Basic Camera :: Rigges   
-------------------------------------------------------------------------------

*For this tutorial, I will assume you have a working knowledge of MOHRadiant. This tutorial only deals with setting up basic camera operation, which allows a camera view from any angle and setting up a path for the camera to follow.*

The first step is to set up a basic room with a light. (See the First Room tutorial by [Surgeon](mailto:Surgeon@planetmedalofhonor.com)). At this point, put an object in the centre of the room for the camera to move around. For the sake of this tutorial I have placed an Allied AI Captain in the room. To do this, simply right click inside the edit area of the editor, select the menus AI/Allied/First Ranger/ the select captain. Position the captain in the centre of the room.

-------------------------------------------------------------------------------  
Part 1 : Fixed Camera  
-------------------------------------------------------------------------------

To set up a fixed camera, position a splinepath node anywhere on your map where you want to camera to exist. You can find the splinepath node by right clicking the edit area of the editor, selecting the "info" menu. I use splinepath in this tutorial as we can expand on this later to create a moving camera. In fact, the camera (view) can be attached to any object that has an origin; this includes vehicles and AI players.Once your splinepath is positioned, select it and press 'N' to get the entity options. Enter a key / value of $targetname / camerapath .

As you can see in the picture below, I have a captain setup, with a purple splinepath selected above his head. This will be the position he is viewed from

|  |
| --- |
| http://www.mohaaaa.co.uk/mohaa/tutorials/basic_camera/image1.gif |

Now we get to scripting. Create a new script called test\_camera.scr and enter in the following lines:

|  |
| --- |
| main:  level waittill prespawn level.farplane = 25000  thread NullPlayer  end  NullPlayer:  // this little section, sets up the player to be a camera! $player nodamage $player notsolid $player physics\_off $player hide $player threatbias ignoreme $player model "fx/dummy.tik" $player glue $camerapath // basically glue the view to the object you want to view from local.xcoord = 70 local.ycoord = 90 local.zcoord = 00 $player.viewangles = (local.xcoord local.ycoord local.zcoord) // (X Y Z) X is up down pitch, Y is left right pitch, Z is rotation  end |

And that's it. You have a basic camera set up that will view your scene from whatever position your splinepath is set.

-------------------------------------------------------------------------------  
Part 2 : Moving Camera  
-------------------------------------------------------------------------------

This is a little trickier, largely due to scripting, but if you just use my script, it should be easy enough. First of all, create a few more splinepaths as you did before, and name each one "path1", "path2", "path3" and so on, respectively, and place them in a line that you want your camera to follow. Basically, each point in the path will represent a large chance in camera direction. Now, we need to join these splinepaths together.

This isn't complicated at all, don't worry!. Simply, click on the first splinepath ("camerapath"), and enter a key / value of target / path1 and press enter. This links the first splinepath to the second one. Now select the second splinepath ("path1") and enter a key / value of target / path2 , and so on with all other splinepaths, until they are all linked. You will see it has worked as an arrow will join them together, visible in the editor. For each splinepath, assign a set on each one by simply entering for each splinepath a key/value of $set / 1.

The image below, is the same scene as before, but you can see five splinepaths instead of just one, and an arrow links them all together.

|  |
| --- |
| http://www.mohaaaa.co.uk/mohaa/tutorials/basic_camera/image2.gif |

That's the editor finished with, we now have a path for our camera to follow. Now the script. I've made the script as simple as possible so as not to get too confusing, and it does the trick, but I'm sure there are easier ways to do it. I have comments to explain the basics of each point. Anyway, here it is:

|  |
| --- |
| main:  level waittill prespawn  level.farplane = 25000 thread NullPlayer thread CameraSetup  end  NullPlayer: // this little section, sets up the player to be a camera!  $player nodamage $player notsolid $player physics\_off $player hide $player threatbias ignoreme $player model "fx/dummy.tik"  end  CameraSetup:  println "-- Configure Camera" local.xcoord = 70 local.ycoord = 90 local.zcoord = 00 $player.viewangles = (local.xcoord local.ycoord local.zcoord)  // (X Y Z) X is up down pitch, Y is left right pitch, Z is rotation local.set = level.camera[local.i].set level.camerapaths = $camerapath.size level.camerapath = exec global/makearray.scr $camerapath local.set = level.camerapath[local.i].set level waittill spawn thread CameraSetLoop self.set  end  flypath local.i:  self.origin = level.camerapath[local.i].origin  self followpath level.camerapath[local.i] self waitmove  end  CameraSetLoop local.name:  for (local.i=1;local.i<level.camerapaths+1;local.i++) { spawn script\_model "targetname" ("CameraObjects" + level.camerapath[local.i].set) "spawnflags" "2" // allow camera to move through path provided $player glue ("CameraObjects" + level.camerapath[local.i].set) // attach view to camera path $("CameraObjects" + level.camerapath[local.i].set) thread flypath local.i // flypath, using current view point }  end |

The first section of code is basically the same as before, the main difference is a loop, which is used to step through each point of the camera view as it travels along the path. Not to bad? Just read the code and it should make enough sense for you to use it! Hey, it works for me.

Just a few final words on this. I intend this tutorial only as a work in progress. I hope others can use my work here to progress and have better camera work in their games! I'm still working on removing the hud display, and changing the camera angle while you move along the path. Oh, and one last thing, the script works!, any time I had problems, it was something to do with the splinepaths, they are temperamental little fellas, so if you're having a problem, just pick apart the sample level I've supplied

[Download the Example map]

Need any help : Ask in the [Forum](http://www.mohaaaa.co.uk/mohaa/forum/index.php)