Computergrafik
Computer Aided Geometric Design
Dokumentation

Fachhochschule Bielefeld
Campus Minden
Studiengang Informatik

Beteiligte Personen:

Name	Matrikelnummer
Jan-Hendrik Sünderkamp	1153536
Peter Dick	1050185
Alexander Jaksties	1077474
Ruben Meinhardt	1082430

---

22. Juli 2018

Inhaltsverzeichnis

1 Aufgabenstellung

Im Verlauf des Projektes sollten folgende Aufgaben gelöst werden:

1. Einlesen von Objekten im .obj-Format und entsprechende Visualisierung
2. Verschieben von einem oder mehreren Punkten des Kontrollnetzes
3. Setzen von Vertice-Gewichten und scharfer Kanten
4. Implementierung der Catmull-Clark Unterteilung für offene und geschlossene Meshes
5. Visualisierung unterteilter Meshes beliebiger Tiefe
6. Berechnung von Catmull-Clark Limitpunkten und -normalen, sowie deren Visualisierung
7. Konsistenzprüfung der Halfedge-Datenstruktur, sowie Ausgabe einer Statistik
8. Implementierung einer Punktglättung
9. Phong Shading und Phong Beleuchtung

2 Implementierte Funktionen

Folgende Funktionen wurden gemäß der Aufgabenstellung implementiert:

1. .obj-Objekte können eingelesen werden. Diese werden von der Anwendung direkt angezeigt.
2. Einzelne oder mehrere Punkte gruppiert sind verschiebbar.
3. Es können beliebige Vertice-Gewichte >0 auf zwei Nachkommastellen genau gesetzt werden. Diese fließen in die Berechnung der Unterteilungsflächen mit ein. Des Weiteren ist das setzen scharfer Kanten möglich, die vom Catmull-Clark-Algorithmus auch als solche behandelt werden.
4. Sowohl offene, wie auch geschlossene Meshes sind mit Catmull-Clark unterteilbar.
5. Die verschiedenen Catmull-Clark-Iterationen können über einen Slider angezeigt werden.
6. Optional können zu jedem unterteilten Mesh die Limitpunkte angezeigt werden.
7. Es kann die Konsistenz eines Meshes überprüft werden. Zusätzlich dazu wird eine Statistik ausgegeben, die Informationen über die Halfedge-Datenstruktur liefert.
8. Abschließend kann mithilfe eines Sliders eine Punktglättung durchgeführt werden. Der Slider bestimmt dabei das Maß der Glättung.

Zusätzlich wurden folgende weitere Funktionen implementiert:

1. Einzelne Punkte, Kanten oder Faces können aus Meshes gelöscht werden. Dabei entstehen Hole-Faces.
2. Punkte können gezielt an einer bestimmten Achse verschoben werden.
3. Meshes sind aus der Anwendung exportierbar.
4. Punkte können als “spitz“ markiert werden und erhalten dann beim Unterteilen ihre Position.

3 Architektur

3.1 Vorgaben

Es galt eine Architektur aufzubauen, die auf einer HalfEdge-Datenstruktur basiert. Das heißt insbesondere:

- Jeder Punkt kennt genau *eine* seiner ausgehenden und *keine* seiner eingehenden Kanten.
- Jede Kante kennt nur ihren Ausgangspunkt, ihre Gegenkante, ihre nachfolgende Kante und die Fläche, an der sie liegt.
- Jede Fläche kennt genau *eine* ihrer umschließenden Kanten.
- Bei einem zusammenhängenden Mesh ist jeder Punkt von jedem anderen über die Kanten erreichbar.
- Kanten, die am Rand des Meshes liegen, kennen eine Fläche, die man als “Hole“ bezeichnet.

In den folgenden Abschnitten wird die erstellte Architektur dargestellt und kurz erklärt.

3.2 Architektur

Die Architektur des Projektes teilt sich im wesentlichen in ein Model- und ein View-Paket. Während das Model eine etwas erweiterte HalfEdge-Struktur bereitstellt, ist das View-Paket im wesentlichen für die Anzeige zuständig.

Im folgenden soll zuerst das Model-Paket und anschließend das View-Paket vorgestellt werden.

3.2.1 Model

Wie bereits erwähnt stellt das Model in Abb. 1 unter anderem die HalfEdge-Struktur bereit. Hierzu dienen die Strukturen graphicVertex, halfEdge und graphicFace, welche alle von der abstrakten GraphicObject-Klasse erben. Dies erleichtert später die Verknüpfung der verschiedenen Level of Detail (LOD) nach Ausführungen von Unterteilungsalgorithmen. Zur einfacheren Datenhaltung werden die genannten Strukturen als HalfEdgeMesh-Klasse gespeichert, die alle Punkte, Kanten und Flächen eines Objektes enthält. Des Weiteren kennt das Mesh sein LOD. Die genannten Objekte werden über die ObjectLoader-Klasse

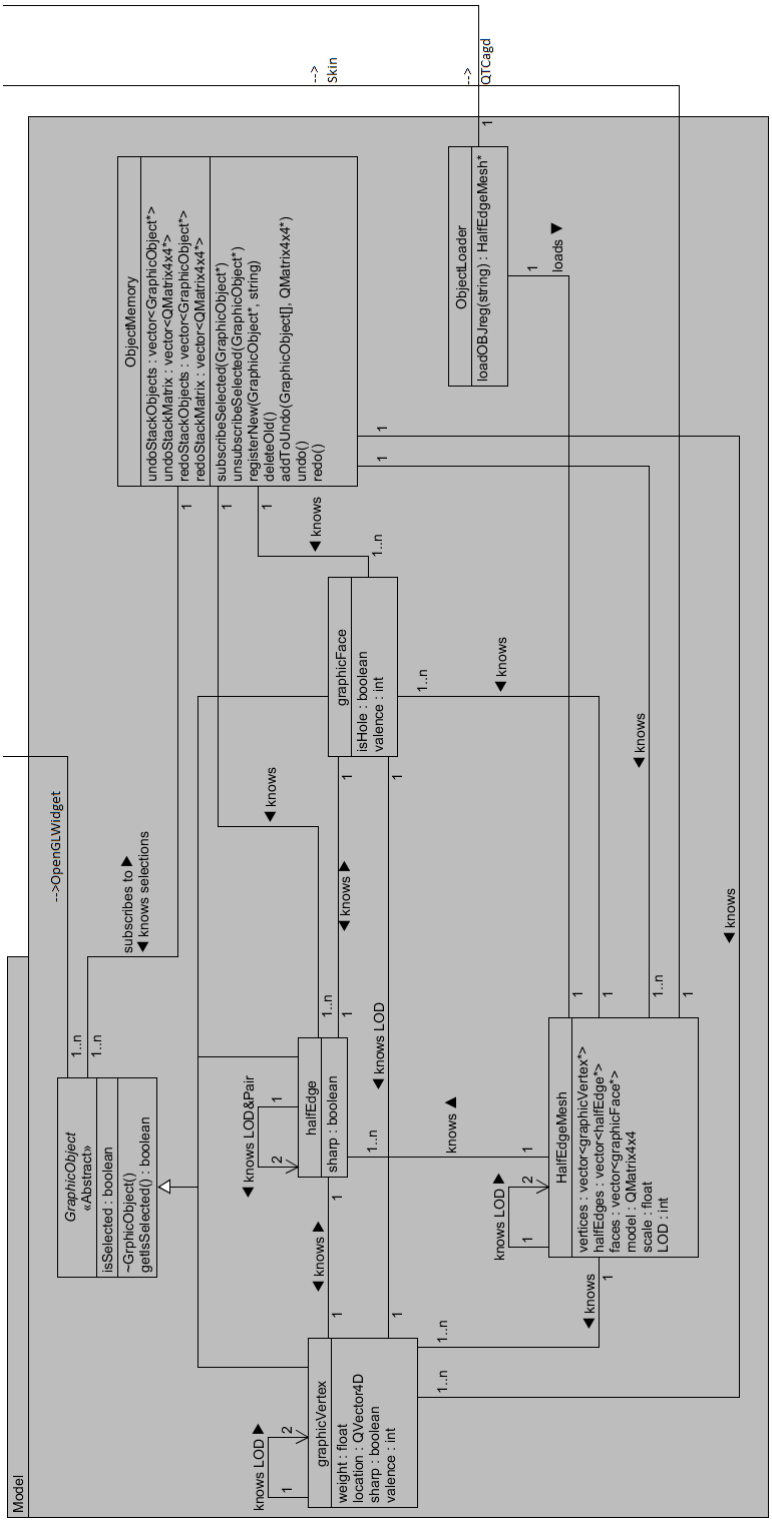

Abbildung 1: Model-Paket der Architektur

geladen.

Die ObjectMemory-Klasse ist zurzeit noch ungenutzt. Mit Verweis auf Kapitel 6 sollte an dieser Stelle z.B. eine Undo-Redo-Funktion realisiert werden. Dies wurde aufgrund von Zeitmangel jedoch nicht umgesetzt.

3.2.2 View

Um den Aufwand für UI-Programmierung zu reduzieren, haben wir uns des QT-Frameworks bedient. Dementsprechend besteht die View zum größten Teil aus externen QT-Abhängigkeiten. Das Hauptfenster der Anwendung wird in QTCagd realisiert, welche eine mit dem QT-Designer erstellte QTCagd.ui einbindet und anzeigt (siehe Abb. 2). Des Weiteren werden dort entsprechende Signale für UI-Eingaben und Auswahlen gesendet.

Das OpenGLWidget hingegen ist ein Teil des Hauptfensters und kümmert sich um sämtliche OpenGL-spezifischen Anzeigen und Operationen. Dies umfasst z.B. das Anzeigen von Vertices, Edges und Faces.

Das Skin-Paket umfasst Objekte, die zum Anzeigen der internen GraficObjects genutzt werden. So existiert z.B ein Sphere-Model, das genutzt wird, um dem Nutzer die Auswahl von Vertices zu erleichtern. Der ContextSensitiveEditView ist das Menü auf der rechten Seite in Abb. 3. Dieses Menü ändert sich basierend auf der momentanen Auswahl.

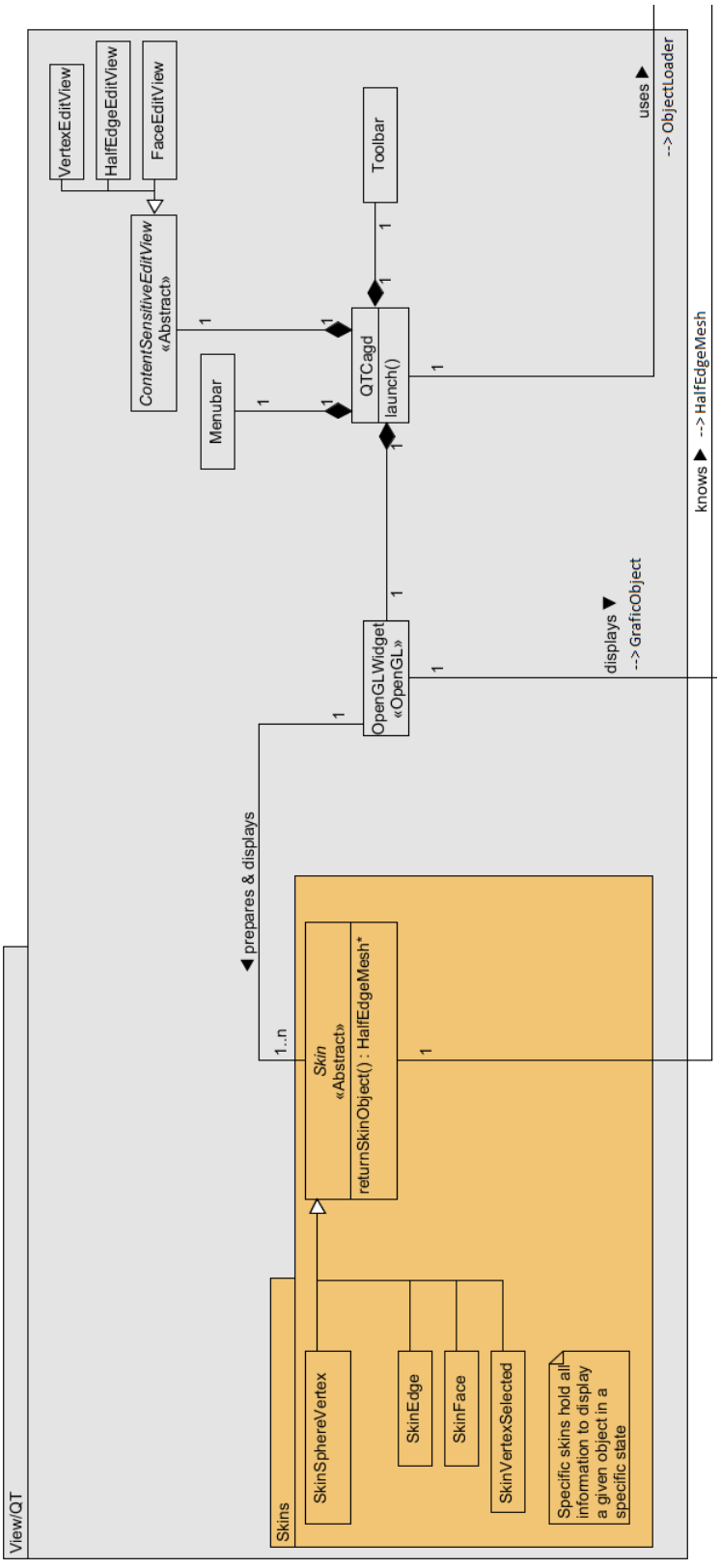

Abbildung 2: View-Paket der Architektur

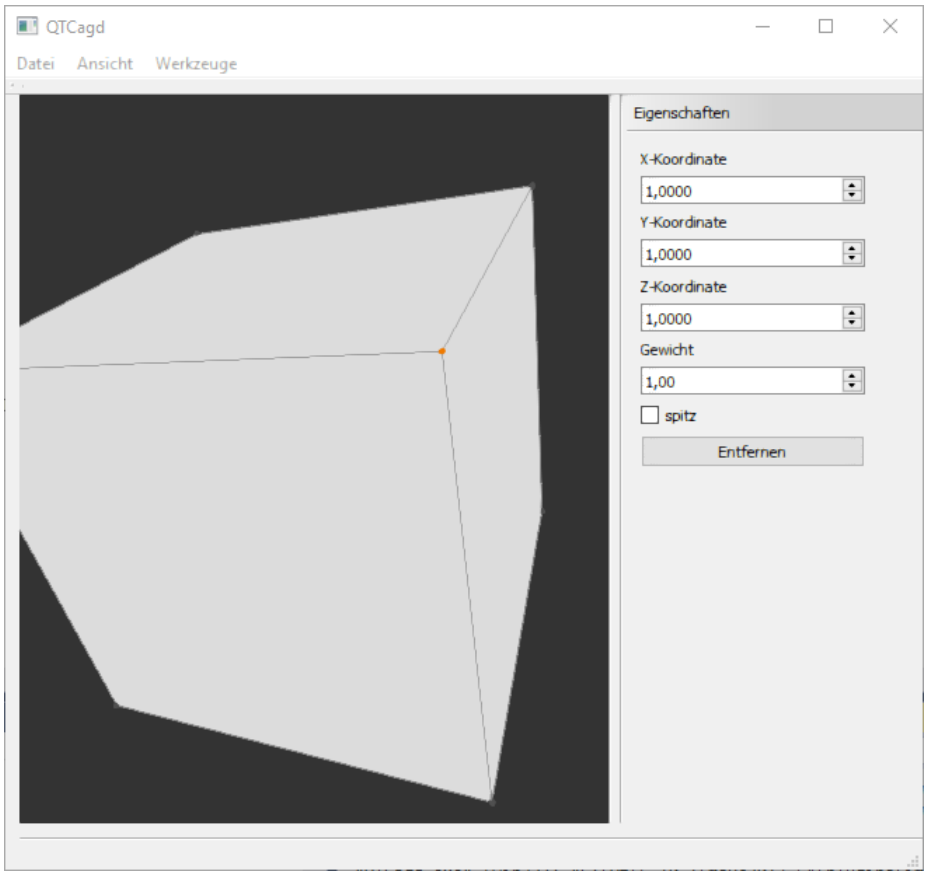

Abbildung 3: UI der Anwendung

4 Erklärung der Algorithmen

4.1 Catmull-Clark-Unterteilung

Im Zuge des Projektes wurde der Catmull-Clark-Algorithmus für Unterteilungsflächen umgesetzt. Darauf soll in den folgenden Unterkapiteln genauer eingangen werden. Zuerst wird der allgemeine Fall betrachtet. Anschließend die Regelungen für scharfe Kanten und spitze Punkte. Abschließend wird noch kurz auf die Limitpunkte eingegangen.

4.1.1 Allgemeiner Algorithmus

Bei dem Algorithmus werden die Flächen des Meshes unterteilt. Dabei entsteht für jeden existierenden Punkt, jede Kante und jede Fläche ein neuer Punkt. Dies funktioniert für beliebige Topologien. Nach der ersten Iteration des Algorithmus gibt es im Mesh nur noch Flächen mit Valenz vier. Des Weiteren enstehen nach der ersten Iteration nurnoch Punkte mit Valenz vier. Die Anzahl der irregulären Punkte *nach* der ersten Iteration bleibt gleich.

Die neuen Punkte werden nach folgenden Regeln gebildet:

- Ein neuer Flächenpunkt errechnet sich aus dem Mittel der die Fläche umschließenden

Punkte.

- Ein neuer Kantenpunkt wird aus dem Mittel der neuen Flächenpunkten und der angrenzenden Punkte der Kante gebildet.
- Neue Eckpunkte werden aus den neuen, angrenzenden Flächenpunkten, den inzidenten Punkten und dem alten Eckpunkt errechnet.

Im folgenden wird auf die genauen Berechnungen eingangen.

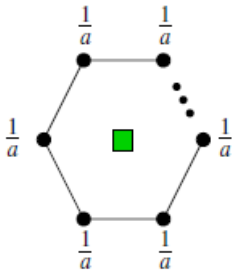

Abbildung 4: Berechnung der Flächenpunkte

Flächenpunkte In Abb. 4 wird die Berechnung der neuen Flächenpunkte(grün) dargestellt. Die Variable a bezeichnet die Flächenvalenz der jeweiligen Fläche.

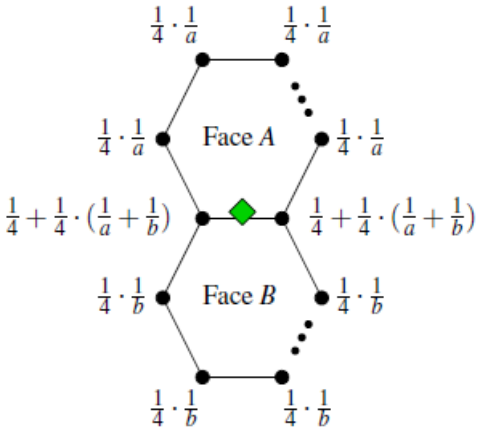

Abbildung 5: Berechnung der Kantenpunkte

Kantenpunkte Die Kantenpunkte(grün) werden wie in Abb. 5 ermittelt. Die Variable a bezeichnet die Flächenvalenz der Fläche A und die Variable b entsprechend für Fläche B. Zu beachten ist hier der Spezialfall in Kap. 4.1.2.

Eckpunkte Die neuen Eckpunkte ergeben sich durch Anteile der neuen Flächenpunkte und der alten, inzidenten Eckpunkte, wie in Abb. 6 dargestellt. Hierbei bezeichnet k die Punktvalenz.

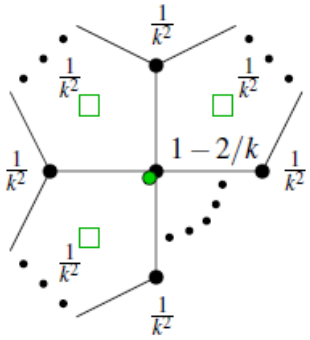

Abbildung 6: Berechnung der Eckpunkte

4.1.2 Scharfe Kanten und spitze Punkte

Eine scharfe Kante verwendet gesonderte Regeln um beim Unterteilen andere Ergebnisse zu erzielen. Dies begründet sich darin, dass bei der Berechnung nur die inzidenten Eckpunkte betrachtet werden. Es entstehen sichtbare Knicke im Mesh.

Ein spezieller Einsatzbereich der scharfen Kanten ist die Umrandung von Löchern im Mesh. Der Einsatz scharfer Kanten verhindert hierbei ein Anwachsen der Lochgröße. Die Kantenpunkte und Eckpunkte, die auf einer scharfen Kante liegen, errechnen sich mit den in Abb. 7 gezeigten Verhältnissen.

Ein spitzer Punkt erhält bei der Unterteilung durch Catmull-Clark seine Ursprungsposition.

4.1.3 Limitpunkte

Durch den Einsatz des Catmull-Clark Algorithmus schrumpft das unterteilte Mesh. Nach unendlich vielen Iterationen konvergiert das Mesh gegen eine Limitfläche. Für einen Punkt des Meshes lässt sich sein zugehöriger Punkt auf der Limitfläche bestimmen, die dazu benötigten Regeln heißen Limitpunktregeln.

Abb. 8 zeigt einen Ausschnitt aus einem Viereckmesh. Die Variablen α , β und γ beschreiben den Eckpunkt, seine inzidenten Eckpunkte und die Eckpunkte in Zweier-Nachbarschaft.

In den Limitpunkt gehen die genannten Komponenten zu folgenden Verhältnissen ein. Hierbei bezeichnet k die Punktvalenz und i den i-ten inzidenten Punkt bzw. Punkt in

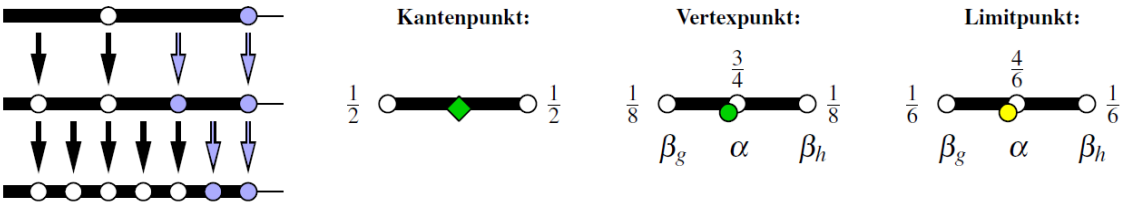

Abbildung 7: Berechnung der scharfen Kanten mit Limitpunkten

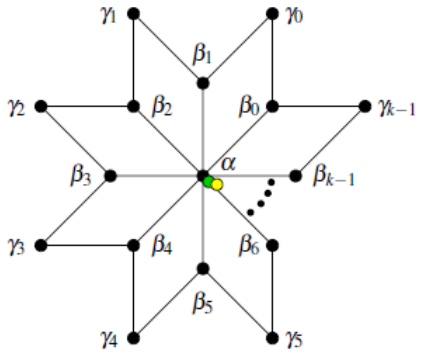

Abbildung 8: Berechnung der Limitpunkte

Zweier-Nachbarschaft.

$$\alpha = 1 - \frac{5}{k+5} \qquad \beta_i = \frac{4}{(k+5)k} \qquad \gamma_i = \frac{1}{(k+5)k} \tag{1}$$

An scharfen Kanten gelten für die Berechnung von Limitpunkten die in Abb. 7 gezeigten Verhältnisse.

4.2 Punktglättung

Bei der Punktglättung werden alle Punkten, der zu einem Punkt inzidenten Flächen, mit dem Ursprungspunkt addiert. Aus dem Mittel ergibt sich der geglättete Punkt. Zwischen dem Ursprungspunkt und dem geglätteten Punkt kann man nun per Interpolation eine beliebige Glättungsstufe erhalten.

4.3 Löschen von Punkten, Kanten und Flächen

Im folgenden wird die Funktion, Punkte, Kanten und Flächen zu löschen näher betrachtet. Insbesondere wird kurz auf Besonderheiten bei den einzelnen Verfahren hingewiesen.

4.3.1 Löschen von Punkten

Das Löschen von Punkten ist der schwierigste Fall beim Löschen. Hierbei müssen einige Schritte unternommen werden und auf einige Spezialfälle geachtet werden. Allgemein betrachtet muss folgendes geschehen:

- Der ausgewählte Punkt muss entfernt werden.
- Alle inzidenten Kanten müssen entfernt werden.
- Alle inzidenten Flächen müssen entfernt werden.
- Die inzidenten Eckpunkte zeigen nicht mehr auf die entfernten Kanten.
- Die Valenz der inzidenten Eckpunkte muss angepasst werden.
- An der entsprechenden Stelle entstehen ein oder mehrere Löcher.

- Die HalfEdge-Datenstruktur an der entsprechenden Stelle muss aktualisiert werden. Dabei muss die Konsistenz trotz entfernter Kanten und eines oder mehrerer neuer Löcher erhalten bleiben.

4.3.2 Löschen von Kanten

Das Löschen von Kanten ist dem Löschen eines Punktes recht ähnlich.

- Die entsprechende Kante (und damit die unterliegenden Halfedges) müssen entfernt werden.
- Die inzidenten Flächen werden entfernt und durch ein oder mehrere Löcher ersetzt.
- Die Valenz der inzidenten Punkte muss entsprechend angepasst werden.
- Die HalfEdge-Datenstruktur muss wie oben aktualisiert werden, um die Konsistenz zu erhalten.

4.3.3 Löschen von Flächen

Das Löschen einer Fläche ist der einfachste Fall. Hierbei wird lediglich die Fläche durch ein Loch ersetzt und die HalfEdge-Datenstruktur mit scharfen Kanten versehen.

5 Probleme bei der Entwicklung

Bei der Arbeit am Projekt sind wir nur auf wenige größere Probleme gestoßen. Die meisten Probleme, die wir hatten, hatten mit der Fehlerfindung in Visual Studio und C++ zu tun. An vielen Stellen war es nicht offensichtlich, an welcher Stelle nun ein Fehler auftrat und warum.

Das Problem wurde insbesondere dadurch verstärkt, dass wir keinerlei Vorkenntnis mit QT hatten, welches wir für die UI nutzen wollten. Dies sollte uns einiges an Arbeit im Bereich der Oberflächengestaltung ersparen. Im Endeffekt haben wir vermutlich das gleiche an Einarbeitungszeit aufwenden müssen. Des Weiteren hatten wir einige Probleme mit dem QT Designer Tool, welches schnelles und einfaches erstellen von UIs ermöglichen soll. Insbesondere dessen Visual Studio Einbindung funktioniert nicht immer, wie wir es erwartet hätten.

Ein weiteres Problem war das Finden einer geeigneten und erweiterbaren Architektur für das Projekt. Insbesondere die Erweiterbarkeit war zu Anfang schwierig, da noch nicht alle Aufgaben und nötigen Funktionen bekannt waren. Dementsprechend durchlief die Architektur einige Iterationen.

Das letzte große Problem war die Performance. Mit wachsendem Funktionsumfang wurde das Projekt leider deutlich langsamer. Bei “größeren“ Meshes -viele Vertices, Edges und Faces- dauerten dann Catmull-Clark Iterationen eine Minute oder länger, bei einem Model mit 50.000 Vertices und 300.000 Halfedges dauert selbst das Laden des Objektes etwa 30 Sekunden.

6 Ausblick

Die Anwendung könnte auf mehrere Arten erweitert werden. Zum einen wäre es möglich weitere Unterteilungsverfahren zu implementieren, wie z.B. Loop-Subdivide oder Doo-Sabin. Andererseits wäre es natürlich auch möglich bestehende Funktionen auszubauen. So könnte auch eine einfache Rotation des Meshes statt der Kamera stattfinden. Des Weiteren wäre die bereits genannte Undo-Redo-Funktion eine sinnvolle Erweiterung für die Anwendung. Außerdem könnte man noch eine einzige Fläche zur weiteren Verarbeitung (verschieben, löschen, scharf setzen,...) unterteilen lassen, statt das ganze Mesh zu unterteilen. Eine abschließende Erweiterungsmöglichkeit wäre eine Extrude-Funktion, die es dem Nutzer erlaubt eine Fläche aus einer existierenden zu extrahieren, welche das Mesh fortsetzt.

Im Bereich der Optimierung liegt ganz klar die Performance im Fokus. Diese könnte vermutlich mit Techniken wie asynchroner Verarbeitung und Multithreading deutlich verbessert werden.

7 Anleitung

7.1 Objekt-Dateien einlesen

Unter „Datei->Öffnen“ kann eine Objekt-Datei geöffnet werden (siehe Abb. 9).

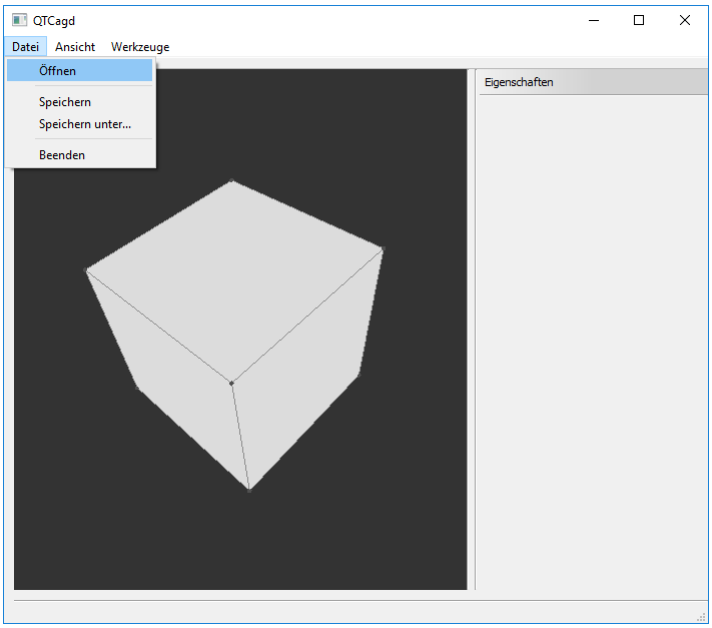

Abbildung 9

7.2 Objekt-Dateien speichern

Unter „Datei->Speichern“ oder mit „Datei->Speichern unter...“ kann eine Objekt-Datei gespeichert werden (siehe Abb. 10).

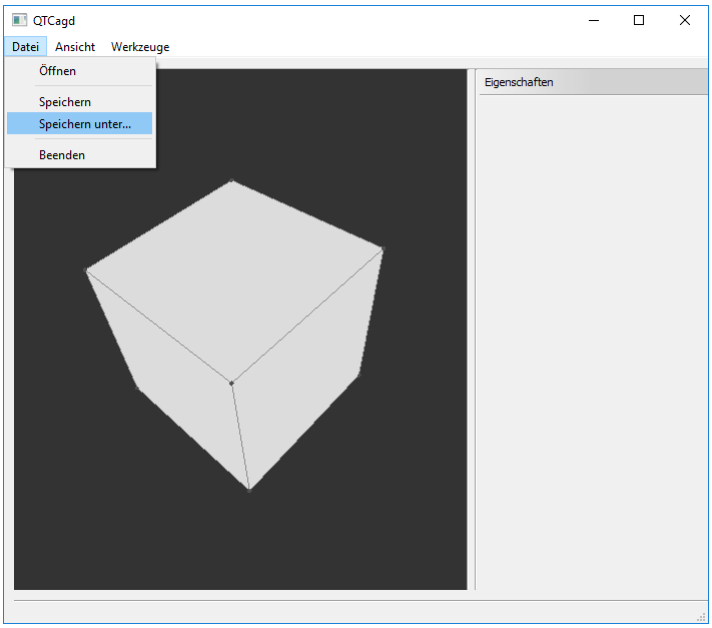

Abbildung 10

7.3 Ansichtsmodi

Es können 3 Ansichtsmodi aktiviert werden und zwar der „Vertex Mode“, der „Edge Mode“ oder der „Face Mode“ (siehe Abb. 11).
Punkte können im „Vertex Mode“ bearbeitet werden.
Kanten können im „Edge Mode“ bearbeitet werden.
Flächen können im „Face Mode“ bearbeitet werden.

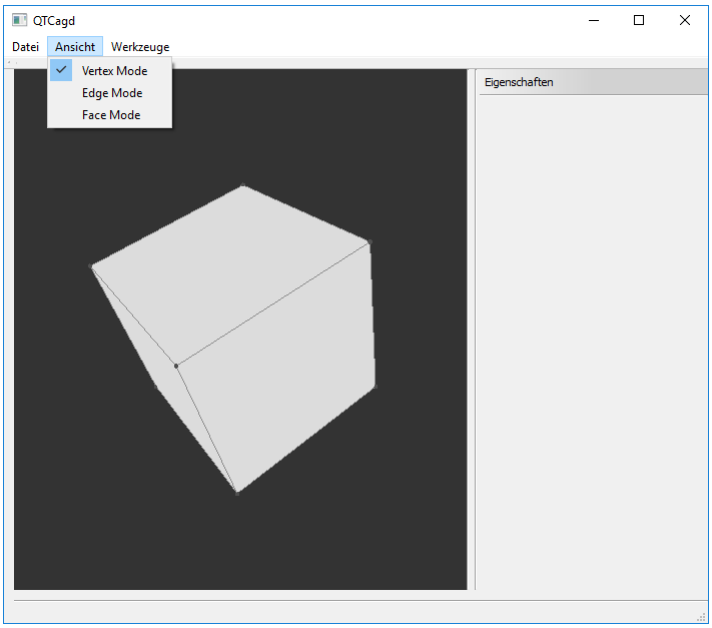

Abbildung 11

7.4 Objekte selektieren

Mit der Linken Maustaste können Grafikobjekte wie Punkte oder Kanten selektiert werden.
Mit Strg + Linke Maustaste kann eine Gruppe von Objekten selektiert werden.
Die Selektierten Objekte sind orange gefärbt (siehe Abb. 12).
Mit der Rechten Maustaste kann die Kamera um das Mesh gedreht werden.

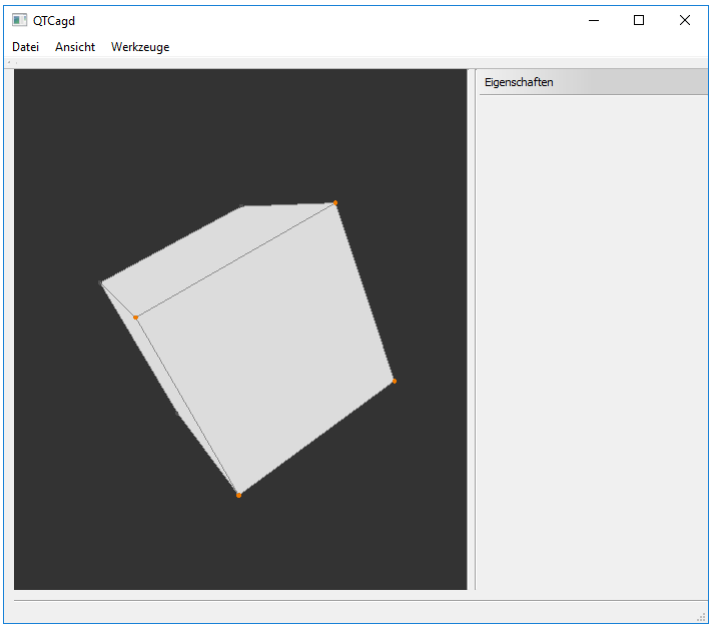

Abbildung 12

7.5 Punkte verschieben

Selektierte Punkte können mit gedrückter (linker) Maustaste und gleichzeitig gedrückter Alt-Taste verschoben werden.
Wird dabei die Taste einer Achse gedrückt (z.B. die X-Taste), so verschiebt sich die Selektion nur um die entsprechende Achse.
Ein selektierter Punkt kann auch mit den Spinboxen „X-Koordinate“, „Y-Koordinate“ und „Z-Koordinate“ im „Eigenschaften“ Menü verschoben werden (siehe Abb. 13).

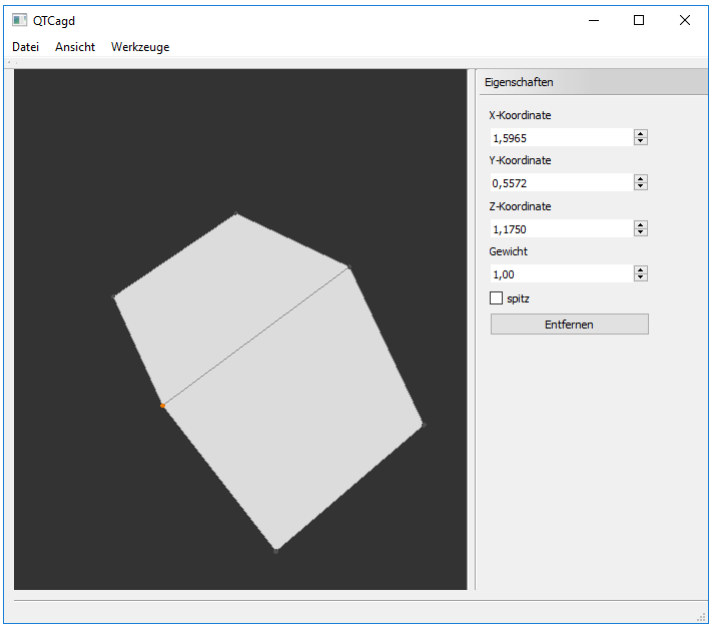

Abbildung 13

7.6 Objekte löschen

Selektierte Punkte, Kanten oder Flächen können mit der „Entf“-Taste gelöscht werden. Ein selektiertes Objekt kann auch mit den Button „Entfernen“ im „Eigenschaften“ Menü gelöscht werden (siehe Abb. 14).

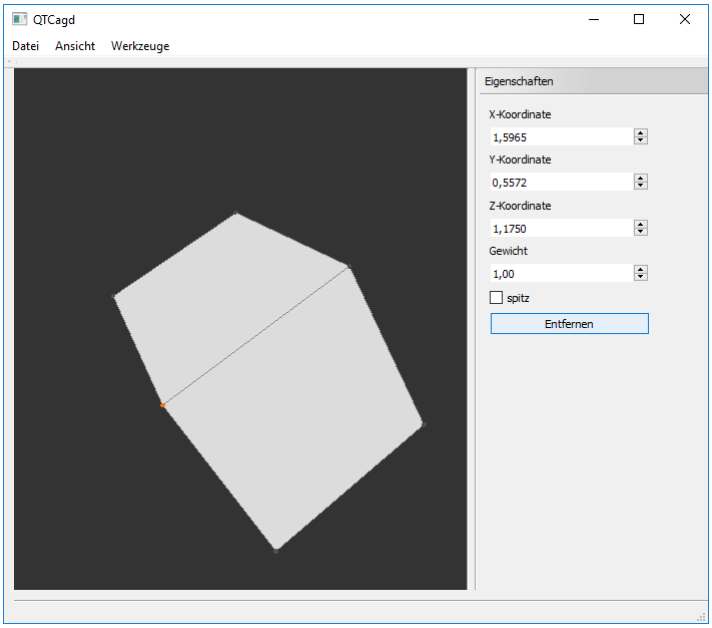

Abbildung 14

7.7 Punkte gewichten

Ein selektierter Punkt kann mit der Spinbox „Gewicht“ im „Eigenschaften“ Menü gewichtet werden (siehe Abb. 15).

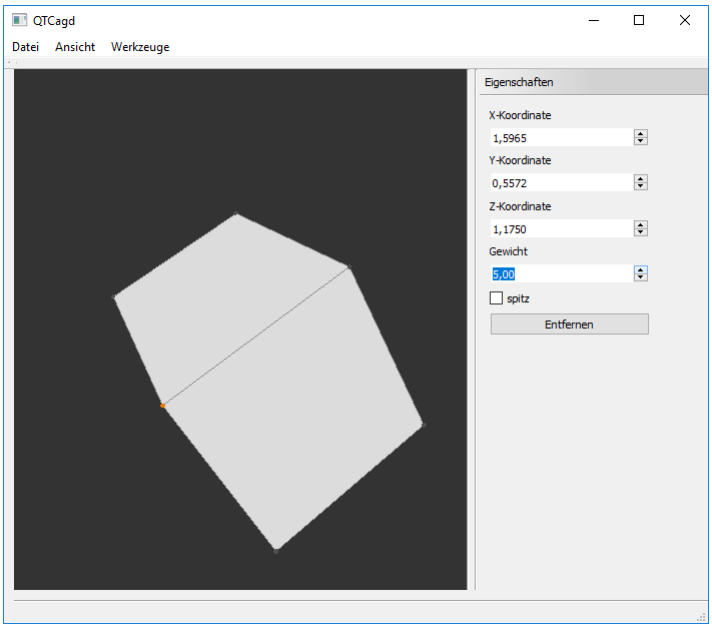

Abbildung 15

7.8 Punkte und Kanten scharf setzen

Ein selektierter Punkt oder eine selektierte Kante kann mit der Checkbox „spitz“ bzw. „scharf“ im „Eigenschaften“ Menü scharf gesetzt werden (siehe Abb. 16).
Die Taste „A“ setzt alle Kanten auf scharf.
Die Taste „D“ setzt alle Kanten auf nicht scharf.

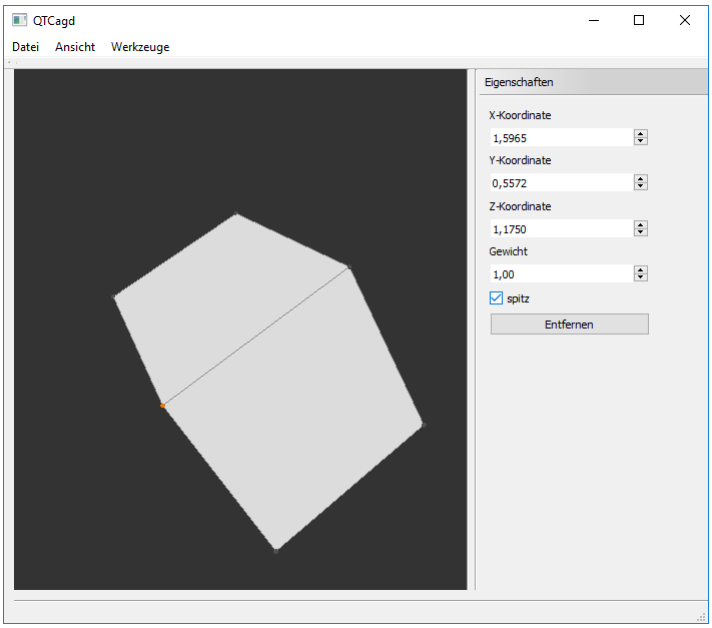

Abbildung 16

7.9 Mesh mit Catmull-Clark unterteilen

Unter „Werkzeuge->Catmull-Clark“ wird das „Catmull-Clark Subdivision“ Menü geöffnet. Der Slider „Level of Detail“ zeigt das Mesh in dem eingestellten Level of Detail an. Der Button „Anwenden“ unterteilt das Mesh (siehe Abb. 17).

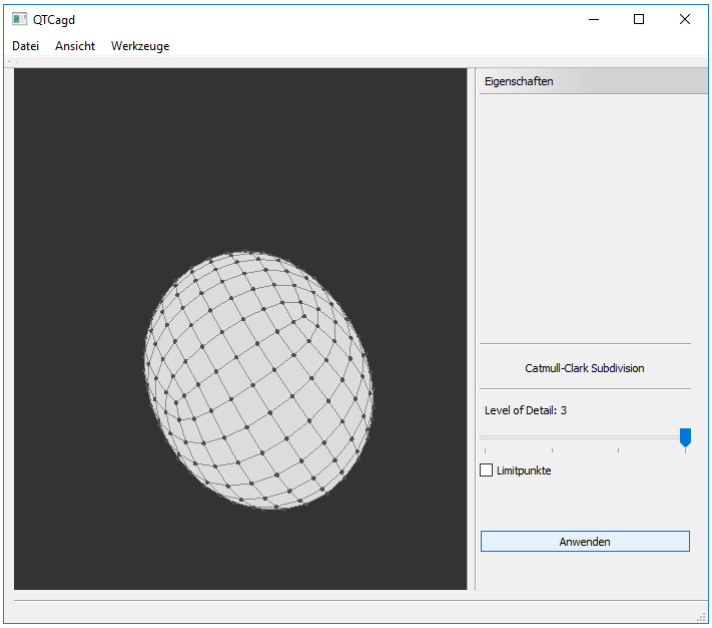

Abbildung 17

7.10 Limitpunkte Anzeigen

Im „Catmull-Clark Subdivision“ Menü können die Limitpunkte angezeigt werden indem die Checkbox „Limitpunkte“ gesetzt wird (siehe Abb. 18).
Wenn die Checkbox „Limitpunkte“ gesetzt ist können keine Objekte geändert werden.

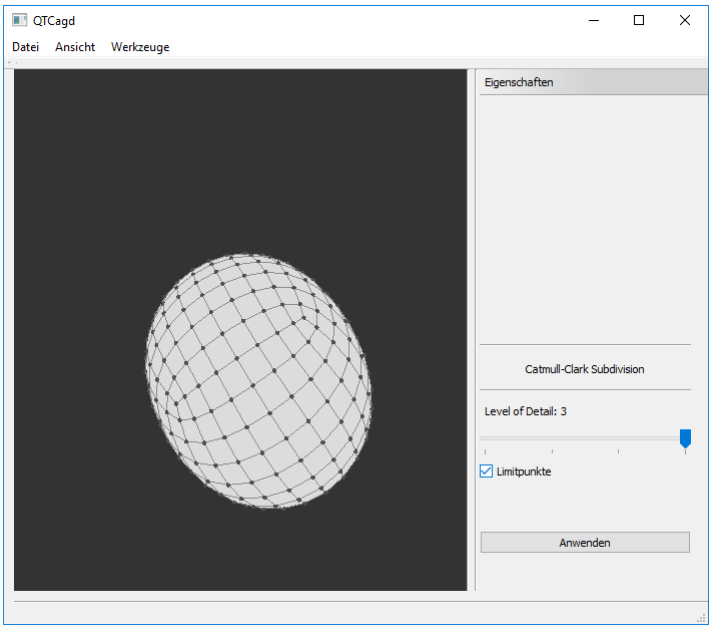

Abbildung 18

7.11 Smoothing

Unter „Werkzeuge->Glättung“ wird das „Glättung“ Menü geöffnet.
Der Slider „Glättung“ glättet das Mesh und zeigt das Mesh in dem eingestellten Glättungs-Level in Prozent an.
Der Button „Anwenden“ übernimmt das geglättete Mesh (siehe Abb. 19).
Wenn der Slider „Glättung“ einen Wert größer als Null hat können keine Objekte geändert werden.

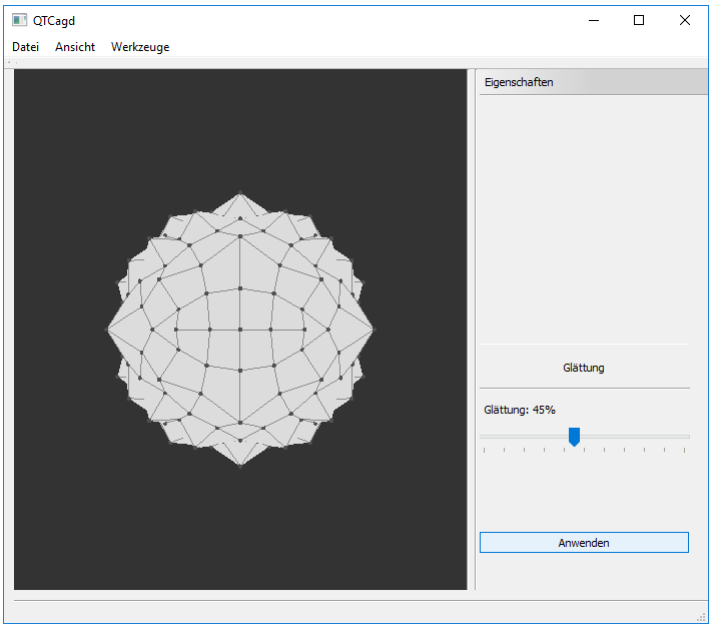

Abbildung 19

7.12 Konsistenzprüfung und Statistik

Mit der Taste „T“ kann ein Mesh auf Fehler untersucht werden und die Statistik angezeigt werden (siehe Abb. 20).
Außerdem wird nach jeder Iteration Catmull-Clark automatisch auf Fehler untersucht.

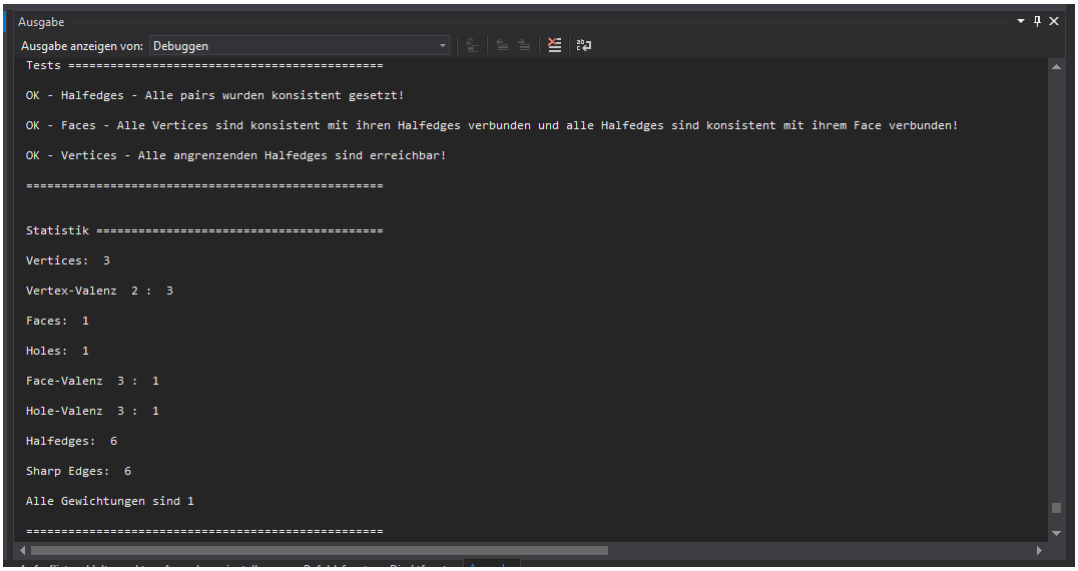

Abbildung 20: Konsistenzprüfung und Statistik für ein Dreieck