## WOJSKOWA AKADEMIA TECHNICZNA

# *LABORATORIUM*

# *GRAFIKA KOMPUTEROWA*

Stopień, imię i nazwisko słuchacza

Grupa szkoleniowa

Stopień, imię i nazwisko prowadzącego

***Grzegorz Pol***

***I7X3S1***

***dr inż. Marek Salamon***

Data wykonania ćwiczenia

***12.11.2008 r.***

***SPRAWOZDANIE***

***Z***

***PRACY LABORATORYJNEJ***

***NR 1***

|  |  |
| --- | --- |
| **Temat:** | Przekształcenia obrazów rastrowych. |

* 1. **Zadania:**

Podczas zajęć laboratoryjnych należało wykonać zadania o poniższej treści:

1. Napisać algorytm sterujący generatorem adresu odczytu w celu uzyskania efektu zasłaniania poziomego obrazu w kierunku prawej strony ekranu
2. Napisać algorytm sterujący generatorem adresu odczytu w celu uzyskania efektu przewijania obrazu wzdłuż przekątnej ekranu w kierunku górnego lewego wierzchołka
3. Napisać algorytm sterujący generatorem adresu odczytu w celu uzyskania efektu przesuwania pionowego obrazu w kierunku górnej krawędzi ekranu
   1. **Użyte metody**

*ReadPixel(int i, int j)* - we wskazane miejsce wyświetla piksel o wartościach *int i* oraz *int j* obrazu wejściowego, który jest zapisany w pamięci.

*ReadTlo(kolor) )* - we wskazane miejsce wyświetla piksel o kolorystyce wskazanej przez *kolor*, czyli ustalone przez nas wcześniej tło.

* 1. **Implementacja algorytmu realizującego efekt zasłaniania poziomego obrazu w kierunku prawej strony ekranu**

public void Efekt1(){

if (p >= L) p = 0; // zerowanie klatek przy przekroczeniu wymiaru rysunku

for(int j=1; j<=L; j++){ // wiersze obrazu

for(int i=1; i<=K; i++) // kolumny obrazu

if(i <= p) // jeżeli kolumna nie jest powyżej licznika klatek

ReadTlo(N); // wczytaj tło

else // w przeciwnym wypadku

ReadPixel(i, j); // odczytaj rysunek

}

}

Obrazek dzielimy na dwie części. Jedną z nich wypełniamy czarnym tłem, a drugą rysunkiem. Szerokość tła zależy od licznika klatek.

W funkcji Efekt1 przeglądamy każdy piksel obrazu. Bierzemy każdy wiersz i sprawdzamy w nim kolumny. Jeżeli nr danej kolumny jest większy od licznika klatek to odczytujemy rysunek bez jakichkolwiek zmian. W innym wypadku uzupełniamy dany piksel jednolitym tłem.

* 1. **Zobrazowanie efektu**



* 1. **Implementacja algorytmu realizującego efekt przewijania obrazu wzdłuż przekątnej ekranu w kierunku górnego lewego wierzchołka**

public void Efekt2(){

if (p >= L) p = 0; // zerowanie klatek przy przekroczeniu wymiaru rysunku

for (int j = 1; j < L; j++){ // wiersze obrazu od pierwszego do ostatniego

for (int i = 1 + p; i <= K; i++) // kolumny obrazu od licznika klatek do ost.

ReadPixel(i, j); // odczytaj rysunek

for (int i = 1; i <= p; i++)// kolumny obrazu od pierwszego do licznika klatek

ReadPixel(i, j); // odczytaj rysunek

}

}

W tym wypadku obrazek jest podzielony jest na cztery części. Warto zauważyć że w każdej klatce wyświetlane są wszystkie piksele. Funkcja2 polega na pobieraniu po kolei każdego wiersza obrazku, a następnie „przesunięć” pikseli w zależności od wartości p czyli licznika klatek.

* 1. **Zobrazowanie efektu**



* 1. **Implementacja algorytmu realizującego efektu przesuwania pionowego obrazu w kierunku górnej krawędzi ekranu**

public void Efekt3(){

if (p >= L) p = 0; // zerowanie klatek przy przekroczeniu wymiaru rysunku

for (int j = 1+p; j <= L; j++){ // wiersze obrazu od licznika klatek do ostatniego

for (int i = 1; i <= K; i++) // wszystkie kolumny

ReadPixel(i,j); // przerysowywanie rysunku

}

for (int j = 1; j <= p; j++){ // wiersze obrazu od 1-szego do licznika klatek

for (int i = 1; i <= K; i++) // wszystkie kolumny

ReadTlo(N); // wypełnianie tłem

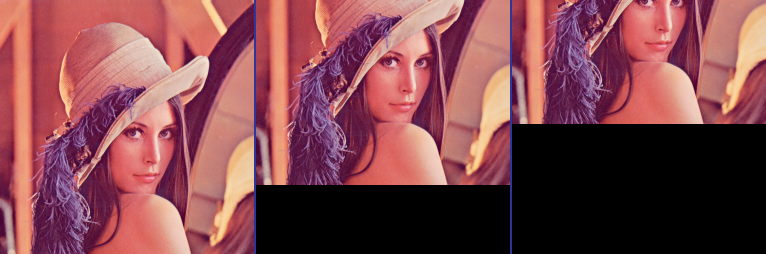
}

}

Obrazek tutaj także dzielimy na dwie części i tak samo jak w przypadku funkcji pierwszej jedną z nich wypełniamy czarnym tłem, a drugą rysunkiem. Wysokość tła zależy od licznika klatek.

W funkcji Efekt3 przeglądamy tak samo jak w przypadku Funkcja1 każdy piksel obrazu. Są tutaj dwie pętle pobierające wiersze. Jedna z nich odpowiedzialna jest na pobieranie wiersza następnego i zapisywanie go w aktualnym. Za każdym razem czynimy to o jeden raz mniej. Za to druga pętla, która wypełnia tłem resztę obrazku wykonuje jeden obieg za każdym razem więcej. W ten sposób uzyskujemy efekt przesuwania obrazku do góry.

**5.2 Zobrazowanie efektu**



**6. Wnioski**

Podsumowując: wszystkie zadania z pkt. 1 zostały przeze mnie pomyślnie zrealizowane. Powyższe ćwiczenia pokazują, że aby przekształcić obraz rastrowy nie potrzeba pisać skomplikowanych algorytmów. Za pomocą kilku nieskomplikowanych pętli jesteśmy w stanie przewinąć, przesunąć, odbić, obrócić bądź to zasłonić taki oto obraz w dowolnym kierunku.