

Schritte um ein Starterpack auf AGS3 umzubauen

Um eine Konsistenz (Einheitlichkeit) über alle Starterpacks zu gewährleisten, ist es ratsam alle Starterpacks mit der Version 3.0.2-SP1 aufzubauen (AGS Editor .NET (Build 3.0.2.44)).

1. Organisieren des leeren Starterpacks für AGS3
2. konvertieren eines vorhandenen Starterpacks in eine Version für AGS2.72, falls es mit einer früheren Version erstellt wurde. Sichern dieser Version und öffnen der konvertierten mit AGS 3 (3.0.2.44)
3. Speichern in AGS3 und Speichern aller Grafiken, die für das Starterpack typisch sind als PNG (es können auch bmp, gif oder tif verwendet werden)
[Auf gar keinen Fall dürfen die Bilder als jpg gespeichert werden]
Es empfiehlt sich die Bilder in logischen Unterordnern zu speichern

Ordnerstruktur

- Objekte werden also praktischerweise mit dem Prefix o gespeichert und Inventargegenstände mit dem prefix i.
4. Öffnen einer neuen Instanz von AGS3 und öffnen des heruntergeladenen Starterpacks. Erstellen von logischen (Unter-)Ordner und einfügen der vorher gesicherten Bilder.
 5. Jetzt können die Räume kopiert und importiert werden. (Ein Raum nach dem anderen)

```
Starterpack
|- Räume
|- Objekte
|  |- oObjektname.png
|
|- Inventar
|  |- iInventargegenstand.png
```

Ab jetzt fängt die eigentliche Arbeit an. Es müssen nun alle Räume nachbearbeitet werden und die Syntax angepasst werden.

6. Jetzt wird der erste importierte Raum bearbeitet. Als erstes werden die Namen der Hotspots und der Objekte aufgeschrieben und notiert. Und danach werden alle Verknüpfungen auf AnyKlick gelöscht. Jetzt werden die Namen der Hotspots (hxxx) und Objekte (oxxx) benannt und dann die Events wieder hinzugefügt (z.B. oFernbedienung, oFernbedienung_AnyKlick()). Als nächstes werden im Skript-Teil oben ein paar Leerezeilen eingefügt und danach die Vorlage zur Aufteilung der Raum- und Hotspot bzw. Objektfunktionen eingefügt.
7. Als nächstes kommt die Vorlage für die Aktionen dran. Jetzt müssen nur noch die ganzen Texte geändert, bzw. neue eingefügt werden. Pro Raum habe ich ca. eine dreiviertel Stunde bis Stunde eingeplant und auch gebraucht.

8. Beispiel:

```
// Hausflur unten (Familie Pantz)

/*****
*   Raum-Funktionen
*****/

/*****
*   Tür-Hotspot-Funktionen
*****/

/*****
*   Objekt-Funktionen
*****/
```

```
/*  
*   Hotspot-Funktionen  
*/
```

```
/*  
*   Regionen-Funktionen  
*/
```

9. Beispiel:

```
if (any_click_move(118, 133, eDirLeft))  
{  
    if (UsedAction(eActWalkTo))  
    {  
        //   Wait(5);  
    }  
    else if (UsedAction(eActLookAt))  
    {  
        Wait(5);  
        player.Say("Das ist der Lieblingstедdy von Marcy.");  
    }  
    else if (UsedAction(eActPickUp))  
    {  
        Wait(5);  
        player.Say("Vielleicht sollte ich ihn einfach nehmen.");  
    }  
    else if (UsedAction(eActUse))  
    {  
        Wait(5);  
        player.Say("Vielleicht sollte ich ihn einfach nehmen.");  
    }  
    else if (UsedAction(eActOpen))  
    {  
        Wait(5);  
        player.Say("Und da ich momentan keine davon brauche, ...");  
    }  
    else if (UsedAction(eActClose))  
    {  
        Wait(5);  
        player.Say("Der Teddy ist sauber vernaft.");  
    }  
    else if (UsedAction(eActPull))  
    {  
        Wait(5);  
        player.Say("Vielleicht sollte ich ihn einfach nehmen.");  
    }  
    else if (UsedAction(eActPush))  
    {  
        Wait(5);  
    }  
}
```

```
        player.Say("Damit er umfällt?");  
    }  
    else Unhandled();  
}
```

10. Jetzt sind noch die Properties an der Reihe. In jedem Raum muss ein neues Property „gi“ auf den letzten Wert eines Türenobjektes. (z.B. Wenn im Raum drei Türen sind und die Objekte die Nummern 0, 1 und 2 haben, dann muss gi den Wert 2 haben.)
11. Bei den Tür-Objekten muss jetzt nur noch die GlobalInt-Nummer der Tür eingetragen werden.
12. Jetzt noch Icons mit transparentem Hintergrund und dem Gesicht des Hauptcharakters erstellen (USER.ICO, SETUP.ICO) (mit Gimp)
13. -FERTIG-