СОФИЙСКИ УНИВЕРСИТЕТ „СВ. КЛИМЕНТ ОХРИДСКИ“

SOFIA UNIVERSITY ST. KLIMENT OHRIDSKI

Курсова работа

на тема:

Реализиране на софтуер, представящ алгоритъма за намиране на cross plot на крива на Bézier

За курса:

Компютърно геометрично моделиране (CAGD)

Лектор: доц. Красимира Влъчкова

Изготвена от:

Петър Петков Кирилов

II курс, Софтуерно инженерство

София, 2024

Условие на задачата

При въвеждане на входни точки от потребителя, програмата трябва да начертава кривата на Bézier с тях като контролни точки. Ако потребителят пожелае, софтуерът начертава и cross plot-а на тази крива.

Работа с програмата

След стартирането на програмата се появява прозорец с фиксирани размери, върху който е изобразена Декартова координатна система. Потребителят може да добавя и премахва контролни точки само и единствено в първи квадрант. Във втори квадрант ще бъде начертана y(t) функцията на Bézier, а в четвърти квадрант – x(t) функцията. Тъй като трети квадрант не служи за изобразяване на елементи, то в неговата област е направено кракто описание за работа със софтуера. По-надолу са описани различните възможности, предоставени на потребителя:

1. ***При натискане на левия бутон на мишката върху първи квадрант*** потребителят създава нова контролна точка на мястото на курсора в момента на натискане. След всяка нова точка се чертае кривата на Bézier с обновените контролни точки (новата точка е добавена към предишните и така получаваме нов брой контролни точки и съответно различна крива на Bézier).

2. ***При натискане на десния бутон на мишката върху първи квадрант*** потребителят изтрива последната добавена контролна точка (ако такава съществува). След всяко такова изтриване, се чертае новата крива на Bézier с обновените контролни точки (точката се премахва от контейнера с контролни точки и така ще получим нова крива на Bézier).

3. ***При натискане на клавиша ’x’*** се начертава/скрива x(t) функцията от cross-plot-а на кривата на Bézier от първи квадрант. X(t) се изобразява в четвърти квадрант.

4. ***При натискане на клавиша ‘y’*** се начертават/скрива y(t) функцията от cross-plot-а на кривата на Bézier от първи квадрант. Y(t) се изобразява във втори квадрант.

5. ***При натискане на клавиша ‘p’*** се начертават/скриват контролните точки във всички видими графики.

6. ***При натискане на клавиша ‘l’*** се начертава/скрива полигона от контролните точки във всички видими графики.
7. ***При натискане на клавиша ‘c’*** се начертава/скрива кривата на Bézier във всички видими графики.

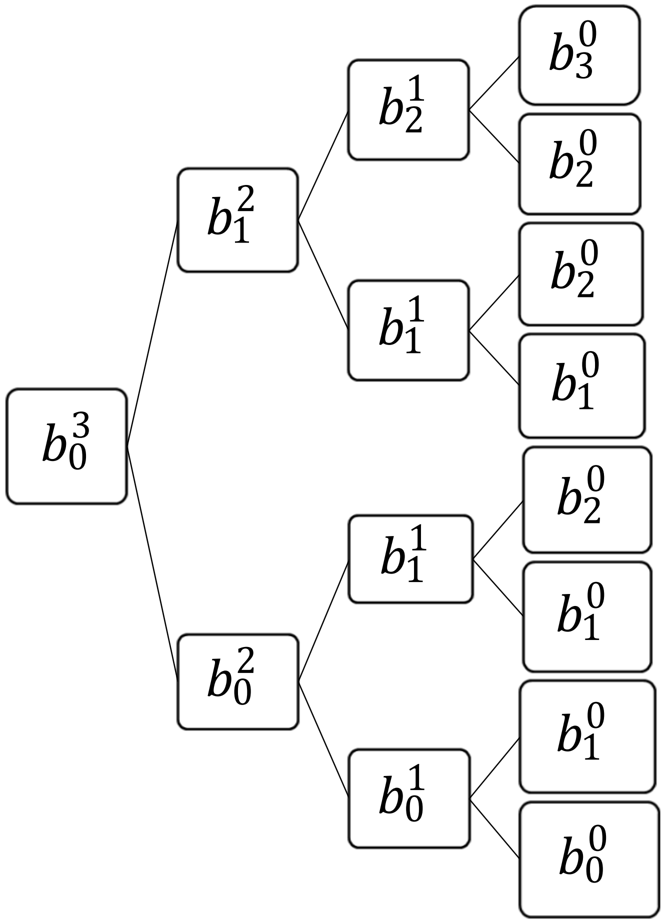
Математическо описание:

❖ **<u>Пресмятане на точките от кривата на Bézier</u> –** това е една от основните функционалности в тази програма, защото чрез нея се изчислява точка от кривата на Bézier при подадени контролни точки и дадена стойност **t**. Алгоритъмът, който е използван тук, е рекурсивният алгоритъм на *de Casteljau:*

$$b^r_i(t) = (1 - t)\, b^{r-1}_i(t) + t b^{r-1}_{i+1}(t),\ r = 1,\dots, n;\ i = 0, \dots, n - r$$

$$b^0_i = b_i$$

На картинката долу е показано как рекурсията изчислява търсената точка b_0^3 при контролни точки (b_0, b_1, b_2, b_3) за някое **t**.
Първоначално алгоритъмът започва от b_0^3, като за да я изчисли са му нужни точките b_1^2 и b_0^2. Затова той започва да изчислява b_0^2 и след това b_1^2. Този процес се повтаря за всички междинни точки, докато не се стигне до контролна точка (тоест r = 0). След като се стигне до последното ниво на горната йерархия (тоест се достигнат контролните точки), рекурсията започва да се връща и по този начин да смята стойностите на междинните точки. Така, когато алгоритъмът получи стойностите на точките b_1^2 и b_0^2, изчислява b_0^3 по гореописаната формула за някаква стойност на **t** и връща тази точка.



❖ **Изчисляване координатите на контролните точки на функционалната крива x(t)**

За всяка контролна точка на функционалната крива:

- стойността на x координатата е същата като съответната от i-тата контролна точка (i-та по ред на поява) на кривата на Bézier.
- y координатата се изчислява по следния начин: Разбива се интервала на четвърти квадрант (за y), който е [h / 2, 0], където h е дължината на прозореца (FRAME_HEIGHT), чрез броя на контролните точки на кривата. По този начин получаваме n + 1 равни подинтервала, където n е броят контролните точки. Краят на всеки един интервал без последния съответства на y-координатата на i-тата контролна точка на функционалната крива. Математически, това може да се изрази чрез следната формула:

$$y_i = (h / 2) + i * ((h / 2) / (n + 1)), \text{ за } i = 1, \ldots, n$$

❖ **<u>Изчисляване координатите на контролните точки на функционалната крива y(t)</u>**

За всяка контролна точка на функционалната крива:

- стойността на y координатата е същата като съответната от i-тата контролна точка (i-та по ред на поява) на кривата на Bézier.
- x координатата се изчислява по следния начин: Разбива се интервала на втори квадрант (за x), който в случая е [0, w / 2], където w е ширината на прозореца (FRAME_WIDTH), чрез броя на контролните точки на кривата. По този начин получаваме n + 1 равни подинтервала, където n е броят контролни точки. Краят на всеки един интервал без последния съответства на x-координатата на i-тата контролна точка на функционалната крива. Математически, това може да се изрази чрез следната формула:

$$x_i = (w / 2) - i * ((w / 2) / n + 1), \text{ за } i = 1, . . ., n$$

Използвана литература

Лекции към курса за бакалаври ,,Компютърно геометрично моделиране”

Използван софтуер

IntelliJ IDEA