

Pflichtenheft
Programmieren eines 3D-Editors
-- Blacksun --

Mitglieder:	Philipp Gruber Reinhard Jeschull Thomas Kuhndörfer Thomas Tischler Stefan Zeltner	pgruber@fh-landshut.de rjeschu@fh-landshut.de tkuhndo@fh-landshut.de ttischl1@fh-landshut.de szeltne@fh-landshut.de
Betreuer:	Prof. Peter Hartmann	
Homepage:	http://sourceforge.net/projects/blacksun/	
letzte Änderung:	Mittwoch, 18. Oktober 2006	

Inhaltsverzeichnis

1 Zielbestimmung.....	3
1.1 Mußkriterien.....	3
1.2 Wunschkriterien.....	3
1.3 Abgrenzungskriterien.....	3
2 Produkteinsatz.....	3
2.1 Anwendungsbereiche.....	3
2.2 Zielgruppe.....	3
3 Produktumgebung.....	3
3.1 Software.....	4
3.2 Hardware.....	4
4 Produktfunktionen.....	4
4.1 Grundeditierungsmöglichkeiten.....	4
4.2 Grundobjekte.....	4
4.3 Model-Manipulation auf Vertex-Ebene.....	5
4.4 Model-Manipulation auf Face-Ebene.....	5
4.5 Model-Manipulation auf Gruppen-Ebene.....	6
4.6 Texturierung.....	6
4.7 Darstellung.....	7
4.8 Sonstige enthaltene Tools.....	7
4.9 Dateioperationen.....	7
4.10 Pluginmanagement.....	8
4.11 Sonstiges	8
5 Produktdaten.....	8
6 Produktleistungen.....	8
7 Benutzeroberfläche.....	8
8 Qualitätsbestimmungen.....	8
9 Entwicklungsumgebung.....	9
9.1 Programmiersprache/IDE.....	9
9.2 Bibliotheken.....	9
9.3 Tools.....	9
10 Ergänzungen.....	9
10.1 Spezielle Projektrollen.....	9

1 Zielbestimmung

Das Ziel des Studienprojektes ist die Erstellung eines oberflächenbasierten Programmes zur Modellierung von statischen 2D/3D-Objekten.

1.1 Mußkriterien

- Erweiterbarkeit durch Plugin-Schnittstelle
- Standardtools zur Bearbeitung von 3D-Objekten (z.B. Translation, Skalierung, Extrudierung)
- Standardtools zur Generierung von Standard-Objekten (z.B. Kugeln, Torus, Plane)
- Englische Anwenderdokumentation
- Englische Entwicklerdokumentation des Klassendiagramms / Frameworks
- Auf intuitive Bedienung ausgelegte GUI
- Schneller OpenGL-Renderer
- Auf Performanz zugeschnittene Mathe-Funktionen
- Texturierungsmöglichkeiten von modellierten Objekten
- Eigenes Dateiformat zur Speicherung der Editor-Daten
- Ex-/Importer zu den gängigsten Modell-Formaten
- Online-Update als Menüpunkt im Programm

1.2 Wunschkriterien

- Fortgeschrittene Werkzeuge (z.B. spezielle Normalenberechnung, Subdivision, Bevel)
- CSG (Constructive Solid Geometry) zur Anwendung von booleschen Operatoren auf Körper
- Materialeditor um komplexere Materialien zu erstellen
- Tool zur Pflanzengenerierung über L-Systeme
- Prozeduraler Texturgenerator (z.B. für Holz, Marmor)
- Umschaltbare GUI-Sprache (Englisch/Deutsch)
- Individualisierung der GUI mit umstellbaren Skins/Styles

1.3 Abgrenzungskriterien

- Die Fähigkeit zur Animation von erstellten Objekten wird ausgeschlossen.

2 Produkteinsatz

Prinzipbedingt kann der Modeller in unterschiedlichsten Branchen zum Einsatz kommen, eine eindeutige Einschränkung auf eine Anwendergruppe wird nicht vorgenommen.

2.1 Anwendungsbereiche

Das Einsatzfeld reicht vom Amateurbereich bis hin zum semiprofessionellen Bereich.

Im Vordergrund steht die Spieleentwicklung im Allgemeinen, welche stets auf einfach erstellbare 3D-Objekte angewiesen ist. Diesen Modellen gemein sind die Eigenschaften, meist aus relativ wenigen Polygonen zu bestehen, mit einem Fokus auf schnell erzielbare Resultate.

2.2 Zielgruppe

Um uns von den etablierten Paketen abzugrenzen, konzentrieren wir uns bei unserem Projekt auf die reine Modellierungskomponente. Dadurch kann das gesamte Anwendungsschema auf diesen Punkt hin optimiert werden.

Damit stellt *Blacksun* eine schnell erlernbare Alternative für jeden Grafikenthusiasten dar, der vor einer Einarbeitung in die großen Pakete zurückschreckt. Besonders wenn sich der betreffende Anwender nur für das Modellieren interessiert und nicht durch zu viele Wahlmöglichkeiten überfordert werden will.

3 Produktumgebung

Das Produkt ist weitgehend unabhängig vom Betriebssystem, sofern folgende Produktumgebung vorhanden ist.

3.1 Software

- OpenGL
- Installierte Qt-Bibliothek (nicht zwingend, da dies Qt-Libraries auch mitgeliefert werden können)

3.2 Hardware

- OpenGL-fähige Grafikkarte

4 Produktfunktionen

Die hier aufgelisteten Funktionen stellen die Funktionalität des Editors dar. Die mit **[optional]** gekennzeichneten Punkte sind im Zeitraum des Studienprojektes vermutlich nicht zu erfüllen bzw. nicht in ihrem vollem Umfang. Sie sind aber trotzdem aufgeführt für eine Weiterarbeit nach dem Studienprojekt bzw. falls es die Zeit während dem Studienprojekt zulässt.

4.1 Grundeditierungsmöglichkeiten

Der User kann Modelle erzeugen, indem dieser Objekte generiert und diese danach bearbeitet. Dies setzt einige Grundfunktionen voraus.

/F00010/ Selektieren von beliebigen Teilen eines Models. Das Selektieren kann über verschiedene Ebenen erfolgen:

- Vertex-Ebene: Es kann jeder Vertex einzeln selektiert werden
- Face-Ebene: Es kann jedes Face (ein Triangle aus 3 Vertices) einzeln selektiert werden
- Group-Ebene: Es kann jede Gruppe einzeln selektiert werden

/F00020/ Beliebiges bewegen/verschieben von selektierten Modelldaten

/F00030/ Beliebiges rotieren von selektierten Modelldaten.

/F00040/ Beliebiges skalieren von selektierten Modelldaten.

/F00050/ Extrudieren einer Fläche per Parallelverschiebung um diese in einen 3-dimensionalen Körper umzuwandeln (Ein Quadrat wird nach dem extrudieren zu einem Quader).

/F00060/ Kopieren, einfügen und ausschneiden von selektierten Modelldaten.

/F00070/ Direktes duplizieren von selektierten Modelldaten.

/F00080/ Löschen der selektierten Modelldaten aus der Szene.

/F00090/ Löschen der kompletten Szene.

/F00100/ Selektieren einer Gruppe **über** ihren **Namen** oder ihr **Material(-namen)**.

/F00110/ Selektieren der **ganzen Szene** (ohne alles selber selektieren zu müssen).

/F00120/ Aufheben der Modelldaten-**Selektierung**, sodass nichts mehr selektiert ist.

/F00130/ Invertieren der **Selektierung**.

/F00140/ Verbergen der selektierten Faces und Gruppen, diese werden in den verschiedenen Ansichten nicht mehr angezeigt. Ebenso kann dies wieder für die ganze Szene rückgängig gemacht werden, sodass **alles** wieder **sichtbar** ist.

/F00150/ Schützen einer **Gruppe** vor jeglicher Veränderung, indem es **gesperrt** wird. Ebenso kann eine Gruppe wieder **entsperrt** werden.

/F00160/ Alle Aktionen können mit dem bekannten **Undo/Redo**-System wieder rückgängig gemacht bzw. wiederhergestellt werden

4.2 Grundobjekte

Dem User des Editors stehen verschiedene Grundobjekte zur Verfügung, mit dessen Hilfe die Szene aufgebaut werden kann. Diese Grundobjekte können natürlich selber beliebig manipuliert werden.

Folgende Grundobjekte stehen zur Verfügung:

- /F01010/ Vertex
- /F01020/ Face (Ein Triangle bestehend aus 3 Vertices)
- /F01030/ Box
- /F01040/ Block (Eine Box bei der die Seiten aus $N \times N$ Teilflächen bestehen)
- /F01050/ Kugel
- /F01060/ GeoSphere **[optional]**
- /F01070/ Zylinder
- /F01080/ Kegel
- /F01090/ Röhre
- /F01100/ Gitter (Eine Fläche bestehend aus $N \times N$ Teilflächen)
- /F01110/ Linie (2D) **[optional]**
- /F01120/ Polyline (2D) **[optional]**
- /F01130/ Kreis (2D) **[optional]**
- /F01140/ Schrift **[optional]**
- /F01150/ Polyhedron (ein N -Flächer) **[optional]**
- /F01160/ Teapot **[optional]**

4.3 Model-Manipulation auf Vertex-Ebene

Bei Vertices handelt es sich um den primitivsten Grundtypen eines 3D-Objekts. Bei der Manipulation auf Vertex-Ebene werden die Eckpunkte eines Modells direkt verändert.

- /F02010/ **Zusammenfassen** beliebig vieler Vertices, sodass diese alle die gleiche Position besitzen (Bei dieser Zusammenfassung werden keine Vertices gelöscht)
- /F02020/ **Ausrichten** der Position beliebig vieler Vertices **am Gitter**. Jede Position wird an die nächste Gitterposition angepasst. Das Gitter wird vom User festgelegt über Einstellungen.
- /F02030/ **Ausrichten** von Vertices **an eine Ebene**, sodass diese eine Fläche bilden.
- /F02040/ **Spiegeln** von Vertices **an einer Ebene**.
- /F02050/ **Manuelles bearbeiten** der Vertexdaten selektierter Vertices. Darunter fallen unter anderem die Position, Texturkoordinate und die Farbe.
- /F02060/ **Erzeugung** einer **triangulierten Fläche** aus einer Menge von (ungebundenen) Vertices.
- /F02070/ **Einfügen** eines **zusätzlichen Vertex** zwischen zwei selektierten Vertices. Dieser wird (falls möglich) in die Gruppe eingebunden und Faces werden geteilt, sodass der neue Vertex darin enthalten ist.
- /F02080/ **Ungebundene Vertices**, die in keinem Face eingebunden sind, können aus der Szene **entfernt** werden.
- /F02090/ Erzeugung einer **konvexen Hülle** aus einer Menge von Vertices, die alle selektierten Vertices enthält.

4.4 Model-Manipulation auf Face-Ebene

Von einem Face (dt. Seite) spricht man, wenn man eine Menge von Vertices (in unserem Falle 3) als ein Polygon betrachtet. Durch dieses Zusammenfassen von Vertices zu einem Face stehen dem Anwender besondere Modellierungsoptionen zur Verfügung, welche bei einzelnen Vertices nicht möglich wären.

- /F03010/ **Umdrehen** der **Vertexreihenfolge** in den selektierten Faces (Rückseite wird zur Vorderseite)
- /F03020/ **Teilen** der selektierten **Faces** in **beliebig viele Subfaces**
- /F03030/ **Vertauschen** der **Ecken** zweier selektierter und benachbarter Faces (die den Faces zugeordneten Vertices werden damit geändert)
- /F03040/ **Abschrägen** der Kanten (Bevel) **[optional]**
- /F03050/ **Erzeugung** eines **Lochs** in jedes selektierte Face
- /F03060/ **Generierung** von **Stacheln** aus jedem selektiertem Face (einfügen eines Vertex in die Mitte des Faces und verschieben dieses Vertex)

4.5 Model-Manipulation auf Gruppen-Ebene

Eine Gruppe ist eine Zusammenfassung von Faces und Vertices. Auch auf diesen Gruppen lassen sich spezielle Operationen anwenden, die nur auf Gruppenebene Sinn machen.

- /F04010/ **Zusammenfassen** der selektierten Faces/Gruppen zu einer **neuen Gruppe**
- /F04020/ Erstellen eines **Rotationskörpers** aus einer selektierten Gruppe
- /F04030/ Automatische **Optimierung der Vertexanzahl** (z.B. Löschen doppelter Vertices, ...). **[optional]**
- /F04040/ **Reduzierung** der **Vertexzahl** um einen beliebigen Prozentsatz. **[optional]**
- /F04050/ Anwendung von **Subdivision**. Diese Technik macht das Modell durch Einfügen zusätzlicher Vertices komplexer, plastischer und runder. **[optional]**
- /F04060/ **Verschieben** jedes **Faces** einer Gruppe vom **Zentrum** dieser Gruppe (bekannt als '**Explosion**').
- /F04070/ **Normalen** an **Ecken** **weich** berechnen.
- /F04080/ Anwendung von **CSG (Constructive Solid Geometry)** zur Nutzung von Booleschen Operationen auf Gruppen. **[optional]**
Solche Boolesche Operationen sind:
 - Vereinigung
 - Differenz
 - Schnittmenge

4.6 Texturierung

Um die Oberfläche eines Models zu beschreiben gibt es die sogenannten Texturen. Das sind Bilddateien, die auf das Modell gelegt werden. Um diese Bilddaten mit anderen materialspezifischen Eigenschaften zu vereinen, verwendet der Editor Materialien. Ein solches Material umfasst alle dafür genutzten Texturen (Multitexturing), Reflektionswert, Materialgrundfarbe und einige mehr, die das Material beschreiben.

- /F05010/ **Hinzufügen** von **Materialien**.
- /F05020/ **Einstellen** von **Materialien** in einem **Material-Editor**.
- /F05030/ **Speichern** und **Laden** von **Materialien** (im Material-Editor).
- /F05040/ **Zuweisen** eines **Materials** an eine Gruppe.
- /F05050/ **Löschen** der **Zuordnung** eines Materials an eine Gruppe.

/F05060/ Beliebige Anpassung der Texturkoordinaten über einen **Texturkoordinaten-Editor**.

/F05070/ Automatische **Generierung** von **Texturkoordinaten**. **[optional]**

Texturgeneratoren sind:

- Planar Mapping
- Box Mapping
- Spherical Mapping
- Cylindrical (cap) Mapping
- Global Mapping

/F05080/ **Einstellen** des **Texturkoordinaten-Offsets** (die Verschiebung) und die Anzahl der **Wiederholungen**

/F05090/ **Generierung** von **Texturen**, z.B. Holz, Marmor, spezielle Muster, ... **[optional]**

4.7 Darstellung

Die frei positionierbaren Fenster, die die Qt-Bibliothek zur Verfügung stellt, werden genutzt um den Aufbau der Benutzerschnittstelle dem Anwender zu überlassen. Neben dem bereits vorgegebenen Aufbau, kann der Benutzer die Fenster neu positionieren und diese Einstellungen dauerhaft speichern.

Um eine bessere Orientierung im 3-dimensionalen Raum anzubieten, hat der Anwender die Möglichkeit das Szenenlayout auf die Bedürfnisse einer Modellierungstätigkeit anzupassen.

/F06010/ Bereitstellen **verschiedener Darstellungsmöglichkeiten** für die 4-Fenster-Ansicht

/F06020/ Beliebiges **einstellen** der **4-Fenster-Ansicht**

/F06030/ **Anpassen** der Viewports, sodass das **ganze Modell sichtbar** ist

/F06040/ **Anpassen** der Viewports, sodass die **selektierten Modellteile sichtbar** sind

/F06050/ **Wiederherstellen** der Viewports auf ihre **ursprünglichen Einstellungen**

/F06060/ **Darstellen** der Daten auf verschiedene Arten, z.B.

- Wireframe (Drahtgittermodell)
- Solid (ohne Textur)
- Texturiert

/F06070/ **Speichern** des Inhalts eines der Viewports in eine **Datei** oder kopieren in die **Zwischenablage**.

/F06080/ Möglichst **beliebig einstellbare GUI** (Menüs, Toolbars, ...).

/F06090/ **Individualisierung** des Aussehens der GUI über Skins/Styles **[optional]**

4.8 Sonstige enthaltene Tools

Zur Übersicht über eine erstelltes Modell können verschiedene Metadaten einer Szene abgerufen werden. Ebenso werden noch spezielle Tools angeboten, die dem User die Arbeit beim Modellieren vereinfachen.

/F07010/ **Kommentieren** des Models

/F07020/ Anzeigen der **Model-Statistik** (z.B. Vertexanzahl, Ausmaße des Models, ...)

/F07030/ Anzeigen der **Model-Hierarchie**

/F07040/ Zuschaltbares **automatisches Spiegeln** an einer Ebene. Eine Änderung wird automatisch übernommen um einfach symmetrische Objekte zu erstellen.

/F07050/ Zuschaltbare **Darstellung** der **Normalen** und ähnliche **wichtige Vektoren** in der 3D-Ansicht.

/F07060/ Tool zur **Generierung** von **Planzen-Modelle** mithilfe der L-Systeme.

4.9 Dateioperationen

Hierbei werden die üblichen Dateioperationen zur Verfügung gestellt, um das editoreigene Format zu verwalten. Darüber hinaus wird eine Import- und Exportmöglichkeit angeboten, um Modelle in anderen Editoren, Spielen und ähnliches weiter nutzen zu können.

/F08010/ **Erstellen** einer **neuen Szene**.

/F08020/ **Öffnen** einer Szene aus einer Datei.

/F08030/ **Speichern** der Szene.

/F08040/ **Einfügen eines Modells** aus eine Datei in die aktuelle Szene.

/F08050/ **Importieren** eines Modells aus einem bestimmten Dateiformat.

/F08060/ **Exportieren** der Szene in ein bestimmtes Dateiformat.

4.10 Pluginmanagement

Es wurde die strategische Entscheidung getroffen, möglichst viele Komponenten durch eine einheitliche Pluginschnittstelle anzubinden. Dadurch können viele Programmteile unabhängig vom Hauptprogramm entwickelt werden, was den Entwicklungsprozess stark vereinfacht und das Programm leicht erweiterbar macht. Zu Plugins zählen z.B. Exporter, Importer, Tools, ...

/F09010/ **Einfügen** von Plugins mithilfe eines Pluginmanagers .

/F09020/ **Löschen (Deinstallieren)** eines Plugins mithilfe des Pluginmanagers.

/F09030/ **Verwaltung** der Plugins mithilfe des Pluginmanagers.

4.11 Sonstiges

/F00010/ **Updaten** des Programms über einen Menüpunkt

/F00020/ Einstellen der **Sprache (englisch, deutsch)** **[optional]**

5 Produktdaten

- noch unbekannt – (siehe später Entwicklungsdokumente der Architektur)

6 Produktleistungen

/L00010/ Bei **fehlererzeugenden Eingaben** erhält der Benutzer eine ausführliche **Fehlermeldung**. Nach Fehlern bei der Benutzereingabe hat dieser die **Möglichkeit**, die **Eingabedaten** zu **modifizieren**. Alle Fehler werden in einer zentralen **Log** im Programm gespeichert.

/L00020/ Das Modell sollte **schnellstmöglich** in den Viewports **dargestellt** werden, dass es **keine** unangenehmen **Ruckler** gibt und das Arbeiten damit beeinträchtigt.

/L00030/ Das Editor sollte selbstständig in festgelegten Intervallen eine **Autospeicherung** vornehmen, um im Falle eines Programmabsturzes den **Datenverlust** zu **minimieren**.

7 Benutzeroberfläche

- noch unbekannt – (siehe später Entwicklungsdokumente)

8 Qualitätsbestimmungen

	Sehr wichtig	Wichtig	Weniger wichtig
Robustheit			X
Zuverlässigkeit		X	
Korrektheit		X	
Benutzerfreundlichkeit	X		
Effizienz		X	
Portierbarkeit	X		
Kompatibilität		X	

9 Entwicklungsumgebung

Es wird darauf geachtet, dass alle Entwicklungstools kostenlos (*Freeware*) sind.

9.1 Programmiersprache/IDE

- C++ mithilfe von Eclipse und dem CDT-Plugin

9.2 Bibliotheken

- OpenGL
- Qt

9.3 Tools

- SVN (Subversion) zur Versionsverwaltung
- Doxygen zur HTML-Dokumentations-Generierung

10 Ergänzungen

10.1 Spezielle Projektrollen

10.1.1 Organisatorische Projektleitung: Thomas Kuhndörfer

- Planung von Meilensteinen und Projektfortschritten
- Ansprechpartner des Projektbetreuers

10.1.2 Technische Leitung: Reinhard Jeschull

- Weisungsbefugnis über die Entwickler

10.1.3 Softwarearchitekt: Stefan Zeltner

- Zusammenfügen der einzelnen Projektmodule
- Design von Modulschnittstellen

10.1.4 Softwarearchitekt: Reinhard Jeschull

- Design von Modulschnittstellen, im speziellen OpenGL-Renderer und Mathe-Lib

10.1.5 Portierungsverantwortlicher: Thomas Tischler

- Prüfung der Lauffähigkeit unter Linux / Windows
- Gegebenfalls Anpassungsarbeiten

10.1.6 Dokumentationsverantwortlicher: Philipp Gruber

- Weisungsbefugnis bei Dokumentationsangelegenheiten
- Korrektheit der erstellten Doxygen-Dokumentation sicherstellen