

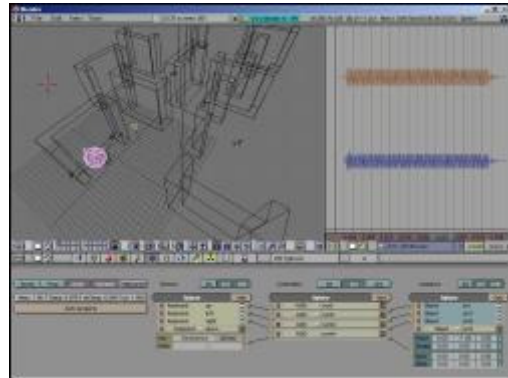


Einführung

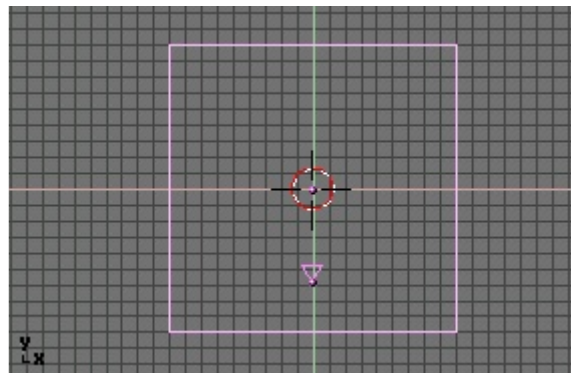
Dieses Tutorial erklärt die grundlegenden Konzepte von Blender 2.1 [bis 2.25]. Es setzt grundlegendes Wissen von der Benutzung von Blender voraus - das hat sich seit Blender 1.8 nicht sehr geändert, und die alten Tutorials sind immer noch anwendbar (obwohl vielleicht einige Buttons an eine andere Stelle verschoben wurden).

Wenn ihr euch nicht mit Blender auskennt, schaut euch bitte zuerst diese Tutorials an:

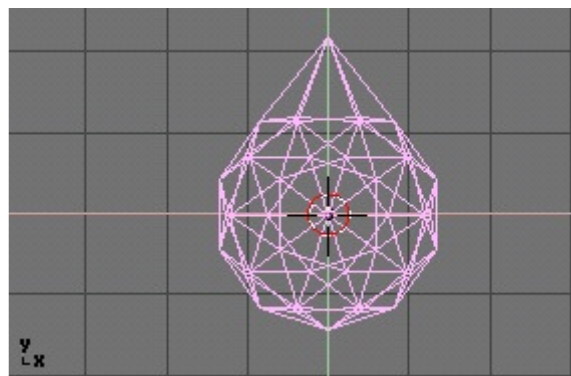
- Benutzeroberfläche
- Navigation
- Die Blender-Fenster
- Ein Schloss bauen



Zuerst machen wir einen Würfel. Löscht die Standardfläche und fügt einen Würfel ein. Vergrößert ihn so, dass er den Bildschirm ausfüllt. Wir verwenden den Würfel als Boden und Wände für unsere Szene.



Jetzt brauchen wir noch einen Spieler. In der Oberansicht fügt ihr eine "Icosphere" mit "Subdivision" 2 ein (Je mehr Unterteilungen, desto mehr Dreiecke hat euer Spieler und desto langsamer kann euer Spiel werden!). Im Editmode zieht ihr eine Nase heraus. Das wird uns später zeigen, in welche Richtung sich unser Spieler bewegt. Sich mit einer perfekten Kugel zu bewegen kann ziemlich verwirrend sein.



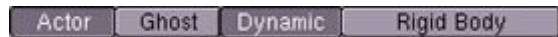
Drückt **P** um die Echtzeitsimulation zu starten. Der Spieler sitzt nur da - nichts passiert! Kein Wunder, denn wir haben noch keine Physik oder Aktionen eingestellt. Das tun wir jetzt.

Dynamik

Stoppt die Simulation mit **Esc**. Wählt euren Spieler aus und geht zu den Realtime Buttons (drückt **F4** [ab Blender 2.3?]). Ganz links seht ihr einen Button mit Namen "Actor".

Ein Spieler ist nicht notwendigerweise der Hauptcharakter eines Spiels. Ein Beispiel für einen Spieler kann ein Feind sein, der euch im Spiel nachläuft.

Für euren Spieler klickt ihr "Actor" an. Wenn ihr das Spiel jetzt starten würdet, würde immer noch nichts passieren. Deshalb schalten wir jetzt die Physiksimulation für unser Objekt ein: der Spieler muss noch angewählt sein, wenn ihr "Dynamic" anklickt. ("Dynamic" erscheint nur, wenn "Actor" aktiviert ist)

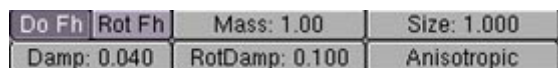


Dreht eure Ansicht etwas (die Oberansicht zeigt den Effekt nicht gut) und drückt **P**. Euer Spieler fällt auf den Boden.

Kräfte und Materialien

Jetzt arbeiten wir etwas mehr mit der Physik. Zuerst einmal, es gibt einen globalen Wert für Gravitation im WorldButtons Fenster. Erstellt eine neue Welt; ein Regler mit der Bezeichnung "Grav" erscheint. Das ist eine Konstante in unserer Welt; alle dynamischen Objekte werden davon beeinflusst. Der Standardwert ist 9.8, das entspricht der natürlichen Erdanziehung.

Zurück zu den Realtime Buttons. Für jedes Objekt könnt ihr eine Anzahl von Einstellungen vornehmen.



1) Mass.

Wenn eine Kraft wirkt, bestimmt die Masse die Beschleunigung, die das Objekt abkriegt. Höhere Masse heißt geringere Beschleunigung.

Wenn ihr große und kleine Objekte in eurem Spiel habt, die miteinander interagieren, solltet ihr aufpassen, dass die Massenverhältnisse stimmen - damit erreicht ihr mehr Realismus in eurem Spiel.

2) Damp.

Bewegungsämpfung. Das ist eine Reibungskraft, die immer auf euer Objekt wirkt - wenn es den Boden berührt oder frei ist. Eine gute Art es zu beschreiben wäre "Luftwiderstand".

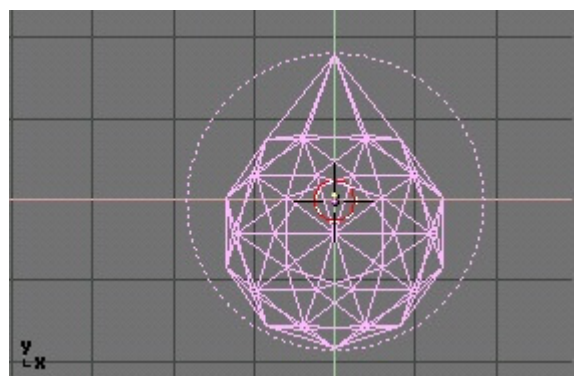
Wenn ein Objekt einen Dämpfungswert hat, wird es immer langsamer werden, falls keine anderen Kräfte darauf wirken.

3) RotDamp.

Drehungsämpfung. Wie die Bewegungsämpfung, aber für die Geschwindigkeit der Drehung.

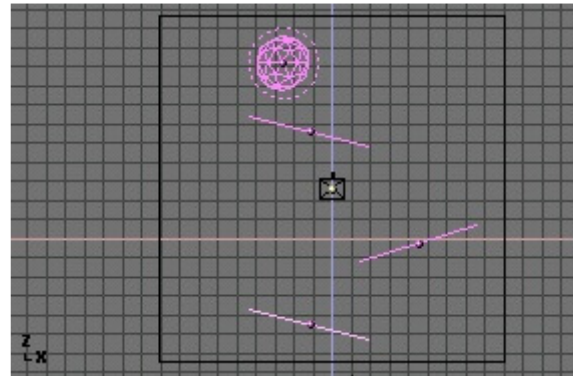
4) Size.

Die aktuelle Physikengine von Blender benutzt Kugeln, um eure dynamischen Objekte zu simulieren. Dieser Wert bestimmt die Größe dieser Kugel. Die Kollisionskugel wird als gepunkteter Kreis dargestellt. Meistens solltet ihr versuchen, euer Objekt ganz in der Kugel einzuschließen. Wenn ihr das nicht tut, können Teile eures Objekts durch Wände und andere Objekte durchgehen.



Stellt die Größe für euren Spieler ein. Die Kugel wird ihr Zentrum im Objektzentrum haben. Um das Objekt relativ zum Objektzentrum zu bewegen, wählt ihr alle Vertices im Editmode und bewegt sie.

Jetzt basteln wir noch etwas am Physiksimulator, bevor es weitergeht. In der Oberansicht fügt ihr eine Fläche ein und wechselt dann in die Vorderansicht. Macht ein paar Duplikate und rotiert und bewegt sie etwas, bis ihr in etwa so eine Situation habt:



Drückt **P** und schaut eurem Spieler beim Fallen zu. Etwas stimmt noch nicht: wenn er eine Fläche berührt, bleibt er daran hängen und beginnt daran herunterzurutschen. Um das zu beheben, werden wir die dynamischen Materialeigenschaften der Flächen einstellen müssen.

Materialien

Neben Farbeigenschaften hat jedes Material auch dynamische Eigenschaften; damit kann man verschiedene Kräfte einstellen, die die Interaktionen zwischen dynamischen Objekten und den Objekten bestimmen, die dieses Material haben.



Macht ein neues Material für eine der Flächen und klickt auf den "DYN" Button.

Als ersten Test probieren wir den "Restitut" Wert aus. Dieser Wert bestimmt, wie viel von der Bewegungsenergie des Objekts an das Objekt zurückgegeben wird, wenn es anstößt. Mit anderen Worten: dieser Wert bestimmt die Elastizität eures Materials. Setzt ihn jetzt auf eins. Wählt die Fläche direkt unter eurem Spieler und löscht seine Rotation mit **Alt R** um sie flach zu machen. Drückt **P** um das Spiel zu starten.

Ihr erwartet, dass er immer weiter springt, oder nicht? Falsch - erinnert ihr euch noch an die "Damp" Einstellung? Die Dämpfung macht euren Spieler dauernd ein bisschen langsamer. Darum springt sie jedes Mal weniger hoch. Setzt "Damp" auf null. Drückt wieder **P** und schaut zu.

Schaut euch dieses Beispiel an: die Flächen wurden so platziert, dass der Ball ziemlich lange weiter springt.

Download:  [bouncing.zip](#)

Setzt den "Damp" Wert wieder auf 0.04. Als nächstes werden wir "Fh" ausprobieren. Dieser Ausdruck wurde aus der Physik ausgeliehen, wo er eine Kraft in vertikaler Richtung beschreibt. In Blender bedeutet er das Gleiche, aber wir müssen einen Einflussbereich definieren. Dieser Einflussbereich wird von "Fh Dist" kontrolliert. Dieser Wert erzeugt ein Polster um euer Objekt, wo die "Fh" Kraft aktiv ist.

Versucht mal das: setzt "Fh Dist" auf [1.0] und "Fh Force" auf 0.4. Jetzt wählt ihr euren Spieler und geht wieder zu den Realtime Buttons. Um Fh anzuschalten, müsst ihr auf "Do Fh" klicken. Startet dann das Spiel (löscht die Rotation der oberen Ebene wieder). Euer Spieler schwebt jetzt über der Fläche wie auf einem Magnetfeld.

Der Einfluss der Fh Kraft ändert sich mit dem Abstand zum Objekt: er variiert von null bis Fh im Intervall Fh Dist.

Der Fh Dist Wert wird auch für die Kollisionsabfrage benutzt. Sobald ein anderes Objekt das Polster berührt, wird eine Kollision registriert. Außerdem erhält das aufprallende Objekt die Farbe des Objekts, auf das es prallt. So kann man das während des Spiels schon kontrollieren.

Das waren die Grundlagen der Spieleengine. Der nächste Schritt ist die Kontrolle über euren Spieler zu lernen. Das wird erklärt in "Controlling a Dynamic Object"

übersetzt von: pat - pat@psycho3d.de - www.psycho3d.de - 22.05.2003 - überarbeitet 30.04.05