SVG editor app

Project

Глава 1

Увод

Това приложение дава възможност на потребителя да отваря файлове от тип SVG(saleable vector graphics) и зареждайки фигурите. Програмата позволява на клиента да прави промени по фигурите във файла, както и по самия файл.

Целта на проекта е да може потребителя да управлява файла и фигурите в него.

Действията които могат да бъдат приложени върху файла включват:

Отваряне на файл – open – което зарежда всички фигури, които програмата поддържа, в паметта на компютъра и после го затваря.

Затваряне – close – „затваря“ текущия файл, като изчиства заредените в паметта фигури.

Запазване – save - запазва всички фигури и промените направени върху тях, като презаписва файла. Файла отново е в svg формат и може да бъде отварян от други програми.

Запазване като – saveas – Създава нов файл отново в svg формат. Отново може да бъде отварян от други програми които поддържат svg.

Принтиране – print – програмата принтира всички фигури, заредени в паметта, на стандартния изход(Terminal).

Програмата може да прилага следните операции върху фигурите, които са заредени от файла:

Създаване на фигури – create - създаването приема като аргумент вида на фигурата както и другите аргументи които са по стандарт на svg файловете. Тази команда позволява да се създаде нов svg файл, без да бъде отварян файл вече готов, т. е. програмата може да създава самостоятелно файлове.

Изтриване на фигура – erase <n> - изтрива фигура с индекс n от фигурите заредени в паметта.

Транслация – translate <horz> <vert> (optional <n>) – транслира всички фигури заредени в паметта с подадените стойности по хоризонтала и вертикала. Ако бъде подаден и номер, само тази фигура ще бъде транслирана.

Проверка кои фигури се съдържат в определена област – within – областта може да бъде кръг, който се определя от координатите на центъра и радиус, или квадрат, който се определя от координати на горен ляв ъгъл и дължина и ширина.

Самата програма има команда help която извежда на стандартния изход всички команди с кратки обяснения, както и команда exit която прекратява програмата.

Глава 2

Проблеми

Основният проблем с който се сблъсках бе зареждането на фигурите от файла. Това че аргументите могат да бъдат както си искат подредени в тага, и това че могат да бъдат таговете на повече от един ред. Както и да идентифицирам каква фигура дефинира тага. За сега програмата поддържа 3 базови фигури:

1. Квадрат
2. Кръг
3. Линия

Структурата на програмата е направена така, че добавянето на нови фигури да стане възможно най – лесно. Задачата бе предизвикателна за изпълнение. Не съм използвал някакви определени методи за решаване на проблемите. Главният метод е проба грешка. Изпробвам решение което съм измислил и ако не сработи го променям, така че да сработи.

Глава 3

Картина, която съдържа диаграма, кръг, скица, Шрифт

Описанието е генерирано автоматичноПроектиране

Диаграмата по горе показва дървото на наследяване на класовете. Имаме базов клас Figures с наследници Line, Rectangle, Circle. Базовият клас е абстрактен и наследниците му имплементират методите спрямо своите нужди. Избрана е тази структура за да може да се направи абстракция в методите които обработват групата от фигури, за да може те да работят без проблем дори и нови фигури да бъдат добавени в последствие.

Всеки едни клас е дефиниран в hpp файл и има разписани методи в cpp файл със същото име. Освен тях има и hpp файл с функции който се използват в main файла, но са изместени за по лесно разчитане на кода и подреденост.

Файлова структура и използвани библиотеки:

1. Main.cpp
   1. Iostream
   2. Vector
   3. Sstream
   4. Fstream
2. Func.hpp
   1. Func.cpp
3. Figure.hpp
   1. Line.hpp
      1. Line.cpp
   2. Rectangle.hpp
      1. Rectangle.cpp
   3. Circle.hpp
      1. Circle.cpp

Глава 4

Възможни тестове:

1. open <filePath>  
   translate 100 100  
   print  
   erase 2  
   saveas temp.svg  
   save  
   close  
   exit
2. create circle 6 6 3 blue black 0  
   create rectangle 10 10 10 10 blue black 10  
   print  
   saveas temp2.svg  
   exit

Глава 5

Възможности за подобрение:

1. Поддържане на всички базови фигури и криви
2. Добавяне на групи
3. Оптимизация на алгоритми
4. Добавяне на визуален интерфейс
5. Визуализация на фигурите
6. По - добро разчитане на фигури от файл

Изготвил документация и проект: Борислав Иванов

Използвана литература:

Статии в Stack overflow.

Cpp reference

W3.org