` jest większa od `1`, to wywoływana jest funkcja `unlock_buttons()`, by umożliwić wykonanie ruchu.
- `quit()` - Zamyka program wywołując funkcje `closePipes()` i `gtk.main_quit()`.
- `closePipes()` - Zamyka otwarty funkcją `fopen()` plik oraz zwalnia obszar pamięci wskazywany przez `pipes`.
- `initPipes()` - Inicjuje potoki, za pomocą których przekazywane są informacje pomiędzy kopiami programu.
- `openOutPipe()` - Otwiera plik do zapisu.
- `openInPipe()` - Otwiera plik do odczytu.
- `sendStringToPipe()` - Przesyła do potoku napis, przekazany jako argument funkcji.
- `sendStringToPipe()` - Pobiera napis z potoku.