



Computergrafik (SS 2018)

Übung 9

fällig Montag 18. Juni, 14:20

- Geben Sie bei jeder Abgabe alle Ihre Matrikelnummern auf jedem Blatt an.
- Verspätet eingereichte Abgaben können nicht gewertet werden.

Aufgabe 1 (Theorie: Bézierkurven)

Eine kubische Bézierkurve C in 2D sei definiert durch die Kontrollpunkte $b_0 = (0, 0)$, $b_1 = (1, 0)$, $b_2 = (1, 1)$, $b_3 = (2, 1)$.

- (a) (1P) Zeichnen Sie die Kontrollpunkte und das Kontrollpolygon von C .
- (b) (2P) Werten Sie C mittels de Casteljau-Algorithmus *grafisch* (d.h. ohne Rechnung) an der Stelle $t = \frac{1}{4}$ aus.
- (c) (2P) Werten Sie C mittels de Casteljau-Algorithmus *grafisch* (d.h. ohne Rechnung) an der Stelle $t = \frac{1}{2}$ aus.
- (d) (2P) Werten Sie C mittels de Casteljau-Algorithmus *rechnerisch* an der Stelle $t = \frac{3}{4}$ aus.
- (e) (1P) Geben Sie den Richtungsvektor (Vorzeichen egal) der Tangente an C an der Stelle $t = 1$ an. Begründen Sie Ihre Angabe geometrisch, ohne Rechnung.
- (f) (1P) Eine lineare Bézierkurve sei gegeben durch $b_0 = (0, 0)$, $b_1 = (3, 3)$. Werten Sie diese Kurve mittels de Casteljau-Algorithmus rechnerisch an der Stelle $\frac{1}{3}$ aus.
- (g) (2P) Eine Bézierfläche, welche in beide Parameterrichtungen linear ist, sei gegeben durch $b_{00} = (0, 0, 0)$, $b_{01} = (0, 1, 0)$, $b_{10} = (1, 0, 0)$, $b_{11} = (1, 1, 2)$. Werten Sie diese Fläche rechnerisch an der Stelle $(\frac{1}{4}, \frac{1}{2})$ aus.



Aufgabe 2 (Praxis: Bézierkurven)

Schauen Sie sich den Code für Übungsblatt 10 an.

- (a) (5P) Die Klasse `BezierCurve` implementiert eine kubische Bézierkurve. Sie besitzt dazu vier Punkte `p0`, `p1`, `p2`, `p3`, die die Kontrollpunkte der Kurve angeben.

Ergänzen Sie diese Klasse um eine Methode `drawFrame`, welche einen einzigen Parameter `t` besitzt, d.h. `this.drawFrame = function (t) { ... }`.

Diese Methode soll die Kontrollpunkte und sämtliche Zwischenpunkte b_i^k , k von 0 bis n , die der de Casteljau-Algorithmus für den Wert t berechnet, berechnen und zeichnen. Die üblichen Verbindungslinien dieser Zwischenpunkte sollen ebenfalls gezeichnet werden. Sie können beliebig viele weitere Hilfsmethoden hinzufügen.

Zum Verwalten von Punkten können Sie die Klasse `Point3D` verwenden. Punkte können mit der Funktion `drawPoint` gezeichnet werden.

Zum Verwalten von Linien können Sie die Klasse `Line3D` verwenden. Linien können mit der Funktion `drawLine` gezeichnet werden.

- (b) (4P) Ergänzen Sie die Klasse `BezierCurve` um Methoden, die die eigentliche Bézierkurve zeichnen.

Die Hauptmethode, die durch andere Programmteile aufgerufen wird, muss den Namen `drawCurve` tragen und keinen Parameter erwarten.

Das Zeichnen der Kurve soll durch Auswertung der Kurve für viele t -Werte und Zeichnen von Punkten an den Ergebnisstellen geschehen. Iterieren Sie dazu über t ; eine geeignete Schrittweite ist z.B. 0,002.

Bei korrekter Implementierung können die Kontrollpunkte mit der Maus verschoben und der t -Wert mit dem Slider eingestellt werden.

Aufgabe 3 (Bonus: Bézierflächen)

Ergänzen Sie den Code so, dass eine Bézierfläche angezeigt werden kann.

Erweitern Sie dazu die Funktion `updateCanvas`. Ein Array von vier Bézierkurven wird Ihnen in `curvesTrans` zur Verfügung gestellt.

Zum Anzeigen der Fläche aktivieren Sie die Checkbox `Bonusaufgabe`.

Um die Fläche per Maus zu rotieren (Virtual Trackball) deaktivieren Sie die Checkbox `Verschiebe Punkte`.