

Macoun



Grundlagen der Spieleentwicklung

Andy Abgottspon

@aaweb – www.aaweb.ch

Hi, I'm Andy!

Bei Tag: **Software Engineer**, The Foundry

Bei Nacht: **iOS developer**



FPS: 10.0137629

Save Settings

Auto Save

1: SETTINGS

Release



Ablauf

- Einführung
- Die Spiel-Engine
- Grafik und Animation
- Weitere Komponenten
- Skripting

Einleitung

You have to learn the rules of the game. And then you have to play better than anyone else.

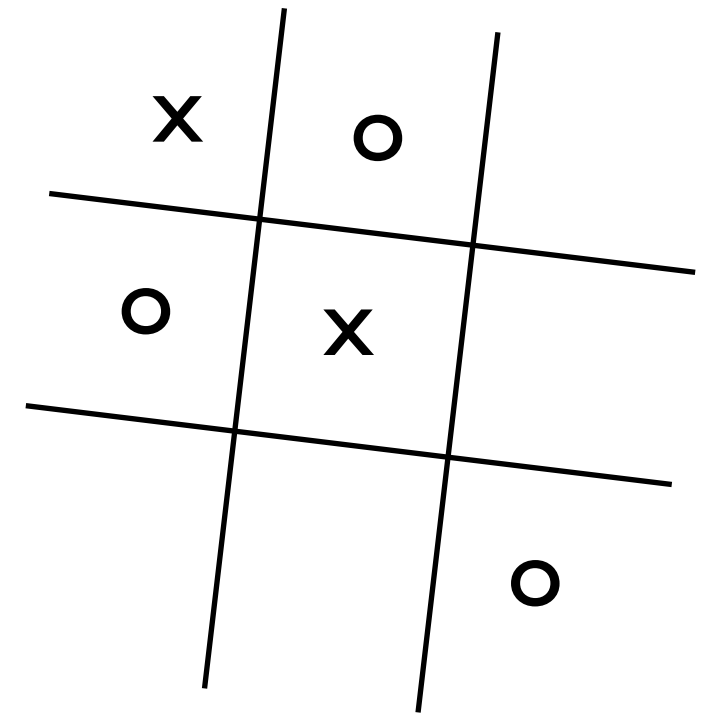
Albert Einstein

Was ist ein Spiel?

“Tätigkeit ohne Zweck und aus Freude, Spaß an ihr selbst, unterhaltende Beschäftigung, Kurzweil, im Unterschied zur Arbeit, Freizeitbeschäftigung” *Wiktionary 2012*

“A Game is a series of interesting choices”
Sid Meier

(Bei)spiele: Tic Tac Toe, Schiffeversenken



Wie designt man ein Spiel?

Geschichte

Physik

Musik

Software-Entwicklung

Psychologie

Marketing

Soziologie

Kunst

Geografie

Business

Bildung

...

Was erwarten die Spieler?

Schlaue KI

Schnell (min. 30 fps)

Komplexe Grafik

Alles über's Netzwerk

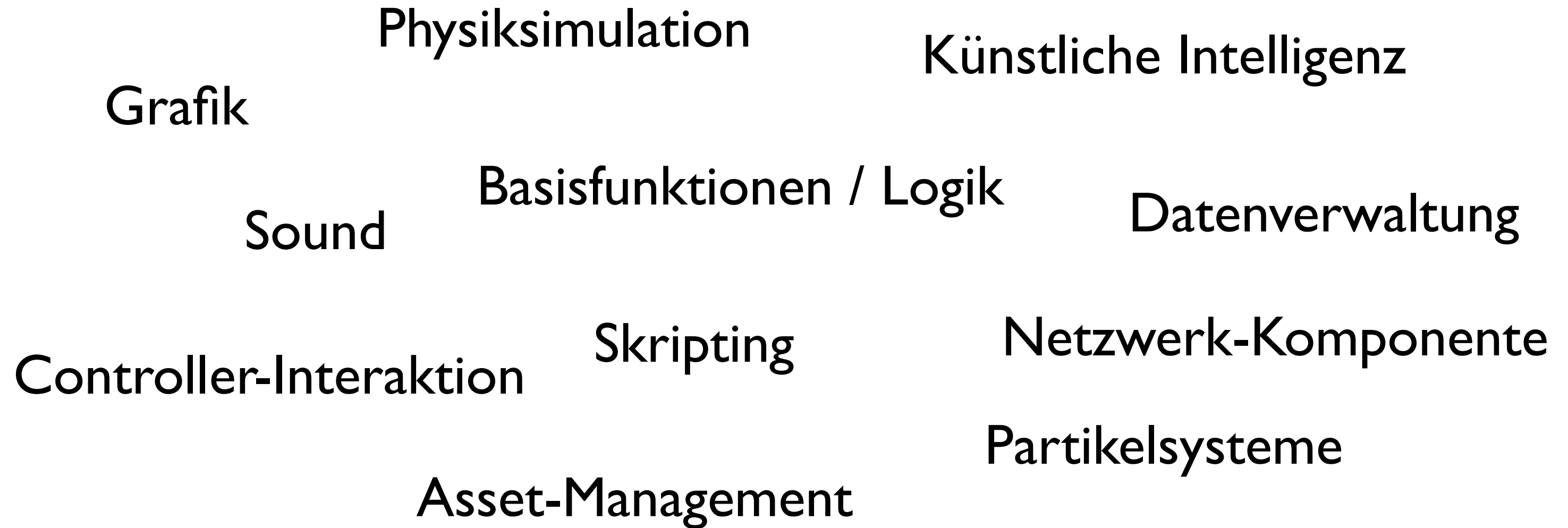
Realistische Physik



It's hard!

Die Spiel-Engine

Komponenten



Spiel-Engines für iOS

- **3D-Engines:**
Unreal Dev Kit For iOS, Unity 3D, Marmalade, Stonetrip S3D, GLBasic, Bork 3D, SIO2 (*), Irrlicht*, ...
- **2D-Engines:**
Game Salad, Corona SDK, Torque 2D for iPhone: iTorque, Game Editor, Sparrow*, Cocos2D*, ...

* Free / Open Source



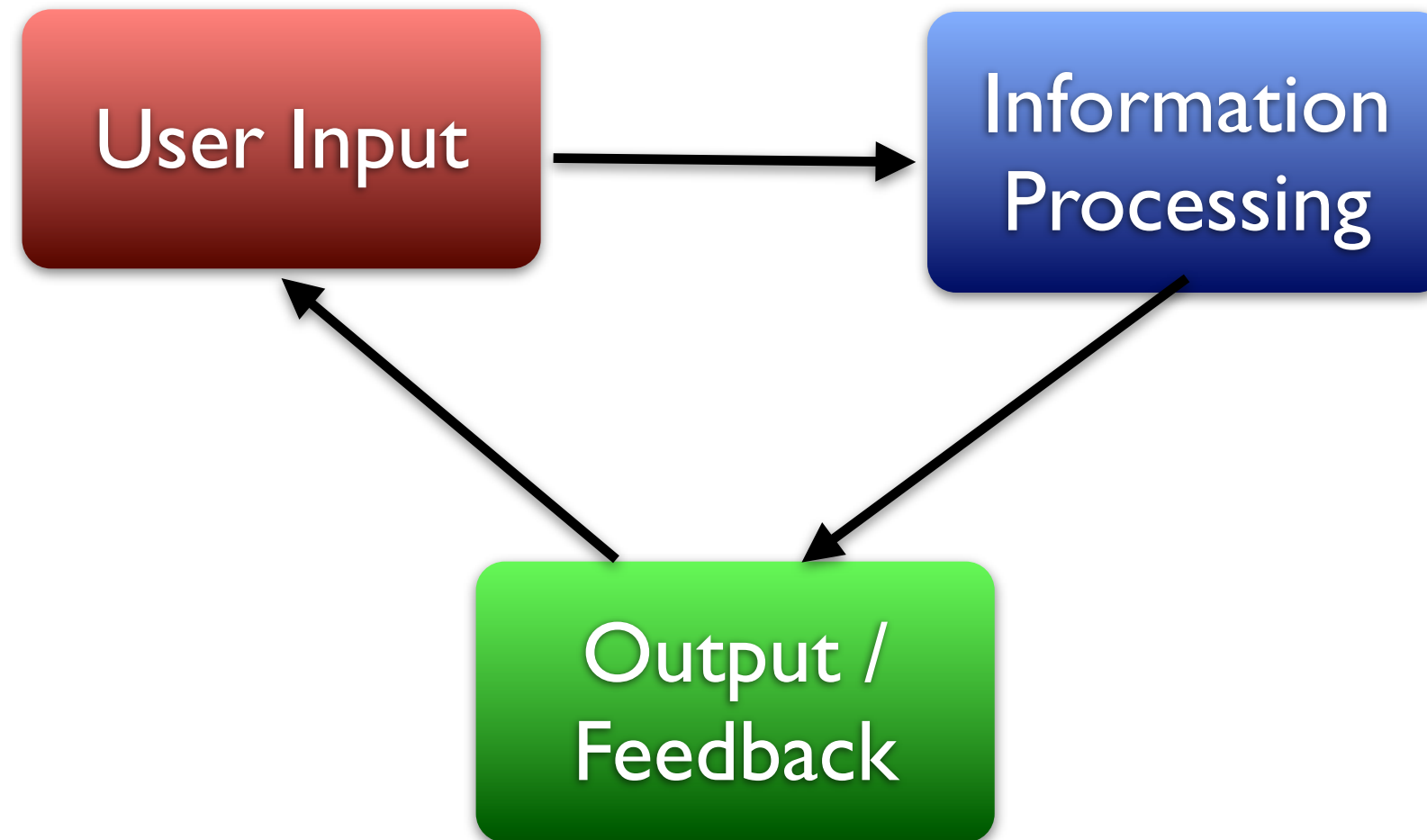
Open Source 3D Engine: **www.gameplay3d.org**

Das Gameplay-Framework

- Multi-Platform, C++: iOS, Android, Blackberry OS, Mac, Windows
- Projekte für Visual Studio, Xcode und Momentics IDE's
- Code Open Source auf github, contributed by RIM
- Was man halt so braucht: Math library, abstraction layer für mehrere Betriebssysteme, file format support, ...

Los geht's!

Die magische Formel



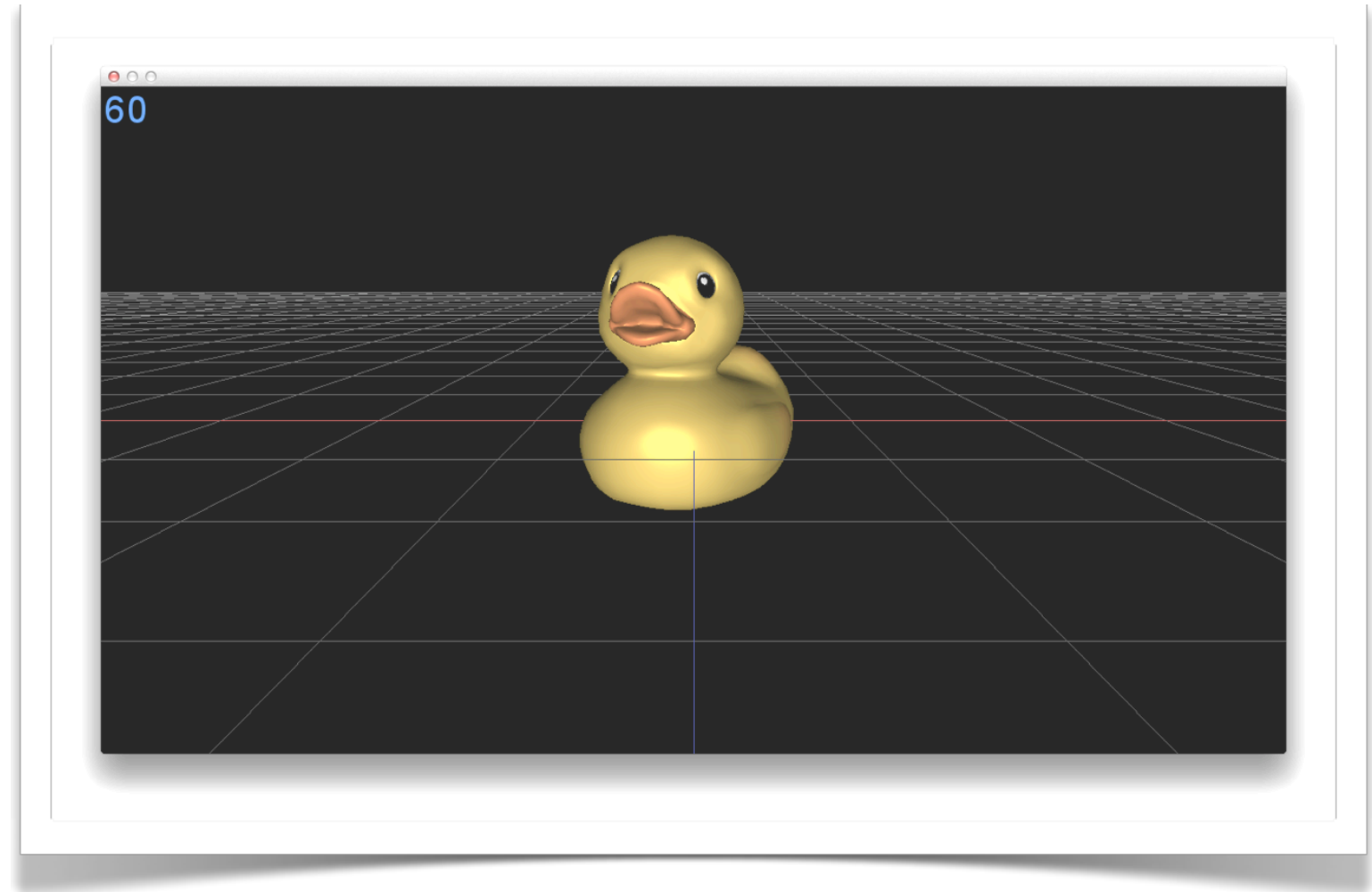
Demo

sample00-mesh

Es quietscht ...

I. Dieses Beispiel enthält:

- 3D-Model laden
- Touch- und Key-Events verarbeiten



Grafik und Animation

Grafik

3D-Modelle zu 2D-Bilder (Rendering-Pipeline)

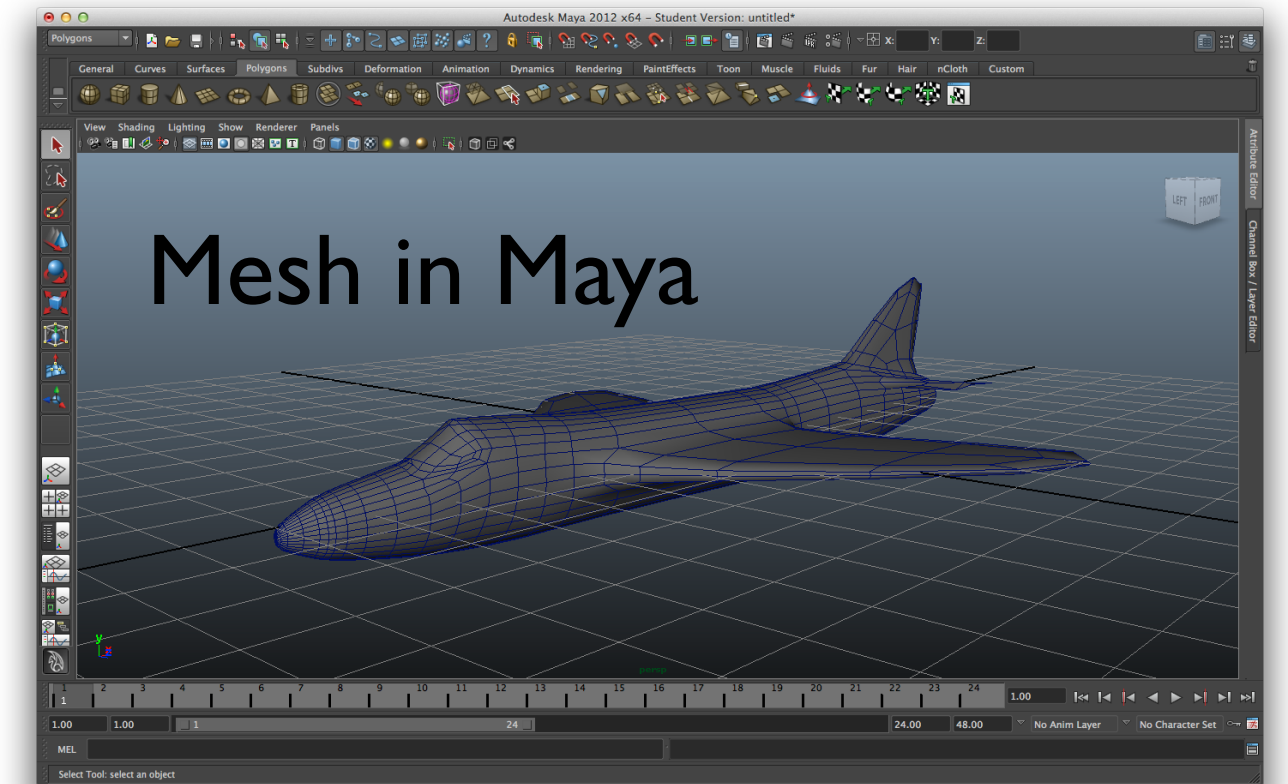
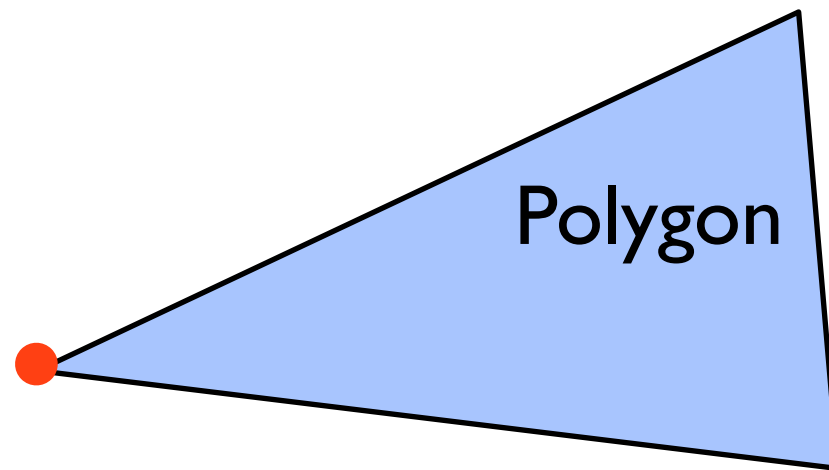
OpenGL (ES): Library für Transformationen, Projektionen und Zeichnen

Geometrie besteht aus Polygonen

Point ●

Vertex ●

Polygon



Was ist ein Shader?

Berechnen bestimmte Informationen bezogen auf ein Element.

Ein Element ist z.B. ein Vertex, Pixel oder Polygon.

Wir brauchen viel Power für diese Berechnungen, da wir ganz viele dieser Elemente haben.

Real-Time Graphics Pipeline



Vertex-Shader =
Mathematische
Funktion per Vertex

Generiert Fragmente
aus den bisher
gesammelten
geometrischen Daten

Berechnet Pixel-
RGB aus den vom
Vertex-Shader
übergebenen Daten

Physik

Arten der Simulation

Physik starrer Körper (rigid body mechanics)

nicht elastische Physik mit Deformationen (soft body dynamics)

Masse-Feder-Modelle (spring dynamics) u. a. als Grundlage der Simulation von Seilen und Stoff (rope bzw. cloth)

Partikelsysteme u. a. als Grundlage der Simulation von Flüssigkeiten (fluid dynamics) und Feuer

Physik-Engines: Bullet, Havok, PhysX, Open Dynamics Engine, Tokamak, Newton Game Dynamics etc.

Demo

sample03-character

Ein Beispiel mit Charakter

1. sample03-character ausprobieren
2. Dieses Beispiel enthält:
 - 3D-Model laden
 - Touch- und Key-Events verarbeiten
 - **Shader**
 - **Animationen**
 - **Physik**
 - **GamePad**



Partikelsysteme

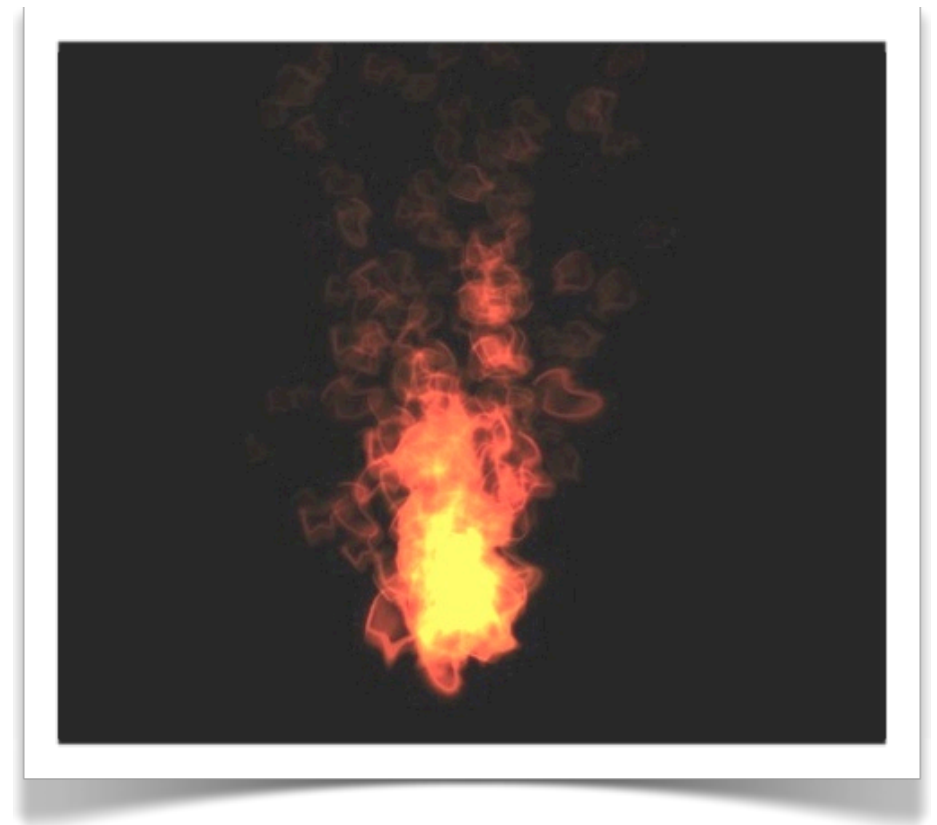
Partikelsystem = Ansammlung einzelner Partikel (Punkte im 2D-/3D-Raum) mit Regeln für deren Verhalten

Partikel haben Eigenschaften:

Position, Farbe, Transparenz, Geschwindigkeit, 3D-Texture, Alter/Lifetime, Zufälliges Verhalten, Krafteinflüsse, etc.

Lebenszyklus:

Generation > *Dynamics* > Death

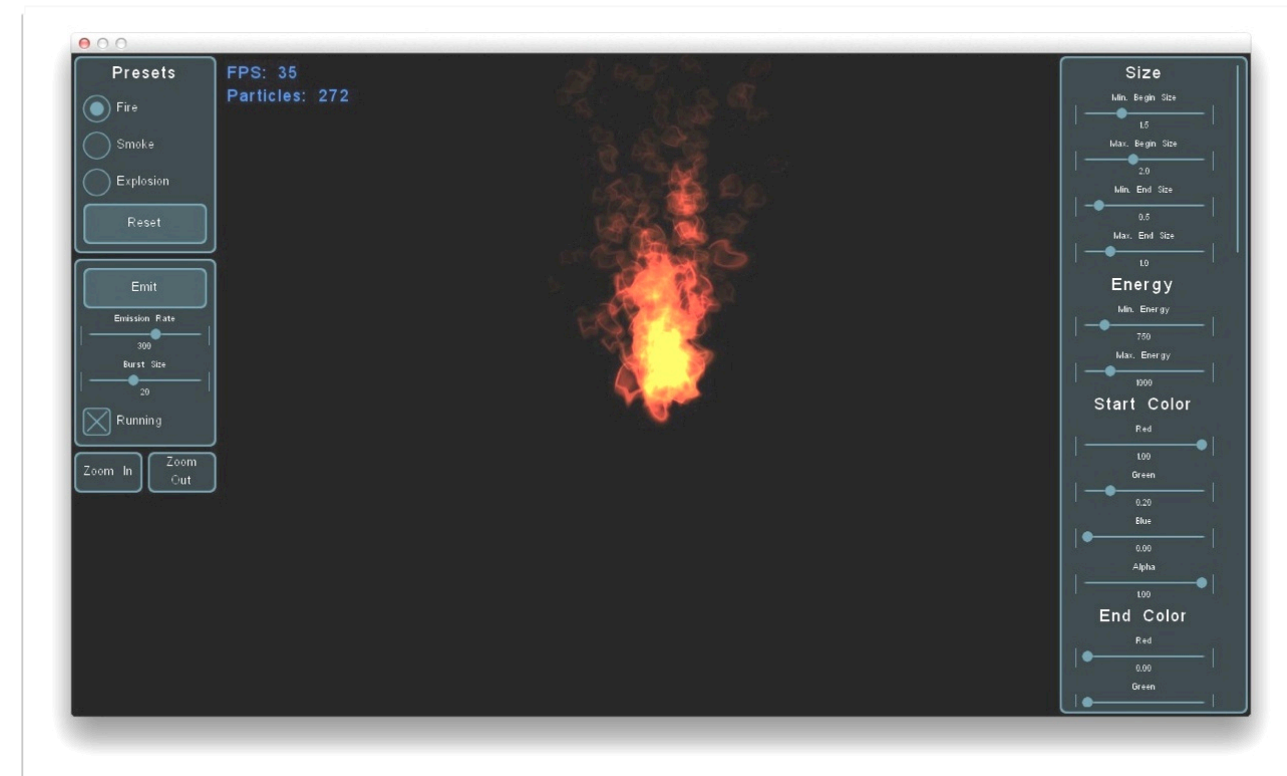


Demo

sample04-particles

Es brennt ...

1. sample04-particles ausprobieren
2. Dieses Beispiel enthält:
 - **Partikeleffekte**
 - **Grafische
Benutzeroberfläche**



Skripting

Was ist eine Skriptsprache?

- Dynamisch und in der Regel simpler
- Zur Laufzeit kompiliert
- Rapid development vs. meistens langsamere Ausführung (Overhead)
- “Paralleles Entwickeln”
- Komplexität und Logik einfacher verlagern

```

152 -- =====
153 -- GAME FUNCTIONS
154 -- =====
155
156 -----
157 -- @brief Initialise the game
158 -----
159 ▼ function initGame()
160     -- Airplane
161     airplane.id = createGameObj("HawkerHunter.obj");
162     setName(airplane.id, "airplane");
163     setTexture(airplane.id, "HawkerHunter.jpg");
164
165     -- Fire
166     addAnimObject("explosion", "");
167     animObjects["explosion"].duration = 2;
168     animObjects["explosion"].postEvent = "crashOverEvent";
169     setVisible("psFire", false);
170     setVisible("psSmoke", false);
171
172     -- Setup skybox, terrain ...
173     setSkyboxTexture("skydome", "jpg");
174     setFlagFog("skybox", true);
175     setFlagFog("terrain", true);
176
177     -- Background sound
178     loadBackgroundSound("background.m4a");
179     --loadBackgroundSound("Cabin.wav");
180     playBackgroundSound();
181
182     -- Animations
183     addAnimObject("respawnAnim", "");
184     animObjects["respawnAnim"].duration = 3;
185     animObjects["respawnAnim"].postEvent = "respawnOverEvent";
186
187     addAnimObject("flash", "");
188     animObjects["flash"].duration = 0.5;
189     animObjects["flash"].postEvent = "flashEndEvent";
190
191     -- Axis-aligned bounding boxes for tree
192 ▼ for i=1,3 do
193     treeAABBs[i] = createGameObj("treeAABB"..i..".obj");
194     setVisible(treeAABBs[i], false);
195 ▲ end
196
197     -- Go create trees (as many as needed at maximum)
198 ▼ for i=1,game.maxObstaclesInAnyLevel*2 do
199     tree = {};
200     tree.id = createGameObj("treetrunk.obj");
201     setTexture(tree.id, "treetrunk.jpg");
202     setFlagFog(tree.id, true);

```



Demo

sample05-lua

Weitere Beispiele

Demo

sample06-racer



gameplay Survival Shooter Demo



gameplay Dragon Demo

Fazit

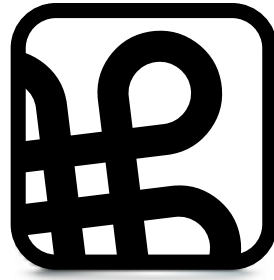
- Spieleentwicklung ist schwierig.
- Mit den richtigen Tools ist aller Anfang ~~leicht~~ leichter.
- Wissen universell anwendbar.
- Viel Spass!

"Because of the nature of Moore's law, anything that an extremely clever graphics programmer can do at one point can be replicated by a merely competent programmer some number of years later."

- John Carmack

Fragen?

Vielen Dank



Macoun