

Tutorial zum Bau von Einheiten in Gebäuden

	Seite
1. Voraussetzungen	-1-
2. Ziele festlegen	-1-
3. Permissions und Subroutines erstellen	-2-
4. Teams definieren und konfigurieren	-7-
5. Gebäude bauen lassen	-10-

1. Voraussetzungen

- Handhabung des Worldbuilder (WB) ist geläufig
- Erstellen von Teams ist bekannt
- Spawn von Einheiten und Teams ist bekannt
- Setzen von WayPoints und anlegen von Routen mittels WayPoints ist geläufig

Soweit ich keine Screen Shots zur Erläuterung der Scripte zur Verfügung habe, verwende ich folgende Syntax dabei:

Scriptname < Name >

Options, dort wo Haken gesetzt sind wird die Option genannt.

Condition → Pfadangabe die zum Befehl führt

Action → Pfadangabe die zum Befehl führt

Alle Worte die im WB eine Bedeutung haben werde ich *kursiv* schreiben.

2. Ziele festlegen

Einheiten und Teams per Spawn auf die Map zu setzen ist eine beliebte Vorgehensweise, weil sie einfach und leicht zu praktizieren ist. Leider sieht es nicht sehr realistisch aus, wenn Einheiten plötzlich hinter Militärgebäuden aus dem „Nichts“ auftauchen oder vom Rand her in die Map laufen.

Lasst uns also ein paar Scripte schreiben die Einheiten in Militärgebäuden bauen.

Was wollen wir bauen?

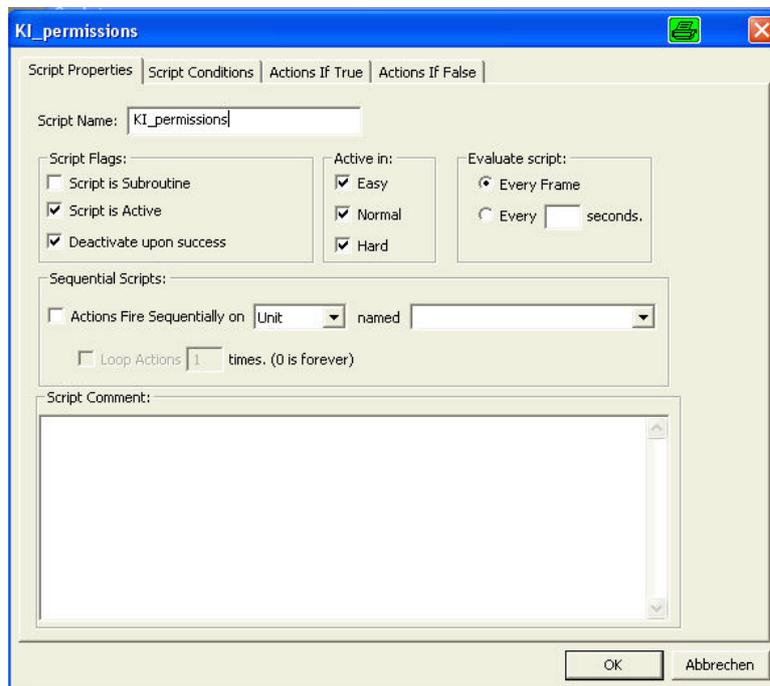
Wir wählen in unserem Beispiel die Goblins des *PlyrWild* aus und lassen drei Instanzen bestehend aus jeweils 2 Batt *GoblinArcher* und *GoblinFighter* in einer *GoblinKaserne* durch die KI bauen. Nachdem diese erstellt sind sollen sie „Erfahrung“ und alle verfügbaren fraktionspezifischen Upgrades erhalten.

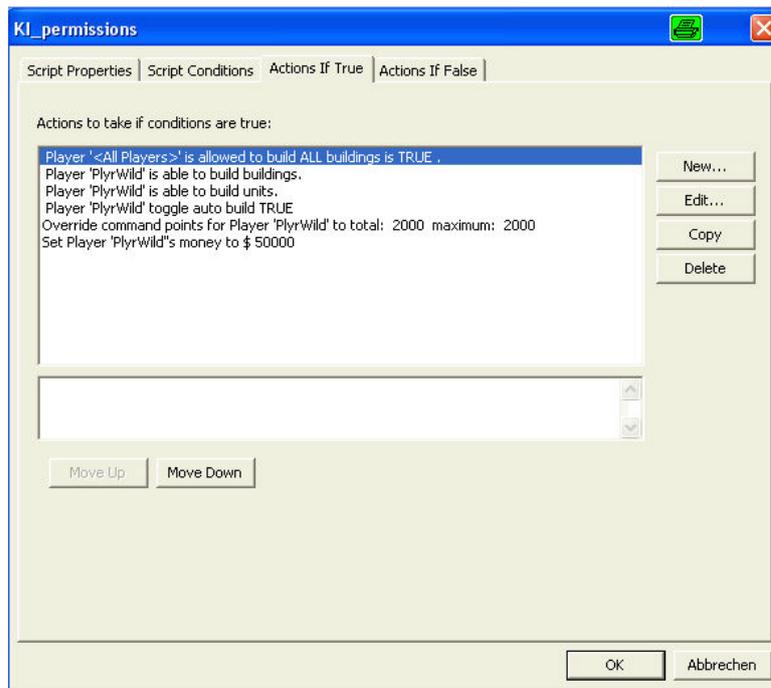
Das Team, das sich dann aus *GoblinArcher* und *GoblinFighter* zusammensetzt nennen wir „*Goblin_attackers*“

Die Kaserne existiert noch nicht und soll durch einen Baumeister erstellt werden, auch das werden wir über Scripte steuern.

3. Permissions und Subroutines erstellen

Damit die KI Gebäude und Einheiten bauen kann müsst ihr dies durch ein Script erlauben. Bei dieser Gelegenheit setzen wir auch die „CommandPoints“ und das „Money“ der KI.





Scriptname <KI_permissions>

Options

Active

Deactivte upon success

Every Frame

Condition

Scripting/TRUE

Action

Player/Building/Allow or Disallow Buildings

Player/Permissions/Player can build Buildings

Player/Permissions/Player can build Units

Player/Permissions/Turn on or off a Player's auto building of Units

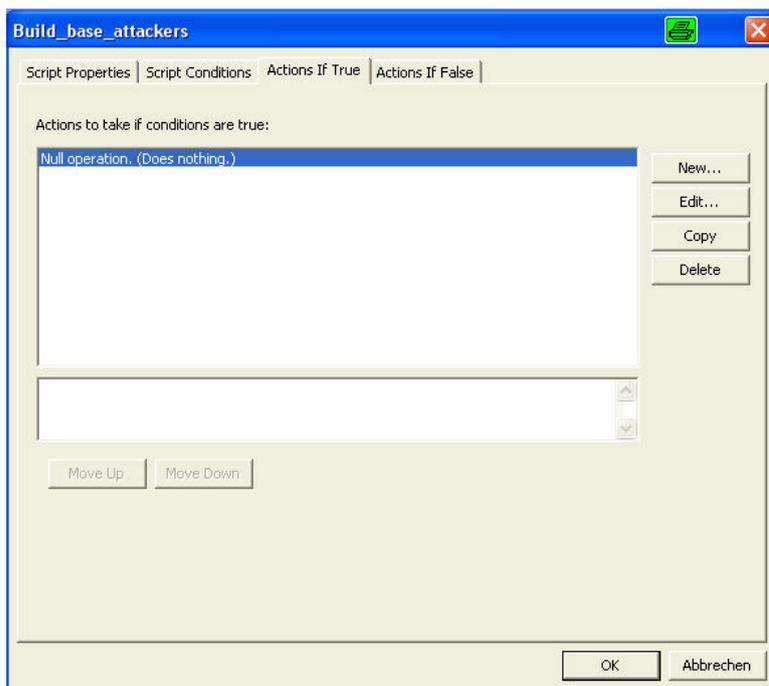
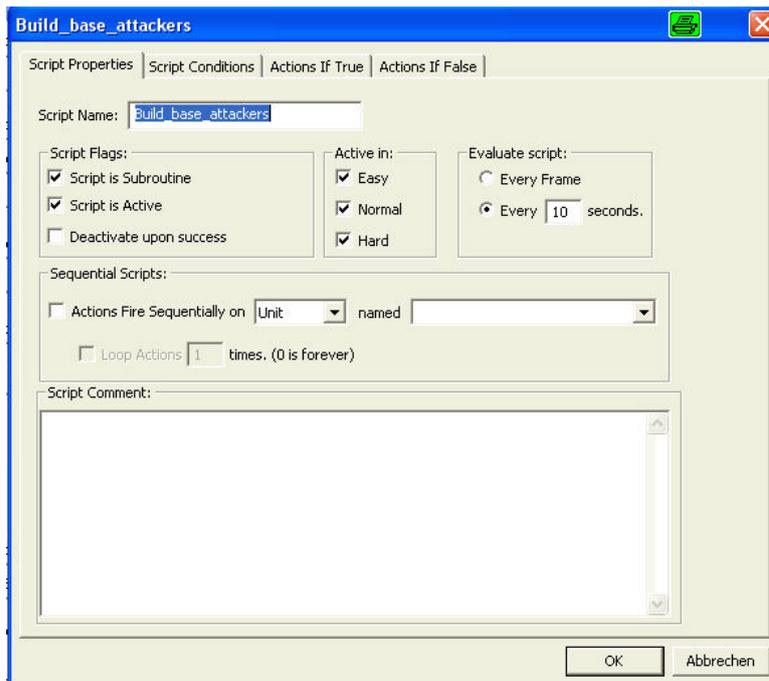
Player/Command Points/Override player command points

Player/Money/Set Player's Money

Jetzt erstellen wir das Script welches erlaubt die Einheiten zu bauen, wenn eine bestimmte Bedingung erfüllt ist. Um es nicht zu kompliziert zu machen ist in diesem Script die Bedingung auf *TRUE* gesetzt, ihr könnt aber hier euere mapspezifische Bedingung eintragen.

Das Script ist eine Subroutine und wird später mit dem Team verbunden das wir erstellen wollen. Der *Action* Part bleibt leer, denn das Script erfüllt seinen Zweck, wenn die *Condition TRUE* liefert.

Unter *Evaluate script* könnt ihr einstellen in welchem Zeitintervall das Script ausgeführt werden soll.



Scriptname <Build_base_attackers>

Options

Script is Subroutine

Script is Active

Every 10 seconds

Condition

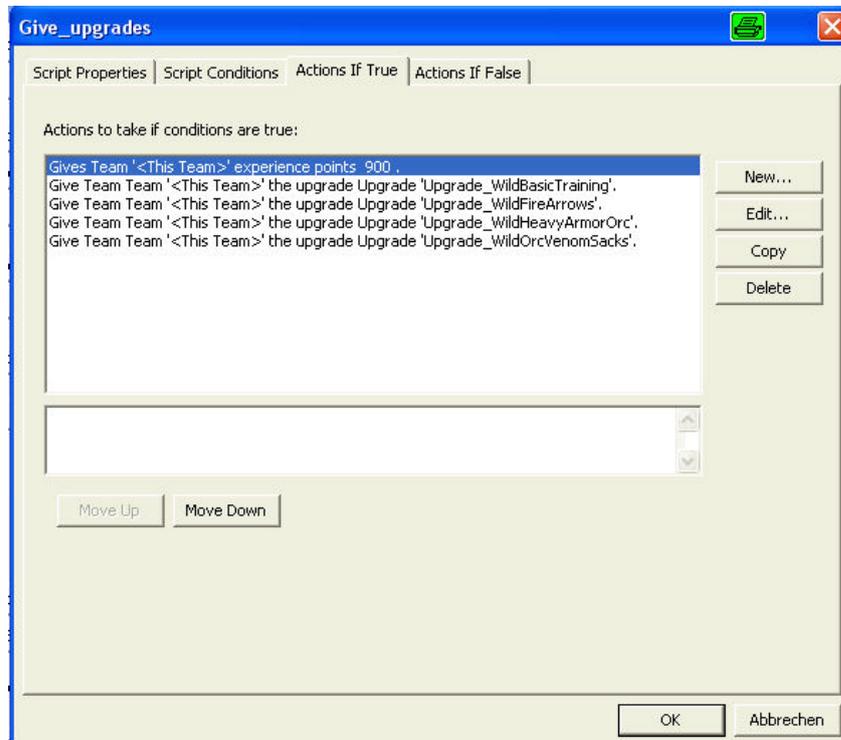
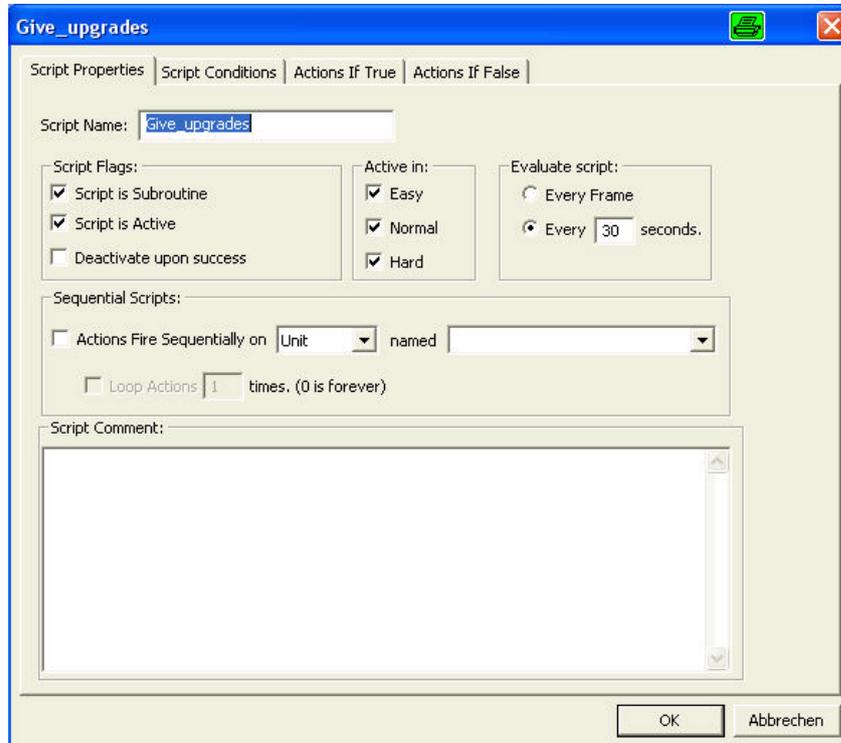
Scripting/TRUE

Action

Null operation (Does nothing)

Das nächste Script gibt dem gerade erstellten Team Erfahrung und die für den PlyrWild verfügbaren Upgrades der GoblinArcher und GoblinFighter. Die *Conditions* habe ich der Einfachheit halber wieder auf *TRUE* gesetzt.

Das Script wird im 30 Sekundentakt ausgeführt, diese Zeit könnt ihr variieren, je nachdem wie viele Instanzen des Teams gebaut werden sollen, aber dazu später mehr.



Scriptname <Give_upgrades>

Options

Script is Subroutine

Script is Active

Every 30 seconds

Condition

Scripting/TRUE

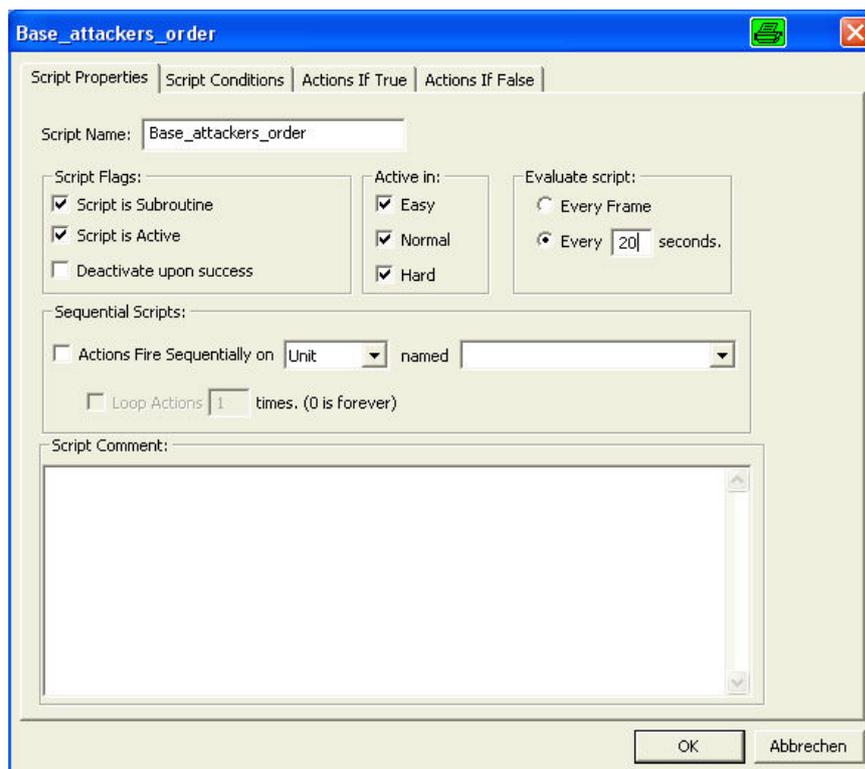
Action

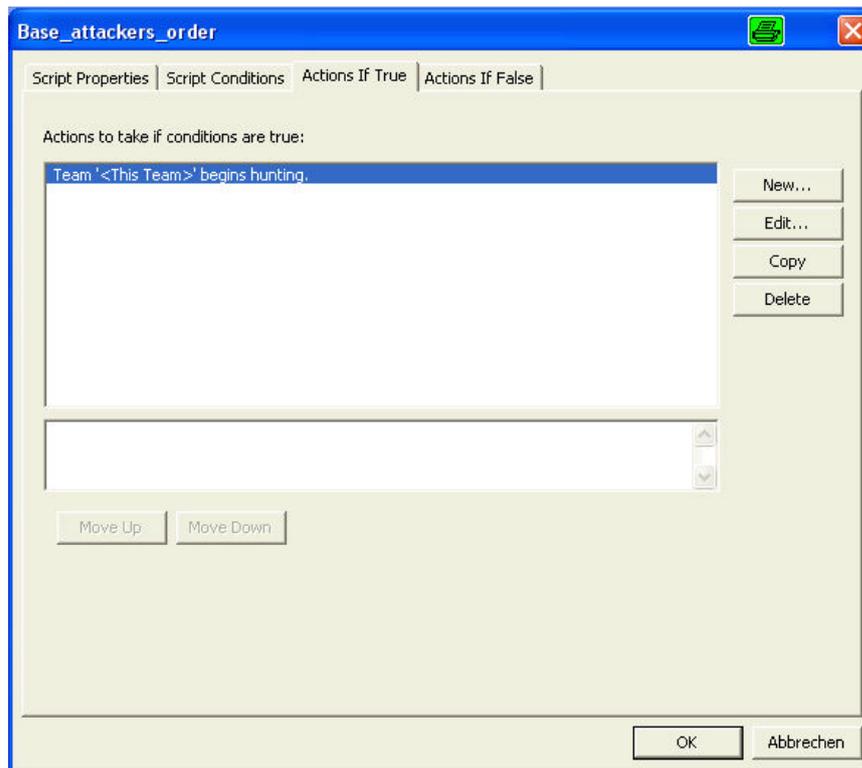
Team Action/Experience/Give the Team experience points

Team Action/Upgrades/Give Team the specified upgrade

Bei der Teamdefinition verwenden wir <This Team>. Diese Teambezeichnung ist ein Platzhalter, der das Script unabhängig vom tatsächlich verwendeten Team macht.

Nun weisen wir das Team an welche Aufgabe es hat nachdem Erstellen.





Scriptname *<Base_attackers_orders>*

Options

Script is Subroutine

Script is Active

Every 20 seconds

Condition

Scripting/TRUE

Action

Team Action/Attack or hunt/Hunt (choose Enemy target....)

Bei der Teamdefinition verwenden wir *<This Team>*. Diese Teambezeichnung ist ein Platzhalter, der das Script unabhängig vom tatsächlich verwendeten Team macht.

Alle 20 Sekunden wird das Team daran erinnert wie sein Auftrag lautet.

4. Teams definieren und konfigurieren

Öffnet im WB „*Edit Teams*“ oder drückt Ctrl-Alt-T. Erstellt unter dem PlyrWild, ein neues Team und nennt es *Goblin_attackers*

Edit Team Template

Identity | Behavior | Generic

Goblin_attackers Maximum: 3

Owner: PlyrWild

Home Position: <none>

Destination: <none>

AI Type: ATTACK_TEAM

Production:

Condition: Build_base_attackers

Execute associated actions

Priority: 10 Success Priority: 0

Build for: 60 Secs Failure Priority Decrease: 0

Min:	Max:	Unit Type:	Properties
2	2	GoblinFighterHorde	...
2	2	GoblinArcherHorde	...
0	0	<none>	...
0	0	<none>	...
0	0	<none>	...
0	0	<none>	...
0	0	<none>	...

Recruitment Options:

Automatically reinforce whenever possible

Team members are AI Recruitable

Team Description:

OK Abbrechen Übernehmen

Unter *Maximum* stellt ihr die Anzahl der Instanzen ein die von diesem Team gleichzeitig auf der Map aktiv sein sollen.

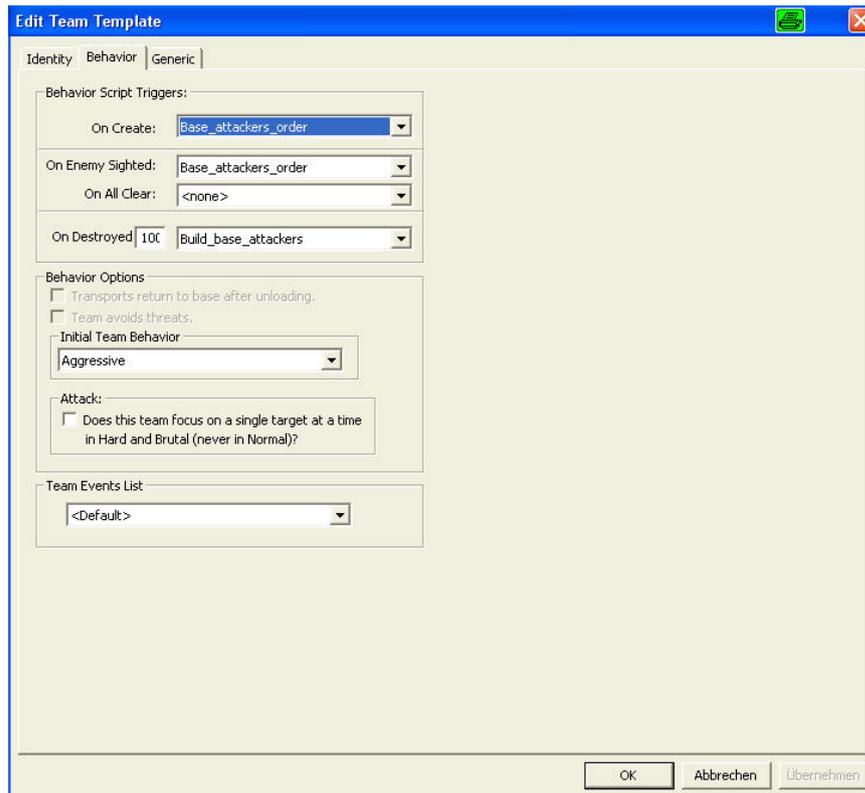
Home Position definiert einen WayPoint an dem sich das erstellte Team versammelt.
Destination ist das Ziel zu dem das Team wandern würde.

Bei den *Production Conditions* setzt ihr die Subroutine ein welche die Bedingung enthält um das Team zu erstellen.

Priority gibt an wie wichtig es ist das Team zu erstellen.

Build for sagt aus wie lange die KI versuchen soll das Team zu erstellen, wird hier „0“ eingetragen wird es nie gebaut.

Unter *Properties* könnt ihr dem Team seine Upgrades mitgeben, die wir aber zur Sicherheit per Script vergeben, falls die KI die erforderlichen Upgrade-Voraussetzungen nicht erfüllt hat! Wenn wir nicht wollen, dass die KI cheated müssten wir das auch noch zuvor per Bedingung abfragen, aber das würde diese Einführung unnötig verkomplizieren. Ähnlich sieht es mit den erforderlichen Credits aus, die für das Erstellen des Teams fällig werden, auch das müsste per Bedingung abgefangen werden,..!



Im *Behavior* des Teams stellen wir ein wie es sich verhalten soll.

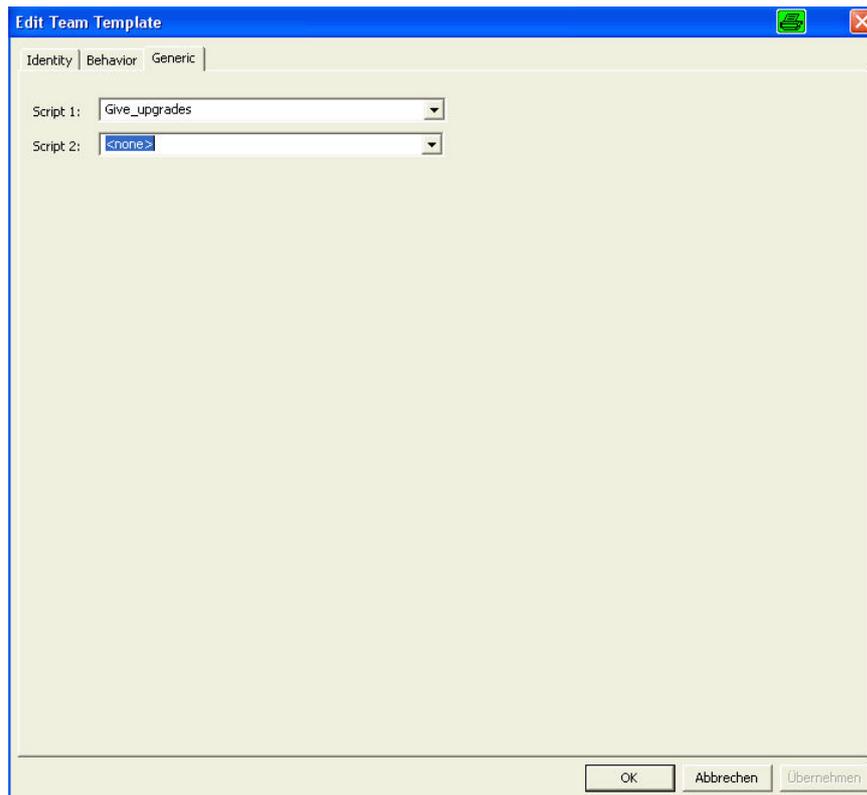
Nachdem das Team gebaut wurde *On Create* fügen wir die Subroutine *Base_attackers_order* ein, das Team wird also anfangen zu „jagen“, dabei wird es alle 20 Sekunden daran erinnert.

On Enemy Sighted führt das Team die dort hinterlegte Subroutine aus, in unserem Fall noch mal die *Base_attackers_order*.

Mit *On Destroy* *<100>* legen wir fest, dass bei der vollständigen Zerstörung des Teams ein neues gebaut werden soll. Es ist auch möglich andere Werte zu verwenden, z.B. 50, würde mit dem Bau beginnen, wenn die Hälfte des Teams zerstört wurde. Es sind aber auch andere Bedingungen möglich, eurer Kreativität sind kaum Grenzen gesetzt.

Initial Team Behavior legt das grundsätzliche Verhalten des Teams fest und gibt ihm einen erweiterten Sichtradius, wenn ihr den *Behavior* auf *Alert* oder *Aggressive* einstellt.

Im *Generic* fügen wir das Script ein welches dem Team die vorgesehenen Upgrades gibt. Das könntet ihr aber auch im *On Create* einsetzen.



5. Gebäude bauen lassen

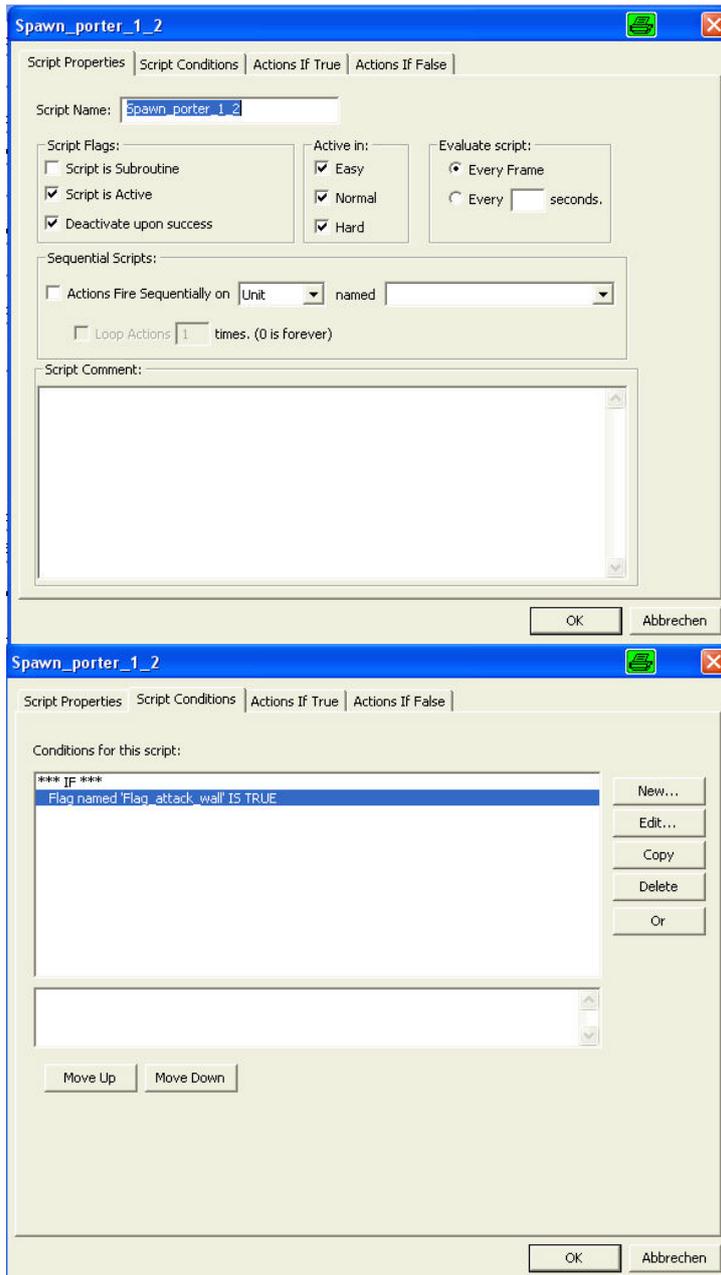
Die hier gezeigte Variante ist die einfachste, denn es soll ja verständlich sein was die Scripte bewirken, ...es geht natürlich auch anders.

Im folgenden Beispiel verwende ich den *PlyrMordor*, was der Wirkungsweise des Scripte aber keinen Abbruch leistet. Es werden zwei *MordorPorter* auf die Map gesetzt, die einen *OrcPit*, einen *TrollPit* und drei *SlaughterHouses* bauen, also eine kleine Base. Erklärt wird aber nur der Bau des *OrcPit*.

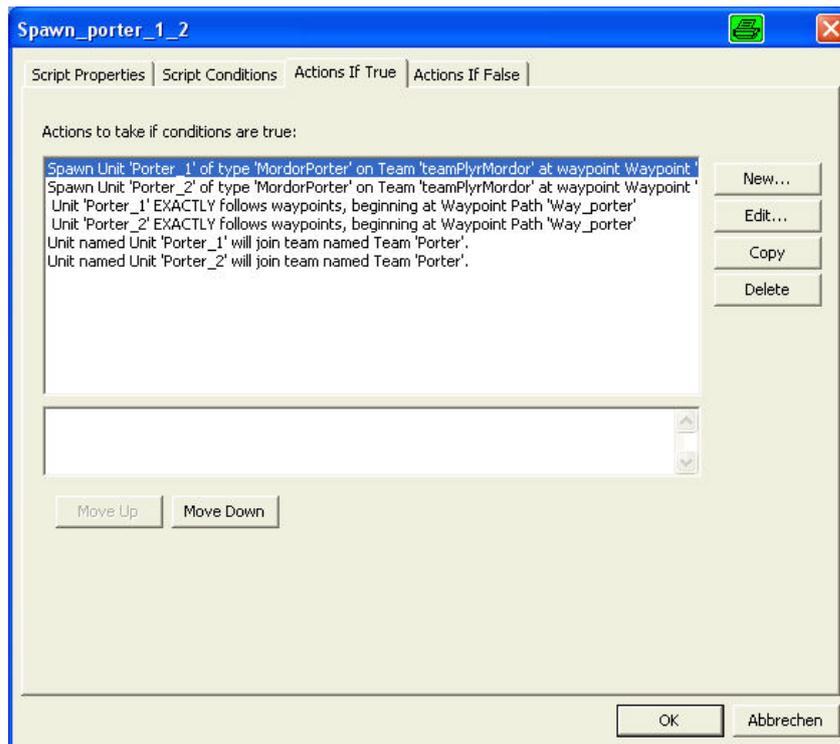
Als erstes legt ihr einen *SpawnPoint* und einen *WayPath* fest, an welchem die Porter per Spawn auf die Map gesetzt werden, danach folgen sie den *WayPath* bis zu dessen Ende.

Dort haben wir einen *WayPoint* gesetzt an dem der Porter den *OrcPit* errichten soll. Um die genaue Ausrichtung (*Angle*) des *OrcPit* festzulegen, setzt ihr ihn zuvor auf die Map und notiert euch den *Angle*.





In den *Conditions* verwende ich den Zustand eines Flags, ihr könnt dort aber auch *TRUE* wählen, wenn es keine Bedingung geben soll.



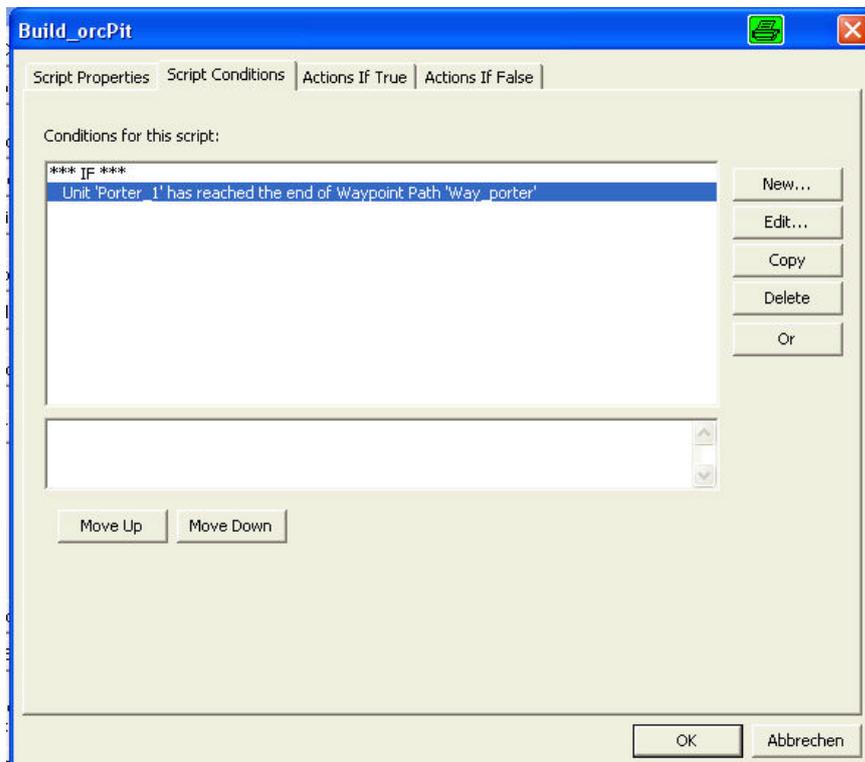
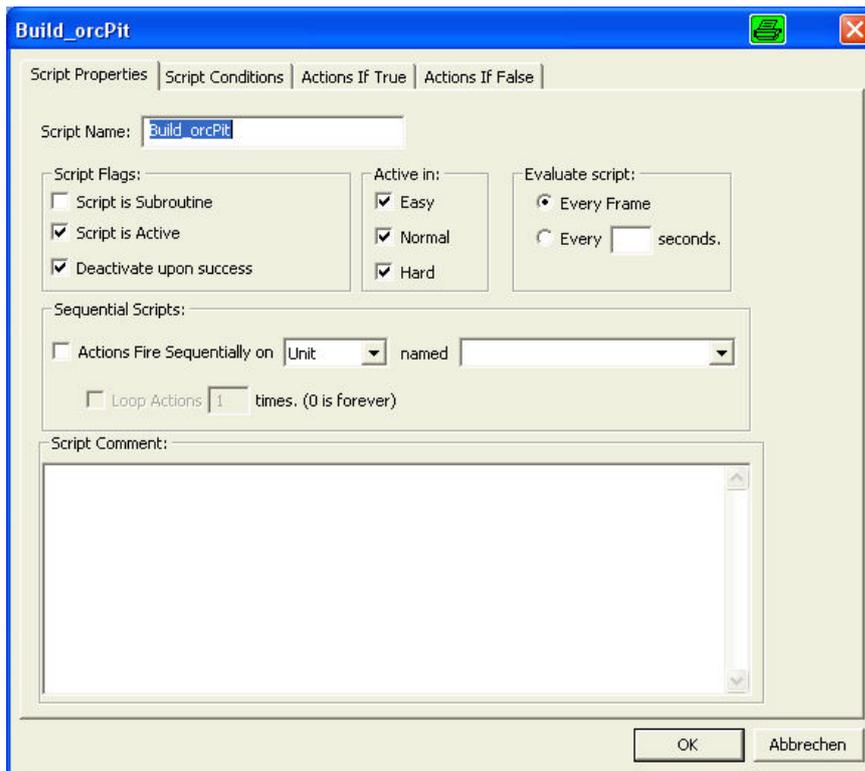
Pfad zu den Scripten:

Unit (include Structures) – Other/Spawn/Spawn – named Unit on Team at a Waypoint
Unit (include Structures) – Action/Move/Unit follow a Waypoint--path exactly
Unit (include Structures) – Other/Allegiance/Transfer a specified Unit to the control of Player

Die ersten beiden Scripte setzen die beiden Porter auf die Map.

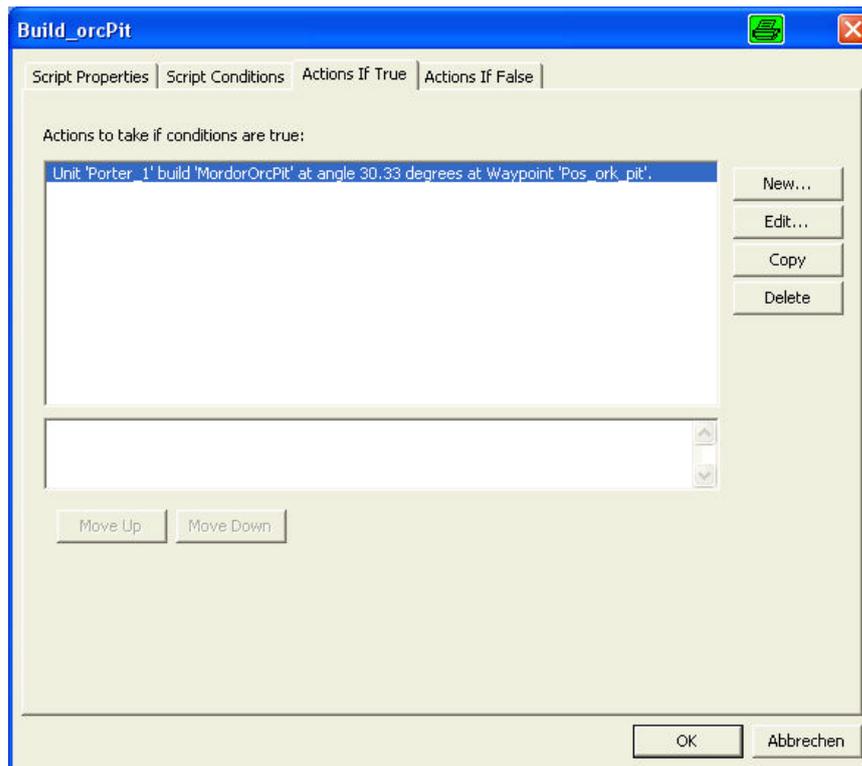
Script drei und vier bewegen die Porter zu ihrem Ziel.

Die letzten beiden Scripte integrieren die Porter zu einem Team. Das müsst ihr nicht unbedingt machen, da ich aber die Gebäude mit Namen versehen habe ist es einfacher die Einheiten in einem Team zu haben. Der Einfachheit halber werde ich aber in diesem Tutorial nicht darauf eingehen.



Pfad zum Script:

Unit (include Structures)/Area or Path/Unit has reached the end of a WayPoint path



Pfad zum Script:

Unit (include Structures) – Action/CommandButton/Build Thing at WayPoint

Sobald das Militärgebäude steht das in der Lage ist euer festgelegtes Team zu produzieren, sprudeln Teams aus dem Gebäude, bis die festgelegte Anzahl von Instanzen erreicht ist. Das Gebäude wird solange Teams nachbauen bis es zerstört wird.

So, das war es auch schon! Nun wünsche ich euch viel Spaß beim Ausprobieren.

Euer ERNESTO vom Edain Team
Im Dezember 2009