**Bomb dropping planes tutorial**

For this tut i assume you have basic knowlege of Radient, it's interface, and techniqes for brushes.

This tut should show you how to create planes that fly on a path you mark and drop a bomb where you wish. you can just imagine the uses...., in any case, to start out, make a simple map, just a skybox with the bottom textured some grass texture really. since this is just a tut, use **ambientlight 70 70 70** to light up our map... make a line of **info>splinepaths** from one end of the map to another. make sure that when using splinepaths that they all have the angle specified toward the next splinepath in the sequence until you're done. Okay, now connect them all together, you can use the common 't' function in [MOHRadiant](http://gronnevik.se/rjukan/index.php?n=Main.MOHRadiant) where you targetname and target a node with **t#** and the next number will appear in sequence once you hit spacebar (clone). Once you've connected them all together, highlight the first one in the series and give it the following keys/values

|  |  |
| --- | --- |
| **Key** | **Value** |
| #set | 1 |
| targetname | bomberpath |
| model | vehicles/p47fly.tik |
| $mdl | vehicles/p47fly.tik |
| angles | 0.00 0.00 0.00 |

Now make a **trigger\_multiple** and put it somewhere in your map. it doesn't matter where. make an arch and place it under it for reference in-game so you know where to go to hit it. give it the keys/values....

|  |  |
| --- | --- |
| **Key** | **Value** |
| $targetname | bombertrigger |
| target | 2 |
| #set | 1 |
| cnt | 1 |

Okay: now that that's done, place an **info>player>start** somewhere in the map, and put the following script in the script file...

main:

exec global/auto.scr

level waittill prespawn

exec global/ambient.scr m2l2

exec global/bomber.scr

$world farplane 5000//fog is optional. quote it out if you like.

$world farplane\_color (.333 .333 .329)

level waittill spawn

level.flyplane = 1

$player item weapons/colt45.tik

$player ammo pistol 200

end

Okay: you can compile at this stage but the plane won't drop any bombs as of now. it'll just spawn and fly through it's path and remove when finished with the move. To make it drop a bomb, we have to make good use of the fact that the plane is big and hollow. So... We use a tricky little optical illusion. make another splinepath path that overlaps the plane's (make sure to leave a but of room between them for now so you can changes their properties as we go along), now change the one overlapping the plane's keys/values to:

|  |  |
| --- | --- |
| **Key** | **Value** |
| targetname | bombpath |
| model | ammo/us\_bomb.tik |
| #set | 1 |

As with the plane connect the first one to the entire path... but bend this path at the middle so that it hits a mid-point splinepath and curves downward. Adjust all the bombpath's splinepaths so that they curve as gracefully as possible toward the ground. Make sure the last one just skims the ground, so that it is neither over nor under it.

Right: that's it. Compile you map and try it out, what should happen is you hit the trigger, the plane flies above, drops it's payload, and departs. :)

Of course, you can make multiple bombs come out of the plane by making separate paths for each one. Making it easy enough to bomb out an entire building with one plane :).

The **#set** value is used to determine between spawns, if you use the spawnset method of spawning ai do not make a set the same as an ai's spawnset.

You can easily change the plane that appears to anything, just change the $mdl and the model value. Make sure it ends with **fly.tik**. the bomb is one-of-a-kind i belive, i've never found another tiki for it so keep that as **us\_bomb**.

- [ReptilianMapper](http://gronnevik.se/rjukan/index.php?n=Profiles.ReptilianMapper)

## Note

This can also be done solely with script. The same way as described above but you spawn all needed entities in script instead of using radiant.  
Apply the keys appropriately. Though creating a spline path in script is more difficult as you have no visuals to base your angles and origin on.

# Strafe2

Strafe2.scr is a script based on the bomber.scr created by BdBodger. Using a slightly different, yet more evident approach.  
Besides there being more options available, in strafe2.scr there is no need to set an extra parallel path of nodes for each bomb.   
  
One path means one plane and its bombs. Add the $bomb key to a node to make the plane drop a bomb when it flies over the node.  
Additionally you can make extra scripts run by using the $setthread key on a pathnode.   
  
More information found here:

//=================================================================================================================

//

// Plane/strafeing system created by bdbodger ( bdbodger@hotmail.com ) ©2004

//

// You are free to use this script for noncommercial use as long as you do not remove this message

//

// and if you edit the script you indicate in the script the changes you have made .

//

// This script is offered as is and author is not responsible for improper use .

//

//=================================================================================================================

//

// example:

//

// main:

//

//

// level waittill spawn

//

// exec global/strafe2.scr

//

//

// thread global/strafe2.scr::strafe # [optional plane sound]

//

// end

//

//=================================================================================================================

//

// After level waittill spawn

// "exec global/strafe2.scr".

//

// Created a info\_splinepath and set it's model value to the plane you want to use. Choose one that ends with fly.tik.

//

// Note: If you set the model of the splinepath nodes to the model you will be useing ( $mdl value )

//

// it will be easier to see the angles of the plane at that point in the path in the editor .

//

// Copy the info\_splinepath into a big line of them , where you want the plane to fly through space.

//

// Link them all together( node1 targets node2 targets node3 etc ). Give the first one a targetname of "strafe\_path"

//

// Set this one's $mdl value to be equal to its model value . set #damage to the amount of damage the plane will

//

// take before being destroyed , health is saved between spawns . 20 second delay before respawnable at full health .

//

// this defaults to 1000 . Rotate each individual info\_splinepath node angles .

//

// The way its rotated will determine what position the plane will be in when it flies through that

//

// point in space and the angles of the first node will be used to setup the guns and fireing targets

//

// ( do not set "angle" set "angles" ).

//

// You can set the "speed" key to multiplicably values. This defaults to 1 . If you think 1 is too fast,

//

// set all the splines to a speed of 0.5. Or set half of them to 0.5, and for that half the plane

//

// will fly at half speed. If you set level.flystrafe = 1 all the planes will fly at a constant speed of

//

// <speed> 1500 <acceleration> 200 <lookahead> 256 However because of the way flypath works with regard to how

//

// lookahead smooths out the path of the plane , the plane may fly too far from the spline\_path node

//

// to drop a bomb if #bomb is set at that node . Give all the nodes of the "strafe\_path" the same #set value .

//

// You can make triggers for the strafe with "trigger\_multiple"s that have a targetname "strafetrigger"

//

// and the same #set number as the strafe\_path ,or you can do: thread global/strafe.scr::strafe #

//

// where # is the #set number in question . You can give each of the nodes on the path a Key:$setthread

//

// and the name of a thread in your level.script to run when the plane flys over that node

//

// You can give any of the nodes a Key:#bomb and a value that is how often the plane should drop a bomb

//

// example: key:#bomb Value: 3 // plane will drop a bomb every third time it is spawned

//

// you can give any node with a #bomb value a Key:#bombstart and a value that is when the first bomb should drop

//

// example: key:#bomb Value:3 Key:#bombstart Value: 2 // plane will drop a bomb the second time it is spawned then

//

// every third time after that . You can give any of the nodes on the path a Key:$bombthread

//

// and the name of a thread in your level.script to run when a droped bomb explodes on the ground from

//

// that node . Set level.script = to the name of your map for strafe sounds , $bombthread and $setthread to work

//

// properly . example: level.script = maps/mymap.scr .

//

//=================================================================================================================

// First info\_splinepath node keys

//=================================================================================================================

//

// Key:$targetname Value: strafe\_path ( not optional )

// Key:#set Value: < integer > ( set each each node to same set# each path to a different set # )

// Key:$mdl Value: < model > ( Plane model default vehicles/p47fly.tik)

// Key:$scrplane Value: < script\_object> ( Script object brush model for plane instead of $mdl )

// Key:angles Value: < x y z > ( pitch yaw roll )

// Key:#planescale Value: < float > ( plane scale default 1 )

// Key:#gunangle Value: < angle > ( gun pitch angle default 30 )

// Key:#guns Value: < boolean > ( guns on or off (1,0) default 1 on )

// Key:#sideoffset Value: < float > ( gun offset from center \* #planescale( damage trigger 100x100 \* scale on center ) default 70 )

// Key:#forwardoffset Value: < float > ( gun forward offset from origin default 100 )

// Key:#heightoffset Value: < float > ( gun height offset from origin default 0 )

// Key:#damage Value: < integer > ( health when new plane spawned or after plane death default 500)

// Key:#destroyable Value: < boolean > ( 1 destoyable 0 not destroyable default 1 overrides #damage )

// Key:$planeexpmdl Value: < emitter > ( plane explosion emitter default animate/fx\_mortar\_higgins.tik)

// Key:#planeexpscale Value: < float > ( plane explosion scale default 1 )

// Key:#pause Value: < time > ( optional time to wait after spawn before flying in sec )

// Key:#diebounce Value: < float > ( if set plane will bounce when shot down before explodeing default 0 )

//

//=================================================================================================================

// All info\_splinepath nodes keys ( read from node )

//

//=================================================================================================================

// some optional keys need only to be set once if at all unless you want to change them at another node

// for example planescale only needs to be set if you want to change the scale of the plane at this node

//=================================================================================================================

//

//===================================== Optional Keys =========================================================

//

// Key:$setthread Value: < thread > ( optional level.script thread to run when plane flys over node passes #set and #node to thread )

// Key:model Value: < model > ( optional editor model for spline\_path node ( editor only ))

// Key:#planescale Value: < float > ( optional new plane scale )

// Key:#bombstart Value: < spawn # > ( optional spawn # delay for bombing if #bomb or #singlebomb set )

// Key:#bomb Value: < # spawns > ( optional value for bombing every # spawns (after #bombstart if #bombstart set))

// Key:#singlebomb Value: < boolean > ( optional value for single bomb drop , resets #bomb to NIL after bombing can be used with #bombstart )

// Key:#randombomb Value: < integer > ( optional Value for random bombing chance of bomb 1 to #randombomb )

// Key:$bombtarget Value: < targetname > ( optional entity targetname for bombing (reset to default bombing angles or $statictarget after bombing))

// Key:$statictarget Value: < targetname > ( optional entity targetname for bombing (is not reset to default and is overridden if $bombtarget set))

// Key:$bombmodel Value: < model > ( optional model for bomb default ammo/us\_bomb.tik )

// Key:#bombscale Value: < float > ( optional bomb scale set each node default 1 )

// Key:bombsound Value: < soundalias > ( optional sound for dropping bomb default drop\_bomb )

// Key:bombdamage Value: < float > ( optional bomb damage default 500 )

// Key:bombradius Value: < float > ( optional bomb damage radius default 500 )

// Key:$expmodel Value: < emitter > ( optional explosion model for bombing default models/emitters/explosion\_mine)

// Key:#expscale Value: < float > ( optional explosion model scale set each node default 1 )

// Key:$bombthread Value: < thread > ( optional level.script thread to run after bomb explosion passes #set and #node to thread)

// Key:#guns Value: < boolean > ( optional value to start/stop guns fireing ( 1 on 0 off ))

// Key:#gunangle Value: < newangle > ( optional new gun pitch angle )

// Key:speed Value: < float > ( optional new speed setting at node default 1 or speed set in editor )

//

//

//=================================================================================================================

// Level Nodes ( Keys can be changed by script see Keys above )

//=================================================================================================================

//

// level.strafenode[set #][node #].origin ( node origin )

// level.strafenode[set #][node #].angles ( node angles )

// level.strafenode[set #][node #].<key> ( read from node each flight )

//

// level.startbomb[set #][node # ] ( Stored after node setup can be changed to delay )

// ( bombing or set #bomb Key to NIL to stop bombing )

//

// level.index[set #][node #] ( counter for #bombstart and #bomb key , 1 to )

// ( startbomb then 1 to #bomb )

//

//=================================================================================================================