Programovacia úloha 1

Téma Bézierova krivka

Termín Odovzdať do **21.3.2013** mailom na **gemeranova@sccg.sk** (archív so zdrojovými súbormi a spustiteľným súborom /podľa pokynov nižšie/)

Cieľom prvej programovacej úlohy je vytvoriť grafické používateľské rozhranie, ktoré budete môcť použiť na zobrazenie určenej krivky a na manipuláciu s ňou. Zároveň je úlohou implementovať dve metódy na vykresľovanie Bézierovej krivky. Vytvorené používateľské rozhranie sa bude dať využiť aj pri ďalších úlohách, preto je na vypracovanie prvej úlohy dlhší čas a hodnotenie používateľského rozhrania má pridelených viacej bodov.

Všeobecné požiadavky

Program bude naprogramovaný v jazyku a prostredí, ktoré bolo vopred dohodnuté s cvičiacim. Podmienkou je, aby sa dala úloha jednoducho skontrolovať. Program musí byť možné spustiť na čistom stroji (knižnice štandardne nedodávané s operačným systémom je potrebné pribaliť), resp. sa musí dať skompilovať. V každom prípade je však potrebné poslať všetky zdrojové súbory.

Ak nebude uvedené inak, výsledný program má umožniť klikaním do plochy pridávať a mazať riadiace vrcholy a ťahaním ich presúvať. Vykresľovaná krivka má byť pri každej zmene prekresľovaná (presúvanie riadiacich vrcholov, zmena parametrov). Žiaden parameter nemá byť nezmyselne obmedzovaný v rozsahu, konkrétne má byť možné zadať najmenšiu možnú zmysluplnú hodnotu. Ak sa bude ovládanie líšiť od vzorovej aplikácie, uveďte ho buď v aplikácii alebo v maile. V aplikáciach nepoužívajte špeciálnu klávesu 'Alt'.

Zadanie

Úlohou je naprogramovať aplikáciu vykresľujúcu Bézierovu krivku. Na vykreslenie použite dve rôzne metódy, pričom v ľubovoľnom momente má byť možné medzi nimi prepínať (bez zmeny krivky). Krivku vykresľujte priamym výpočtom za použitia de Casteljauovho algoritmu. Ako druhú metódu implementujte buď zvyšovanie stupňa alebo prerozdelenie krivky. V nasledujúcom texte predpokladajme, že stupeň krivky je n.

Priamy výpočet Parametrom vykreslenia je hustota vzorkovania. Nech premenná s označuje počet vzorkovacích bodov (min. je 2). Potom ak je krivka definovaná na intervale $[0, 1]$, tento pokryjeme s rovnako vzdialenými bodmi, pričom prvý je bod $t_0 = 0$ a posledný bod $t_{s-1} = 1$. Pre každý bod s_i – hodnotu parametra, vypočítame bod krivky $\mathbf{v}(s_i)$. Následne body spojíme lomenou čiarou aproximujúcou krivku.

Bod krivky počítajte de Casteljauovým algoritmom: nech body $\mathbf{v}_0, \ldots, \mathbf{v}_n$ sú riadiace vrcholy krivky. Potom parametru t prislúcha bod krivky $\mathbf{v}_0^n$, pričom platí rekurentný vzťah na jeho výpočet

$$\mathbf{v}_i^k = (1-t)\mathbf{v}_i^{k-1} + t\mathbf{v}_{i+1}^{k-1} \qquad \text{pre} \qquad i = 0, \ldots, n-k \tag{1}$$

Body $\mathbf{v}_i^0$ sú identické s riadiacimi vrcholmi ($\mathbf{v}_i^0 = \mathbf{v}_i$).

Hint: Výpočet sa dá implementovať efektívnejšie ako použitím dvojrozmerného poľa. Vystačíme si len s dvoma sadami vrcholov – vrcholy predchádzajúceho riadku $\mathbf{v}_i^{k-1}$ a aktuálne počítaného riadku $\mathbf{v}_i^k$. Navyše je možné vypočítať bod krivky *in-situ*, čiže na mieste. Hodnota $\mathbf{v}_0^{k-1}$ sa totiž používa iba pri výpočte hodnoty $\mathbf{v}_0^k$ a teda túto možno uložiť na jej miesto. Následne hodnota $\mathbf{v}_1^{k-1}$ sa využije už iba pri výpočte bodu $\mathbf{v}_1^k$ a teda tento môžeme opäť uložiť namiesto hodnoty $\mathbf{v}_1^{k-1}$. Postup ďalej pokračuje na ďalšie body.

Zvyšovanie stupňa Parametrom vykreslenia je počet zvýšení stupňa r (min. 0). Bézierova krivka je polynomickou krivkou a teda krivka stupňa n je reprezentovateľná ako krivka stupňa $n+1$. Nech $\mathbf{v}_i$ sú riadiace vrcholy Bézierovej krivky pri reprezentácii ako krivky stupňa n a $\mathbf{w}_i$ riadiace vrcholy tej istej krivky pri reprezentácii ako krivky stupňa $n+1$. Platí

$$\mathbf{w}_i = \frac{i}{n+1}\mathbf{v}_{i-1} + \frac{n-i+1}{n+1}\mathbf{v}_i \tag{2}$$

Pre zadanú hodnotu r je teda potrebné vypočítať riadiace vrcholy krivky reprezentovanej ako krivky stupňa $n+r$, kde n je pôvodný stupeň krivky. Zobrazenie potom pozostáva z vykreslenia riadiaceho polygónu krivky so zvýšeným stupňom. Pôvodná krivka sa nijak nemení, zvyšovanie stupňa sa vykonáva na dočasnej kópii, ktorá je použitá len pre vykreslenie.

Poznámka: výsledkom je veľmi hrubá aproximácia výslednej krivky, keďže jediné body skutočne patriace krivke sú koncové body polygónu.

Prerozdelenie krivky Bézierovu krivku stupňa n možno v ľubovoľnom jej bode rozdeliť na dva segmenty a tieto reprezentovať ako separátne krivky stupňa n. Úlohou je pre vstupný parameter m (min. 0) vykonať postupne m operácií prerozdelenia na všetkých segmentoch krivky a následne vykresliť lomenú čiaru danú riadiacimi polygónmi jednotlivých segmentov. Napríklad pre hodnotu $m = 1$ dostávame 2 segmenty, pre hodnotu $m = 2$ dostávame 4 segmenty (oba predchádzajúce segmenty sme prerozdelili), pre $m = 3$ dostávame 8 segmentov atď.

Na výpočet riadiacich vrcholov segmentov krivky sa používa de Casteljauov algoritmus (pozri zadanie k priamemu výpočtu). Ak zoberieme prvý a posledný bod z každého riadku algoritmu, dostaneme dve postupnosti bodov – tieto tvoria hľadané riadiace vrcholy. Konkrétne, nech $\mathbf{v}_i^k$ sú body z de Casteljauovho algoritmu pre nejakú hodnotu parametru t. Potom postupnosti $(\mathbf{v}_0^k \mid k = 0, \ldots, n)$ a $(\mathbf{v}_{n-k}^k \mid k = n, \ldots, 0)$ predstavujú riadiace vrcholy segmentov krivky pre intervaly $[0, t]$ a $[t, 1]$. Oba segmenty sú definované opäť nad intervalom $[0, 1]$.

> Poznámka: Pri výpočte môžete použiť hodnotu $t = 1/2$, ktorá zodpovedá rovnomernému prerozdeleniu krivky alebo poskytnúť používateľovi možnosť hodnotu zvoliť. Metóda už pre relatívne malé hodnoty parametra m poskytuje veľmi presné výsledky, čo je spôsobené jednak tým, že pri prerozdelení sa vypočíta bod krivky a taktiež exponenciálnym počtom segmentov vzhľadom na vstupný parameter.

Ďalšou časťou úlohy je pre používateľom zvolený parameter t vykresliť bod krivky, vypočítaný de Casteljauovým algoritmom. Pre tento bod vizualizujte aj postup výpočtu – lomené čiary prislúchajúce jednotlivým riadkom de Casteljauovho algoritmu. Následne vykreslite dotykový a normálový vektor krivky v tomto bode. Veľkosť vykreslených vektorov nemusí zodpovedať analyticky vypočítaným vektorom – dĺžku môžete upraviť podľa potreby tak, aby sa vektory zmestili na obrazovku. Objekty vykresľujte pri všetkých vizualizačných prístupoch, ktoré ste implementovali.