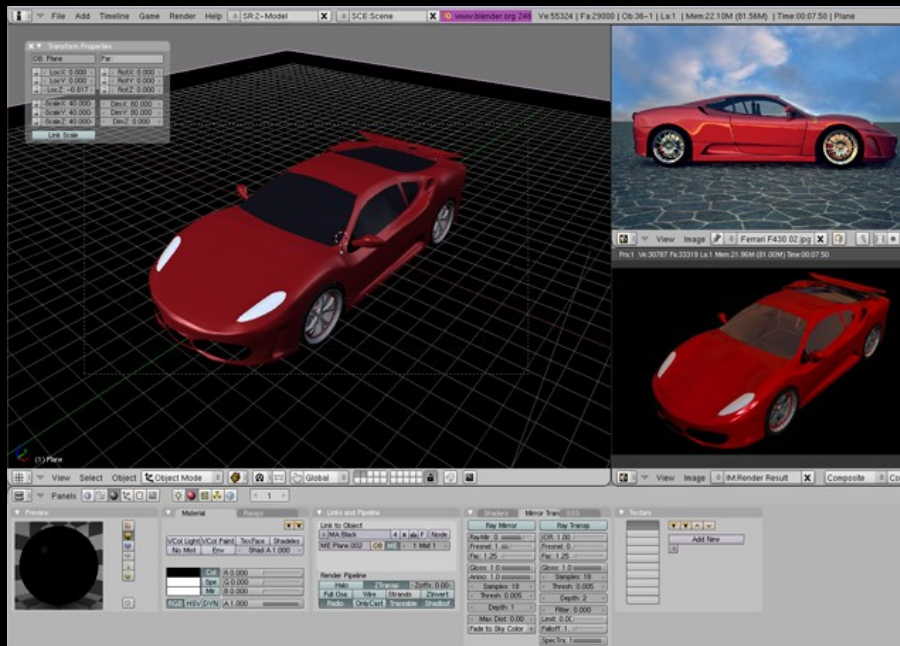




Blender 4 Game Development



Philipp Vath

Sommersemester 2008

Softwarepraktikum

Informatik Uni Freiburg



Übersicht

- Motivation
- 3D-Modelle (Vom Aufbau bis zur Darstellung)
- Blender Modeling Workshop
 - Interface
 - Modeling
 - Texturing
- Export nach XNA
 - OBJ-Formate
 - ContentProcessor



Motivation

Warum 3D-Modelle (in Computerspielen) ?

- 1. Kreativer Spaß und Blickfang:
 - ➔ Programmierem Verhalten Darsteller zuweisen
 - ➔ Das eigene Computerspiel „bevölkern“
Ideen in 3D-Objekte umsetzen und darstellen
- 2. 3D-Grafik als Feature
macht das Spiel im Bereich Look & Feel besser
dreambuildplay.com bewertet bspw. eingesendete Computerspiele mit
 - „40% Fun factor (exciting, entertaining action)“
 - „30% Innovation (push creative and technical limits)“
 - „30% Production Quality (as polished as possible – so aufgemotzt wie möglich)“
- 3. Wissen über 3D-Grafik
für Computerspiele, Simulationen oder andere Echtzeit-Visualisierungen

Spaß und Blickfang



Action: Pyroblazer



Action-Roleplay: Numen



Racing: Need for Speed Pro Street





3D-Grafik als Feature

- Entwicklungszyklus eines Games ohne 3D-Grafik
 - Software programmiert durch Programmierer
 - Spieler (Tutoren? ;-) geben Programmierer Feedback
- Entwicklungszyklus eines Games mit 3D-Grafik (Potentiale und Feedback)
 - + Programmierer hat Möglichkeit der Visualisierung
 - + Artist hat Möglichkeit der Interaktivität/Simulation
 - + Artist und Programmierer geben einander Feedback
 - + Spiel wird anschaulicher und besser
 - Was nützt Fahrverhalten ohne Auto und umgekehrt Auto ohne Fahrverhalten im fertigen Produkt ?
- Woher bekomme ich 3D-Modelle ?
 - online
 - Artist (von außen oder innerhalb des Teams (selbst bauen))
 - von mir ;-)



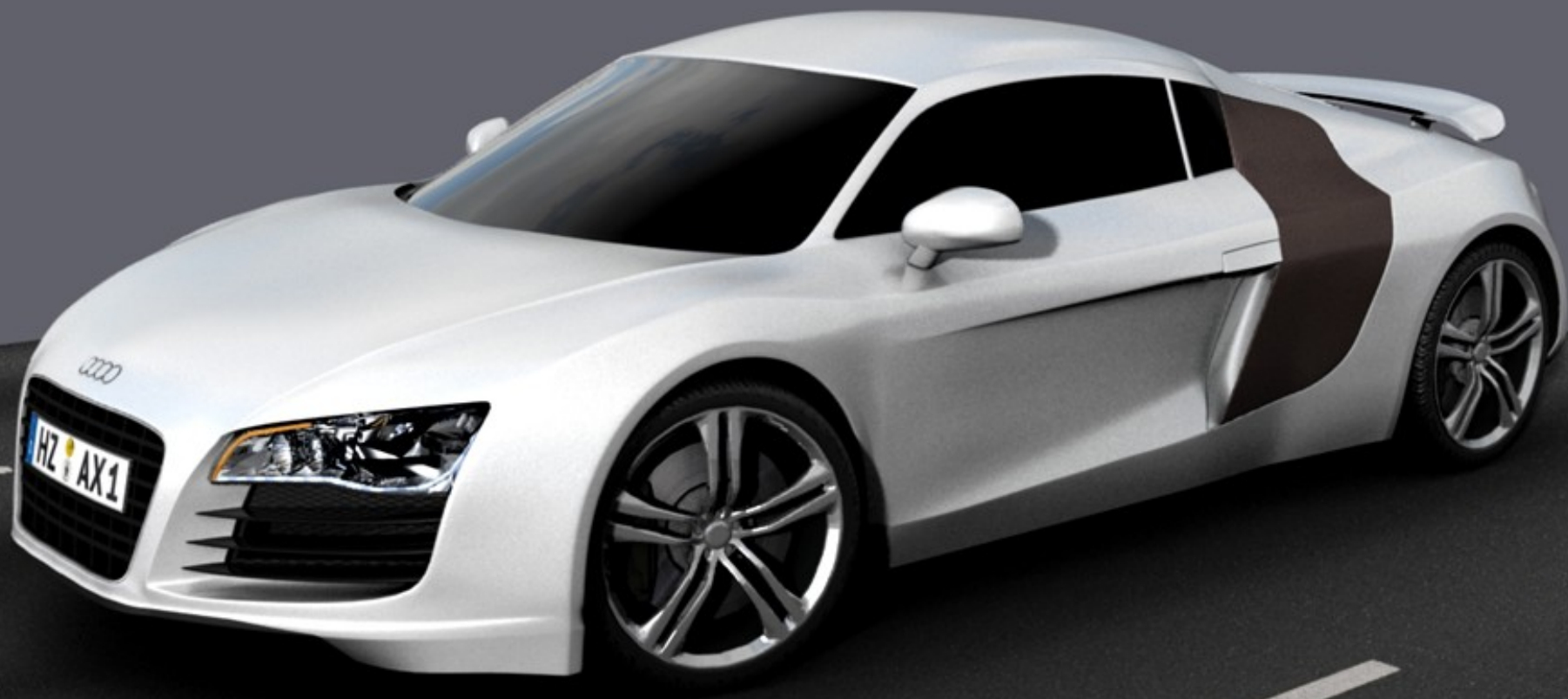
Ferrari F430



(c) Philipp Vath



Audi R8

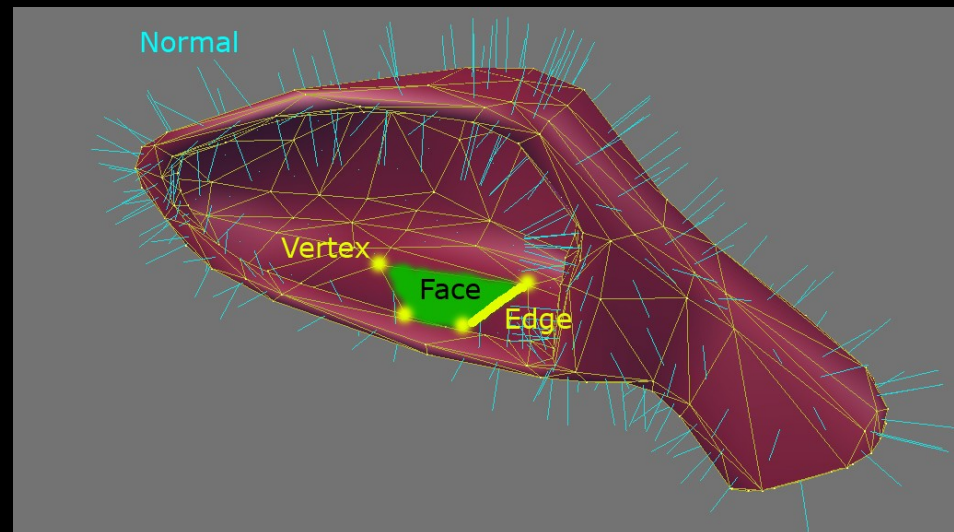


(c) Alex Glawion



3D-Drahtgittermodell

- Was ist ein 3D-Modell ? Woraus ist es aufgebaut ?
-Die Bausteine in der 3D-Grafik:
- 1 Modell: Meshes
- 1 Mesh: Faces (Flächen)
- 1 Face: Edges (Kanten) und Normalen
- 1 Edge: Vertices (Punkte)





Wie kommt mein Modell auf den Bildschirm?

- Wie entstehen die 3D-Modelle in meinem Computerspiel ?

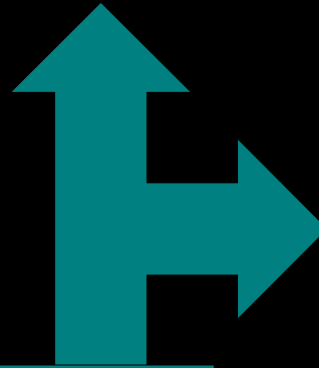
Idee

Design

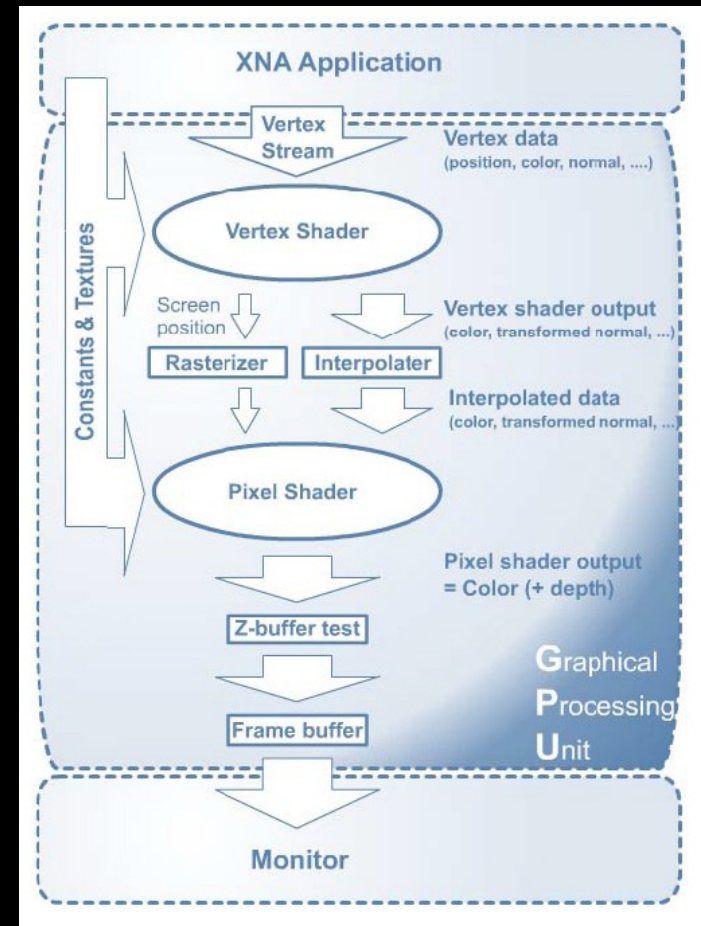
Geometrie

Modeling

Shading/Texturing



Rendering-Pipeline:





Übersicht Teil 1: Modeling Workshop

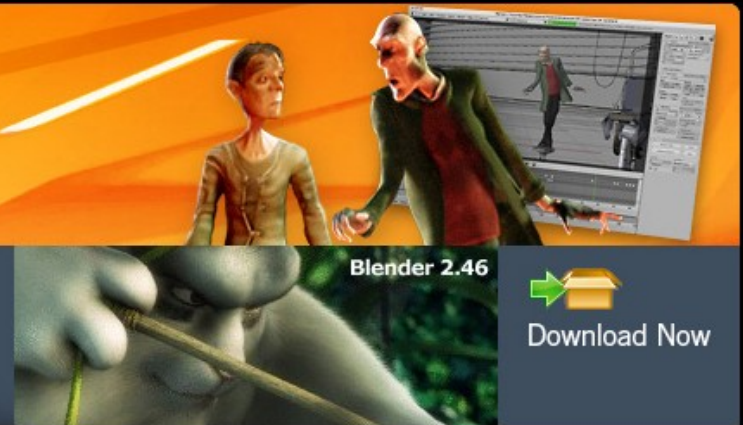
- 3D Blender
- Interface
- Blueprints
- Modeling
- Texturing



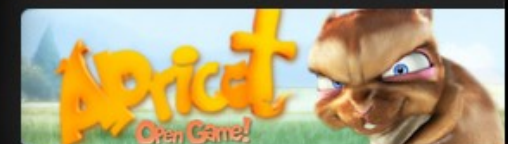
Blender

model · shade · animate · render · composite · interactive 3d

Blender is the free open source 3D content creation suite, available for all major operating systems under the **GNU General Public License**.



- Womit erzeuge ich mein 3D-Modell ?
 - 3D Creation Suite Blender
 - Download von www.blender.org
- Warum Blender ?
 - Kostenlos, OpenSource
 - Leistungsstark in Modeling, Texturing, Animation, Export und Rendering
 - Große Community, hilfreiches Forum und umfassende Dokumentation
 - Sehr schneller Zuwachs an neuen Features
 - Game-like Workflow
 - Starker UV-Editor
- Blender OpenSource
Creative Commons License Gedanke



Art Gallery

Forums

Get Involved

Foundation / Institute



Idee für ein Modell

- Idee
 - Was für ein Modell (Charakter, Maschine, Pflanze ?)
 - Maschine: Science Fiction Vehicle oder Auto ?
 - Aussehen und Form
- Blueprint
 - Erstellen oder Downloaden und Anpassen (PodRacer)



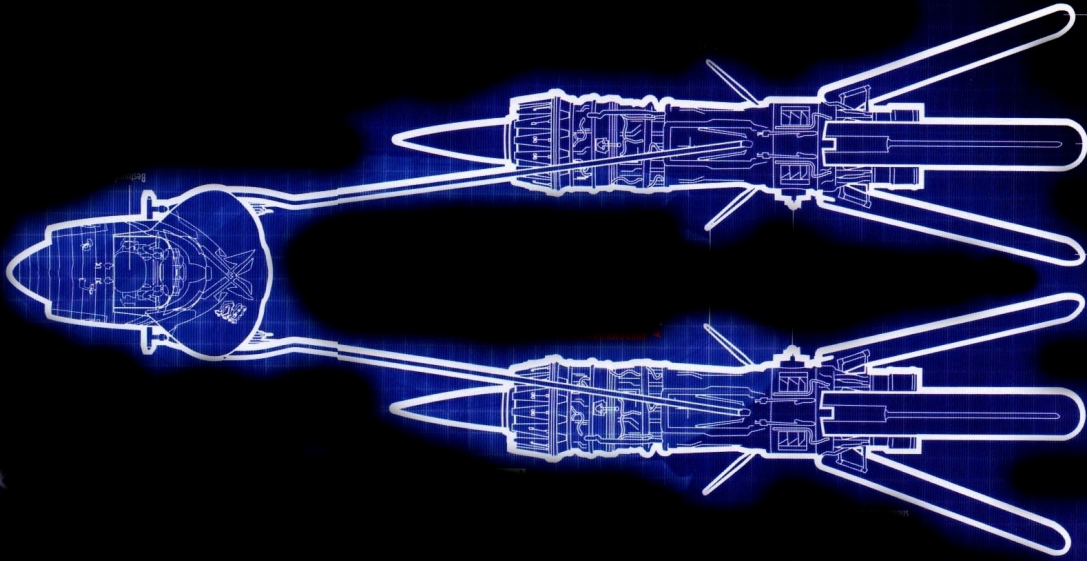


Blueprints

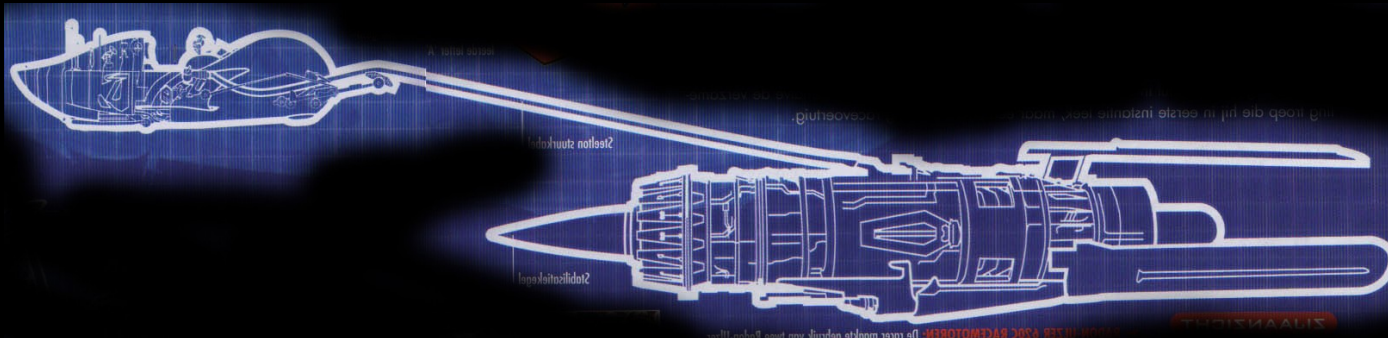
Workshop-Dateien online:

<http://home.arcor.de/designo477/workshop/blender.rar> bzw.

<http://home.arcor.de/designo477/workshop/blenderBig.rar>



Top



Side



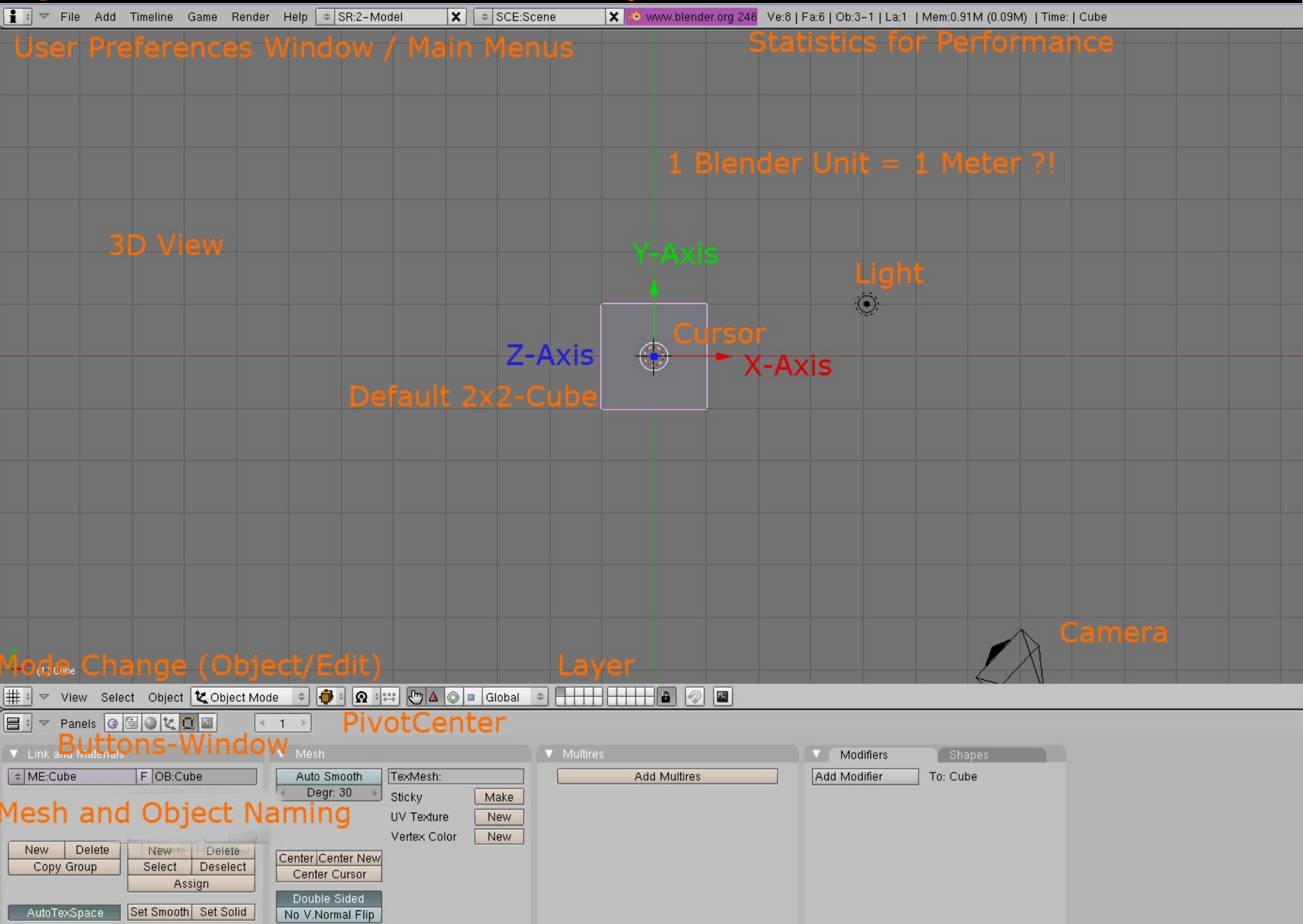
Front



Benennen der Objekte

- 3D-Objekte müssen später im Code eindeutig identifizierbar sein
 - wichtig, sobald mehr als 1 Objekt vorhanden ist
- Eventuell sollen Kollegen mit der Datei besser umgehen können ?
- Namen für alle einzelnen Objekte und Materialien vergeben
 - Chassis
 - Connector
 - Jet
 - JetInner
 - L,R etc. für links rechts

Quickstart: Interface in Top-View





Quickstart: Interface und Navigation

- Interface
 - Bedient sich nach Eingewöhnung wie ein Game!
 - Window-Areas (Anpassbarkeit)
 - Window-Types
 - Preferences: Autosave, Laptop-Benutzer: Emulate Numpad
 - Save User-Preferences (Strg+U)
- Navigation (View)
 - Rotieren: MMB
 - Pan: Shift+MMB
 - Zoom: Scrollrad
 - Front: Num1 bzw. 1
 - Right: Num3 bzw. 3
 - Top: Num7 bzw. 7
 - Rear, Left, Back mit Strg+1, Strg+3, Strg+7
 - Camera: Num0 bzw. 0
- Tutorials für Anfänger
 - QuickStart Guide: wiki.blender.org/index.php/QuickStart
 - Hotkey Referenz: www.blender.org/education-help/



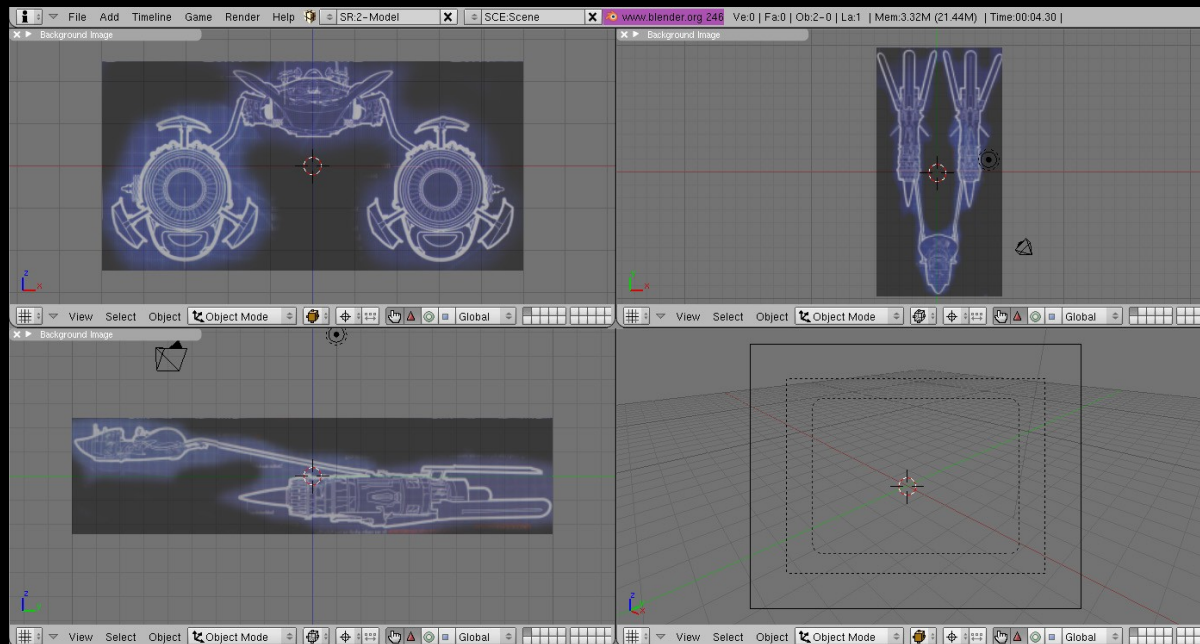
Basics: Objekte in Blender

- Selektieren
- Transformieren
 - Transform Widget
 - Hotkeys: Grab G, Rotate R, Scale S
 - Transform Properties Window (N)
- Center
 - Auf Cursor verschieben, Neu berechnen, Objekt zum Center schieben
 - Wichtig: Center des Hauptobjektes bei (0,0,0)
 - Als Pivot benutzen
- Cursor verschieben
 - Einfügepunkt
 - Als Pivotpunkt benutzen
 - Reset auf (0,0,0)
- Objekte
 - Hinzufügen (Plane, Cube, UVSphere, Cylinder, Monkey)
 - Löschen



Blueprints einfügen

- Fenster in 4 Fenster unterteilen
 - Strg+Up bringt fokussiertes Window auf „FullScreen“
- Als Hintergrund hinzufügen
- Anpassen und Skalieren





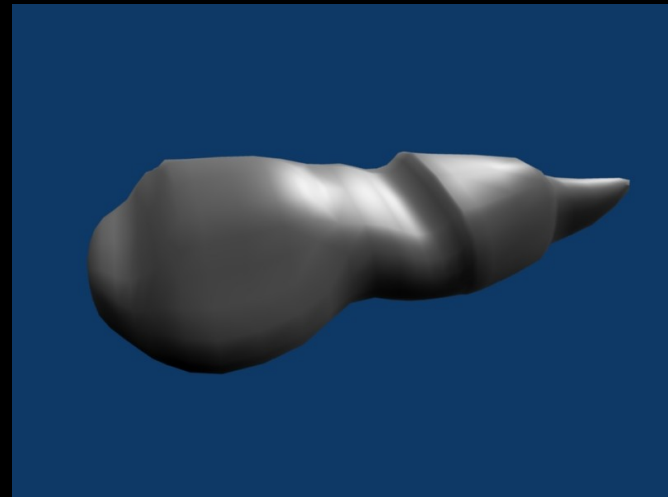
Basics: Box Modeling

- Idee: Aus einer Box mehrere Boxen extrudieren und solange deformieren oder zusammenfügen, bis fertiges Modell entsteht
- Editmode vs. Object Mode (Deformieren vs. Transformieren)
- Selektionsmodi
 - Box select (B) , Circle select (BB)
 - Vertex, Edge, Face select mode
- Extrude (Geometrie aus vorhandener erweitern)
 - Standard: Extrude Region + Grab
 - Schrumpfen / Expandieren: Extrude Region + Scale
 - Eindellen: Extrude + Rotate
- Dem Blueprint anpassen (Grab, Rotate, Scale)
 - Achsen locken (durch entsprechenden Key (X, Y, Z))
- Create Vertex, Face, Edge (Strg+LMB, F)



Advanced: Box Modeling und Rendern

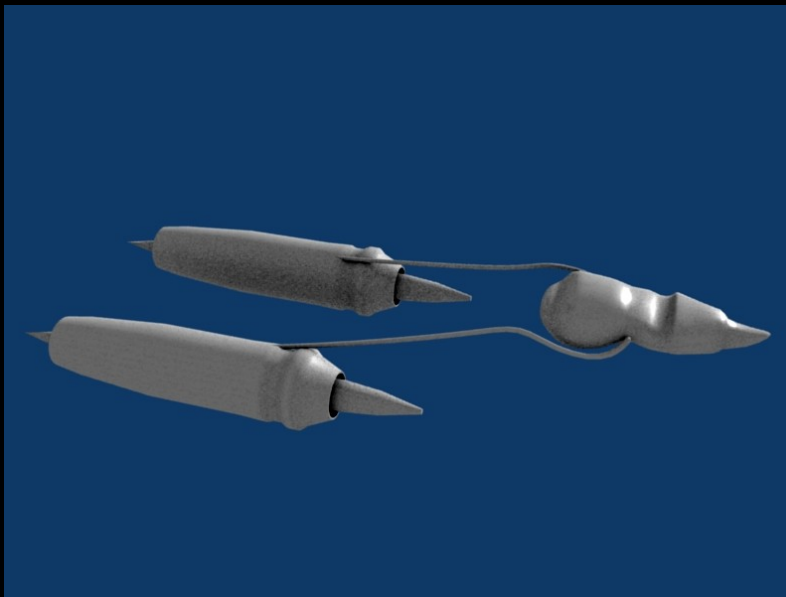
- Specials-Menü (W)
 - Verschmelzen zweier Vertices (W, M) oder (Alt+M)
- Am häufigsten: Alles abwählen (A) + Box select (B)
- Detail (Kanten hinzufügen):
 - (Loop) Cuts (K)
 - Subsurf-Modifier
- Kamera einstellen
 - In Kamerasicht:
 - Shift-F: W,S,A,D,R,F,
 - Mausbewegung & Scrollrad
- Rendern (F12)
 - Auflösung
 - Anti-Aliasing
 - Ausgabeformat





Advanced II: Modeling und Rendering

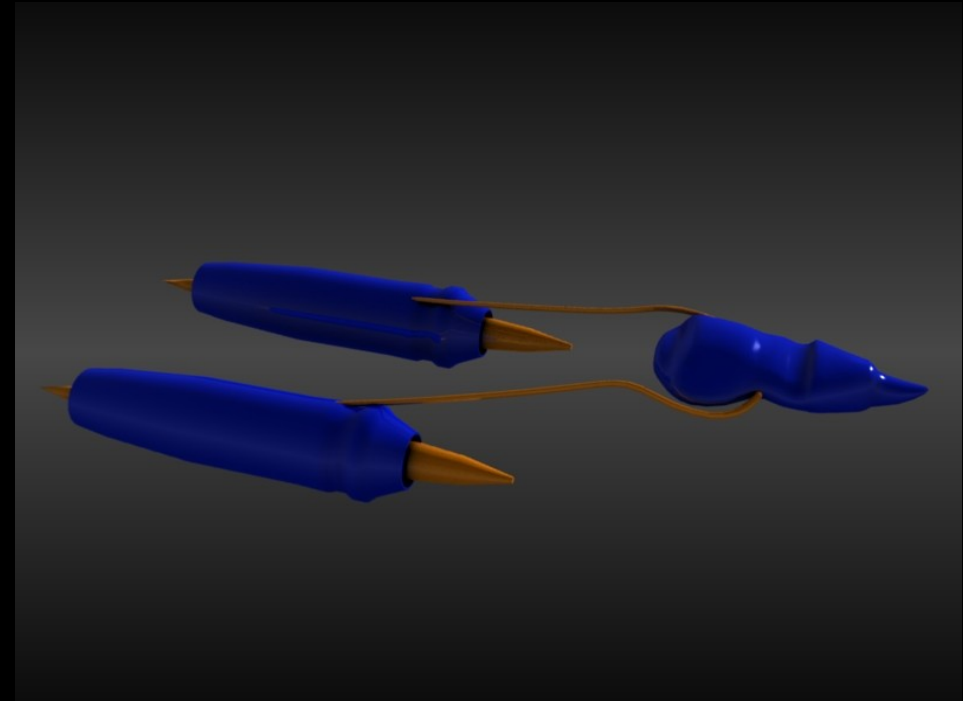
- Mirror modifier (symmetrisches Modellieren)
- Details hinzufügen durch Cuts
- Schattieren mithilfe von Ambient Occlusion
 - Objekte beleuchten und schattieren sich gegenseitig





Einfaches Shading mit Materials

- Material erstellen, Bilder laden
- Farbe (Col)
 - Color picker
- Reflektionsstärke (Ref)
- Glanzlicht (Spec)
- Härte (Hard)
- RayMirror (Ray Tracing for Mirror Reflection)



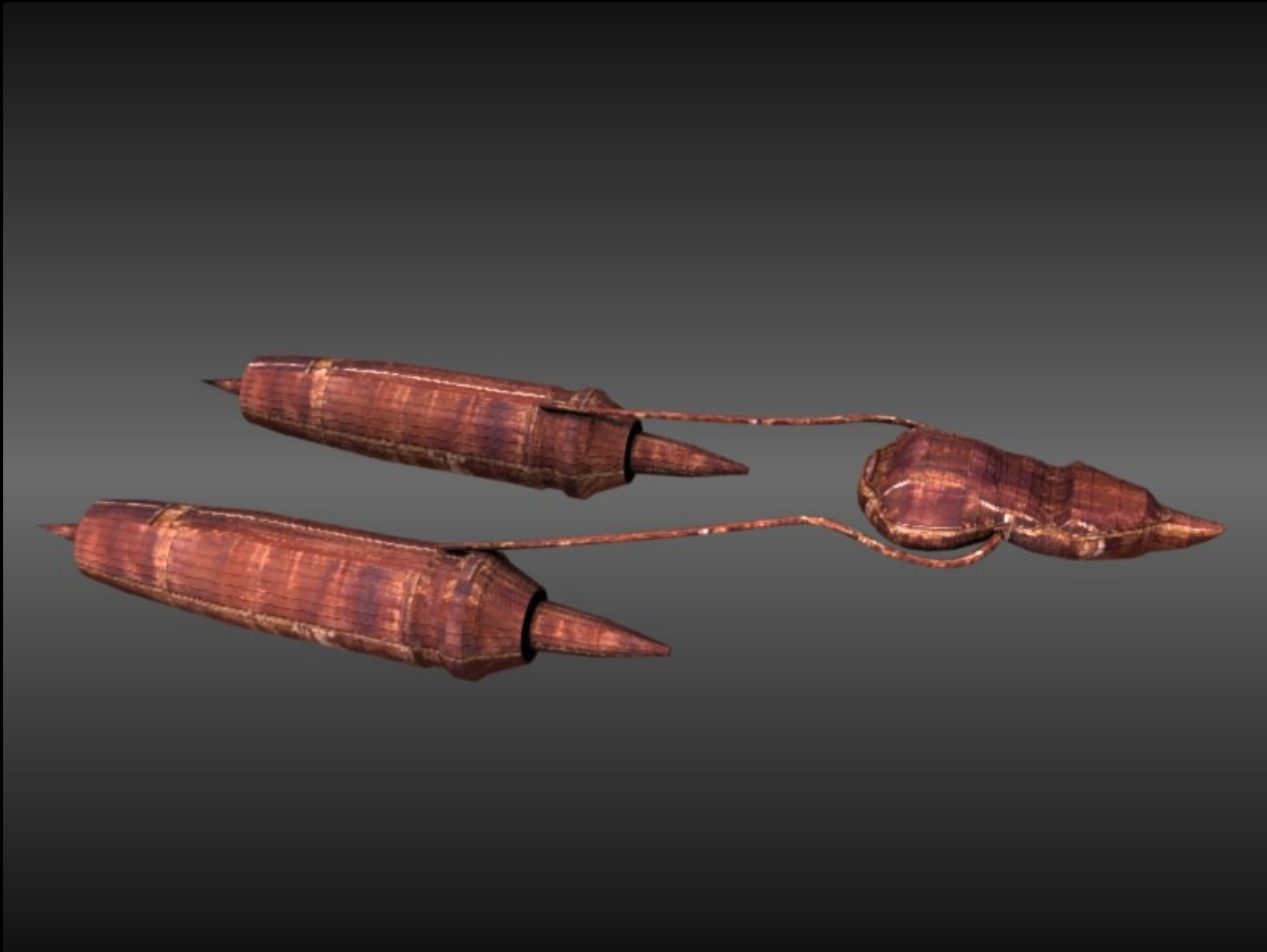


Texturing

- Textur für Material erstellen
 - Zum Material hinzufügen
 - Bild laden
 - Projektion festlegen (häufig: Orco Cube oder UV Flat)
 - Welchen Zweck erfüllt die Textur, d.h. auf welchen Shader-Parameter wird sie gemappt ? (MapTo)
 - Farbe (Col)
 - Oberflächenrauheit, Bump-Mapping (Nor)
 - Heightmap (2D SW => 2D Höhen/Tiefen)
 - Normalmap (3D RGB => 3D Normalenvektoren)
 - Alpha-Maske (Stencil)
 - Reflektionsstärke (Ref)
 - Glanzlichstärke (Spec)
 - Transparenz (Alpha)
- Workshop: Texturen für Farbe und Bump-Mapping



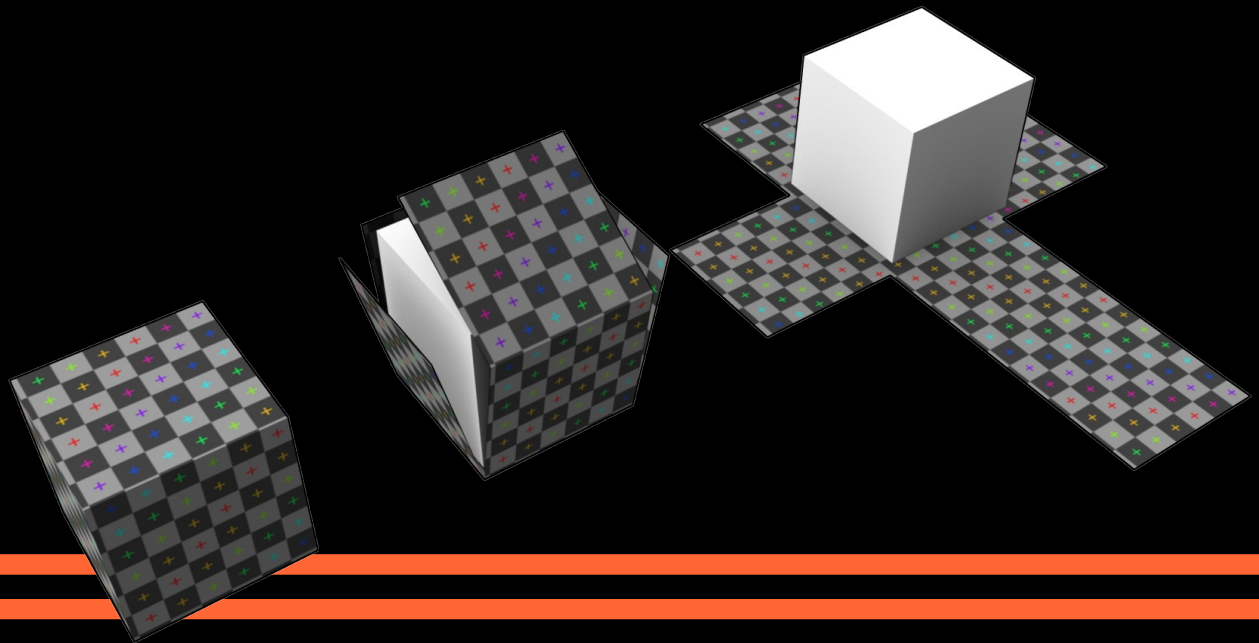
Farb- und Bumpmapping





UV Texturen

- UV Texturkoordinaten
 - Jedem 3D-Vertex wird eine 2D-Texturcoordinate zugewiesen, dazwischen wird interpoliert
- UV Unwrapping
 - Festlegen der UV-Texturkoordinaten (Mapping) gemäß einer Projektionsmethode
 - Cube Projection
 - Project from View
 - Smart Projection



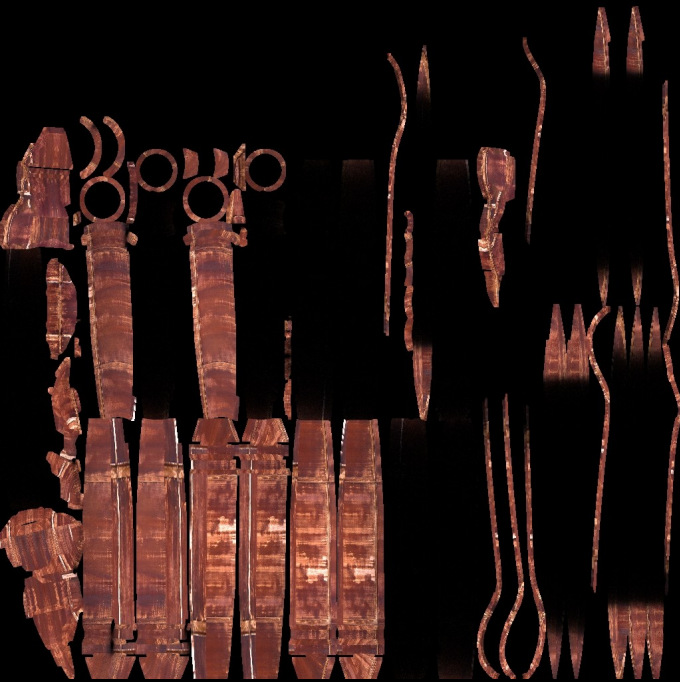


Baking Textures

- Modifier anwenden
- Mehrere Objekte in 1 umwandeln (optional) (Strg+J)
- Unwrappen (U)
- Normalen neu berechnen (Strg+N)
- Baken der Texturen
 - Colormap: Farben der Oberfläche
 - (Height-/) Bumpmap (2D):
 - Relief der Oberfläche (Höhen und Tiefen)
 - Normalmap (3D):
 - Noch detaillierteres Oberflächenrelief (3D statt 2D)
- „Selected To Active“-Button: Baken auf anderes (einfacheres) Mesh

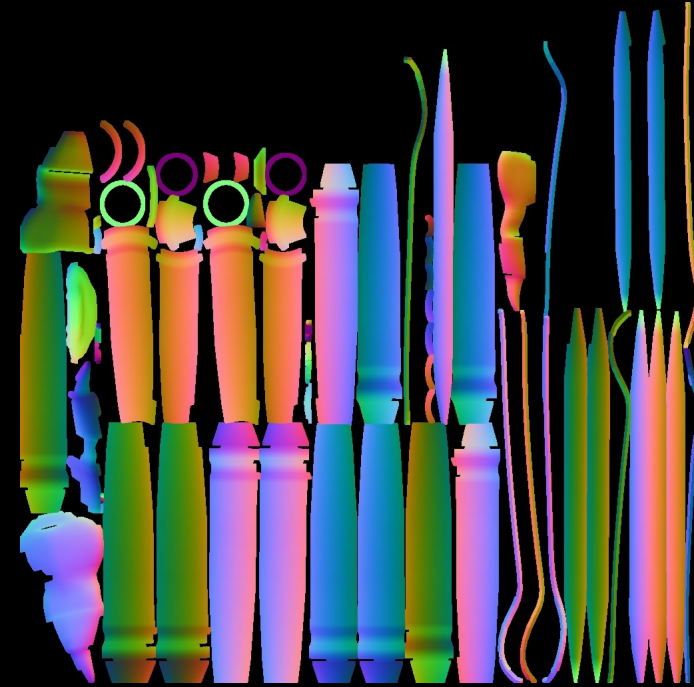
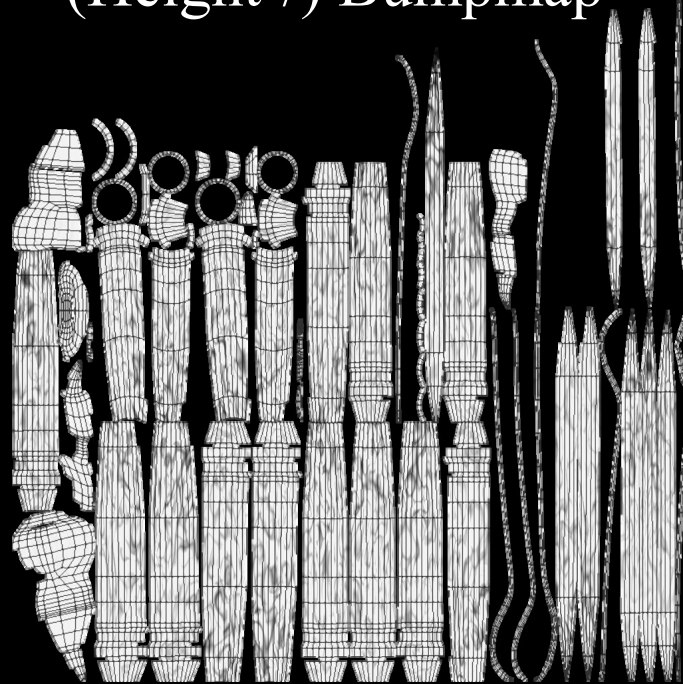


Baked Textures



Colormap

(Height-/) Bumpmap



Normalmap



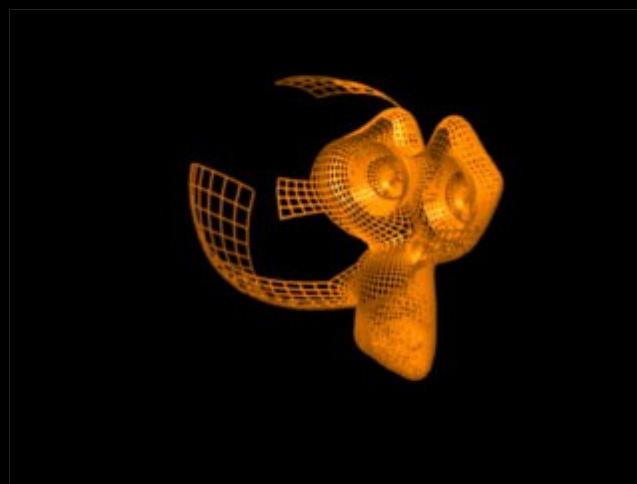
Übersicht Teil 2: Export

- Dateiformate
- OBJ-Formate und Wings 3D
- OBJ und FBX Content Processor für XNA
- Am besten FBX (direkt XNA kompatibel)
- Laden und Rendern in XNA

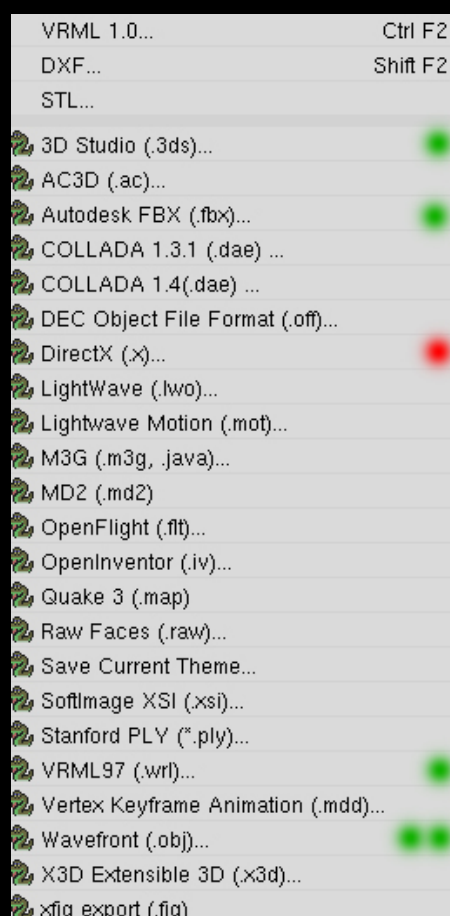


Blender Import und Export

- Import



- Export





OBJ Format(e)

- Blender: Export als OBJ
 - Triangularisieren und Normalen exportieren !
- Intuitiv definiert:
 - Material Dateilink
 - Vertices (v)
 - 3D-Vektoren
 - Vertex UV-Texturkoordinaten (vt)
 - 2D-Vektoren
 - Vertex normals (vn)
 - 3D-Vektoren
 - faces (f)
 - v1//vn1 v2//vn2 v3//vn3
 - jedoch manchmal auch v1/vn1 v2/vn2 v3/vn3
 - bzw. v1/vt1/vn1 v2/vt2/vn2 v3/vt3/vn3
 - sollten Triangles sein statt Quads !
- Es existieren also verschiedene Formate
 - Verschiedene Tools ausprobieren
 - Dateiformat Parser schreiben



OBJ und FBX > Content Processor > XNA

- OBJ-Importer = Custom Model Importer von XNA
 - Download: <http://creators.xna.com/en-us/sample/custommodelimporter>
- Blender:
 - Triangulate
 - export Normals
 - don't copy images (buggy script)
- Editor:
 - In der OBJ - .mtl Materialdatei müssen alle „Ni“ zu „Ns“ Parametern konvertiert werden
- FBX-Importer = Integrierter Content Processor von XNA
- Blender:
 - Export Mesh
 - Apply all modifiers
 - Export HQ Normals



Content Processor konfigurieren

- Bedienung
 - vollautomatisch
 - daher optional
 - außerdem weitgehend selbsterklärend
- Parameter
 - Dateitypen
 - Komprimierungsstufen
 - Pfade
 - Farbeinstellungen
- Verschiedene Content Processors:
 - Model Loader
 - Texture Loader
 - Effect Loader



Car Showcase

- Model Viewer
- Generische Struktur:
 - Actor
 - Model
 - Meshes
 - Materials
 - Textures
- Starte Demo ...



Quellen und Kontakt

- <http://www.blender.org>
- <http://www.dreambuildplay.com>
- <http://www.pcgames.de/>
- http://en.wikipedia.org/wiki/UV_mapping
- http://de.wikipedia.org/wiki/Bump_mapping

Bei Fragen zu Blender:

E-Mail an: Philipp.Vath at jupiter.uni-freiburg.de



Danke für eure Aufmerksamkeit



visit www.alphredo.com to see yet another blender open movie project