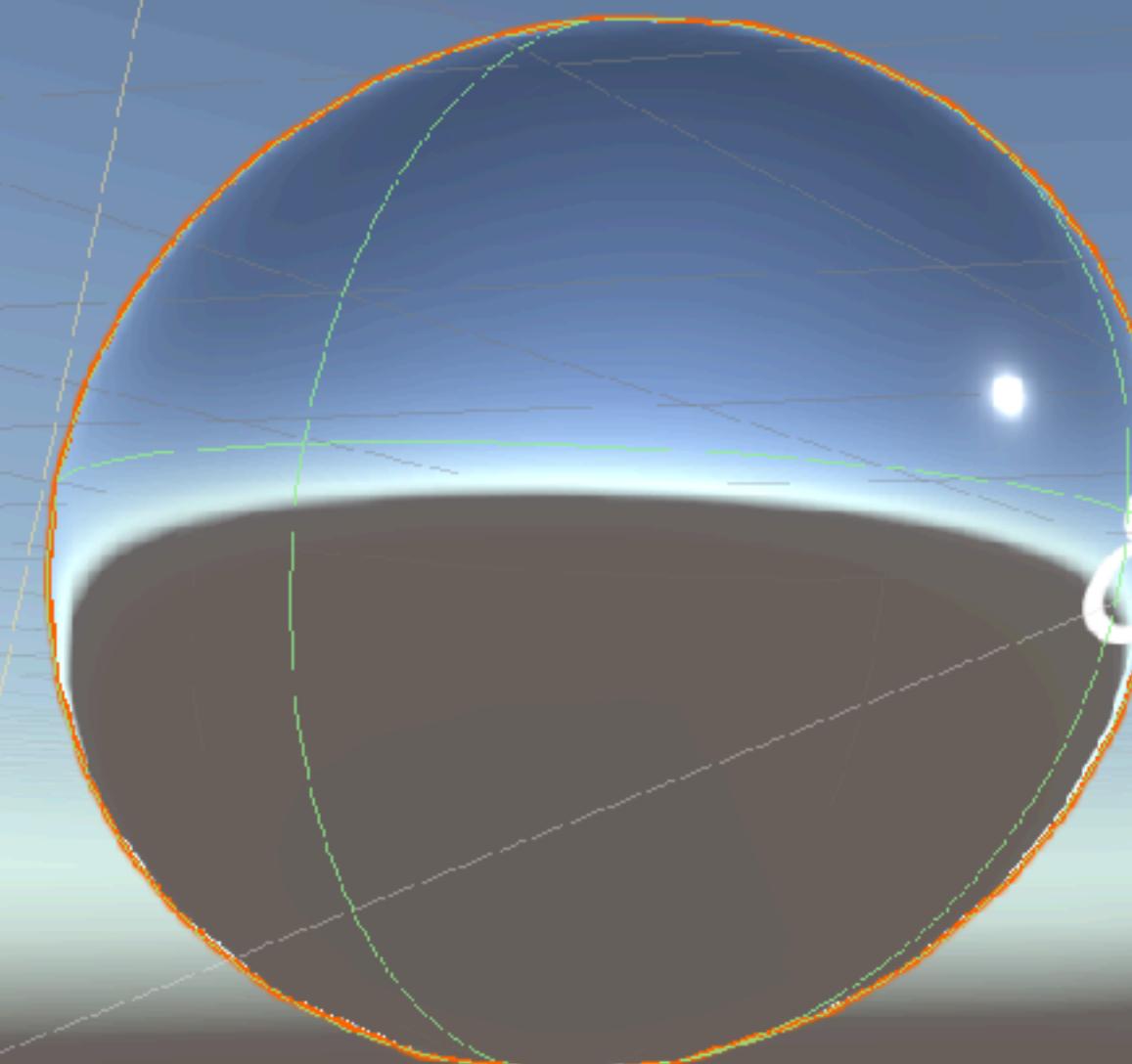




# Hello World

*Unity Einführung*

8.3.2021





Dichtestress, 2020

<https://www.zhaw.ch/index.php?id=13049>

**Unity Version 2019.4.21f1 (LTS)**

**oder neuer**

**oder älter, spielt nicht so eine Rolle**

**Bei den Modulen mindestens Build Support für die eigene Platform herunterladen, sonst nach belieben.**

👉 Game Engines

👉 Einführung Unity

👉 Tutorial: Walking Sim

 Game Engines

 Einführung Unity

 Tutorial: Walking Sim

 Game Engines?

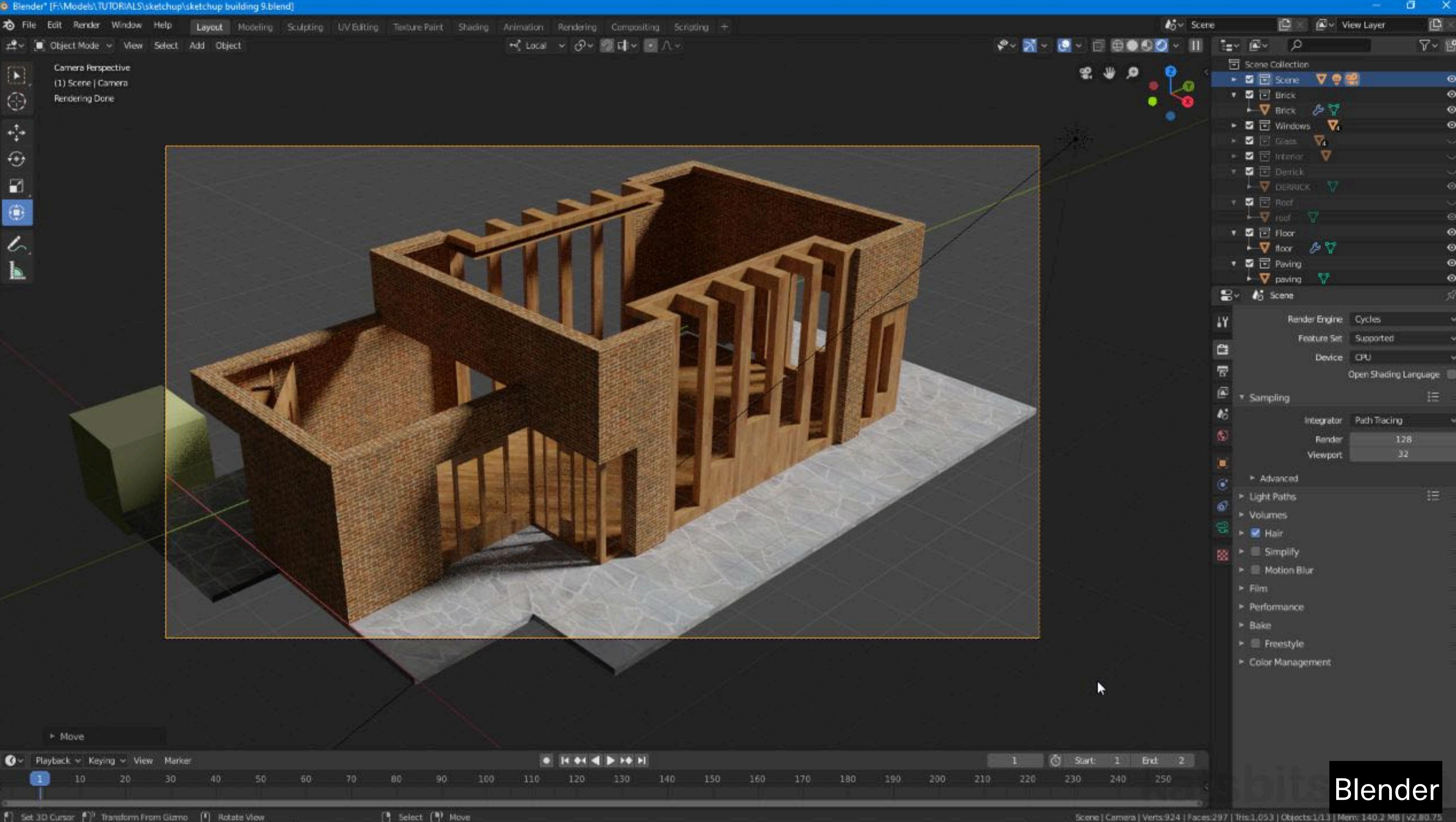
 Einführung Unity

 Tutorial: Walking Sim

# Unity ist eine Game Engine

# Was ist eine Game Engine?

# Was kann eine Game Engine?





Vectorworks

World Insert Layers



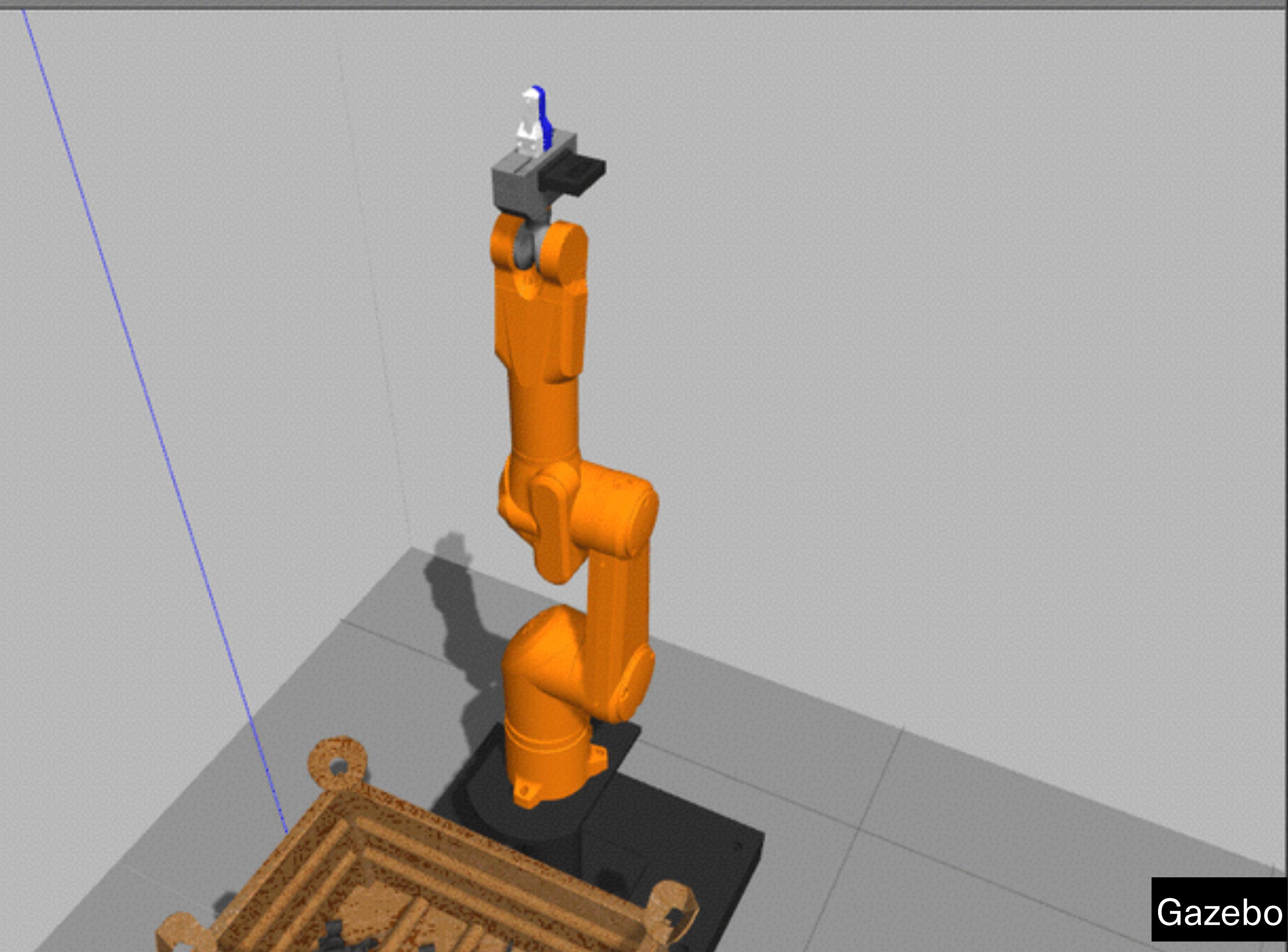
- pieza\_b1\_clone\_0
- pieza\_b1\_clone\_7
- pieza\_b1\_clone\_8
- pieza\_b1\_clone\_9
- pieza\_b1\_clone\_10
- pieza\_b1\_clone\_11
- pieza\_b1\_clone\_12
- pieza\_b1\_clone\_13
- pieza\_b1\_clone\_14
- pieza\_b1\_clone\_15
- pieza\_b1\_clone\_16
- pieza\_b1\_clone\_17
- pieza\_b1\_clone\_18
- pieza\_b1\_clone\_19
- ▼ ground\_plane

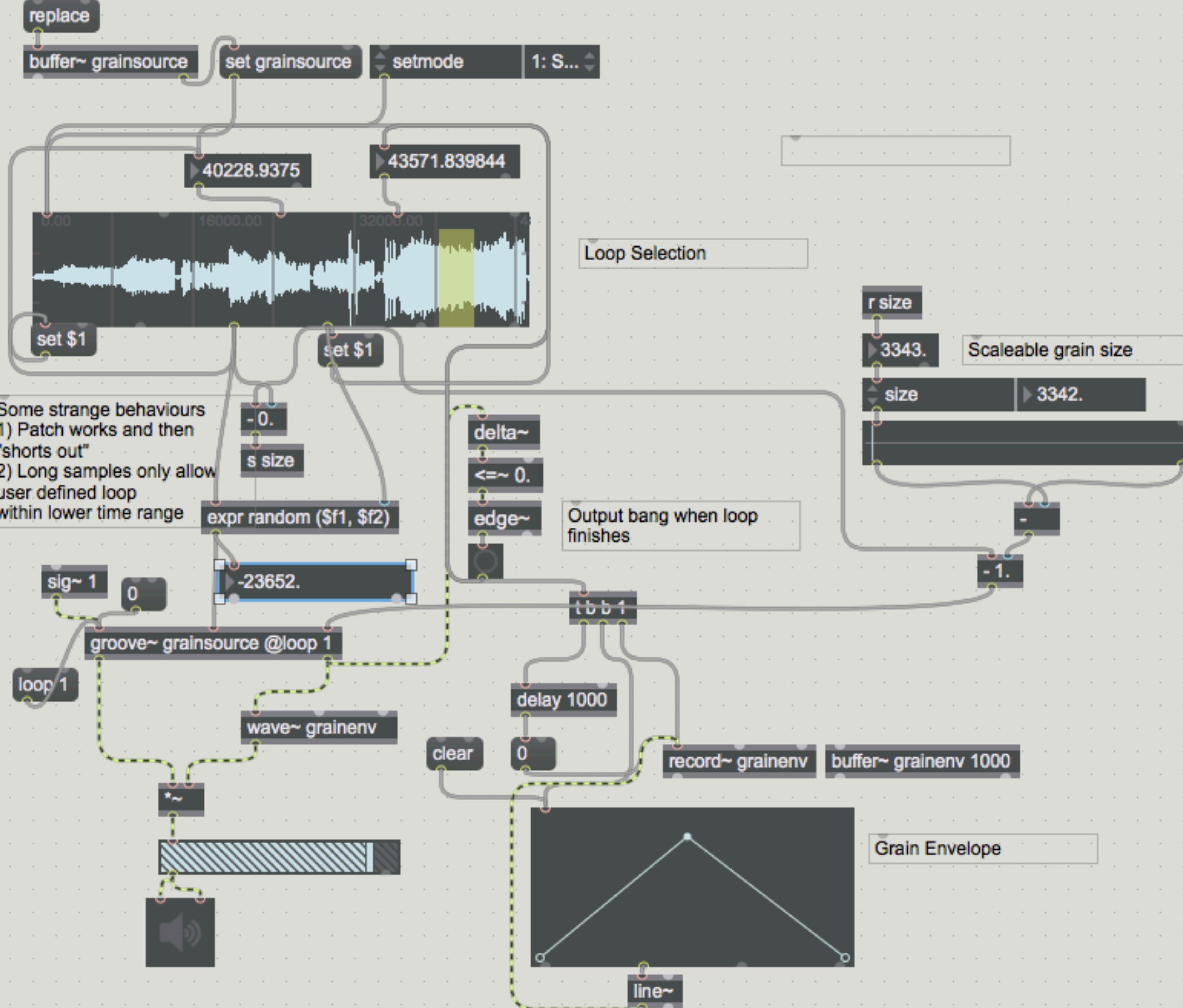
link

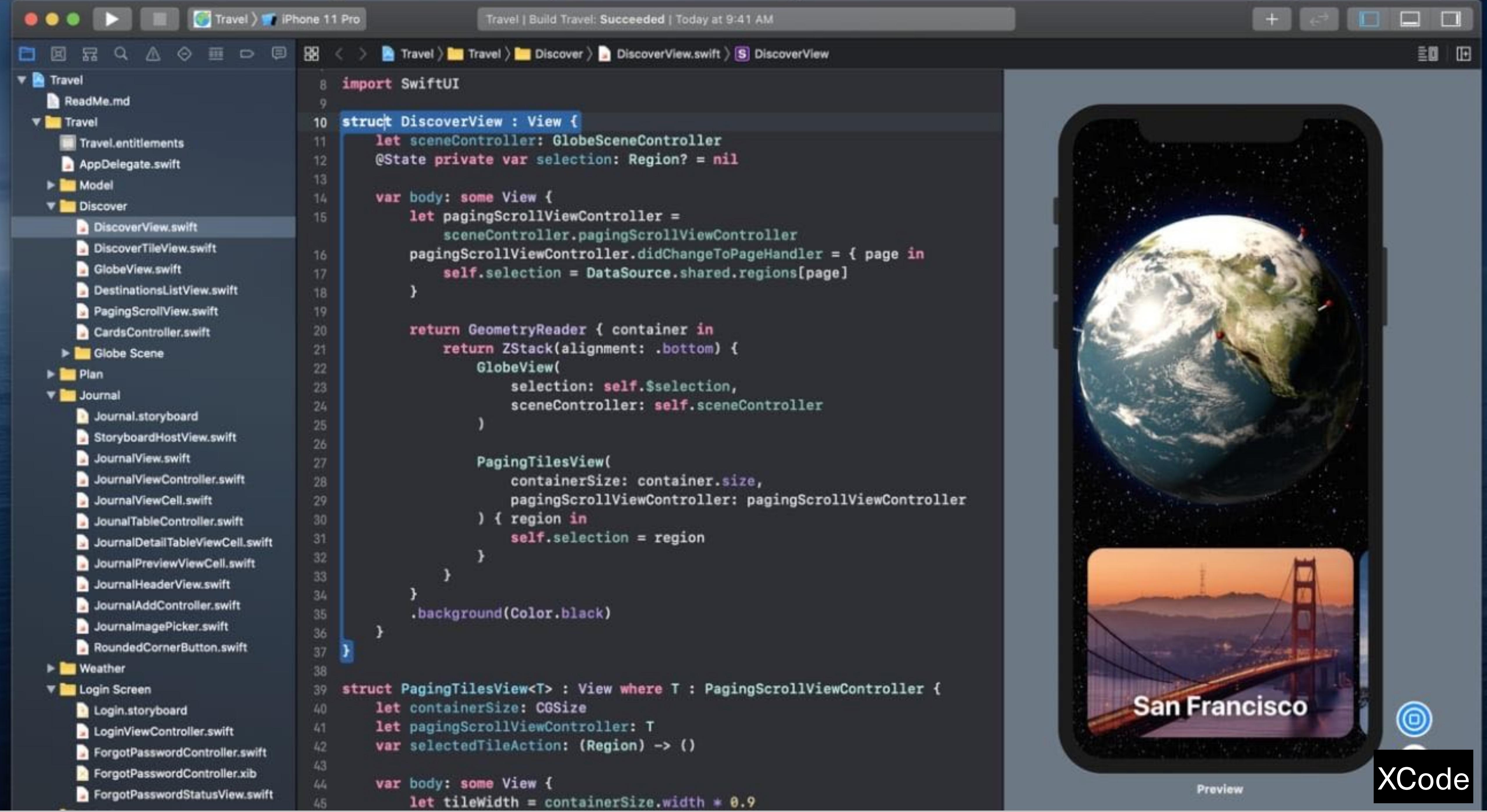
- Muro1

...

Property	Value
name	ground_plane
is_static	<input checked="" type="checkbox"/> True
self_collide	<input type="checkbox"/> False
► pose	
► link	ground_plane::link







Rendering Engine

Animation

Physics Engine

Artificial Intelligence

Input

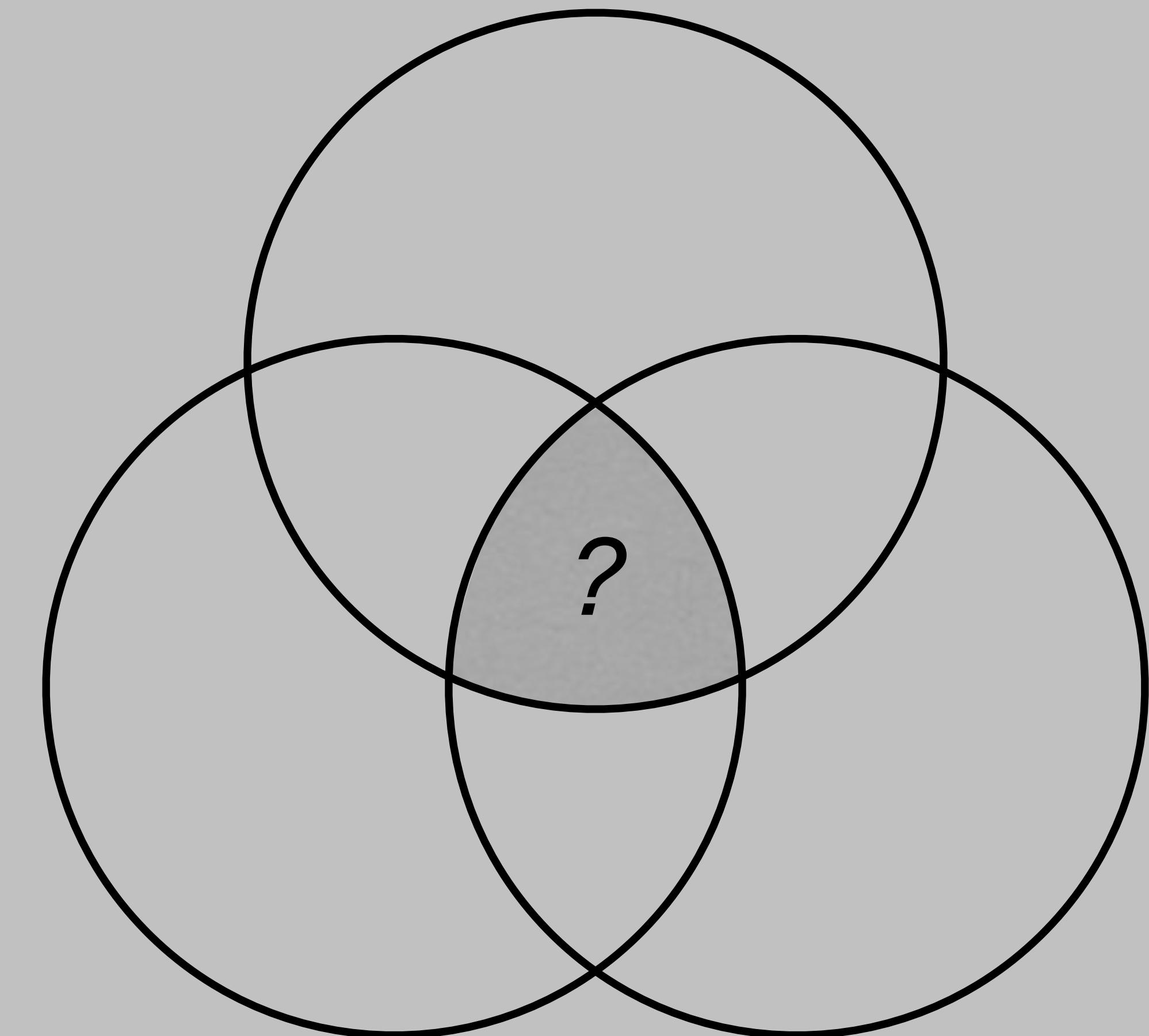
Networking

Sound

Memory Management

Scripting

Threading



Für was ist das Tool gedacht?

*“Unity ist darauf ausgelegt, Games zu machen.”*

*“Unity ist darauf ausgelegt, (alle?/nur?) Games zu machen.”*

Was ist ein Game für *Unity Technologies Inc.*?

# Beispiel-Prioritäten einer Game Engine

- 👉 Rendering- und Berechnungs-Effizienz (Echtzeit)
- 👉 Photorealistisches 3D
- 👉 Einfache Implementation von Aspekten und Dingen aus bestehenden “Games”
- 👉 ...

 Game Engines

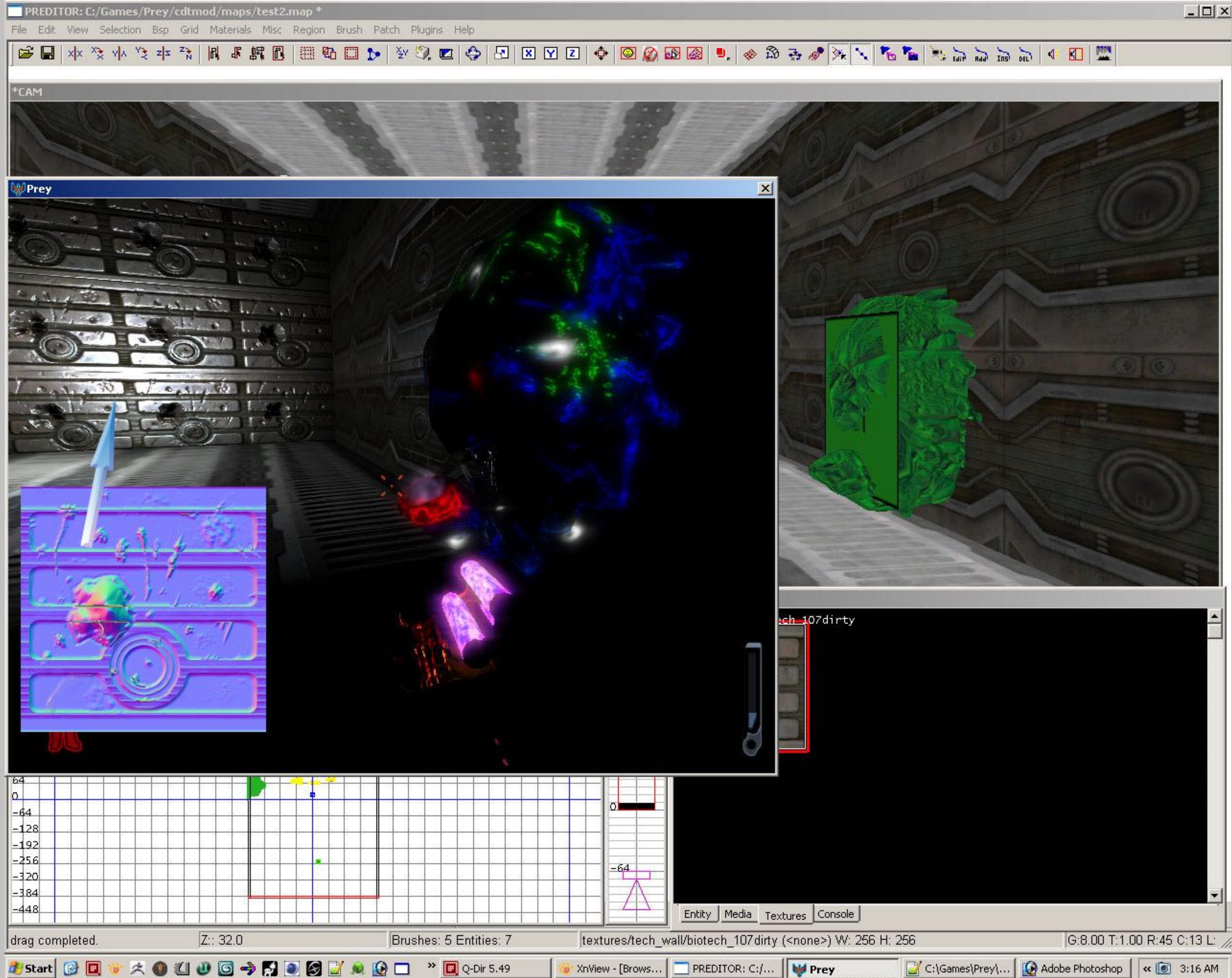
 Wieso Unity?

 Einführung Unity

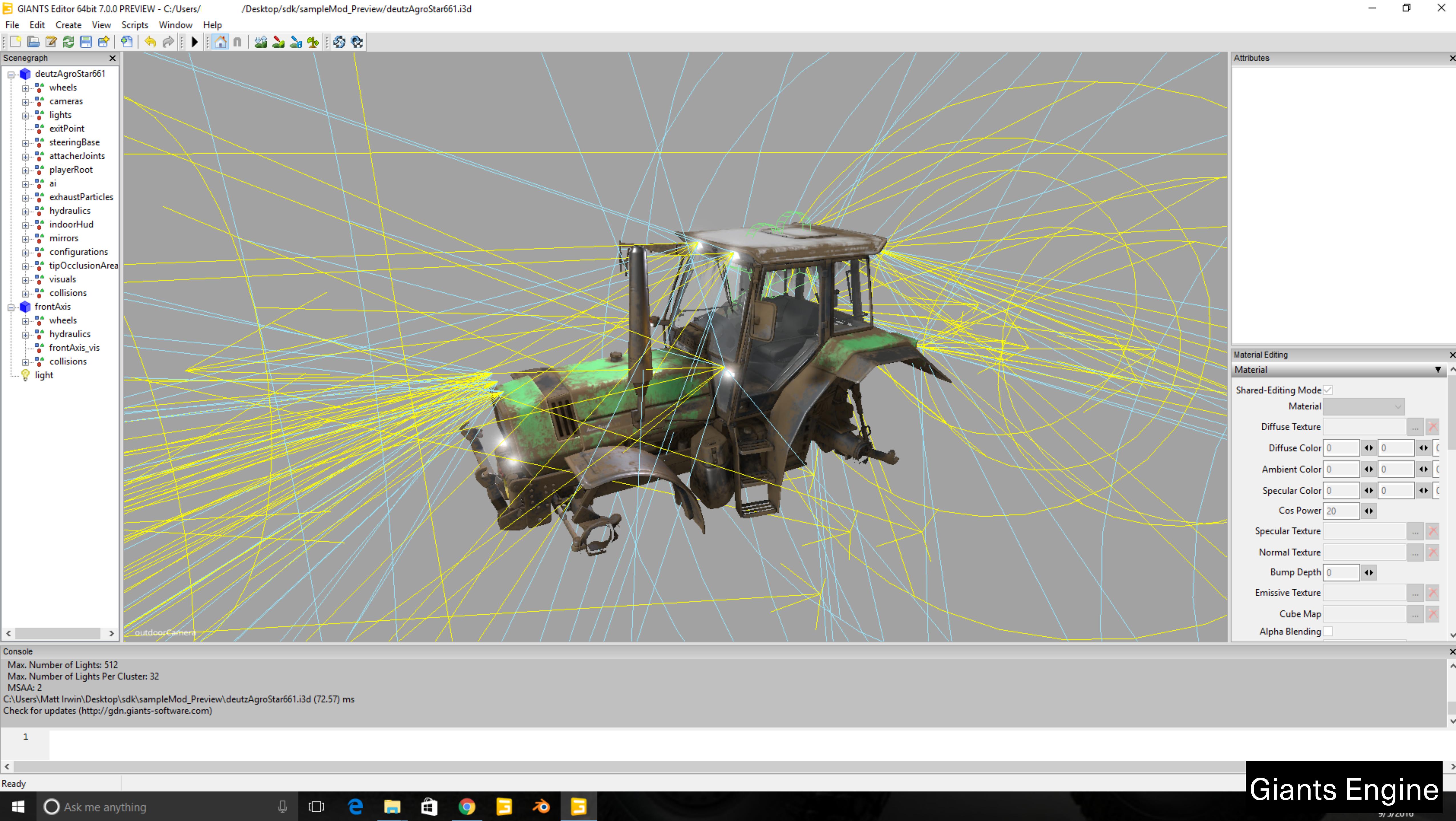
 Tutorial: Walking Sim

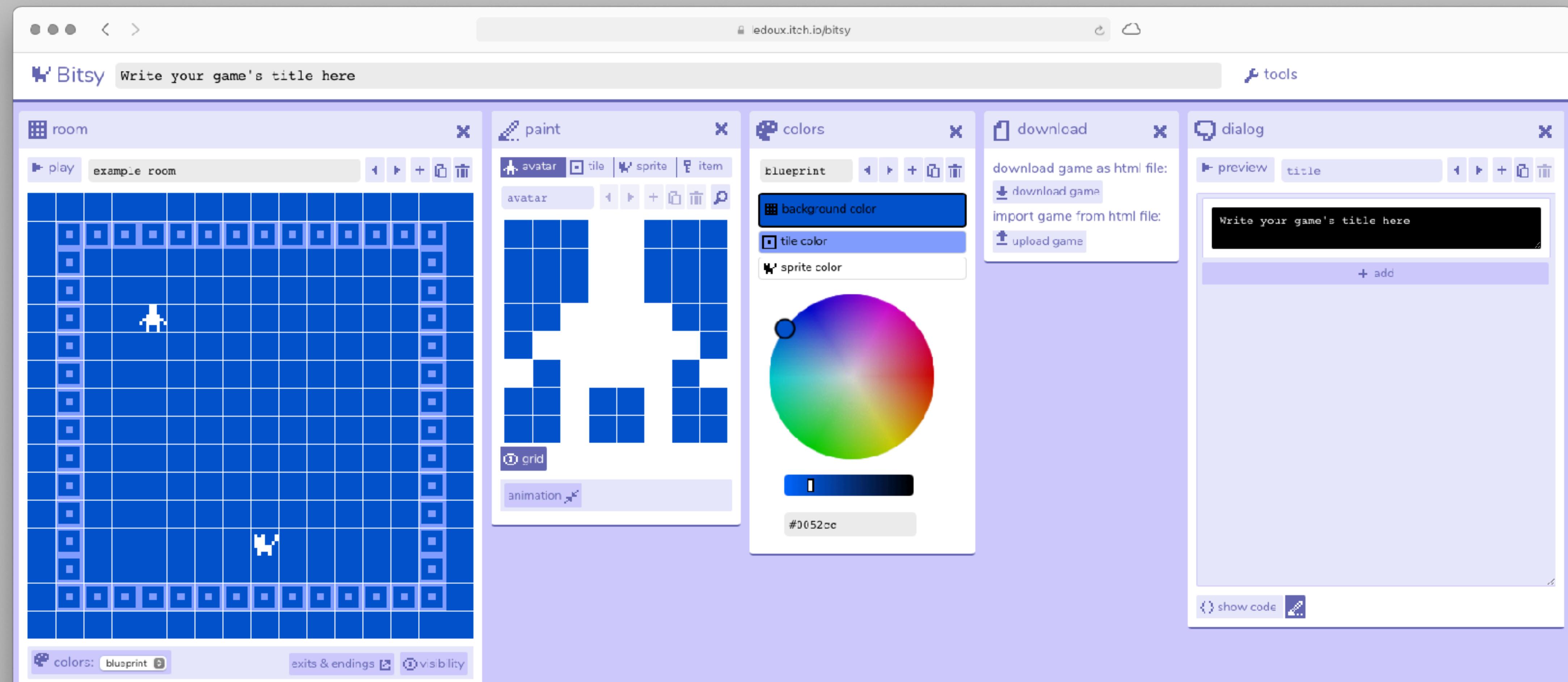


Unreal Engine 4



id Tech 4





# Bitsy

**Unity** (Unity Technologies)

**Unreal Engine** (Epic Games)

**Godot** (Open Source!)

**GameMaker** (YoYoGames)

**Scratch** (MIT Media Lab)

**Phaser**

**Löve**

**HaxeFlixel**

**Bitsy**

**Pico-8**

**Flickgame**

**Ren'Py**

**Twine**

# Prävalenz, Ressourcen & Community

 Game Engines

 Einführung Unity

 Tutorial: Walking Sim



# Navigation

Q      W      E      R      T

Q, W, E, R, T für diese Werkzeuge in Folge:



Sicht Rotieren: Alt + Drag (zentriert) / Rechtsklick (umhersehen)

Sicht Zoomen: Alt + Ctrl + Drag / Mausrad

Sicht Bewegen: Alt + Cmd / Mittlere Maustaste

First-Person Fly Navigation: Rechtsklick + WASD, Shift, Q & E

Ausgewähltes Objekt zentrieren: F

# Hierarchy/Scene Graph

Scene

GameObject

GameObject

GameObject

GameObject

GameObject

GameObject

# Hierarchy/Scene Graph

Scene

  Transform

    Transform

      Transform

      Transform

      Transform

      Transform

# Komponenten eines 3D Objektes

Transform (notwendig für jedes GameObject): Position, Rotation, Scale

Mesh Filter: 3D Geometrie

Mesh Renderer: Wie wird die Geometrie dargestellt (Material, Licht, Schatten, ...)

Weiteres, z.B.: Box Collider, Mesh Collider, Rigidbody, eigenes Script!, ...

 Game Engines

 Einführung Unity

 Tutorial: Walking Sim

# Aufbau

1. Modell(e) importieren
2. First Person Controller
3. *Spiegel*
4. Szenenwechsel
5. Build & Veröffentlichen

# Aufbau

1. Modell(e) importieren

2. First Person Controller

3. *Spiegel*

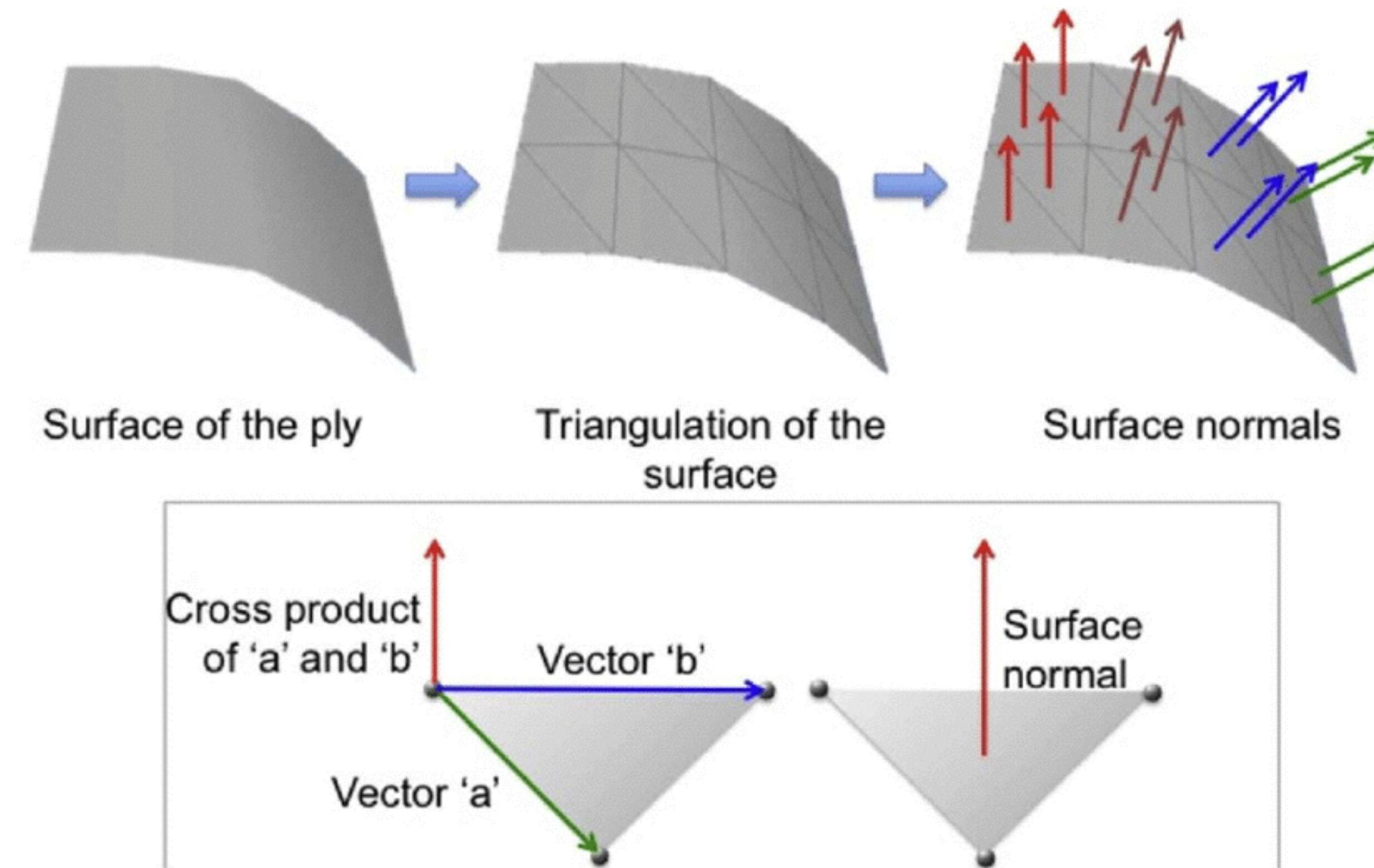
4. Szenenwechsel

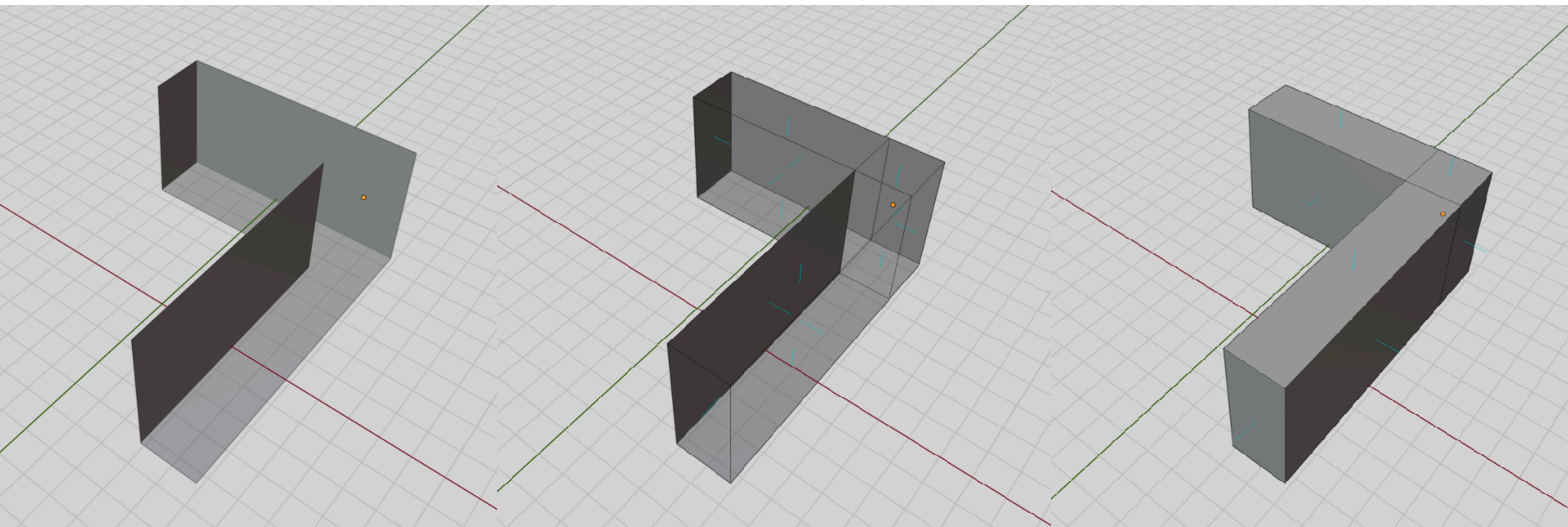
5. Build & Veröffentlichen

# Modelle importieren ➡ Geometrie

## 1. Flächennormalen überprüfen

☞ Unity rendert (normalerweise) nur Vorderseite (positiv)



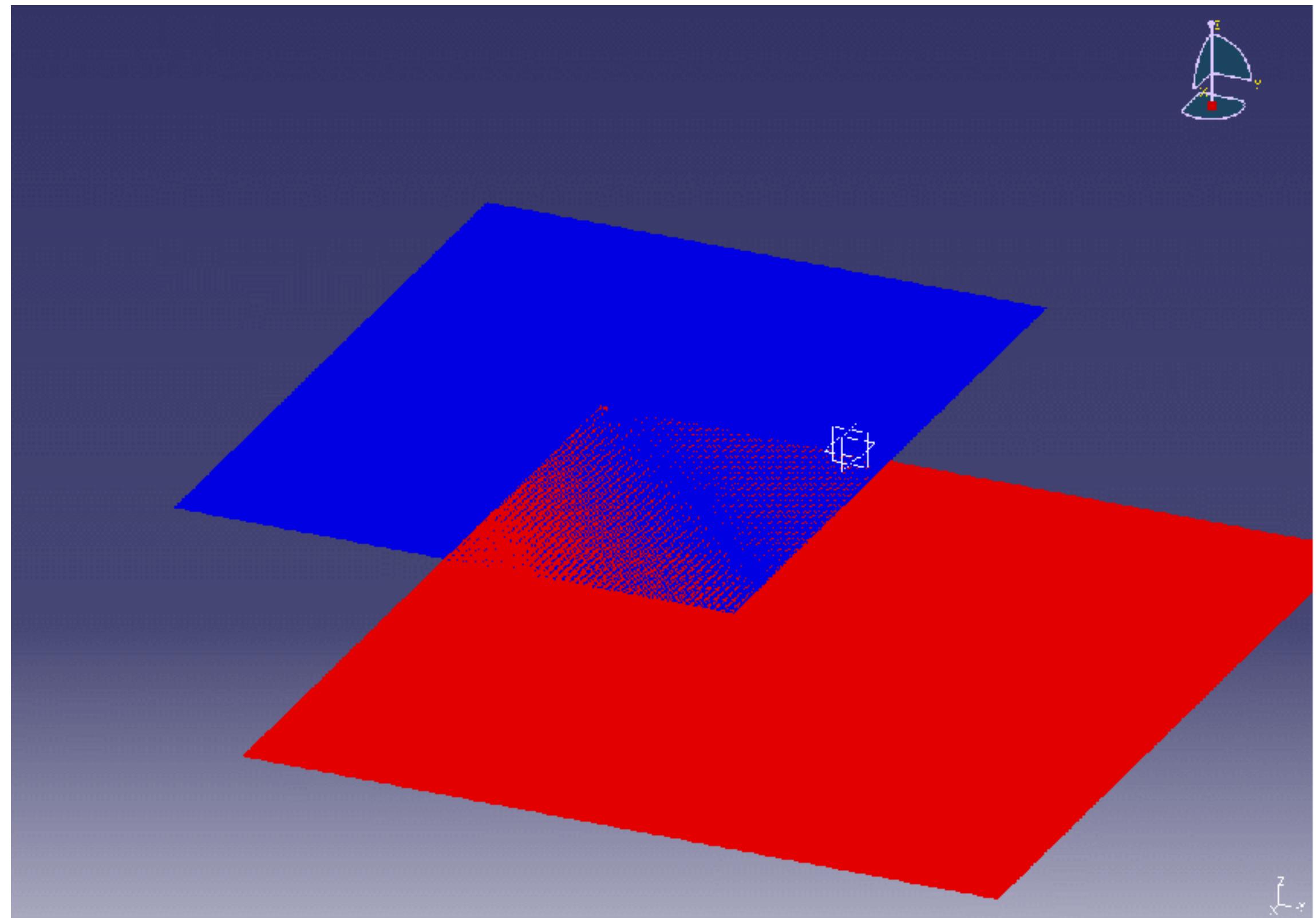


# Modelle importieren ➔ Geometrie

1. Flächennormalen überprüfen

2. Doppelte Geometrie vermeiden

➔ Z-Fighting



# Modelle importieren ➔ Geometrie

1. Flächennormalen überprüfen
2. Doppelte Geometrie vermeiden
3. Polygon-Anzahl möglichst tief halten

# Modelle importieren ➡ Geometrie

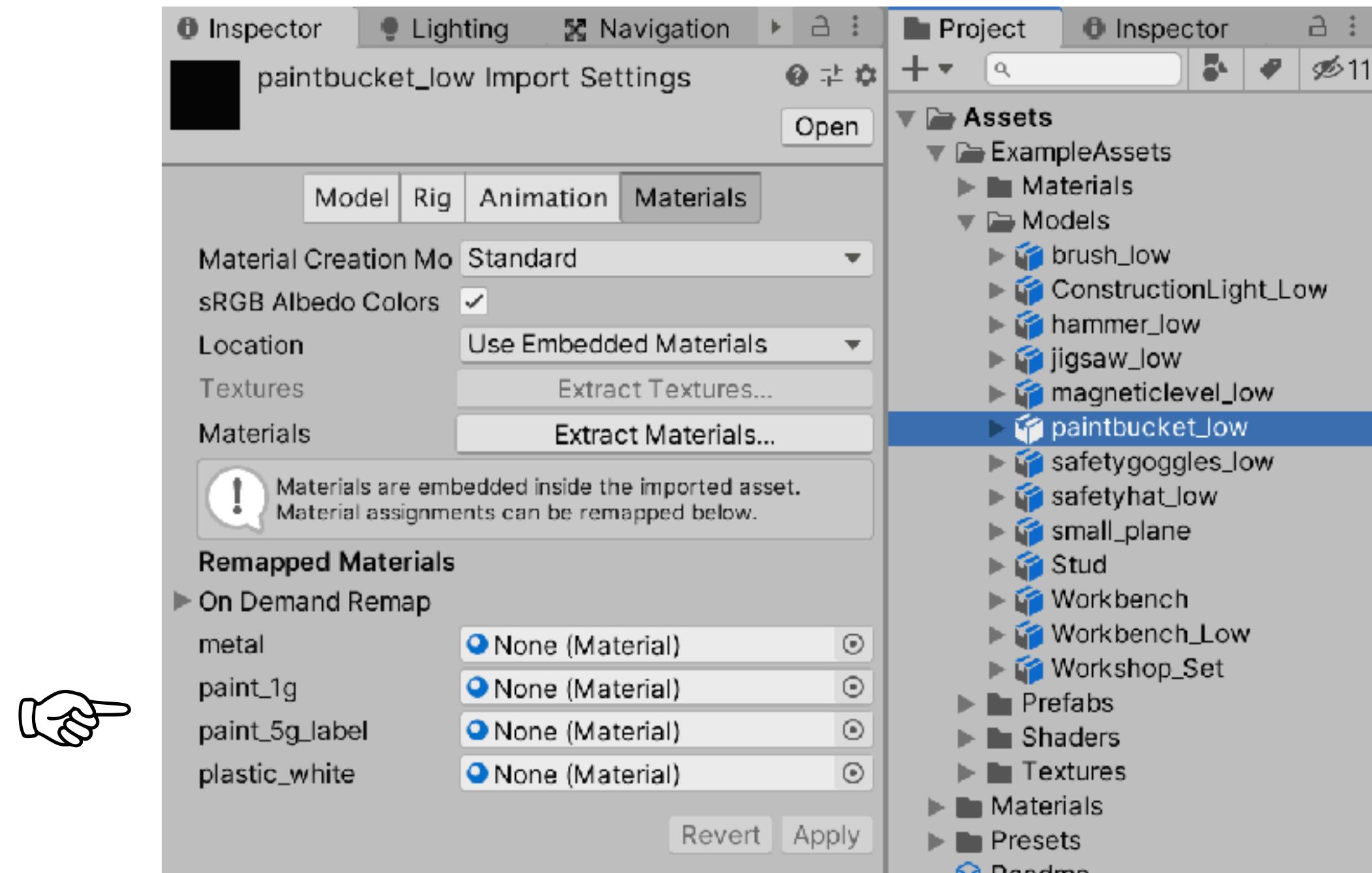
1. Flächennormalen überprüfen
2. Doppelte Geometrie vermeiden
3. Polygon-Anzahl möglichst tief halten
4. .fbx Files sind 

# Modelle importieren ➔ Materialien

1. Materialien für das Modell müssen in Unity ersetzt/zugewiesen werden

# Modelle importieren ➡ Materialien

1. Materialien für das Modell müssen in Unity ersetzt/zugewiesen werden
2. Unity kann Materialien von importierten .fbx erkennen und ersetzen



# Modelle importieren ➡ Materialien

1. Materialien für das Modell müssen in Unity ersetzt/zugewiesen werden
2. Unity kann Materialien von importierten .fbx erkennen und ersetzen
3. Ausprobieren! :)

# Modelle importieren ➔ Licht

1. Jedes Licht in der Szene kostet Performance

➔ Rendert die ganze Szene aus Sicht des Lichts

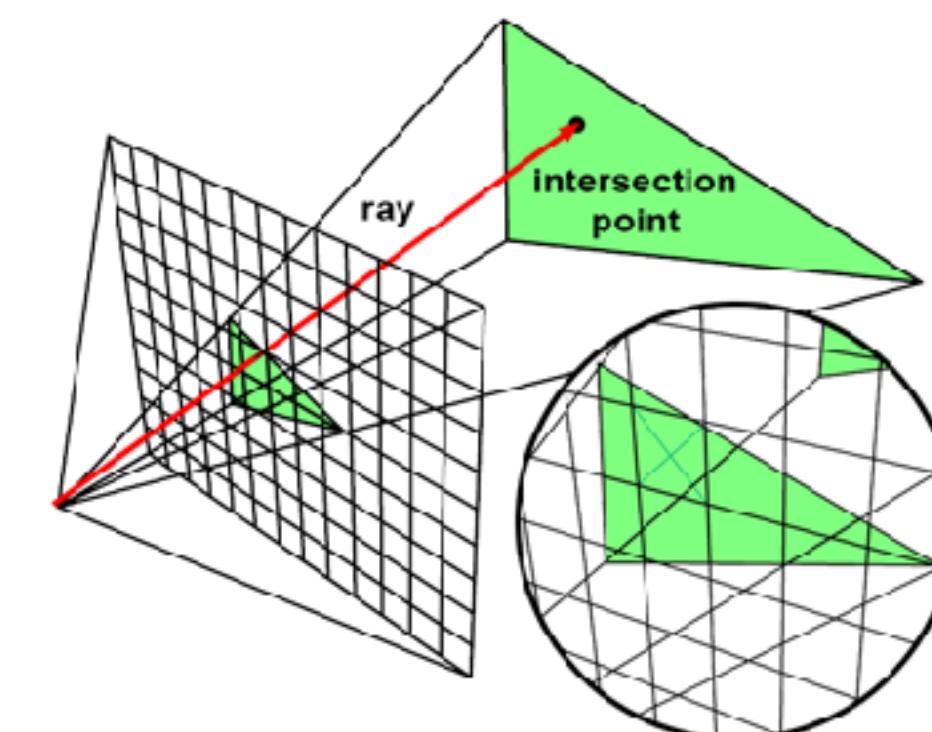
# Modelle importieren ➡ Licht

1. Jedes Licht in der Szene kostet Performance

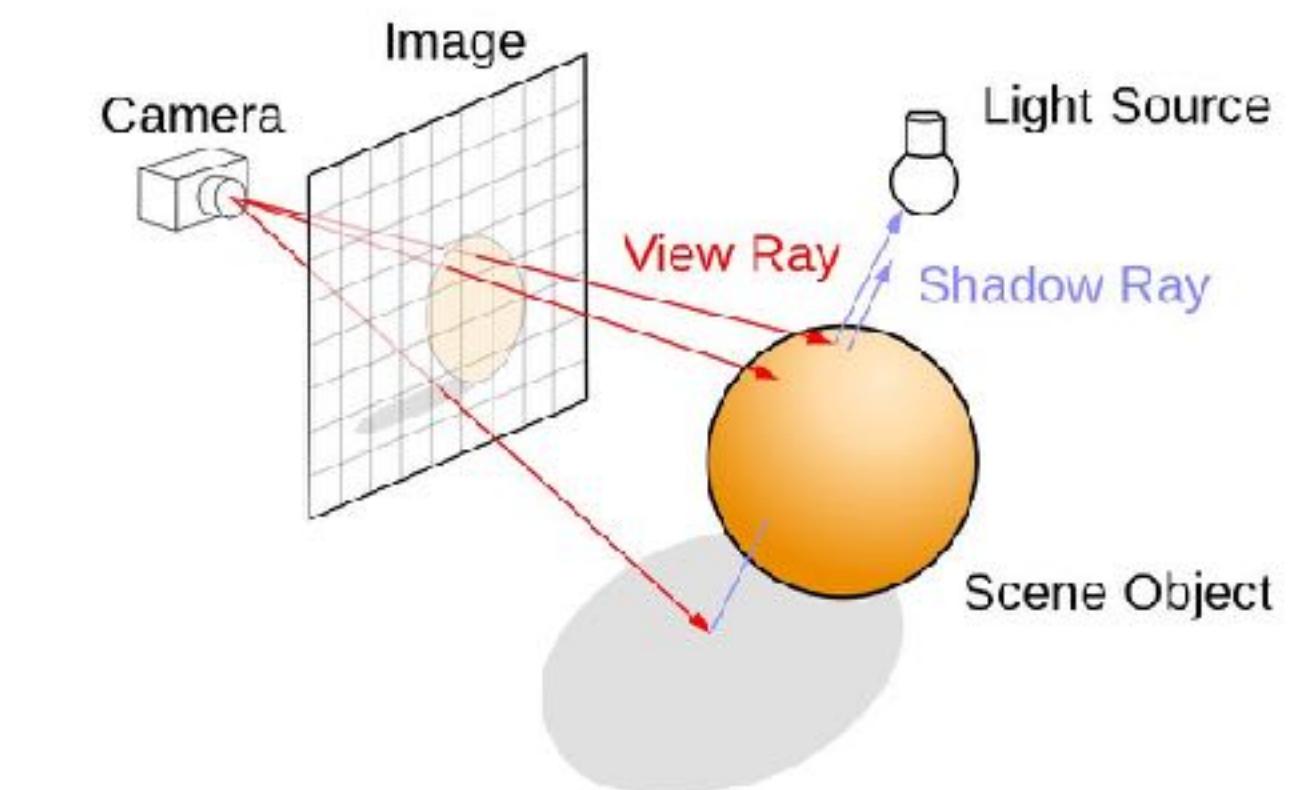
➡ Rendert die ganze Szene aus Sicht des Lichts

2. Licht Settings erforschen (*Window > Rendering > Light Settings*)

➡ Light baking



“Rasterisierung”



“Ray Tracing”

# First Person Controller

1. *FirstPersonDrifter.unitypackage* herunterladen und importieren  
👉 <https://github.com/stekra/Unity-Intro-Packages>
2. Prefab “*Player*” in die Szene ziehen
3. Allfällige andere Kameras in der Szene löschen

# Aufbau

1. Modell(e) importieren
2. First Person Controller
3. *Spiegel*
4. Szenenwechsel
5. Build & Veröffentlichen

# Spiegel ↗ In Videospielen



*Resident Evil: Revelations 2 (2015)*

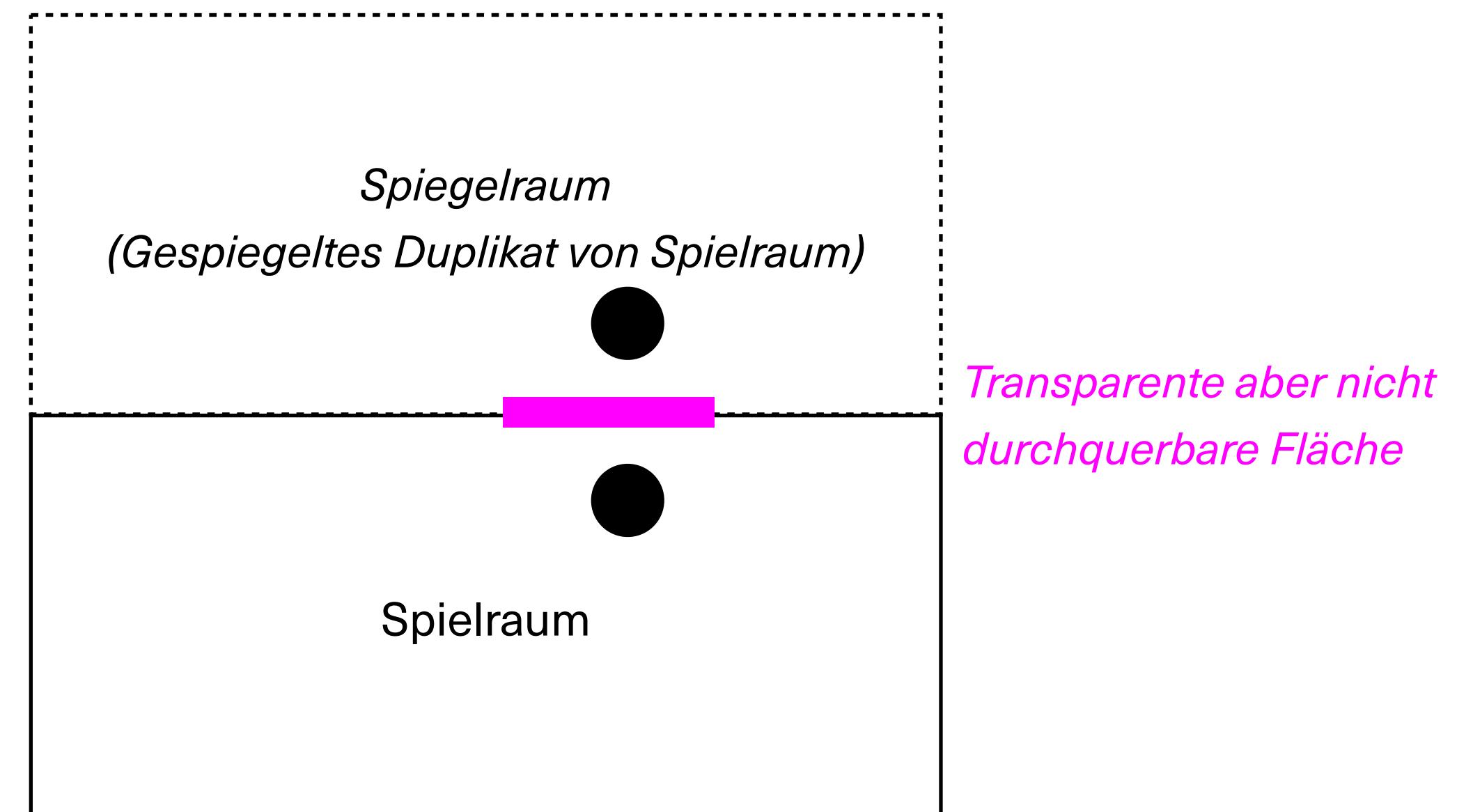
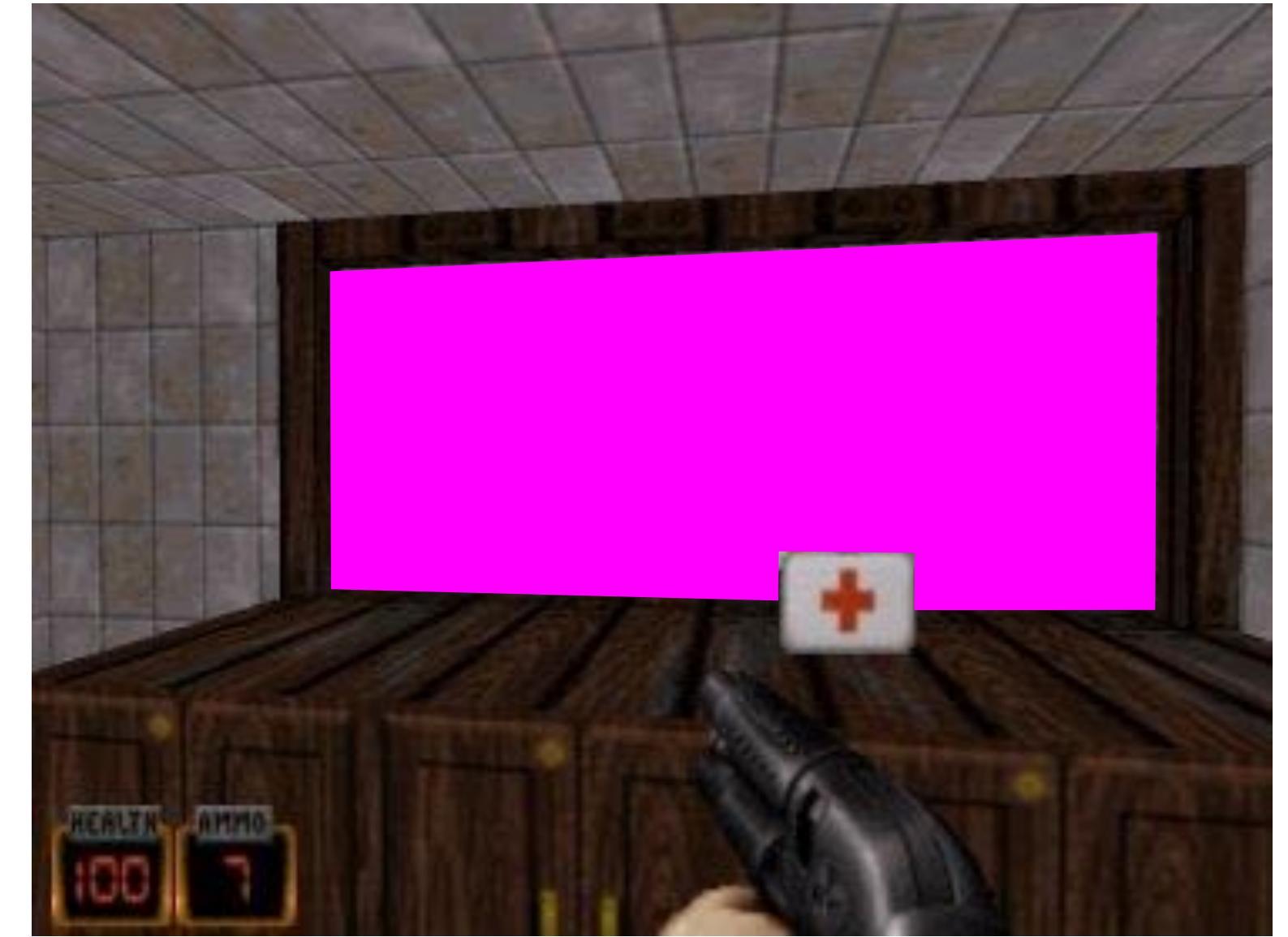


*Doom (2016)*

# Spiegel ↗ In Videospielen



Duke Nukem 3D (1996)



# Spiegel ➔ Planare Spiegelungen

1. *Mirrors.unitypackage* herunterladen und importieren

➔ <https://github.com/stekra/Unity-Intro-Packages>

2. Object Mirror

Selektiv zu spiegelnde Objekte wählen, Spiegelachsen wählen, Objekte werden dupliziert und synchronisiert bewegt

3. View Mirror

Spiegelfläche spiegelt gesamte Sicht in beliebigem Winkel  
(nur ein View Mirror pro Szene möglich)

# Spiegel ➔ Sphärische Spiegelungen

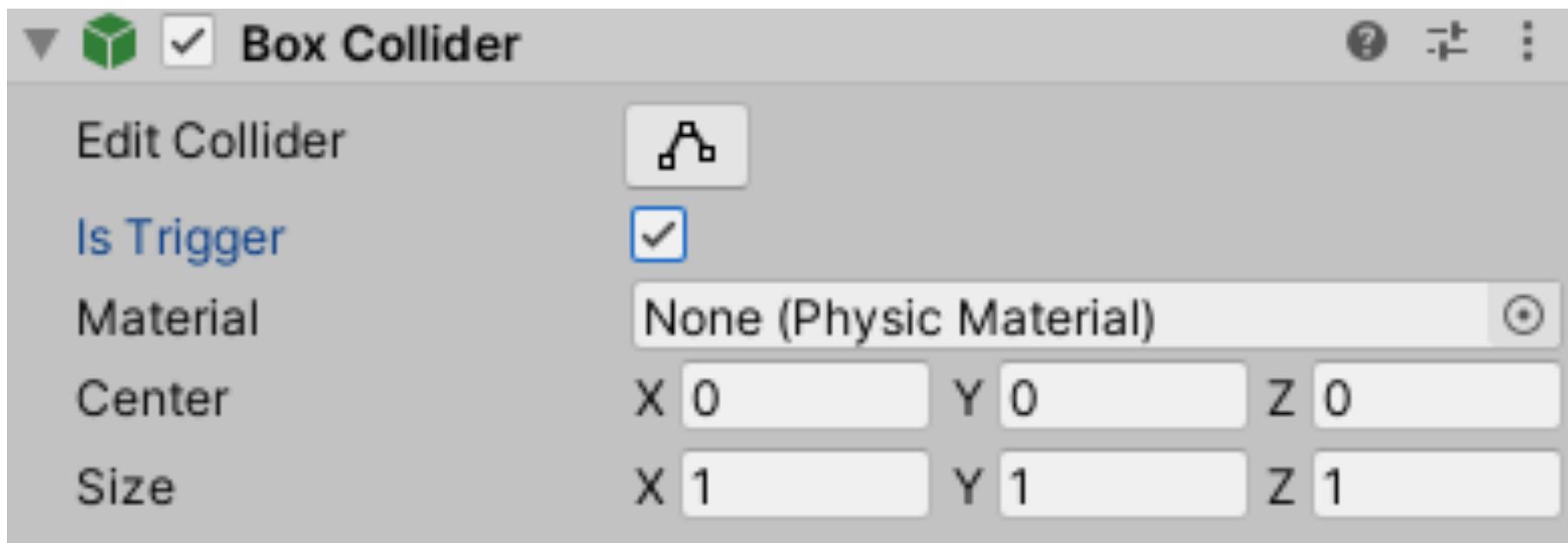
1. Einem neuen GameObject die Komponente *Reflection Probe* geben
2. In der Komponente *Type* auf *Realtime*, *Refresh Mode* auf *Every Frame* und *Time Slicing* auf *No Time Slicing* stellen
3. Spiegelndes Material kreieren und einem (möglichst kugelförmigen) Objekt zuweisen
4. Reflection Probe Objekt möglichst beim Kugelzentrum platzieren

# Aufbau

1. Modell(e) importieren
2. First Person Controller
3. *Spiegel*
4. Szenenwechsel
5. Build & Veröffentlichen

# Szenenwechsel

1. Neues GameObject mit Collider (z.B. Box Collider) erstellen
2. In der Collider Komponente “Is Trigger” aktivieren



# Szenenwechsel

1. Neues GameObject mit Collider (z.B. Box Collider) erstellen

2. "Is Trigger" aktivieren

3. Neues Skript anhängen

```
1  using UnityEngine;
2  using UnityEngine.SceneManagement;
3
4  public class ChangeSceneOnTrigger : MonoBehaviour
5  {
6      public string SceneToChangeTo;
7
8      private void OnTriggerEnter(Collider other)
9      {
10         if (other.CompareTag("Player"))
11         {
12             SceneManager.LoadScene(SceneToChangeTo);
13         }
14     }
15 }
16 }
```

# Szenenwechsel

1. Neues GameObject mit Collider (z.B. Box Collider) erstellen
2. “Is Trigger” aktivieren
3. Neues Skript anhängen
4. Alle gewünschten Szenen in den Build Settings (*File > Build Settings...*) hinzufügen

# Aufbau

1. Modell(e) importieren
2. First Person Controller
3. *Spiegel*
4. Szenenwechsel
5. Build & Veröffentlichen

# Build

1. Im Unity-Hub unter *Installs* für gewünschte version “Add Modules” auswählen und gewünschte Module hinzufügen
2. In den Build Settings die Platform wählen
3. In den Build Settings und Player Settings allenfalls Anpassungen machen (z.B. Icon hinzufügen)
4. Teilen!

# WebGL

Einfach zu teilen

👉 Spieler muss nichts herunterladen, wenig Kompatibilitätsprobleme

Auf Performance achten!

Veröffentlichen, z.B. auf *itch.io*

# macOS Catalina 10.15 und neuer

Optionen:

Rechtsklick > Öffnen

Apple Certified Developer werden und Applikation “notarisieren” lassen.

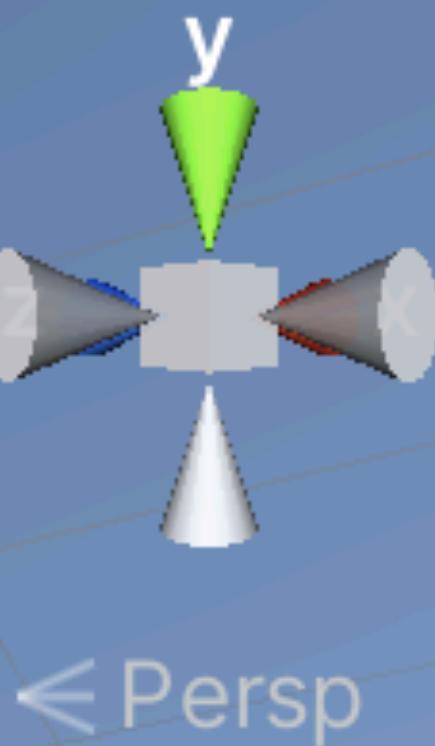
# iOS

Builden exportiert ein .xcodeproj

In XCode lässt es sich (relativ) einfach auf dem eigenen Gerät testen

Zum Veröffentlichen muss es auf dem iOS App Store publiziert werden.

Dazu muss man ebenfalls im Apple Developer Program sein und die App muss den App Store Guidelines konform sein.



Slides und Scripts erhältlich unter <https://github.com/stekra/Unity-Intro-Packages>  
Unity Fragerunde @ Montag, 26.5.2021  
Fragen jederzeit an [stefan.kraft1@zhdk.ch](mailto:stefan.kraft1@zhdk.ch) (Antwortzeit nach Kapazität)