Symfonia – Migration

Das Nachschlagewerk zum Upgrade auf Version 2

by totalwarANGEL

Inhaltsverzeichnis

1 Einführung

Mit dem Sprung auf Version 2 habe ich die Gelegenheit genutzt Dinge, die nicht optimal waren, zu verbessern. Des weiteren gibt es neue Features. Dieses Dokument ist eine Checkliste für ein sauberes Upgrade auf Version 2.

2 Änderungen an bestehenden Features

2.1 Entity Scripting Values

Die Entity Scripting Values wurden in ein neues Konzept überführt. Sie sind nicht mehr Hauptbestandteil des Bundles, sondern Mittel zum Zweck. Das neu entstandene Entity Properties Bundle kann die wichtigsten Eigenschaften von Entities abfragen und ändern. Dabei wird ein Wrapper erstellt, der seine Gültigkeit behält.

```
Hakim = QSB.EntityProperty:New("hakim");
Hakim:Hurt(50, true);
```

Über die Variable Hakim kann zu einem späteren Zeitpunkt auf weitere Methoden zugegriffen werden. Die alten API-Funktionen sind als Fallback vorhanden. Für eine genaue Liste siehe die Dokumentation.

2.2 Time Line

Die API-Funktionen der TimeLine wurden aus der Dokumentation entfernt. Sie haben ohne jegliche Prüfung nur die Methoden der TimeLine-Klasse aufgerufen. Zusammen mit den doppelten Kommentaren unnötiger Boilerplate Code.

Aus `API.TimeLineXXX` wird `QSB.TimeLine:XXX`. Alle Argumente bleiben gleich. Die alten API-Funktionen sind als Fallback vorhanden.

Für weitere Informationen siehe die Dokumentation ein.

2.3 Funktionsreferenzen in Custom Behavior

Die MSF-Behavior wurden überarbeitet. Ihre Originalfunktion bleibt erhalten. Im Skript bekommt man die Möglichkeit eine Funktionsreferenz statt einem String anzugeben. Alle weiteren Argumente werden zu Argumenten der referenzierten Funktion.

```
Reward_MapScriptFunction(ReplaceEntity, "hakim", Entities.U_KnightWisdom);
```

2.4 Überarbeitete Briefings

Briefings und Cutscenes wurden voneinander getrennt. Ein Briefing hat permanent breite Bars, welche aber transparent gemacht werden können.

Cutscenes werden jetzt über CS-Dateien erzeugt, welche in die Map kopiert werden müssen. Dies ist weniger Performancelastig und ermöglicht bessere Kameraflüge.

Pages von Briefings haben jetzt automatisch eine endlose Duration, wenn diese nicht angegeben wird. Eine Ausnahme bilden Pages mit Multiple Choice. Ihre Duration ist immer endlos.

Für eine Kameraanimation wird nur noch eine Seite benötigt. Die Page hat normale Positionsangaben und zusätzlich eine Subtable `FlyTo`. Hat die Page keine Duration oder ist die Duration kleiner, wird die Animationsdauer als Duration gesetzt.

```
AP {
    ...
    FlyTo = {
        Position = "destination",
        Angle    = 23.5,
        Rotation = -99,
        Zoom     = 5000,
        Duration = 20.0
    },
}
```

Pages besitzen nun auch einen „Zurück“-Button. Zusätzlich heißt der „Überspringen“-Button jetzt „Weiter“. Auf der letzten erreichbaren Seite des Briefings heißt der Button „Beenden“. Entsprechende englische Texte werden automatisch verwendet. Zusätzlich gibt es die Methoden `OnForward` (Weiter) und `OnReturn` (Zurück), die beim Klick ausgelöst werden.

Die Syntax für Multiple Choice wurde vereinfacht. Anstatt mehrfach verschachtelter Tables kann die Liste der Antworten jetzt direkt übergeben werden.

```
AP {
    ...
    MC = {
        {"Antwort 1", 5},
        {"Antwort 2", "IDOfPage8"},
    },
}
```

Zusätzlich können Pages jetzt mit dem Feld `Name` einen Namen erhalten, den man anstelle des Index für MC und Sprünge verwenden kann.

```
AP {
    ...
    Name = "Bockwurst",
}
```

Für weitere Informationen siehe die Doku von „BundleBriefingSystem“!

2.5 Debug-Kommandos

Die Befehle `winall` und Co. wurden abgeschafft. Die Detailanzeige eines Quests ist entfallen.

Quests können nun nicht nur nach Namensteilen, sondern auch nach Status gesucht werden.

Die Kommandos `gexec` und `lexec` wurden umbenannt in `>` und `>>` und können nur noch Funktionen aufrufen. (Beispiel: F 0 1 → F(0, 1)) Für komplexe Anwendungen LUA-Debugger nutzen.

Analog heißen `gload` und `lload` jetzt `<` bzw `<<`.

3 Neu hinzugekommene Features

3.1 Reward_RandomRequest

Die Zufallsaufgaben haben ihren Weg zurück in die QSB gefunden. Im Gegensatz zum Original werden aber nicht nur Lieferanfragen gestellt.

```
Goal_RandomRequests(_DeliverGoods, _PayGold, _Claim, _KnightTitle, _Reputation,
_Time, _StartText, _SuccessText, _FailureText);
```

Für weitere Informationen, siehe die Dokumentation „AddOnRandomRequests“!

3.2 Erweiterung Trigger_Briefing

`Trigger_Briefing` wurde um einen Parameter erweitert, der den Typ des Briefings festlegt. Es kann entweder auf alle („All“) oder auf einen bestimmten Typ reagiert werden („Success“ oder „Failure“).

Neu ist, Briefings, je nach verwendetem Behavior, als anderer Typ an einen Quest angebunden werden. `Reprisal_Briefing` bindet ein Briefing, dass als „Failure“ gebunden wird. Zusätzlich dazu bindet `Reward_Briefing` ein Briefing als „Success“.

3.3 Nachfolger der Laufschrift

Nachdem es immer wieder gewünscht wurde, verfügt die QSB nun über einen offiziell unterstützten Nachfolger der alten Laufschrift: `API.SimpleTypewriter`.

```
API.SimpleTypewriter(_Text, _Callback)
```

Der Text kann in zwei Sprachen angegeben werden. Der Text ist eine fortlaufende Zeichenkette. Sie wird automatisch zerlegt. Du musst Dir darüber keine Gedanken machen!

Siehe dazu die Dokumentation in „BundleDialogWindows“!

3.4 Wetter-Bundle

Es gibt nun ein Bundle für einfache Wettermanipulation.

Ein einzelnes Event kann erzeugt und abgespielt werden. Es verwendet ein dynamisches Display Set (z.B. me_winter_sequence.xml) und beeinflusst auch Wachstum und andere Dinge.

Solch ein Event kann auch als sich endlos wiederholendes Event eingestellt werden.

In jedem Fall wird ein Event, solange es noch läuft, automatisch nach dem Laden eines Spielstandes wiederhergestellt und mit der verbleibenden Zeit abgespielt.