Nietypowy pokaz slajdów

Damian Łączak , Krzysztof Stachańczyk , Ernest Jęczmionek

# Opis projektu

Celem projektu było stworzenie programu do wyświetlania zdjęć. Miał on mieć możliwość na wybranym przez użytkownika tle pełnoekranowego wyświetlania zdjęć w ramkach polaroid ustalonego koloru z wybranym opisem. Zdjęcia mają mieć możliwość obrotu oraz płynnego przechodzenia jedne w drugie.

# Założenia wstępne

* Program będzie uruchamiany na systemie Windows 7 lub nowszym.
* Formatami obsługiwanymi zdjęć będą bmp, png, tga, jpg, gif, psd, hdr i pic.

# Analiza projektu

### Specyfikacja danych wejściowych

Konfiguracja programu

<program>

<frame>true/false</frame>

<background><nazwa pliku tła></background>

<mode><przejścia 1-normalne, 2-gładkie, 3-losowe></mode>

<speed><mnożnik szybkości przejść></speed>

<delay><opóźnienie między zdjęciami></delay>

</program>

Konfiguracja poszczególnego zdjęcia – wszelkie wartości liczbowe podajemy w liczbach naturalnych

<images>

<image>

<filename><nazwa zdjęcia></</filename>

<center>

<x><Współrzędna x środka zdjęcia></x>

<y>><Współrzędna y środka zdjęcia></y>

</center>

<size>

<x><Szerokość zdjęcia></x>

<y>><Wysokość zdjęcia></y>

</size>

<rotation>

<Kąt obrotu zdjęcia>

</rotation>

<description>

<opis zdjęcia: Float>

</description>

<frame><!—Pozwala wybrać kolor ramki w formacie RGB -->

<r><Czerwony 0-255></r>

<g>><Zielony 0-255></g>

<b>><Niebieski 0-255></b>

</frame>

</image>

</images>

### Dane wyjściowe

Oczekiwanym efektem jest wyświetlenie pokazu slajdów.

### Struktury danych

Ze względów wydajnościowych postanowiliśmy do przechowywania zdjęć w pamięci użyć szablonowej klasy vector biblioteki standardowej języka C++. Struktura ta była wypełniona obiektami klasy Image( nasza własna klasa na podstawie sf::Image) co pozwalało na łatwe operacje na zdjęciach, gdyż jest to najbardziej bazowa klasa biblioteki SFML, z której korzysta wiele innych klas.

### Zdefiniowanie zadań

* przygotowanie struktury zdjęć do późniejszej obróbki
* przygotowanie ekranu
* Stworzenie struktury obrazka
* Stworzenie obramowania
* Wczytanie XMLa
* Interpolacja bilinearna
* Alfablending
* Scalenie oprogramowania
* Sporządzenie dokumentacji

### Wybór narzędzi

Jako środowiska programistycznego używaliśmy Visual Studio 2015. Jako biblioteki graficznej użyliśmy SFML 2.4.1, a jako systemu kontroli wersji użyliśmy git’a.

Dodatkowo użyliśmy wtyczki do Visual Studio o nazwie Visual Leak Detector pozwalającą na znajdowanie wycieków pamięci.

Wybór narzędzi był podyktowany znajomością ich przez zespół oraz szerokie rozpowszechnienie w środowisku programistycznym.

# Podział pracy i analiza czasowa

Damian Łączak

* Stworzenie struktury obrazka
* Stworzenie obramowania
* Wczytanie XMLa
* Scalenie oprogramowania

Krzysztof Stachańczyk

* Interpolacja bilinearna
* Alfablending
* Zrównoleglenie obliczeń

Ernest Jęczmionek

* Przygotowanie struktury zdjęć do późniejszej obróbki
* Przygotowanie ekranu
* Sporządzenie dokumentacji

# Opis algorytmów

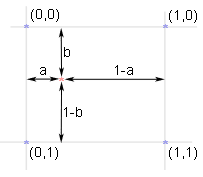
### Interpolacja bilinearna

Do rotacji oraz skalowania użyliśmy interpolacji bilinearnej ( opis na podstawie skalowania).

Obraz ma początek w środku układu współrzędnych.

Aby otrzymać współrzędne wyjściowe :

Przyjmijmy, że rysunek przedstawia nowo powstały punkt.



Jak sama nazwa wskazuje będziemy musieli przeprowadzić interpolację w dwóch osiach. Im sąsiad jest bliższy tym wkłada większą wagę do rozwiązania.

Oś pozioma:

Pozostała nam oś pionowa, musimy wybrać pochodną dwóch wcześniej wyliczonych interpolacji.

### Alfablending

Algorytm polega na zdyskretyzowanym przejściu pomiędzy pixelami:

, gdzie

-wartość RGB pixela wejściowego, – wartość RGB pixela docelowego, – wartość RGB pixela przejściowego

# Kodowanie

Dokumentacja projektu oraz diagramy znajdują się w folderze DOC.

# Testowanie

Testowanie odbywało się przy pomocy osób trzecich, dzięki czemu mogliśmy sprawdzić działanie programu podczas używania go przez osobę z nią niepowiązaną.

# Wdrożenie i wnioski

Wszystkie założenia projektowe zostały spełnione. Zrównoleglenie porzuciliśmy ze względu na zbyt duże koszty obliczeniowe przy dostępie do pamięci. W przyszłości moglibyśmy rozwinąć program na platformy inne niż Windows.