Projekt : Mapa wysokosci

Autor : Rafal Braun

May 14, 2014

Contents

1 Wstep

Na poczatku, zanim zostana wykonanie jakiekolwiek czynnosci zwiazane z wlasciwym rysowaniem mapy, dane wejsciowe w postaci plikow mapa.txt oraz przekroj.txt zostaja wczytane za pomoca odpowiedniech wywolan systemowych a nastepnie sparsowane w pozadanej przez program formie. Oznacza to, ze kolejno wczytywane kody ASCII zostaja zamienione na liczby w postaci heksadecymalnej umieszczone w rejestrach. Nastepnie odczytane liczby zostaja kolejno umieszczone w petli na stosie w celu wygodnego ich odczytania. Dane zostaja zachowane na stosie, gdyz musza byc wykorzystane zarowno w czescie rysujacej mape jak i przekroj.

2 Parsowanie

Dane wejsciowe sa wczytywane kolejno bajt po bajcie. Po natrafieniu na spacje lub znak nowej linii program przechodzi do funkcji odczytujacej jedna, dwie, trzy lub cztery liczby zaleznie od rozmiaru ostatnio wczytanej liczby. Nastepnie poprzez odjecie wartosci 0x30 (zamiana kodu ASCII na odpowiadajaca liczbe), pomnozenie wartosci w rejestrach przez odpowiednie wagi (1,10,100,1000) oraz dodanie rezulatatow otrzymujemy liczbe odpowiadajaca jej zapisowi w formaci ASCII w pliku.

3 Rysowanie mapy wysokosci

Rysowanie odbywa sie w petli poprzez odczytanie obecnie przetwarzanego piksela ze stosu, przeskalowanie wysokosci (pomnozenie razy 400/255) oraz zapis do pliku poprzez obliczenie jego pozycji odpowiadajacej wartosci licznika w petli. Zapisywana jest wartosc wysokosci odczytana ze stosu poprzez zapis wysokosci (wynik jest widoczny tylko w przedziale wysokosci 0-255) w odcieniu szarosci (stad wynika organiczenia). Odbywa sie to dzieki zapisaniu powielonej wartosci na kazdym z trzech bajtow odpowiadajacych za kolor. Dodatkowo przy zwiekszaniu kazdej wartosci y konieczne jest dodanie bajtu 0x00. Wynika to z koniecznosci wyrownania do wartosci podzielnych przez 4 kazdej linii poziomej (specyfikacja pliku bmp).

4 Rysowanie czerwonej linii

Do rysowania linii uzywany jest algorytm Bresenhama. Dziala poprzez zapisanie w odpowiednich miejscach bufora posredniczacego (ktory zostaje potem w calosci zrzucony do pliku wraz z naglowkiem) danych w formie pikseli. Za jeden piksel odpowiadaja 3 bajty. Aby uzyskac czerwony kolor w formacie RGB pierwszy bajt koloru zostaje uzupelniony wartoscia 0xff, reszta 0x00.

```
Pseudkod algorytmu:

    dx - odleglosc x2 od x1
    dy - odleglosc y2 od y1
    e - wartosc wyrazenia bledu

Lista krokow

    K01:  dx = x2 - x1
    K02:  dy = y2 - y1
    K03:  e = dx / 2
    K05:  Zapal piksel x1,y1
    K04:  Wykonuj dx razy kroki K06-K10
    K06:  x1 = x1 + 1
    K07:  e = e - dy
    K08:  Jesli e > 0, idz do kroku K11
    K09:  y1 = y1 + 1
    K10:  e = e + dx
    K11:  Zapal piksel x1,y1
    K12:  Koniec
```

5 Rysowanie przekroju

Rysowanie przekroju bocznego odbywa sie poprzez obliczenie pozycji(wysokosci) kolejnych pikseli na obrazku po odczytaniu ze stosu, zapis na odpowiednich miejscach i polaczeniu kolejno rysowanych pikseli.

6 Opis testowania

Testowanie odbywa sie za pomoca napisanego wczesniej pliku w jezyku python. Wystarczy uruchomic skrypt za pomoca polecenia './test.sh' aby pojawily sie foldery z wynikami testow. Skrypt generuje za pomoca petli podwojnej i rownania okregu pliki w formacie txt, ktore zostaja potem uzyte do generacji folderow z wynikami programu, losowe szumy, ostry spadek oraz rowna powierzchnie.