

xWord

Program ułatwiający tworzenie krzyżówek typu "Jolka" Dokumentacja techniczna

1. Wstęp

Przed rozpoczęciem czytania dokumentacji warto zajrzeć do instrukcji dla użytkownika. xWord składa się z:

-6 plików źródłowych ".c":

crosswords.c, main.c, main_gui.c, promptWindow.c, window_new.c, wordList.c

-6 plików nagłówkowych ".h":

crosswords.h, main_gui.h, promptWindow.h, window_new.h, wordList.h, config.h

-1 arkusza stylów ".css"

mainStyle.css

-plików słownika w folderze *dictionaries*

-obrazków oraz ikon w folderze *graphics*

W plikach nagłówkowych znajdują się deklaracje struktur i funkcji używanych w plikach źródłowych o tej samej nazwie. Każda para plików odpowiada modułowi programu. Moduły obsługujące okna zawierają deklaracje struktury danych zawierającej wszystkie elementy okna. Wskaźnik do tej struktury jest przekazywany wszystkim funkcjom działającym na danym oknie.

2. Krótki opis modułów i plików

-*crosswords* - moduł zawiera obsługę krzyżówki - strukturę planszy, zapisywanie/wczytywanie planszy do pliku oraz dopasowywanie słów pasujących do danego schematu.

-*main.c* - teoretycznie główny plik programu, w praktyce zawiera jednak tylko inicjalizację biblioteki GTK oraz wywołanie funkcji *run* z modułu *main_gui* odpowiedzialnej za obsługę głównego okna. Moduł ten zawiera również główną pętlę programu - *gtk_main()*

-*main_gui* - moduł obsługujący główne okno programu. Zawiera funkcje tworzące i obsługujące pasek narzędziowy oraz graficzny obraz planszy. W tym module otwierane są wszystkie okna dialogowe i znajduje się obsługa głównych zdarzeń programu. Struktura odpowiedzialna za przechowywanie głównego okna programu ma nazwę: *mainGui*.

-*promptWindow* - moduł obsługujący okno z podpowiedzią. Struktura zarządzająca tym oknem to *PromptWindowGui*.

-*window_new* - moduł odpowiedzialny za okno tworzenia nowej krzyżówki. Struktura przechowująca okno nazywa się *windowNewGui*.

-*wordList* - moduł zawierający strukturę danych *WordArray* - dynamiczną tablicę, w której przechowywane są wyrazy.

-*config.h* - plik zawierający informacje konfiguracyjne(głównie dyrektywy *#define*) takie jak: nazwy i rozmiary okien, maksymalny rozmiar używanych wielkości itp.

-*mainStyle.css* - arkusz stylów zawierający ustawienia czcionek i kolorów tła. *GtkLabel#texts* to nazwa nadana etykietom-komórkom krzyżówki, a *GtkLabel#textsMatchedLetter* to nazwa pól zawierających zielone litery w oknie podpowiedzi.

3.Struktura interfejsu graficznego głównego okna

Główne okno programu składa się z pojemnika *box* w którym umieszczone są: przycisk przechwytyjący focus - *targetFocus*, pasek narzędzi *toolBar* (zawierający przyciski typów *GtkToolButton* oraz *GtkToggleToolButton*) oraz obszar roboczy *layout* na którym rysowana będzie krzyżówka. Przycisk przechwytyjący focus był konieczny, aby zniwelować nieestetyczne zaznaczenie przycisków paska narzędziowego.

Krzyżówka zbudowana jest z siatki *grid*, w której komórkach umieszczone są widżety *GtkEventBox* oraz *GtkLabel*. Służą one odpowiednio do odbierania zdarzeń kliknięcia na daną komórkę oraz wyświetlania litery. W *eventBoxie* znajduje się obrazek przedstawiający tło komórki. Dodatkowo każda komórka siatki zawiera 8 obrazków(4 narożniki i 4 krawędzie) stanowiących zieloną ramkę zaznaczenia wyrazu - domyślnie obrazki te nie są wyświetlane.

Klikanie na pola krzyżówki powoduje wywoływanie funkcji *cellClickedEvent*, która w zależności od aktywnych przycisków paska narzędzi wykonuje odpowiednie działanie. Wcisnięcie klawisza wywołuje funkcję *keyPressedEvent*, która w zależności od rodzaju wciśniętego klawisza wykonuje odpowiednią akcję.

Informacja ogólna: funkcje odpowiedzialne za obsługę zdarzeń mają w swojej nazwie słowo *Event*, a funkcje łączące sygnały zawierają w nazwie część *connect*.

4.Struktura *WordArray* (*wordList.c*, *wordList.h*)

Struktura *WordArray* to dynamiczna tablica przechowywująca wyrazy. Składa się ona z dwuwymiarowej tablicy typu *char* zaalokowanej na stercie reprezentowanej przez wskaźnik *char*** oraz aktualnego rozmiaru i aktualnego maksymalnego rozmiaru. Dodawanie elementów do struktury odbywa się za pomocą funkcji *pushWord*. Działa ona następująco - jeśli aktualny rozmiar tablicy jest mniejszy od aktualnego maksymalnego rozmiaru, to nowy element dopisywany jest na końcu tablicy. W przeciwnym wypadku, tablica zostaje realokowana, a jej maksymalny rozmiar zwiększa się dwukrotnie i element dodawany jest na jej pierwszym wolnym miejscu.

5. Przykład: tworzenie głównego okna programu (*main_gui.c, main_gui.h*)

Przy tworzeniu każdego okna wywoływana jest najpierw funkcja *create* np. *createMainWindow* (w przypadku głównego okna programu). W tej funkcji tworzone jest okno, ustawiane są jego wymiary oraz określone inne ważne parametry (takie jak np. tytuł, modalność, typ okna itp.). Następnie tworzony jest główny pojemnik zawierający pozostałe widżety i wywoływana jest funkcja tworząca pasek narzędzi - *createToolBar*. Po tym wszystkim dodany zostaje obszar roboczy *layout* oraz podłączamy obsługę zdarzeń za pomocą funkcji *connectSignals*. Część tych funkcji będzie w przyszłości odłączana od swoich sygnałów, zapamiętujemy więc konieczne numery *id* zdarzeń.

Ciekawą funkcją jest funkcja:

```
setMenuButtonsToggledWithout(mainGui *gui, GtkWidget *widget)
```

której parametrami są: struktura głównego menu, oraz przycisk paska narzędziowego *widget*. Funkcja ta służy do dezaktywowania wszystkich poza jednym (podanym jako argument) *GtkToggledToolButton* w pasku narzędziowym. Jest ona wywoływana automatycznie kiedy zmienia się stan któregoś z przycisków i jej celem jest utrzymywanie tylko jednego aktywowanego przycisku an raz. Co ciekawe, aby program się nie zapętlał, na początku działania tej funkcji musimy odłączyć ją od sygnałów, które obsługuje.

W programie xWord użyto również funkcji *timeout* z biblioteki *glib*, aby wywoływać konkretne zdarzenie co pewien interwał czasowy - jest ona wykorzystywana przy mrugającym kursorze zaznaczonego pola. (domyślny odstęp między mrugnięciami wynosi 300ms).

6. Struktura pliku .xw

Plik .xw ma następującą strukturę - pierwsza linijka zawiera nazwę krzyżówki, w drugiej linii znajdują się dwie liczby całkowite - szerokość i wysokość planszy krzyżówki, a w następnych liniach opisana jest plansza. Pola planszy opisywane są w taki sposób:

- jeśli w danym miejscu planszy znajduje się czarny kwadrat, to opisujemy to znakiem '0'

- gdy w danym miejscu znajduje się puste pole bez tekstu, to opisujemy to znakiem '1'

- gdy w danym miejscu znajduje się litera niebędąca polskim znakiem, to opisujemy pole tą literą

- gdy w danym miejscu znajduje się polski znak, to opisujemy to pole znakiem o kodzie z przedziału 0,..., 9 - 'A' odpowiada 0, 'Ć' odpowiada 1 itd. (alfabetycznie)

7.Wyszukiwanie wyrazów (*crosswords.c, crosswords.h*)

Słownik znajduje się w katalogu *dictionaries*. Jest on podzielony na 20 plików, w każdym z plików znajdują się wyrazy o długości zadanej przez nazwę pliku. Funkcją odpowiedzialną za dopasowywanie wyrazów ze słownika jest:

*WordArray *wordMatch(char *pattern)*

Funkcja ta zwraca tablicę *WordArray*, a jej argumentem jest napis stanowiący wzór wyrazów do dopasowywania. Funkcja dopasowuje wyrazy ze słownika do wzoru w następujący sposób - wybiera wyrazy o długości wzoru, które mają takie same litery na takich samych pozycjach co wzór. Wyjątkiem jest znak '.', który oznacza, że na danym miejscu może znajdować się dowolna litera. Przykład: wyraz "mama" pasuje do wzoru "m..a".

8.Zakończenie

Do dokumentacji nie dołączam szczegółowej listy funkcji oraz opisów funkcji - w plikach nagłówkowych znajdują się wszystkie potrzebne definicje, a nazwy funkcji oraz ich argumentów powinny mówić same za siebie. (starałem się, żeby tak było, jednak możliwe, że w niektórych miejscach mi się to nie udało - wtedy przejrzenie definicji funkcji powinno rozwiązać wszelkie wątpliwości).

Piotr Maślankowski (pmaslankowski@gmail.com)