HANS-GEORG SCHUMANN

SPIELE PROGRAMMIEREN MIT UNITY FÜR KIDS

2. AUFLAGE

MIT SPASS 3D-SPIELE SELBST ENTWICKELN

INHALT

EINLEITUNG

Eigene Fantasiewelten erschaffen, in denen man sich frei bewegen kann. Selbst gebauten Figuren begegnen. Abenteuer selbst erfinden, den Verlauf von Ereignissen und diese selbst bestimmen: Wie wäre das?

Um ein Spiel selbst zu erstellen, muss man vom Programmieren anfangs eigentlich noch gar nichts verstehen. Denn zuallererst braucht man eine Idee und dann einen Plan.

Wovon soll das Spiel handeln? Welche Geschichte soll es erzählen? Personen, Orte und Ereignisse, all das führt zu einem Plan, der umfasst, was zu diesem Spiel gehören soll. Und erst, wenn der Plan »steht«, kann die eigentliche Umsetzung in ein Programmprojekt beginnen. Dann allerdings sollte man schon möglichst gut programmieren können.

Wie **du** auf eine gute Idee kommst, wie **du** deine eigene Geschichte »strickst« und wie **du** einen Plan für ein Spiel aufstellst, wirst du in diesem Buch nicht erfahren. Hier bekommst du einen Kasten voller Werkzeuge, mit dem du deinen Plan in ein Spiel umsetzen kannst.

Wir wollen auch gar nicht so hoch hinaus: Ein professionelles Game wird heutzutage ja von einer ganzen Gruppe von Leuten erstellt, darunter Designer, Künstler, Techniker und nicht zuletzt natürlich Programmierer.

Trotzdem dauert die Arbeitszeit häufig mindestens Monate, wenn nicht Jahre. Die Beteiligten machen einen Vollzeitjob, es ist ihr Beruf. Hier hast du als Einzelgänger nur eine Chance, wenn deine Spiel-Idee so hervorragend und einmalig ist, dass sie andere überschattet.

Bleiben wir also auf dem harten Boden der Tatsachen und planen nicht ein gigantisches Meisterwerk, sondern kümmern uns um solide Grundlagen. Wenn du die beherrschst, hast du durchaus Voraussetzungen, auch einmal an einem professionellen Spielprojekt mitzuwirken.

WELCHE WERKZEUGE BENÖTIGEN WIR?

Um Spiele im 2D- und 3D-Bereich zu erstellen, brauchen wir als Herzstück eine sogenannte Game-**Engine**. Sie muss mit physikalischen Gesetzen umgehen können, damit die Spielwelt mit ihren Figuren und Ereignissen möglichst echt wirkt. Und sie muss komplexe grafische Effekte beherrschen, damit das Ganze auch optisch etwas hermacht.

Das brauchst du:

- Mit **Unity** (früher Unity3D) haben wir nicht nur eine vollwertige Engine, die so vielfältige Möglichkeiten bietet, dass dieses Buch allein nicht reicht, um alles zu beschreiben. Mit dem visuellen Editor lässt sich ein Spiel bequem erstellen – und das nicht nur für Windows, sondern auch für andere Plattformen, wie z.B. Android oder iOS.

Wir verwenden hier die derzeit aktuelle Unity-Version 2020 (für eventuelle Neuerungen schaust du am besten unter *unity.com* nach).

Die visuelle Entwicklungsumgebung, die Unity mit sich bringt, bietet schon viele Möglichkeiten, sich Elemente für ein Spiel zusammenzustellen. Damit aber alles wie gewünscht funktioniert und um die volle Leistung der Engine abzurufen, braucht man ein Programmiersystem:

- **Visual Studio** bietet die Möglichkeit, die für Unity nötigen Skripts in einer passenden Programmiersprache, nämlich C#, zu erstellen. Dieses System gibt es kostenlos bei Microsoft (unter *visualstudio.microsoft.com*). Es kann aber auch über die Unity-Installation automatisch so eingerichtet werden, dass es aus Unity heraus einsetzbar ist.

Damit wir auch eigene Welten und eigene Akteure für unsere Spiele erstellen können, gibt es in Unity weitere leistungsstarke Hilfsmittel, wie z.B. einen Editor u.a. für Landschaften und Bäume. Viele mit anderen Programmen hergestellte Figuren und Elemente lassen sich leicht in Unity importieren und dort einsetzen.

WAS BIETET DIESES BUCH?

Richtig los geht es wie bei jedem Buch in Kapitel 1. Bist du ein erfahrener C#-Programmierer, so wird es für dich kein Problem sein, auch dort zu starten. Willst du aber dein C#-Wissen lieber noch mal überprüfen oder festigen, dann kannst du dies in einem vorgeschalteten Kapitel tun, das du zum Download auf der Homepage *www.mitp.de/0170* findest.

Ansonsten erfährst du hier u.a.,

- wie man in Unity Spiele mit Sprites programmieren kann,
- wie man den 1st-Person und den 3rd-Person-Modus einsetzt,
- wie man Landschaften gestaltet und einfache Gebäude baut,
- wie man klettert, schwimmt und taucht,
- wie man einfache Figuren erstellt und sie animiert,
- wie man künstliche Intelligenz nutzen kann,
- wie man Effekte erzeugt und Sound einsetzt.

Im **Anhang** gibt es dann noch zusätzliche Informationen, z.B. wie man Unity und Visual Studio installiert und wie man Fehler vermeidet.

WIE ARBEITE ICH MIT DIESEM BUCH?

Um dir den Weg vom ersten 2D-Projekt bis zu einem 3D-Game mit Akteuren einfacher zu machen, gibt es einige zusätzliche Symbole, die ich dir hier gern erklären möchte:

ARBEITSSCHRITTE

≫ Wenn du dieses Zeichen siehst, heißt das: Es gibt etwas zu tun. Damit kommen wir beim Programmieren Schritt für Schritt einem neuen Ziel immer näher.

Grundsätzlich lernt man besser, wenn man Objekte selbst einsetzt oder einen Programmtext selbst eintippt oder ändert. Aber nicht immer hat man große Lust dazu. Weil alle Projekte im Buch auch in einem Download-Paket verfügbar sind, findest du hinter einem Programmierschritt auch den Namen des jeweiligen Projekt-Ordners (z.B. GAME1). Wenn du also bestimmte Projekte nicht selbst erstellen willst, kannst du stattdessen die dazugehörigen Dateien herunterladen (zu finden unter *mitp.de/0170*).

FRAGEN UND AUFGABEN

Am Ende eines Kapitels gibt es jeweils eine Reihe von Fragen und Aufgaben. Diese Übungen sind nicht immer ganz einfach, aber sie helfen dir, deine Spiele noch besser zu entwickeln. Lösungen zu den Aufgaben findest du auch im Download von der mitp-Seite (dort findest du auch die Programme zu den Aufgaben).

Du kannst sie dir alle im Editor von Windows oder auch in deinem Textverarbeitungsprogramm anschauen. Oder du lässt sie dir ausdrucken und hast sie dann schwarz auf weiß, um sie neben deinen PC zu legen.

NOTFÄLLE

Vielleicht hast du irgendetwas falsch gemacht oder etwas vergessen. Oder es wird gerade knifflig. Dann fragst du dich, was du nun tun sollst. Bei diesem Symbol findest du eine Lösungsmöglichkeit. Notfalls kannst du aber auch ganz hinten im Anhang B nachschauen, um ein paar Hinweise zur Pannenhilfe zu finden.

WICHTIGE STELLEN IM BUCH

Hin und wieder findest du ein solch dickes Ausrufezeichen im Buch. Dann ist das eine Stelle, an der etwas besonders Wichtiges steht.

Wenn du ein solches »Wow« siehst, geht es um ausführlichere Informationen zu einem Thema.

WAS BRAUCHST DU FÜR DIESES BUCH?

Du findest das Unity-Paket als komplette Entwicklungsumgebung zu Download und Installation auf der entsprechenden Seite von **Unity**:

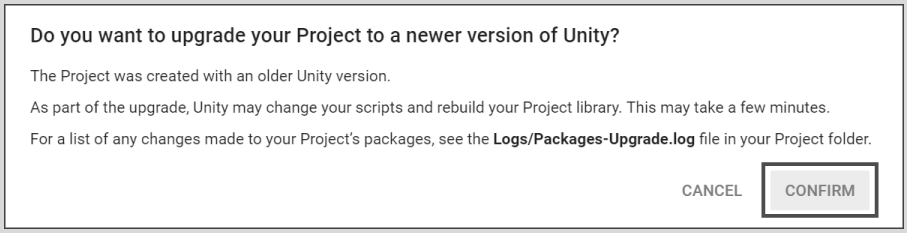
https://unity.com/

Zusätzlich gibt es dort noch eine Sammlung von Hilfsmitteln, die sogenannten Assets. Du findest dort auch eine Menge Zusatzmaterial. Während Unity dich nichts kostet, solange du nicht damit Geld verdienen willst, sind viele Zusätze nicht kostenlos.

Die Beispielprojekte in diesem Buch findest du ebenso wie die Lösungen zu den Aufgaben auf der Homepage des Verlags in der gerade aktuellen Version:

http://www.mitp.de/0170

Die Programmbeispiele sind mit einer Unity-Version erstellt, die vielleicht schon nicht mehr die allerneueste ist, wenn du dieses Buch kaufst oder eine Weile besitzt. Denn Unity wird natürlich ständig weiterentwickelt. Wenn du eines der Projekte in einer neueren Unity-Version öffnest, bekommst du eine solche Meldung:



Mit einem Klick auf CONFIRM wird das alte Projekt für die aktuelle Unity-Version umgewandelt.

BETRIEBSSYSTEM

Die meisten Computer arbeiten heute mit dem Betriebssystem Windows. Am besten geeignet ist Version 10. Mit Unity lassen sich allerdings nicht nur Spiele für Windows entwickeln, sondern auch für andere Systeme wie Googles Android, Apples iOS, ja sogar Microsofts Xbox und Sonys Playstation.

SPEICHERMEDIEN

Auf jeden Fall benötigst du etwas wie einen USB-Stick oder eine SD-Card, auch wenn du deine Programme auf die Festplatte speichern willst. Auf einem externen Speicher sind deine Arbeiten auf jeden Fall zusätzlich sicher aufgehoben.

Gegebenenfalls bitte deine Eltern oder Lehrer um Hilfe.

WIE GUT KENNST DU C#?

Vielleicht kennst du bereits eine andere Programmiersprache, wie z.B. JavaScript, Python oder auch Basic. Dann sollte dir der Umstieg auf C# nicht schwer fallen. Im Idealfall hast du bereits in C# programmiert und bist mit den Grundlagen vertraut.

Aber auch wenn du keine Ahnung vom Programmieren, aber unbändige Lust auf die Spiele-Programmierung mit Unity hast, kannst du einfach so in Kapitel 1 einsteigen und versuchen, mitzukommen. Programmiert wird erst ab Kapitel 2.

HINWEISE FÜR LEHRER

Dieses Buch lässt sich selbstverständlich auch für den Informatik-Unterricht verwenden. Dort setzt natürlich jeder Lehrer seine eigenen Schwerpunkte. Aber wenn es z.B. um eine Programmier-AG oder einen Informatikkurs mit Schwerpunkt Spieleprogrammierung geht, lässt sich dieses Buch in Ergänzung zu Ihrem Lehrmaterial gut einsetzen. Mit Unity steht Ihnen und Ihren Schülern ein mächtiges Entwicklungswerkzeug zur Verfügung.

Was die Programmierung angeht, so wird hier die Sprache C# genutzt, die der Programmiersprache Java recht ähnlich, aus meiner Sicht aber einfacher zu erlernen und einzusetzen ist, weil C# die Vorteile vieler anderer Sprachen vereint und nur wenige Nachteile hat.

Die einzelnen Spielprojekte werden vorwiegend über den visuellen Editor von Unity erstellt, die Programmierung in C# dient der Erstellung von Scripts, um den Spielverlauf zu steuern. Zahlreiche Fragen und Aufgaben mit Lösungen helfen, Gelerntes zu festigen und zu vertiefen.

ÜBUNGSMEDIEN

Für den Informatik-Unterricht sollte jeder Schüler ein eigenes externes Speichermedium haben, um darauf seine Versuche zu sichern. So wird verhindert, dass sich auf der Festplatte des Schulcomputers mit der Zeit allerlei »Datenmüll« ansammelt. Außerdem dient der eigene Datenträger dem Datenschutz: Nur der betreffende Schüler kann seine Daten manipulieren.

REGELMÄßIG SICHERN

Es kann nicht schaden, die Programmdateien, an denen gerade gearbeitet wird, etwa alle **zehn** Minuten zu speichern. Denn Computer pflegen gern gerade dann »abzustürzen«, wenn man seine Arbeit längere Zeit nicht gespeichert hat.

1 DAS ERSTE PROJEKT

Du möchtest natürlich gleich hier an Ort und Stelle dein erstes Spiel erstellen. Daraus wird nichts. Noch nicht. Als Erstes brauchst du Geduld. Und Ausdauer. Hier starten wir aber schon die Maschine, mit der wir später einiges zaubern werden, das sich sehen und spielen lässt. Wir fangen mit einem Projekt an. Und wir spielen auch schon mal ein bisschen mit einem Objekt herum.

In diesem Kapitel lernst du

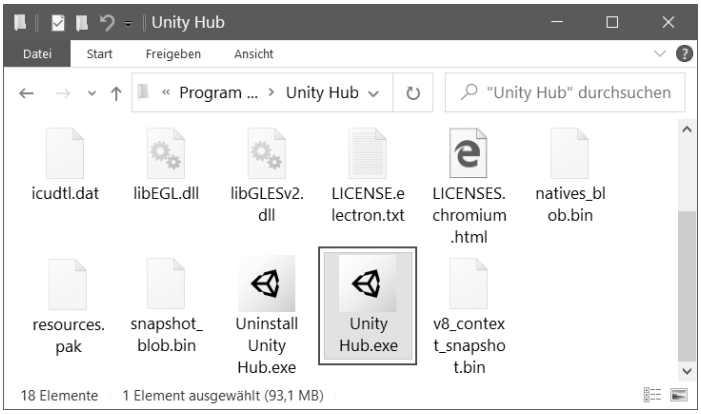
- wie man Unity startet
- deine »Werkbank« kennen
- wie man Spiel-Objekte in einem Projekt einsetzt
- ein bisschen über Spiele-Physik
- etwas über 2D und 3D
- wie man die Position und Größe von Objekten ändert
- wie man Unity beendet

UNITY STARTEN

Bevor wir mit dem Basteln anfangen können, muss das Game-Entwicklungssystem **Unity** installiert werden. Wie das geht, erfährst du in **Anhang A**. Wenn du dir das Einrichten nicht allein zutraust, solltest du dir von jemandem helfen lassen.

Eine Möglichkeit, Unity zu starten, ist diese:

≫ Öffne den Ordner, in den du Unity untergebracht hast (bei mir ist das der Unterordner UNITY HUB im Ordner PROGRAMME auf Laufwerk C:).



Hier suchst du unter den vielen Symbolen eines von denen heraus, die wie eine Art schwarzer Würfel aussehen, und zwar das mit dem Namen UNITY HUB.EXE.

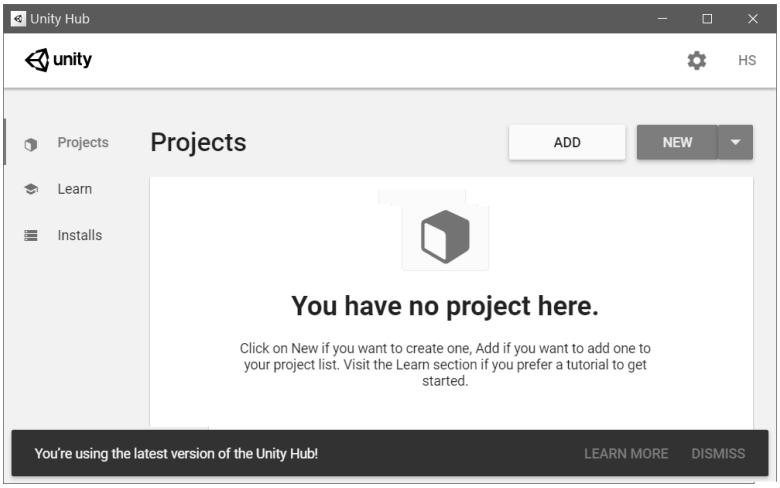
≫ Starte das Programm mit einem Doppelklick auf das Symbol.

Ich empfehle dir, eine **Verknüpfung** auf dem Desktop anzulegen:

- Dazu klickst du mit der rechten Maustaste auf das entsprechende Unity-Symbol (Unity Hub.exe). Im Kontextmenü wählst du KOPIEREN.
- Dann klicke auf eine freie Stelle auf dem Desktop, ebenfalls mit der rechten Maustaste. Im Kontextmenü wählst du VERKNÜPFUNG EINFÜGEN.
- Es ist sinnvoll, für das neue Symbol auf dem Desktop den Text `unity hub.exe - Verknüpfung` einfach durch `Unity` zu ersetzen.

Von nun an kannst du auf das neue Symbol doppelklicken und damit Unity starten.

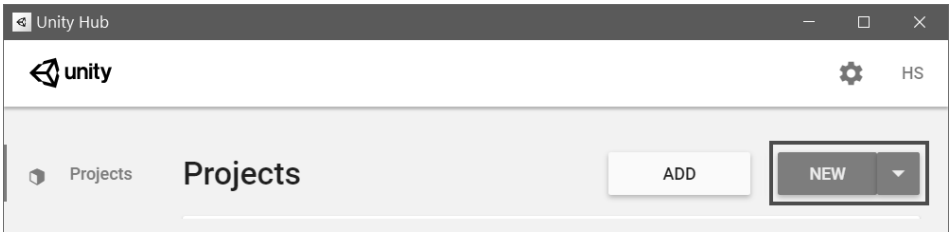
Je nach Computer kann es eine Weile dauern, bis Unity geladen ist. Einige Zeit später landest du in einem neuen Fenster.



Startest du Unity zum allerersten Mal seit der Installation, dann musst du es wahrscheinlich noch aktivieren. Wenn du das nicht allein hinkriegst, schau mal in Anhang A nach.

Dort hast du nun die Möglichkeit, ein Projekt zu öffnen (wenn vorhanden). Oder eines neu zu erstellen.

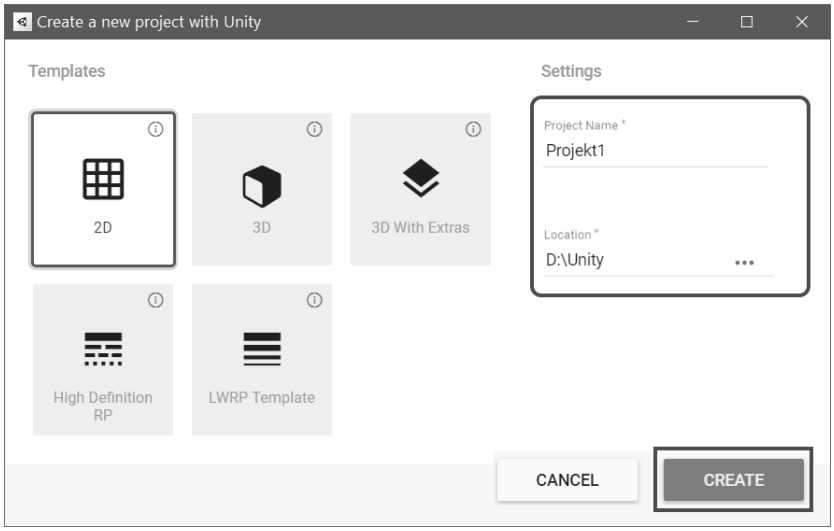
≫ Klicke auf NEW.



≫ Im neuen Dialogfeld wählst du die Einstellung 2D. Mit 3D beschäftigen wir uns später.

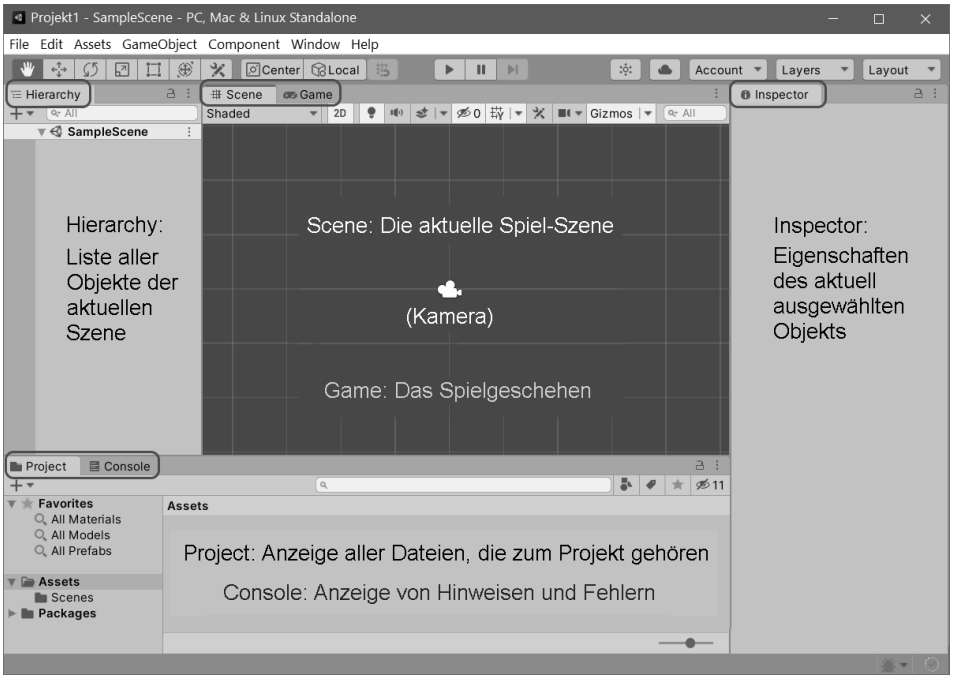
≫ Gib im Feld für PROJECT NAME den Namen des neuen Projekts ein. Und dann bei LOCATION den Ordner, wo du dein Projekt unterbringen willst. (Wenn du alles so stehen lässt, schafft sich Unity seinen eigenen Ordner für deine Spiel-Projekte.)

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



≫ Klicke dann auf CREATE.

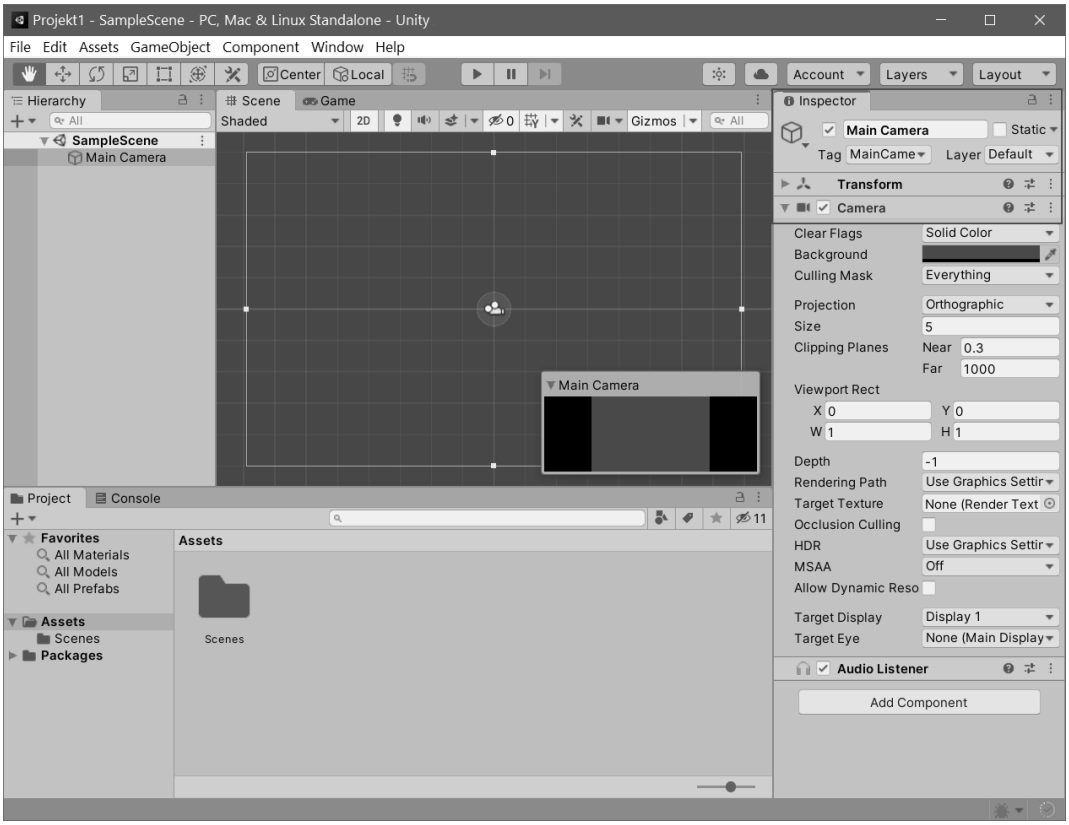
Und endlich zeigt uns Unity sein wahres Gesicht (was eine ganze Weile dauern kann). Schauen wir uns erst einmal die Aufteilung der vier wichtigsten Fensterbereiche an:



◇ Im SCENE-Fenster sieht man erst einmal nur die Kamera. Für ein Spiel brauchen wir dann noch mindestens ein weiteres Objekt wie eine Kugel oder eine Figur.

- Dahinter liegt das GAME-Fenster: Hier siehst du dein Spiel in Echtzeit ablaufen, wenn du es durch einen der darüberliegenden Buttons gestartet hast.
- Im HIERARCHY-Fenster ist bis jetzt nur die MAIN CAMERA aufgelistet. Dort stehen dann später alle Objekte, die zur Szene eines Spiels gehören (jedes Spiel könnte also mehrere Szenen haben).
- Das PROJECT-Fenster erfasst die Ordner mit dem gesamten Zubehör für das ganze Spiel. Dazu gehören natürlich u.a. auch Programmteile. Bilder, die du als Spielobjekt einsetzen willst (wie z.B. eine Kugel oder eine Figur) lassen sich einfach mit der Maus aus einem Ordnerfenster unter Windows hier hineinziehen. Womit die entsprechende Datei ins Projekt kopiert wird.
- Dahinter findest du das CONSOLE-Fenster, das sich bei Fehlern meldet. Außerdem lassen sich dort Werte, z.B. von Spiel-Objekten, anzeigen.
- Um sich die Eigenschaften eines Objekts nicht nur anzuschauen, sondern auch bearbeiten zu können, gibt es das INSPECTOR-Fenster.

➢ Klicke mal im HIERARCHY-Fenster auf MAIN CAMERA. Dann verändert sich nicht nur die Darstellung im SCENE-Fenster, sondern der INSPECTOR zeigt plötzlich eine ganze Menge an (ändern solltest du allerdings daran nichts).

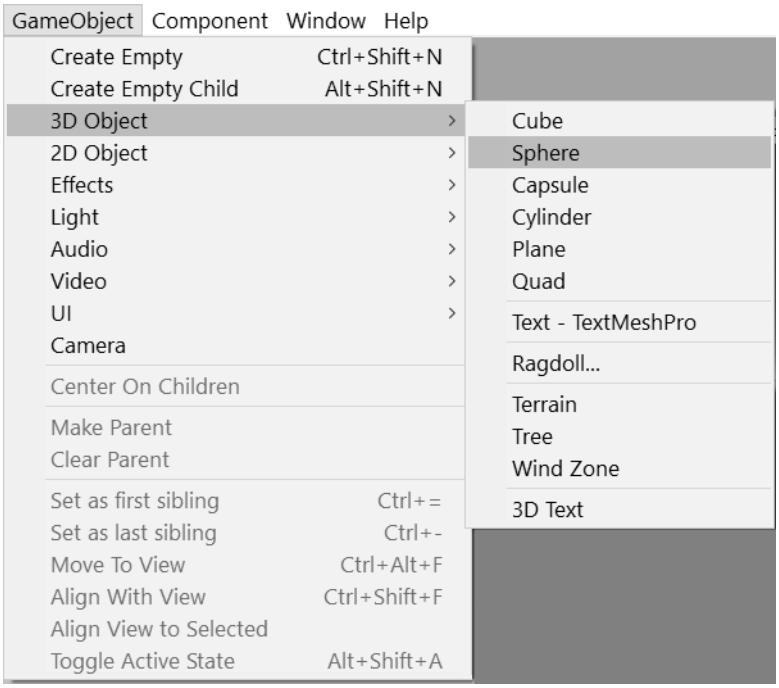


Um das Hauptmenü und die darunterliegenden Symbole geht es erst mal nicht, damit bekommen wir später noch genug zu tun.

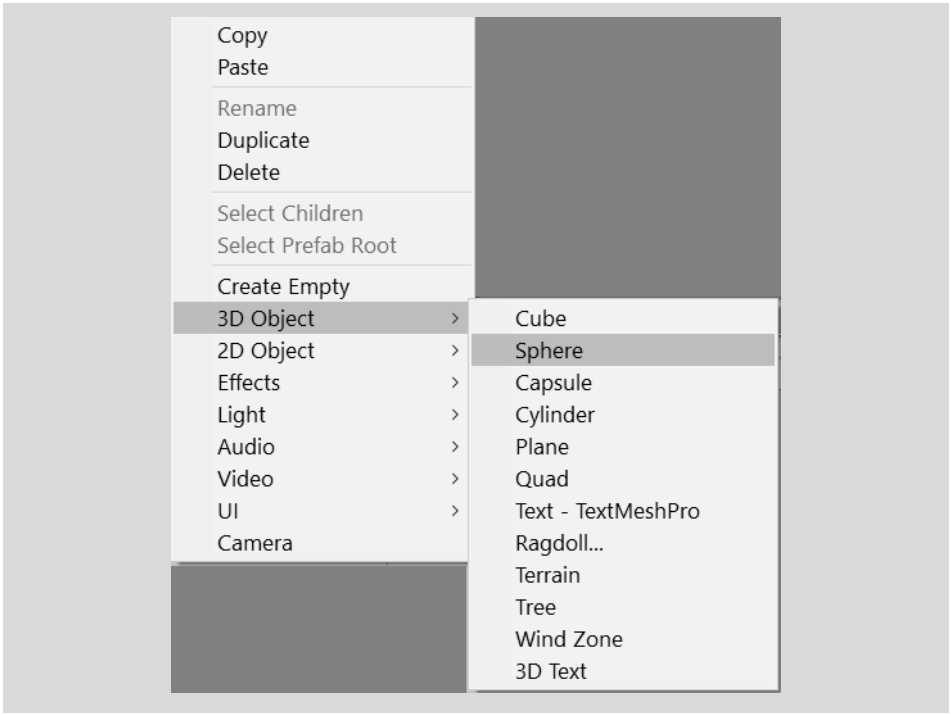
EIN OBJEKT ZUM SPIELEN

Wir beginnen mit etwas Einfachem. Wir brauchen eine Kugel und die soll sich über das Spielfeld bewegen lassen. Mit der Maus oder mit den Tasten zum Beispiel.

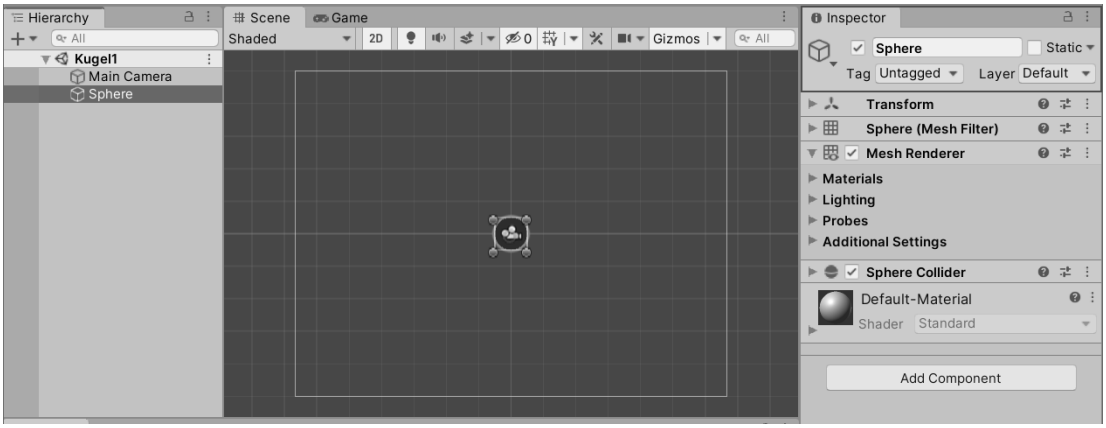
≫ Kommen wir zum Hauptmenü. Klicke dort auf GAMEOBJECT und dann auf den Eintrag 3D OBJECT. Im Zusatzmenü bekommst du nun eine Auswahl. Klicke auf den Eintrag SPHERE (= Kugel).



Alternativ zum Hauptmenü lässt sich ein solches Objekt auch direkt im HIERARCHY-Fenster erstellen. Dazu öffnest du mit der **rechten** Maustaste ein Kontextmenü, und über 3D OBJECT ein weiteres. Dort klickst du auf das gewünschte Objekt, wie z.B. SPHERE.



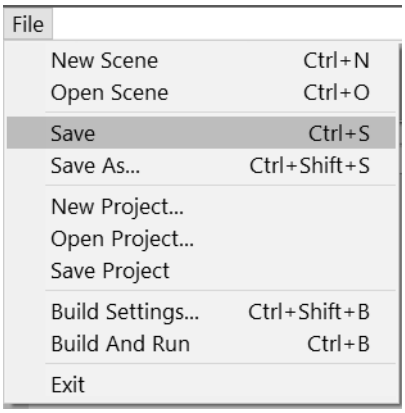
Anschließend taucht im SCENE-Fenster etwas auf, das wie ein Kreis aussieht. Außerdem zeigt der INSPECTOR zahlreiche Informationen über unser neues Spiel-Objekt.



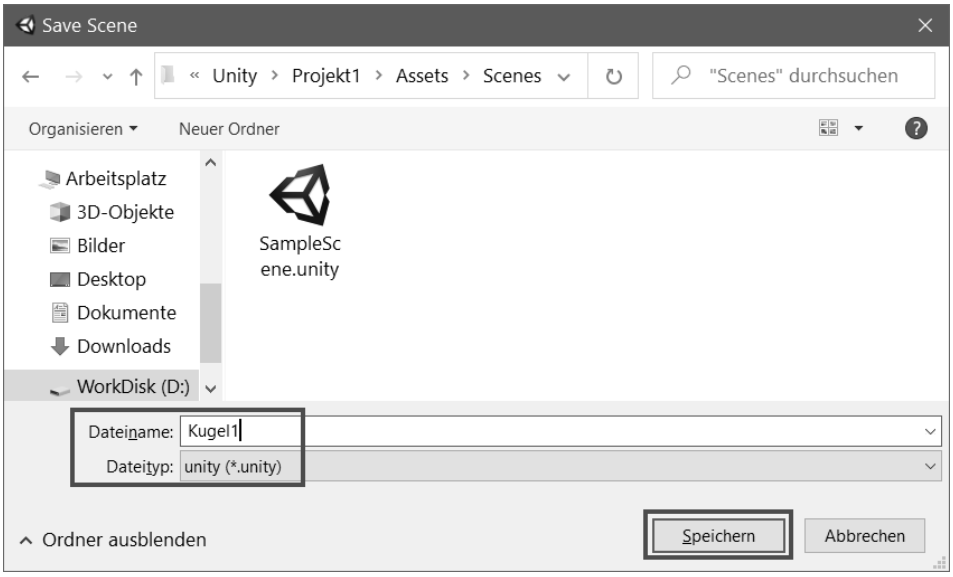
Dargestellt sind nun zwei Objekte: Die Kamera (auf die wir noch zu sprechen kommen) und darunter oder dahinter die von uns erzeugte Kugel.

Nun ist es an der Zeit, die ganze Szene schon einmal zu speichern.

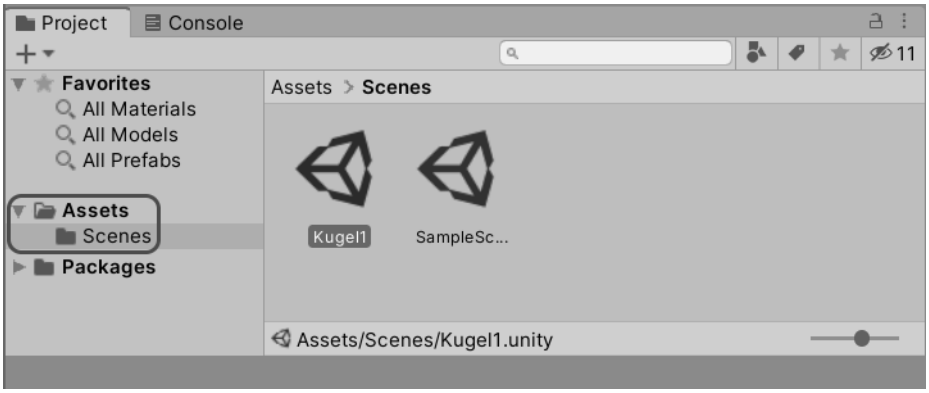
≫ Klicke auf FILE und SAVE.



Bei Unity heißt diese Szene zuerst SAMPLESCENE. Passt dir der Name nicht, dann klicke auf SPEICHERN UNTER und gib im Dialogfeld einen Namen ein, z.B. `Kugel1` (wenn dir nichts Besseres einfällt). Die Kennung UNITY wird automatisch angefügt. Dann klicke auf SPEICHERN.

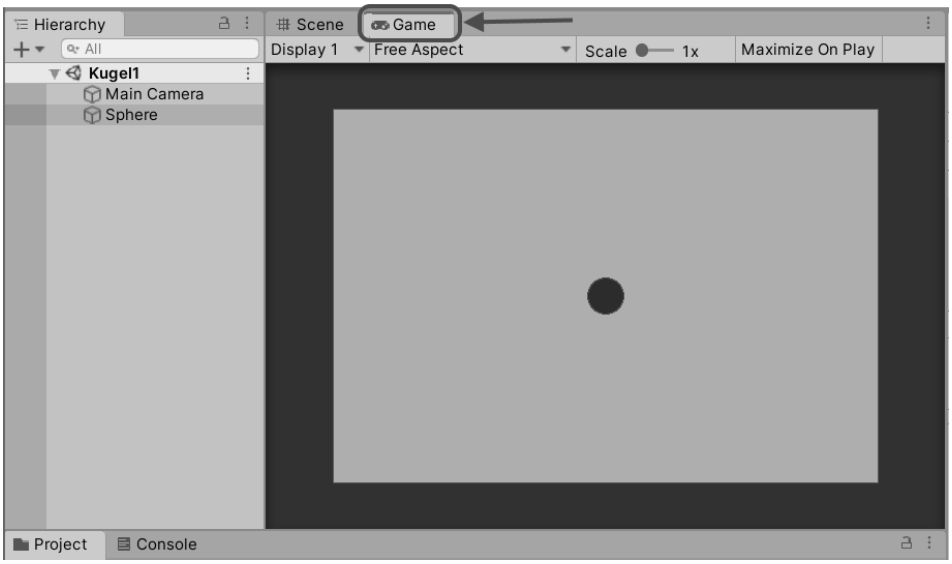


Wenn du anschließend im PROJECT-Fenster auf ASSETS klickst, siehst du da den Ordner SCENES, darin befindet sich das Symbol für die Szenen-Dateien.



Und nun schauen wir uns diese Szene einmal genauer an, und zwar in einem anderen Fenster.

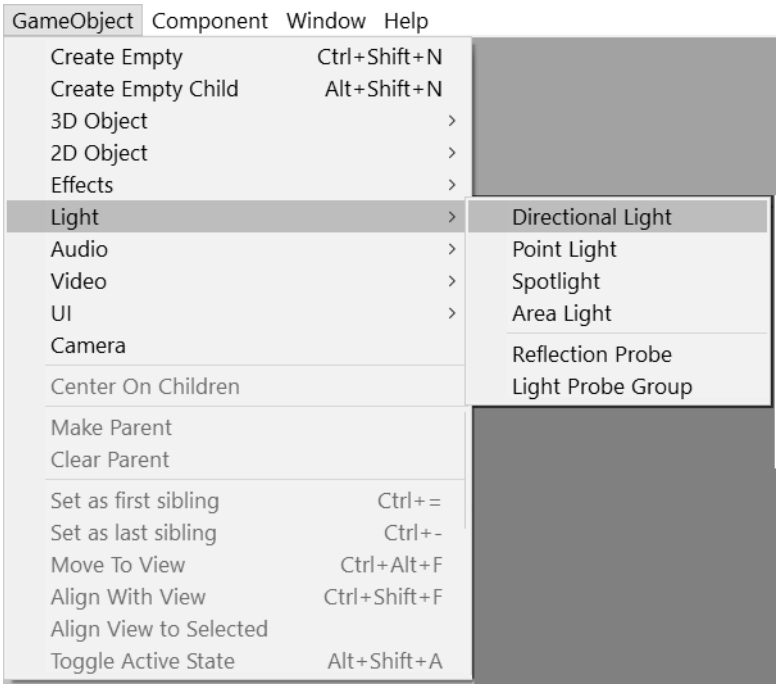
≫ Dazu klickst du auf den Reiter mit dem Text GAME (direkt rechts neben dem SCENE-Reiter).



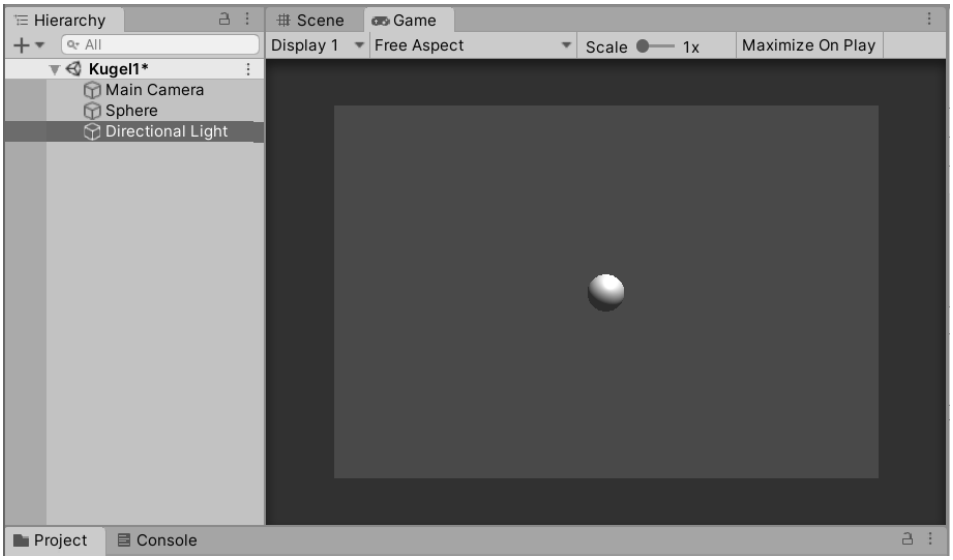
Du landest im GAME-Fenster. Das hat sich bis jetzt hinter dem SCENE-Fenster versteckt.

Hier kannst du sehen, wie dein Spiel in Aktion aussieht. Bis jetzt ist da noch alles dunkel und es bewegt sich nichts. An der Dunkelheit können wir sofort etwas ändern.

≫ Klicke im Hauptmenü auf GAMEOBJECT und dann auf LIGHT. Im Zusatzmenü klickst du auf den Eintrag DIRECTIONAL LIGHT.

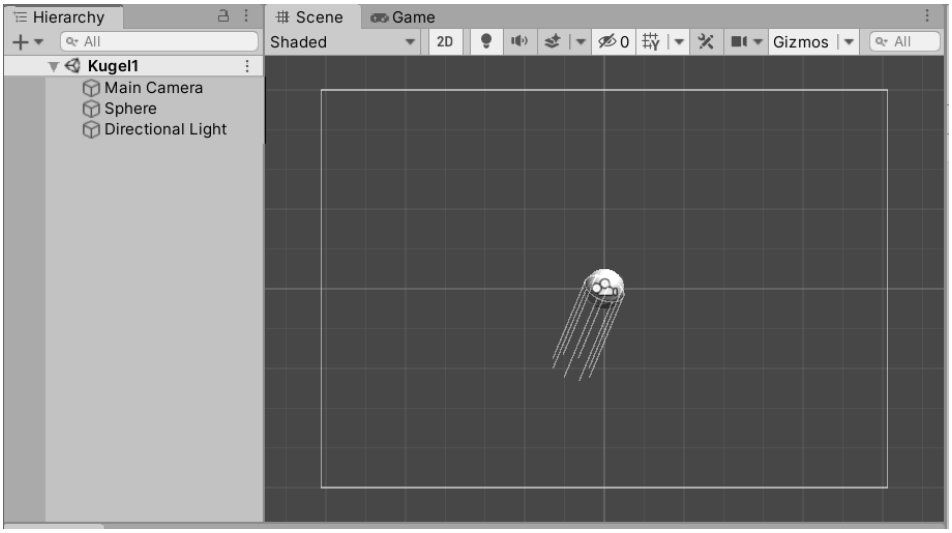


Kurz darauf siehst du deine Kugel in einem anderen Licht. Nur im GAME-Fenster?



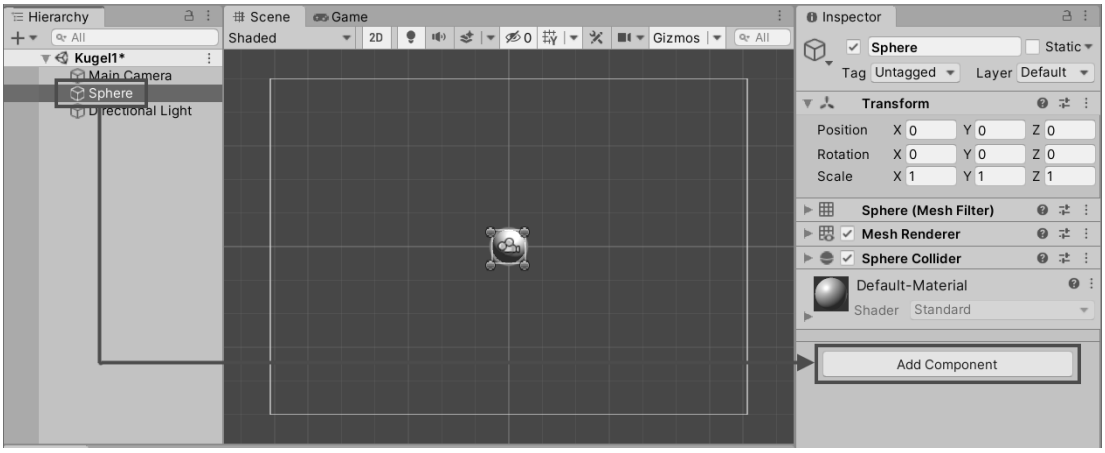
≫ Klick dich doch mal zurück zum SCENE-Fenster.

Dort gibt es auch eine Änderung, wie du sehen kannst. Das neue Spiel-Objekt wird als Symbol aus Strahlen dargestellt.



Da haben wir nun eine Kugel, die sich sonnt. Aber sie soll sich bewegen. Doch wie kriegen wir sie dazu? Zuerst einmal sollten wir diesem »Ding« physikalische Eigenschaften geben. Dass das Objekt aussieht wie eine Kugel, heißt noch nicht, dass es sich auch wie eine Kugel aus einem bestimmten Material verhält.

≫ Markiere also im HIERARCHY-Fenster (links) den Eintrag SPHERE. Dann suche im INSPECTOR-Fenster (rechts) nach einem Button mit der Aufschrift ADD COMPONENT und klicke darauf.



STICHWORTVERZEICHNIS

C

D

E

F

L

M

N

O

T

U

V

W

X

Y

Z