

MODELOWANIE GEOMETRYCZNE

Zestaw 1

Krzywe kubiczne

Krzywe kubiczne (tj. krzywe stopnia trzeciego) są najczęściej używanymi krzywymi w grafice komputerowej. Są to krzywe parametryczne, które dane są przez cztery punkty kontrolne $P_0, P_1, P_2, P_3 \in \mathbb{R}^2$ (lub \mathbb{R}^3) oraz macierz przejścia od pewnej bazy do bazy potęgowej $(t^3, t^2, t, 1)$. Krzywą kubiczną $C : [0, 1] \rightarrow \mathbb{R}^2$ (lub \mathbb{R}^3) można zapisać w postaci:

$$C(t) = (P_0, P_1, P_2, P_3) \cdot M \cdot \begin{pmatrix} t^3 \\ t^2 \\ t \\ 1 \end{pmatrix}, \quad (1)$$

gdzie $t \in [0, 1]$ oraz $M \in \mathbb{R}_4^4$ jest macierzą przejścia z pewnej bazy do bazy potęgowej.

W zależności od użytej macierzy przejścia będziemy mówić o różnych rodzajach krzywych kubicznych:

1. krzywa Béziera

$$M_B = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad (2)$$

2. krzywa Catmulla–Roma

$$M_{C-R} = \frac{1}{2} \begin{bmatrix} -1 & 2 & -1 & 0 \\ 3 & -5 & 0 & 2 \\ -3 & 4 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}, \quad (3)$$

3. krzywa kardynalna

$$M_c = \begin{bmatrix} -\alpha & 2\alpha & -\alpha & 0 \\ 2-\alpha & \alpha-3 & 0 & 1 \\ \alpha-2 & 3-2\alpha & \alpha & 0 \\ \alpha & -\alpha & 0 & 0 \end{bmatrix}, \quad (4)$$

gdzie $\alpha \in [0, 1]$ jest napięciem krzywej,

4. krzywa b-splain

$$M_b = \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \quad (5)$$

Do opisu pojedynczego segmentu krzywej wystarczy podać cztery punkty kontrolne P_0, P_1, P_2, P_3 oraz znać macierz M danej bazy. Zazwyczaj mamy do czynienia z krzywymi sklejanyymi, składającymi się z wielu segmentów. W pełnej ogólności możemy segment opisywać oddzielną czwórką punktów:

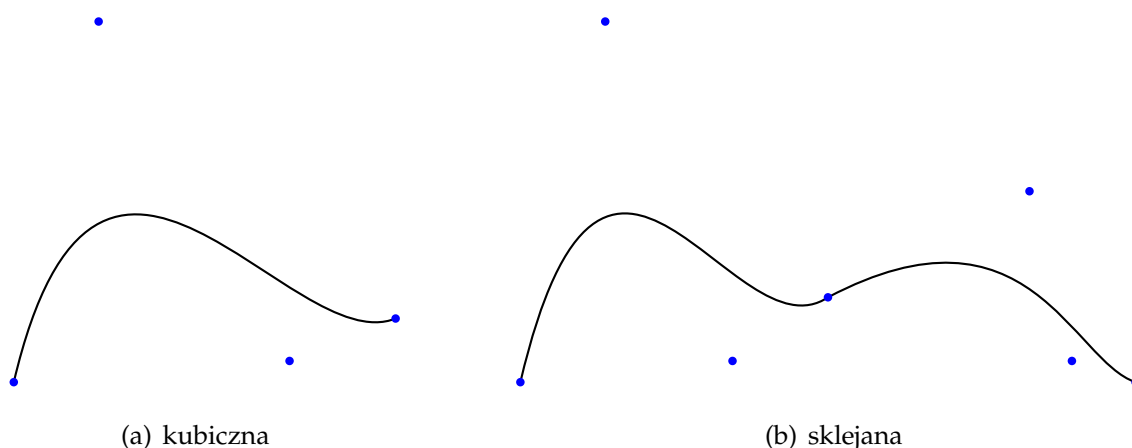
$$(P_0^{(1)}, P_1^{(1)}, P_2^{(1)}, P_3^{(1)}), \dots, (P_0^{(i)}, P_1^{(i)}, P_2^{(i)}, P_3^{(i)}), \dots \quad (6)$$

W praktyce część punktów jest jednak współdzielona pomiędzy kolejnymi segmentami. To znaczy istnieje taka liczba $c < 4$, że $P_{4-c+j}^{(i)} = P_j^{(i+1)}$. Mówimy wtedy, że w danej reprezentacji kolejne segmenty współdzielą c punktów kontrolnych.

Kubiczne krzywe Béziera

Dla kubicznych krzywych Béziera związek pomiędzy punktami kontrolnymi a kształtem krzywej jest następujący: krzywa przechodzi przez punkty P_0 i P_3 oraz styczna w punkcie P_0 jest równa trzykrotności wektora $\overrightarrow{P_0P_1}$, zaś styczna w P_3 jest równa potrojonnemu wektorowi $\overrightarrow{P_2P_3}$. Rysunek 1(a) przedstawia przykładową kubiczną krzywą Béziera.

W przypadku sklejania kubicznych krzywych Béziera, aby połączenie dwóch segmentów było ciągłe musi zachodzić warunek: ostatni punkt kontrolny pierwszego segmentu musi być taki sam jak pierwszy punkt drugiego segmentu. Jeśli chcemy aby segmenty były połączone gładko, to muszą zachodzić warunki: połączenie musi być ciągłe oraz trzeci, czwarty punkt kontrolny pierwszego segmentu oraz drugi punkt kontrolny drugiego segmentu muszą być współliniowe. Rysunek 1(b) przedstawia przykładową sklejaną kubiczną krzywą Béziera.



Rysunek 1. Przykładowe krzywe Béziera

Do znajdowania punktu na krzywej Béziera oprócz wzoru (1) można użyć algorytmu rekurencyjnego zwanego algorytmem de Casteljau (Algorytm 1). Rysunek 2 przedstawia interpretację geometryczną algorytmu de Casteljau. Algorytm de Casteljau nie tylko daje nam możliwość znajdowania punktu na krzywej Béziera, ale również podział krzywej na dwie krzywe Béziera z punktami kontrolnymi $P_0^0, P_0^1, P_0^2, P_0^3$ dla „lewej” krzywej oraz $P_0^3, P_1^2, P_1^1, P_1^0$ dla „prawej” krzywej.

Kubiczne krzywe Catmulla–Roma

Dla kubicznej krzywej Catmulla–Roma związek między punktami kontrolnymi a kształtem krzywej jest następujący: krzywa interpoluje punkty kontrolne P_1, P_2 oraz styczna w punkcie P_1 jest połową wektora $\overrightarrow{P_0P_2}$, zaś styczna w P_2 połową wektora $\overrightarrow{P_1P_3}$. Rysunek 3(a) przedstawia przykładową kubiczną krzywą Catmulla–Roma opartą na tych samych punktach kontrolnych co krzywa Béziera z Rysunku 1(a).

Ze związku między punktami kontrolnymi a kształtem krzywej widzimy iż, aby sklejana krzywa Catmulla–Roma była ciągła potrzeba i wystarcza, aby kolejne segmenty współ-

Algorytm 1: Algorytm de Casteljau znajdowania punktu na kubicznej krzywej Béziera

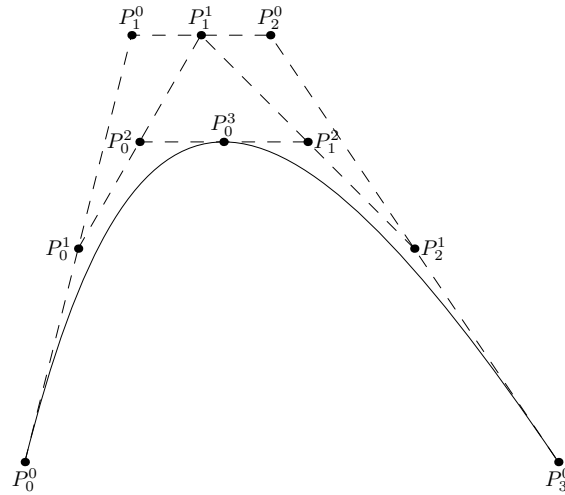
Dane: $P_0^0, P_1^0, P_2^0, P_3^0$ oraz $t \in [0, 1]$

Wynik: punkt na krzywej

for $i = 1, \dots, 3$ **do**

for $j = 0, \dots, 3 - i$ **do**
 $P_j^i = (1 - t) \cdot P_j^{i-1} + t \cdot P_{j+1}^{i-1}$

return $C(t) = P_0^3$



Rysunek 2. Algorytm de Casteljau

dzieliły trzy punkty kontrolne. Ponadto sklejana krzywa Catmulla–Roma jest gładka. Rysunek 3(b) przedstawia przykładową sklejaną krzywą Catmulla–Roma opartą na tych samych punktach kontrolnych co sklejana krzywa Béziera z Rysunku 1(b).

Podobnie jak dla krzywej Béziera również dla krzywej Catmulla–Roma istnieje algorytm znajdowania punktu na krzywej inny niż użycie wzoru (1). Algorytm ten nosi nazwę algorytmu Barry’ego–Goldmana (Algorytm 2). W algorytmie tym współczynnik α_j^i określony jest następująco:

$$\alpha_j^i(t) = \begin{cases} \frac{1}{i}(i + j - 1 - t), & \text{gdy } i \leq 2, \\ 1 - t, & \text{gdy } i = 3. \end{cases} \quad (7)$$

Algorytm 2: Algorytm Barry’ego–Goldmana znajdowania punktu na kubicznej krzywej Catmulla–Roma

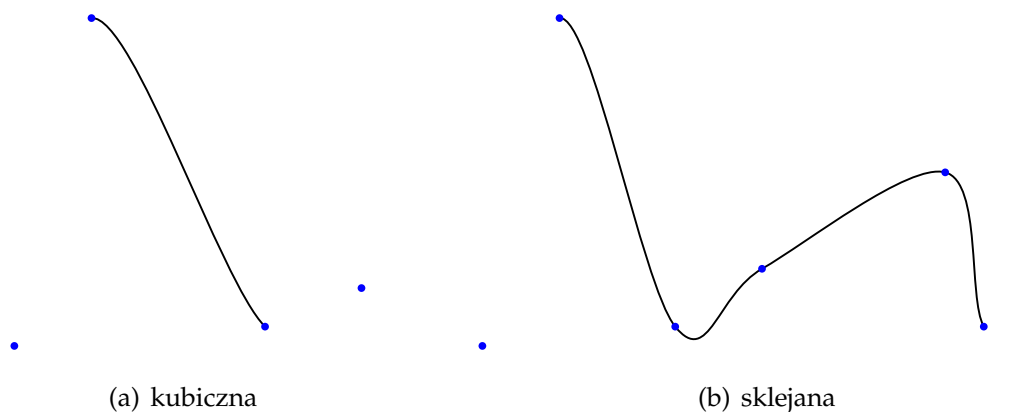
Dane: $P_0^0, P_1^0, P_2^0, P_3^0$ oraz $t \in [0, 1]$

Wynik: punkt na krzywej

for $i = 1, \dots, 3$ **do**

for $j = 0, \dots, 3 - i$ **do**
 $P_j^i = \alpha_j^i(t)P_j^{i-1} + (1 - \alpha_j^i(t))P_{j+1}^{i-1}$

return $C(t) = P_0^3$



Rysunek 3. Przykładowe krzywe Catmulla-Roma

Literatura

- [1] Agoston, M.K.: Computer Graphics and Geometric Modeling: Implementation and Algorithms. Springer, London, (2005)
- [2] Farin, G.: Curves and Surfaces for CAGD: A Practical Guide, 5th Edition. Morgan Kaufmann Publishers, San Francisco, (2002)
- [3] Koprowski, P.: Glimpses of geometry and graphics.
<http://www.pkoprowski.eu/ggg/index.html>
- [4] Salomon, D.: Curves and Surfaces for Computer Graphics. Springer, New York, (2006)

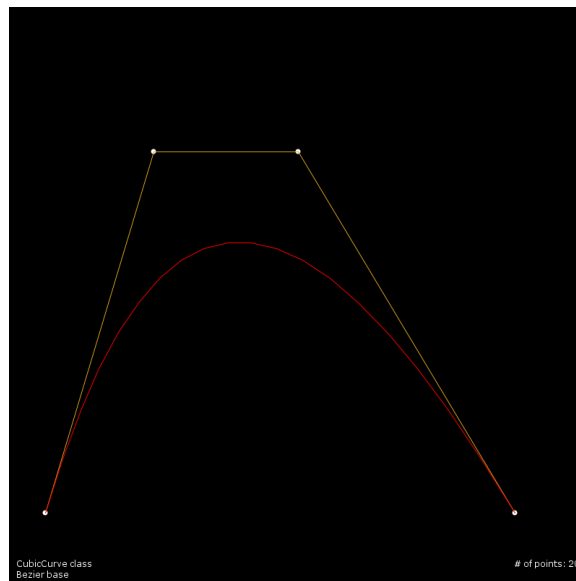
Zadanie

Napisać aplikację prezentującą różne krzywe kubiczne. Aplikacja powinna posiadać następujące funkcjonalności:

- określenie liczby punktów aproksymujących krzywą,
- rysowanie krzywej kubicznej o zadanych punktach kontrolnych,
- rysowanie punktów kontrolnych krzywej,
- włączenie/wyłączenie rysowania krzywej,
- włączenie/wyłączenie rysowania punktów kontrolnych,
- zmiana położenia punktów kontrolnych za pomocą myszki,
- wybór typu krzywej kubicznej: Béziera, Catmulla–Roma, kardynalna, b-splain,
- w przypadku krzywej kardynalnej mamy mieć możliwość podania wartości α ,
- w przypadku krzywej Béziera zaimplementować rysowanie z wykorzystaniem wzoru (1) oraz algorytmu de Casteljau,

- w przypadku krzywej Catmulla–Roma zaimplementować rysowanie z wykorzystaniem wzoru (1) oraz algorytmu Barry’ego–Goldmana.

Przykładowy wygląd obszaru renderingu aplikacji przedstawia Rysunek 4.



Rysunek 4. Zawartość okna po uruchomieniu aplikacji