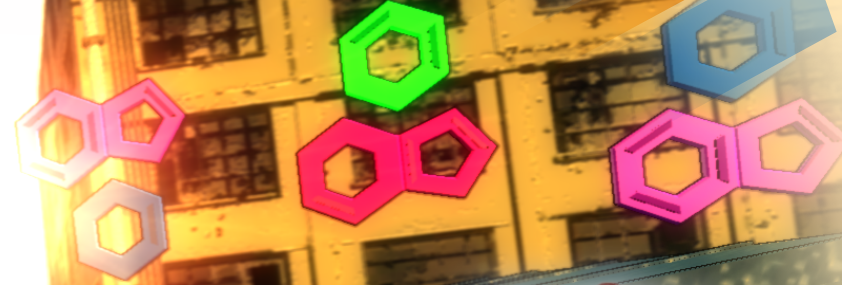


CRAZY DOGS EVOLUTION RUN



Vorweg

- Zeichnen?
- Modellieren/Animieren?
- Level bauen?
- Programmieren?
- Zocken?

Überblick

- Ideenfindung
- Projektstruktur
- Game Design
- Concept Art
- Modellierung
- Level Design
- Gameplay Creation / Programming
- UI Design & Programming

Ideenfindung

- Telefonkonferenzen
- GDrive Dokument + Concepts
- Game Design (Document) komplett überarbeitet
- Verbesserung, Erweiterung, Ausarbeitung der Spielidee im Team

Projektstruktur

- Dokumente: GDrive
- Tasks: Redmine
- VCS: Git (653 Commits/Änderungen an 5809 Dateien)
- Termine: GCalendar
- Kommunikation: Skype, Teamspeak

Game Design - Allgemein

- Kompetitiver 2.5-D Side-Scroller
- Lokaler Multiplayer
- Geschicklichkeit und Herausforderung
 - Wettrennen und Wettkampf
 - Highscore & Statistik

Game Design - Level

- Variierende Level
- Kämpfe, Powerups & Fallen

Level mit Hindernissen + Kämpfe

Level bestehen aus

mehreren Sektoren

bestehen aus mehreren Modulen

zwischen 2 Sektoren: Charakterwechsel

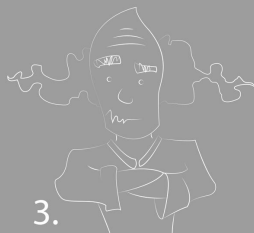
Game Design - Charaktere

- Variierende Steuerung
- Geplant: Angriff, Spezialfähigkeiten
- 3 verschiedene Charakter-Klassen/Steuerungen
- Wechselnde Charaktere während des Levels

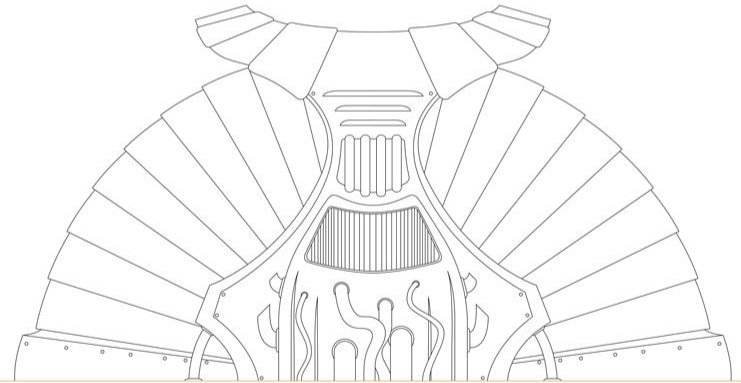
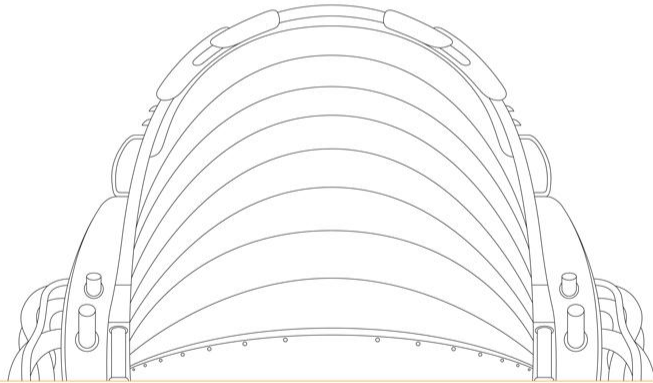
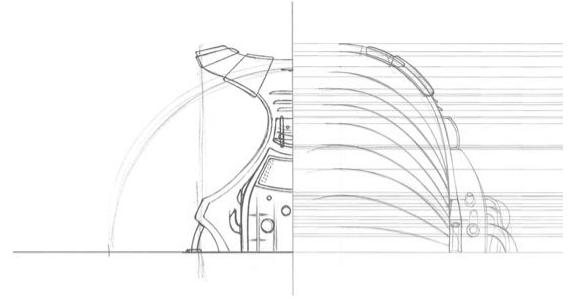
→ neue Steuerung

- Token
 - Gen-Punkte + Zeit
- Ressourcen
 - Leben + Boost

Hauptfigur (Der Verrückte Professor)



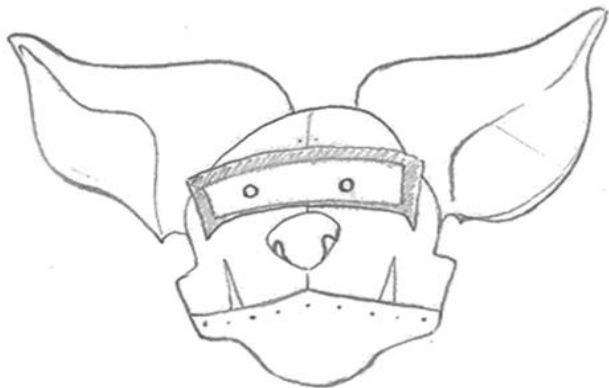
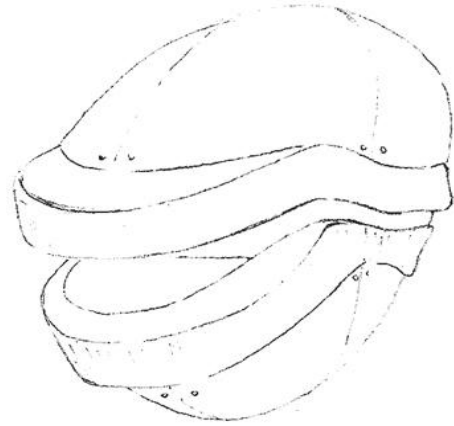
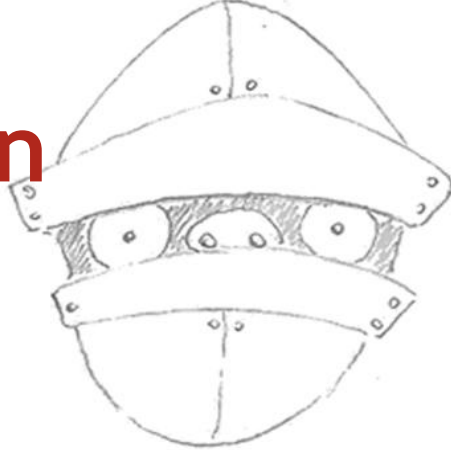
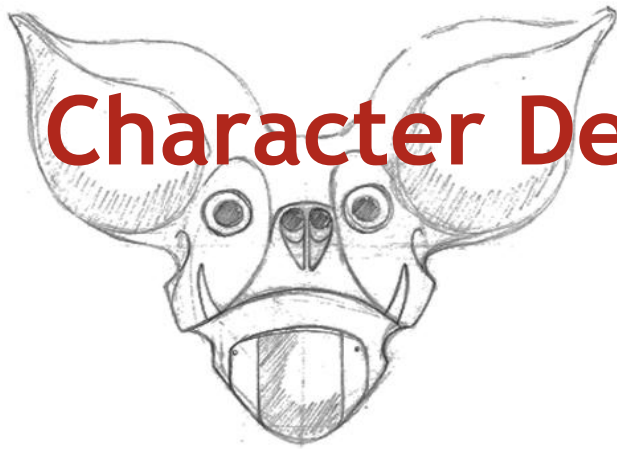
Environment Design

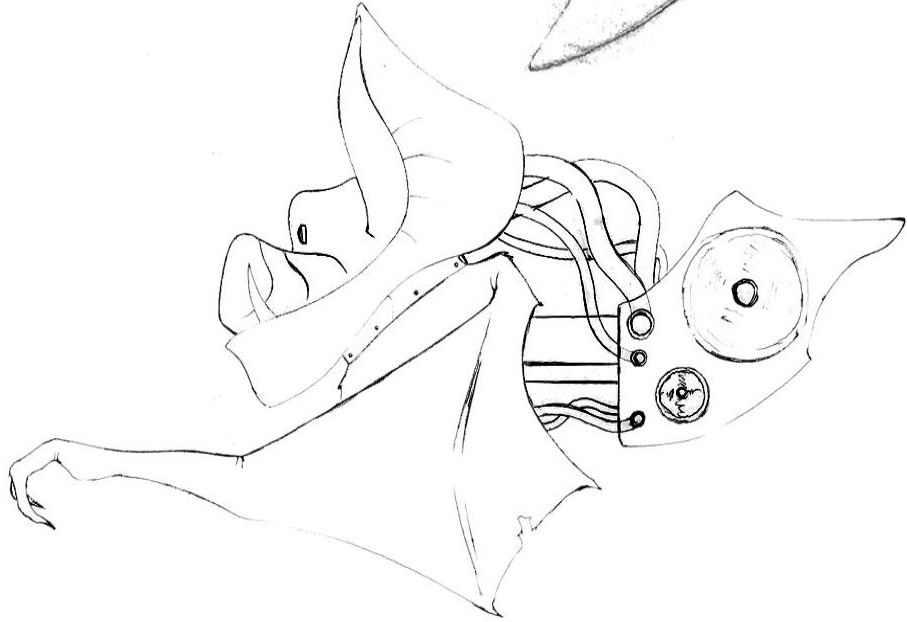
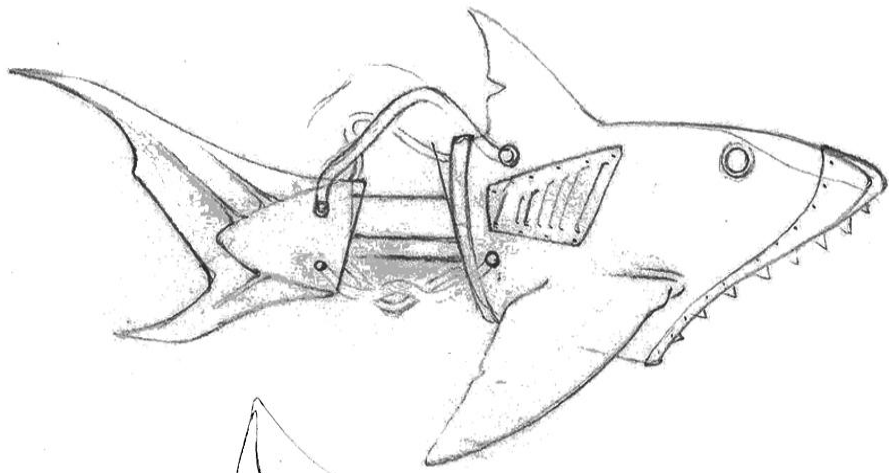


Concept Art -> Final Model

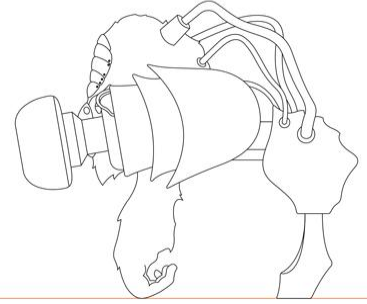
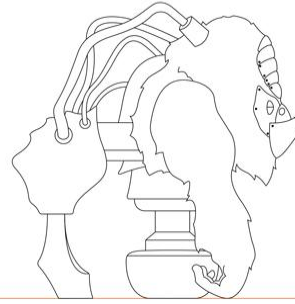
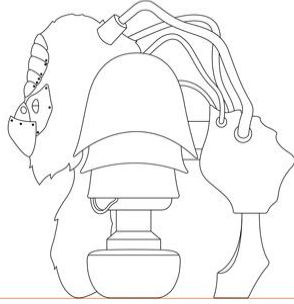
- Recherche und Ideenfindung
- Skribbles
- Ausarbeitung
- Modellieren
- Texturierung
- Rigging
- Weight Painting
- Animation
- Exportieren
- Importieren

Character Design



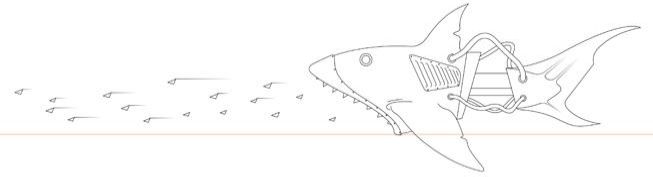
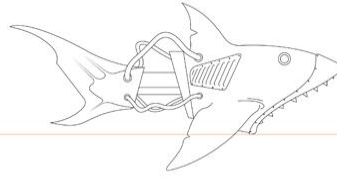
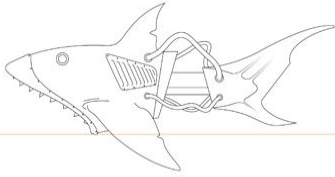
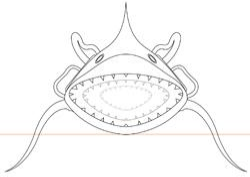


kampflattacke



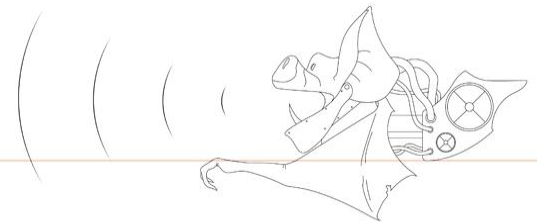
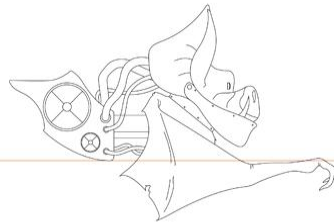
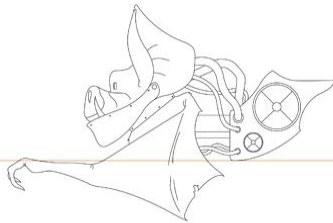
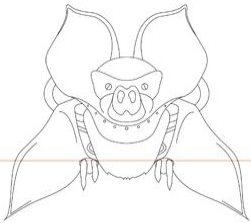
0

kampflattacke

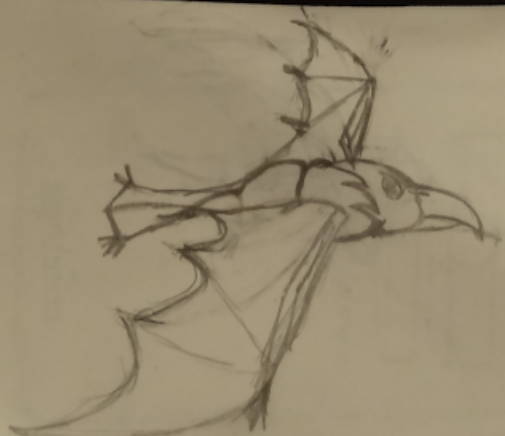


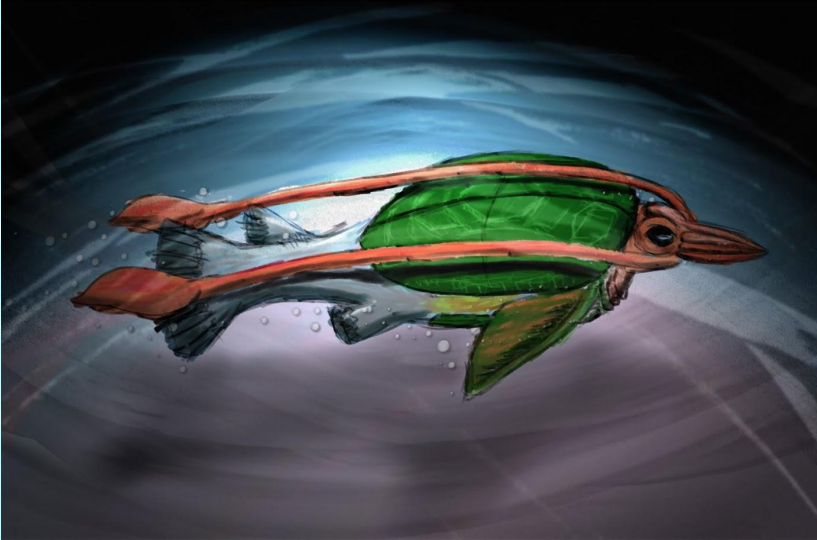
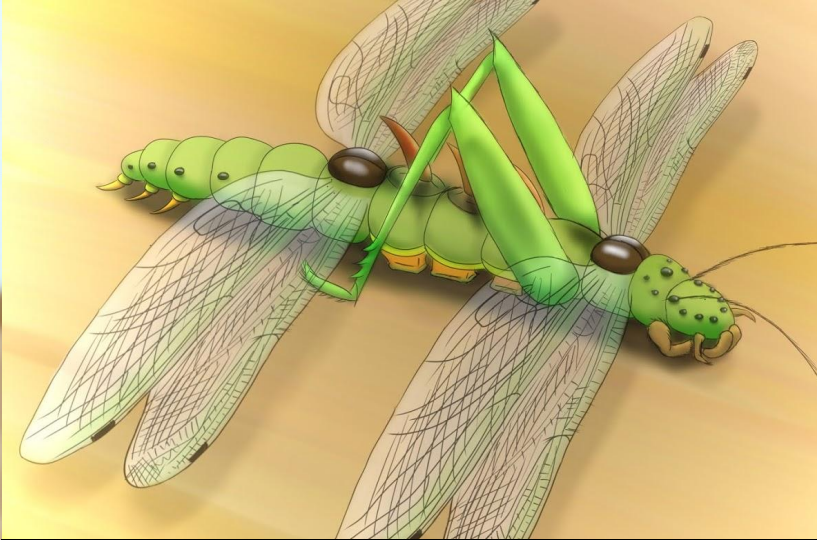
0

kampflattacke

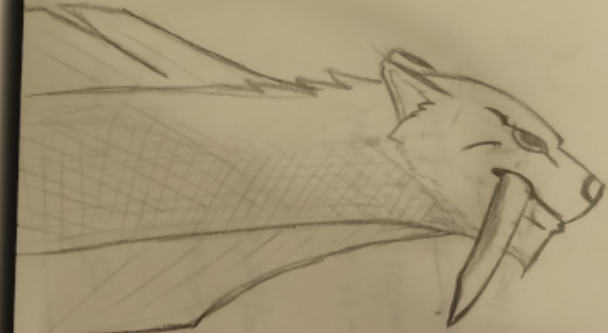


0

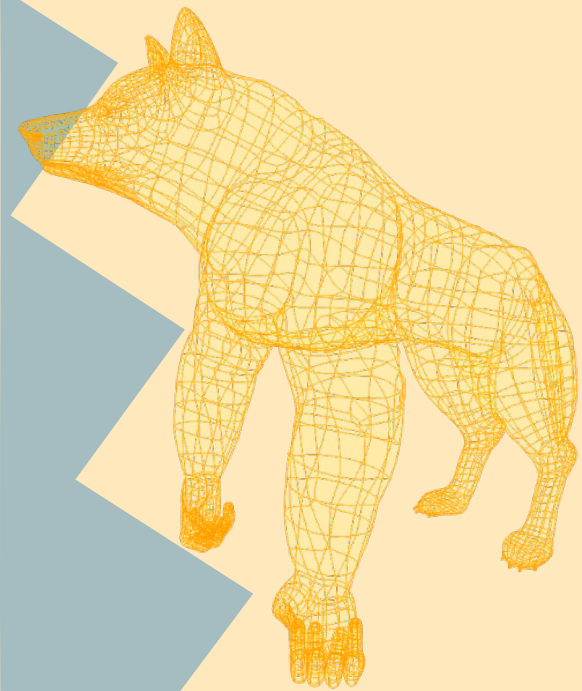




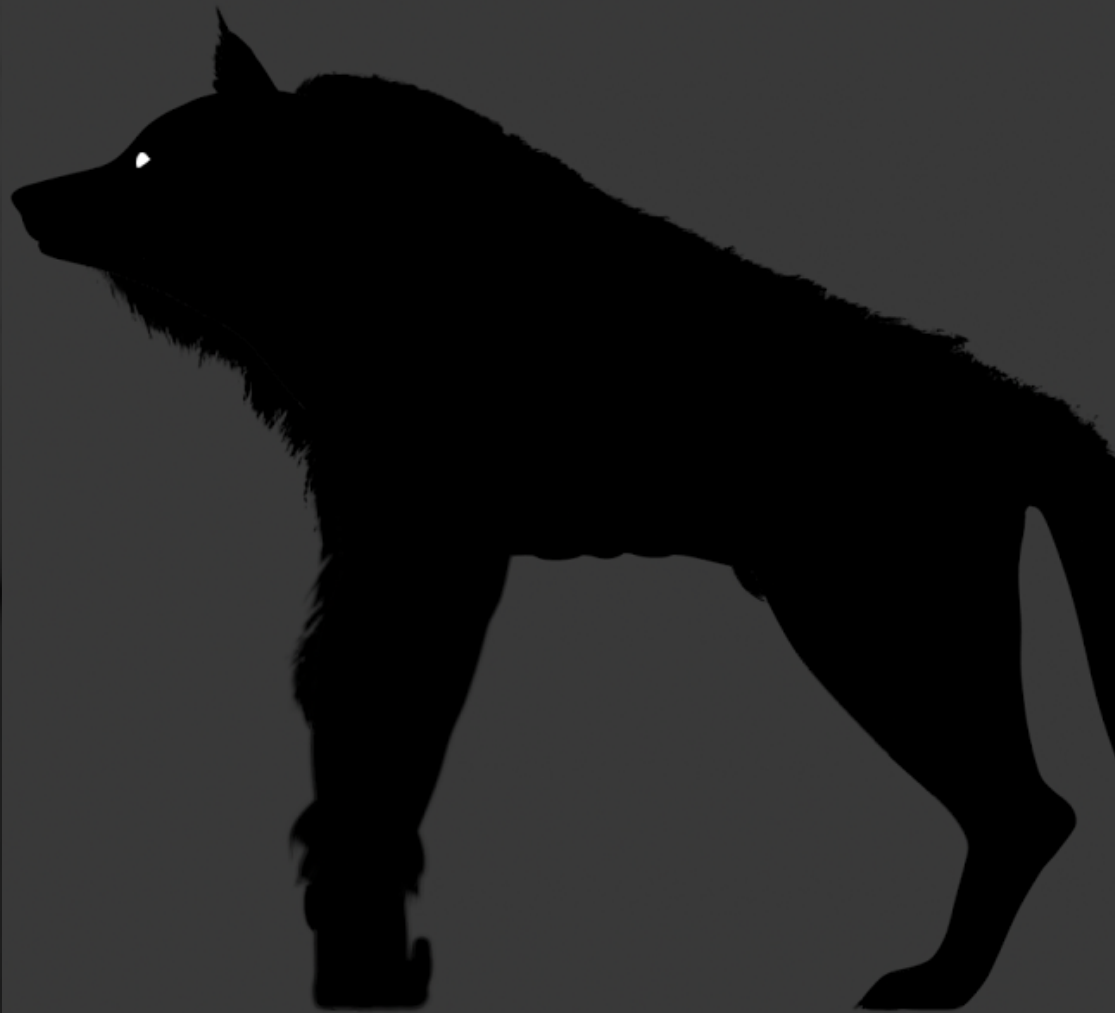
Studie: Wolf



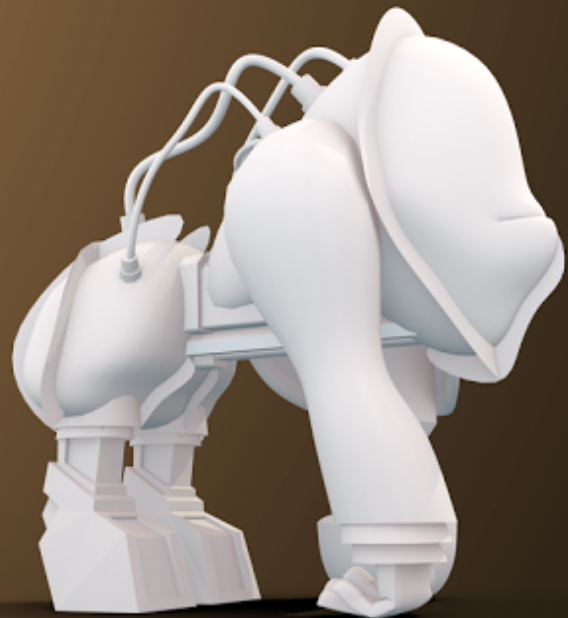
Typologie



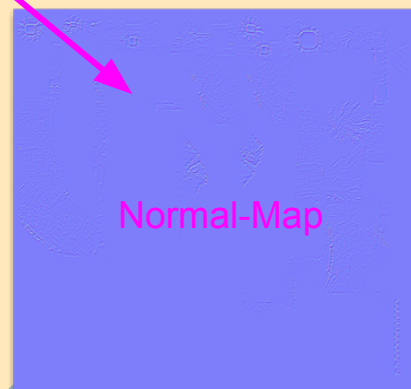
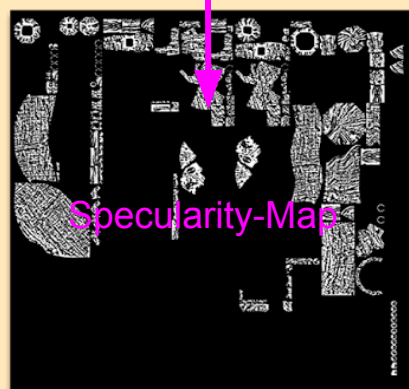
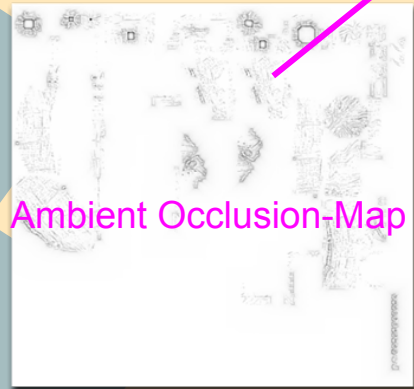
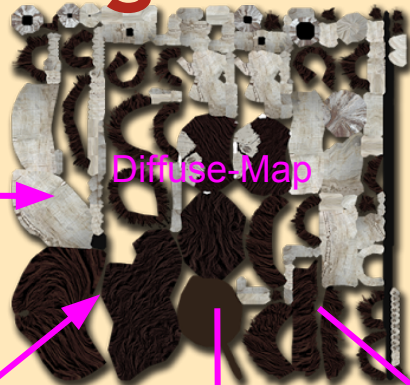
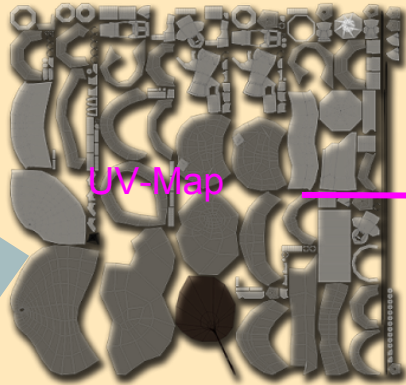
Rendering



Studie: Gorillabot

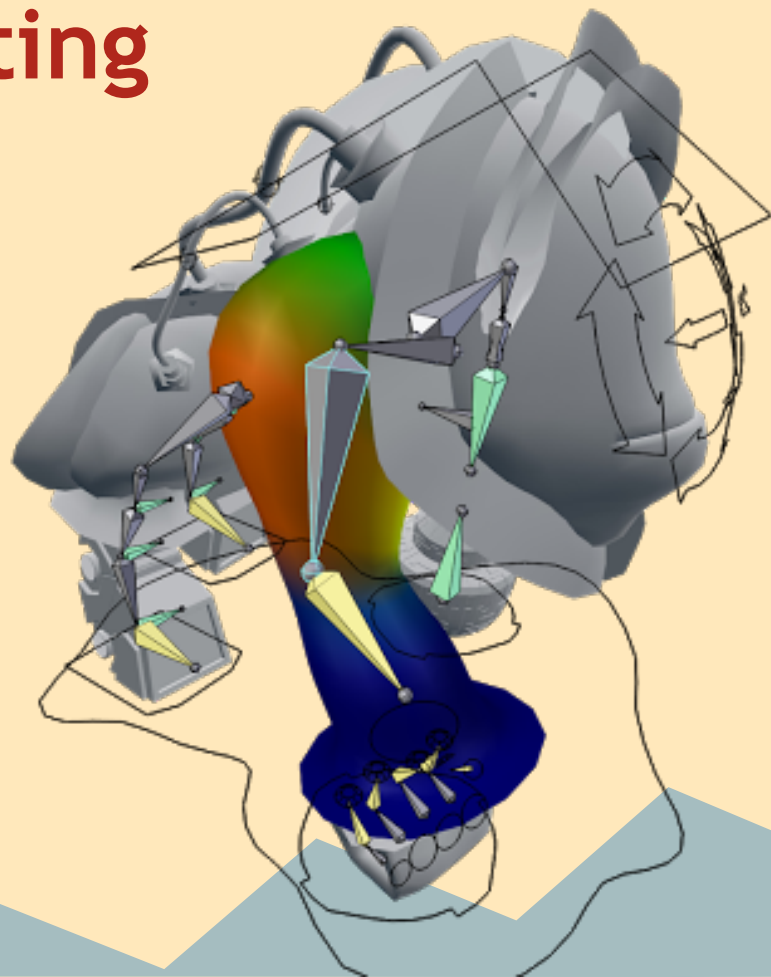
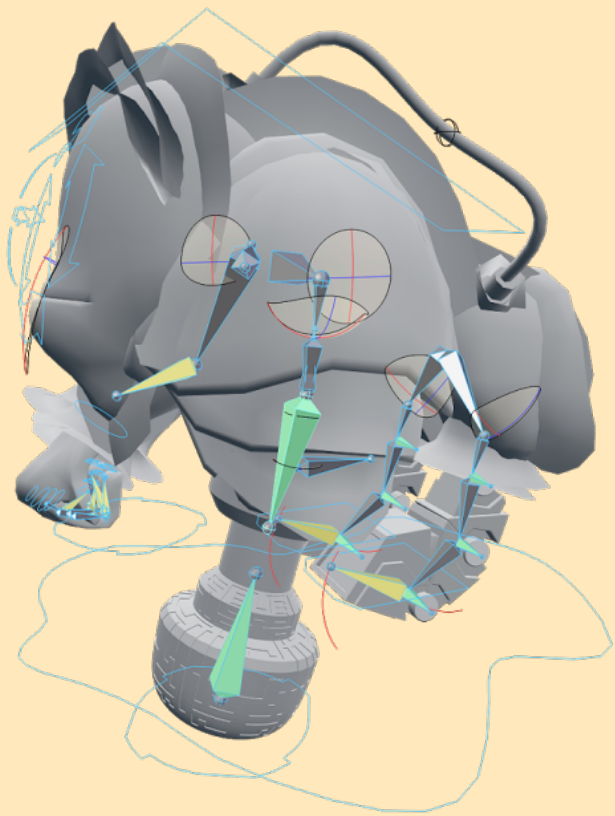


Texturierung





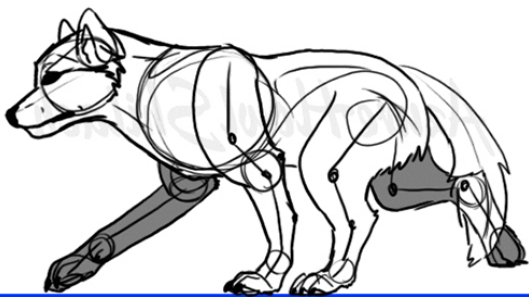
Rigging & Weight Painting



Animation



1



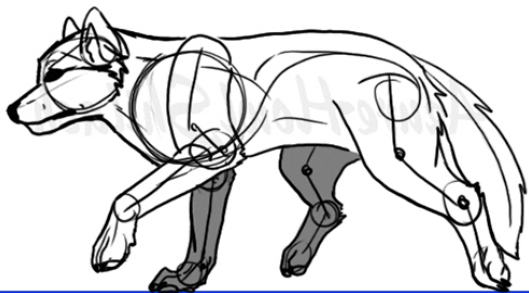
2



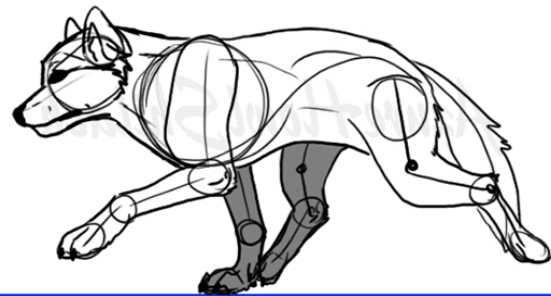
3



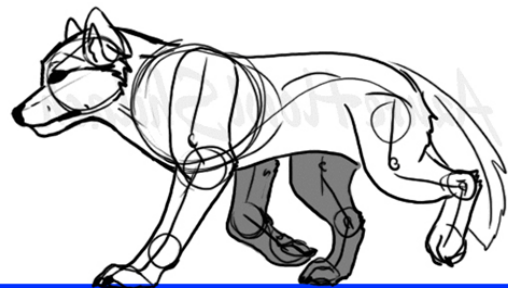
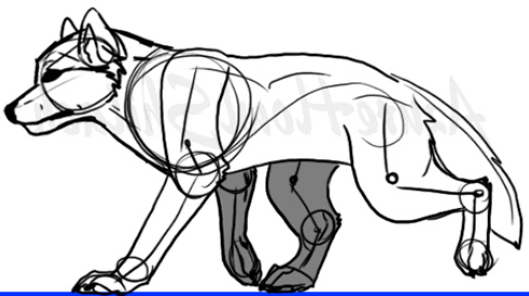
4

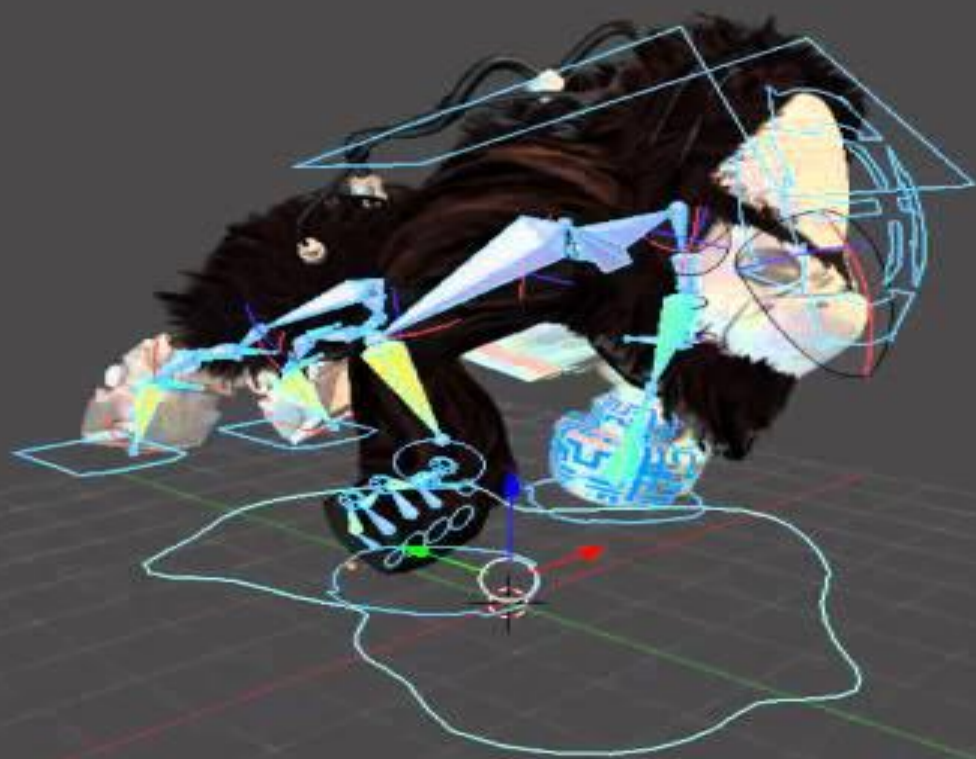


5



6

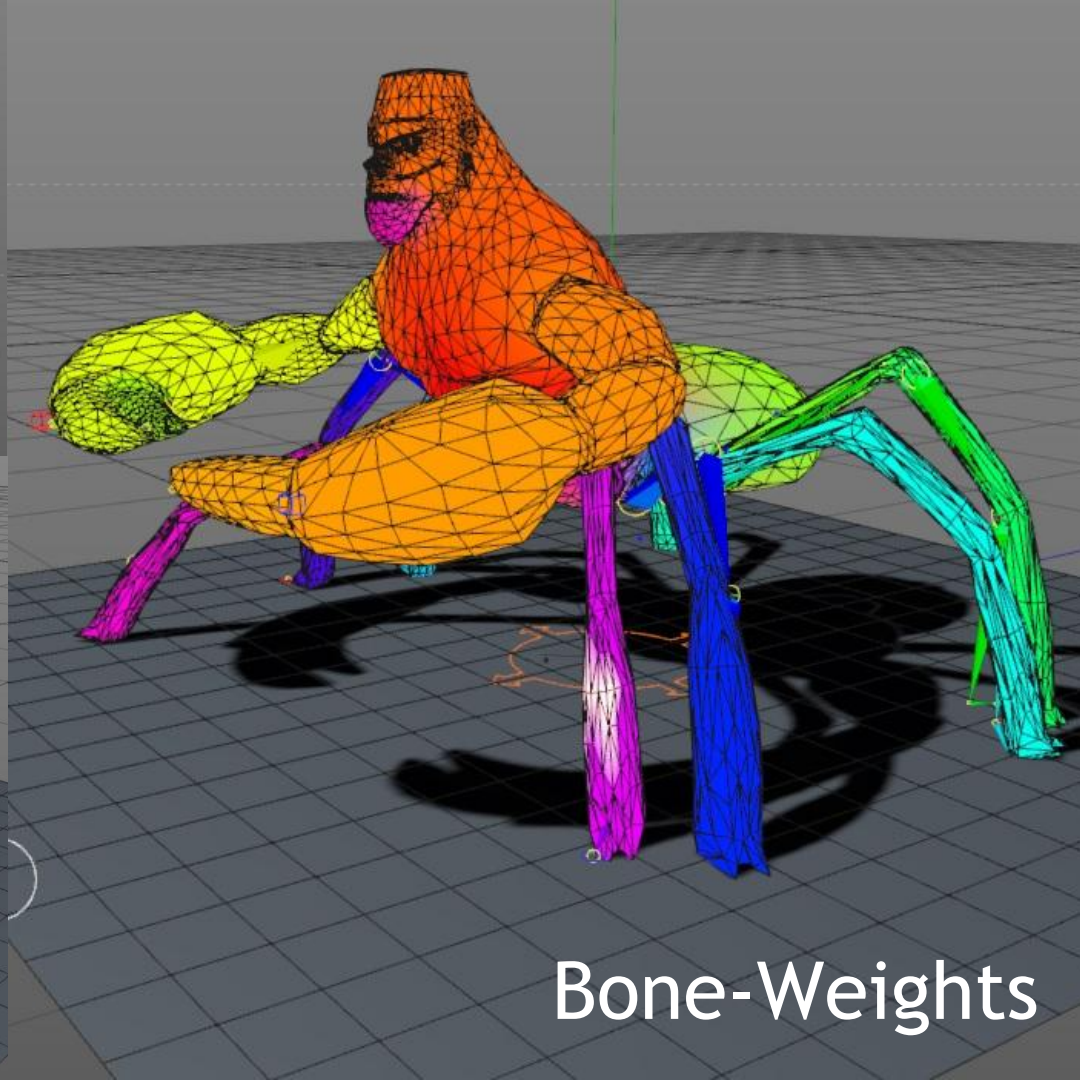
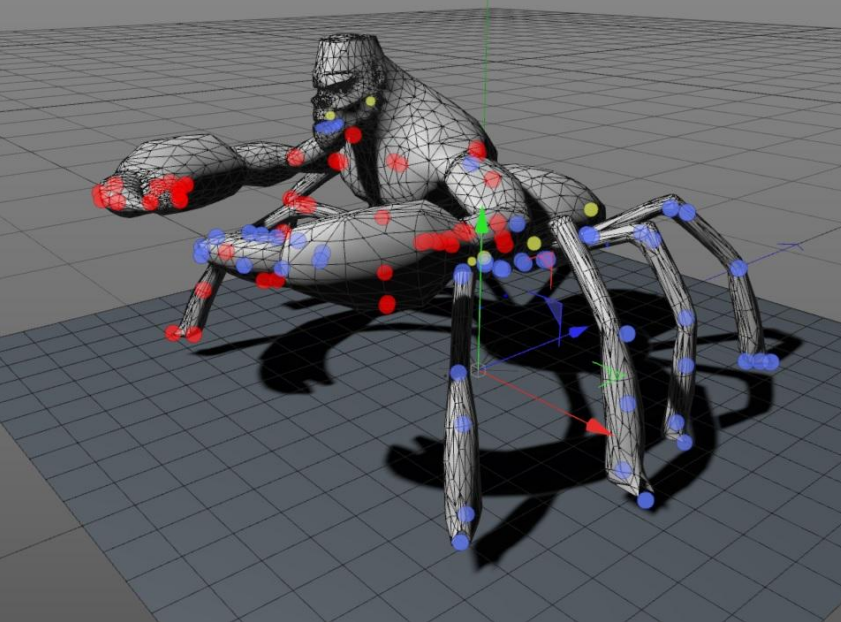
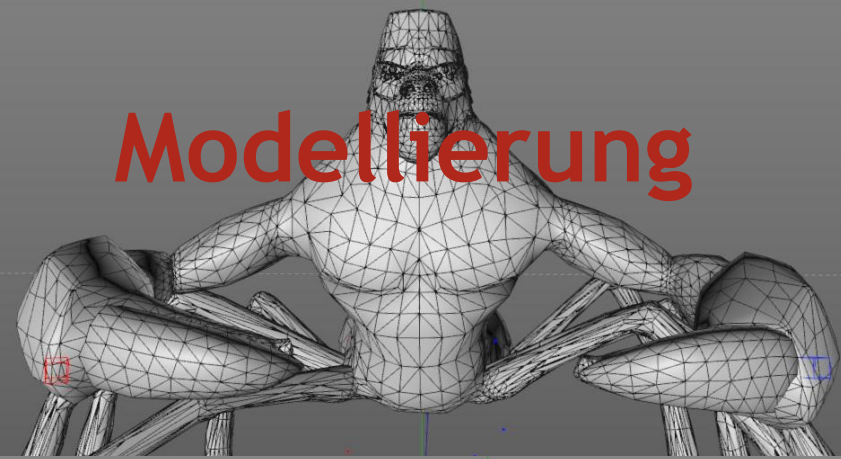




Studie - Gorcrasp



Modellierung

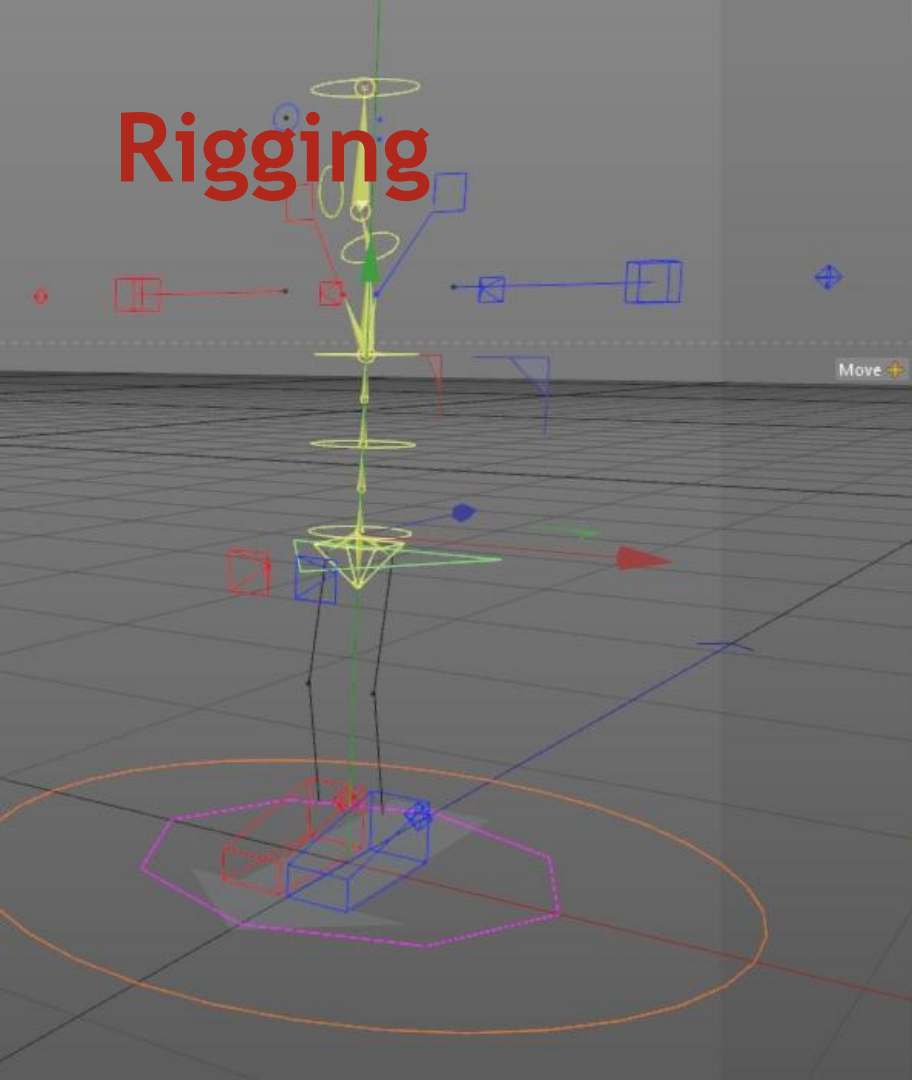


Bone-Weights

Texturierung



Rigging



Character [Character]

Basic Coord. **Object** Display

Object Properties

Build Adjust Binding Animate

Template **Advanced Biped**

Components

Arm (IK/FK Bendy) Leg (IK/FK Bendy)

Eyes

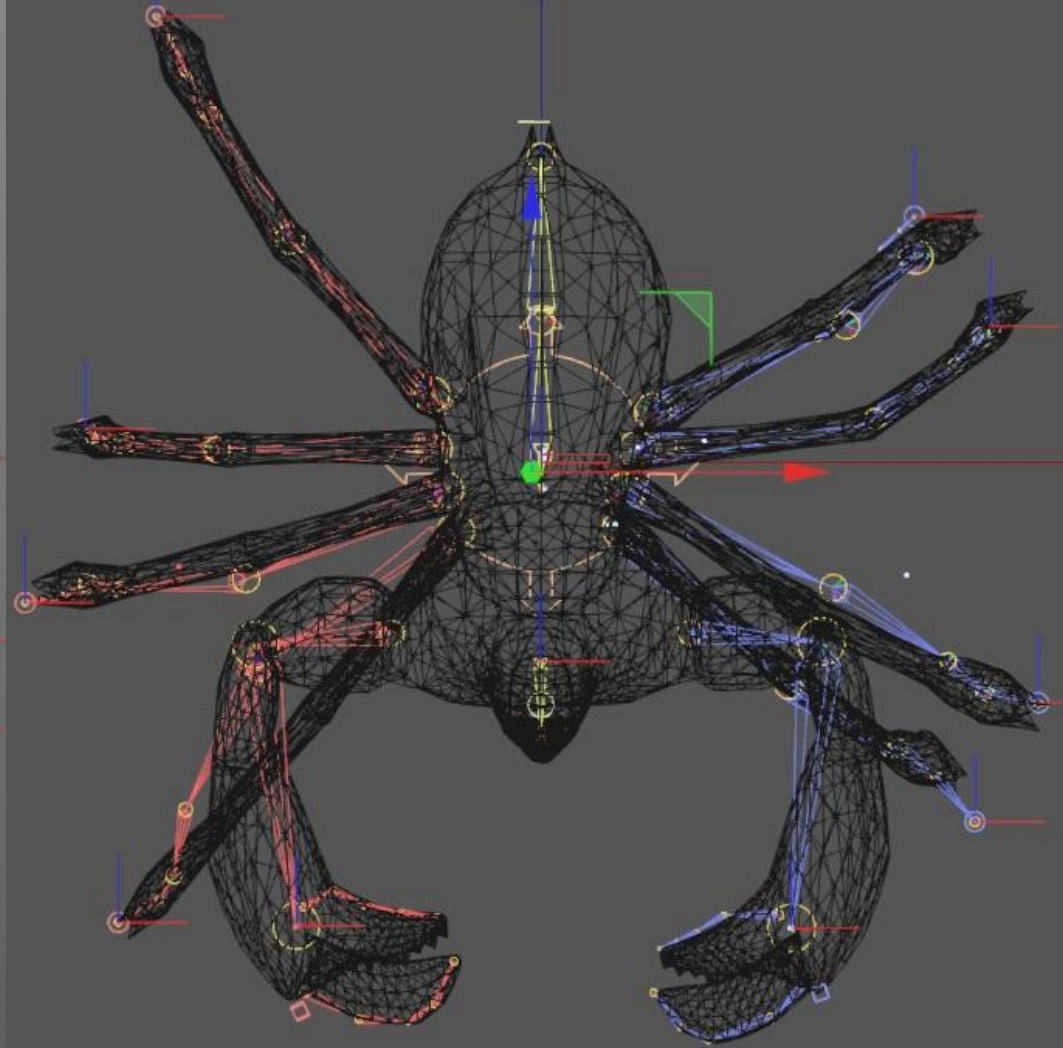
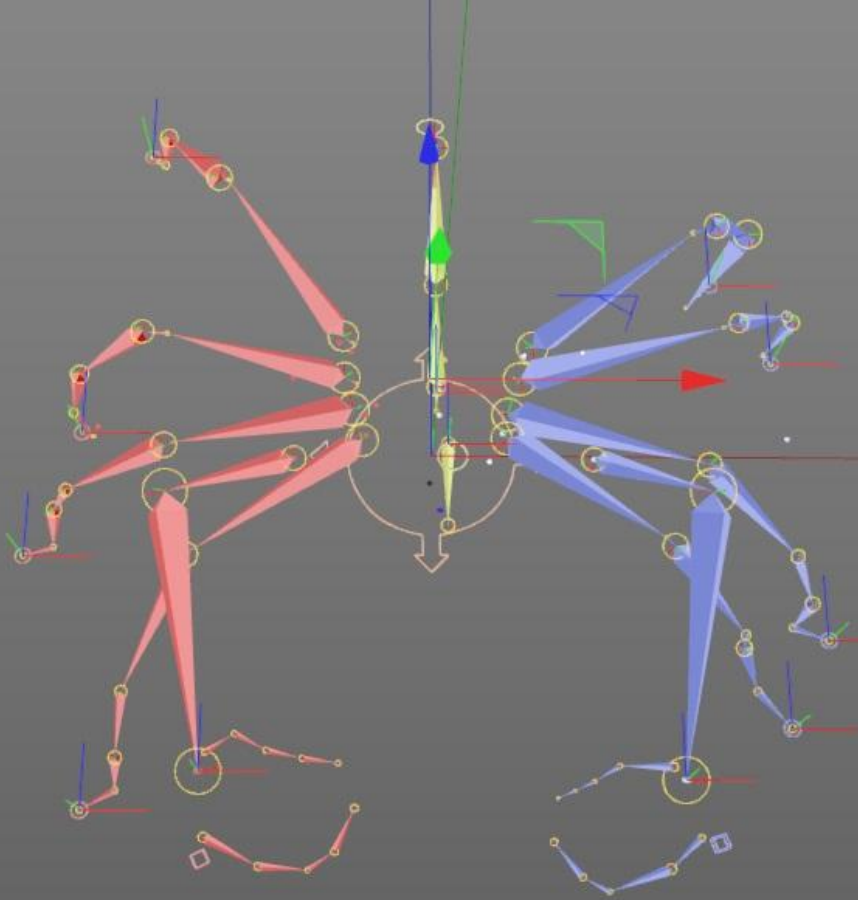
Bones

BLND_Neck Count 1

hierarchy: character (Root, Spine, L_Arm, L_Hand, R_Arm, L_Leg, R_Leg, Jaw, L_Eye)

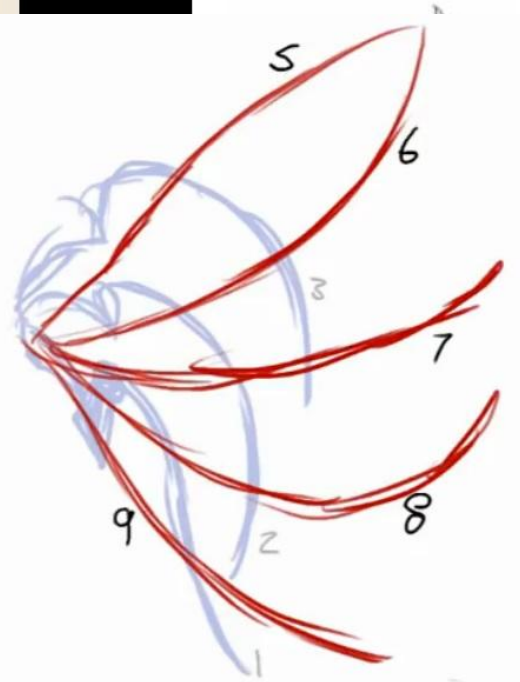
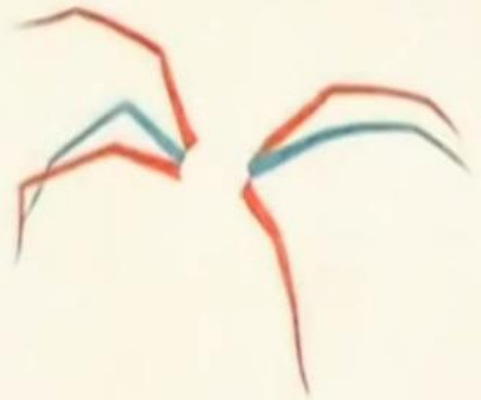
Mode Edit User Data

Content Browser Structure Layers Attributes

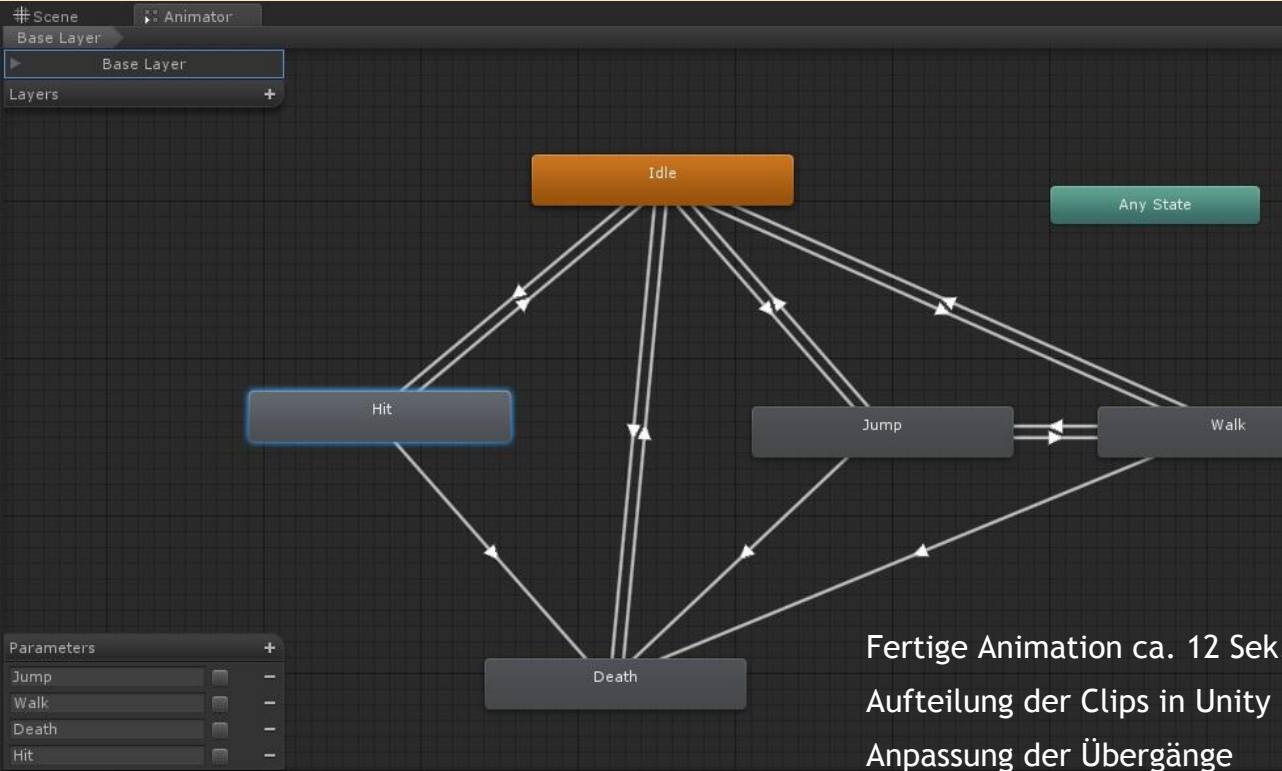


Animation

THE 3RD SET MIMIC THE FRONT ONES



Animation



Fertige Animation ca. 12 Sek.
Aufteilung der Clips in Unity
Anpassung der Übergänge

Inspector

Gorcrasp_Anim_Tex02 Import Settings

Model Rig Animations

Import Animation

Bake Animations

Anim. Compression Off

Clips	Start	End
Gorcrasp_Idle	0.0	30.0
Gorcrasp_Boring	30.0	180.0
Gorcrasp_Walk	180.0	197.0
Gorcrasp_Jump	198.0	220.0
Gorcrasp_Hit	220.0	240.0
Gorcrasp_Death	240.0	285.0
Menu_Gorcrasp_Idle	0.0	30.0
Menu_Gorcrasp_Boring	30.0	180.0

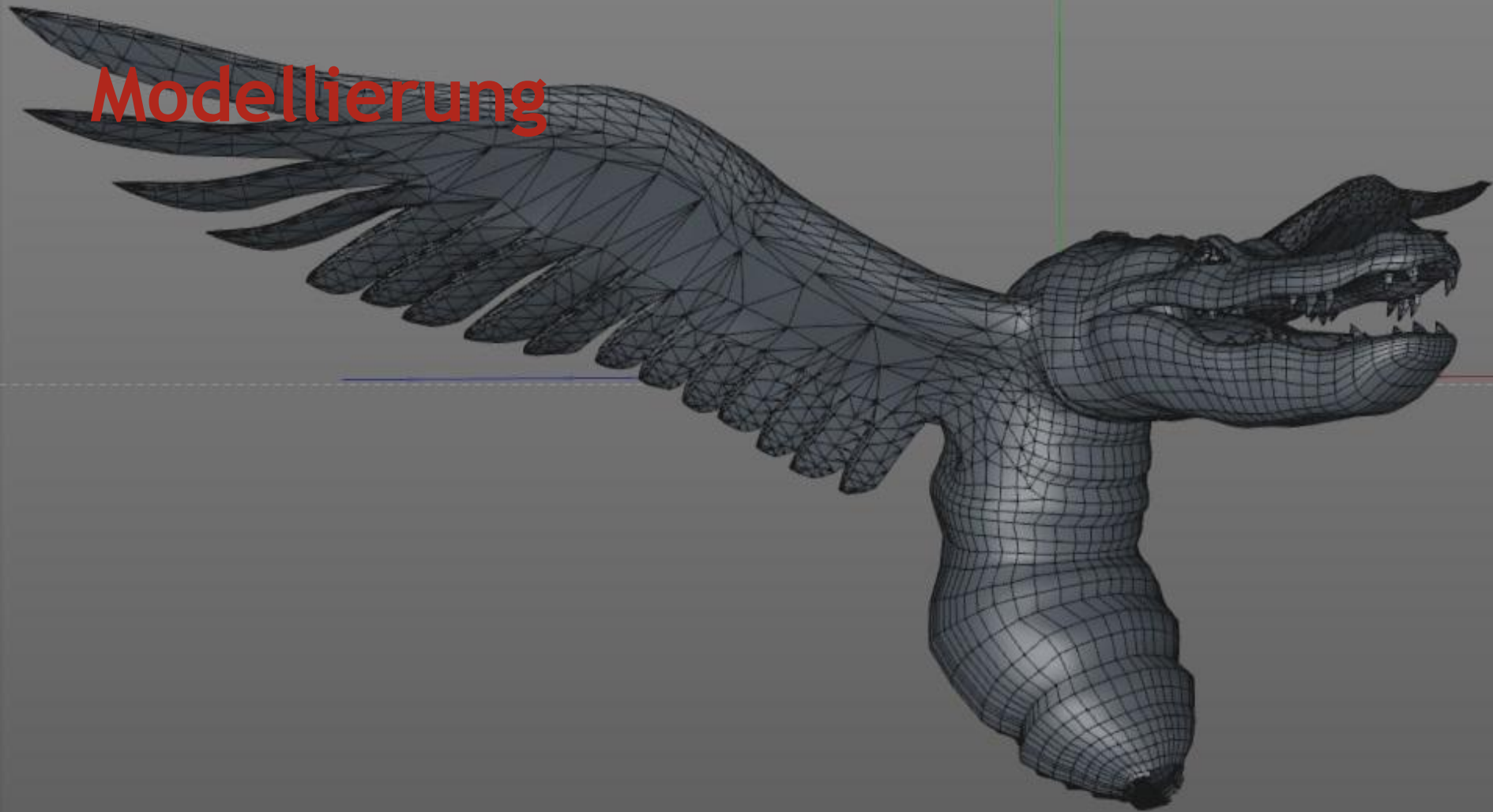
Preview 1.00

0:07 (039.6%)

Studie: Beecrocha



Modellierung



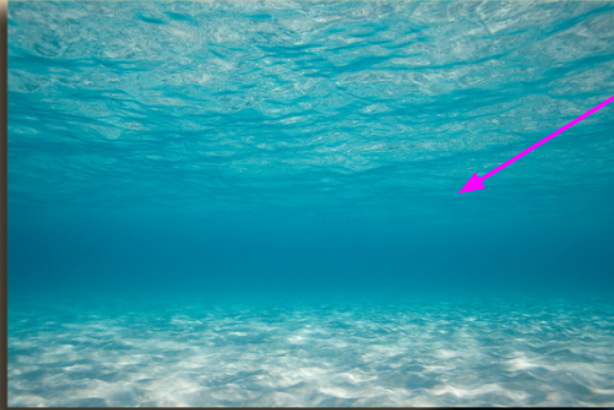
Texturierung



Wassereffekte



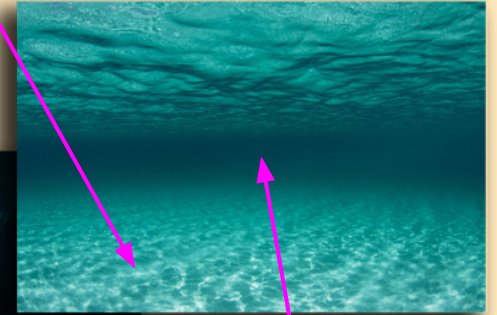
Wassereffekte



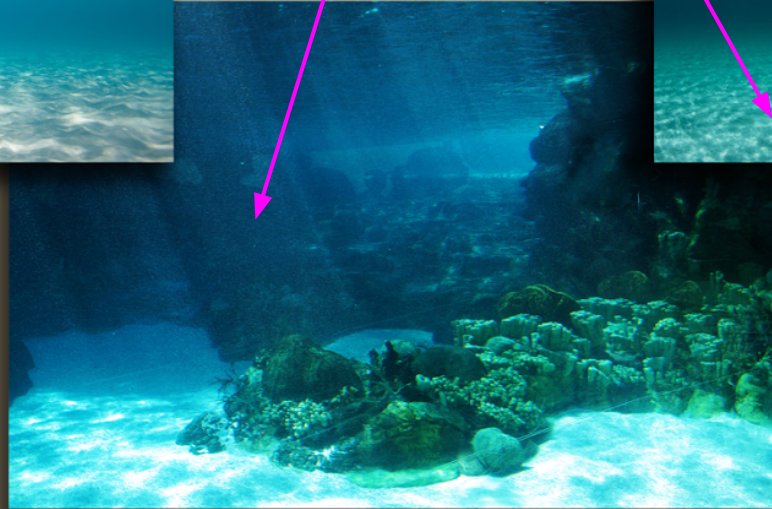
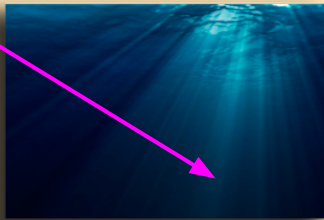
Reflektionen

Staubpartikel

Caustic



Sonnenstrahlen



Nebel

Reflektionen

Oberflächenreflektion → Unity Water
(Standard Asset)

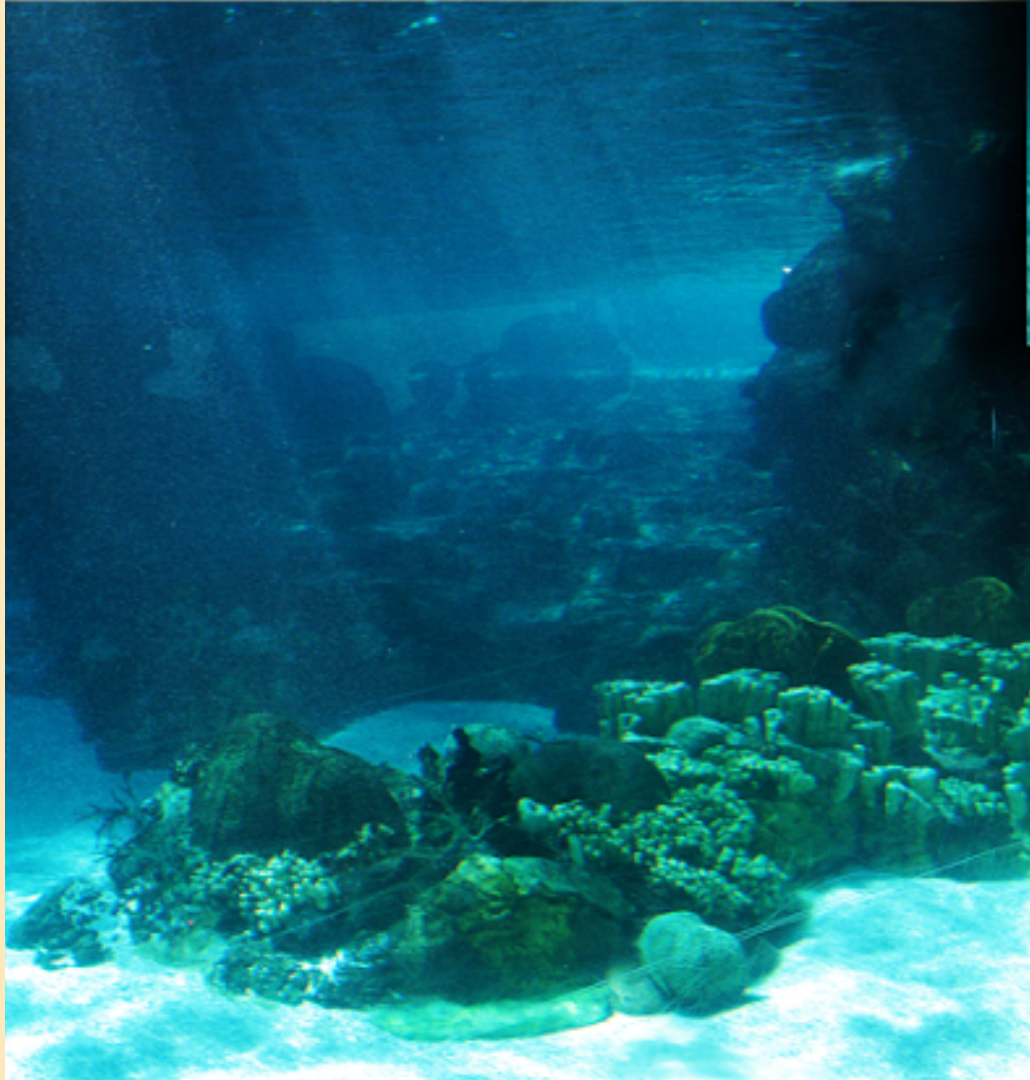
Sun Shafts (Sonnenstrahlen)



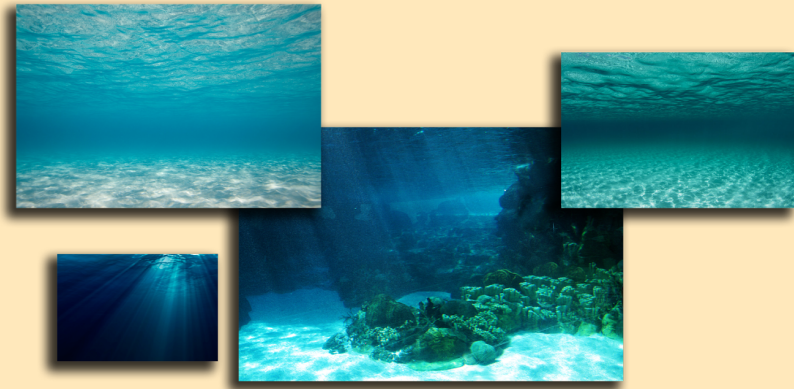
Nebel

Farbe → Color Correction Curves

Trüb → Global Fog



Verschwommen



Blur VS. Normalmap Distortion VS. Depth Of Field

Blur → einheitliches Weichzeichnen ← keine gute Optik

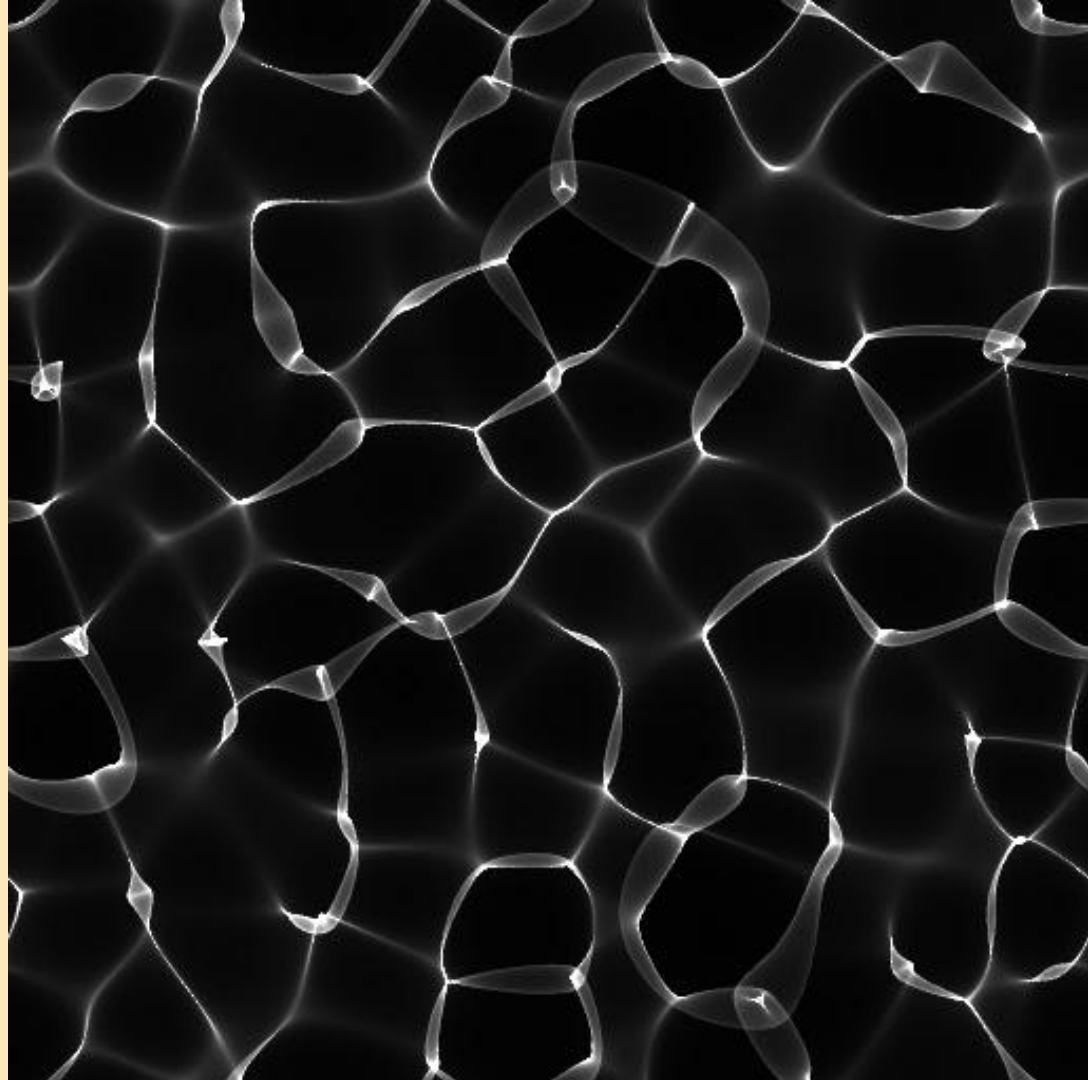
Depth Of Field → Gut einstellbare Tiefenunschärfe mit Focus auf Player

Normalmap Distortion → Animierte Wasserverwirbelungen ← gute Optik

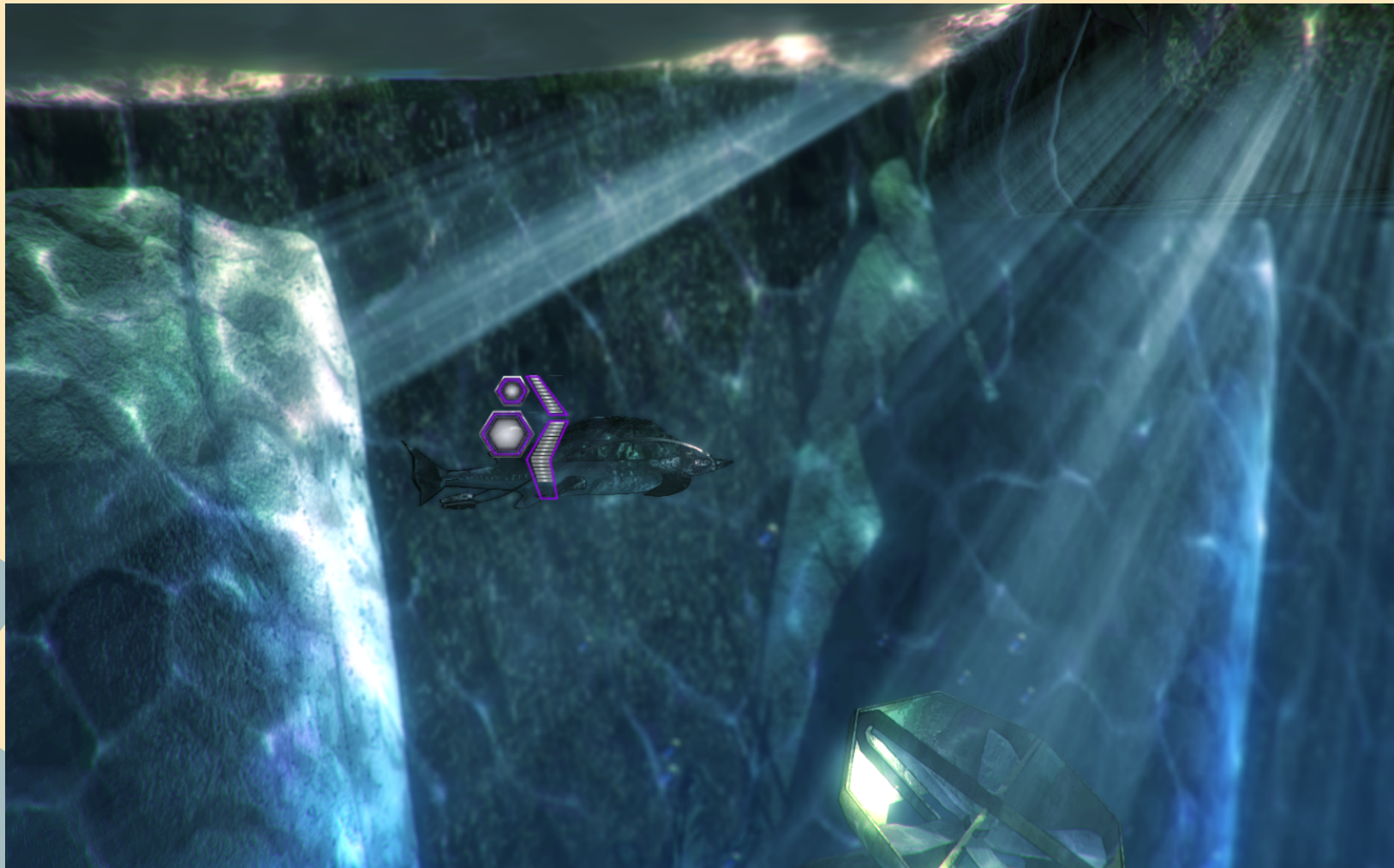
Normalmap Distortion mit Depth Of Field nicht Kompatible →
Tiefeninformation auf Plane mit Normalmap fast 0

Kaustiken

Projektion einer Caustic-
Animation auf das Wasserlevel
Kann mit prozeduralen Texturen
gerendert werden







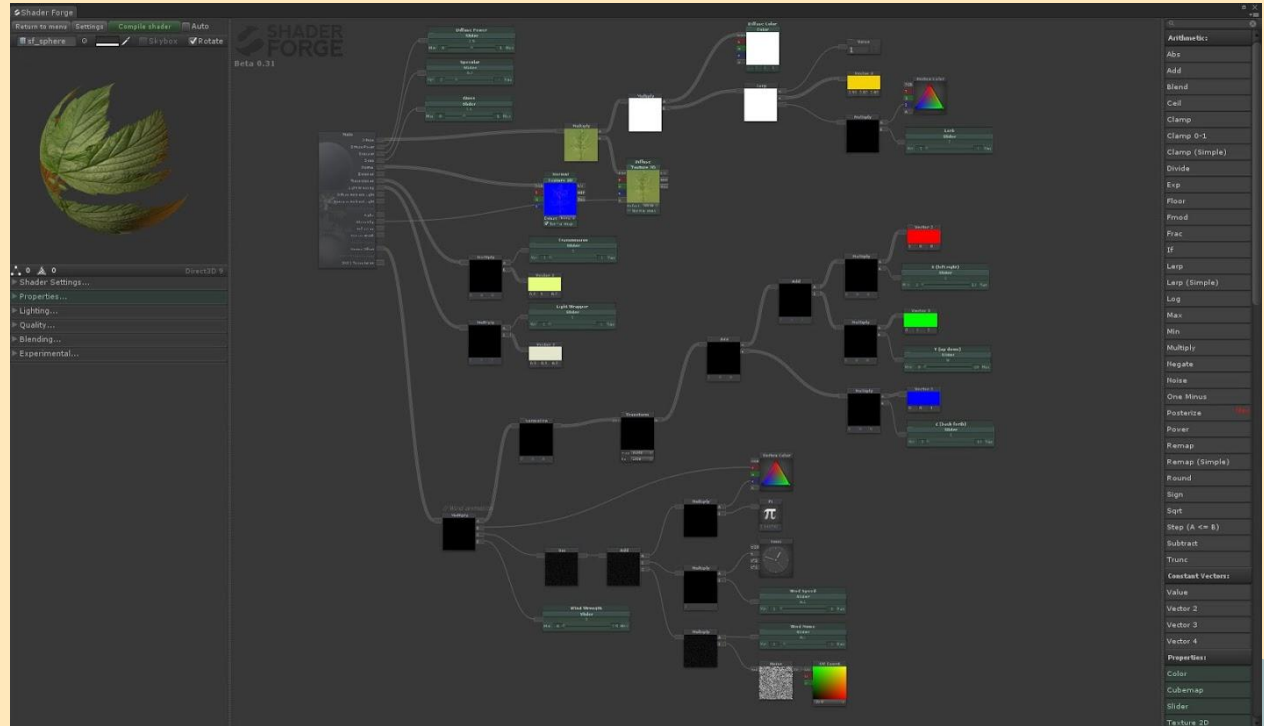
Teile der Wassereffekte konnten auch auf Air und Ground übertragen werden:

- Color Correction Curves
- Depth Of Field <- in abgeschwächter Form
- Sun Shafts
- Bloom
- Vignette

Shader Forge

(node-based shader editor)

Leaf Shader



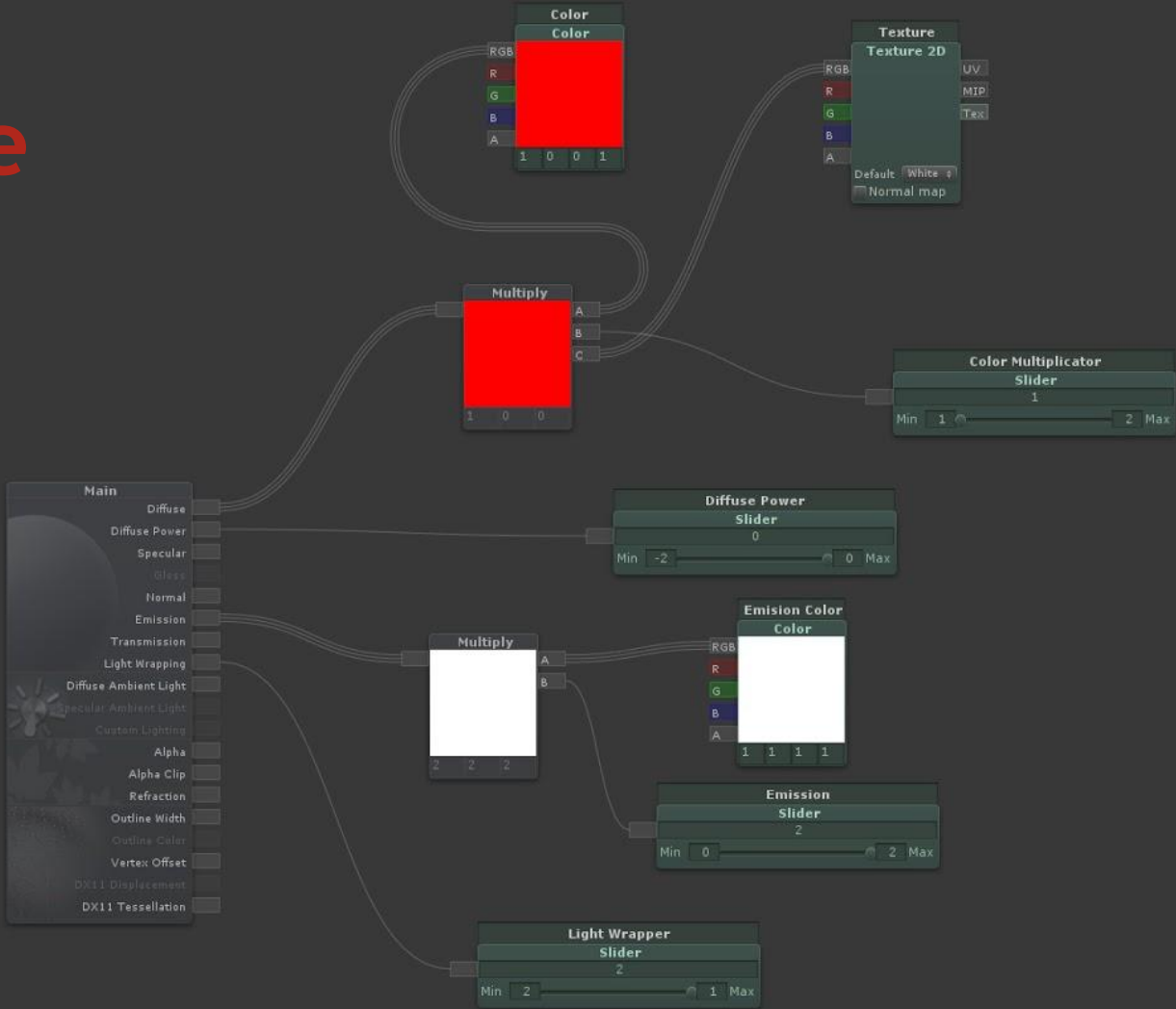
Leaf Shader

generiert über 400
Zeilen Code

```
Vegetation.shader
278 }
279 Cull Off
280 Offset 1, 1
281
282 Fog (Mode Off)
283 CULLPROGRAM
284 #pragma vertex vert
285 #pragma fragment frag
286 #define UNITY_PASS_SHADOWCASTER
287 #include "UnityCG.cginc"
288 #include "Lighting.cginc"
289 #pragma fragmentoption ARB_precision_hint_fastest
290 #pragma multi_compile_shadowcaster
291 #pragma exclude_renderers xbox360 ps3 flash d3d11_wx
292 #pragma target 3.0
293 uniform float4 _TimeEditor;
294 #ifndef LIGHTMAP_OFF
295 sampler2D unity_Lightmap;
296 float4 unity_LightmapST;
297 #ifdef DOUBLELIGHTMAP_OFF
298 sampler2D unity_LightmapInd;
299 #endif
300 #endif
301 uniform sampler2D _Diffuse; uniform float4 _Diffuse_ST;
302 uniform float _MindStrength;
303 uniform float _MindSpeed;
304 uniform float _MindNoise;
305 uniform float _XleftRight;
306 uniform float _Yupdown;
307 uniform float _Backferth;
308 struct VertexInput {
309     float4 vertex : POSITION;
310     float4 uv0 : TEXCOORD0;
311     float4 vertexColor : COLOR;
312 };
313 struct VertexOutput {
314     V2F_SHADOW_CASTER;
315     float4 uv0 : TEXCOORD1;
316     float4 vertexColor : COLOR;
317 };
318 VertexOutput vert (VertexInput v) {
319     VertexOutput o;
320     o.uv0 = v.uv0;
321     o.vertexColor = v.vertexColor;
322     float4 node_289 = o.vertexColor;
323     float4 node_292 = _Time + _TimeEditor;
324     float2 node_1250 = o.uv0;
325     float2 node_1200_stew = node_1250.rg + 0.2127*node_1250.rg.w*0.2713*node_125
326     float2 node_1200_rnd = 4.759*sin(489.128*(node_1200_stew));
327     float node_1200 = frac(node_1200_rnd.x*node_1200_rnd.y*(1+node_1200_stew.w))
328     v.vertex.xyz += (normalise(mul(_World2Object, float4(1,0,0)+_Xlef
329     o.pos = mul(UNITY_MATRIX_MVP, v.vertex);
330     TRANSFER_SHADOW_CASTER(o);
331     return o;
332 }
333 fixed4 frag(VertexOutput i) : COLOR {
334     float2 node_2351 = i.uv0;
335     float4 node_1 = tex2D(_Diffuse, TRANSFORM_TEX(node_2351.rg, _Diffuse));
336     clip(node_1.a - 0.5);
337     SHADOW_CASTER_FRAGMENT(i)
338 }
339 #ENDCG
340 }
341 #Fallback "Transparent/Custom/Diffuse"
342 CustomEditor "ShaderForgeMaterialInspector"
343 }
```

Shader Forge

Laser Shader

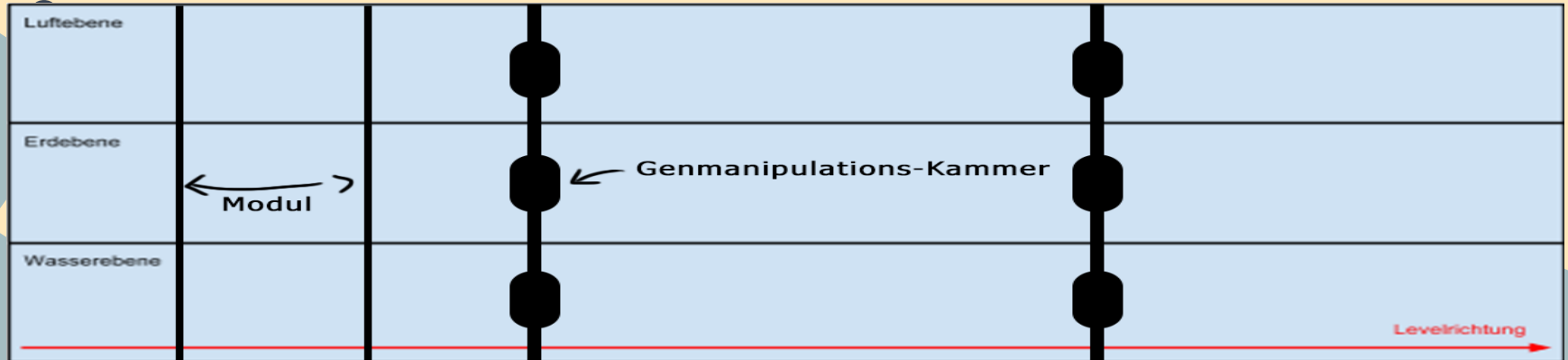


Leveldesign - Allgemein

- Bewegung auf “Gleisen” daher keine Bewegung in die Tiefe
- Spiel auf Geschwindigkeit ausgelegt, daher keine Bewegung in die Höhe
- Also: 2D-Leveldesign in eine Richtung

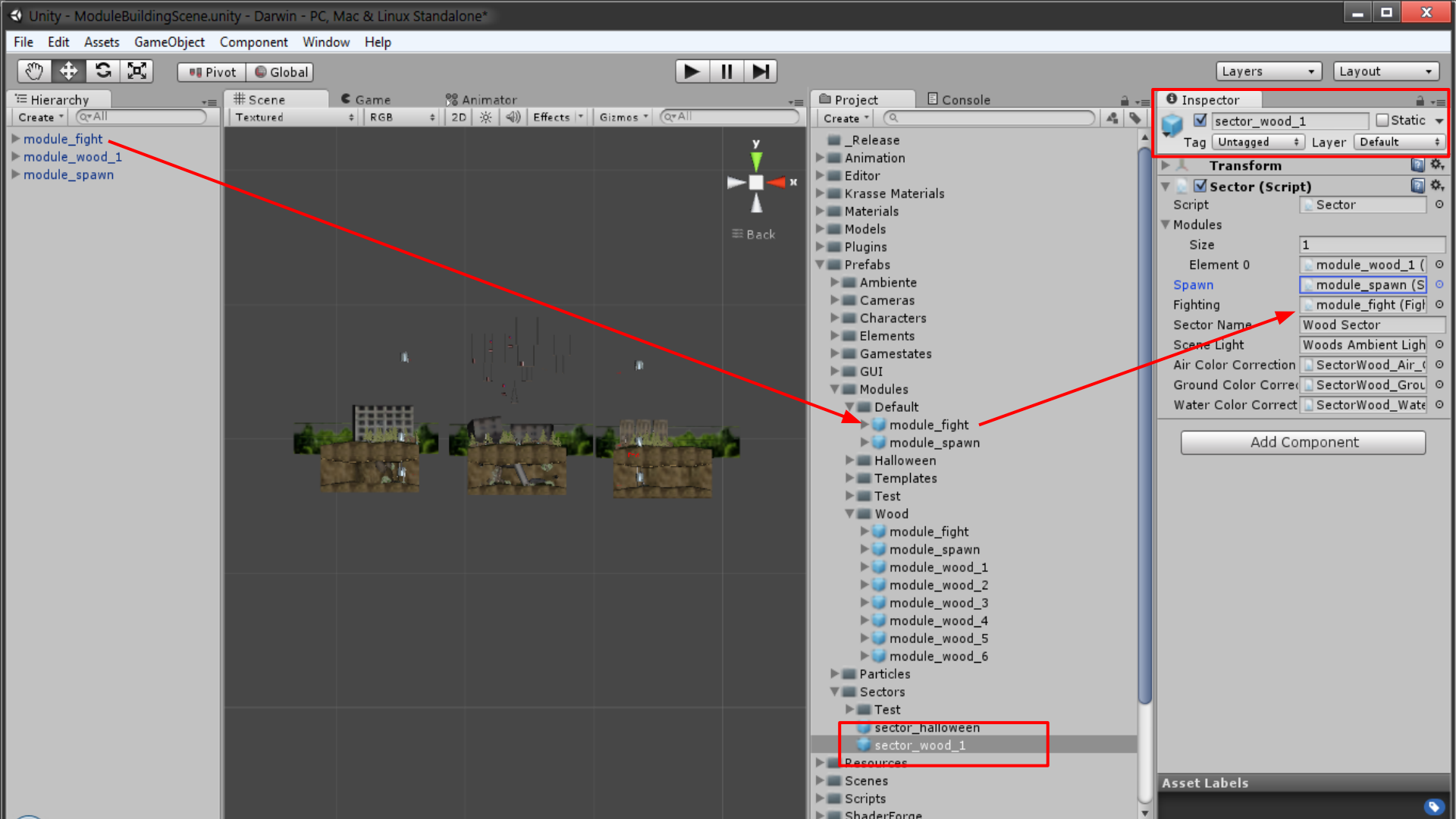
Leveldesign - Levelaufbau

- Level à 3 min mit 5 bis 7 Sektoren
- Ein Fight-/Spawnmodul, mehrere Spielmodule pro Sektor (Alle drei Ebenen pro Modul)



Leveldesign - Workflow

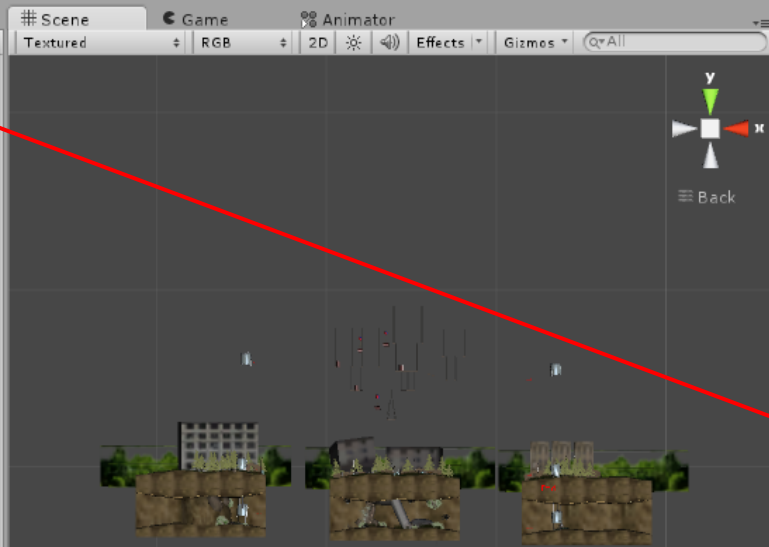
- Erstellen der Modell mit Animationen und Texturen in Blender/C4D
- Exportieren als fbx und Importieren in Unity
- Arrangieren der Modelle und Verknüpfen mit Logik
- Level mit Script als solches kennzeichnen und als Modul einem Sektor zuordnen



Hierarchy

Create Q All

- ▶ module_fight
- ▶ module_wood_1
- ▶ module_spawn



Project

Create Q

- ▶ _Release
- ▶ Animation
- ▶ Editor
- ▶ Krasse Materials
- ▶ Materials
- ▶ Models
- ▶ Plugins
- ▶ Prefabs
- ▶ Modules
 - ▶ Default
 - ▶ module_fight
 - ▶ module_spawn
 - ▶ Halloween
 - ▶ Templates
 - ▶ Test
 - ▶ Wood
 - ▶ module_fight
 - ▶ module_spawn
 - ▶ module_wood_1
 - ▶ module_wood_2
 - ▶ module_wood_3
 - ▶ module_wood_4
 - ▶ module_wood_5
 - ▶ module_wood_6
- ▶ Particles
- ▶ Sectors
 - ▶ Test
 - ▶ sector_halloween
 - ▶ sector_wood_1
- ▶ Resources
- ▶ Scenes
- ▶ Scripts
- ▶ ShaderForge

Inspector

sector_wood_1 Static

Tag Untagged Layer Default

Transform

Script Sector

Modules

Size 1

Element 0 module_wood_1 (

Spawn module_spawn (S

Fighting module_fight (Figh

Sector Name Wood Sector

Scene Light Woods Ambient Ligh

Air Color Correction SectorWood_Air_c

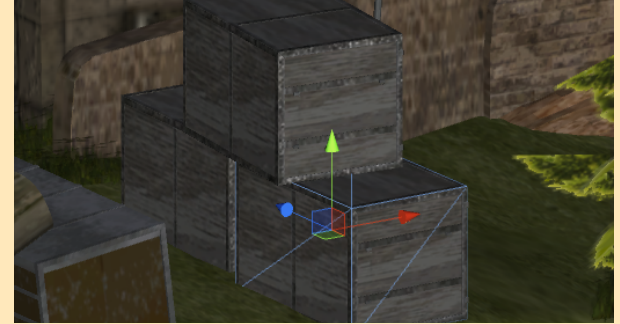
Ground Color Correc SectorWood_Grou

Water Color Correct SectorWood_Wate

Add Component

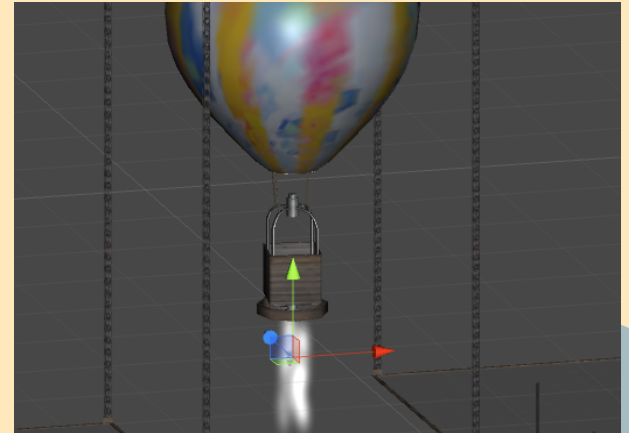
Leveldesign - Hindernisse

- Levelarchitektur/Passiv
 - Kisten, Fässer, Zäune
 - Gruben, Erhebungen
- Bewegend/2-Wege
 - Sichel, Minen, Kugelwillis, Düsenjäger



Leveldesign - Hindernisse

- Tödlich/Explosiv
 - Minen, Bomben, Laser
- Effektzonen
 - Windzonen
 - Wasserströmungen



Leveldesign - Dekoration

- **Objekte, Bilder im Hintergrund**
 - Erzeugt Tiefe und den Eindruck einer echten Welt
- **Beleuchtung, Lichteffekte**
 - Hervorheben von Objekten und Kaschieren von Fehlern
- **Color Correction**
 - Erzeugen je nach Thema unterschiedliche Stimmung

Inputhandling

- Inputhandling

- momentan nur Xbox-Controller “unterstützt”
- Verschiedene Treiber => verschiedene Mappings

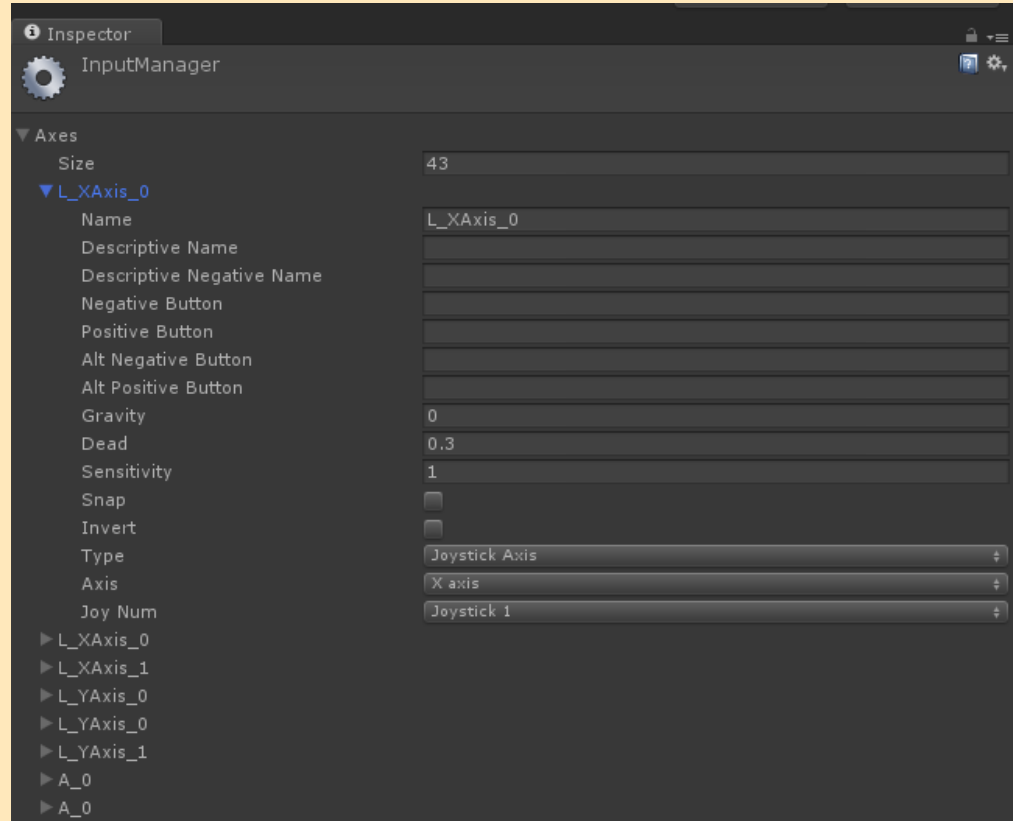


Xbox 360 Controller Layout for Unity
Using Tattie Bogle Mac OSX 360 Controller Driver



Inputhandling

- Inputmanager
 - Tastenzuweisung
 - Controllermapping
 - Spieler eindeutig identifizierbar

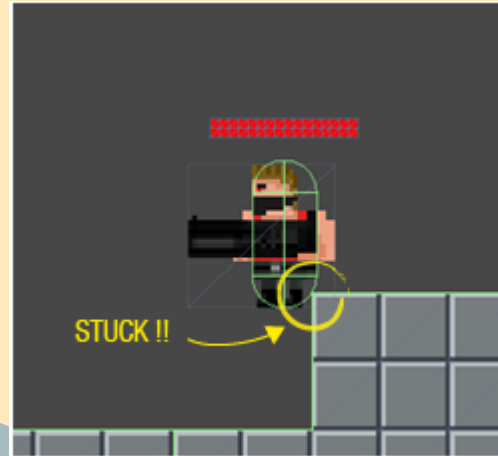


Inputhandling

- Problemstellungen
 - Simultaner Input
 - Gui Input
 - Zuweisung des Spielers
 - Achsen wie Knöpfe behandeln
 - Anmeldung eines Controllers

Movement und Collision

- Warum ein eigenes System?
 - Beide Systeme basieren auf der Physikengine
 - => Weniger Kontrolle, mehr Sonderfälle, mehr Arbeit
 - Rigidbodys
 - werden kinematisch verwendet
 - Charactercontroller
 - Unterstützt nur Capsule Collider
 - Collider ist nicht rotierbar
 - 90° Kanten



Movement und Collision

● Problemstellungen

- weitere Anforderungen während der Entwicklung
 - Slopedetection
 - Terrainrotation
- Movement
 - Input in Spielerbewegung umsetzen
 - Verknüpfung mit dem Collisionssystem (bsp.Movementzone)
 - nicht Einhaltung des Updatezyklus => Jittering
 - Schwierigkeitsgrad vs Usability
 - Keine Zauberformel => Trial-and-Error

Movement und Collision

● Problemstellungen

- weitere Anforderungen während der Entwicklung
 - Slopedetection
 - Terrainrotation
- Movement
 - Input in Spielerbewegung umsetzen
 - Verknüpfung mit dem Collisionssystem (bsp.Movementzone)
 - nicht Einhaltung des Updatezyklus => Jittering
 - Schwierigkeitsgrad vs Usability
 - Keine Zauberformel => Trial-and-Error



Movement und Collision

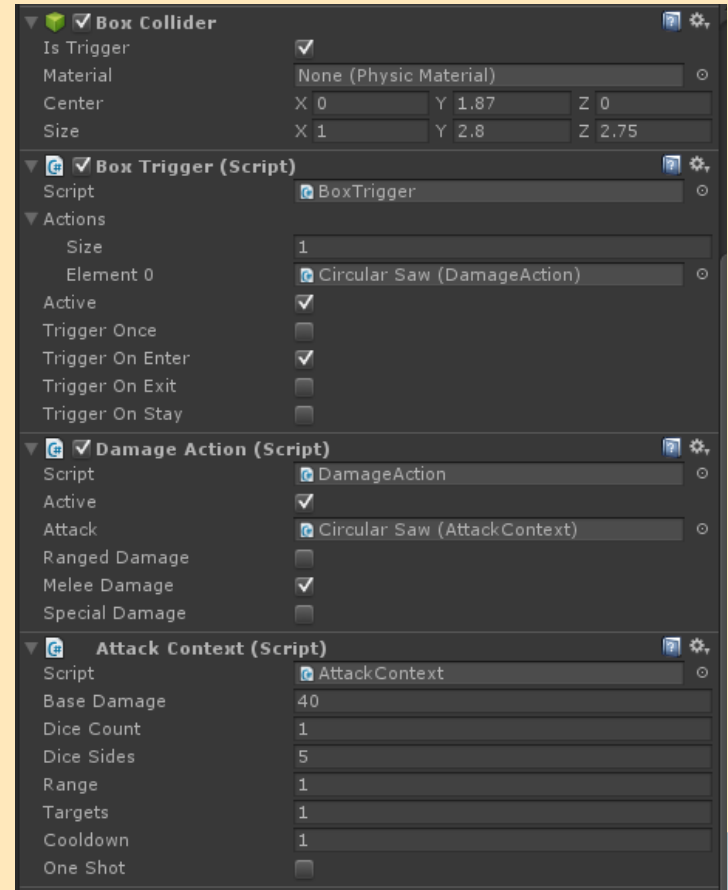
- Collision Messages
 - Kollisionsnachricht wird versandt
 - Objekte kollidieren
- Trigger Messages
 - Triggernachricht wird versandt
 - Objekte kollidieren nicht

Unity Collision Matrix

	Static Collider	Rigidbody Collider	Kinematic Rigidbody Collider	Static Trigger Collider	Rigidbody Trigger Collider	Kinematic Rigidbody Trigger Collider
Collision messages						
Static Collider		Y				
Rigidbody Collider	Y	Y	Y			
Kinematic Rigidbody Collider		Y				
Static Trigger Collider						
Rigidbody Trigger Collider						
Kinematic Rigidbody Trigger Collider						
Trigger messages						
Static Collider						Y Y
Rigidbody Collider				Y	Y	Y
Kinematic Rigidbody Collider				Y	Y	Y
Static Trigger Collider		Y	Y		Y	Y
Rigidbody Trigger Collider	Y	Y	Y	Y	Y	Y
Kinematic Rigidbody Trigger Collider	Y	Y	Y	Y	Y	Y

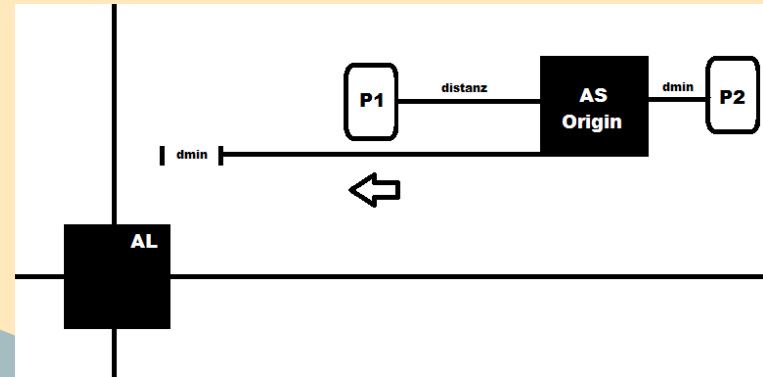
Levelelemente

- Triggersystem
 - Trigger → Boxen, Raycasts, Partikel
 - Actions → Schaden, Partikel, Sounds
 - Trigger löst eine oder mehrere Actions aus



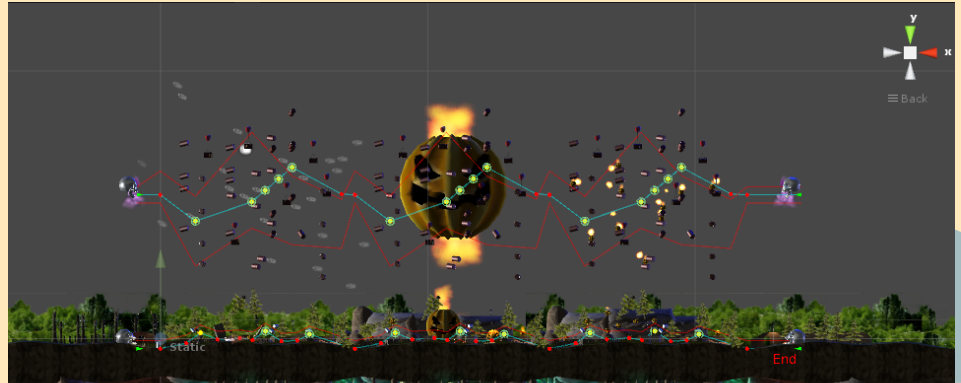
Levelelemente

- Sound
 - Übliches Set-Up
 - Audiolistener an der Camera
 - Audiosources im Level platziert
 - Splitscreen Sonderfall
 - Nur 1 Audiolistener aber 2 Spieler mit verschiedenen Standorten
 - => Lösung: Bewegende



Levelgenerierung

- Level: Level-Generator + Sektoren
- Level-Generator: Seed → Level-Daten
- Sektoren: Sammlung von Modulen
- Module: Spawn, Fight, Normal
 - Connectoren
 - Pathes
 - Spawns/Despawns



Levelgenerierung

- Auswahl eines Seeds
- Prüfen der maximalen Sektoren/Modulen
- Sektor-Generierung
 - Spawn-Modul Instanzieren
 - Level-Module Instanzieren
 - Fight-Modul Instanzieren
 - Path Instanzieren
 - Gen-Punkte Instanzieren

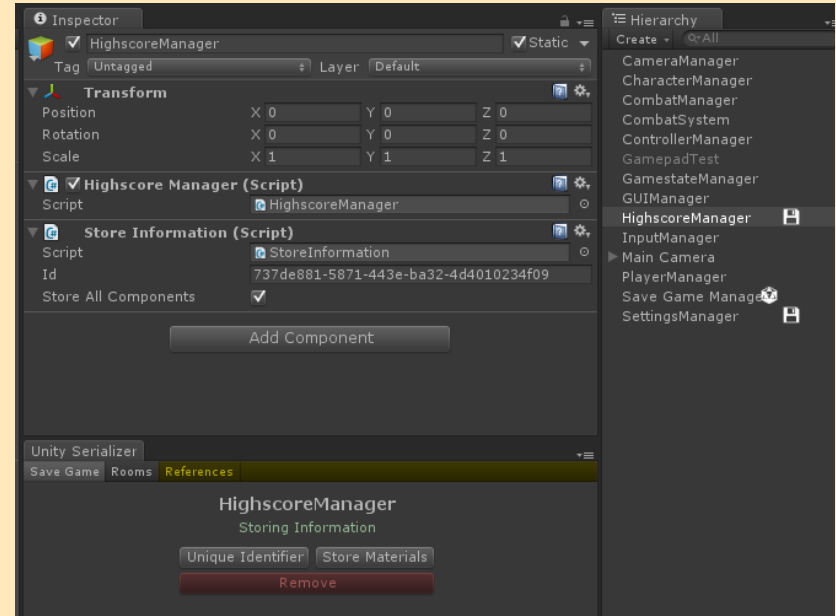
Serialization

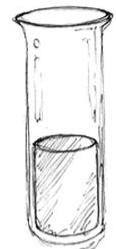
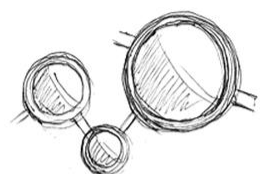
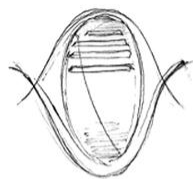
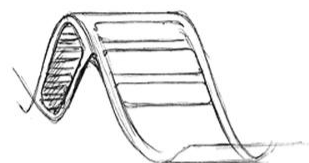
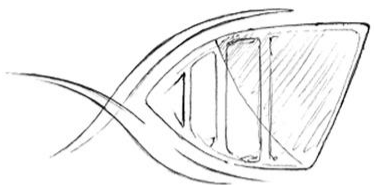
- Unity-Serializer

- kostenloses Plugin
- einfach zu Nutzen
- komplette Speicher-/Lade-Logik

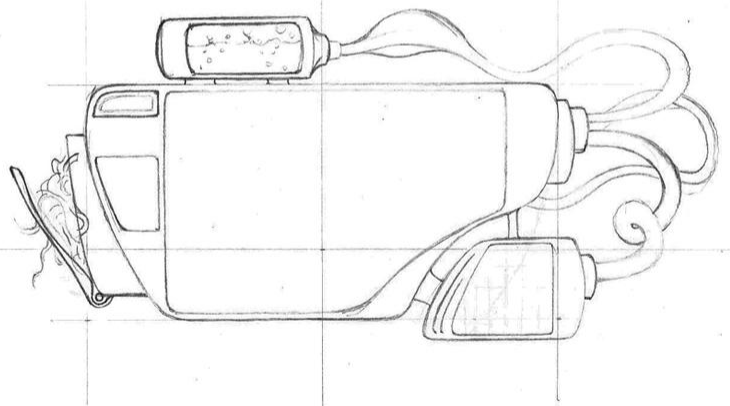
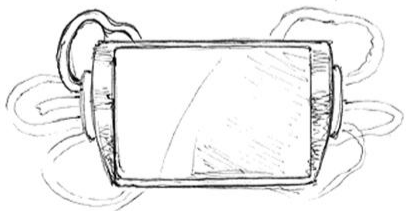
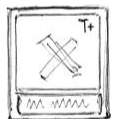
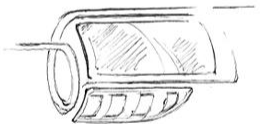
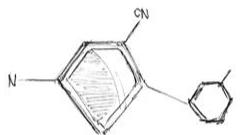
- Speichern und Laden

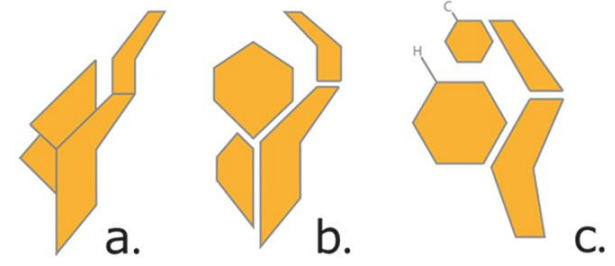
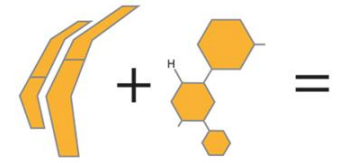
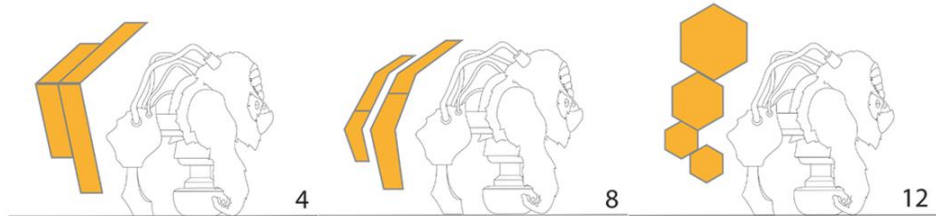
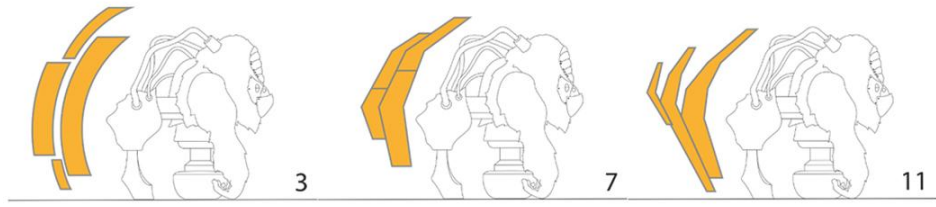
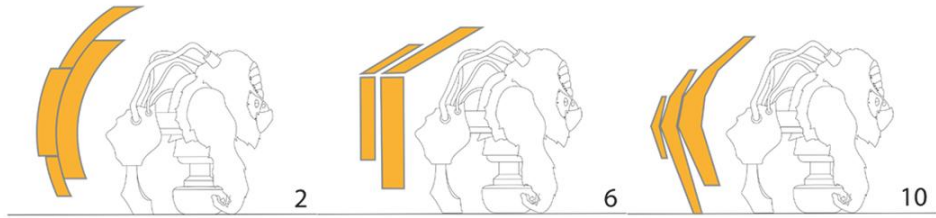
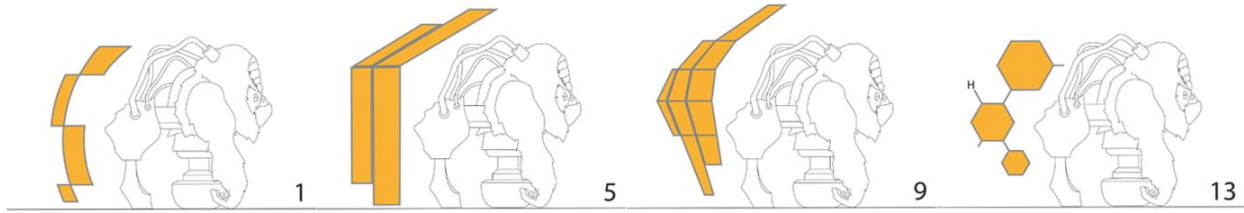
- Highscores und Seeds

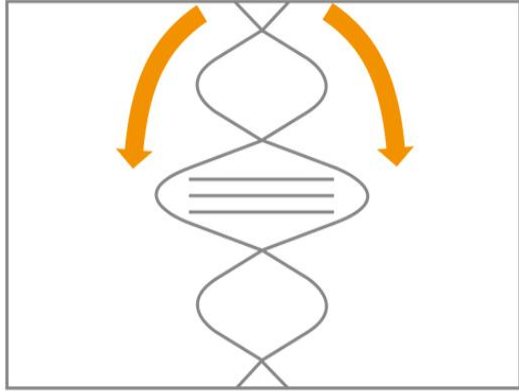
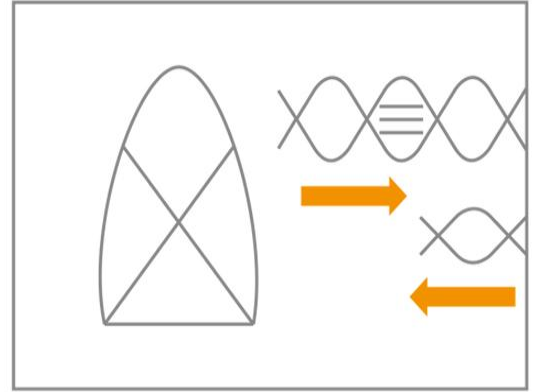
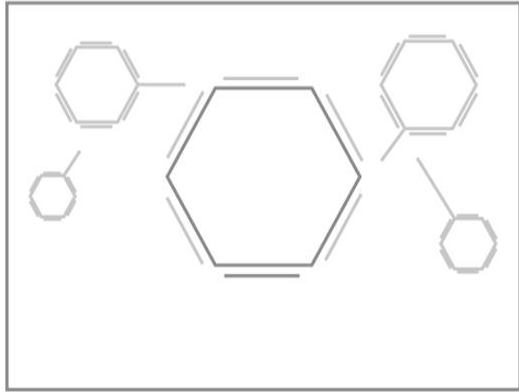
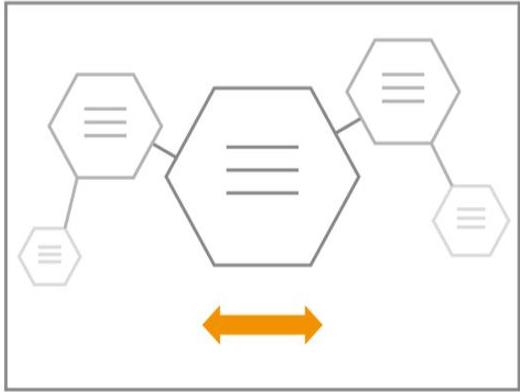


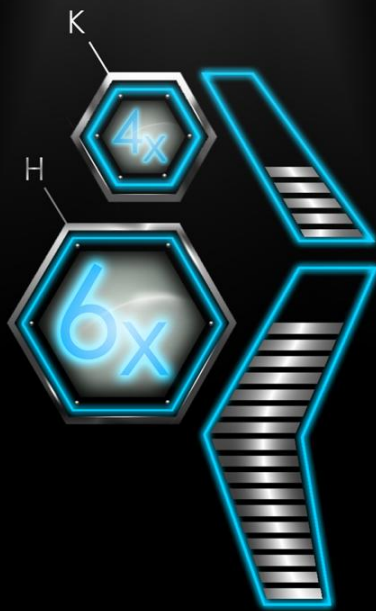
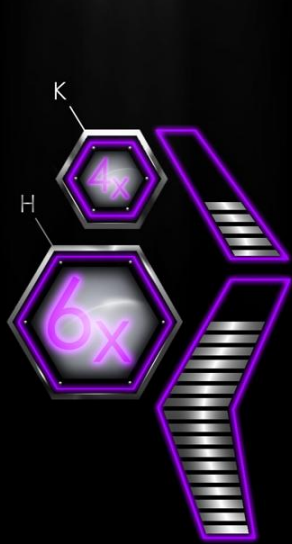


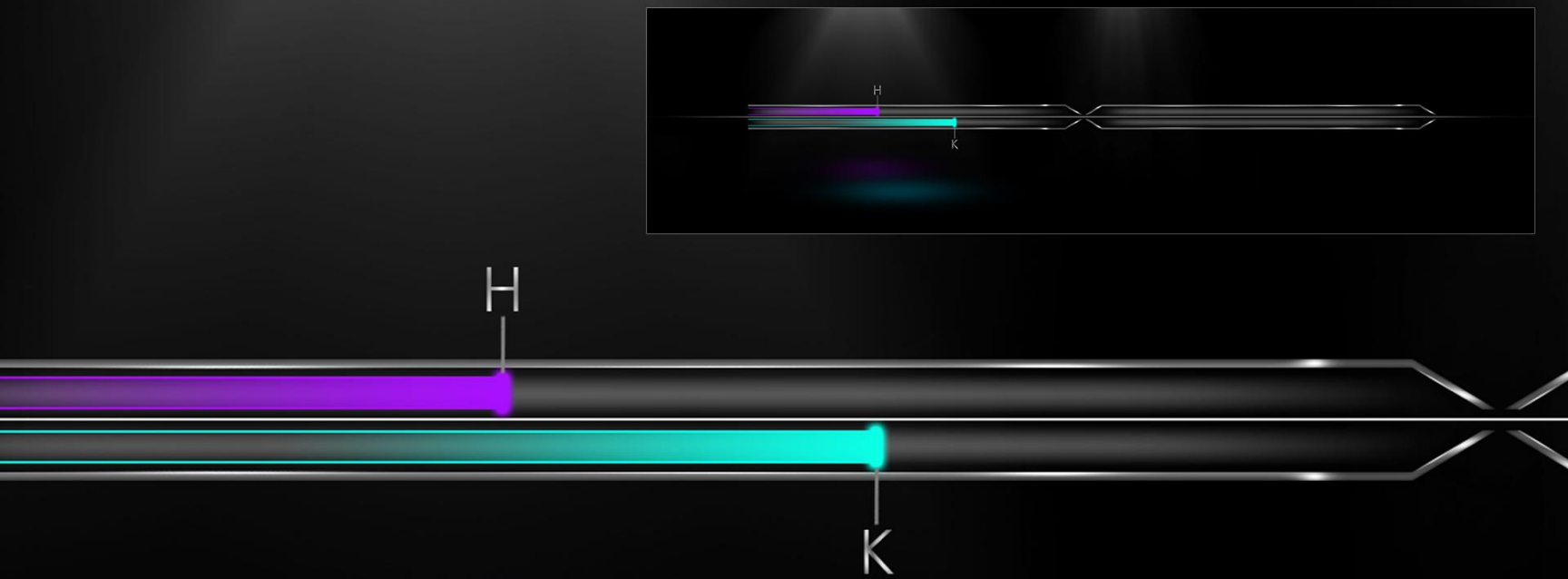
UI Design











Verlangsamen



Beschleunigen



Steuerung Manipulieren

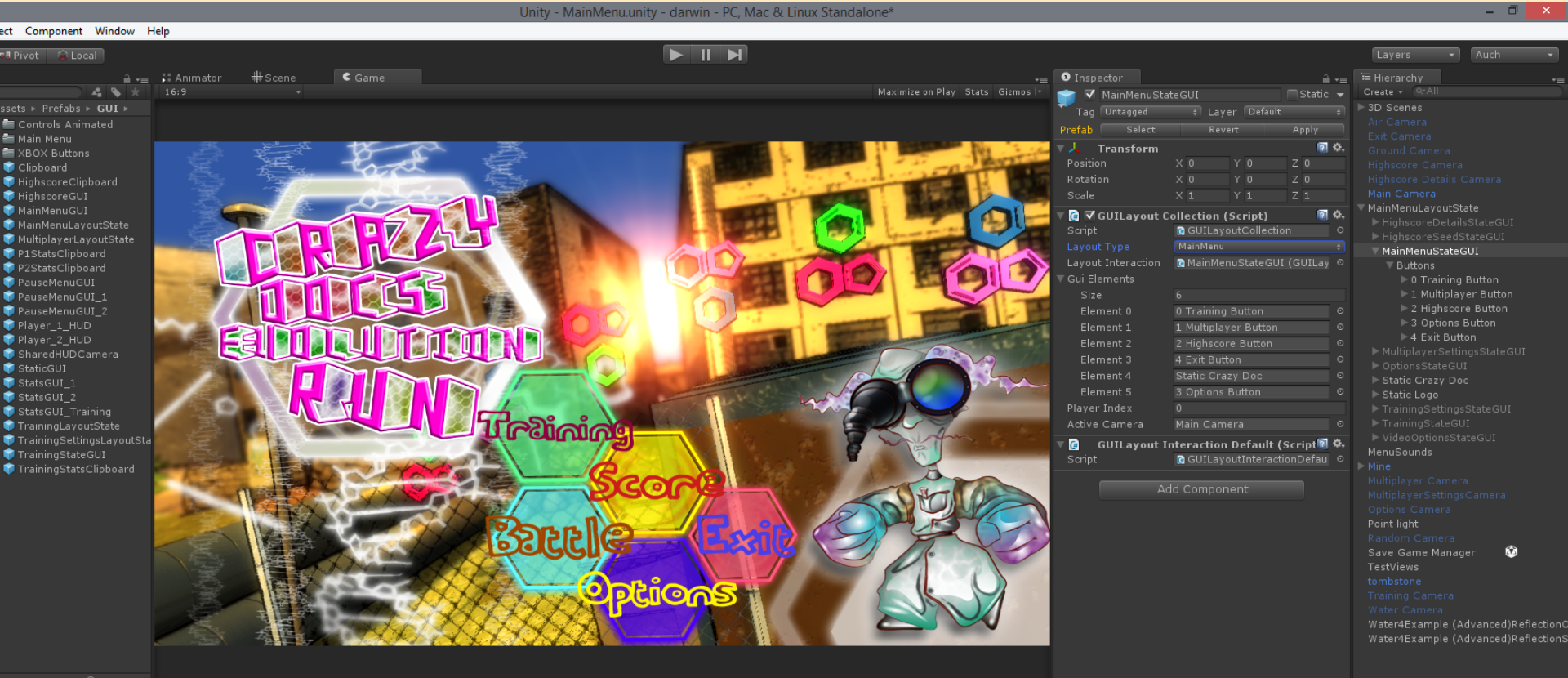


Sicht Einschränken





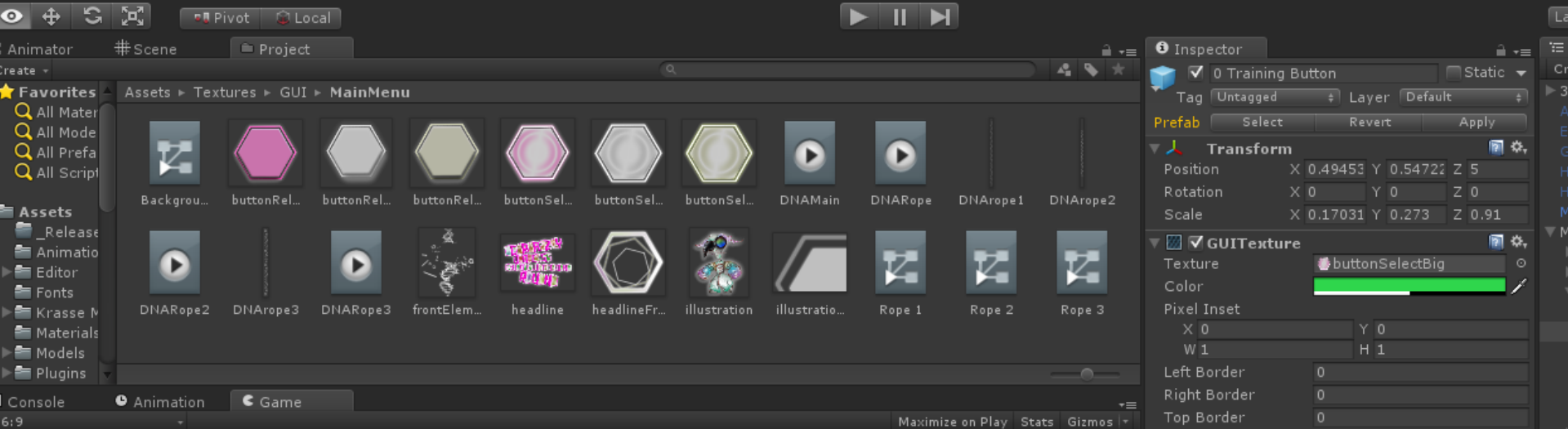
UI Erstellen



UI Erstellen

Unity - MainMenu.unity - darwin - PC, Mac & Linux Standalone*

File Edit Assets GameObject Component Window Help



UI Erstellen

Assembly-CSharp - Scripts\GUI\GUIManager.cs* - MonoDevelop-Unity

File Edit View Search Project Build Run Version Control Tools Window Help

Debug Default MonoDevelop-Unity

Solution

- LevelData.cs
- LevelGenerator.cs
- Player
 - Player.cs
 - PlayerManager.cs
 - PlayerStats.cs
- Settings
 - SettingsManager.cs
- Types
 - GamestateType.cs
 - MovementType.cs
- Util.cs
- Display
 - Camera
 - CameraConfiguration.cs
 - CameraController.cs
 - CameraManager.cs
 - MainMenuCamera.cs
- GUI
 - Layouts
 - EndgameHighscore

GUIManager ▶ No selection

```
15 // The general GUI manager for UI handling...
18 public class GUIManager : MonoBehaviour, IInputListener
19 {
20     public static GUIManager Get () ...
29
30     void Start () ...
39
40     void Update () ...
44
45     //////////////////////////////////////
46
47     // Adds a LayoutCollection to the list. Uses the...
51     public void AddLayout (GUILayoutCollection layout) ...
62
63     // Sets the current GUI layout for a player...
68     public void SetCurrentGUILayout (int playerIndex, GUILayoutCollectionType layoutType) ...
105
106     // Resets the GUI layout for both players...
109     void ResetGUILayout () ...
119
120     // Clears the GUI layout for both players...
123     public void ClearGUILayout () ...
136
```

Zocken?

