



Triggerbeschreibung 2.0

Was ist das hier ?

Nachfolgend haben wir für alle Anfänger im Level- oder Szenariodesign aufgeschrieben, was Trigger eigentlich sind, was man damit macht und wofür das gut sein soll. Zu diesem Text gehören noch die Texte zur Beschreibung der Effekte und der Bedingungen, die Sie auf unserer WebSite finden.

"Wieso ist das getrennt? Gehört doch zusammen!" So oder ähnlich könnte man unsere Dreiteilung kommentieren. Aber wenn Sie tatsächlich Anfänger sind sollten Sie unserer Vorgehensweise Vertrauen schenken. Einige wenige Elemente die erst mit der AddOns dazukamen sind nicht enthalten.

Wir beschäftigen uns hier mit der Bearbeitung von Einzelspielerszenarios, bei denen über einen Menüführung die Triggererstellung erfolgen kann.

Was geht denn so ?

Fast alles was Sie in der mitgelieferten Kampagne schon gesehen haben. Nun bedarf es dazu aber einer kleinen aber bedeutenden Feststellung: Viele der Szenarien der Kampagne sind von A bis Z geskriptet, das heißt die künstliche Intelligenz des Spiels wird nicht verwendet sondern jeder Handschlag einer Figur wurde von den Designern vorgegeben oder die Steuerungsdatei der künstlichen Intelligenz wurde angepasst. Mit aktiver Standardintelligenz ist jedenfalls jedwedes sinnvolle beeinflussen von KI-Gegner definitiv nicht möglich.

Aber in wie weit das überhaupt nötig ist scheint die große Frage. Eher angebracht ist die Anlage von Aufgaben und Bedingungen, die der Spieler erledigen muss – und damit ist nicht nur der alles entscheidende Angriff auf den Gegner gemeint, also die Zerstörung seiner Objekte und Einheiten, sondern auch sekundäre Ziele und kleine Erschwernisse.

In unserer bisherigen Arbeit haben wir allerlei unlogische Dinge im Eventbereich entdeckt, leider auch recht ärgerliche Fehler – zum Beispiel die fehlerhafte Verarbeitung von deutschen bzw. nicht angelsächsischen Sonderzeichen.

Was brauchen Sie?

Wenn vorhanden – und wie wir auch schon in den anderen beiden Teilen Effektbeschreibung und Bedingungsbeschreibung gesagt haben – also, wenn vorhanden, sind grundsätzliche Programmierkenntnisse und natürlich ein bisschen Englisch sehr hilfreich. Logisches Denken und auch Phantasie erst recht. Geduld und Zeit sowieso. Und auch die Bearbeitung unseres Leitfadens ist hilfreich.

Wenn Sie diesen Text hier komplett lesen und durcharbeiten werden keine Probleme auftreten. Sie sind am Schluß in der Lage die Zusammenhänge zu verstehen. Was wir nicht vermitteln können ist Kreativität und selbständiges Denken.

Trigger einfach

„Der Trigger ist eine Hülse, in der Bedingungen und Effekte enthalten sind. Der Trigger stellt diese Informationen dem Programm – also dem Spieler – zu festgelegten Konditionen, meist zeitlicher Natur – zur Verfügung. Der Spieler seinerseits wird nun durch das bewußte oder unbewußte Handeln die Bedingungen erfüllen und damit dann die Effekte auslösen.“
(Ensemble Studios, 1998, AOE)

Na bitte. Dies entspricht in seiner Struktur im wesentlichen den Grundprinzipien von Programmiersprachen jeder Art. Am gängigsten hat sich hier die Grundformulierung <Wenn....Dann > gezeigt. Umgangssprachlich ist "Wenn" eine Bedingung und "Dann" der Effekt. Beides zusammen ist dann der Trigger.

Im Spiel ist also ein Trigger eine Kombination aus einer Bedingung oder mehreren Bedingungen, die, wenn Sie in ihrer Summe zutreffen, einen Effekt oder mehrere Effekte auslösen. Folgender Trigger ist in fast jedem Szenario so oder ähnlich enthalten:

Bedingungsseite	Effektseite
Bedingung: IsDead Towncenter Player 2	Effekt: SetPlayerWon Player 1

"Wenn das Dorfzentrum von Spieler 2 zerstört ist, setze den Spieler 1 als Sieger fest."

Während des Spiels prüft das Programm in kontinuierlichen Abständen ob das Dorfzentrum des Spielers 2 vorhanden ist. Ist es nicht (mehr) vorhanden, ist die Bedingung <WAHR> und das Programm startet den Effekt, den Spieler 1 (Player1) zum Sieger zu erklären (SetPlayerWon). Ist das Dorfzentrum vorhanden ist die Bedingung <FALSCH> und es passiert nichts.

Grundsatz 1 : Um einen Effekt zu starten, muss die Bedingungsseite <WAHR> sein.

Grundsatz 2 : Um einen Effekt nicht zu starten, muss die Bedingungsseite <FALSCH> sein.

Letzteres hört sich merkwürdig an, aber so sollten Sie das im Kopf behalten. Sie müssen Ihren angelegten Trigger genau daraufhin prüfen.

Trigger einfach mehrfach

Neben solchen eindimensionalen Triggern – eine Bedingung, ein Effekt – sind auch mehrdimensionale Trigger möglich, bei denen mehrere Bedingungen einen oder mehrere Effekte auslösen.

Bedingungsseite	Effektseite
Bedingung1: IsDead Towncenter Player 2	
Bedingung2: IsDead Towncenter Player 3	Effekt: SetPlayerWon Player 1

"Wenn das Dorfzentrum von Spieler 2 und Spieler 3 zerstört ist, setze den Spieler 1 als Sieger fest."

Wesentlich hierbei: Nur wenn beide Bedingungen zutreffen, ist die Bedingung für das Auslösen des Effektes insgesamt <WAHR> und der Effekt wird ausgelöst. Ist eine der beiden Bedingungen <FALSCH> ist in der Summe die ganze Bedingung <FALSCH> und der Effekt wird nicht ausgelöst.

Trigger einfach kompliziert

Damit das ganze dann auch schön beweglich bleibt und kompliziert wird, kann man die Bedingungsseite noch durch so genannte Operatoren verhunzen. Im Beispiel eben ist nämlich ein Teil der Bedingungsseite unerwähnt geblieben, der grundsätzlich immer dort steht und im normalen Text nicht auffällt : UND.

Bedingungsseite	Effektseite
Bedingung1: IsDead Towncenter Player 2	
UND	
Bedingung2: IsDead Towncenter Player 3	Effekt: SetPlayerWon Player 1

"Wenn das Dorfzentrum von Spieler 2 UND Spieler 3 zerstört ist, setze den Spieler 1 als Sieger fest."

Den UND Operator kann man nun auch durch einen anderen ersetzen: ODER

Bedingungsseite	Effektseite
Bedingung1: IsDead Towncenter Player 2	

ODER

Bedingung2: IsDead Towncenter Player 3 Effekt: SetPlayerWon Player 1

"Wenn das Dorfzentrum von Spieler 2 ODER Spieler 3 zerstört ist, setze den Spieler 1 als Sieger fest."

Ein weiterer Operator ist das schöne Wort NICHT.

Bedingungsseite

Effektseite

Bedingung1: IsDead Towncenter Player 2

NICHT

Bedingung2: IsDead Towncenter Player 3 Effekt: SetPlayerWon Player 1

"Wenn das Dorfzentrum von Spieler 2, NICHT von Spieler 3 zerstört ist, setze den Spieler 1 als Sieger fest."

Sollte man meinen, ist aber nicht so gemeint. Sondern:

"Wenn das Dorfzentrum von Spieler 2 NICHT zerstört ist oder das Dorfzentrum von Spieler 3 NICHT zerstört ist, setze den Spieler 1 als Sieger fest."

Diese Bedingung ist nicht besonders sinnvoll – Spieler 1 wäre bei Spielstart schon Sieger (?!) –, aber : Wenn eine der Bedingungen nicht wahr ist, ist die gesamte Bedingungsseite doch <WAHR> und der Effekt wird ausgeführt. Hört sich dämlich an, ist aber leider so. Wir sind bei unseren Kunstwerken bisher ohne diesen Operator ausgekommen. (Kopfschmerzen!)

Trigger insgesamt

Man ahnte es schon: Es kann schwierig werden. Unsere Empfehlung lautet für die Operatoren: Verzichten Sie auf die NICHT Verknüpfung. Machen Sie Ihre Bedingungsseite so einfach wie möglich. Wie Sie später sehen werden, ist eine Verbindung von Triggern einfacher als die ganzen Überlegungen, wie man nun die Bedingungen verknüpfen könnte.

Fassen wir zusammen und ergänzen wir:

Ein Trigger besteht aus einer oder mehreren Bedingungen und einem oder mehreren Effekten.

Bedingungen müssen in ihrer Summe immer <WAHR> sein um die Effektseite auszulösen.

Bedingungen eines Triggers sind als Standard alle mit UND verknüpft.

Oder : alle Bedingungen eines Triggers können mit einem ODER verknüpft werden.

Oder : alle Bedingungen eines Triggers können mit einem NICHT verknüpft werden. (Finger weg.)

Alle Bedingungen eines Triggers können nicht mit unterschiedlichen Operatoren verknüpft werden.

Was machen wir nun damit?

Die bis hierher getroffenen Feststellungen sollen der theoretische Aspekt des Ganzen sein. Die folgende Anleitung ist dann in Verbindung mit unserem Leitfadenden Sie hoffentlich auch abgearbeitet haben - das Übungspaket, an dessen Ende Sie eigentlich alles verstanden haben sollten um mit den Bedingungsbeschreibungen und der Effekteliste von unserer WebSite selbst ein Szenario mit Triggern und somit ein wenig mehr Spannung zu versehen.

Die Editorbearbeitung

Innerhalb des Editors wird die Triggererstellung fast vollständig im Rahmen einer Menüführung gesteuert. Starten Sie den Editor aus dem Hauptmenü des Spieles über <Einzelspieler><Szenario-Editor> und wählen Sie dann im Menü <Trigger> die Option <Trigger>. Die Menüpunkte <Gruppeneditor> und <Armeeeditor> kommen später an die Reihe.

Auf dem Bildschirm erscheinen zwei Dialogfenster, "Trigger-Editor" das kleine Fenster, "Trigger" das große Fenster. Das kleine Fenster links ist immer zu sehen, das größere prinzipiell auch, nur wird dort die jeweilig benötigte Eingabemaske eingeblendet, je nach der Auswahl bei den Optionen "Bear" im kleinen Fenster :



Damit hätten wir die wesentlichsten Ansichten zusammen der Triggerbearbeitung zusammen.

Die Triggerbearbeitung

Für jedes Spiel sind zumindest zwei Trigger üblich. Zum einen der, der dem Spieler den Sieg verkündet, zum anderen natürlich der, der dem Spieler die Niederlage verkündet. Am ersteren werden wir jetzt mal die Erstellung eines Triggers ausführen. Nachfolgend sind alle Arbeiten, die Sie erstmal ausführen sollten in **blau** dargestellt – der erste Versuch soll ja auch klappen. Also.

Starten Sie das Spiel

Starten Sie den Editor

Legen Sie eine neue Zufallskarte an mit 4 Spielern.

Speichern Sie Ihr Kunstwerk nun unter dem Namen "Triggertest" – oder was Sie möchten.

Sollten Sie jetzt schon Probleme haben: Wickeln Sie erstmal unseren Leitfaden zur Szenarioerstellung ab. Was dann hier folgt betrachten Sie als Wiederholung.

Wählen Sie im Editormenü die Option Trigger/Trigger

Klicken Sie im kleinen Fenster "Trigger-Editor" auf die Schaltfläche <Einfügen>.

Wenn dies Ihr erster Versuch mit Triggern ist, erscheint im Listenfeld des kleinen Dialogfensters nun <Trigger_0>. Diese Nummer wird im Laufe des Spiels mit jedem neu angelegten (eingefügten) Trigger hochgezählt.

Um einen Trigger zu bearbeiten klicken Sie im kleinen Editorfenster auf seinen Namen! Damit wird er zum aktuell ausgewählten Trigger und alle folgenden Arbeiten beziehen sich ausschließlich auf diesen Trigger. Diese Vorgehensweise gilt auch für den Bedingungs- und Effektedialog.

Im kleinen Dialogfenster sind dann noch die Schalter <Entf.> und <Kopie> zu bewundern : der eine entfernt den aktuell markierten Trigger ohne Nachfrage, der andere kopiert den aktuell markierten Trigger und fügt ihn mit dem Zusatz "Kopie_von_" in die Triggerliste ein. Es werden dabei auch alle Bedingungen und Effekte kopiert!

Nach der Anlage des Triggers hat sich auch im großen Dialogfeld etwas geändert:



Eine (fast) leerer Trigger wurde angelegt mit der Standardbedingung "Always" und dem Standardeffekt "SetIdleProcessing". Betrachten wir zunächst die linke Seite ist diesem Dialogfeld.

Da wäre das Feld <Name>. Es ist erforderlich und sehr empfehlenswert, jedem Trigger einen eigenen, wenn möglich aussagekräftigen Namen zu verpassen, sonst findet man ihn später kaum wieder. Die drei Optionen darunter bestimmen die Behandlung des aktuell angezeigten Triggers.

<Aktiv>

Wie oben erwähnt, wird zyklisch während des Programmablaufs geprüft, ob die im Trigger festgelegte Bedingung zutreffen und ergo abgearbeitet werden müssen. Wenn Sie den Haken hier vor <Aktiv> entfernen, ist der gesamte Trigger für das Programm nicht sichtbar und wird somit nicht geprüft und natürlich auch nicht abgearbeitet. Dieser Status ist aber in vielen Fällen jedoch wünschenswert. Während des Spiels kann ein solcher Trigger nämlich aktiviert und auch deaktiviert werden.

<Schleife>

Dies legt fest, ob der Trigger wiederholt ausgeführt werden soll – letztlich unendlich oft, wenn man ihn nicht deaktiviert oder die Bedingung nicht mehr erfüllt wird, denn: es wird nicht einfach der Effekt immer wieder wiederholt, sondern der ganze Trigger wird erneut geprüft. Das bedeutet aber auch, dass die Bedingungslage immer wieder geprüft wird. Trifft die Bedingung nicht mehr zu, endet die Schleifenausführung! Damit wären wir vom Verständnis her bei einem wichtigen Satz: Ist ein Trigger abgearbeitet, die Bedingung ist also erfüllt und der Effekt erfolgt, wird der Trigger vom Programm deaktiviert und wird nicht mehr geprüft – außer im Falle einer Schleife eben.

<Sofort ausführen>

Üblicherweise dauert es ca. 3 Sekunden, bis ein Trigger im Spielablauf abgearbeitet wird. Die Einstellung <Sofort ausführen> sorgt dafür, dass sich diese Ausführungszeit praktisch auf Null verkürzt, der Trigger also sehr schnell geprüft und ggf. ausgeführt wird. Eine solche Einstellung ist bei den so genannten Eröffnungstriggern erforderlich, die die gewünschten Grundeinstellungen zu Spielanfang festlegen sollen. Ansonsten sollte man die nächste Auswahlmöglichkeit heranziehen, denn wenn Sie alle Trigger auf <Sofort ausführen> setzen landen Sie logischerweise wieder bei höheren Ausführungszeit.

<Priorität>

Dieser Schieberegler legt fest, ob der aktuelle Trigger in seiner Klasse nun sehr wichtig ist oder nicht. Der Trigger zur Feststellung des Sieges – oder der Niederlage – ist keinesfalls eminent wichtig. Lässt man die Einstellung auf normal, vergehen wie gesagt bestenfalls 3 Sekunden bis zur Auslösung des Effektes, wenn die Bedingung zutrifft. Eine hohe Priorität ist eigentlich nur bei Trigger erforderlich, die zyklisch etwas auf der Karte bewegen oder bei Kamerafahrten. Letztlich ist es eine Frage der Leistungsfähigkeit Ihres Rechners und der Menge der Trigger, ob und wann Sie die Priorität eines Triggers vom Standard her ändern. Zurück zum Trigger, den wir zur Festlegung der Siegbedingung und des Siegeffektes anlegen wollen. Als erstes ändern wir seinen Namen. Achtung:

Triggernamen dürfen keine Sonder- oder Leerzeichen enthalten !!!!

Markieren Sie im Listenfeld des kleinen Dialogfensters den Eintrag <Trigger_0>

Markieren ggf. Sie dort die Option <Trigger>

Markieren Sie im Feld <Name> des großen Dialogfensters den Text "Trigger_0"

Überschreiben Sie ihn mit "Main_Sieg"

Drücken Sie die Entertaste (!)

Sie werden als erstes merken, dass das markieren eine fummelige Sache ist – wie in allen Masken des editors übrigens. Hier hilft nur Übung. Nach der Testeingabe ist es unbedingt notwendig, die Entertaste zu betätigen, da sonst der eben eingegebene Text verworfen wird. Im übrigen ist jetzt der neue Name des Triggers im kleinen Dialogfenster links im Listenfeld erschienen.

Wählen Sie im kleinen Dialogfenster "Trigger Editor" die Option <Beding.> (Bedingung).

Die Anzeige des großen Dialogfensters ändert sich – Abb. siehe oben. Im linken Listenfeld des großen Dialogfeldes sind die zu dem aktuellen Trigger gehörigen Bedingungen aufgelistet – momentan nur <Always>. Auch hier gilt das Markierungsprinzip: Die Bedingung, die Sie bearbeiten wollen, muss hier im Listenfeld markiert sein (weiße Schrift).

<Einfügen> : fügt eine weitere Bedingung hinzu (Standardbedingung "SetIdleProcessing")

<Löschen> : löscht die aktuell markierte Bedingung

<Kopie> : kopiert die aktuell markierte Bedingung und fügt Sie der Liste der Bedingungen hinzu

Die Kopie einer Bedingung wird im Gegensatz zu den Triggern mit genau dem gleichen Namen wie das Original versehen – Vorsicht also! Verwechslungsgefahr! Die beiden Optionen:

<ODER> : siehe hierzu die Ausführungen am Anfang dieses Textes

und

<NICHT> : siehe hierzu ebenfalls die Ausführungen am Anfang dieses Textes

Die vernünftige Anwendung dieser beiden Optionen werden wir in unserem Text <Triggerbeispiele>, den Sie auch auf unserer Website finden, erläutern. Jetzt aber achten Sie bitte darauf, keine der beiden Optionen zu markieren.

Das Listenfeld <Type> im rechten Bereich des großen Dialogfeldes enthält alle möglichen Bedingungen des Spiels. Wenn Sie den gelben Pfeil anklicken, wird die Liste nach oben geöffnet. Wenn sich der Mauszeiger *in* diesem Feld befindet, können Sie über die Tastatur die alphabetische Liste scrollen, sonst natürlich mit dem Maustaste rollen und mit einem Rechtsklick die gewünschte Bedingung auswählen. Wenn für diese Bedingung Parameter erforderlich sind – zum Beispiel bei der Bedingung <Timer> die Angabe von Sekunden – werden die erforderlich Parameterfelder unter der Liste angezeigt.

Das kleine Feld über der Liste der angelegten Bedingungen entspricht dem Feld <Name> bei den Triggern. Hier können Sie der aktuellen Bedingung einen anderen Namen geben – was Sie auch tun sollten!

Markieren Sie im kleinen Listenfeld die Bedingung <Always>

Öffnen Sie das Dialogfeld <Type>

Platzieren Sie den Cursor im nun aufgeklappten Listenfeld.

Wählen Sie auf der Tastatur den Buchstaben <I>

Bewegen Sie den Markierungsbalken des Listenfeldes zum Befehl <IsDead>.

Klicken Sie auf diesen Befehl.

Unter dem Listenfeld sind nun zwei Schalter aufgetaucht. Dies geschieht sowohl hier als auch bei den Effekten des Öfteren. Damit werden in der Regel Objekte auf der Karte oder Bereiche festgelegt.

Bewegen Sie den Kartenausschnitt so, dass das Dorfzentrum des Spielers 2 (rot) sichtbar wird.

Wählen Sie dieses Dorfzentrum an und dessen Markierungsrahmen sichtbar wird.

Klicken Sie nun im großen Dialogfeld auf den Schalter <SourceUnits>.

Damit ist das Dorfzentrum des Spielers 2 als Prüfobjekt festgelegt.

Soweit sollte klar sein: Wenn das gewählte Objekt <IsDead> ist, also zerstört, ist die Bedingung erfüllt. Der Schalter <Show> zentriert übrigens die Ansicht auf das gewählte Objekt.

Damit wäre die erste Bedingung angelegt und könnte schon als Siegbedingung ausreichen. Aber wir werden noch zwei weitere Bedingungen hier hinzufügen.

Achten Sie darauf dass die eben angelegte Bedingung im Listenfeld markiert ist.

Benennen Sie mittels des Feldes über diesem Listenfeld diese Bedingung um in "TC IsDead Spieler 2".

Klicken Sie auf den Schalter <Kopie>

Eine weitere Bedingung wurde hinzugefügt, die als Kopie natürlich dem Original entspricht.

Achten Sie darauf dass die eben angelegte Kopie im Listenfeld markiert ist.

Bewegen Sie nun den Kartenausschnitt so, dass das Dorfzentrum des Spielers 3 (gelb) sichtbar wird.

Markieren Sie dieses Dorfzentrum.

Klicken Sie im großen Dialogfeld auf den Schalter <SourceUnits>.

Ändern Sie den <Spieler> 2 auf <Spieler> 3

Damit haben wir festgelegt, dass auch der dritte Mitspieler sein Dorfzentrum für den Sieg von Spieler 1 verlieren muss. Das Anlegen der dritten Bedingung für Spieler 4 sollte nun kein Problem für Sie sein, wobei Sie bitte nicht die Namensänderungen vergessen: "TC IsDead Spieler 3" und "TC IsDead Spieler 4". Dann wären wir mit den Bedingungen für diesen Trigger durch und kommen zum Effekt.

Wählen Sie im kleinen Dialogfenster "Trigger Editor" die Option <Effekte>

Wieder ändert sich die Anzeige im großen Dialogfeld hin zu den Effekten. Der Aufbau und die Funktionalität entspricht bis auf die Angaben ODER/NICHT dem des Bedingungsdialoges.

Markieren Sie im kleinen Listenfeld den Effekt <SetIdleProcessing> (Markierungsgrundsatz!).

Öffnen Sie das Dialogfeld <Type>.

Platzieren Sie den Cursor im nun aufgeklappten Listenfeld.

Wählen Sie auf der Tastatur den Buchstaben <s>.

Suchen Sie den Befehl <SetPlayerWon> und wählen Sie ihn aus.

Das Listenfeld wird geschlossen, unterhalb davon wird jetzt ein weiteres Listenfeld angezeigt, in dem Sie den Spieler auswählen, der zum Sieger erklärt werden soll – Spieler 1 in diesem Fall, was aber der Einstellungsstandard ist, also ist keine Einstellung erforderlich.

Wählen Sie kleinen Dialogfenster "Trigger Editor" die Option Trigger

Der Bildschirm sollte dann so aussehen:



Speichern Sie Ihr Kunstwerk.

Damit wäre der Trigger für den Sieg des Spielers 1 angelegt. Wie Sie der Liste der Effekte entnehmen können, ist der Effekt ziemlich schlicht. Aber hier können Sie natürlich noch kreativ werden.

Die Anlage des Niederlagetriggers ist analog zu diesem hier. Ein neuer Trigger mit dem Namen "Main_Niederlage", als Bedingung ebenfalls den Befehl <IsDead>, dann natürlich wird das Dorfzentrum von Spieler 1 ausgewählt, als Effekt den Befehle <SetPlayerDefeated>. Fertig.

Aber das sollten Sie nun alleine können.

Der Gruppeneditor

Spätestens wenn Sie die ersten 10 Trigger erstellt haben würden Sie der besseren Übersicht wegen das Ganze ein wenig sortieren. Sie haben allen Triggern sinnvolle Namen gegeben (!!!!!), aber trotzdem beginnt die Übersicht zu schwinden. Also nehmen wir das Hilfsmittel des Gruppeneditors zur Hand.

Im kleinen Dialogfenster "Trigger Editor" ist ihnen sicher das Listenfeld <Gruppe> aufgefallen, in dem zur Zeit lediglich "Nicht gruppiert" enthalten ist. Mittels des Gruppeneditors können Sie nun weitere Gruppen anlegen, dort Trigger nach Sinn und Zweck zusammenfassen, hier Dialogfenster "Trigger-Editor" entsprechend anzeigen lassen.

Sie sollten zunächst zu 10 Trigger erzeugen, ohne Inhalt, ohne besondere Namen, nur zu Demonstrationszwecken. Anschließend starten Sie den Gruppeneditor im Menü <Trigger>. Das sieht dann am Bildschirm so aus (hier verkleinert):



Übersetzt: In der Gruppe "Nicht gruppiert" sind die im Listenfeld daneben angezeigten Trigger enthalten.

Klicken Sie auf den Schalter <Einfügen> im linken Dialogfenster "Triggergruppen-Editor".

Im Listenfeld links erscheint die Bezeichnung <Group1> über der Bezeichnung <Nicht gruppiert>, das rechte Dialogfenster ändert seine Ansicht : Im Feld <Name> steht Group1, das Listenfeld Trigger ist leer.

Wählen Sie die Option <Trigger> im linken Dialogfeld unten.

Der Bildschirm sieht dann so aus : (auch hier verkleinert):



Im rechten Dialogfeld stehen jetzt im rechten Listenfeld unter dem Feld <Gruppe> die Trigger, die in der Gruppe "Nicht gruppiert" enthalten sind. Im linken Listenfeld <Aktuell> steht noch nichts. <Aktuell> bedeutet, hier stehen die Trigger der Gruppe, die im kleinen Dialogfeld "Triggergruppen Editor" markiert ist, hier jetzt "Group_1" – und diese Gruppe ist wie gesagt noch leer. Mittels der beiden Schalter zwischen den beiden Listenfeldern im rechten Dialogfeld können Sie den jeweils markierten Trigger nun der neuen Gruppe zuordnen bzw. auch wieder daraus entfernen.

In der Regel sollten Sie eine Triggerabteilung, die ursächlich zusammengehört, nach der weitestgehenden Fertigstellung in einer Gruppe zusammenfassen. Wie Sie in unserem Szenario "Handelswege" feststellen können, sind hier zum Beispiel die Starteinstellungen alle in einer Gruppe zusammengefasst.

Aus langer Erfahrung mit anderen Spieleditoren wissen wir, das spätestens nach 7 Tagen niemand mehr einer pure Triggerauflistung sinnvoll auflösen kann. Alleine schon deswegen empfehlen wir diese Methode dringend.

Der Armeeditor

Die vielen Bedingungen und Effekte, die sich auf eine <Army...> beziehen, bedürfen eines Eintrages in diesem Bereich – es sei denn, Sie erstellen diese Armee direkt bei der Anlage der Trigger. Wenn nicht, ein Wort zum Verständnis:

Jede hier angelegte Armee ist letztlich nur eine leere Hülse, der Sie mittels der Effekte innerhalb der Trigger Einheiten oder Objekte zuordnen – ein Oberbegriff sozusagen. Grundsätzlich ist jede Armee einem Spieler zugeordnet, im Prinzip aber kann jede Armee per Trigger von allen Spielern verwendet werden. Sie können einer Armee so ziemlich jede bewegliche Einheit – aber auch unbewegliche Objekte – zuordnen. der Logik folgend sollten Sie das aber vermeiden. Starten Sie den Armeeditor im Menü <Trigger>.

Um eine Gruppe von Einheiten auf der Karte zur Armee zu befördern, markieren Sie diese Einheiten. Alle auf einmal oder bei gehaltener Umschalttaste einzeln. Dann klicken Sie im Dialogfenster ins Feld <Name>, legen



einen solchen fest, anschließen klicken Sie auf den Schalter <Armee erstellen>. Im Listenfeld <Aktuelle Armeen> ist nun diese Armee aufgelistet.

Um eine Armee zu erstellen, die später im Spiel mit Einheiten gefüllt werden soll, erstellen Sie eine Geisterarmee. Hierzu legen Sie erst wieder einen Namen fest, dann klicken Sie auf den Schalter <Geisterarmee>. Wie die zuerst angelegte Armee aus existenten Einheiten ist auch diese Armee in den dafür vorgesehenen Effekten auswählen, um Sie mit Einheiten zu füllen (ArmyDeploy)

Was man dann mit diesen Armeen machen kann, erklären wir in unserem Text "Triggerbeispiele", den Sie auf unserer WebSite finden.

Citybuilders © 2006-2018

Dieser Text ist keine vom Hersteller des Spiels <Age of Empires 3> autorisierte Fassung eines oder eines mit dem Produkt ausgelieferten Handbuchs oder sonstige Arbeitsrichtlinie sondern stellt eine Beschreibung da, die wir aufgrund der Arbeit mit dem Programm erstellt haben. Die Angaben und Arbeitsanweisungen werden ohne Gewähr auf Richtigkeit, Vollständigkeit oder Funktionsfähigkeit erteilt und stellen nur Vorschläge da, die wir nach bestem Wissen erarbeitet haben. Es wird keinerlei Haftung für jedweden Schaden der aus der Anwendung der hier gemachten Angaben entsteht übernommen. Alle genannten Markennamen oder Markenhinweise sind Eigentum der jeweiligen Markeninhaber