

Spieler Inventar

by Alexander Pippa aka RnB *rrr*bbb* <alexander.pippa@rnbcreative.at>, 11/09/01

Aufbau

Als Beispiel wollen wir einen Schlüssel aufheben, einen Trigger auf **aktiv** stellen, den Schlüssel im Inventar (rechts oben im HUD) anzeigen und bei aktivieren des Triggers wieder aus dem Inventar streichen. Zum Aufbau: Wieder ein Raum (StartRoom), mit Jumpoint. Jetzt noch einen Trigger (Trigger_01) der zu Beginn deaktiviert ist (**this->t_enable(false);**) sowie einen einfachen Block (Key) an die Wand der unseren Schlüssel darstellen soll.



Einstellungen beim Key

Um den Key später zu verstecken macht ihn zu einem dynamischen Objekt (F5+d). Schaltet sämtliche Kollisionen und **Cast no Shadows** ein. Soviel zum Schlüssel selbst...

... Jetzt plaziert ihr vor den Key einen **action_button_trigger** (Trigger_02).

Command `maxpayne_hudmode->MPHM_PrintInventory(%s, %i);`

Der ganze Command wird verwendet um Inventargegenstände im HUD anzuzeigen.

- **%s**: Steht für den Text der aus der Stringtable.txt-Datei entnommen wird. Mehr dazu weiter unten.
- **%i**: Gibt den Slot an, indem der Gegenstand angezeigt werden soll. Dabei gibt es die Slots 1,2 und 3.

Den Key dem Inventar hinzufügen

So dazu gehen wir bei den Messages zur T_Activate-Kategorie 1.Spalte. Zuerst einmal den Key verstecken: **parent::key->DO_Hide(true);** Jetzt wollen wir unseren eigenen Gegenstand anzeigen. Dazu öffnet die **stringtable.txt** im **data/database**-Verzeichnis. Scrollt hinunter bis zu diesem Abschnitt. Befindet sich in der Mitte.

[QuickLoadQueryText]	String = "Press F9 again to Quick
[STRING_CARRYING]	String = "";
[STRING_BAR_KEY]	String = "Bar Key";
[STRING_CELLAR_KEY]	String = "Cellar Key";
[STRING_STAIRCASE_KEY]	String = "Staircase Key";

Z.B. **STRING_BAR_KEY** währe beim Command dann das %s. Der Text der dann rechts in Anführungszeichen steht, wird dann im Inventar angezeigt. Jetzt lasst uns einen eigenen String einfügen. Sieht dann so aus:

[QuickLoadQueryText]	String = "Press F9 again to Qui
[STRING_CARRYING]	String = "";
[STRING_BAR_KEY]	String = "Bar Key";
[STRING_CELLAR_KEY]	String = "Cellar Key";
[STRING_STAIRCASE_KEY]	String = "Staircase Key";
[STRING_TUTORIAL_KEY]	String = "Unser Tutorial Key";

So das würde jetzt **Unser Tutorial Key** anzeigen. Lasst uns jetzt den Command beim action_button_trigger hinzufügen:
maxpayne_hudmode->MPHM_PrintInventory(STRING_TUTORIAL_KEY, 1);
Das wird den Key jetzt beim aufheben mit der USE-Taste ins Inventar bei Slot1 einfügen. Fügt jetzt noch diese Message ein um den Trigger_01 zu aktivieren:

parent::trigger_01->t_enable(true);

Den Key wieder löschen

Um den Key beim aktivieren des Trigger_01 wieder zu löschen muss man beim gleichen Slot einen leeren String senden. Solltet ihr mehrere Slots besetzt haben rücken die Hinteren Gegenstände vor. Sendet jetzt einen leeren String.

[QuickLoadQueryText]	String = "Press F9 again to Quick
[STRING_CARRYING]	String = "";
[STRING_BAR_KEY]	String = "Bar Key";
[STRING_CELLAR_KEY]	String = "Cellar Key";
[STRING_STAIRCASE_KEY]	String = "Staircase Key";
[STRING_TUTORIAL_KEY]	String = "Unser Tutorial Key";

Also das 1.Spalte bei Trigger_01 einfügen:
maxpayne_hudmode->MPHM_PrintInventory(STRING_CARRYING, 1);
Den Trigger mit
this->t_enable(false); wieder deaktivieren.

Nachwort

Um das ganze zu verfeinern kann man jetzt die richtigen Sounds noch einfügen und beim Aktivieren des Trigger_01 eine Tür öffnen die vorhin noch geschlossen war. Sounds wie "There was a key on the table" befinden sich im Verzeichnis **data\database\sounds\characters\max**