

# ParaWorld Triggering Manual

by AyCe

**Version 1.0**

- 1. Grundlegendes
  - 1.1 Planung
  - 1.2 Das Menü
- 2. Triggering
  - 2.1 Ein Trigger
  - 2.2 Variablen
- 3. Conditions-Liste
- 4. Actions-Liste



# 1. Grundlegendes

## 1.1 Planung

Triggering in Paraworld ist garnicht so schwer. Voraussetzungen sind jedoch umfassende Editor-Kenntnisse, da schlechte Maps auch mit Trigger nicht besser werden. Wenn man das Leveldesign fertig hat, und die Map soweit spielbereit ist, kann man Trigger adden, um besondere Features hinzuzufügen. Doch zuvor sollte man sich einen Plan machen: Was soll getriggert werden? Kann man bestimmte Quests immernoch erfüllen, wenn ein bestimmtes Objekt zerstört ist? Ich schlage vor, das Triggering in „Module“ zu unterteilen. Anhand meiner Map „BfPW – Highland“ werde ich bestimmte Dinge erklären, zum Beispiel eben jene Module.

Zuerst hab ich die gameplayspezifischen Trigger in zusammengefasst, also die Zeitverzögerung am Anfang, oder die Erweckung der Untoten.

Danach habe ich die Spieleraktionen in einer Node abgelegt, dh. Das Spawnen, und das Unitkaufen.

Auch die Kontrollpunkte haben einen eigenes Modul, diesmal sogar zwei, eines für die Kontrolle und eines für den Timer der kommt, wenn man alle unter seiner Kontrolle hat.

Dann gibt es noch die Node für die Upgrades, dh.

Verbesserung der Türme und Unterstützungstruppen, und zum Schluss das Bot-Modul. In jedem der team- oder spiellerspezifischen Module gibt es mehrere

Untermodule, mit wiederum einer Node für jeden Spieler.

Dieses ganze Modulsystem dient vorallem der Übersicht.

Ein Beispiel: Du willst machen, dass man, wenn man in ein bestimmtes Gebiet kommt, von einem Spino angegriffen

wird, einmalig nur. Dafür brauchst du kein extra Modul, du kannst den Trigger direkt in Root platzieren. Wenn du jedoch einen Spawn für mehrere Spieler machen willst, solltest du schon für jeden Spieler eine Node anlegen. Wenn du überdies noch eine AI-Steuerung einbaust, lohnt sich das Modul-System.

## 1.2 Das Menü

Genug geredet, auf zum Triggern! Erstmal eine Erklärung zum Menü. Das Auslösermenü findet ihr hier:

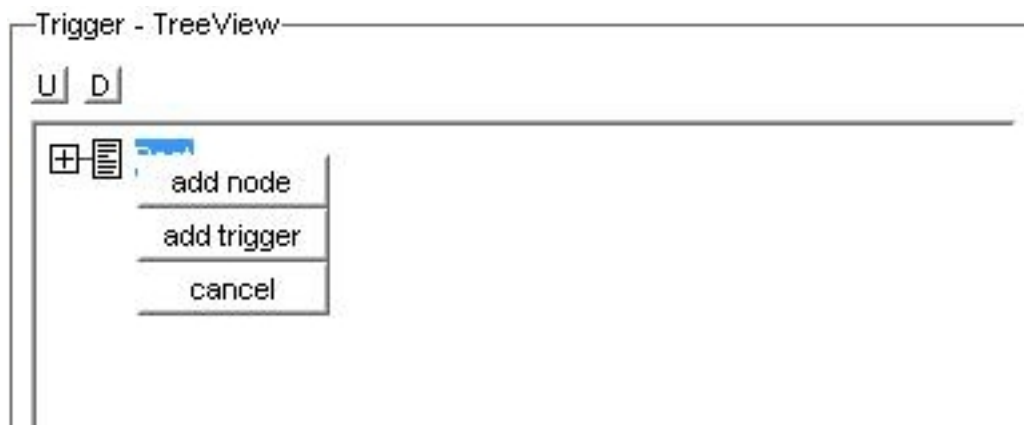


Jetzt hat sich ein Fenster geöffnet, vergrößert es auf ca. 80 % des Bildschirms, um gut zurecht zu kommen. Nun zu seinen Fähigkeiten: Außer damit Trigger zu erstellen, könnt ihr diese auch noch unter **Sort** sortieren (was jedoch auch manuell geht) und Variablen verwalten. Ansonsten bietet die Leiste leider nichts Interessantes.



Das Variablen-Menü öffnet ihr mit **Variables -> Edit Variables**. Alles andere könnt ihr getrost ignorieren, ihr werdet es nicht benötigen oder sonst einen Nutzen darin finden... denke ich, aber probiert es ruhig aus wenn ihr wollt! ;)

Jetzt zu den Triggern und Nodes: Ihr seht einen Eintrag „**Root**“ in der Liste. Dies ist das Wurzelverzeichnis eurer Trigger im Level. Macht einen Rechtsklick darauf, um folgendes Menü zu erhalten:



Erklärung:

**add node** fügt eine Unternode hinzu (vorhin auch als Modul bezeichnet) welche der Strukturierung dient. Trotz der Nodes dürfen keine zwei Trigger denselben Namen haben.

**add trigger** erstellt einen blanko-Trigger.

**cancel** bricht den Rechtsklick ab.

Mit **Up** und **Down** kannst du Trigger und Nodes separat nach oben/unten verschieben. Du kannst jedoch keinen Trigger in den Node-Bereich oder umgekehrt schieben.

Sonstiges:

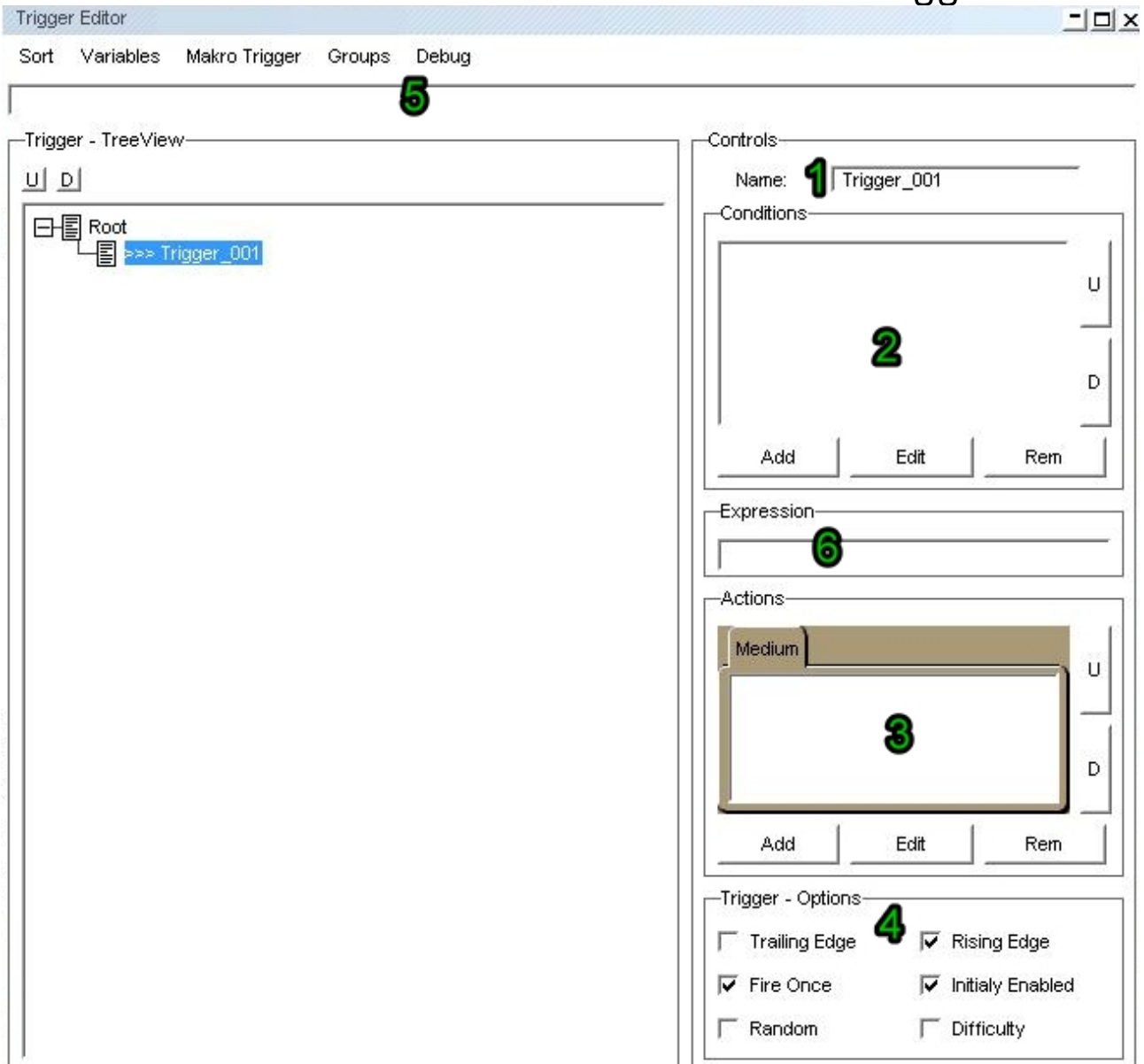
- Trigger und Nodes verschiebt ihr so: Klickt auf darauf, und haltet die Maustaste gedrückt, jedoch ohne sie zu bewegen. Nun habt ihr nach ca. einer Sekunde einen Geist davon an eurem Cursor. Haltet weiterhin gedrückt und zieht es auf die Zielform.
- Wenn ihr eine Condition oder Action hinzufügen wollt, klickt niemals auf „Weiter“ wenn ihr nicht etwas ausgewählt habt, da ansonsten der Editor crasht.
- Doppelklickt ihr auf einen Trigger, so ändert sich das >>> in ..., was zur Folge hat, dass dieser Trigger für das Level komplett deaktiviert ist, d.h. er kann auch mit Actions nicht aktiviert werden.
- Nodes zu deaktivieren wird nicht empfohlen.
- Es sollte bei rekursivem Triggering immer eine Sekunde

als Condition zusätzlich definiert sein, da es sonst zu Fehlern führen kann.

## 2. Triggering

### 2.1 Ein Trigger

Ich erkläre euch kurz die Bestandteile eines Triggers:



1: Das ist der Name, welcher auch in der Liste links Anzeigt wird, bestätigt ihn nach Eingabe mit Return. Seid kreativ. :)

2: Die Conditions werden hier aufgelistet. Mit **Add** fügt ihr

eine hinzu, mit **Edit** könnt ihr die angewählte verändern und mit **Rem** entfernt ihr diese. Mit **U** und **D** könnt ihr die Position dieser Verändern (**Up/Down**)

3: Die Actions. Die Bedienung ist dieselbe, mit Ausnahme, dass wenn ihr „**Difficulty**“ aktiviert habt, ihr durch die Reiter „**Easy**“ „**Medium**“ und „**Hard**“ schalten könnt, um verschiedene Actions hinzuzufügen. (Um in einem Kampagnenlevel z.B. beim leichten Modus weniger Feinde zu spawnen als beim schweren, diese Dinge eben.)

4: Die Triggeroptionen: **Trailing Edge** und **Rising Edge** ignoriert ihr am besten, keine Ahnung was sie bewirken. **Fire Once** bedeutet, dass der Trigger nur einmal ausgelöst wird, was jedoch nicht heißt, dass ihr ihn nicht mit einer Action reaktivieren könnt, wenn ihr den Haken entfernt gilt dies jedoch nicht. Dies ist manchmal verbuggt, die Reaktivierungs-Variante sollte bevorzugt werden. **Initially Enabled** heißt, dass der Auslöser „scharf“ ist, dh. sobald die Conditions erfüllt sind, wird er ausgelöst. Ansonsten muss er eben erst (re)aktiviert werden.

**Random** lässt den Trigger bei Auslösung eine zufällige Action aus den vorhandenen auswählen. **Difficulty** hab ich ja schon erklärt.

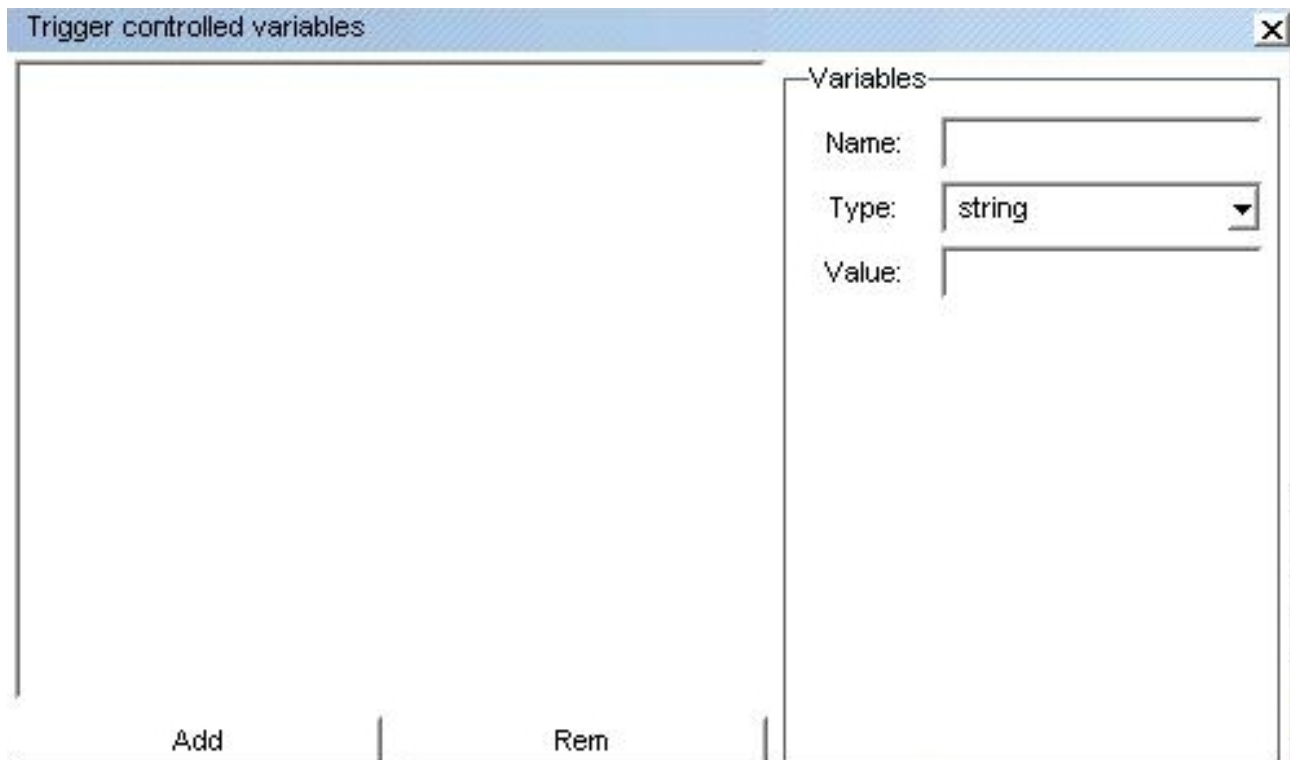
5: Beschreibungstext für den Trigger. Wie der Name, so dient dies nur der Übersicht und Erklärung für Mapper.

6: Dies ist extrem wichtig, sobald ihr mehr als eine Condition habt. Damit könnt ihr definieren, wie diese gehandhabt werden. && bedeutet UND und || bedeutet ODER. Mit Klammer könnt ihr es näher definieren. Neben jeder Condition steht eine Nummer, diese benutzt ihr in diesem Feld. Beispiel: Cond. 1 und 2 ODER stattdessen Condition 3 => (1 && 2) || 3

Bei Nodes gibt es nur den Namen.

## 2.2 Variablen

Unter **Variables** -> **Edit Variables** könnt ihr eure Levelvariablen verwalten.



The screenshot shows a window titled "Trigger controlled variables". It has a close button in the top right corner. The window is split into two main areas. The left area is a large, empty rectangular box. Below this box are two buttons: "Add" and "Rem". The right area is titled "Variables" and contains three input fields: "Name:", "Type:", and "Value:". The "Type:" field is a dropdown menu with "string" selected.

Mit **Add** fügt ihr eine hinzu, mit **Rem** löscht ihr die selektierte. Wenn ihr eine „Rohvariable“ erstellt habt, könnt ihr diese anwählen, und mit den Feldern rechts bearbeiten. Unter **Name** legt ihr ihren Namen fest, mit **Type** den Typ. Empfohlen ist nur **string**, eine Zeichenkette und **int**, eine Ganzzahl. **float** ist eine Gleitkommazahl, dh. sie geht nach dem Komma weiter. **bool** ist verbuggt. Schließlich könnt ihr mit **Value** den Wert definieren, welcher durch Trigger jedoch geändert werden kann. Dieser kann dann ausgegeben werden, oder einfach nur als Counter fungieren.