Dr. Michael J. Winckler
IWR, Raum 302
INF 368
69120 Heidelberg
Michael.Winckler@iwr.uni-heidelberg.de
http://www.iwr.uni-heidelberg.de/teaching/Mathe-Star/

Arbeitsblatt 13 *Terrainvisualisierung* (Version 1.10)

1 Gitterbasierter Ansatz

1.1 Bestandteile

- Evaluator-Funktion `Height(x,y)`, mit der an jeder Stelle des Geländes die aktuelle Höhe abgerufen werden kann (z-Koordinate).
- Festes, meist quadratisches oder rechteckiges Raster aus Koordinatenpunkten. Die Höheninformation wird i.d.R. einmal ausgewertet und danach in einem 2D-Array abgespeichert.
- Auf Dreiecksstreifen basierendes, zugeschnittenes Auswertungsschema.

1.2 Zusammenwirken

Die Auswertungsfunktion wird zur Bestimmung des Höhenwertes an den Gitterpunkten verwendet. Daraus ergibt sich ein 3D-Gitter, das in regelmässigen x- und y-Rastern das Gelände abtastet und auf Basis einer festen Darstellungsroutine (z.B. auf Basis von Quadstrips) die Visualisierung erzeugt.

1.3 Vor- und Nachteile

1.3.1 Vorteile

- Schnelle Visualisierungsroutine durch Dreiecks- oder Viereckszüge
- Effiziente Auswertung des Höhenprofils

1.3.2 Nachteile

- Starres Feinheitsgerüst bzw. einheitliche Feinheit der Darstellung
- Kein dem Terrain angepasstest Datenmodell
- Darstellung unabhängig vom Beobachterstandpunkt
- für grosse Datensätze/Terrains nicht einsetzbar

2 Rekursiver Ansatz

Der Rekursive Ansatz geht in seiner einfachsten Version von einem einzelnen darzustellenden Grundgebiet aus, das als einzelnes Dreieck angegeben wird. Dabei wird in der x-y-Ebene ein Auswertungsgebiet definiert und zu den drei Eckpunkten die Höhendaten aus der Auswertungsfunktion bestimmt.

Danach wird dieses Dreieck an eine Zeichenfunktion übergeben, deren Aufgabe es ist, entweder das Dreieck direkt zu zeichnen oder eine Unterteilung in zwei kleinere Teildreiecke vorzunehmen, für die diese Vorgehensweise rekursiv übernommen wird.

Die Entscheidung, ob ein Dreieck gezeichnet werden soll oder noch weiter unterteilt wird, wird auf Basis einer Bewertungsfunktion getroffen, deren Formulierung den eigentlichen Algorithmus des rekursiven Ansatzes ausmacht.

2.1 Grobstruktur des Algorithmus

Wir gehen im folgenden von einem Gebiet aus, das aus einem einzelnen Dreieck besteht. Das Dreieck ist durch seine drei Eckpunkte P_0, P_1, P_2 festgelegt, deren Koordinaten sich aus den Gebietskoordinaten x und y sowie aus der Höhenkoordinate z ergeben. Das Dreieck ist so in die Datenstruktur eingepasst, dass der Abstand zwischen P_0 und P_2 im Grundgebiet (x-y-Ebene) am grössten ist.

proc *RekurssiveSubdivide*$(P0, P1, P2, lev)$ $\equiv$

$PM = (P0 + P2)/2$	Mittelpunkt P0-P2
$h = Height(PM.x, PM.y)$	
$PH = (PM.x, PM.y, h)$	wirklicher Terrainpunkt
if $((abs(PM - PH) < delta) \vee (lev > LEVMAX))$	gleich Zeichnen
$Draw(Triangle(P0, P1, P2))$	
else	
$RecursvieSubdivide(P0, P1, PH, lev + 1)$	Subdiv 1
$RecursiveSubdivide(P1, P2, PH, lev + 1)$	Subdiv 2
fi	

end

2.2 Design der Bewertungsfunktion

Die im oberen Beispiel verwendete Bewertungsfunktion deutet an, dass ein zentrales Bewertungselement der Unterschied zwischen dem interpolirten Zwischenwert zwischen den beiden Eckpunkten der längsten Dreiecksseite und dem wirklichen Terrainpunkt an dieser Stelle sein kann.

Das Mass dieses Unterschieds kann man sowohl in absoluten Werten als auch in Screen-Kooridnaten berechnen. Beispielsweise lässt sich so absch“ytzen, wieviele Pixel sich durch eine weitere Unterteilung des Dreiecks überhaupt „ndern würden. Wenn eine Unterteilung am erzeugten Bild im Wesentlichen nur einige wenige Pixel ändert, macht eine Unterteilung evtl. keinen sinn.

Die Bewertungsfunktion kann somit Gegebenheiten wie die Position des Boebachters, die Neigung des Terrains gegen die Waagerechte bzw. den Horizont und die Entfernung von der Kamera miteinbeziehen, um eine noch bessere Darstellung zu erzielen.

3 Aufgaben

Aufgabe 13.1

Schreibe bzw. erweitere (d)eine Terrain-Klase zur Beschreibung eines 3D-Terrains. Implementiere dabei eine einfache Funktion zur Erzeugung eines Höhenwertes. Diese Funktion werden wir später noch erweitern. Implementiere einen dierkten Visualisierungsansatz, in dem ein Bereich des Terrains auf Basis von Dreiecksstreifen gezeichnet wird (ohne Verfeinerungen).

Implementiere zudem eine Dreiecksklasse zur Verwaltung von Dreiecken. Diese Klasse werden wir später zudem in einen Listencontainer verpacken oder mit einer eigenen Listenstruktur versehen (einfach oder doppelt verkettet). Zunächst muss ein Dreieck nur seine Eckpunkte und seine Farbe kennen und sich auf den Bildschirm zeichnen können.

Aufgabe 13.2

Erweitere das Terrain nun um eine rekursiv-statische Zeichenmethode. Dazu kann man dem Terrain eine Methode geben, ein Dreiecksgebiet darzustellen. Diese Methode unterteilt das Dreieck (gemäss einem inneren Kriterium) rekursiv in Unterdreiecke und sendet diesen schliesslich den Befehl, sich zu zeichnen (wenn z.B. die Genauigkeitsanforderungen erreicht sind).

Zu diesem Zweck ist es notwendig, den Dreiecken ein *Level* als Member Data zu geben, damit jedes Dreieck weiss, zu welchem Darstellungslevel es gehoert. Dies wird bei Algorithmen wie ROAM und SOAR wichtig, die versuchen, die beim Unterteilen auftretenden Cracks zu vermeiden.