Programowanie animacji

Zestaw 2

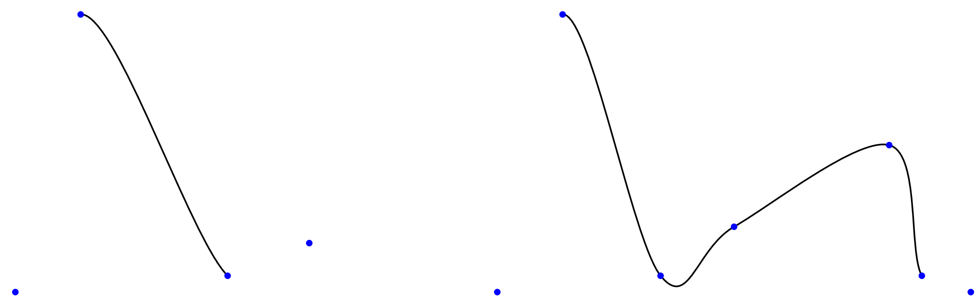
1. Napisać klasę reprezentującą sklejaną kubiczną krzywą Catmulla-Roma. W klasie mają znaleźć się odpowiednie konstruktory, metody rysujące krzywą (metoda korzystająca ze wzoru (1) oraz metoda korzystająca z algorytmu Barry'ego-Goldmana).

   Dla przypomnienia kubiczna krzywa Catmulla-Roma dana jest przez cztery punkty kontrolne $P_0, P_1, P_2, P_3 \in \mathbb{R}^3$, a jej wzór jest następujący:

$$P(t) = (P_0, P_1, P_2, P_3) \cdot \begin{bmatrix} -1/2 & 1 & -1/2 & 0 \\ 3/2 & -5/2 & 0 & 1 \\ -3/2 & 2 & 1/2 & 0 \\ 1/2 & -1/2 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} t^3 \\ t^2 \\ t \\ 1 \end{pmatrix}, \tag{1}$$

   gdzie $t \in [0, 1]$.

   Krzywa Catmulla-Roma interpoluje punkty kontrolne P_1 i P_2 oraz jej styczna w punkcie P_1 jest połową wektora $\overrightarrow{P_0P_2}$ zaś styczna w P_2 połową wektora $\overrightarrow{P_1P_3}$. Kolejne dwa segmenty sklejanej kubicznej krzywej Catmulla-Roma współdzielą trzy punkty kontrolne oraz krzywa taka jest gładka (jest klasy C^1).



(a) kubiczna (b) sklejana

Rysunek 1: Przykładowe krzywe Catmulla-Roma.

   Do wyznaczania punktu na krzywej można korzystać ze wzoru (1), ale istnieje też inny algorytm. Algorytm ten nosi nazwę algorytmu Barry'ego-Goldmana (Algorytm 1). W algorytmie tym współczynnik α_j^i określony jest następująco:

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

2. Napisać program, w którym po ustalonej krzywej sklejanej będzie poruszać się obiekt. Należy zadbać o to, aby obiekt był dobrze zorientowany podczas ruchu po tej krzywej.

   Dla przypomnienia w celu zorientowania obiektu potrzebujemy wyznaczyć lokalny układ współrzędnych na krzywej, który definiuje nam macierz rotacji.

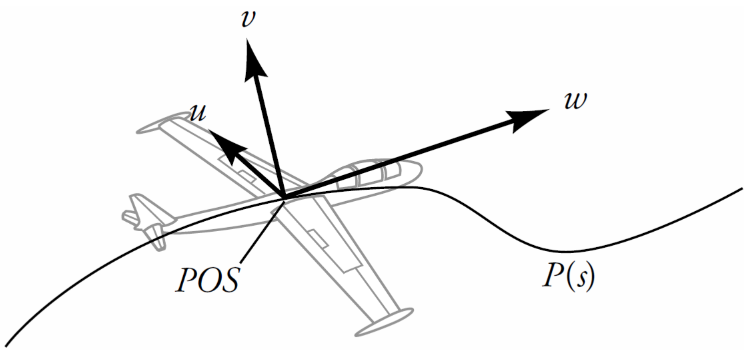
Algorytm 1: Algorytm Barry'ego-Goldmana znajdowania punktu na kubicznej krzywej Catmulla-Roma

Dane: $P_0^0, P_1^0, P_2^0, P_3^0$ oraz $t \in [0, 1]$.
Wynik: Punkt na krzywej.

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

return $C(t) = P_0^3$

- W przypadku 2D układ wyznaczony może być jako wektor styczny i normalny krzywej. Jeśli wektor $(x, y)^T$ jest wektorem stycznym, to wektor normalny wynosi $(y, -x)^T$.
- W przypadku 3D możemy skorzystać z układu Freneta (trójścianu Freneta).



Wektory u, v, w możemy obliczyć następująco:

$$w = \frac{P'(s)}{\|P'(s)\|}, \tag{3}$$

$$u = \frac{P''(s) \times P'(s)}{\|P''(s) \times P'(s)\|}, \tag{4}$$

$$v = u \times w. \tag{5}$$