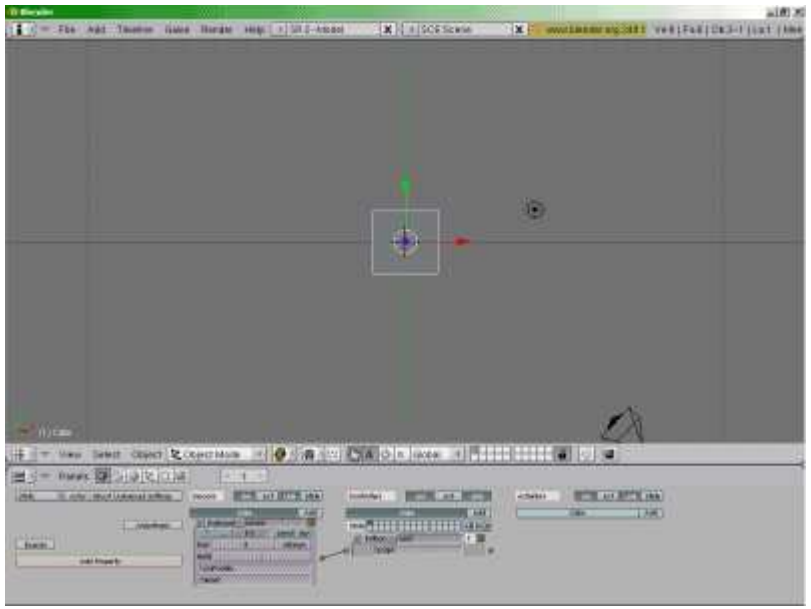
Blender-Tutorial: MP3, Ogg-Vorbis und MIDI in der Game Engine

Stand: 21.01.2010

Blender-Version:	2.49b
Python-Version:	2.6.4
Pygame-Version:	1.9.1
Betriebssystem:	Windows XP SP3
Autor:	Torsten Funk
Website:	www.torsten-funk.de
Download:	tutorial.zip
PDF-Version:	tutorial.pdf



Hinweis:
Bilder können durch Anklicken vergrößert werden!

Ziel dieses Tutorials
Abspielen von MP3-, Ogg-Vorbis- und MIDI-Dateien in der Game Engine mit *Pygame* und Veröffentlichen eines Spiels incl. aller Dateien.

Vorbemerkungen
Für dieses Tutorial wird folgendes angenommen:

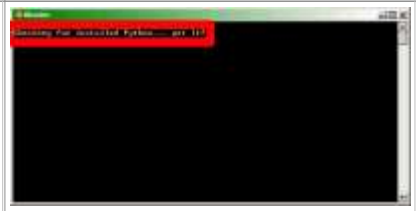
- Blender ist installiert
- Python ist **nicht** installiert
- Pygame ist **nicht** installiert

Hier ein paar Sound-Dateien zum testen:

- beispiel.mp3
- beispiel.ogg
- beispiel.mid

Arbeitsschritte	***Screenshot***

01 - Python installieren



Falls Python noch nicht installiert ist, hole dies nun nach. Eine ausführliche Anleitung dazu findest Du bei Wikibooks.
Wichtig ist hier das setzen der Umgebungsvariable!

Für dieses Tutorial wurde folgende Datei installiert:
python-2.6.4.msi

Weitere Downloads findest Du hier:

http://www.python.org/download/

Ob Blender richtig erkennt, ob und wo Python installiert ist, siehst Du im DOS-Fenster (Konsole) von Blender, direkt nach einem Neustart von Blender.

02 - Pygame installieren

Falls Pygame noch nicht installiert ist, hole dies nun nach. Eine ausführliche Anleitung dazu findest Du auf pygame.org.

Für dieses Tutorial wurde folgende Datei installiert:
pygame-1.9.1.win32-py2.6.msi

Weitere Downloads findest Du hier:
http://www.pygame.org/download.shtml

Pygame fragt bei der Installation nach dem Installations-Pfad von Python, um sich darüber zu installieren.

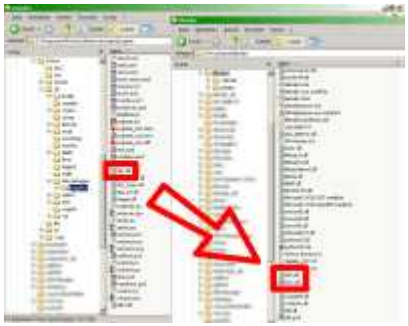
Ob Blender Pygame findet, zeigt sich erst in Schritt 21.

03 - SDL.dll kopieren

Hinweis: Dieser Schritt war bei mir nicht nötig! Er stammt aus einer alten Version des Tutorials unter Blender 2.48a, Python 2.5.2 und Pygame 1.8.1. Falls Du aber die Sound-Wiedergabe nicht zum laufen bekommst, solltest Du diesen Schritt probieren.

Im Python-Unterordner **\Lib\site-packages\Pygame** befindet sich die Datei **SDL.dll**. Kopiere sie in den Ordner mit der Blender-Installation.

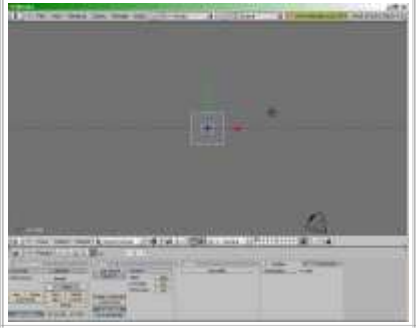
Sicherheitshalber kannst Du die bisherige Datei umbenennen, z.B. in **SDL.old** - so bleibt sie Dir für alle Fälle erhalten.



04 - Blender starten

Starte Blender neu, so dass Du mit einer 'frischen' Datei anfängst.

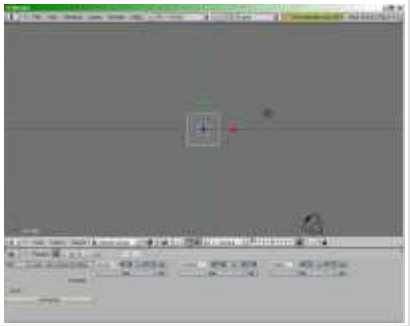
Es existiert bereits ein selektiertes Würfel-Objekt (Cube).



05 - Logic-Buttons

Wechsel mit **F4** in die Logic-Buttons.

Unten in den Buttons werden wir nun die Spielelogik definieren.



06 - Sensor aktivieren

Blender muss nun wissen, welches Signal verarbeitet werden soll, bzw. woher das Signal stammt, welches das Abspielen der Sound-Datei startet. Dazu wird ein 'Sensor' benötigt.

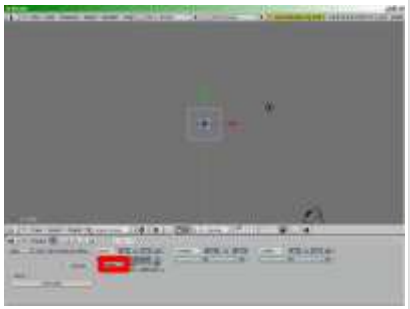
Füge mit einem Klick auf **Add** im Bereich **Sensors** einen Controller hinzu.

Hinweis: Die Logik, die Du nun definierst, wird hier als Beispiel an das vorhandene Objekt *Cube* gehangen.

07 - Sensor definieren

Später soll in der Game Engine eine Sound-Datei abgespielt werden. Zum Auslösen des Abspielens gibt es mehrere Möglichkeiten - gewählt wurde hier als Beispiel das Drücken der Taste *S* auf der Tastatur. Daher muss als erstes definiert werden, dass Blender auf Tastendrücke reagieren soll.

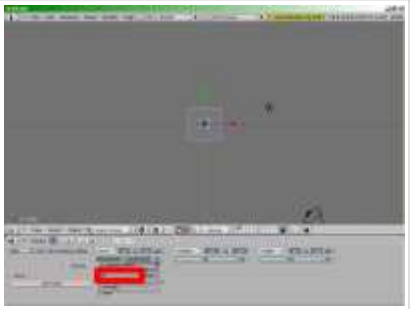
Klicke dazu auf **Always**. In der aufklappenden Liste wähle **Keyboard**.



08 - Taste definieren

Nun wird definiert, welche Taste auslösen soll.

Klicke in das Feld **Key** und anschließend die Taste **S**.



09 - Controller aktivieren

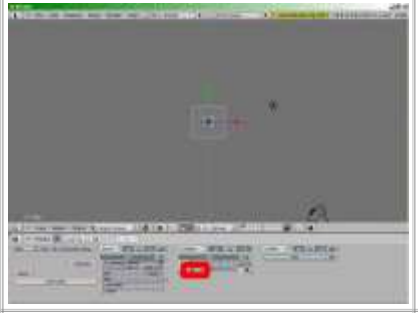
Blender muss nun wissen, womit der Tastendruck verarbeitet werden soll. Dazu wird ein Controller benötigt.

Füge mit einem Klick auf **Add** im Bereich **Controllers** einen Controller hinzu.

10 - Controller definieren

Mit einem Klick auf **AND** können wir den voreingestellten Controller ändern in **Python**. Damit weiß Blender, dass der Tastendruck durch ein Python-Skript verarbeitet werden soll.

Hinweis: Wenn wir später das Skript geschrieben haben, müssen wir hier im Feld *Script* den Skript-Namen eintragen.



11 - Sensor und Controller verlinken

In den Logic-Bricks kann es gleichzeitig mehrere Sensoren, Controller und Aktuatoren geben. Daher ist es nötig, Verbindungen zwischen ihnen herzustellen, damit z.B. definiert ist, welcher Sensor welchen Controller auslöst und an welchen Aktuator dieser sein Ergebnis schickt.

Verbinde dazu die zwei kleinen Knöpfe zwischen Sensor und Controller.

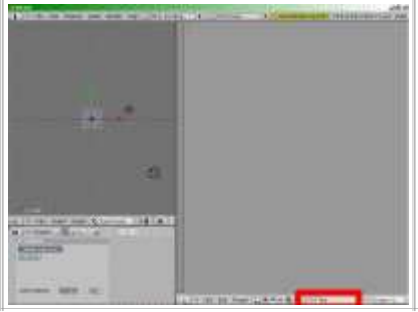
12 - Ansicht wechseln

Da nun das Skript geschrieben werden soll, empfiehlt es sich, in die Scripting-Ansicht zu wechseln.

Wähle also oben aus der Screen-Liste den Eintrag **Scripting**.

13 - Skript umbenennen

Der Text-Editor enthält bereits einen Text namens 'Text'. Benenne diesen in **Test** um.

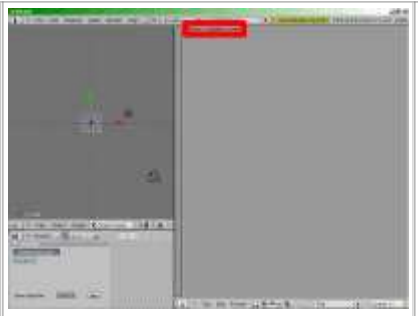


14 - Skript verfassen: Pygame-Mixer importieren

Füge dem Skript folgende Zeile hinzu:

```
import pygame.mixer
```

Damit werden die Pygame-Bibliotheken für den Pygame-Mixer geladen, welche zum Abspielen von Sound-Dateien nötig sind.



15 - Skript verfassen: Pygame-Mixer initialisieren

Füge dem Skript folgende Zeile hinzu:

```
pygame.mixer.init()
```

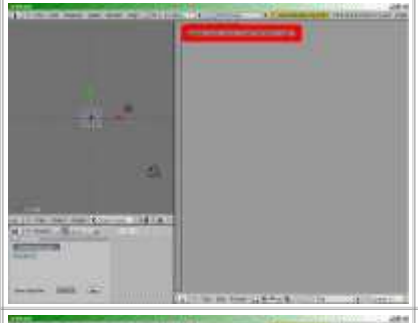
Mit diesem Funktionsaufruf wird der Pygame-Mixer initialisiert.

16 - Skript verfassen: Sound-Datei laden

Füge dem Skript folgende Zeile hinzu:

```
pygame.mixer.music.load("beispiel.ogg")
```

Definiert eine Sound-Datei.

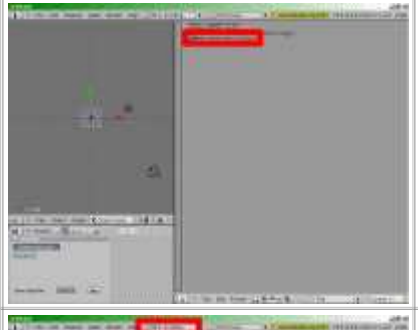


17 - Skript verfassen: Sound-Datei abspielen

Füge dem Skript folgende Zeile hinzu:

```
pygame.mixer.music.play()
```

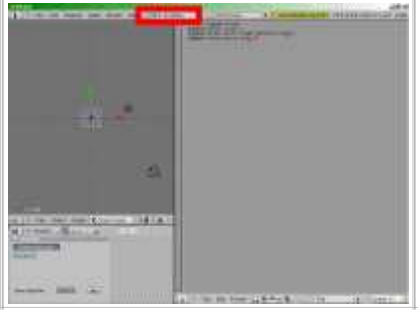
Startet die Wiedergabe der definierten Sound-Datei.



18 - Ansicht wechseln

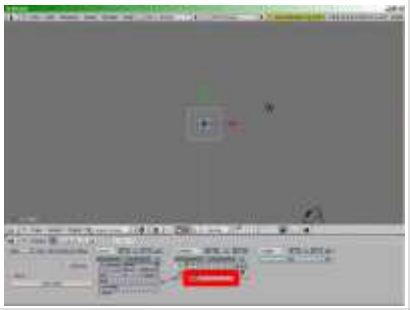
Wie oben erwähnt, muss in den Logic-Bricks der Name des Skripts angegeben werden.

Wähle also oben aus der Screen-Liste den Eintrag **Model**. So kommst Du zurück zu den Logic-Bricks.



19 - Skript-Name im Controller eintragen

Trage im Feld 'Script' den Namen des Skriptes **Test** ein.

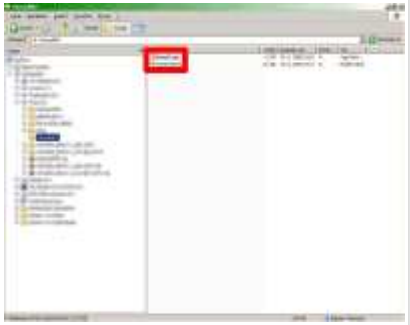


20 - Speicherplatz der Sound-Datei

Nun bleibt noch die Frage, wo Blender bzw. das Python-Skript nach der Sound-Datei sucht.

Solange die Blender-Datei noch nicht gespeichert wurde, wird die Sound-Datei im Installations-Ordner von Blender gesucht. Wird sie dort nicht gefunden, gibt es entsprechende Fehlermeldungen in der Konsole.

Speicher die Blender-Datei in einem Ordner, schließe Blender, kopiere eine Sound-Datei hinein und öffne anschließend die Blender-Datei neu. Es wird dann relativ von der Blender-Datei aus nach der Sound-Datei gesucht.



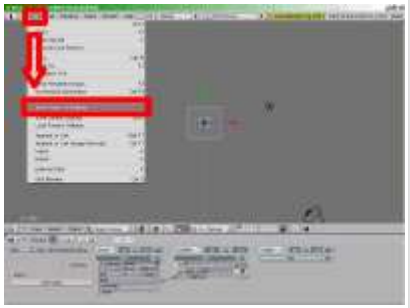
21 - Vorerst Fertig!

Wenn Du nun die Game Engine aktivierst und die Taste 'S' drückst, wird die Sound-Datei abgespielt.

22 - Als Runtime speichern

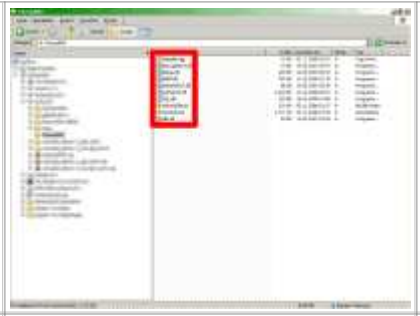
Wenn Du das Spiel veröffentlichen willst, damit auch andere Dein Werk anschauen können, muss Deine Blender-Datei als ausführbare Runtime (exe-Datei) gespeichert und alle benötigten Dateien dazugepackt werden.

Wähle dazu im File-Menü den Eintrag **Save Game As Runtime...**. Speicher die Datei z.B. als **tutorial.exe** in dem Ordner ab, in dem auch die Blender-Datei und die Sound-Datei liegen.



23 - Dateien anschauen

Wie Du nun siehst, hat Blender nicht nur die exe-Datei erstellt, sondern auch auch ein paar dll-Dateien dazugepackt.



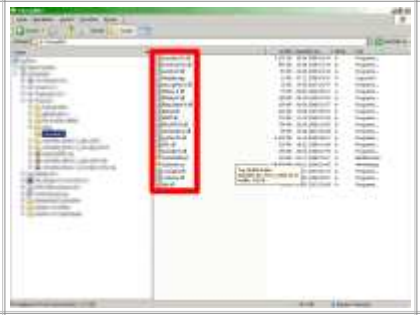
24 - Runtime testen

Wenn Du nun die Runtime startest, erscheinen Fehlermeldungen, dass noch weitere Dateien zum Ausführen benötigt werden.

25 - Benötigte Dateien kopieren

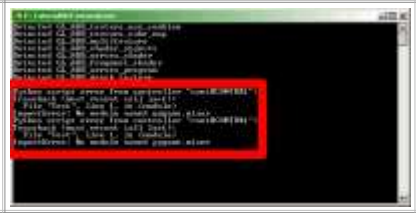
Kopiere folgende Dateien aus dem Installations-Ordner von Blender in den Ordner mit der Runtime:

- python26.dll
- vcomp90.dll



26 - Runtime testen

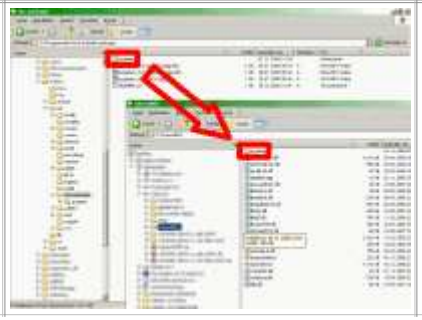
Wenn Du nun die Runtime startest, erscheint in der Konsole eine Fehlermeldung. Diese verrät uns, dass die Runtime die Pygame-Dateien nicht finden kann, denn diese werden nun relativ vom Speicherort der Runtime gesucht.



27 - Pygame-Ordner kopieren

Im Installations-Ordner von Python befindet sich im Ordner **\Lib\site-packages** der Unterordner **pygame**. Kopiere ihn komplett als Unterordner in den Ordner, in dem Deine Blender-Datei zum Tutorial liegt.

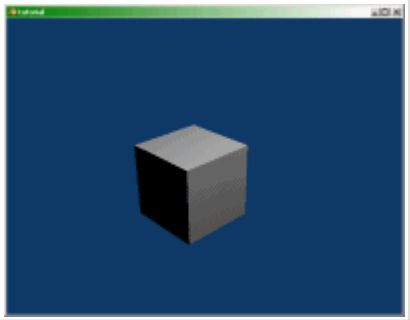
Hinweis: Die meisten Dateien im Unterordner *Pygame* werden zum Abspielen von Sound-Dateien nicht benötigt - z.B. kannst Du die Unterordner *docs* und *excamples* löschen.



28 - Fertig!

Wenn Du nun die Datei **Tutorial.exe** ausführst und die Taste **S** drückst, wird die Sound-Datei abgespielt.

Den kompletten Ordner kannst Du nun packen und veröffentlichen. Eine lokale Blender-, Python- oder Pygame-Installation ist zum Ausführen nicht erforderlich.



Lob & Tadel bitte über das Kontakt-Formular auf www.torsten-funk.de.