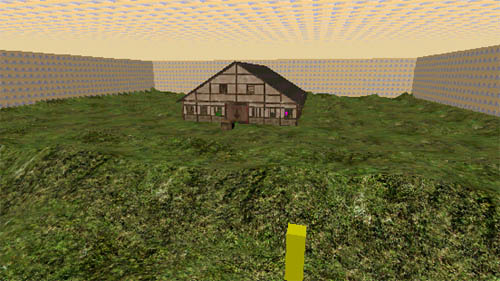
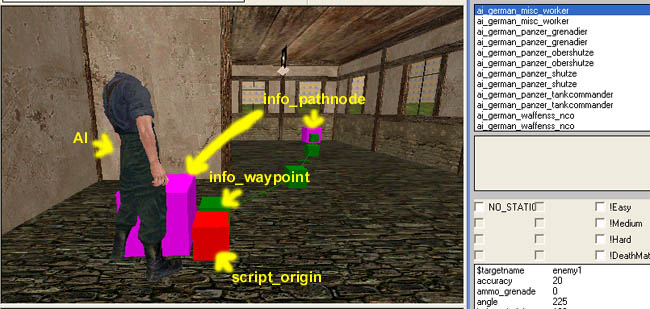
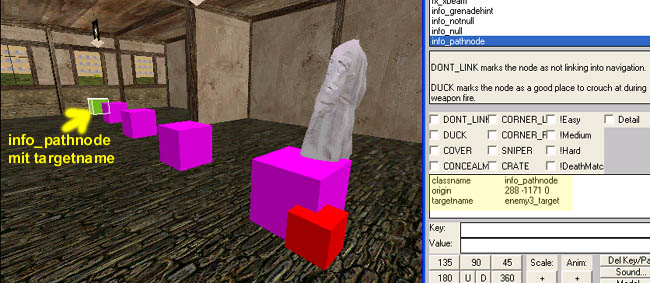
**AI Spawning System (SP)**

© 3/2003 T.Reim | Dieses Tutorial ist für Singleplayer Maps.

Der Grundgedanke zum Tutorial ist eine Map mit Script zu erstellen, in der die Anzahl der Gegner eingestellt werden kann, sowie das Aussehen, die Waffe, und die Angriffsart. Ihr findet hier eine Anwendung zu folgenden Script- und Spiel-Befehlen. Spielbefehle:  
spawn, runto, walkto, waittill movedone, waittill death, enable\_ai, disable\_ai, AI Parameter, missioncomplete  
Scriptbefehle:  
if else, switch, Variable hochzählen ++, Variable vergleichen <, Variablen/Werte übergeben, Parameter ändern

1. Erstellt eine Skybox mit einem Haus (mindestens 2 Fenster und eine Tür) und dem ganzen drumherum, und macht die worldspawn Einstellung sowie den playerstart in einigem Abstand vor das Haus. Direkt vor dem Haus erstellt ihr noch eine 64x64x64 Kiste.  
   
2. In dem Raum des linken Fensters setzt ihr einen script\_origin mit dem $targetname = enemy1\_spawner. (den script\_origin dort hinsetzen wo ihr ihn nicht sehen könnt als Player, da dort die AI gespawnt werden) Von dem script\_origin weg erstellt ihr einen info\_waypoint path mit den targetname = enemy1\_path bis zum Fenster. Am ende erstellt ihr einen normalen info\_pathnode. Neben dem script\_origin erstellt ihr noch einen AI german worker mit dem $targetname = enemy1 und noch einen normalen info\_pathnode  
   Ungefähr so:  
   
3. In dem Raum oder Gang wo die Tür nach draussen ist, erstellt ihr auch ein script\_origin mit $targetname = enemy2\_spawner. Von dem script\_origin weg, erstellt ihr wieder einen info\_waypoint path mit targetname = enemy2\_path, der durch die Tür nach draussen bis vor die Kiste geht. Vor der Kiste wo der letzte waypoint liegt setzt ihr ein info\_pathnode mit den Parametern duck und crate im Entity Menü, sowie den angle angeben, in der Richtung die auf die Kiste zeigt. Am script\_origin erstellt ihr einen AI german soldier mit $targetname = enemy2 und einen normalen info\_pathnode.  
   
4. In dem rechten Raum erstellt ihr ein script\_origin mit $targetname = enemy3\_spawner. Jetzt wollen wir mal den AI nicht über einen waypoint path laufen lassen, sondern über info\_pathnode. Dazu erstellen wir eine Bahn an pathnodes bis zum Fenster. Dem letzten info\_pathnode am Fesnter geben wir den targetname = enmy3\_target im Entity Menü. Hier erstellen wir auch neben dem script\_origin einen AI german scientist mit $targetname = enemy3. Das wars auch schon im Radi.  
   

**Das Script**

Am besten ihr öffnet parallel zum lesen dieser Seite das Script in einer Textdatei ([test\_ai\_spawning\_system.txt](http://www.steinhuder-meer-portal.de/wp-content/uploads/mohaa/test_ai_spawning_system.txt" \o "neues Fenster: Script als Text-Datei" \t "_blank)), da ich hier die Zeilen im einzelnen erkläre. Wir fangen in der Zeile main an:

main:  
Anfang des main thread für den Level, der die Global Scripte die für den Level benötigt werden enthält

exec global/loadout.scr maps/test\_ai\_spawning\_system.scr  
Befehl der dem Spiel angibt welches Script zur Map gehört

exec global/ai.scr  
ruft das Global Script für AI auf, dieses wird immer benötigt wenn AI in der Map vorhanden sind

level waittill prespawn  
Befehl zum warten bis der Level geladen ist

level.script = maps/test\_ai\_spawning\_system.scr  
Befehl der dem Spiel angibt welches Script zum Level gehört

exec global/ambient.scr obj\_team3  
ruft das Global Script für den Ambient Sound auf (music/eureMap.mus)

level waittill spawn  
Befehl zum warten bis der Level gestartet ist

// Anzahl der enemy  
level.e1 = 10  
gibt an das die Levelvariable e1 gleich 10 ist, dies ist die Anzahl der Gegner die gespawnt werden, hier müsst ihr einstellen wieviele ihr haben wollt, die 3 AI die schon in der Map sind müsst ihr von dieser Zahl wieder abziehen oder bei level.e2 dazu zählen

level.e2 = 0  
gibt an das die Levelvariable e2 gleich 0 ist, diese benötigen wir damit wir die gespawnten AI hochzählen können, um später zu kontrollieren ob die eingestellte Anzahl erreicht ist

$player item weapons/Springfield.tik  
$player item weapons/ThompsonSMG.tik  
$player item weapons/BAR.tik  
$player item weapons/M2frag\_grenade\_sp.tik  
$player item weapons/bazooka.tik  
$player ammo rifle 300  
$player ammo smg 300  
$player ammo mg 400  
$player ammo grenade 6  
Zuweisung der Waffen und der Munition für den Spieler

$player useweaponclass smg  
gibt an welche Waffe der Spieler beim Mapstart in der Hand halten soll, hier ist es die Thompson SMG

$player nodamage  
setzt den Spieler auf Unverwundbarkeit, diese Zeile müsst ihr beim fertigen Script wieder rausnehmen, ich benutze diesen Befel nur zum Maps testen. Den Befehl nodamage könnt ihr auch auf andere Objekte wie AI oder Flugzeuge anwenden z.B. $stuka1 nodamage

$enemy1 exec global/disable\_ai.scr  
$enemy2 exec global/disable\_ai.scr  
$enemy3 exec global/disable\_ai.scr  
mit dem Global Script disable\_ai werden die AI ausgeschaltet und reagieren somit nicht auf Angriffe und greifen selber auch nicht an

thread enemy\_start  
ruft den thread enemy\_start auf, damit die AI eingeschaltet werden und loslaufen

end  
beendet den main thread

/////////////////////////////////////////// enemy start  
enemy\_start:  
Anfang des enemy\_start thread, in diesem thread schalten wir die ersten 3 AI ein, und rufen die nächsten threads auf damit die AI wenn sie tot sind neu gespawnt werden

wait 1  
wartet 1 sekunde bis die nächste Zeile verarbeitet wird

$enemy1 thread deathsystem1 enemy1 $enemy1\_path $enemy1\_spawner 1  
ruft den thread deathsystem1 für den enemy1 auf, und übergibt die Werte enemy1, $enemy1\_path, $enemy1\_spawner und die Zahl 1 ; eine Erklärung kommt in dem thread spawnsystem1

$enemy1 runto $enemy1\_path  
aufrufen des Befehls für das rennen zum Ziel, das Ziel ist der info\_waypoint path mit dem targetname $enemy1\_path

$enemy1 waittill movedone  
warten bis der AI $enemy1 am ende des waypoint path angekommen ist

$enemy1 exec global/enable\_ai.scr  
einschalten des AI $enemy1 damit er auf Angriffe reagiert oder selber angreift

wait 1  
wartet 1 sekunde bis die nächste Zeile verarbeitet wird

$enemy2 thread deathsystem1 enemy2 $enemy2\_path $enemy2\_spawner 2  
ruft den thread deathsystem1 für den enemy2 auf, und übergibt die Werte enemy2, $enemy2\_path, $enemy2\_spawner und die Zahl 2 ; eine Erklärung kommt in dem thread spawnsystem1

$enemy2 walkto $enemy2\_path  
aufrufen des Befehls für das gehen zum Ziel, das Ziel ist der info\_waypoint path mit dem targetname $enemy2\_path

$enemy2 waittill movedone  
warten bis der AI $enemy2 am ende des waypoint path angekommen ist

$enemy2 exec global/enable\_ai.scr  
einschalten des AI $enemy2 damit er auf Angriffe reagiert oder selber angreift

wait 1  
wartet 1 sekunde bis die nächste Zeile verarbeitet wird

$enemy3 thread deathsystem1 enemy3 $enemy3\_target $enemy3\_spawner  
ruft den thread deathsystem1 für den enemy3 auf, und übergibt die Werte enemy3, $enemy3\_target, $enemy3\_spawner und keine Zahl ; eine Erklärung kommt in dem thread spawnsystem1

$enemy3 runto $enemy3\_target  
aufrufen des Befehls für das rennen zum Ziel, das Ziel ist der info\_pathnode mit dem targetname $enemy3\_target

$enemy3 waittill movedone  
warten bis der AI $enemy3 am ende des waypoint path angekommen ist

$enemy3 exec global/enable\_ai.scr  
einschalten des AI $enemy3 damit er auf Angriffe reagiert oder selber angreift

end  
beendet den enemy\_start thread

/////////////////////////////////////////// general deathsystem  
// local.d1 = ai name  
// local.d2 = ai target  
// local.d3 = ai spawner origin  
// local.d4 = ai uniform attack type gun  
//////////////////////////////////////////  
diese Zeilen geben an welche Werte über die local Variablen übergeben werden und dienen nur als Gedankenstütze

deathsystem1 local.d1 local.d2 local.d3 local.d4:  
anfang des thread deathsystem1 und bereitstellen der local Variablen local.d1, local.d2, local.d3, local.d4, eine Erklärung was welche local Variable beinhaltet steht in den Zeilen drüber. Dieser thread überwacht die 3 AI ob sie tot sind und leitet dann den thread ein in dem die AI neu gespawnt werden

self waittill death  
wartet bis das Objekt self, in unserem Fall einer der 3 AI tot ist

wait .1  
wartet eine zehntel sekunde bis die nächste Zeile verarbeitet wird

self.targetname = ""  
setzt den targetname des Objektes self auf nichts (einer der 3 AI)

wait 3  
wartet 3 sekunden

thread spawnsystem1 local.d1 local.d2 local.d3 local.d4  
ruft den thread spawnsystem1 auf, und übergibt die Werte die in den local Variablen enthalten sind

end  
beendet den thread deathsystem1

/////////////////////////////////////////// general spawnsystem  
// local.name = ai name  
// local.s2 = ai target  
// local.s3 = ai spawner origin  
// local.s4 = ai uniform attack type gun  
// enemy type 1 = worker StG44  
// enemy type 2 = soldier MP40  
// enemy type default = scientist Mauser KAR 98K  
//////////////////////////////////////////  
diese Zeilen geben an welche Werte über die local Variablen übergeben werden und dienen nur als Gedankenstütze

spawnsystem1 local.name local.s2 local.s3 local.s4:  
Anfang des thread spawnsystem1 und bereitstellen der local Variablen, eine Erklärung der Variablen steht in den Zeilen oben drüber. Da local Variablen nur für einen thread gelten, müssen diese wieder angegeben werden aber mit einem anderen local Variablennamen (s3, s2…)

if (level.e2 < level.e1)  
if Abfrage, mit der Bedingung level Variable e2 kleiner als level Variable e1, das heisst wenn level.e2 kleiner als level.e1 ist werden die folgenden Befehle ausgeführt, also die Anzahl der gespawnten enemies kleiner als die vorgegebene Anzahl der zu spawnenden enemies ist

{  
Anfang der Befehle wenn die Bedingung zutrifft

level.e2++  
level Variable e2 wird um 1 erhöht, da in den folgenden Befehlen ein AI gespawnt wird

switch (local.s4)  
switch Befehl der local Variable s4, hier wird eine Auswahl getroffen in der wir angeben wollen welche Uniform, Waffe und Angriffsart der AI haben soll. Die local Variable s4 hat den Wert 1, 2 oder keine Angabe. Diese haben wir in dem thread enemy\_start für jeden AI am ende der thread Befehlszeile angegeben.

{  
Anfang der switch Befehle

case 1:  
Wert 1 der local Variable s4, wenn local.s4 1 ist, werden diese Befehle verarbeitet

spawn models/human/german\_misc\_worker.tik "targetname" local.name "type\_attack" "turret" "gun" "StG44"  
spawnt einen AI mit den angegebenen Parametern (diese findet ihr im Radi in den AI Einstellungen eines normalen AI). Die local Variable name gibt an welchen targetname der AI haben soll, den targetname haben wir im thread enemy\_start mit übergeben.

break  
bricht die Befehlskette der Zuteilung zum case 1 ab, hiernach werden die Befehle nach dem Ende der gesamten switch Befehle weiter verarbeitet

case 2:  
Wert 2 der local Variable s4, wenn local.s4 2 ist, werden diese Befehle verarbeitet

spawn models/human/german\_wehrmact\_soldier.tik "targetname" local.name "type\_attack" "turret" "gun" "MP40"  
spawnt einen AI mit den angegebenen Parametern (diese findet ihr im Radi in den AI Einstellungen eines normalen AI). Die local Variable name gibt an welchen targetname der AI haben soll, den targetname haben wir im thread enemy\_start mit übergeben.

break  
bricht die Befehlskette der Zuteilung zum case 2 ab, hiernach werden die Befehle nach dem Ende der gesamten switch Befehle weiter verarbeitet

default:  
default Wert der local Variable s4, wenn local.s4 nicht angegeben ist oder nicht auf eine der anderen case zutrifft, werden diese Befehle verarbeitet

spawn models/human/german\_misc\_scientist.tik "targetname" local.name "type\_attack" "cover" "gun" "Mauser KAR 98K"  
spawnt einen AI mit den angegebenen Parametern (diese findet ihr im Radi in den AI Einstellungen eines normalen AI). Die local Variable name gibt an welchen targetname der AI haben soll, den targetname haben wir im thread enemy\_start mit übergeben.

break  
bricht die Befehlskette der Zuteilung zum default ab, hiernach werden die Befehle nach dem Ende der gesamten switch Befehle weiter verarbeitet

}  
Ende der switch Befehle, an diese Stelle wird nach jedem break gesprungen und die folgenden Befehle ausgeführt

local.name exec global/disable\_ai.scr  
die local Variable name beinhaltet den targetname des neuen AI und wird somit genutzt um alle weiteren Einstellungen nach dem spawnen des AI zu machen, hier wird der AI mit dem Global Script disabled (ausgeschaltet), das heisst der AI reagiert nicht auf Angriffe und greift auch niemanden an

local.name.origin = local.s3.origin  
damit das Spiel auch weiß wo der AI erscheinen soll, müssen wir ein origin angeben. Diese haben wir im Radi erstellt und $enemy1\_spawner usw. benannt. Um auf den Origin Wert eines Objektes zugreifen zu können wird der Entity Wert origin im Script mit dem targetname und einem Punkt verknüpft. Diese Befehlszeile heisst das der origin Wert des eben gespawnten AI den origin Wert hat, des in der local Variable s3 übergebenen Originnamens.

local.name thread deathsystem1 local.name local.s2 local.s3 local.s4  
aufrufen des thread deathsystem1 und übergeben der local Variablen, dieser Aufruf übergibt alle Werte wieder an die Überwachung im deathsystem1 thread, damit ein neuer AI gespawnt wird wenn der eben gespawnte AI tot ist.

// ai parameter  
local.name.mindist = 128  
local.name.health = 100  
local.name.maxdist = 2048  
local.name.leash = 512  
local.name.fixedleash = 0  
local.name.sight = 2048  
local.name.noticescale = 100  
local.name.accuracy = 20  
local.name.hearing = 2048  
local.name.ammo\_grenade = 3  
einstellen der AI Parameter, hier wieder verknüpft mit dem Punkt, es gibt noch mehr Parameter die ihr einstellen könnt, diese findet ihr in den AI Parameter Fenster im Radi, ihr braucht nur den Parameter-Namen mit dem Punkt und dem targetname local.name verknüpfen (die Einstellung wird nicht im Radi vorgenommen, dort könnt ihr nur nachschauen welche es noch gibt)

local.name runto local.s2  
aufrufen des Befehls für das rennen zum Ziel, das Ziel ist der info\_waypoint path oder der info\_pathnode mit dem targetname local.s2, diese haben wir im thread enemy\_start mit übergeben für jeden einzelnen AI

local.name waittill movedone  
warten bis der AI am ende des waypoint path oder am pathnode angekommen ist

local.name exec global/enable\_ai.scr  
einschalten des AI damit er auf Angriffe reagiert oder selber angreift

}  
beenden der erfüllten Bedingung aus der if Abfrage

else  
aufrufen der Befehle wenn die Bedingung der if Abfrage nicht erfüllt ist, also die Anzahl der enemy erreicht ist

{  
Anfang der Befehle wenn die Bedingung nicht erfüllt ist.

wait 20  
wartet 20 Sekunden

thread level\_end  
aufrufen des thread level\_end, hier kann auch was anderes kommen, je nachdem wie ihr das Spawnsystem in eurer Map einstezt

}  
Ende der if Abfrage wenn die Bedingung nicht erfüllt ist

end  
Ende des thread spawnsystem1

/////////////////////////////////////////// level ende  
level\_end:  
Anfang des thread level\_end, der die Befehle für das erreichen des Levelendes beinhaltet

exec global/missioncomplete.scr test\_ai\_spawning\_system  
aufrufen des Global Scriptes für den Abschluss eines Levels, hier müsst ihr den Mapnamen eintragen von der Map die als nächstes geladen werden soll. Wenn ihr den Mission success Bildschirm nicht haben wollt, müsst ihr am ende der Zeile noch eine 1 schreiben z.B. “exec global/missioncomplete.scr test\_meinemap 1″ (der aktuelle Health und die Waffen werden dadurch auch auf den nächsten Level übergeben, wenn ihr beim nächsten Level die Waffen nicht haben wollt schreibt ihr vor der neuen Waffenzuweisung im main thread des Mapscriptes $player takeall und $player heal 1)

end  
beendet den thread level\_end

**thread Funktionsprinzip**

main:  
aufrufen der Global Scripte, setzen der benötigten Variablen, Player Waffenzuweisung, AI ausschalten und starten der AI einleiten

enemy\_start:  
hier werden die 3 AI zu ihrem Ziel geschickt und eingeschaltet. Sowie sozusagen für das Script konfiguriert, die targetnamen der AI, der script\_origin und der info\_waypoint/info\_pathnode werden als Wert für die weitere Verarbeitung auf den thread deathsystem1 übergeben

deathsystem1:  
überwachen des AI bis er tot ist, und dann wird der thread spawnsystem1 für das spawnen des nächsten AI aufgerufen

spawnsystem1:  
Kontrolle ob die Höchstanzahl an AI erreicht ist, wenn nicht wird der nächste AI gespawnt und die AI Parameter gesetzt und der Ziellauf des AI eingeleitet, die Werte werden wieder zurück an den deathsystem1 übergeben damit der AI überwacht wird. Wenn die Höchstanzahl erreicht ist wird das Levelende eingeleitet.

level\_end:  
Abschluss des Levels mit dem Global Script, laden des nächsten Levels

Download: [Beispielmap – AI Spawning System](http://www.steinhuder-meer-portal.de/wp-content/uploads/mohaa/test_ai_spawning_system.pk3)