COMPUTERGRAFIK 2014 · UNIVERSITÄT OSNABRÜCK

Niels Meyering

WAS IST BLENDER?

- 3D-Grafik-Software
- Open Source (GPL-lizenziert)

FEATURES

- Modellierung
- Animation
- Rendering (zwei eingebaute Renderer)
- Compositing

- Partikelsysteme
- Physik-Simulation
  - Fluid
  - Cloth
  - Solid Physics
  - Rauch

- Python-Scripting
- Im- und Export
- Videoschnitt
- Game-Engine

u.v.m.

EINORDNUNG

CINEMA 4D

- MAXON Computer GmbH (Friedrichsdorf)
- kommerziell; Studio-Version 595€ (Stand Juli 2014, maxonshop.de)
- kostenlose Demo-Version für Studenten verfügbar
- siehe auch CG-Vorlesung 2010

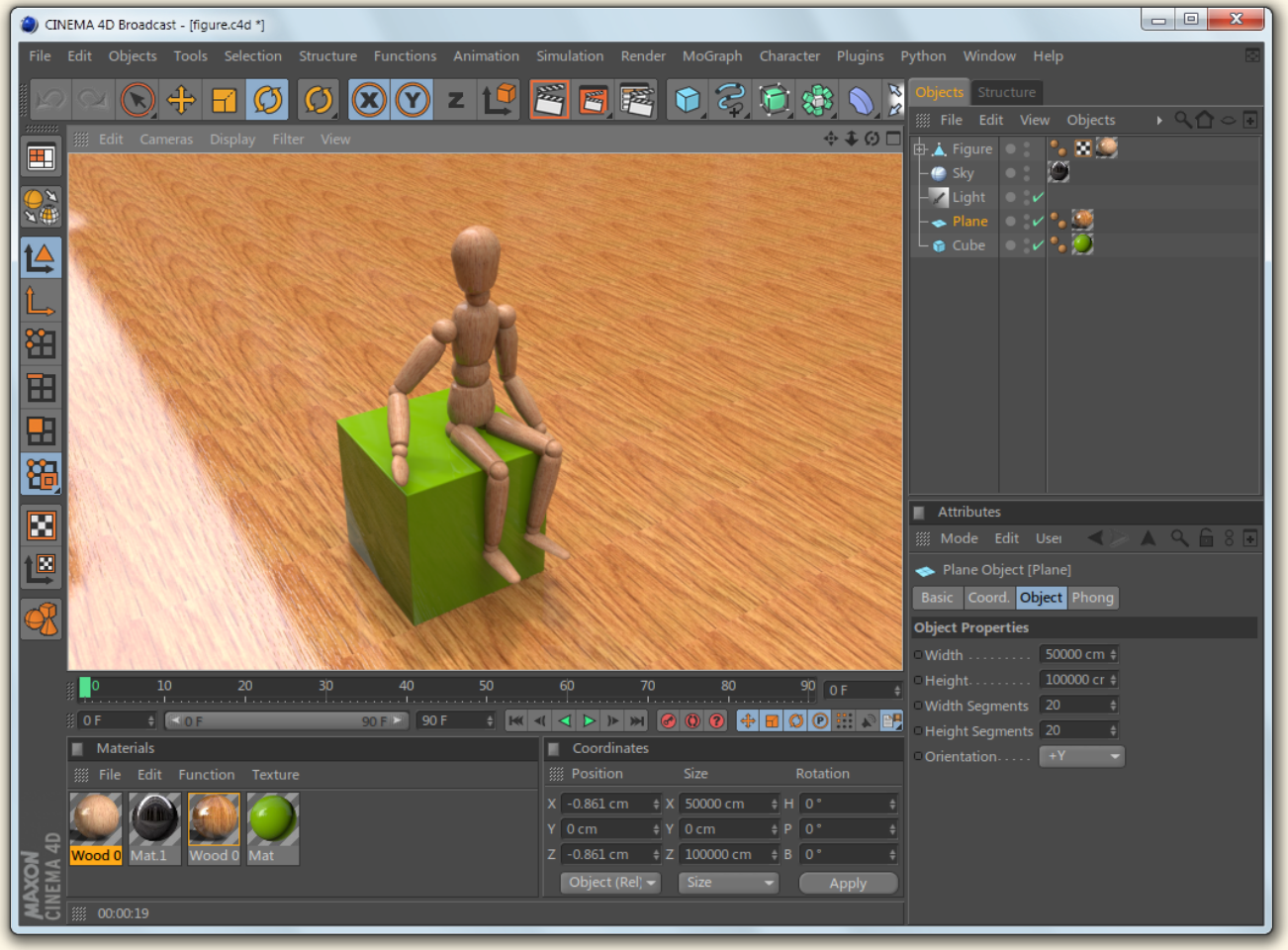


Bild: Wikipedia

3DS MAX

- Autodesk, Inc.
- kommerziell; ab 3.900€, 195€/Monat (Stand Juli 2014, autodesk.de)

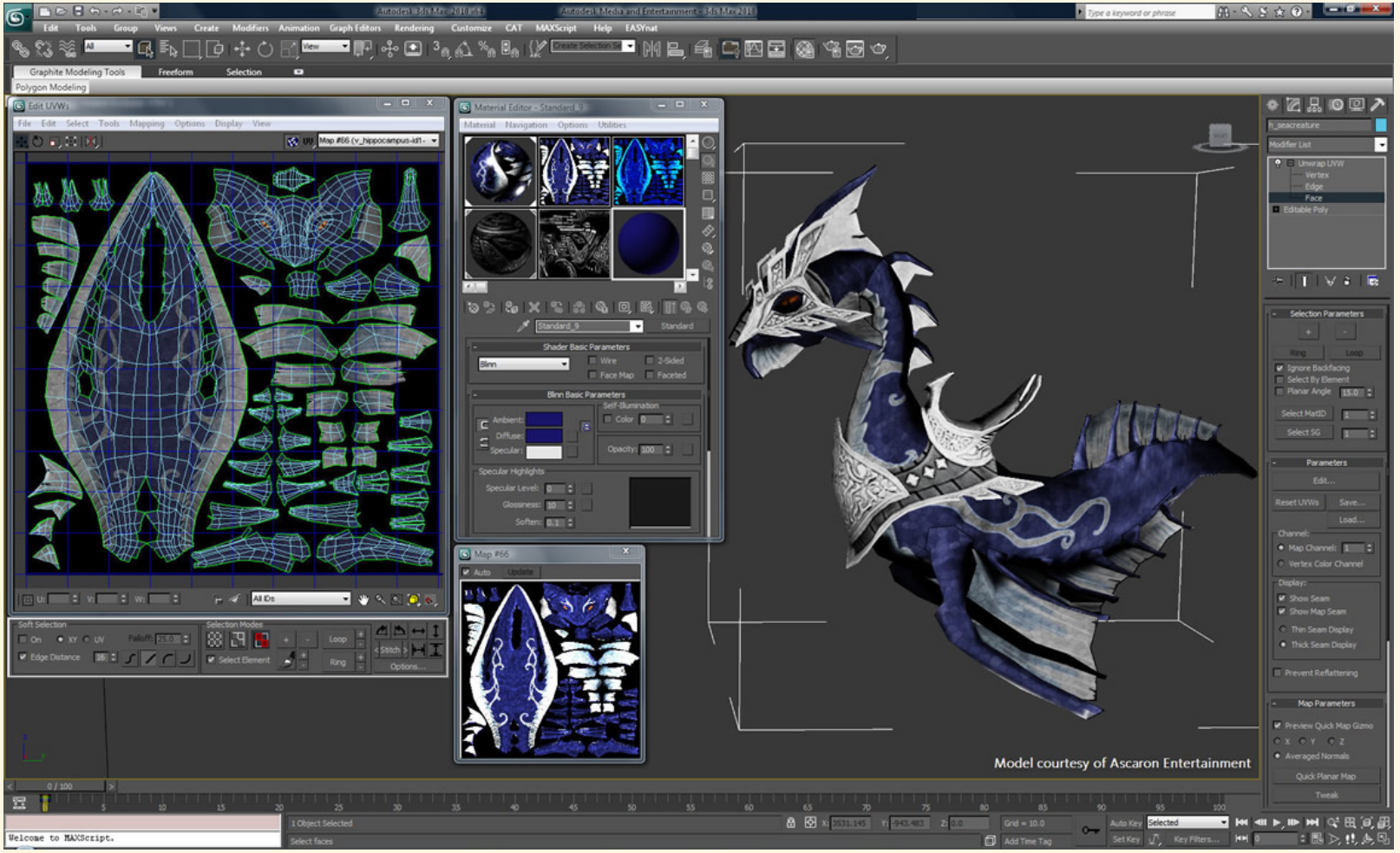


Bild: Autodesk

MAYA

- Autodesk, Inc.
- kommerziell; ab 6.000€, 195€/Monat (Stand Juli 2014, autodesk.de)

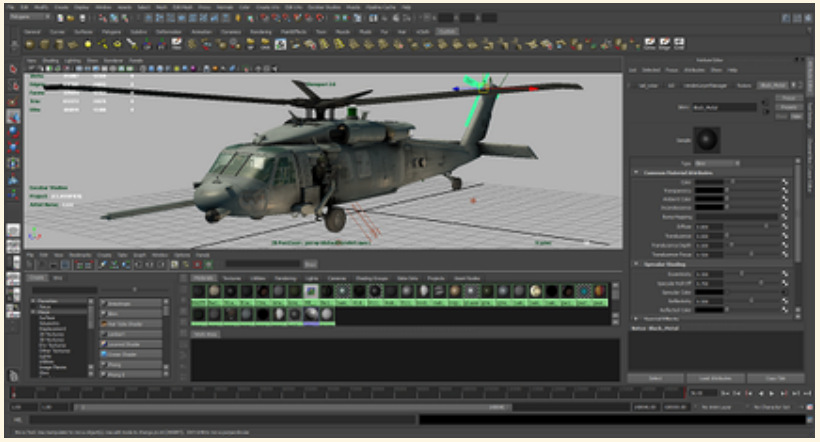


Bild: Wikipedia

LIGHTWAVE 3D

- NewTek, Inc.
- kommerziell; $1495 USD (Stand Juli 2014, lightwave3d.com)

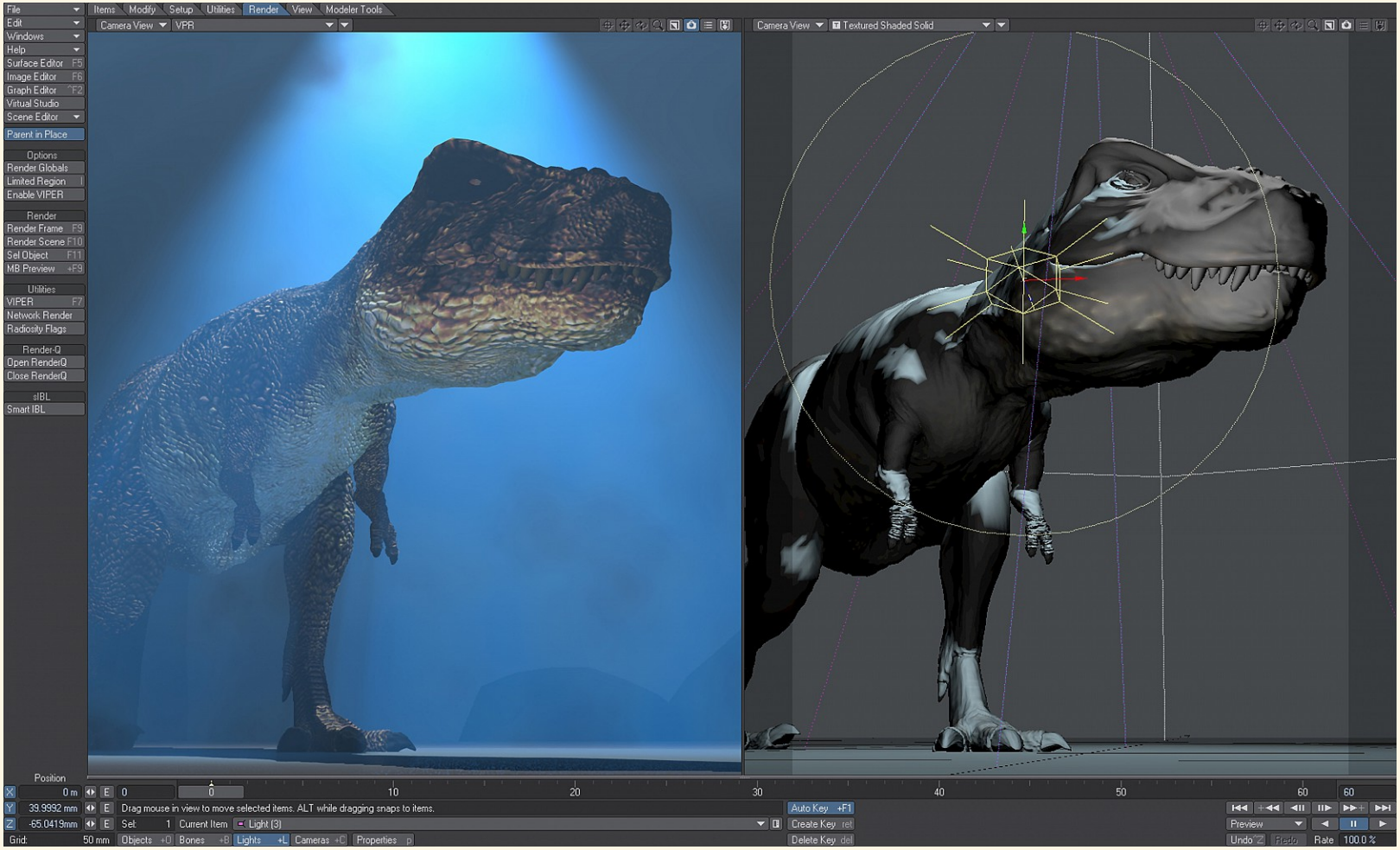
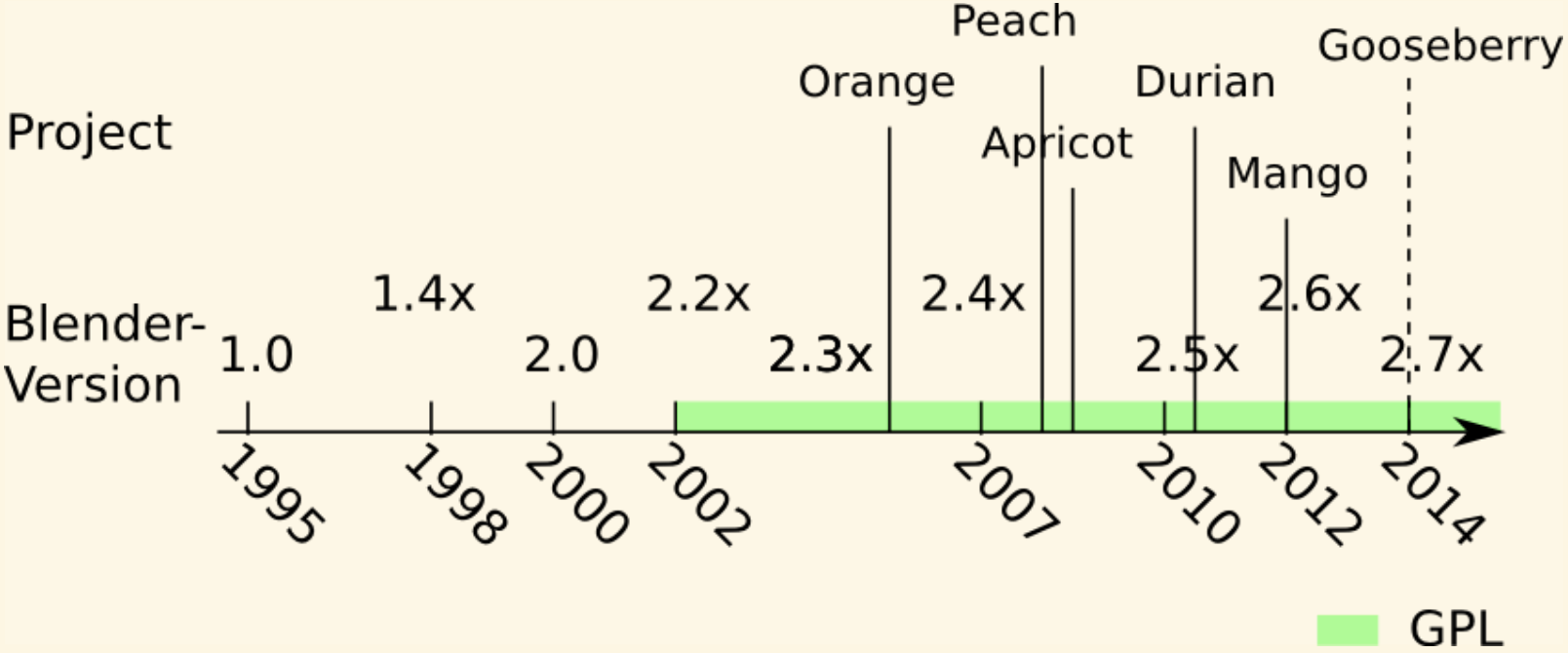
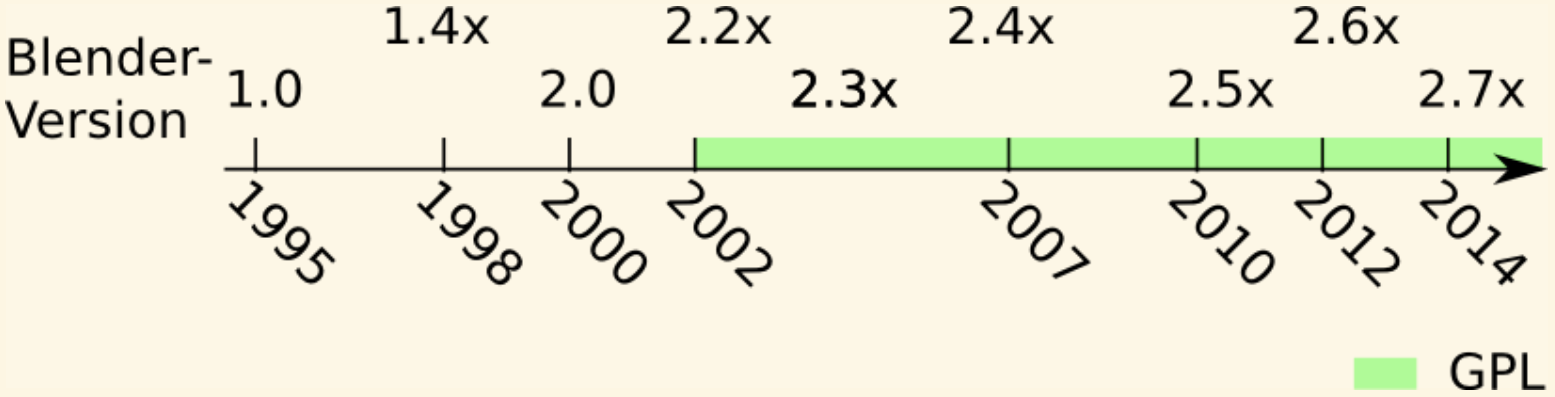

File
Edit
Windows
Help
Surface Editor F5
Image Editor F6
Graph Editor ^F2
Virtual Studio
Scene Editor
Parent in Place
Options
Render Globals
Limited Region
Enable VIPER
Render
Render Frame F9
Render Scene F10
Sel Object F11
MB Preview +F9
Utilities
VIPER F7
Network Render
Radiosity Flags
Render-Q
Open RenderQ
Close RenderQ
sIBL
Smart IBL
Items
Modify
Setup
Utilities
Render
View
Modeler Tools
Camera View
VPR
Camera View
Textured Shaded Solid
Position
Drag mouse in view to move selected items. ALT while dragging snaps to items.
Auto Key +F1
Create Key ret
Delete Key del
Preview
Undo ^Z
Redo
Rate
100.0 %
Grid:
50 mm
Objects +O
Bones +B
Lights +L
Cameras +C
Properties p

Bild: down.cd

BLENDER-GESCHICHTE

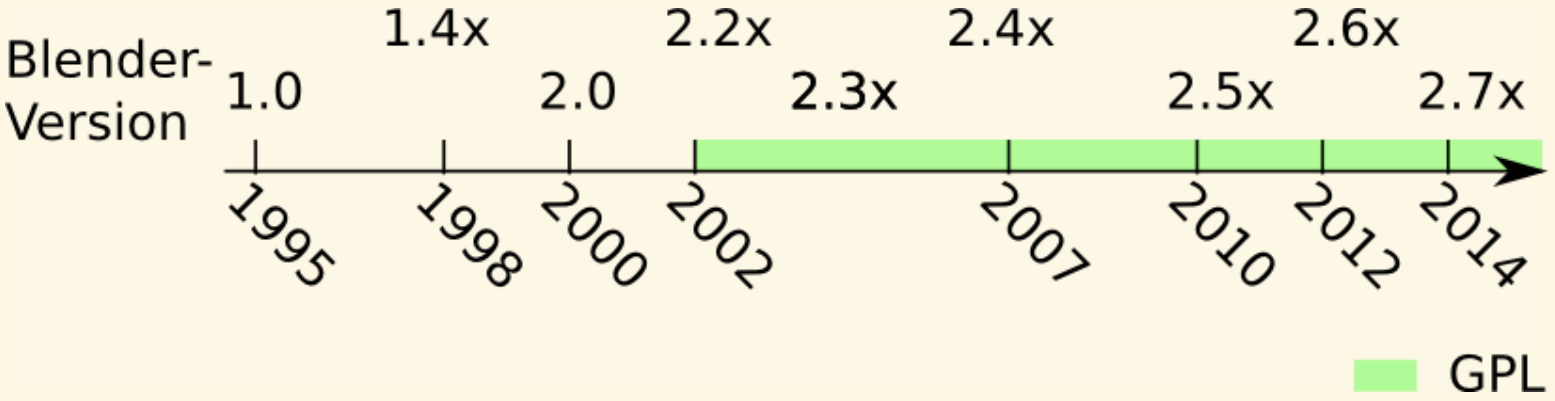


1995



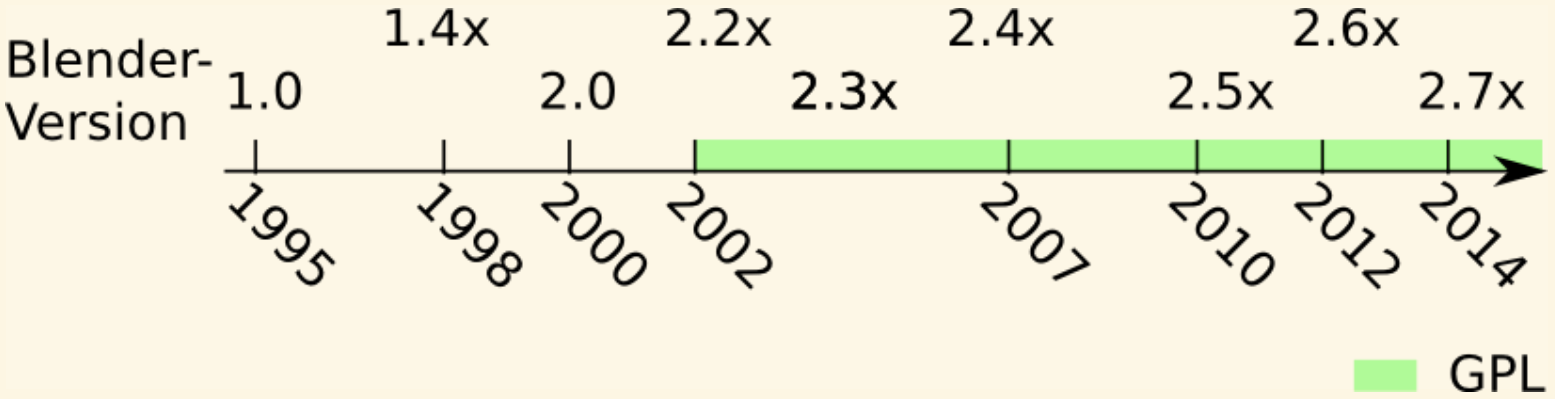
- Ton Roosendaal bei Animationsstudio NeoGeo (Niederlande)
- interne 3D-Modeling Software wird neu geschrieben

1998



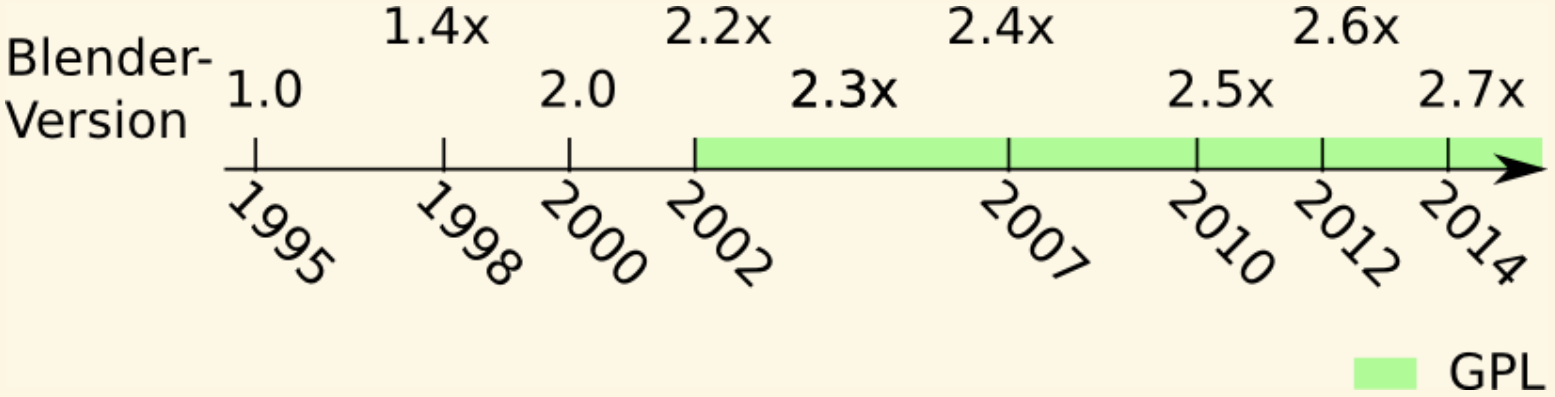
- Ton Roosendaal gründet Not A Number (NaN)
- Ziel: Blender-Entwicklung fördern
- Finanzierung durch kommerzielle Dienste im Umfeld von Blender

2000



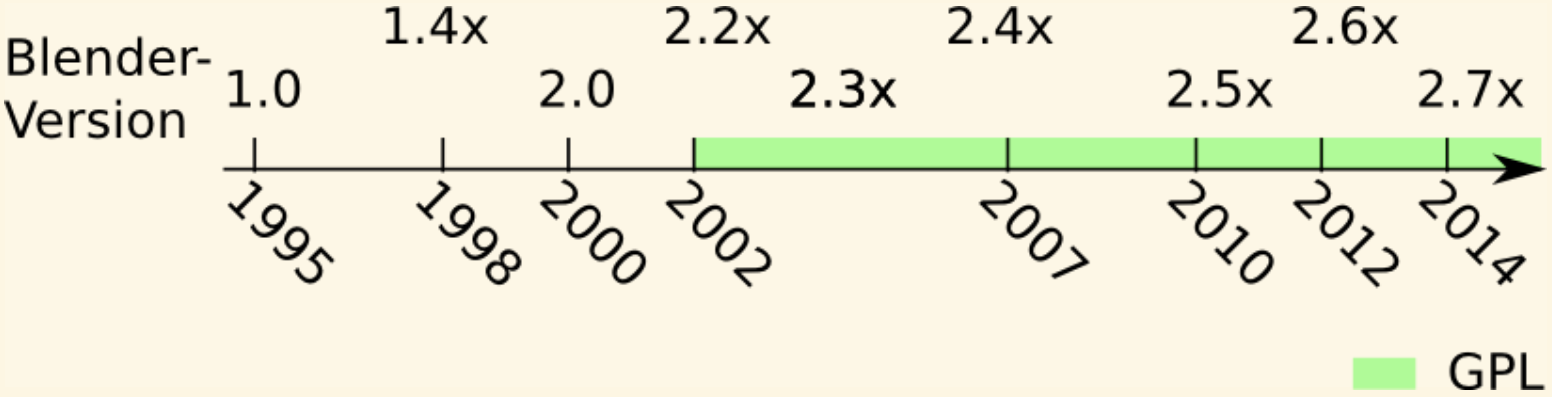
- neue Initiative für ein freies 3D-Tool mit kommerziellen Versionen
- finanziert durch Investment-Gelder

2002



- NaN schließt wegen finanzieller Schwierigkeiten
- Gründung der **Blender Foundation**
- Ziel: Open Source Blender
- "Free Blender"-Kampagne sammelt 100.000€
- 13.10.2002: **Blender released unter GPL**

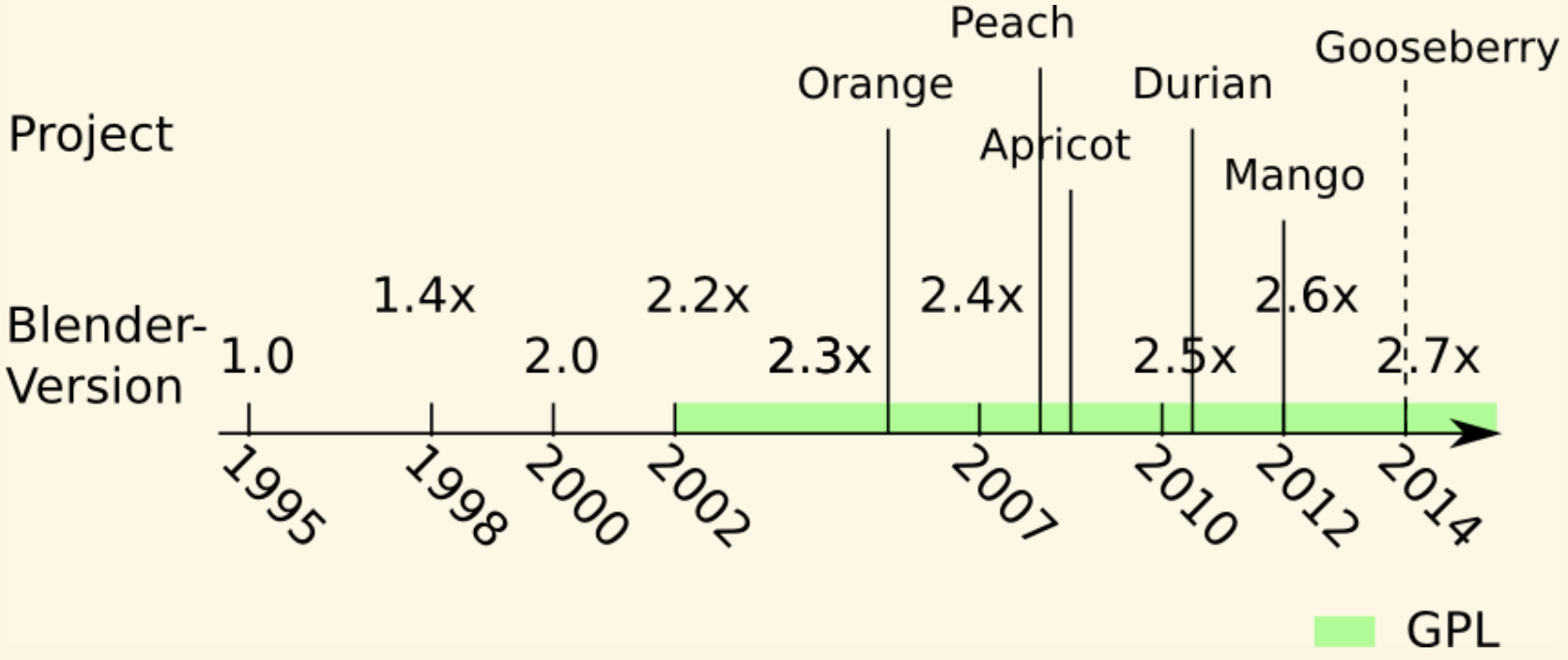
2007



- Gründung: **Blender Institute**
- permanenter Firmensitz und Studio
- kommerzielle Seite der Blender Foundation
- u.a. Organisation weiterer "Open Projects"

OPEN PROJECTS

- organisiert von Blender Foundation
- treiben Entwicklung und Innovation voran
- Werbung für Blender
- Film und Assets üblicherweise unter Creative Commons verfügbar



2005

PROJECT ORANGE, 'ELEPHANTS DREAM'

- Blender 2.37 / 2.4x
- **neues** Charakter-Animationssystem
- **neues** Node-basiertes Compositing
- Renderer-Upgrades

2008

PROJECT PEACH, 'BIG BUCK BUNNY'

- Blender 2.4x
- Haar-/Fell-Modellierung und Rendering
- Tools für Cartooncharakter
- Performance

PROJECT APRICOT, 'YO FRANKIE!'

- Game-Engine
- Blender als Tool in Pipeline für andere Game-Engines

2010

PROJECT DURIAN, 'SINTEL'

- Blender 2.5
- Sculpting
- Global Illumination

2012

PROJECT MANGO, 'TEARS OF STEEL'

- Blender 2.6
- Motion Tracking
- Camera Solver
- Masking
- Cycles
- Color Grading

2014

PROJECT GOOSEBERRY

- Blender 2.7
- Spielfilmlänge
- internationale Kooperation
- Development-Ziele
  - Blender Cloud
  - Asset Management
  - performante Physiksimulation

DEMO

WEITERFÜHRENDE LINKS

- http://www.blender.org/get-involved/ - alle Links für Interessierte zum Einstieg
- http://wiki.blender.org/ - zentraler Einstiegspunkt
  - Dokumentation, Tutorials: Blender, Python-API
  - Release Notes
- http://blenderartists.org/forum/
- Blender Cookie - Video-Tutorials, Workshops, Tools; kostenlos und kostenpflichtig
- http://blendtuts.com/ - Video-Tutorials
- http://www.blendswap.com/ - Models und Blender-Szenen aus der Community

Computergrafik 2014

Lukas Kalbertodt

Inhalte

- Spiel-Engine Grundlagen
- Was ist Unity?
- Live-Vorführung

Was ist eine Spiel-Engine?

Was ist eine Spiel-Engine?


```
if(world.isRaining()) {
  car.getMaterial().setSpecularExp(128);
}
```

```
glGenBuffers(1, &handle);
glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, handle);
GLenum error = glGetError();
```

```
sub esp, 4
mov eax, [ebp+8]
mov esp, ebp
pop ebp
```

Was ist eine Spiel-Engine?

- Abstrahiert technische Aspekte
  → Programmieren auf höherem Level
- Übernimmt performancekritische Aufgaben
- Bringt meist große Funktionalität mit
  - Physiksimulation
  - Realistische Grafik
  - Künstliche Intelligenz
- Soll Künstlern Freiheit geben
  - Gestaltung von dem Level, Charakteren, …

Werkzeug für Künstler

- Künstler sollen sich auf eine Sache konzentrieren können
- „Strand geht zu Waldboden über“
  - Terrain mit mehreren Texturen/Materialien
  - Texturen Überlagerung
- „Durch das Kirchenfenster scheint buntes Licht“
  - Besondere Behandlung von transparenten Objekten
  - Komplexe Lichtberechnung

Aufbau eines Spiels

```
while(true) {
  processInput();     // Eingaben verarbeiten
  simulateWorld();    // Physik simulieren, Events abarbeiten
  renderWorld();      // Welt mit OpenGL anzeigen
}
```

➔ Aufbau nach dem EVA Prinzip

Performance in Spielen

- Soll wie interaktiver Film wirken
  - Normaler Kinofilm mit ca. 25 FPS
  - Spiel braucht mindestens 30 FPS
- Hohe Anforderungen an das Spiel:
  - Wunderschöne Grafik, realistische Physik
  - Direktes Feedback auf Eingaben, flüssige Anzeige (> 30 FPS)

    ➔ Ein Schleifendurchlauf in weniger als 33 ms!
- Moderne Grafikkarten sind extrem leistungsstark

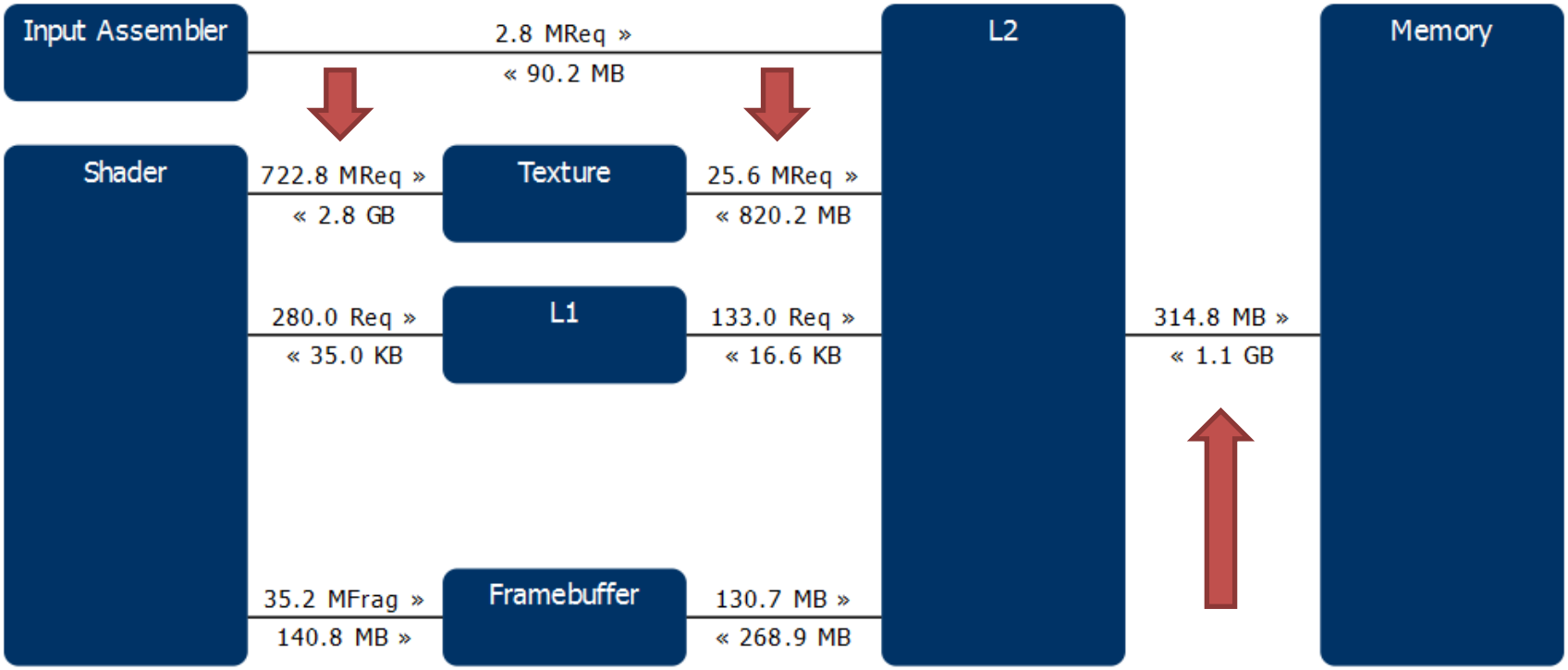
Was macht die Grafikkarte?

Test: „Crysis 2“, niedrige Grafik, 40 fps, ein Frame:

- Gezeichnete Primitives: `1.300.457`
- Bearbeitete Fragments: `36.916.705`
- Daten zwischen RAM und GPU: `1,4 GB`
- Daten innerhalb der GPU: `4 GB`

[Getestet auf einer GeForce GTX 460 (2010)]

Was macht die Grafikkarte?

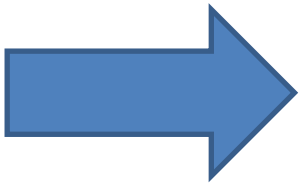


Screenshot: Nsight in Visual Studio

5,5 GB/Frame ⇔ 220 GB/s

Sonstiges Anforderungen

- Plattformunabhängig (PC, Konsolen, Mobil)
- Nutzbarkeit für alle Spiel-Genres



Spiel-Engines sind höchst komplex und schwer zu programmieren!

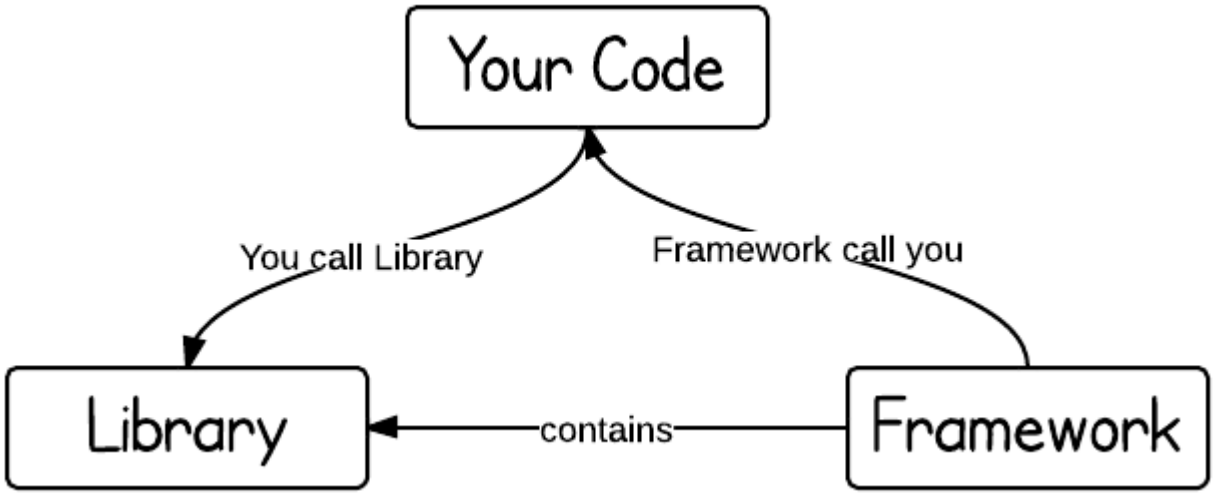
Struktur von Engines

Library

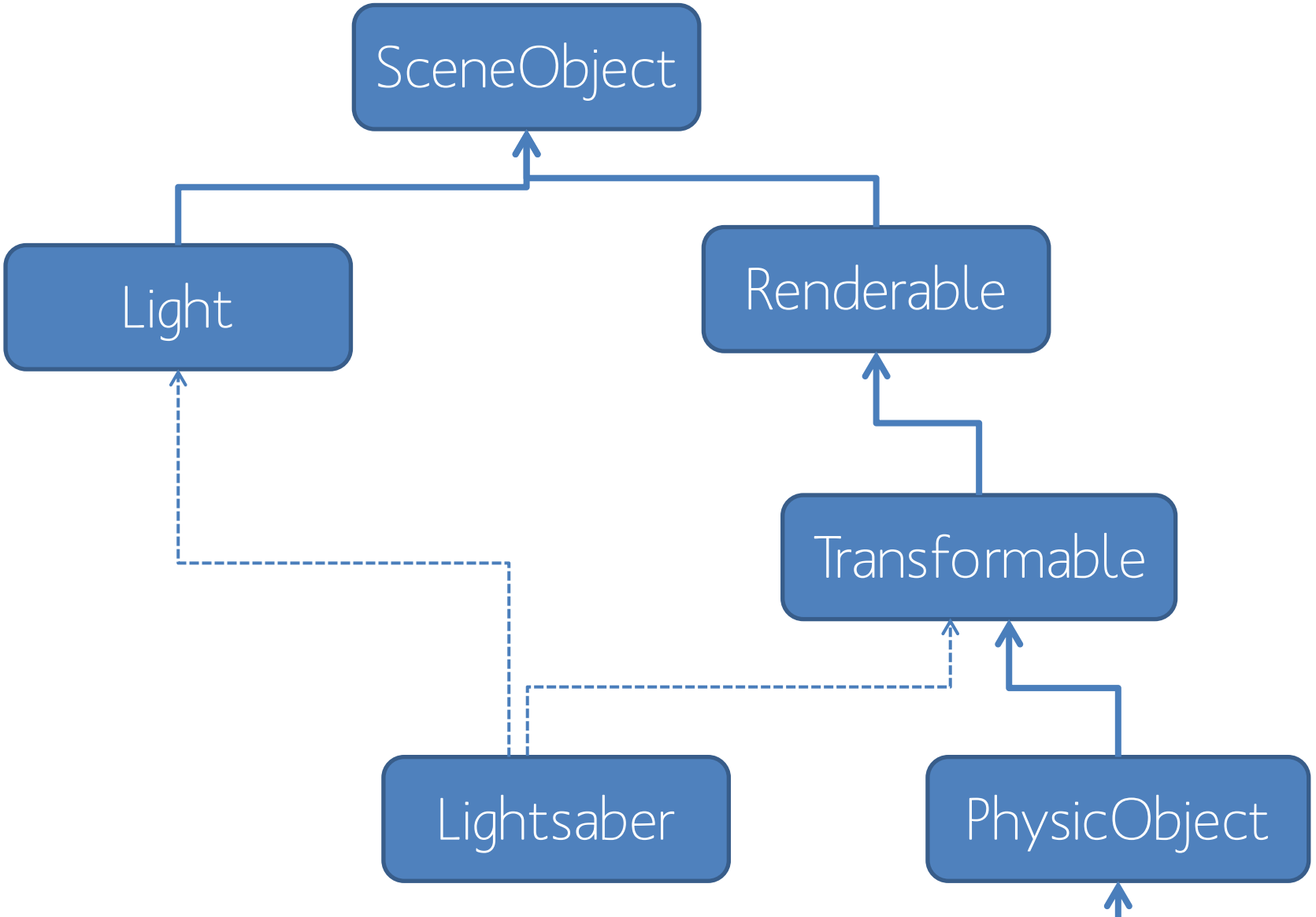
- Wird aus eigenem Code aufgerufen
- Eigener Code bildet Rahmen

Framework

- Ruft eigenen Code auf
- Bildet Rahmen der Applikation

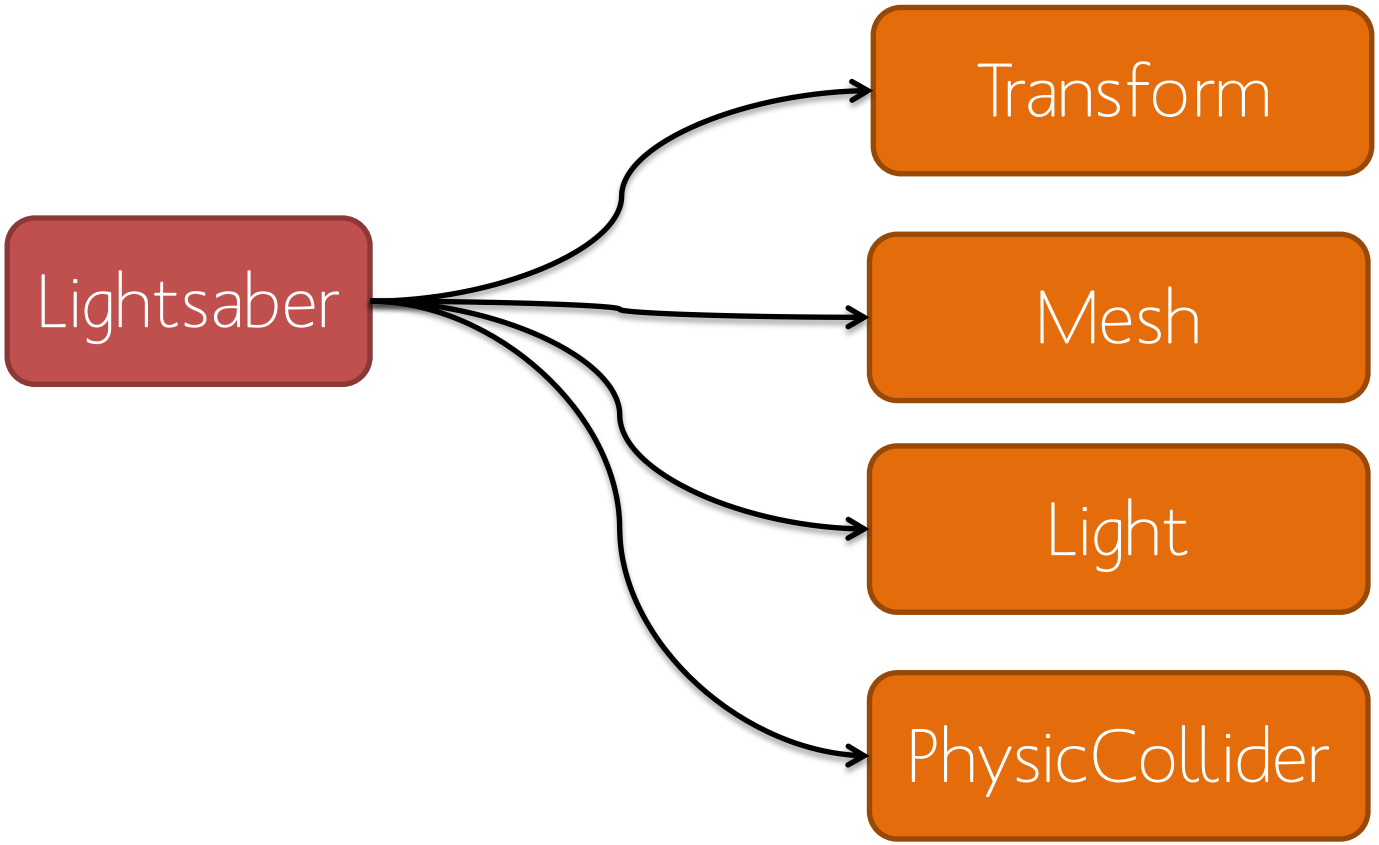


Programm-Design: Szene



Entity Component System

- Ersetzt „ist-ein“ durch „hat-ein“ (Komposition)

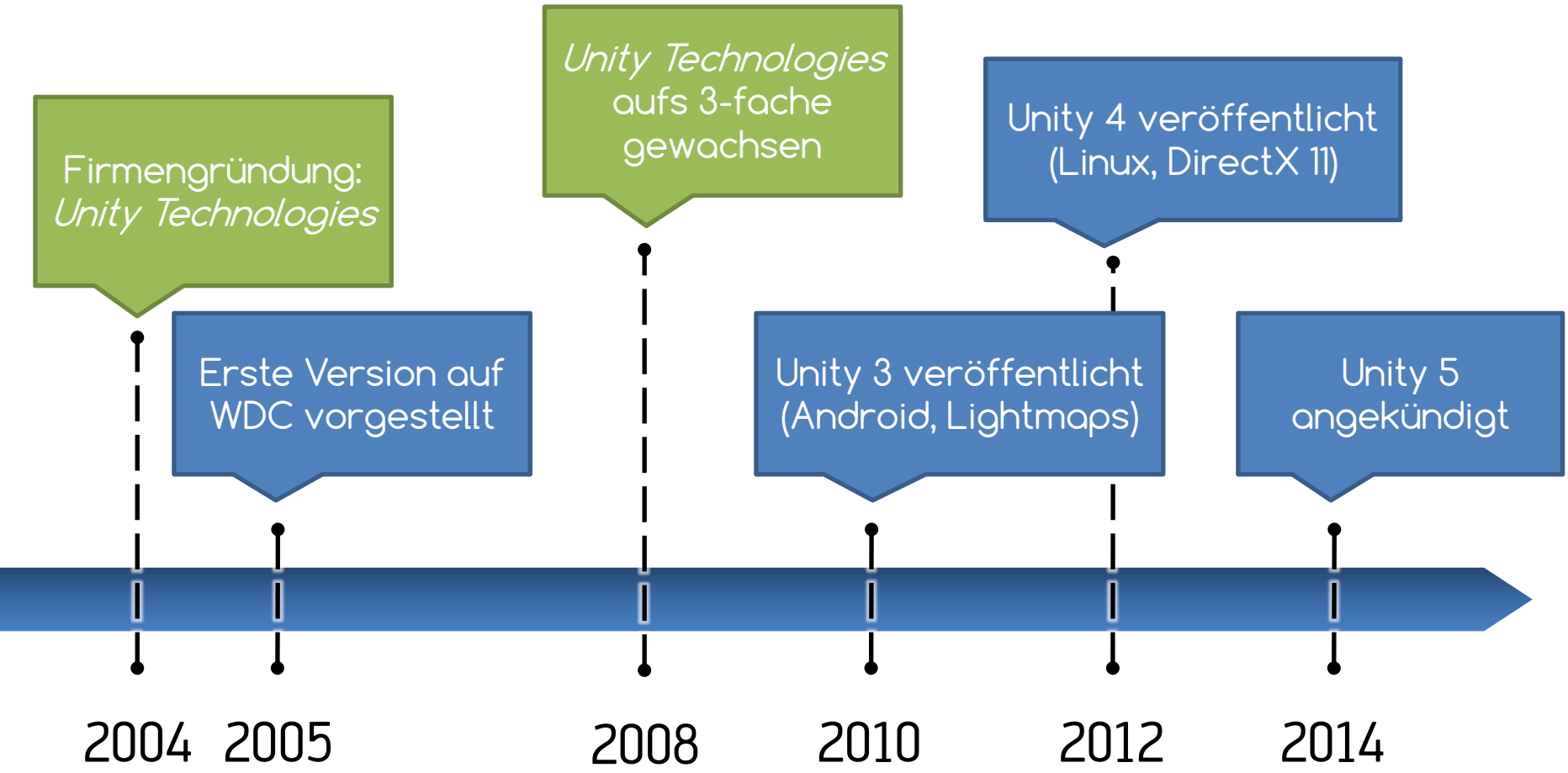


Was ist Unity?

Was ist Unity?

- In den letzten Jahren sehr populär geworden
  - 2,9M registrierte Entwickler (630 aktiv)
  - 10,7M Editor Sitzungen im Juni
- *Freemium* Preis Modell
  - Kostenlose Version für Hobbyentwickler
  - Pro-Version mit erweiterten Funktionen

Geschichte von Unity



➔ Wir nutzen Unity 4.5 im Praktikum

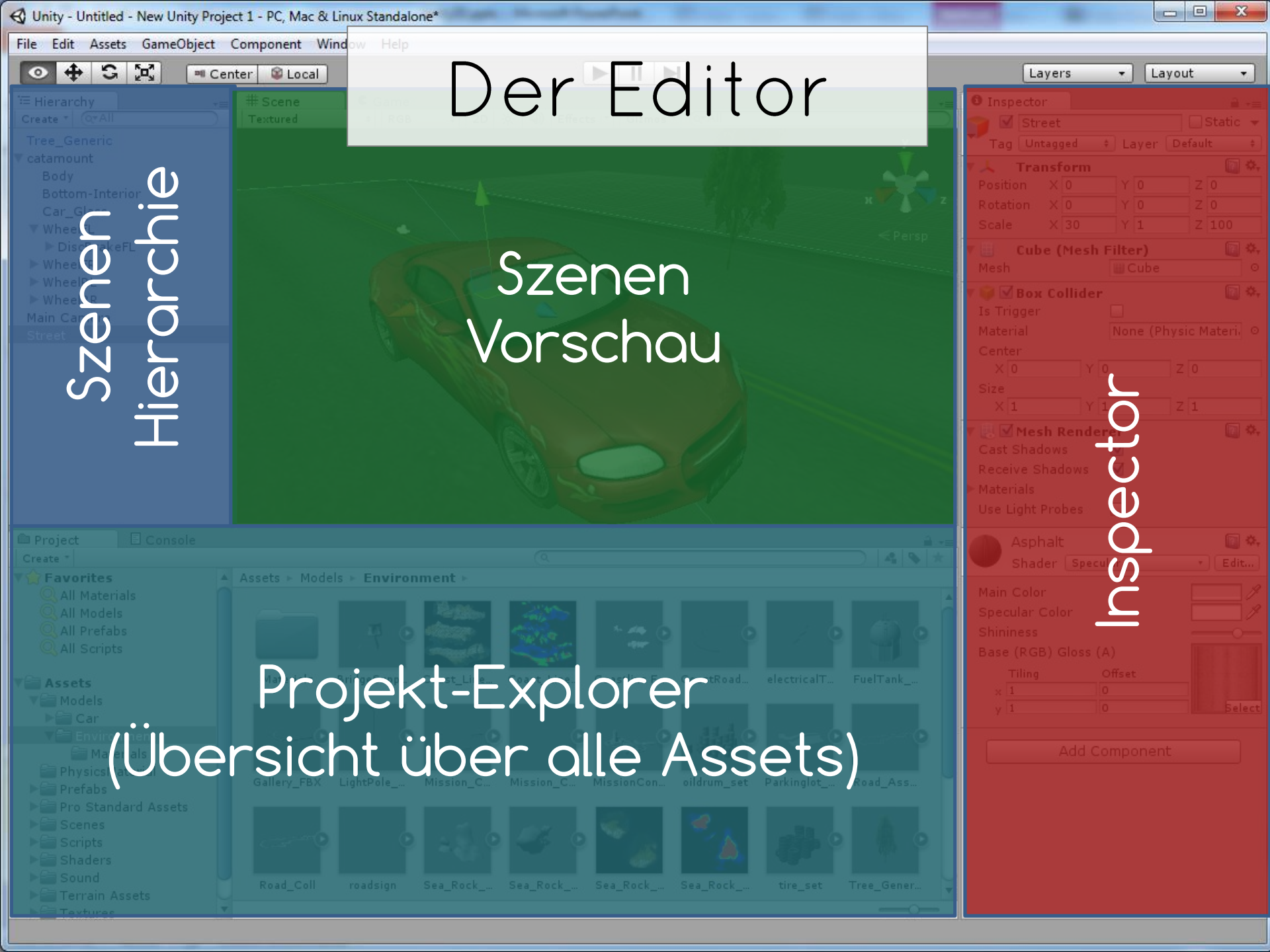
Unity

- Wichtigste Komponente: Der Editor
  - Ermöglicht interaktives Erstellen der Szene
  - Alle Einstellungen werden hier vorgenommen
  - Läuft nur unter Windows und OSX
- Unity komplett als Framework aufgebaut
  - Man schreibt nur sog. *Behavior-Scripts*
  - Kein eigener kompilierter Code o.ä.
  - Unity übernimmt Resourcen-Verwaltung und linkt alles zusammen

Assets und Stuktur der Szene

- Assets sind externe Daten
  - Texturen, Materialien, Models, Audio, Scripte
  - Oft in externen Programmen erzeugt
- Alles in der Szene ist ein GameObject
  - Kann verschiedene Komponenten besitzen
  - Aussehen durch Materialien und Models
  - Verhalten durch Scripts und Phyisk-Komponenten
- Flexibilität durch „Entity Component System"

Der Editor


Szenen Hierarchie
Szenen Vorschau
Inspector
Projekt-Explorer (Übersicht über alle Assets)

Scripts

- Programmiert wird nur in Behavior-Scripts

C#	• Objektorientiert • Von Microsoft entwickelt
JavaScript	• Sprache für Webseiten • Dynamisch typisiert
Boo	• Ähnlich wie Python • C# Runtime

Scripts

- Script ist ein *Asset*
- Als Komponente einem Objekt zugewiesen
- Wird bearbeitet in „MonoDevelop“