



AOQML-Updates 2011

Stand: Juli 2011

Dieses Dokument richtet sich an alle AOQML-Schreiber, aber natürlich auch an alle AOQML-Interessierten. Grundlegende AOQML-Kenntnisse sind von Vorteil.



Agenda

1. Wie geht's dem NPC nach einem Kampf?
2. Fertigkeitswerte temporär ändern
3. Questauftreten regulieren
4. Umgang mit Wareninstanzen
5. Werkzeug für jeden Schreiberling: `<var-dump/>`



Wie geht's dem NPC nach einem Kampf?

- verschiedene Infos zu einem NPC lassen sich jetzt abfragen, Infos stehen in Variablen bereit:

rivals:#1:status → „geflohen“, „verloren“, „gewonnen“

rivals:#1:lep → Zahl der Lebenspunkte

rivals:#1:wounds → Anzahl der Wunden

- um NPC-Freund abzufragen, anstatt ‚rivals‘ einfach ‚friends‘ benutzen



Wie geht's dem NPC nach einem Kampf?

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <scene xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:noNamespaceSchemaLocation="http://eisentrutz.antamar.eu/aoqml.xsd">
4 <fight>
5   <rivals>
6     <npc npcid="331"/>
7     <npc npcid="331"/>
8   </rivals>
9   <victory takeDroppedWeapons="true">
10    ...
11 </victory>
12 <escape>
13   ...
14 </escape>
15 <defeat>
16   ...
17 </defeat>
18 </fight>
19
20 </scene>
```

Variablen für den ersten NPC-Gegner heißen
rivals:#1:status
rivals:#1:lep
rivals:#1:wounds

Variablen für den zweiten NPC-Gegner heißen
rivals:#2:status
rivals:#2:lep
rivals:#2:wounds

Variablen haben den Scope 'scene'



Wie geht's dem NPC nach einem Kampf?

```
4 <fight>
5   <rivals>
6     <npc npcid="331"/>
7     <npc npcid="331"/>
8   </rivals>
9   <victory takeDroppedWeapons="true">
10    <switch name="rivals:#1:status">
11      <case val="verloren">
12        ... tot am boden ...
13      </case>
14      <case val="geflohen">
15        ... Ork ist abgehauen ...
16      </case>
17    </switch>
18  </victory>
19  <escape>
20    ...
21  </escape>
22  <defeat>
23    ...
24  </defeat>
25 </fight>
```

In dem Beispiel wird geprüft, ob nach dem Kampf, im Siegesfall des Helden, der Gegner #1 am Boden liegt oder ob er geflohen ist



Wie geht's dem NPC nach einem Kampf?

```
</victory>
<escape>
  ...
</escape>
<defeat>
  <switch name="rivals:#2:lep">
    <case min="30">
      ...
    </case>
  </switch>
  <switch name="rivals:#1:wounds">
    <case val="0">
      Ork hat keine Wunden.
    </case>
  </switch>
</defeat>
</fight>
```

In diesem Beispiel wird geprüft, ob der zweite NPC noch mindestens 30 LEP hat.

Außerdem wird geprüft, ob der erste NPC keine Wunden erlitten hat.



1. Wie geht's dem NPC nach einem Kampf?
2. Fertigkeitswerte temporär ändern
3. Questauftreten regulieren
4. Umgang mit Wareninstanzen
5. Werkzeug für jeden Schreiberling: `<var-dump/>`



Fertigkeitswerte temporär ändern

- FeW lassen sich ändern für eine bestimmte Zeit

```
<set talent="Athletik" inc="3" duration="10 days" />  
<set talent="Sinnenschärfe" dec="5" duration="14 days" />
```

- erhöht Athletik-FeW um 3 für 10 Tage
- verringert Sinnenschärfe-FeW um 5 für 14 Tage



1. Wie geht's dem NPC nach einem Kampf?
2. Fertigkeitswerte temporär ändern
3. **Questauftreten regulieren**
4. Umgang mit Wareninstanzen
5. Werkzeug für jeden Schreiberling: `<var-dump/>`



Questauftreten regulieren

- jede Quest wird mit `<quest status="ended" />` beendet
 - die Angabe vom *frequency*-Attribut reguliert die Wahrscheinlichkeit, wie häufig die Quest beim Held erneut auftreten darf
- z.B. *frequency*="rar" bedeutet, die Wahrscheinlichkeit liegt unter 10%, dass die Quest wieder startet, wo sie normalerweise starten würde



Questauftreten regulieren

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <scene xmlns:xsi="http://www.w3.org/2001/XMLSchema-i
3     xsi:noNamespaceSchemaLocation="http://mihl2.h
4
5 <p>Endlich angekommen, siehst du auch schon die Bere
6 nicht eine junge Frau, die sich weinend aus dem Fens
7 Als du dich dem Haus näherst, bemerkt sie dich.
8 Sie will ins Haus verschwinden und das Fenster schli
9 <p><q>Moment, nicht so schnell. Heißt du Luise?</q>,
10 <p>Sie hält einen Moment inne, schliesst das Fenster
    <q>Das geht Euch nichts an!</q></p>
11 <p>Darauf nimmst du den Brief aus der Tasche, hebst
12 <q>Der hier ist von Mark.</q></p>
13 <p>Hastig stürzt sie zum Fenster: <q>Her damit, endl
14 <p>Du lässt dir den Zettel aus der Hand reißen und b
15 <set attribute="EP" inc="20"/>
16 <quest status="ended" frequency="rar" />
17
18
19 </scene>
```

- durch frequency="rar" wird in der Quest festgelegt, dass die Quest nicht sehr häufig erneut startet



Questauftreten regulieren

- das *withhold*-Attribut ermöglicht eine temporäre Sperre der Quest (für den Held)

```
<p>Hastig stürzt sie zum Fenster: <q>Her damit,  
<p>Du lässt dir den Zettel aus der Hand reißen  
<set attribute="EP" inc="20" />  
<quest status="ended" withhold="100 days" />
```

- `withhold="100 days"` sperrt die Quest für 100 (Ingame-)Tage für den Held



Questauftreten regulieren

- Kombination von *frequency* und *withhold* erscheint sinnvoll

Beispiel:

```
<quest status="ended" withhold="100 days" frequency="rar" />
```

- bewirkt, dass die Quest die nächsten 100 Tage beim Held nicht erneut startet und danach nur mit geringer Wahrscheinlichkeit



1. Wie geht's dem NPC nach einem Kampf?
2. Fertigkeitswerte temporär ändern
3. Questauftreten regulieren
4. Umgang mit Wareninstanzen
5. Werkzeug für jeden Schreiberling: `<var-dump/>`



Umgang mit Wareninstanzen

- jeder Gegenstand ist „einzigartig“ in Antamar
- hat der Held 5 Säbel dabei, ist jeder Säbel unterschiedlich (auch wenn das für den Spieler nicht immer sichtbar/merkbar ist)
- Unterschiede gibt es zum Beispiel in der Qualität
- Qualitätsstufe kann bei der Aufnahme von Gegenständen angegeben werden:

```
<take item="Schwert" quality="75" />
```

Zahl zwischen 0 (ganz mies) und 100 (sehr gut)



Umgang mit Wareninstanzen

- Zugriff auf eine Wareninstanz kann man als AOQML-Schreiber via *item-select* ermöglichen

```
<item-select item="Heiltrank" target="heiltrank-benutzen" />
```

- der Spieler bekommt alle seine Heiltränke, die er dabei hat, angezeigt und kann nun einen davon auswählen (anklicken)

Der Klick führt dann in die Szene *heiltrank-benutzen.xml*

- [Heiltrank](#) (Superior)
- [Heiltrank](#) (gut)
- [Heiltrank](#) (brauchbar)
- [Heiltrank](#) (Murks)



Umgang mit Wareninstanzen

Die selektierte Wareninstanz lässt sich dann manipulieren

- Qualität der Ware auslesen:

```
<get object="selected-item" attribute="quality" />
```

- Qualität der Wareninstanz ändern:

```
<set object="selected-item" attribute="quality" dec="15" />
```

verringert die Qualität um 15 (auf minimal 0 Punkte), analog mit *inc* zum Erhöhen der Qualität

- gewählte Wareninstanz wegwerfen:

```
<drop object="selected-item" />
```



1. Wie geht's dem NPC nach einem Kampf?
2. Fertigkeitswerte temporär ändern
3. Questauftreten regulieren
4. Umgang mit Wareninstanzen
5. Werkzeug für jeden Schreiberling: `<var-dump/>`



var-dump

- `<var-dump/>` hilft bei der Fehlersuche (Debuggen) einer (komplexeren) Queste
- `var-dump` zeigt alle gesetzten Variablen mit ihren Werten an, wenn der Befehl auf dem Testserver ausgeführt wird
- Beispiel auf der nächsten Folie



var-dump

```
<store name="NachSchatzGefragt" scope="quest">true</store>
<store name="StoryErzaehlt" scope="quest">ja</store>

<p>Es wird noch eine scene-Variable gesetzt.</p>
<store name="zahl" scope="scene">0</store>
<set name="zahl" inc="0...100" />

<p>Var-dump wird ausgeführt:</p>

<var-dump />
```

- gesetzte 'quest' und 'scene'-Variablen werden ausgegeben
- Mit `<var-dump hero="true" />` können auch alle Heldvariablen ausgegeben werden

Es wird noch eine scene-Variable gesetzt.

[zahl=25]

Var-dump wird ausgeführt:

Debugging scene-Vars:

zahl : 25

Debugging quest-Vars:

NachSchatzGefragt : true

StoryErzaehlt : ja

sys:locationId : 422

sys:locationType : ort

sys:trace : (running)



Linksammlung

- AOQML-Changelog unter <http://forum.antamar-community.de/viewtopic.php?f=77&t=8624>
- Alle aktuellen Infos zu den Tags im Wiki <http://wiki.antamar.eu>
- Frohes AOQML-Programmieren! :-)