BERGISCHE UNIVERSITÄT WUPPERTAL
FAKULTÄT FÜR MATHEMATIK
UND NATURWISSENSCHAFTEN

BERGISCHE
UNIVERSITÄT
WUPPERTAL

Angewandte Informatik
Dr. Martin Galgon
M.Sc. Jose Jimenez

Bildgenerierung

Wintersemester 2023 / 2024

Übungsblatt 6

Aufgabe 15 *(Painter's Algorithm)*

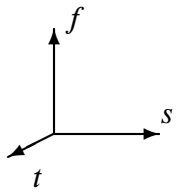
Der *Painter's Algorithm* ist ein besonders einfacher Algorithmus zur Elimination verdeckter Kanten und Flächen. Er ist anwendbar, wenn die Objekte sich nicht gegenseitig durchdringen, und basiert auf folgender Strategie: Man zeichnet einfach *alle* Objekte (Polygone müssen dabei ausgefüllt werden), wobei *„von hinten nach vorne"* gearbeitet wird. Dadurch werden die durch weiter vorne liegende Objekte verdeckten (und damit unsichtbaren) Teile der hinteren Objekte automatisch „übermalt".

Stellen Sie mit Hilfe des Painter's Algorithm den Graphen der Funktion

$$f(s,t) = 10e^{-\left(\frac{s^2}{25}+\frac{t^2}{100}\right)}$$

auf dem Bereich $(s,t) \in [-5;5] \times [-5;5]$ dar.

Approximieren Sie dabei den Graphen durch ein geeignetes Netz von Dreiecken und verwenden Sie zur Darstellung die sogenannte *Kabinett-Projektion*.



Dabei handelt es sich um eine Parallelprojektion, wobei die s- und die f-Achse waagerecht bzw. senkrecht dargestellt werden und die t-Achse um 30° geneigt und um den Faktor ½ verkürzt ist.

Verwenden Sie das Rahmenprogramm `painters.cc` unter `/home/bildgen/Aufgaben/painters` und implementieren Sie die Funktionen:

```
Matrix4x4 berechneMpar(double umin, double umax, double vmin, double vmax,
                       double& ratio)
void erzeugeFlaeche(double xmin, double xmax, double zmin, double zmax, int num,
                    const std::function<double(double, double)>& func,
                    std::vector<Dreieck>& dreiecke )
```

Aufgabe 16 *(Silhouetten-Algorithmus)*

Implementieren Sie den Silhouetten-Algorithmus und stellen Sie wie in Aufgabe 15 mit dessen Hilfe den Graphen der Funktion f dar.

Verwenden Sie die gleiche Parallelprojektion wie in Aufgabe 15.

Das Rahmenprogramm `silhouetten.cc` im Verzeichnis `/home/bildgen/Aufgaben/silhouetten` dient als Grundlage. Implementieren Sie darin die Funktionen

```
Matrix4x4 berechneMpar(double umin, double umax, double vmin, double vmax,
                       double& ratio)
void erzeugeKurven(double xmin, double xmax, double zmin, double zmax,
                   int num, int pieces,
                   const std::function<double(double,double)>& func,
                   std::vector<std::vector<Vec3D>>& kurven )
void maleSilhouetten(Drawing& pic,
                     const std::vector<std::vector<Vec3D>>& kurven,
                     const Matrix4x4& mpar, double ratio)
```

bzw. die fehlenden Teile der Funktionen. Weitere Informationen zu den Ein- und Ausgabeparametern finden Sie im Rahmenprogramm.

Aufgabe 17 *(z-Buffer-Verfahren)*

In der Datei `proj3.cc` unter `/home/bildgen/Aufgaben/projektion-3` finden Sie ein fast fertiges Programm, das die Lösung zu Aufgabe 10 um das z-Buffer-Verfahren erweitert. Ergänzen Sie die folgenden Funktionen:

- `clip3DPoint`: Im Gegensatz zu Aufgabe 13 wird hier keine Linie gekappt, sondern lediglich getestet, ob ein gegebener Punkt im kanonischen Bildraum liegt.
- `drawPointZ`: Malt einzelne Pixel und aktualisiert den z-Buffer.

Abgabe: Mi., 06.12.2023, 13:15 Uhr

Senden Sie Ihre Lösungen der Theorie-Aufgaben und Ihre Programme per E-Mail an `bildgen@studs.math.uni-wuppertal.de` .