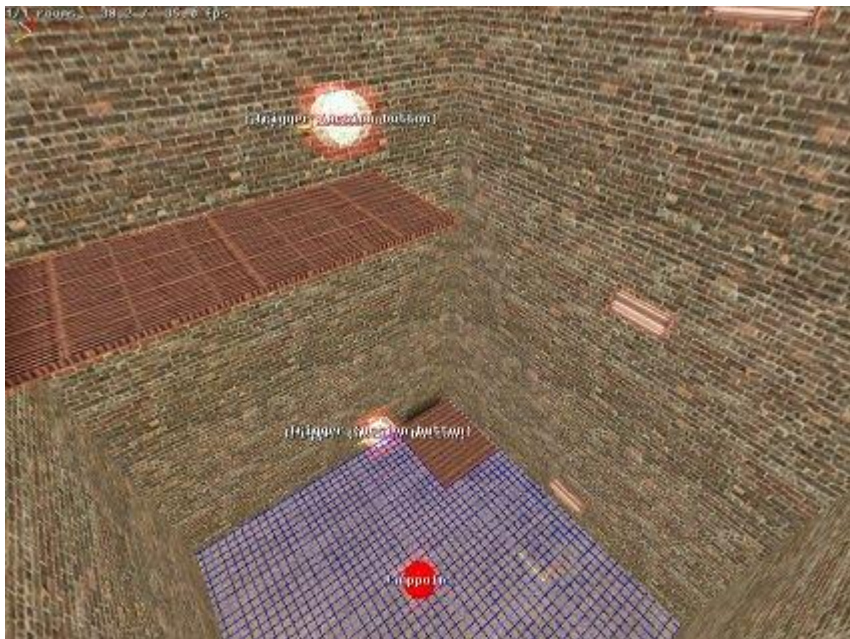


Aufzug - 2 Schalter

by Alexander Pippan aka RnB *rrr*bbb* <alexander.pippan@rnbcreative.at>, 13/08/01

Grundaufbau

Macht zunächst einen funktionierenden Raum der über 2 Stockwerke geht. Plaziert nun im oberen Stockwerk eine Plattform. Macht jetzt euren Aufzug dorthin, wo er starten soll. Benennt ihn in **Aufzug** um. Jetzt macht an beide Haltestellen des Aufzuges eine Art Console hin. Plaziert vor die Consolen jeweils 2 **action_button_trigger** dicht neben einander und benennt die unteren 2 in **Trigger_01**, **Trigger_01_A** und die oberen 2 in **Trigger_02**, **Trigger_02_A** um.



Animationen

Macht jetzt euren Aufzug zu einem dynamischen Objekt (Transform-Mode + d). Macht jeweils eine Keyframe am Start und am Endpunkt. Nennt diese in **start** und **end** um. Erstellt jetzt 2 Animationen. Eine zwischen 'start' und 'end' und eine zwischen 'end' und 'start'. Benennt diese jetzt in **up** and **down** um.

Messages

Jetzt der etwas kompliziertere Teil. Zuerst einmal warum wir jeweils 2 Trigger gesetzt haben. Und zwar: Sollte der Spieler während der Fahrt vom Aufzug fallen etc. oder nicht hinauf kommen, muss er den Lift auch wieder herunter fahren. Da wir jetzt bei dem ersten Trigger nur die **Up**-Animation aktivieren können ist der 2.Trigger notwendig.

So zuerst schreibt bei allen Triggern außer dem **Trigger_01** das in die

Startup-Kategorie: **this->T_Enable(false)**; Damit sind jetzt alle Trigger bis auf den einen zu Beginn inaktiv. Jetzt müssen wir über die Trigger die richtige Animation auslösen. Dazu bei der Kategorie **T_Activate** in der 1.Spalte:

- Trigger_01: **parent::aufzug->DO_Animate(up)**;
- Trigger_01_A: **parent::aufzug->DO_Animate(down)**;
- Trigger_02: **parent::aufzug->DO_Animate(down)**;
- Trigger_02_A: **parent::aufzug->Do_Animate(up)**;

So das aktiviert jetzt die richtigen Animationen. Die Steuerung wann der jeweilige Trigger aktiviert ist erfolgt über die Animationen. Dazu sehen wir uns die Messages unseres Aufzuges an. Geht hier unter die Kategorie **Animation(up)**. Fügt hier in der 1.Spalte folgendes ein:

- **parent::Trigger_01->T_Enable(false)**;
- **parent::Trigger_02_A->T_Enable(false)**;

Diese 2 Messages jetzt in der 3.Spalte:

- **parent::trigger_01_A->T_Enable(true)**;
- **parent::Trigger_02->T_Enable(true)**;

So aktiviert man jetzt den Trigger_01 damit der Aufzug nach oben fährt, deaktiviert er sich sofort. Gleichzeitig wird der Trigger_02_A, der zum hochholen des Aufzuges deaktiviert. Ist der Aufzug jetzt oben angekommen, aktiviert er den Trigger_02 mit dem man wieder hinunter kommt und den Trigger_01_A, der sollte man unten geblieben sein, dazu dient den Aufzug wieder herunter zu fahren.

Das gleiche nur umgekehrt machen wir jetzt bei der Kategorie **Animation(down)**. Hier in der ersten Spalte:

- **parent::Trigger_02->T_Enable(false)**;
- **parent::trigger_01_A->T_Enable(false)**;

Jetzt noch in der 3.Spalte:

- **parent::Trigger_01->T_Enable(true)**;
- **parent::Trigger_02_A->T_Enable(true)**;

Das Ganze scheint jetzt zwar ein wenig kompliziert, allerdings überlegt man jetzt einmal was man genau gemacht hat scheint es doch recht einfach.

Nachwort

Jetzt wisst ihr wie man einen einfachen Aufzug mit 2 Schaltern baut. Achtung: Solltet ihr unter den Aufzug gelangen wird der Aufzug nicht zurück fahren. Das könnt ihr allerdings noch selber hinzufügen.

Wer jetzt noch während der Fahrt einen einfachen Sound haben will, der soll

bei der Kategorie **Animation(up)** und **Animation(down)** jeweils in der ersten Spalte das :

this->A_Play3DSound(dynamic, elevator_running_loop, "");

und in der 3.Spalte das:

this->A_StopAll3DSounds("");

eingeben.

© 2001 *rrr*bbb*, MPM