



Einstieg in Unity Teil 1 – 3D und erste Szene

Description

Dies ist der Beginn einer dreiteiligen Serie, um in die Unity-Engine einzusteigen. Dabei werden wir vor allem die Grundlogik der Spieleentwicklung in Unity kennenlernen. Es geht vor allem darum, schnell Erfolgserlebnisse zu sammeln, ohne langatmig alles zu erklären. Außerdem werden wir uns vor allem mit Würfeln beschäftigen, nicht mit einem Spiel oder Spiellogik.

Kein Spiel? Was soll der Quatsch?!

Das ist eine sehr gute Frage. Spiele sind komplex, selbst die vermeintlich einfachen. In einem Einstiegskurs wie diesen sollte es aber vor allem um zwei Dinge gehen:

1. Kennenlernen der wesentlichen Elemente von Unity.
2. Kleine Erfolgserlebnisse, um sich weiter zu motivieren.

Wer etwas mehr über Unity und die Hintergründe dieser Serie erfahren will, [kann gerne den Einführungsartikel lesen](#). Für alle anderen gilt: Reden wir nicht lange herum und starten!

Download und Installation

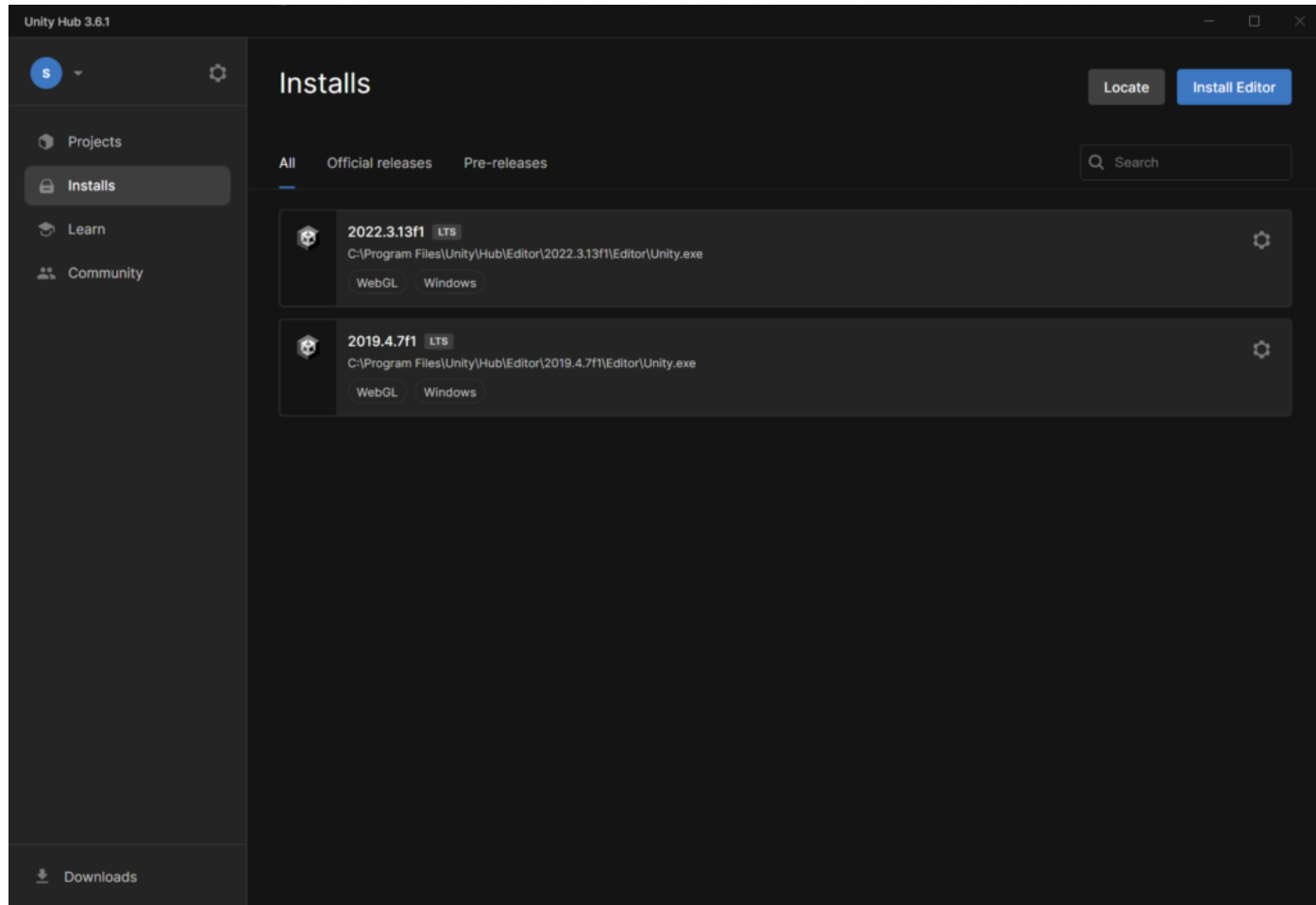
Ich gehe davon aus, dass du bereits Programme unter deinem Betriebssystem installiert hast. Deshalb halte ich diesen Abschnitt kurz.

- Gehe [auf diese Seite](#) und lade die Version für dein Betriebssystem herunter.
- Die Datei sollte relativ klein sein. Der Rest wird später heruntergeladen, wenn er gebraucht wird.
- Unter Windows muss man Adminrechte haben, um es installieren zu können.
- Ggf. wird Visual Studio Code und .NET-Kram zusätzlich installiert. Das ist wichtig!

Anschließend hast du Icons auf Deinem Desktop und kannst mit dem Start der *Unity-Hub* loslegen.

Projekt starten

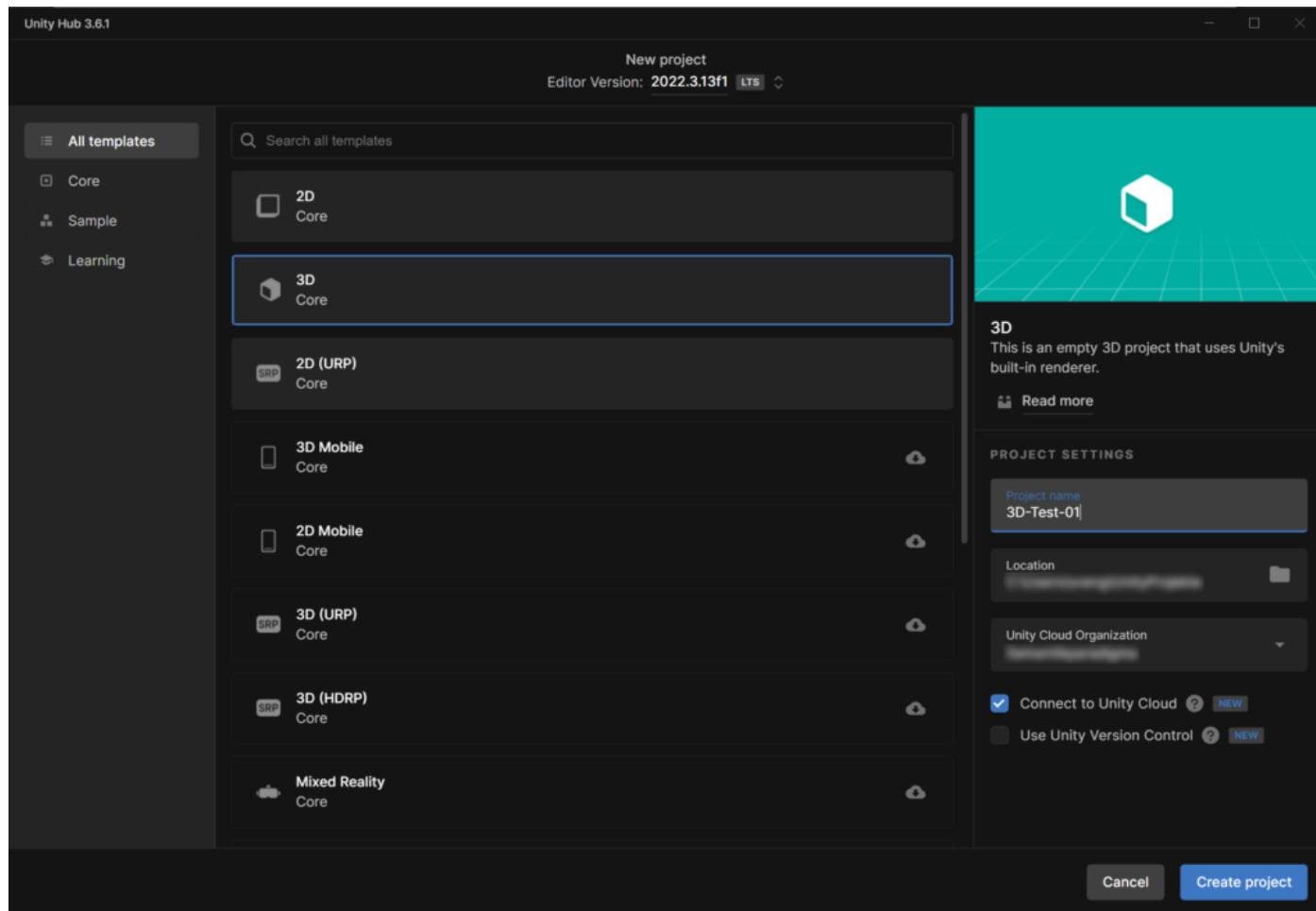
Wenn du den Unity-Hub startest, hast du links mehrere Reiter. Du siehst deine Projekte, installierte Versionen von Unity etc. Je nach Version können ab 2024 unterschiedliche Lizenzbedingungen vorliegen. Ich arbeite derzeit mit der 2022.3.13f1.



Unity Hub

Du kannst gerne eine neuere Version nehmen, es sollte keinen Unterschied machen.

Wir klicken links auf *Projects* und dann auf *New project*. Nun bekommen wir zahlreiche Templates vorgeschlagen, was recht verwirrend sein kann. Wir wählen *3D Core*. Damit erhalten wir die Basis für 3D-Spiele, ohne unnötigen Schnickschnack.



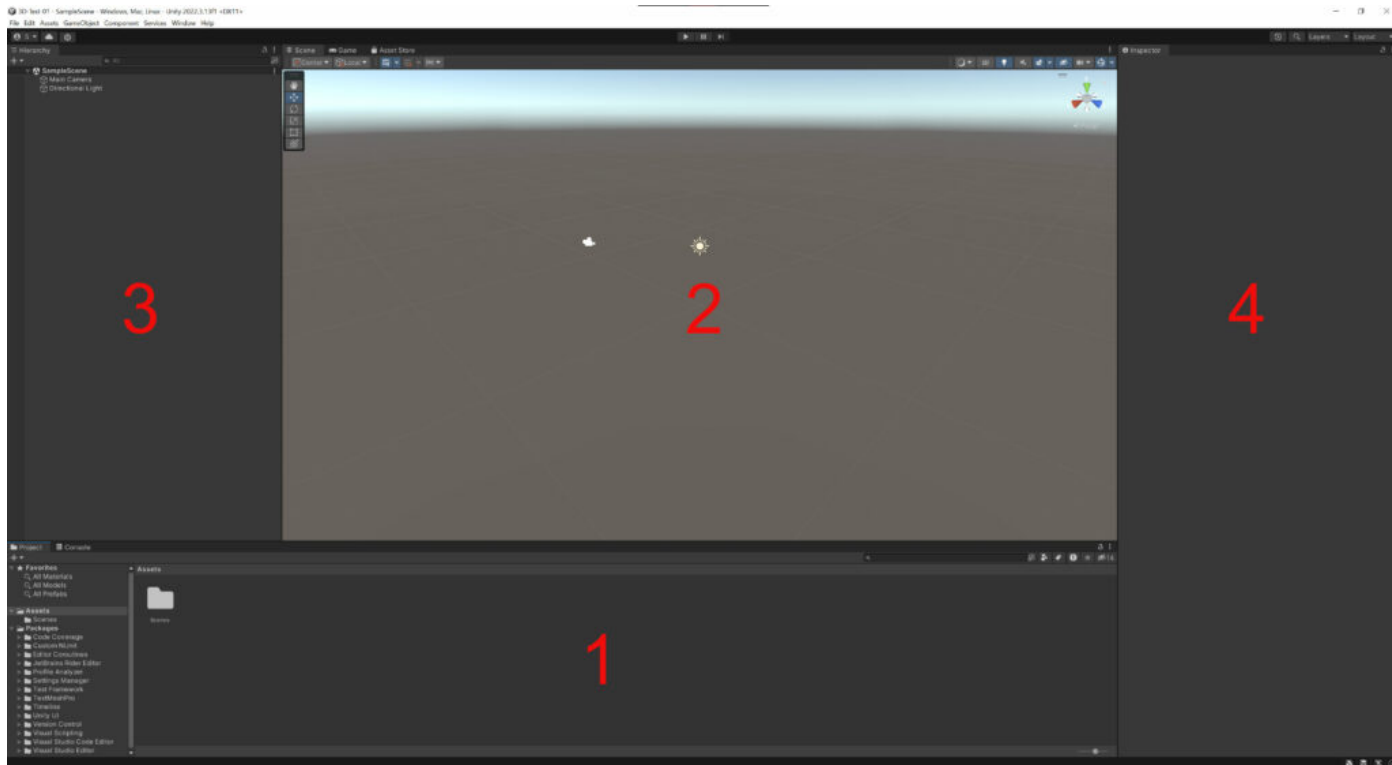
Unity Projekt starten

Wenn wir alles eingestellt haben, klicken wir auf *Create project* und können gleich loslegen.

Wir schauen uns um

Das mit „gleich loslegen“ ist so eine Sache. Wenn man ein neues Projekt startet, kann es durchaus eine Minute oder länger dauern, bis alle Daten kopiert und geladen wurden. Wird das Projekt später neu geladen, geht es ein bisschen schneller.

Wie auch immer: Irgendwann ist es geladen und wir starren auf ein graues Fenster, welches in mehrere Fenster unterteilt wurde.



Unity nach dem Start eines neuen 3D Core-Projekts

Diese Fenster sind vom Start weg viergeteilt. Ich habe sie so nummeriert, wie sie i. d. R. verwendet werden.

1. Hier haben wir unsere **Ressourcen** für das Projekt. Texturen, Grafiken wie Sprites, 3D-Objekte, Materialien, Sounds, aber auch Code. Wir können hier unsere eigenen Ordner erstellen und alles so organisieren, wie uns der Sinn danach steht.
2. Die meisten Ressourcen ziehen wir irgendwann in die **3D-Welt**. Wenn wir starten, haben wir einen Himmel, eine Kamera und ein Licht. Oberhalb des Fensters gibt es drei Reiter. Die zwei wichtigen sind *Scene* und *Game*.
3. Links von der 3D-Welt haben wir unsere **Hierarchie**. Hier werden wir alles aufgelistet sehen, was sich in unserer 3D-Welt befindet. Die 3D-Welt selbst wird meistens ein Level sein. Nach dem Start steht da *SampleScene* und darin befinden sich die bereits erwähnte Kamera und das Licht.
4. Rechts sehen wir den **Inspector**. Darin können wir die Eigenschaften unserer Ressourcen und den Dingen in der Welt bestimmen, die wir einsetzen. Etwa die Position der Kamera, die Größe von Objekten (Skalierung) und vieles mehr. Per Code können wir sogar neue Eigenschaften hinzufügen, die wir dann nicht mehr im Code manipulieren müssen, sondern bequem im Fenster verändern können.

Bewegen in der 3D-Welt

Um uns in der 3D-Szene zu bewegen, nutzen wir die Maustasten. Die linke Taste dient dazu, Objekte auszuwählen und zu manipulieren. Mit der rechten Maustaste können wir uns drehen. Halten wir die mittlere Maustaste gedrückt, können wir hoch und runter fliegen. Mit drehen des Mausekaders kommen wir vor und zurück.

Wenn wir die Kamera anklicken, sehen wir gleich drei Dinge.

1. Wir sehen ein weiteres kleines Fenster rechts unten, die uns die Kameraansicht zeigt. Das ist sehr hilfreich.
2. Um das Objekt haben sich drei Pfeile mit unterschiedlichen Farben gelegt.
3. Der Inspektor zeigt alle Eigenschaften der Kamera an.

Mit den drei Pfeilen können wir die Position der Kamera verändern. Das gilt für jedes Objekt in der 3D-Welt. Grün ist **Y**, Rot ist **X** und Blau ist **Z**. Im Inspektor kannst du auch die Werte eingeben.

Wir können nicht nur die Position verändern, sondern auch die Rotation oder Skalierung. In der Szene gibt es dafür links oben entsprechende Icons, um umzuschalten.

Üblicher Workflow

Beim Workflow unterscheide ich zunächst zwischen Ressourcen und Code.

Ressourcen einbauen

Die Ressourcen wie Grafiken, 3D-Modelle etc. hast du entweder selbst erstellt oder irgendwo besorgt. Diese kommen unten in dein Projektfenster (1). Grundsätzlich muss man zwischen *Texturen* und *Materialien* unterscheiden. Die Texturen kommen von dir, das Material musst du erstellen. Wie das geht, zeige ich gleich.

Der Unterschied ist folgender: Die Textur ist das eigentliche Bild auf der Oberfläche. Ein Material kann aus mehreren Ebenen von Texturinformationen bestehen, die zusammen sozusagen die physikalischen Eigenschaften einer Oberfläche bestimmen. Das bekannteste ist das *Normal Mapping*. Dadurch wirkt das Material plastischer.

Wenn du die Ressource hast, kannst du sie in deiner Szene, also bspw. deinem Level, platzieren und die Eigenschaften im *Inspektor* definieren.

Das Zuweisen eines Materials auf ein Objekt erfolgt normalerweise sogar vorher. In der 3D-Szene kannst du später überprüfen, ob alles funktioniert hat.

Arbeiten mit Code

Normalerweise erfolgt der Code durch die Erstellung von *C#-Skripten*. Das werden wir im zweiten und dritten Teil dieser Serie tun.

Man erstellt also ein Skript, Unity füllt es dann mit den wichtigsten Informationen. Wenn man einmal auf das Skript klickt, bekommt man eine Vorschau im Inspektor. Mit Doppelklick sollte sich automatisch *Visual Studio* oder *Visual Studio Code* öffnen, je nachdem, was du eingestellt hast. Der Ablauf ist dann wie folgt:

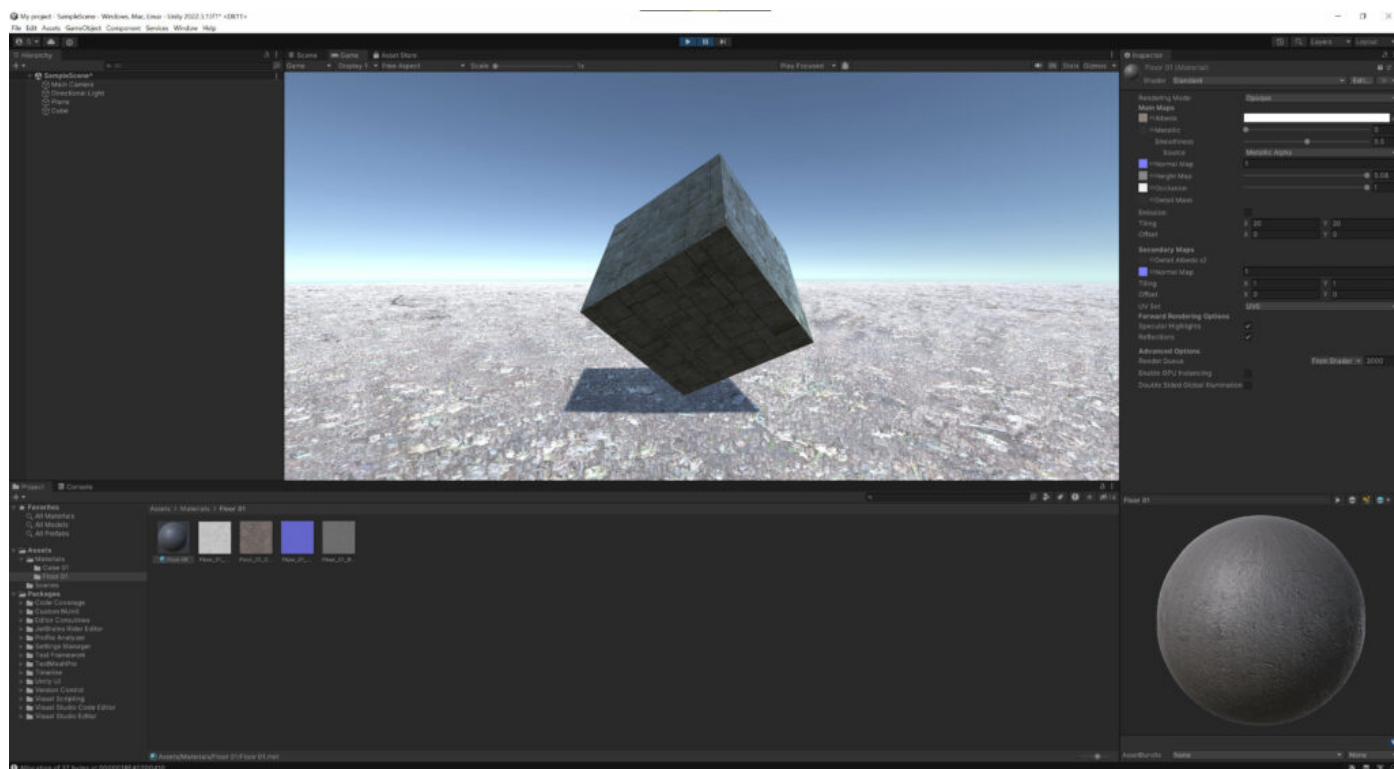
1. Du schreibst den Code.

2. Du speicherst ihn ab und wechselst zu Unity.
3. Unity prüft den Code auf Fehler, etwa die Syntax.
4. Wenn es Fehler gibt, werden diese in der Konsole angezeigt.
5. Damit das Spiel gestartet werden kann, musst du die Fehler zunächst beheben.
6. Damit der Code ausgeführt wird, musst du ihn einem Objekt bzw. einer Ressource zuweisen.

Im Detail gehen wir darauf im zweiten Teil ein.

Boden und Würfel

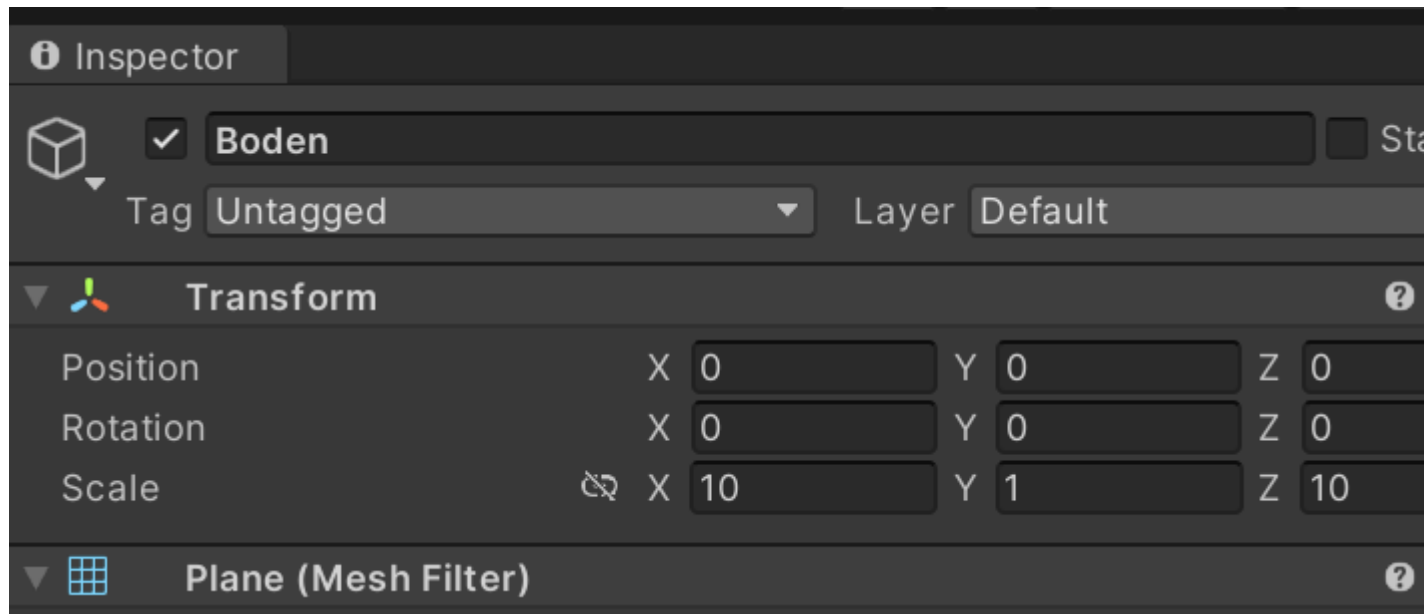
Das ist unser heutiges Ziel:



Unser Ziel: Boden, Würfel und Texturen

Wir können in Unity nicht nur fertige 3D-Objekte laden, sondern auch einfache Geometrien erzeugen.

Den Boden erstellen wir, indem wir ganz oben im Menü auf *GameObject ? 3D Object ? Plane* klicken. In der Hierarchy ist „Plane“ ausgewählt, wir können es gleich in *Boden* oder *Floor* umbenennen. Anschließend können wir es markieren, die Position auf 0, 0, 0, setzen und die Skalierung auf 10, 1, 10.



Inspector-Einstellungen für den Boden

Auf dieselbe Weise erstellen wir den Würfel. *GameObject ? 3D Object ? Cube*. Den Würfel positionierst du nun so, dass er vor der Kamera sichtbar ist. Die Kamera ist bei mir auf 0, 4, -18, der Würfel bei 0, 6, -4. Er ist etwas weiter hinten und recht weit oben, was vor allem im dritten Tutorial Sinn ergeben wird. Du kannst natürlich die Positionen frei wählen.

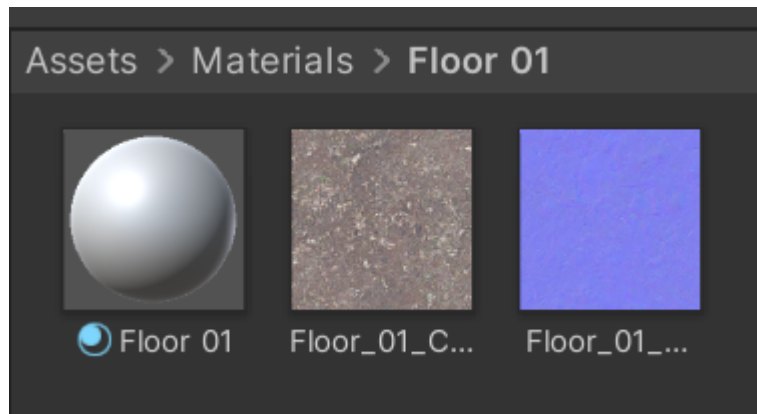
Materialien

Für Boden und Würfel brauchst du je eine Textur. Normalerweise sind das JPG oder PNG-Bilder. Für den Test ist die Auflösung egal, ich habe je 1024x1024 Pixel genommen. Es ist für solche Anwendungen vorteilhaft, wenn die Texturen kachelbar sind, man also keinen Anfang und kein Ende sieht, wenn man sie kachelt. Das wirst du gleich beim Boden bemerken.

Im Beispiel habe ich zwei Ebenen verwendet, du kannst aber gerne darauf verzichten. Die Textur reicht.

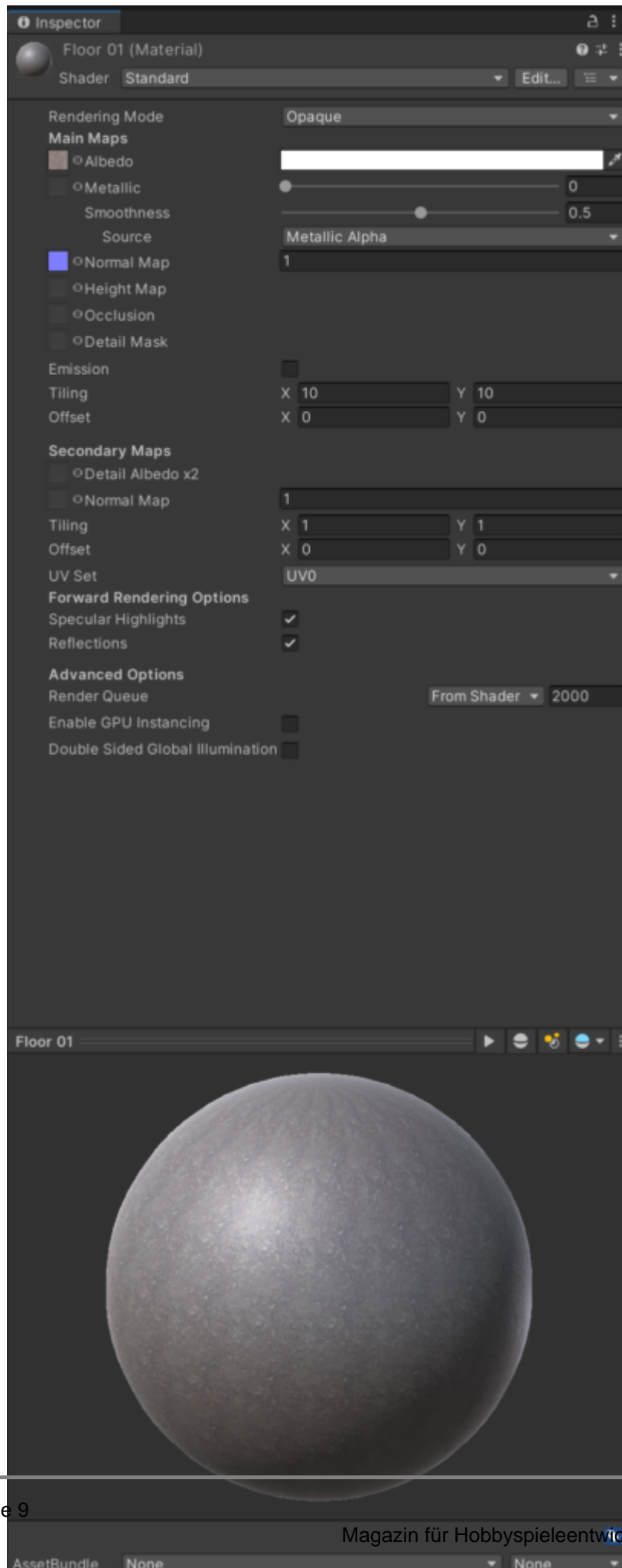
Am besten erstellst du unter **Assets** einen Ordner mit dem Namen *Materials*. Rechtsklick, *Create ? Folder*.

Darin zwei weitere Ordner mit den Namen *Floor 01* und *Cube 01*. Gehe in den Ordner *Floor 01* und lege hier deine Textur ab. Dann wieder Rechtsklick, *Create ? Material*. Nenne es *Floor 01*.



Floor 01 Material

In meinem Beispiel habe ich eine Textur und die Normal Map. Jetzt weisen wir beides dem Material zu. Wenn du auf das Material klickst, siehst du, dass noch nicht viel los ist. Klicke auf die Textur und halte dabei die Maustaste gedrückt. Jetzt ziehe es zum **Inspektor** auf das kleine Rechteck, wo *Albedo* steht. Wenn Du eine Normal Map hast, wiederholst du den Vorgang mit der Normal Map und legst sie auf das gleichnamige Rechteck. Höchstwahrscheinlich wird dich Unity darauf hinweisen, dass es noch keine Normal Map ist. Klicke daraufhin auf *Fix Now*. Da wir den Boden hochskaliert haben, stellen wir beim *Tiling* noch 10, 10 ein.



Floor 01 Material Inspektor

Nun klicken wir wieder auf das Material bei den Ressourcen, halten es gedrückt, ziehen es auf den Boden in der 3D-Szene und lassen los. Fertig! Du hast den Boden texturiert.

Jetzt kannst Du den Vorgang für den Würfel wiederholen, wobei das Tiling entfällt. Die Textur sollte sich auf alle sechs Seiten des Würfels legen.

Nächster Teil

Im zweiten Teil geht es an den Code. Wir werden den Würfel in alle drei Richtungen um die eigene Achse drehen lassen. Dabei werden wir nicht nur C#, sondern auch weitere Funktionen von Unity kennenlernen.

Weiterführende Links

[Einstieg in Unity Teil 2 – Der drehende Würfel](#)

[Einstieg in Unity Teil 3 – Die Macht der Programmierung](#)

[Warum soll ich programmieren lernen?](#)

[Arcade-Spiele – die Mutter aller Genres?](#)

[Frustspirale – Designschnitzer in Spieleklassikern](#)

[Mehr als nur eine Party](#)

Externe Links

[Unity Dokumentation](#)

Date Created

29. November 2023

Author

sven