|  |  |
| --- | --- |
| |  | | --- | | **Barking Dogs :: Wacko** | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| The barking we gonna do will be adjustable in many different parameters: We can choose, which barking sounds we want to use, how often the dogs will bark once they've started, for how long they bark, how long they rest until they start barking again and some more. For all these settings we will not give plain values but rather choose min and max values, and then randomize the outcome. This will produce a barking which will never be the same and therefore sounds quite realistic.  ------------------------------------------------------------------------------- Mapping -------------------------------------------------------------------------------  Start with a new map or open the one that should get the dogs. Let's say it's called doggie.map. In the map, we gonna need a spot, where the barking comes from - a speaker. This is easily be done by rightclicking in a 2D-view, just where the barking should be loudest, and choosing a script\_model in the entity list. With this new entity (still) selected, press 'n'. The 'Entity' window opens and here you add some key/value pairs:   |  |  | | --- | --- | | key: | model | | value: | fx/dummy.tik | |  |  | | key: | $targetname | | value: | bark\_o\_mat |   Finally check the always\_draw box. This will make the speaker to be heard even if it's not visible.  You will want to add a info\_player\_deathmatch, so you can test the map in deatchmatch mode. Finally, compile the map and move the doggie.bsp from whereever it is to mohaa/main/maps/dm.  So much for mapping!  ------------------------------------------------------------------------------- Scripting -------------------------------------------------------------------------------  If your map already has a script, you will have to copy the barking pieces into it and you'll probably know what to put where. Else, you'll have to start a new script for your map. For a map called doggie.map, the script will have to be named doggie.scr. Open a new textfile in notepad and copy/paste the following code into it:  main:   local.master = spawn ScriptMaster   local.master aliascache b\_1 sound/characters/dog\_bark\_1.wav soundparms 1.5 0.0 0.8 0.4 160 4000 voice loaded maps "dm "   local.master aliascache b\_2 sound/characters/dog\_bark\_2.wav soundparms 1.5 0.0 0.8 0.4 160 4000 voice loaded maps "dm "   local.master aliascache b\_3 sound/characters/dog\_bark\_3.wav soundparms 1.5 0.0 0.8 0.4 160 4000 voice loaded maps "dm "   local.master aliascache b\_4 sound/characters/dog\_bark\_4.wav soundparms 1.5 0.0 0.8 0.4 160 4000 voice loaded maps "dm "   local.master aliascache b\_5 sound/characters/dog\_bark\_5.wav soundparms 1.5 0.0 0.8 0.4 160 4000 voice loaded maps "dm "    level waittill spawn    thread dog\_bark end  dog\_bark:   //by wacko (woof, woof)   while(1)   {     level.bark\_times = randomint 10 + 1     wait (randomint 21 + 30)     for (level.times\_barked = 1; level.times\_barked <= level.bark\_times; level.times\_barked++)     {       level.bark\_num = randomint 5 + 1       level.bark\_style = "b\_" + level.bark\_num       $bark\_o\_mat playsound level.bark\_style       level.bark\_wait = randomfloat 1 + .5       wait level.bark\_wait     }   } end  Save this as doggie.scr into mohaa/main/maps/dm, just where doggie.bsp is. Now you can start your map and you'll hear the dog! You will have to wait a bit, because the dog won't start immediately...  ------------------------------------------------------------------------------- Adjusting The Script -------------------------------------------------------------------------------  The scripting above is how I wanted my dog to bark, but you might want to change the parameters, so here's the code (in blue) again with explanations (in green) showing where you have to change something to achieve a different barking:  main:   local.master = spawn ScriptMaster   local.master aliascache b\_1 sound/characters/dog\_bark\_1.wav soundparms 1.5 0.0 0.8 0.4 160 4000 voice loaded maps "dm "   local.master aliascache b\_2 sound/characters/dog\_bark\_2.wav soundparms 1.5 0.0 0.8 0.4 160 4000 voice loaded maps "dm "   local.master aliascache b\_3 sound/characters/dog\_bark\_3.wav soundparms 1.5 0.0 0.8 0.4 160 4000 voice loaded maps "dm "   local.master aliascache b\_4 sound/characters/dog\_bark\_4.wav soundparms 1.5 0.0 0.8 0.4 160 4000 voice loaded maps "dm "   local.master aliascache b\_5 sound/characters/dog\_bark\_5.wav soundparms 1.5 0.0 0.8 0.4 160 4000 voice loaded maps "dm "  These five lines copy the barking wave-files into the cache and give them several parameters telling the engine how to work with them:   |  |  | | --- | --- | | local.master aliascache |  | | b\_# |  | | sound/characters/dog\_bark\_1.wav | The wave file to be used. | | soundparms |  | | 1.5 | Base volume, higher values are louder | | 0.0 |  | | 0.8 |  | | 0.4 |  | | 160 | MinDist; being nearer than this, the player will hear the full volume. | | 4000 | MaxDist. From MinDist to MaxDist the volume fades away, being further away than MaxDist from the speaker, the player won't hear it | | voice |  | | loaded |  | | maps "dm " | In quotes the beginning letters of the mapnames (including folder) where this sound shall be cached. "dm " is for all maps in mohaa/main/maps/dm.  For further info about aliascache, please read the ubersound.scr in pak0.pk3/ubersound. |     level waittill spawn    thread dog\_bark end  dog\_bark:   // by wacko (woof, woof)   level.bark\_times = randomint 10 + 1  here the number of barks per 'attack' is created: level.bark\_times will be a value from 1 to 10. 'randomint 10' returns a value of 0 to 9. The dog will bark 'level.bark\_times' once it decides to start barking.     wait (randomint 21 + 30)  before starting to bark, the dog will wait for some seconds: minimum waittime is 30, maximum waittime is 50 seconds (20+30).    for (level.times\_barked = 1; level.times\_barked <= level.bark\_times; level.times\_barked++)   {     level.bark\_num = (randomint 5 + 1)  each time, the dog barks, it will 'choose' one of the 5 cached barking sounds. If you don't like one of them, delete it's line above, adjust the 'h\_#' numbers in the remaining lines and change the randomint command here to the number of bark sounds.      level.bark\_style = "b\_" + level.bark\_num     $bark\_o\_mat playsound level.bark\_style     level.bark\_wait = (randomfloat 1 + .5)  each bark needs some time to be played. Randomfloat(1) produces a value from 0.001 to 0.999. So the script waits for 0.001 + 0.5 = 0.511 seconds to 0.999 + 0.5 = 1.499 seconds. This way, some barks won't be played fully, and sometimes the dog will take breath before its next bark. Imho, this sounds quite realistic, but off course you can play with these values.      wait level.bark\_wait  here's where the script is waiting after having started a single bark.    }   goto dog\_bark end |