

LINQ for Geometry - VORLÄUFIGES DOKUMENT

**Implementierung der Half-Edge Datenstruktur zu Manipulation und
Handling Dreidimensionaler Meshes insbesondere durch den Einsatz
von LINQ und LAMBA Ausdrücken in Microsofts C#**

Dominik Steffen

Erstbetreuer: Prof. Christoph Müller, Fakultät DM

Zweitbetreuer: Prof. Wilhelm Walter, Fakultät DM

18. März 2013

Inhaltsverzeichnis

1	Einleitung	1
1.1	Fragestellung	1
1.2	Warum C# ?	1
1.3	Die halfedge data structure (kurz HES) als Algorithmus	1
1.3.1	Die Basis der HES	1
1.3.2	Speicherverbrauch im Gegensatz zu Face basierten Lösungen	1
1.3.3	Vorteile der HES	1
1.4	Aktueller Forschungsstatus	1
1.5	Probleme der aktuellen Forschung	1
1.6	Einführung zu LINQ in C#	1
1.7	Einführung zu Lambda in C#	1
2	Hauptteil	2
2.1	Gegenüberstellung nativer (OpenMesh.org) und gemanagter HES	3
2.2	Geschwindigkeitsunterschiede von nativem und Code in der C# Umgebung	3
2.3	Implementierung und Funktion der Handler für die einzelnen Komponenten der HEDS	3
2.3.1	Edges und Handler	3
2.3.2	Half-Edges und Handler	3
2.3.3	Vertices und Handler	3
2.3.4	Faces und Handler	3
2.4	LINQ und Lambda Ausdrücke und ihre Stärken und Schwächen bei der Selektierung großer Datenmengen	3
2.5	Stern- und Umlaufenumeratoren (Iteratoren)	3
2.5.1	Verwendete „Design-Patterns“ und Softwarelösungen	3
2.5.2	Linq und Lambda Ausdrücke in den Enumeratoren	3
2.6	Der Import von Geometriedaten im „Wavefront Object“ Format	3
2.6.1	Warum das Wavefront Object Format	3

2.6.2	Importer für das Wavefront Format	3
2.7	Manipulation von Mesh Daten in der Datenstruktur	3
2.7.1	Manipulation von Vertices	3
2.7.2	Manipulation von Kanten	3
2.7.3	Manipulation von Faces	3
2.8	Evtl. beispielhafte Implementierung von Standard Geometriemanipulationsalgorithmen	3
3	Schluss	4
3.1	Ergebnis der Arbeit	4
3.1.1	In wie weit ist das Programm produktiv Nutzbar	4
3.1.2	Welche Schnittstellen müssen noch geschaffen werden	4
3.2	Abschließender Vergleich von Native und Managed Code	4
3.2.1	Wie groß sind die Unterschiede Tatsächlich ausgefallen	4
3.3	Zukünftige Entwicklung	4
	Literaturverzeichnis	5

1 Einleitung

1.1 Fragestellung

Ist es möglich die „halfedge data structure“ (kurz HES) in einer gemanagten Programmiersprache wie C# unter der Berücksichtigung von Linq und Lambda Ausdrücken so performant zu gestalten dass damit grundlegende Geometriemanipulation in der Computergrafik erfolgen kann?

1.2 Warum C# ?

1.3 Die halfedge data structure (kurz HES) als Algorithmus

1.3.1 Die Basis der HES

Verbindungen und Beziehungen in der HES

1.3.2 Speicherverbrauch im Gegensatz zu Face basierten Lösungen

1.3.3 Vorteile der HES

1.4 Aktueller Forschungsstatus

1.5 Probleme der aktuellen Forschung

1.6 Einführung zu LINQ in C#

1.7 Einführung zu Lambda in C#

2 Hauptteil

2.1 Gegenüberstellung nativer (OpenMesh.org) und gemanagter HES

2.2 Geschwindigkeitsunterschiede von nativem und Code in der C# Umgebung

2.3 Implementierung und Funktion der Handler für die einzelnen Komponenten der HEDS

2.3.1 Edges und Handler

2.3.2 Half-Edges und Handler

2.3.3 Vertices und Handler

2.3.4 Faces und Handler

2.4 LINQ und Lambda Ausdrücke und ihre Stärken und Schwächen bei der Selektierung großer Datenmengen

2.5 Stern- und Umlaufenumeratoren (Iteratoren)

2.5.1 Verwendete „Design-Patterns“ und Softwarelösungen

2.5.2 Linq und Lambda Ausdrücke in den Enumeratoren

2.6 Der Import von Geometriedaten im „Wavefront Object“ Format

3

2.6.1 Warum das Wavefront Object Format

2.6.2 Importer für das Wavefront Format

Face basierter Import - Edge basiertes Handling

3 Schluss

3.1 Ergebnis der Arbeit

3.1.1 In wie weit ist das Programm produktiv Nutzbar

3.1.2 Welche Schnittstellen müssen noch geschaffen werden

3.2 Abschließender Vergleich von Native und Managed Code

3.2.1 Wie groß sind die Unterschiede Tatsächlich ausgefallen

3.3 Zukünftige Entwicklung

Literaturverzeichnis