Teilthema 1: Polygone

a) Geben Sie die Definition für den Begriff „Polygon“ und für die dazu verwendeten Hilfsobjekte an. (RP, 3)

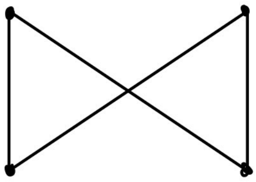
- Eine Menge von Tupel von Punkten wird als Polygonzug bezeichnet. Die Tupel von Punkten sind eine Kante des Polygonszuges. Der Polygonzug ist geschlossen, wenn der erste und letzte Punkt identisch sind. Das von dem Polygonzug umrandete Gebiet ist das Polygon.

b) Definieren Sie „einfaches Polygon“ (RP, 2 P)

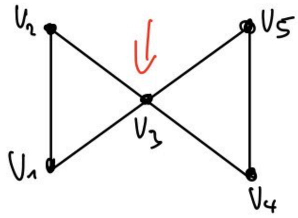
- Ein Polygon ist einfach, wenn der Schnitt von 2 Kanten entweder die leere Menge ist oder ein Eckpunkt ist und jeder Endpunkt einer Kante darf maximal zu zwei Kanten des Polygons gehört.

c) Gemäß Aufgabe b) gibt es zwei Möglichkeiten, wie ein Polygon die geforderten Kriterien für „einfach“ verletzen kann“. Geben Sie je ein solches nicht-einfaches Beispiel-Polygon an (RO, 2 P)

- Schnittpunkt ist kein Endpunkt

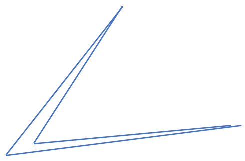


Eckpunkt hat mehr als zwei Kanten



d) Geben Sie ein planares Polygon an (Skizze) bei dem der Orientierungstest basierend auf der Windungszahl mit dem Schwerpunkt als Referenz nicht möglich ist. (TR, 2 P)

- Der Schwerpunkt des Polygons darf nicht im Objekt sein!

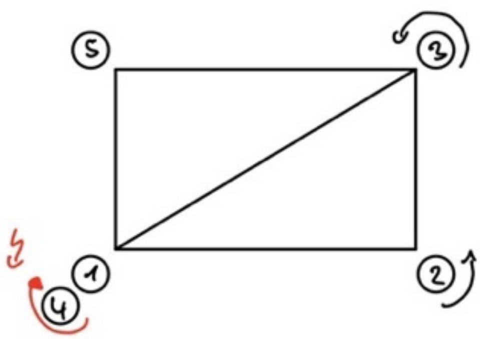


e) Definieren Sie „konvexes Polygon“ (RP, 2 P)

- Bei einem konvexen Polygon ist jeder Eckpunkt von jedem Eckpunkt aus ''sichtbar''. D.h. dass die Verbindungslinie zwischen zwei beliebigen Eckpunkten innerhalb des Polygons verläuft (bei benachbarten Eckpunkten ist diese Verbindungslinie natürlich identisch mit eine Polygonkante).
- Anschaulich gesehen ist es nach außen gewölbt.

f) Geben Sie ein konvexes planares Polygon an, bei dem der Test auf Konvexität aus der VL für einfache planare Polygone fehlschlägt. Erläutern Sie dazu schrittweise den Ablauf bis zum Fehler. (VZ, 4 P)

- Vom 1 bis zum 4 Punkt funktioniert der Test, da wir eine einheitlich Kurvenrichtung (links) haben. Wenn wir jedoch zum 5. Punkt gehen, schlägt der Test fehl da wir ein Richtungswechsel haben.



g) Wieso werden in der Computergrafik besonders gerne Dreiecke anstatt allgemeiner Polygone verwendet? (RP, 1 P)

- Dreiecke sind immer planar & konvex.

h) Erläutern Sie die Funktionsweise von Backface-Culling, beantworten Sie dabei folgende Teilfragen:

i. Wieso wird es verwendet? (Hinweis: Es gibt 2 Gründe)

- Es wird verwendet, um die von der Kamera nicht gesehene Polygone nicht zu rendern.

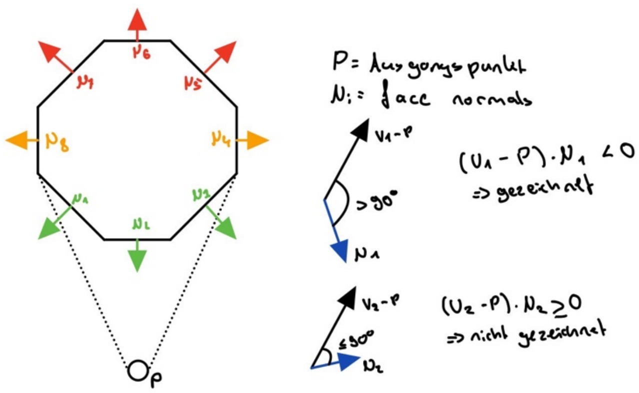
- Performance Gewinn → da abgewandte faces nicht verarbeitet werden müssen.
  Die geringere Anzahl zu zeichnender Flächen erhöht die Darstellungsgeschwindigkeit.
- Man kann das Objekt besser erkennen. (rein visuell)

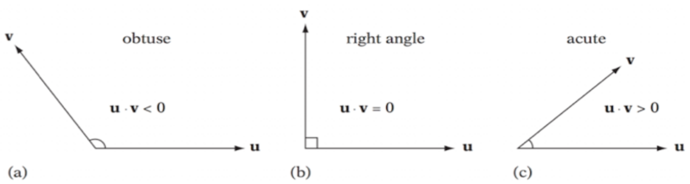
ii. Welches mathematische Kriterium wird verwendet? Erläutern Sie die dabei verwendeten Bezeichner.

- Das Skalarprodukt wird verwendet um die Winkelbeziehung zwischen der Kamera P, einem Eckpunkt V eines Polygons und N seine Normale zu bestimmen mit der Formel (V – P) * N.

iii. Erläutern Sie anhand einer Skizze warum dieses Kriterium geometrisch sinnvoll/korrekt ist

- Das Vorzeichen des Skalarprodukts sagt, ob der Winkel zwischen den Vektoren
  a) **stumpf**, potenziell zur Kamera gewandt
  b) **ein rechter Winkel**, wird als abgewandt festgelegt
  c) **spitz**, abgewandt zur Kamera, also zeigt in die gleiche Richtung wie die Kamera
  => Polygon nicht sichtbar aus Kameraperspektive





Teilthema 2: Polygonale Netze

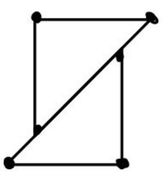
a) Definieren sie den Begriff „Polygonales Netz“. Geben Sie zu jedem Kriterium ein Beispiel-Netz an, das dieses Kriterium nicht erfüllt. (RP, 4 P)

- Ein Polygonales Netz ist eine Menge von geschlossenen, planaren und einfachen Polygone.
- Annahme: Ein Polygon ist ein Dreieck! (immer definieren)

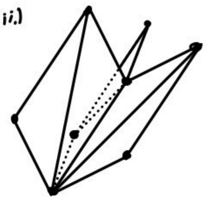
i. 2 Faces haben entweder keinen Punkt, einen Punkt oder eine ganze Kante gemeinsam.
ii. Jede Kante gehört zu einem oder maximal 2 Faces.
iii. Die Menge aller Kanten, die zu einem Face gehören bilden entweder eine leere Menge oder mehrere geschlossene und einfache Polygonzüge.
iv. Jeder Punt hat keine oder genau zwei Kanten, die zum Rand gehören.

Negativbeispiele:

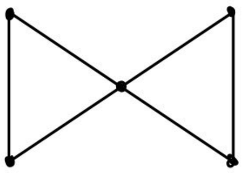
i.)

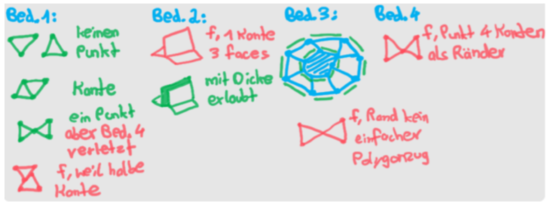


ii.)



iii.)
iv.)





b) Erläutern Sie die Funktionsweise einer Eckenliste, beantworten Sie dabei folgende Teilfragen: (RP, 3 P)

I. Wie ist die grundlegende Funktionsweise, welche Daten werden dazu abgespeichert?

- Trennung von Geometrie und Topologie
- Die Eckenliste besteht aus einer Punktliste mit geometrischen Punkten und wahlfreiem Zugriff in beliebiger Reihenfolge, also enthält keine topologischen Informationen. Ein Polygon wird als Liste von Indizes auf die Punktliste definiert, in der die topologischen Informationen enthalten sind

II. Welche Vorteile bietet eine Eckenliste gegenüber einer expliziten Speicherung?

- Eindeutigkeit der Punkte, keine Gefahr von float-Ungenauigkeiten
- Weniger Redundanz von Punkten → weniger Speicherverbraucht

c) Was ist der Unterschied zwischen Topologie und Geometrie? (RP,2P)

- Geometrie beschreibt die räumliche Lage von Objekten wie z.B. ein Punkt im Raum definiert durch Koordinaten
- Topologie beschreibt die Struktur von Objekten wie Nachbarschaftsbeziehungen, Orientierung wie z.B. von einem Polygon

d) Angenommen basierend auf einer Eckenliste soll die geschätzte Normale eines Punktes aus den angrenzenden Flächennormalen berechnet werden. Wie groß ist die Laufzeit in O-Notation um diese Punkt-Normale zu berechnen? Begründen Sie ihre Antwort. (VZ, 3 P)

- Die Laufzeit beträgt O(n), weil in einer Eckenliste von einem Punkt zu den angrenzen Polygonen zu kommen nicht möglich ist. Man muss durch alle Polygone iterieren und prüfen, ob der Punkt in dem Polygon vorhanden ist

e) Erläutern Sie mit welcher Strategie man - basierend auf einer Ecken- liste - die geschätzte Normale eines Punktes aus den angrenzenden Flächennormalen besonders performant berechnen kann, wenn alle Punkt-Normalen berechnet werden sollen. (RO, 3 P)

I. Geben Sie dazu den Aufwand jedes Teilschrittes in O-Notation an und kumulieren sie anschließend auf.

1. Iteration über alle Punkte und vertice_normals auf (0,0,0) initialisieren O(M)
2. Iteration über alle Faces und für jeden Punkt des Faces die Flächennormale addieren O(N)
3. Iteration über alle vertice_normals um die zu normieren. O(M)

- MAX(O(N), O(M), O(M)) → O(N)
- N: Anzahl der Polygone
- M: Anzahl der Vertices

f) Welches Grundproblem hat jede polygonale Netzdarstellung? (RP, 1 P)

- Netzauflösung bestimmt, wie eckig ein eigentlich rundes oder glattes Objekt wirkt.

g) Mit welchen beiden alternativen Strategien kann man die in f) beschriebene Problematik mindern? (RP, 2 P)

- algorithmische polygonale Verfeinerung
- mathematische Objekte wie Kugel, etc. mit „unendlicher Auflösung"

Klausuraufgaben:

a.) Explizite Speicherung: Welche Probleme gibt es? Eine Skizze und erklären am Beispiel.

- Einzelnen Polygone werden mit der Sequenz von Punkten gespeichert

Nachteile:

- Wegen floating point Ungenauigkeit immer die Frage, ob wirklich der gleiche Punkt gemeint ist
- Hoher Speicherverbraucht für Redundanz
- Keine Speicherung gemeinsamer Punkte

b.) Eckenliste: Was muss an der Eckenliste ändern/ergänzen, damit man die Vertex-Normale in O(n) berechnen kann. (TR)

- Keine Speicherung gemeinsamer Punkte

Teilthema 3: Geometrische Grundprimitive

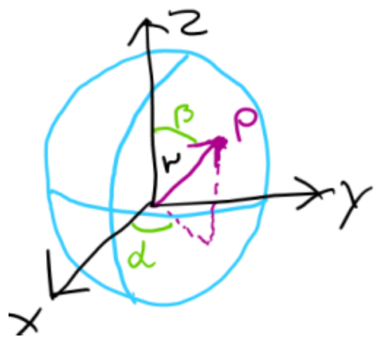
a) Erläutern Sie die Funktionsweise von Kugelkoordinaten, beantworten Sie dabei folgende Teilfragen: (RP, 4 P)

I. Wie ist die grundlegende Funktionsweise mathematisch detailliert (also incl. Formeln) an. Erläutern Sie die verwendeten Bezeichner
   - $\alpha: Winkel\ Querachse,\ \ \beta: Winkel\ Flachachse,\ \ r: radius$
   - $x = r * \cos(\alpha) * \sin(\beta)$
   - $y = r * \sin(\alpha) * \sin(\beta)$
   - $z = r * \cos(\beta)$

     → P(x,y,z)

II. Welche Wertebereiche gelten für die jeweiligen Winkelangaben?
   - $0 \leq \alpha \leq 2\pi$
   - $0 \leq \beta \leq 2\pi$

III. Ergänzen Sie Ihre Ausführungen mit einer beschrifteten Skizze.



b) Welche Nachteile bietet die Tesselierung einer Kugel nach Längen- und Breitengraden? (VZ, 2 P)

- unterschiedliche Flächen der einzelnen Faces → keine Konstante Auflösung
- Unterschiedliche Polygone → Vierecke und Dreiecke

c) Geben Sie mindestens 3 Gründe an, warum einfache Objekte wie Kugel, Zylinder, etc. in der Computergrafik in ihrer mathematischen Darstellung verwendet werden, obwohl Grafik-Karten nur Dreiecke performant verarbeiten können. (VZ, 3 P)

- Level-Of-Detail Möglichkeit: Beliebige Auflösung bei Tesselierung
- Einfache und performante Kollisionsberechnung
- Einfache und performante Transformation (z.B. Verschiebung: nur Mittelpunkt)

d) Was ist ein Sweep- Körper? (RP, 1 P)

- Die Idee: Verschiebe eine Kurve auf einer gekrümmten Leitkurve durch den Raum.

e) Welche beiden Varianten zur Modellierung von Rotationskörpern gibt es?
Welche Polygone werden für die Netzdarstellung jeweils benötigt? (RP, 2 P)

1. Die Achse schneidet die Kurve nicht: (offener Körper)
   - Netz-Modellierung mit Vierecken

2. Die Achse schneidet Anfangs- und Endpunkt der Kurve: (geschlossener Körper)
   - Netz-Modellierung mit Vierecken und Dreiecken

Teilthema 4: Polygonale Verfeinerung

a) Geben Sie die Definition des Prinzips der polygonalen Verfeinerung an. (RP, 4 P)

- Einen Polygonzug durch Hinzufügen von Punkten verfeinern, diese werden bei iterativer Anwendung eine glatte Kurve erzeugen.
- Der polygonale Verfeinerungsprozess ist ein Schema das Kontroll-Polygone Pj,k erzeugt nach der Formel:

$$\mathbf{P}_j^k = \sum_{i=0}^{n_{k-1}} \alpha_{i,j,k} \mathbf{P}_i^{k-1}$$

  und jeder Kontrollpunkt Pj,k als Linearkombination der Kontrollpunkte des vorherigen Kontrollpolygons Pj,k-1 berechnet nach einer Gewichtung alpha

b) Erläutern Sie den Chaikins-Algorithmus in der Sichtweise „pro Kante". Beantworten Sie dazu die folgenden Teilaufgaben: (RP, 4 P)

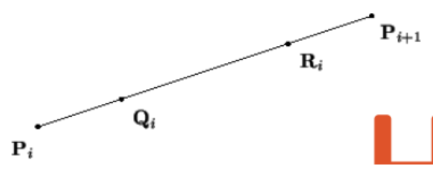
I. Geben Sie formal an, wie das Kontrollpolygon verfeinert wird und erläutern Sie die verwendeten Bezeichner.

- Ein gegebenes Kontrollpolygon {P0, P1, ..., Pn}, wird verfeinert, indem eine Sequenz von neuen Kontrollpunkten {Q0, R0, Q1, R1, ..., Qn−1, Rn−1} iterativ erzeugt wird:

II. Geben Sie das Berechnungsprinzip „pro Kante" als Formel an

III. Ergänzen Sie Ihre Ausführungen mit einer beschrifteten Skizze.

- $\mathbf{Q}_i = \frac{3}{4}\mathbf{P}_i + \frac{1}{4}\mathbf{P}_{i+1} \qquad \mathbf{R}_i = \frac{1}{4}\mathbf{P}_i + \frac{3}{4}\mathbf{P}_{i+1}$



IV. Wie viele Punkte werden aus N gegebenen Punkten in einer Iteration erzeugt?

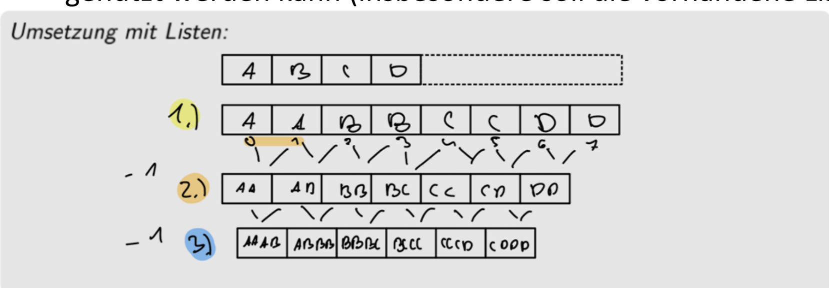
- 2*n-2

c) Erläutern Sie den Chaikins-Algorithmus in der Sichtweise „verdoppeln und mitteln". Beantworten Sie dazu die folgenden Teilaufgaben: (RP,6P)

I. Erläutern Sie strukturiert das Grundprinzip (z.B. gemäß einem „Kochrezept", was wird in welcher Reihenfolge ausgeführt?)

- 1. Verdoppele alle Punkte
- 2. Mittle benachbarte Punkte
- 3. Mittle nochmals benachbarte Punkte
- 4. Gehe zu Schritt 1.

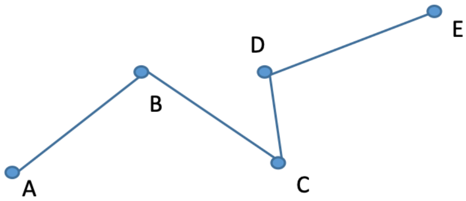
II. Erläutern Sie wie der Speicherplatz in einer Liste bei der Implementierung möglichst effizient genutzt werden kann (insbesondere soll die vorhandene Liste nur vergrößert werden).



- Das Verdoppeln der Punkte wird in inplace gemacht. Die Größe der Liste wird verdoppelt und dann wird von hinten iteriert, um die Punkte richtig zu verschieben. Siehe Skizze Schritt 1
- Das Mitteln wird von vorne durchiteriert und es wird ausgenutzt, dass die polygonale Verfeinerung nur die letzte Kontrollsequenz benötigt wird, sodass Werte passend überschrieben werden können. Siehe Skizze Schritt 2

d) Führen Sie eine Iteration des Chaikins-Algorithmus in der Sichtweise „verdoppeln und mitteln“ für folgenden Polygonzug aus: (RP, 6 P)

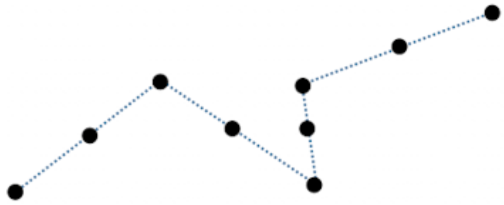
A	B	C	D.	E

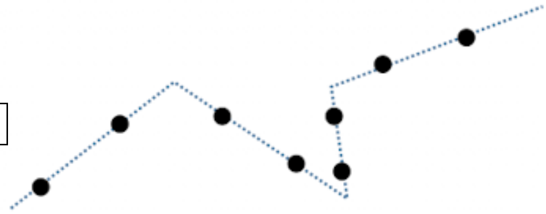


A	A	B	B	C	C	D	D	E	E

AA	AB	BB	BC	CC	CD	DD	DE	EE

AAAB	ABBB	BBBC	BCCC	CCCD	CDDD	DDDE	DEEE





e) Wie kann mit dem Chaikins-Algorithmus für jeden vorgegebenen Polygonzug eine beliebig glatte Kurve (also eine, bei der auf dem Bildschirm keine Polygonale „Eckigkeit“ mehr sichtbar ist) erzeugt werden? (TR, 2 P)

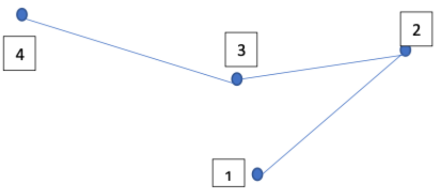
- So viele Unterteilungsiterationen machen dass die längste Linie kleiner als ein Pixel ist

f) Wie ist es möglich bei bekannter Bildschirmauflösung eine Chaikins-Kurve zu erzeugen die an allen bis auf einer Stelle keine sichtbare Eckigkeit mehr hat? (TR, 2 P)

- Man erzeugt eine beliebige Kurve, die jedoch an einer Stelle eine Ecke (90°) hat und diese Ecke und die benachbarten Punkte mit sehr vielen Punkten mit den gleichen geometrischen Positionen an diesen 3 Punkten, jedoch topologisch verschieden, erzeugt. Das sollte dazu führen, dass bei beliebig häufiger Mittelung, diese Punkte weiterhin an der gleichen Stelle bleiben. Es soll so oft gemittelt werden, bis alle Kanten kleiner als ein Pixel sind außer die 2 Kanten, die zu dieser oben genannten Ecke gehören.

Klausuraufgaben:

a.) 4-Punkt-Shema anhand von Beispiel-Kurve herleiten. Mit Formel etc. Erklärung zu dem Parameter. Dazu noch erklären was kubische Präzision ist und wieso man die gerne hätte.



b.) Wie oft darf pro Iteration (Lane Riesenfeld) gemittelt werden, dass bei einer punktlänge N die Kurve nicht ganz verschwindet.

- Beim Mitteln darf Liste nicht kleiner sein als Ursprungsliste (N mal)

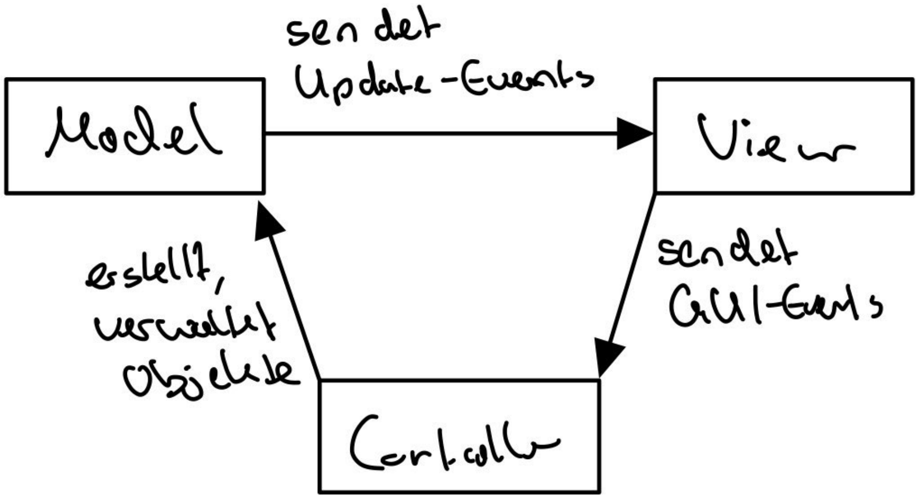
Teilthema 5: sonstiges

a) Was ist ein Level-Set? (RP, 2 P)

- Eine Menge von Punkten an der eine reellwertige Funktion s(x) den konstanten Wert c annimmt heißt Level-Se

b) Erläutern Sie anhand einer Skizze ausführlich das MVC-Konzept im Kontext der Computergrafik (RP, 5 P)

- Model enthält
  - Geometrische Objekte in lokalen Koordinaten
  - Appearance
  - Algorithmen
- View
  - Enthält GUI-Elemente und Grafikbibliothek wie OpenGL
  - Iteriert durch alle Objekte und zeichnet mit allen zugehörigen Transformationsmatrizen
- Control enthält
  - Szenengraph und erstellt Objekte
  - Event-handling



-