

Odchudzanie plików .eps

Dokumentacja końcowa

1. Opis funkcjonalności

Udało się zrealizować, działający na systemach windows i linux, program odchudzający pliki .eps. Program analizuje instrukcje rysujące prostokąty i łamane, po czym zapisuje wyjściowy plik z uproszczonymi instrukcjami. Dokumentacja zrealizowana została za pomocą generatora doxygen. Ze względu na upływający termin ukończenia projektu, program jest w stanie upraszczać tylko pliki .eps, które spełniają poniższe wymagania:

- Zawierają następujące definicje:
 - `"/gs { gsave } bind def"`
 - `"/bg { 0 0 0 srgb } def"`
 - `"/srgb { setcolor } bind def"`
 - `"/r { 4 2 roll moveto 1 copy 3 -1 roll exch 0 exch rlineto 0 rlineto -1 mul 0 exch rlineto closepath } bind def"`
 - `"/p2 { gsave bg fill grestore newpath } bind def"`
- Korzystają jedynie z instrukcji rysujących: r p2, lineto, rlineto, moveto, newpath (reszta instrukcji rysujących nie zostanie zapisana w wyjściowym pliku)

Liczba linii kodu projektu: 1155

Zrealizowane zadania projektowe z dokumentacji wstępnej:

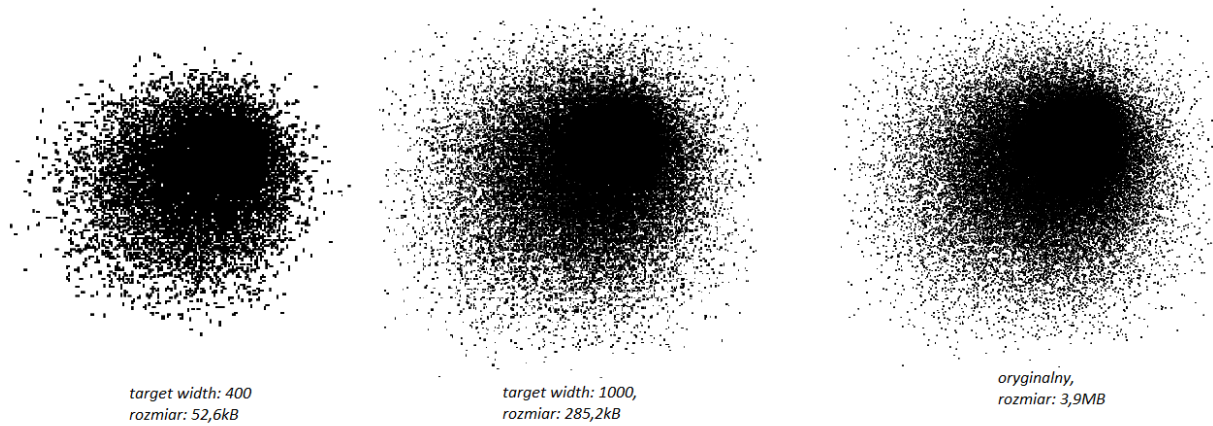
- Interpretacja instrukcji eps
 - zapisanie instrukcji rysujących w postaci zdefiniowanych w programie obiektów wygodnych do analizy (tylko częściowo)
 - napisanie instrukcji na podstawie stanu końcowego obiektów po ich przetworzeniu
- Upraszczenie instrukcji rysujących punkty:
 - łączenie punktów, które się nakładają, lub odstęp pomiędzy nimi jest bardzo mały w stosunku do rozdzielczości
 - usuwanie odosobnionych punktów, które są zbyt małe by je zobaczyć w docelowej rozdzielczości
- Upraszczenie instrukcji rysujących odcinki:
 - w jednym łańcuchu, scalanie dwóch krótkich odcinków o podobnym współczynniku kierunkowym

Niezrealizowane zadania projektowe:

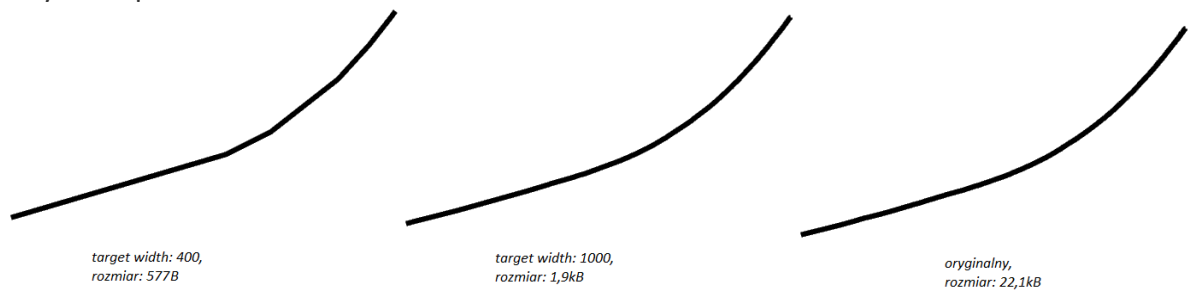
- usuwanie krótkich, mało widocznych łańcuchów (zrezygnowałem ze względu na małą przydatność)

Efekty programu na plikach przykładowych zamieszczonych w zadaniu:

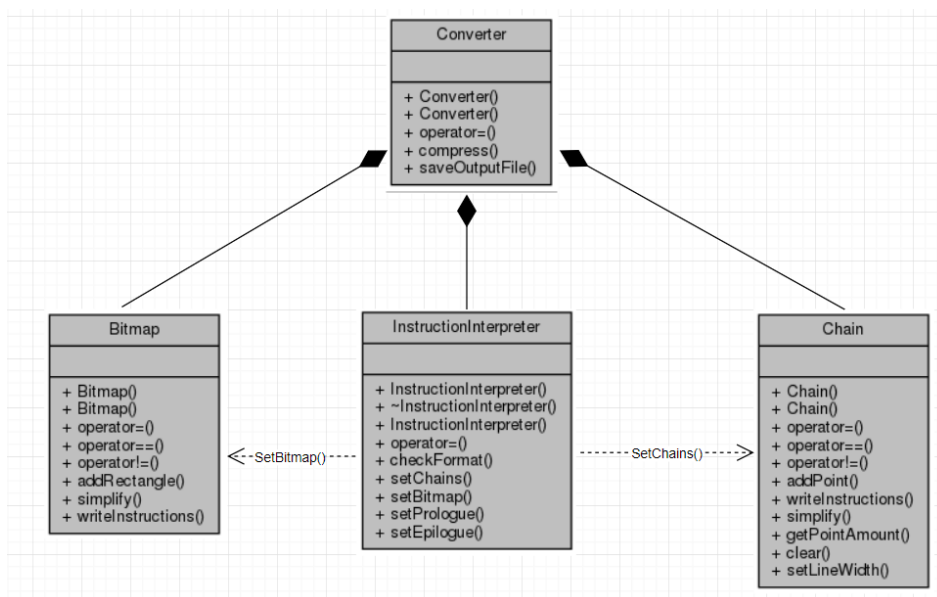
- EPSpunkty.eps



- Przykład.eps



2. Opis klas



- Converter

Klasa zarządzająca klasami InstructionInterpreter, Chain i Bitmap. Przechowuje obiekt Bitmap jako wskaźnik, a obiekty Chain w postaci wektora. Wywołuje funkcje upraszczające i piszące instrukcje klas Chain i Bitmap. Wywołuje funkcje piszące prolog i epilog w InstructionInterpreter. Tworzy na tej podstawie plik wyjściowy.

- **Chain**

Klasa przechowująca obiekty tworzone przy pomocy instrukcji lineto i rlineto. Upraszczenie polega na mierzeniu różnicy pomiędzy długością dwuodcinkowej łamanej i prostej po usunięciu środkowego punktu. Jeżeli ta różnica jest mniejsza od wartości visible_length punkt środkowy jest usuwany. Wypisuje swoje instrukcje rysujące.

- **Bitmap**

Klasa przechowująca obiekty tworzone przy pomocy instrukcji r p2. Upraszczenie polega na zapisie prostokątów do bitmapy o jednostkowej wielkości równej visible_length oraz usuwaniu prostokątów 1x1 w bitmapie. Następnie Bitmap znajduje zapisane prostokąty i je rysuje. Bitmapa jest przechowywana w postaci własnej tablicy dwuwymiarowej, która przechowuje elementy w wektorze.

- **InstructionInterpreter**

Klasa analizująca plik wejściowy. Sprawdza ona, czy plik spełnia wymagania do analizy (tylko częściowo zrealizowane), wyszukuje odpowiednie instrukcje rysujące i dodaje odpowiednie obiekty do obiektów Chain i Bitmap przechowywanych w Converter. Converter przekazuje wektor Chain i wskaźnik do Bitmap przez referencje w metodach setChains i setBitmap. Ustawia na podstawie pliku wejściowego prolog i epilog pliku wyjściowego (tylko częściowo zrealizowane).

3. Czas poświęcony na zadania

Zadanie	Liczba godzin zamierzona	Liczba godzin rzeczywista
Klasa Chain	2	3
Klasa Compress (zawierająca obiekty)	3	3
Interpretacja prologu eps	4	2
Klasa Bitmap	-	3
Translacja instrukcji rysujących na obiekty	4	4
Łączenie punktów	3	6
Usuwanie małych punktów	2	3
Usuwanie krótkich łańcuchów	3	-
Łączenie odcinków w łańcuchu	4	4
Translacja obiektów na instrukcje rysujące i zapis pliku .eps	4	6
Dokumentacja	2	2

Łączny czas zamierzony: 31; łączny czas rzeczywisty: 36

Rzeczywisty czas na wykonanie zadań, mniej więcej, pokrywa się z zamierzonym:

Interpretacja prologu eps została znacznie uproszczona, ze względu na brakujący czas, więc została wykonana krócej.

Łączenie punktów zajęło znacznie dłużej, bo pomysł na jej realizację zmieniał się w trakcie implementacji.

Zwiększony czas pracy nad projektem po części wynika z niedostatecznego zapoznania się z formatem Encapsulated PostScript, przez co niektóre części programu musiały być zmieniane.