

Podstawy Grafiki Komputerowej

Projekt nr 1 - FOTO INDEKS.

1. Tytuł projektu i autorzy projektu.

Zadaniem jakie należało wykonać w ramach projektu pt. Foto Indeks było napisanie programu do tworzenia indeksów zdjęć w postaci miniatur. Projekt został zrealizowany przez trzyosobowy zespół:

- Hubert Jarosz
- Kamil Gwiżdż
- Maciej Czech

2. Opis projektu.

Głównym celem projektu jest stworzenie programu, który tworzy indeksy zdjęć. Program po uruchomieniu prosi o wskazanie ścieżki do katalogu, z którego chcemy zaimportować i zindeksować zdjęcia oraz ścieżkę do katalogu docelowego, w którym program zapisze wynikową bitmapę składającą się z 20 miniatur wraz z podpisami oraz dodatkowy komentarz generowany w dolnej części bitmapy. Treść komentarza jest wypełniana przez użytkownika przed każdorazowym stworzeniem kolejnego indeksu. Każde zdjęcie jest podpisane nazwą pliku z którego pochodzi, jeśli nazwa jest zbyt długa to zostaje skrócona i poprzedzona znakiem tyldy (~).

3. Założenia wstępne.

- Program został wyposażony w prosty interfejs graficzny.
- Użytkownik wczytuje ścieżkę do katalogu ze zdjęciami, ścieżkę docelową oraz komentarze widniejące na każdym indeksie
- Program wczytuje pliki w formacie BMP i JPG
- Wynikowe indeksy są tworzone w postaci bitmap o rozmiarach 800x600px
- Każdy indeks zawiera 20 miniatur (posortowanych alfabetycznie) - cztery wiersze po pięć miniatur.

4. Analiza projektu.

a. Specyfikacja danych wejściowych

Dane wejściowe programu to dwie wyżej opisane ścieżki ścieżka źródłowa powinna zawierać zestaw zdjęć w formacie BMP lub JPG.

b. Opis oczekiwanych danych wyjściowych

Dane wyjściowe to plik/pliki w formacie BMP zawierające 20 miniatur z podpisami oraz komentarz.

c. Zdefiniowanie struktur danych

Program nie korzysta ze skomplikowanych czy niestandardowych struktur danych. W dużej mierze operuje na podstawowych typach danych języka c++ oraz obiektach z biblioteki WxWidgets takich jak np.: wxBitmap, wxImage, wxFileName czy wxString.

d. Specyfikacja interfejsu użytkownika

Interface graficzny jest minimalistyczny i prosty, stworzenie go takim było celowym zabiegiem, aby nie przesłaniać innowacyjności i wydajności programu oraz sprytu i pomysłowości ekipy pracującej nad jego logiką.

e. Wyodrębnienie i zdefiniowanie zadań

Krótką pogadanka w temacie rozpoczęcia pracy nad projektem zaowocowała podziałem naszej pracy na następujące bloki:

1. Znalezienie technologii, która umożliwi zrealizowanie założeń projektu oraz pozwoli na proste i szybkie skonfigurowanie środowiska programistycznego.
2. Opracowanie przepływu danych w programie oraz metodyki działań pracy w zespole.
3. Implementacja i testowanie programu.

f. Decyzja o wyborze narzędzi programistycznych

Środowisko jakie wybraliśmy to CodeBlocks ze względu na prostotę w skonfigurowaniu i podpięciu biblioteki WxWidgets. Realizacja projektu nie wymagała korzystania ze skomplikowanych narzędzi i rozbudowanego środowiska programistycznego.

5. Podział pracy i analiza czasowa.

W związku z natłokiem spraw zarówno okołouczałnianych, jak i pozauczelnianych, praca nad projektem była dosyć spontaniczna, więc nie chcemy jej na siłę systematyzować w tym punkcie. Zadania delegowaliśmy na bieżąco i instynktownie na podstawie znajomości własnych umiejętności.

6. Opracowanie i opis niezbędnych algorytmów

Cały program opiera się na jednym prostym algorytmie. Najpierw pobiera on ścieżki do wszystkich zdjęć z katalogu źródłowego. Następnie dla każdego pliku, na który wskazuje ścieżka, wykonuje operacje przeskalowania obrazu tj. tworzy miniaturkę o wymiarach 150x110px. Miniaturka wraz z podpisem, będącym nazwą pliku jest wklejana na wyjściową bitmapę. Po przeskalowaniu zestawu 20 zdjęć algorytm prosi o podanie treści komentarza, który będzie umieszczony na indeksie poniżej miniaturki i zapisuje bitmapę w formie pliku BMP. W momencie, gdy ilość zdjęć nie będzie podzielna przez 20, czyli indeks nie będzie posiadał 20 miniaturki, algorytm również poprosi o komentarz i zapisze niekompletny indeks.

7. Kodowanie

Program napisany z wykorzystaniem biblioteki WxWidgets. Kodowanie sprowadziło się do stworzenia prostego interfejsu graficznego z wykorzystaniem ów biblioteki oraz zaimplementowania powyżej opisanego algorytmu w języku C++. Kod zawiera intuicyjną konwencję nazewnictwa zmiennych oraz komentarze ułatwiające zrozumienie logiki jego działania

8. Testowanie

Testowanie programu przeprowadzaliśmy w dużej mierze równoległe z procesem implementacji. Dzięki temu udało się naprawić niektóre błędy logiczne i implementacyjne na wczesnym etapie. Ponadto po ukończeniu pisania programu przeprowadziliśmy testy, które pomogły naprawić kilka niedociągnięć. Testy przeprowadzaliśmy manualnie opierając się o metodykę pt. "Co by tu można zepsuć?"

9. Wdrożenie, raport i wnioski

Wszystkie podstawowe wymagania projektu zostały spełnione, dodatkowo dołączyliśmy interfejs graficzny, aby ułatwić obsługę programu dla użytkownika. Podczas pracy nad projektem zdobyliśmy praktyczne umiejętności pracy w zespole i organizacji zadań. W kwestii niedociągnięć i aspektów do poprawienia w przyszłości priorytetem mogłaby być poprawa szaty graficznej tj. np. rodzaj czcionki w komentarza i podpisów, obramowanie wokół miniaturki, lub udostępnić możliwość parametryzowania ilości miniaturki na indeksie, koloru tła.

