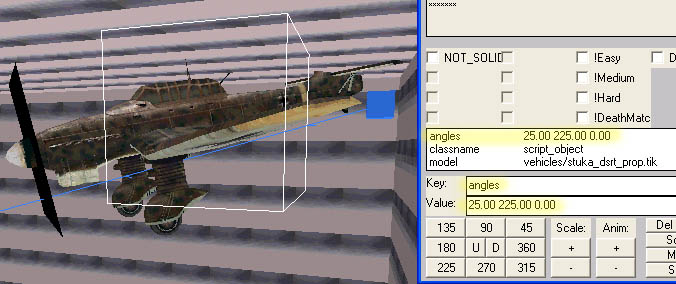
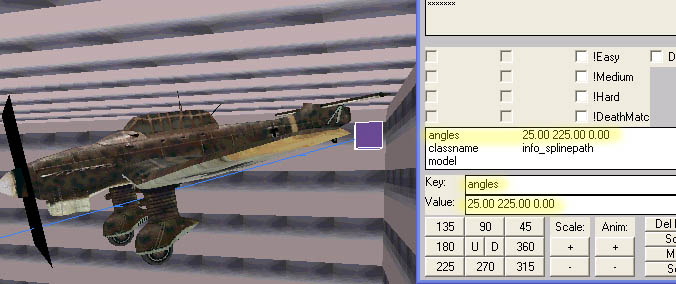
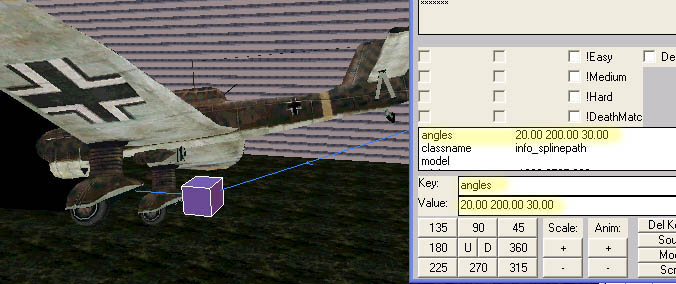
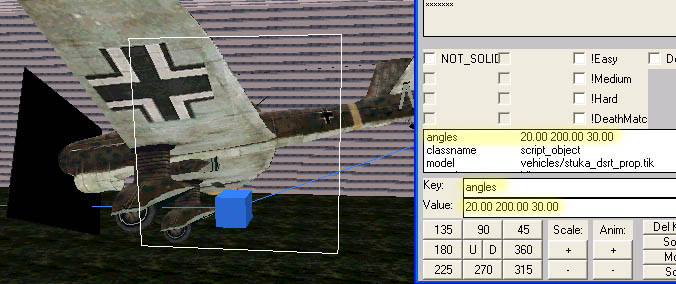
**Flieger abschiessen (SP/MP)**

© 1/2003 T.Reim | Dieses Tutorial ist für Singleplayer und Multiplayer Maps.

Jetzt wollen wir mal etwas vom Himmel zaubern, ein Flugzeug das wir abschiessen können. Ihr könnt es in jeder Map-Art einsetzen, auch als Objective. Anfangen werden wir im Radiant. (ein ähnliches Beispiel findet ihr in der SP map m1l3b)

1. Erstellt eine große Skybox (8000×3000; hoch 1500) mit irgendeiner Fussbodentextur, setzt euren Playerstart, und baut euch eine kleine MG Stellung auf, mit Sandsäcken oder Mauer. (wie alles geht steht ja in den anderen Tutorials) In diesem Beispiel ist kein Nebel usw. vorhanden. In eurer Map wo ihr dieses einsetzen wollt, sollte Nebel enthalten sein und ihr solltet im Spiel nicht den Rand der Map sehen können, damit das Flugzeug auch den Eindruck macht als ob es von weitem angeflogen kommt.
2. Erstellt einen info\_splinepath quer durch eure Skybox mit mehreren Punkten, der die Flugbahn des Flugzeugs ist, wie in den anderen Tutorials beschrieben, aber mit folgender Ausnahme: die model bezeichnung bleibt leer. (targetname = flugbahn)
3. Nun wollen wir die Flugbahn einstellen. Erstellt ein script\_object am Anfang des splinepath. Unter model gebt ihr folgendes ein: vehicles/stuka\_dsrt\_prop.tik (jetzt sollte die Stuka im 3D Fenster erscheinen)  
   Ihr müsst jetzt die angles werte für das Flugzeug einstellen, passend zur Flugbahn (nicht die angle werte, man beachte das “s”). Dazu setzt ihr das script\_object neben jeden splinepath punkt und stellt die angles Werte so ein wie das Flugzeug an dieser Stelle fliegt. Bei jedem splinepath Punkt übertragt ihr die jeweiligen angles Werte dem splinepath Punkt. (der erste Winkel ist die Neigung nach unten, der zweite die Drehung von oben gesehen und die dritte die Drehung um die eigene Achse in Kurven)  
   am Startpunkt des splinepath:  
     
     
   an einem anderen Punkt des splinepath:  
     
     
   Wenn Ihr alle durch habt, schiebt ihr das script\_object (euer Flugzeug) wieder an den Anfang des splinepath und übernehmt die angles Werte des ersten splinepath Punktes wieder zurück für das Flugzeug, damit es wieder die Winkel hat für den Start des Fluges. (ist alles etwas mühselig, ich weiß)
4. Jetzt noch schnell diese Einträge im Entity Menü des script\_object machen (euer Flugzeug)  
   $inair = inair\_trigger1  
   $path = flugbahn << der targetname des splinepath, den wir in 2. erstellt haben  
   $targetname = stuka1  
   angles = . . . << die selben des ersten splinepath Punktes
5. und das wars im Radi. Jetzt kommt das Script dazu.  
   main:  
   setcvar "" ""  
   setcvar "" ""  
   setcvar "" ""  
   setcvar "" ""  
   setcvar "" ""  
   setcvar "" ""  
   setcvar "g\_scoreboardpic" ""  
   if(level.roundbased)  
   {  
   // set the parameters for round based match  
   level.dmrespawning = 0 // 1 or 0  
   level.dmroundlimit = 5 // round time limit in minutes  
   level.clockside = kills // set to axis, allies, kills, or draw  
   }  
   level waittill prespawn  
   //\*\*\* Precache Dm Stuff  
   exec global/DMprecache.scr  
   exec global/exploder.scr  
   level.script = "maps/dm/flieger\_abschiessen.scr"  
   exec global/ambient.scr obj\_team3  
   thread global/exploder.scr::main  
   // Flugzeug thread einleiten  
   $stuka1 thread stuka1a\_setup <<< aufrufen des thread für das Flugzeug (targetname des script\_object)  
   level waittill spawn  
   level waittill roundstart  
   end  
   /////////// Flugzeug setup - loop - damage - fire  
   stuka1a\_setup:  
   self.IsInAir = 1  
   self.GoingDown = 0  
   self setsize "-100 -100 -100" "100 100 100"  
   self.mins = ( -100 -100 -100 )  
   self.maxs = (100 100 100)  
   self.origin = self.path.origin  
   self.angles = self.path.angles  
   self hide  
   thread flugbahn\_loop  
   self thread DamageManager  
   end  
   flugbahn\_loop:  
   while ( (self.GoingDown != 1) && (self != NULL) )  
   {  
   if (self == NULL)  
   end  
   self solid  
   self takedamage  
   self loopsound m1l3b\_stuka\_idle <<< Motorengeräusch des Fliegers, könnt ihr auch einen anderen nehmen  
   self show  
   self followpath self.path  
   self waitmove  
   if ( (self == NULL) || (self.GoingDown == 1) )  
   end  
   self notsolid  
   self hide  
   self nodamage  
   self stoploopsound  
   wait 10 <<< hier müsst ihr die Zeit einstellen, die der Flieger braucht um zum Anfang zurück zu fliegen  
   }  
   end  
   DamageManager:  
   local.damage\_events = 0  
   while (local.damage\_events < 10) <<< hier die Anzahl der Treffer einstellen, bis der Flieger explodieren soll  
   {  
   self waitTill damage  
   local.damage\_events ++  
   println "z: local.damage\_events " local.damage\_events  
   wait .5  
   }  
   self.GoingDown = 1  
   self playsound explode\_plane\_flying <<< Sound für die Explosion, könnt ihr auch einen andere nehmen  
   self thread planeonfire  
   end  
   planeonfire:  
   local.temp = spawn effectentity model models/fx/fx\_flying\_stuka\_explosion.tik  
   local.temp notsolid  
   local.temp anim start  
   local.offset = (0 0 92)  
   local.temp.origin = (self.origin + local.offset)  
   local.start\_time = level.time  
   while(level.time < local.start\_time + 5) <<< hier die Zeit einstellen, wie lange der Flieger nach der Explosion noch zu sehen sein soll (im Nebel verschwindet)  
   {  
   local.temp.origin = (self.origin + local.offset)  
   waitframe  
   }  
   local.temp remove  
   self stoploopsound  
   self hide  
   end

Download: [Beispiel Map – Flieger abschiessen](http://www.steinhuder-meer-portal.de/wp-content/uploads/mohaa/flieger_abschiessen.pk3)