

Hans-Georg Schumann



3D-Spiele programmieren

mit

Unity



Ganz einfach ohne Vorkenntnisse



Inhaltsverzeichnis

	Einleitung	9
E.1	Spielentwicklung	9
E.2	Und was ist Unity?	9
E.3	Voraussetzungen	10
1	Das erste Projekt	11
1.1	Unity starten	11
1.2	Ein Objekt zum Spielen	16
1.3	Gravitation und Kollision	26
1.4	2D oder 3D?	32
1.5	Ausblick	39
2	Script-Programmierung	43
2.1	Ein Script erstellen	43
2.2	Klassen und Methoden	49
2.3	if-Strukturen	53
2.4	Schubsen oder schieben?	57
2.5	Mal schwerelos, mal »bouncy«	60
2.6	Import und Export	63
2.7	Ausblick	69
3	Spielfigur als Sprite	71
3.1	Ein neues Spielobjekt	71
3.2	Bilder fürs Sprite	73
3.3	Ein Script für die Figur	76
3.4	Character Controller	79
3.5	Material und Textur	85
3.6	Ausblick	90
4	Jump & Run	91
4.1	Steuersystem	91
4.2	Das richtige Bild	99
4.3	Eigene Methoden	104
4.4	Laufen, Springen, Schubsen	107
4.5	Bouncy Ball	110
4.6	Trigger	112

4.7	Texturen	115
4.8	Ausblick	118
5	Sightseeing in 3D	119
5.1	Einfache 3D-Szene	119
5.2	Bewegte Kamera	124
5.3	Springen und Drehen	126
5.4	Player mit Kamera	130
5.5	3rd oder 1st Person?	135
5.6	Fertig-Player aus der Packung?	140
5.7	Ausblick	146
6	Landschaften	147
6.1	Von der Ebene zum Terrain	147
6.2	Ein Gelände gestalten	151
6.3	Rundgang und Asset-Suche	156
6.4	Landschaftspflege	164
6.5	Vegetation	168
6.6	Noch mehr Details?	174
6.7	Ausblick	179
7	Erde, Wasser, Luft	181
7.1	Auf und ab	181
7.2	Grenzkontrollen	184
7.3	Wind	189
7.4	... und Wasser	192
7.5	Entschlackungskur	195
7.6	Kugel mit Rigidbody	196
7.7	Kollision mit Folgen	199
7.8	Ausblick	202
8	Bauwerke	203
8.1	Baumaterial	203
8.2	Platten legen	209
8.3	Prefab-Transport I	216
8.4	Prefab-Transport II	222
8.5	Innenansichten	224
8.6	Steigungen	229
8.7	Ausblick	233

9	Klettern und Schwimmen	235
9.1	Ein Kletter-Trigger	235
9.2	Der Player lernt klettern	238
9.3	Ein kleiner Schubs	242
9.4	See-Landschaft	247
9.5	Unterwasser-Atmosphäre	251
9.6	Waten, Schwimmen, Tauchen	256
9.7	Bewegungskontrolle	261
9.8	Ausblick	264
10	Animation und Navigation	265
10.1	Ein kleines Monster	265
10.2	Animator und Keyframes	269
10.3	Das »Ding« bewegt sich	274
10.4	Trigger-Animation	280
10.5	Ein Navigator für die Kreatur	282
10.6	Verfolgung an/aus	293
10.7	Hindernislauf	296
10.8	Ausblick	299
11	Leben oder Tod	301
11.1	Angriff und Verteidigung	301
11.2	Tödliche Kugeln	306
11.3	Animationen organisieren	309
11.4	Stehen – Gehen – Sterben	315
11.5	Tod des Players?	321
11.6	Die Kreatur wird zum Monster	324
11.7	Ausblick	331
12	Strahlen, Partikel und Sound	333
12.1	Raycasting	333
12.2	Todesstrahlen	339
12.3	Partikelsysteme	343
12.4	Flammenwerfer	352
12.5	Geräusche	353
12.6	Noch mehr Sound?	358
12.7	Ausblick	362
13	Game Tuning	363
13.1	Die Kreatur rüstet auf	363
13.2	Gesundheits-Balken	366
13.3	Energiekontrolle für den Player	373

13.4	... und für die Kreatur	379
13.5	Game Over	382
13.6	Aufmarsch der Gegner	386
13.7	Play the Game	390
13.8	Ausblick	396
14	Anhang	397
A.1	Unity installieren	397
A.2	Projekte und Links	407
A.3	Debugging	408
A.4	Kurze Checkliste	409
	Stichwortverzeichnis	411

Das erste Projekt

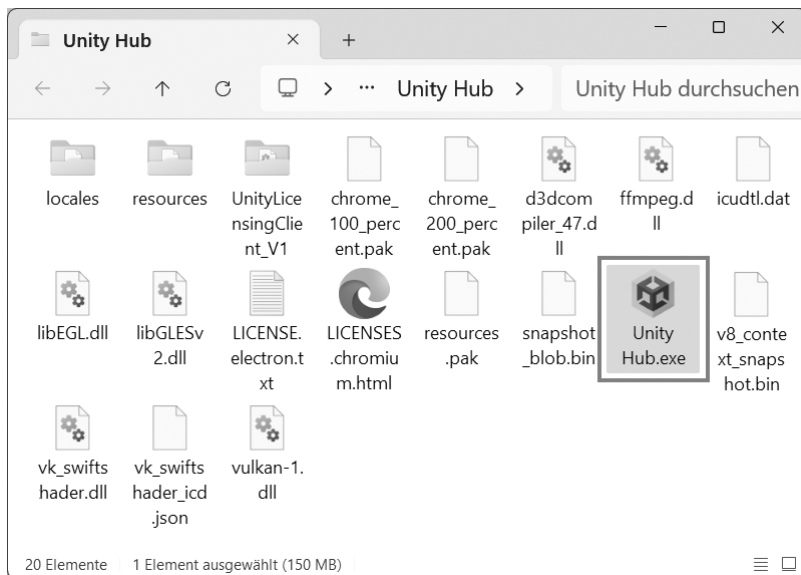
Bevor Sie Ihr erstes Spiel erstellen können, müssen Sie sich noch ein bisschen gedulden. Erst einmal machen Sie sich etwas mit der »Maschine« vertraut, mit der Sie später ein Werk »zaubern« wollen, das sich sehen und spielen lässt. Schon hier beginnen wir mit einem Projekt. Und wir spielen auch schon mal ein bisschen mit einem Objekt herum.

1.1 Unity starten

Bevor wir mit dem »Basteln« anfangen können, muss das Game-Entwicklungssystem *Unity* installiert werden. Wie das geht, steht im *Anhang*. Danach kann es direkt losgehen.

Es gibt mehrere Wege, um Unity zu starten. Einer ist dieser:

1. Öffnen Sie den Ordner, in den Sie Unity installiert haben (bei mir ist das der Unterordner `UNITY HUB` im Ordner `PROGRAMME` auf Laufwerk `C:`).



- Suchen Sie nun unter den vielen Symbolen eines heraus, das wie eine Art schwarzer Würfel aussieht, es muss den Namen **Unity hub.exe** tragen. Dann starten Sie das Programm mit einem Doppelklick auf das Symbol.



Start-Symbol

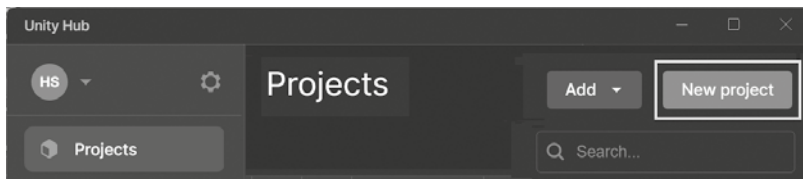
Weil wir Unity ja sehr oft starten werden, empfehle ich hier, eine Verknüpfung auf dem Desktop anzulegen:

- Klicken Sie dazu mit der rechten Maustaste auf das entsprechende Unity-Symbol (Unity Hub.exe). Im Kontextmenü wählen Sie **KOPIEREN**.
- Dann klicken Sie auf eine freie Stelle auf dem Desktop, ebenfalls mit der rechten Maustaste. Im Kontextmenü wählen Sie **VERKNÜPFUNG EINFÜGEN**.

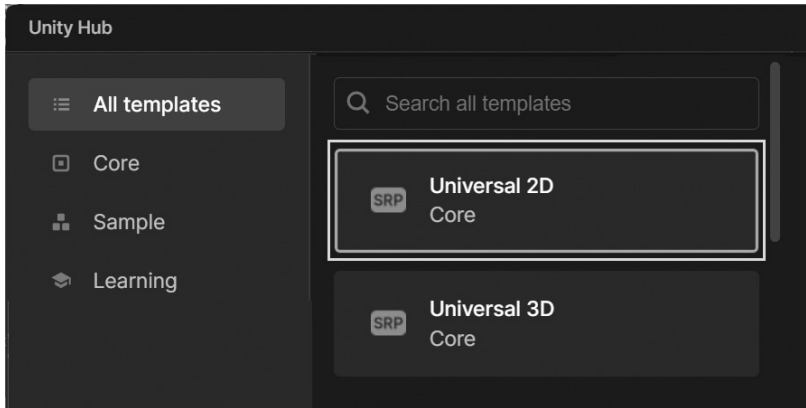
Es ist sinnvoll, das neue Symbol auf dem Desktop umzubenennen, z.B. von **UNITY HUB.EXE – VERKNÜPFUNG** in einfach nur **UNITY**.

Von nun an doppelklicken Sie einfach auf das neue Symbol, und Unity wird gestartet.

Je nach Computer kann es eine Weile dauern, bis Unity Hub geladen ist. Einige Zeit später erscheint ein neues Fenster:

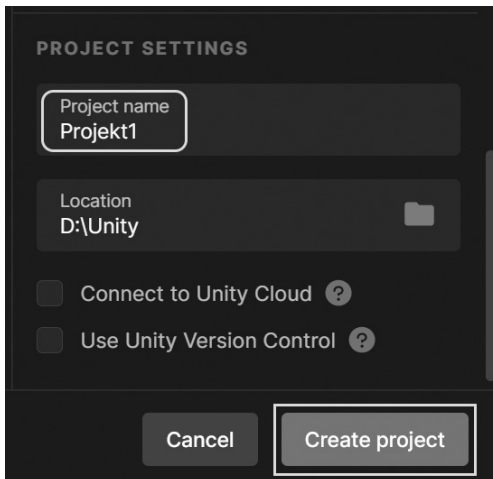


- Klicken Sie dazu auf **NEW PROJECT**.



4. Wählen Sie im neuen Dialogfeld die Einstellung **UNIVERSAL 2D**.
Mit 3D beschäftigen wir uns später noch ausführlich.
5. Geben Sie dann im Feld für **PROJECT NAME** einen Namen für Ihr neues Projekt ein. Bei **LOCATION** sollte der Ordner stehen, in dem das Projekt untergebracht werden soll (wenn Sie nichts eingeben, schafft sich Unity seinen eigenen Ordner für Ihre Spielprojekte.)

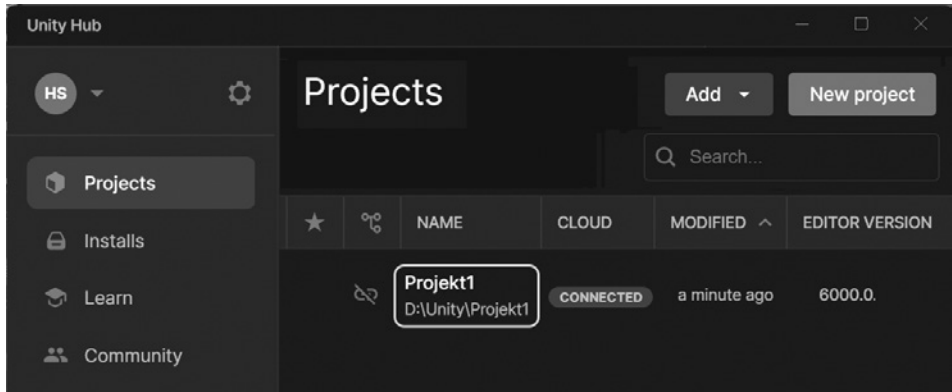
Ich benutze einen Ordner **UNITY** und nenne mein erstes Projekt schlicht und einfach **PROJEKT1**.



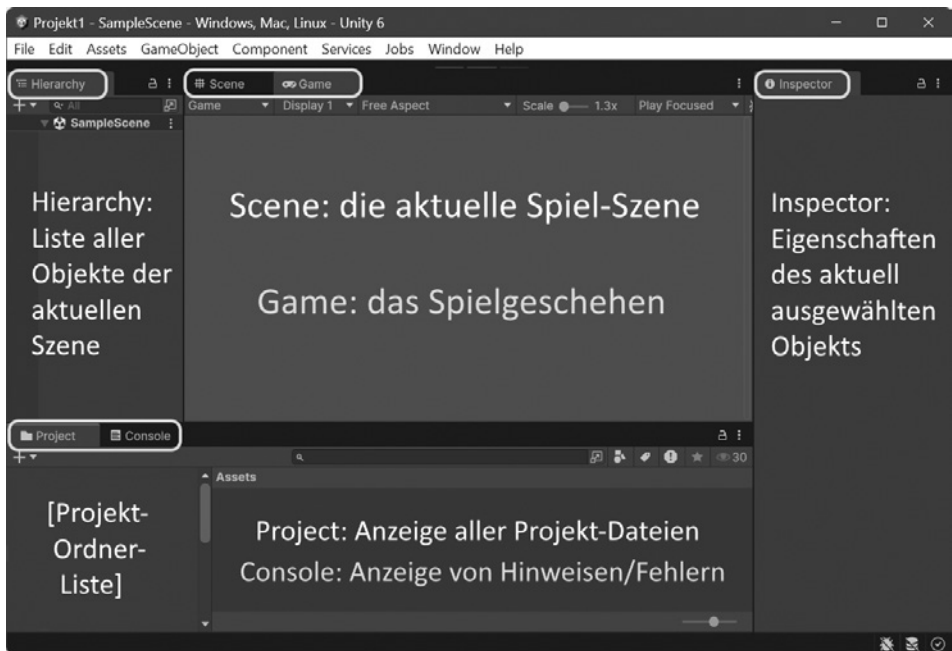
6. Klicken Sie zum Abschluss auf **CREATE PROJECT**.
Es dauert nun eine Weile, bis Ihr Projekt in der Liste unter **PROJECTS** auftaucht.

Kapitel 1

Das erste Projekt



Anschließend zeigt uns Unity endlich seine Arbeitsumgebung. Schauen wir uns erst einmal die Aufteilung der wichtigsten Fensterbereiche an:

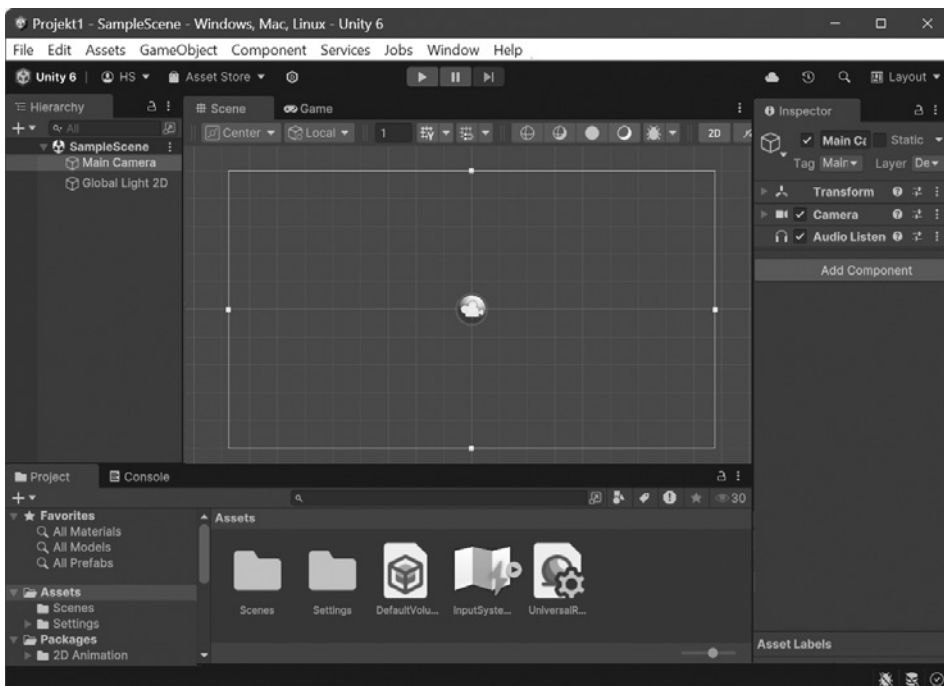


- Im GAME-Fenster ist es erst einmal leer. Da sehen Sie später Ihr Spiel in Echtzeit ablaufen, wenn Sie es durch einen der darüberliegenden Buttons gestartet haben.
- Dahinter liegt das SCENE-Fenster. Und es gibt auch schon zwei Objekte: Kamera und Licht. Doch für ein Spiel brauchen wir dann noch mindestens ein weiteres Objekt wie eine Kugel oder eine Figur.

- Im HIERARCHY-Fenster sind bis jetzt nur MAIN CAMERA und ggf. GLOBAL LIGHT aufgelistet. Dort stehen dann später auch alle Objekte, die zur Szene eines Spiels gehören (jedes Spiel könnte mehrere Szenen haben).
- Das PROJECT-Fenster erfasst die Ordner mit dem gesamten Zubehör für alle Spielszenen. Dazu gehören natürlich u.a. auch Programmteile. Bilder, die Sie als Spiel-Objekt einsetzen wollen (wie z.B. eine Kugel oder eine Figur), lassen sich einfach mit der Maus aus einem Ordnerfenster unter Windows hier hineinziehen. Damit wird die entsprechende Datei ins Projekt kopiert.
- Dahinter findet sich das CONSOLE-Fenster, das sich u.a. bei Fehlern meldet. Außerdem lassen sich dort Daten anzeigen, z.B. von Spiel-Objekten.
- Um sich die Eigenschaften eines Objekts nicht nur anzuschauen, sondern auch bearbeiten zu können, gibt es das INSPECTOR-Fenster. Damit werden wir des Öfteren zu tun haben.

Schalten Sie mal vom GAME-Fenster ins SCENE-Fenster um.

In der Mitte ist das Symbol für die Kamera.



Wenn Sie im HIERARCHY-Fenster auf MAIN CAMERA klicken, zeigt der INSPECTOR plötzlich eine ganze Menge an (ändern sollten Sie aber daran nichts).

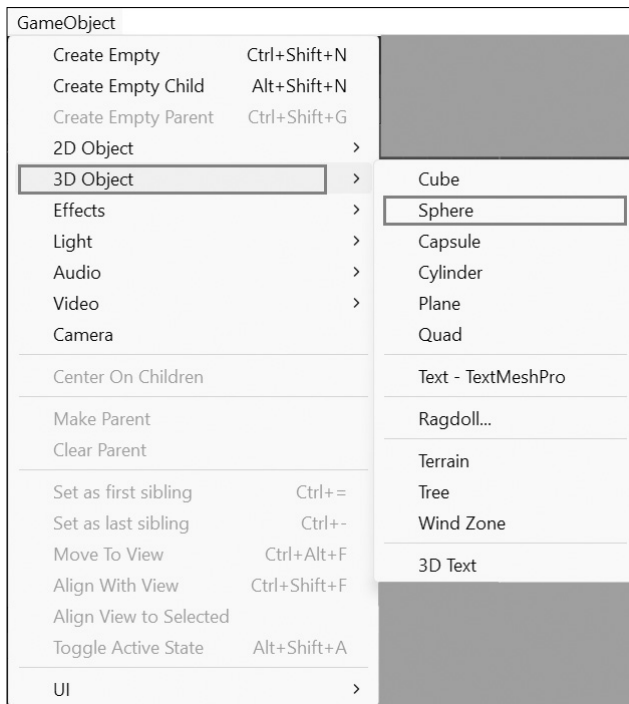
Mit dem Hauptmenü bekommen wir immer wieder zu tun. Die Bedeutung der meisten Menüpunkte klären wir nach und nach.



1.2 Ein Objekt zum Spielen

Wir beginnen mit etwas Einfachem. Dazu brauchen wir eine Kugel, und die soll sich über das Spielfeld bewegen lassen, z.B. mit der Maus oder mit den Tasten.

1. Klicken Sie oben im Hauptmenü auf **GAMEOBJECT** und dann auf den Eintrag **3D OBJECT**. Im Zusatzmenü bekommen Sie nun eine Auswahl. Klicken Sie auf den Eintrag **SPHERE** (= Kugel).



Anschließend taucht im **SCENE**-Fenster etwas auf, das bei genauerem Hinsehen wie ein Kreis aussieht – aber irgendwie auch recht mickrig. Außerdem zeigt der **INSPECTOR** zahlreiche Informationen über unser neues Spiel-Objekt.