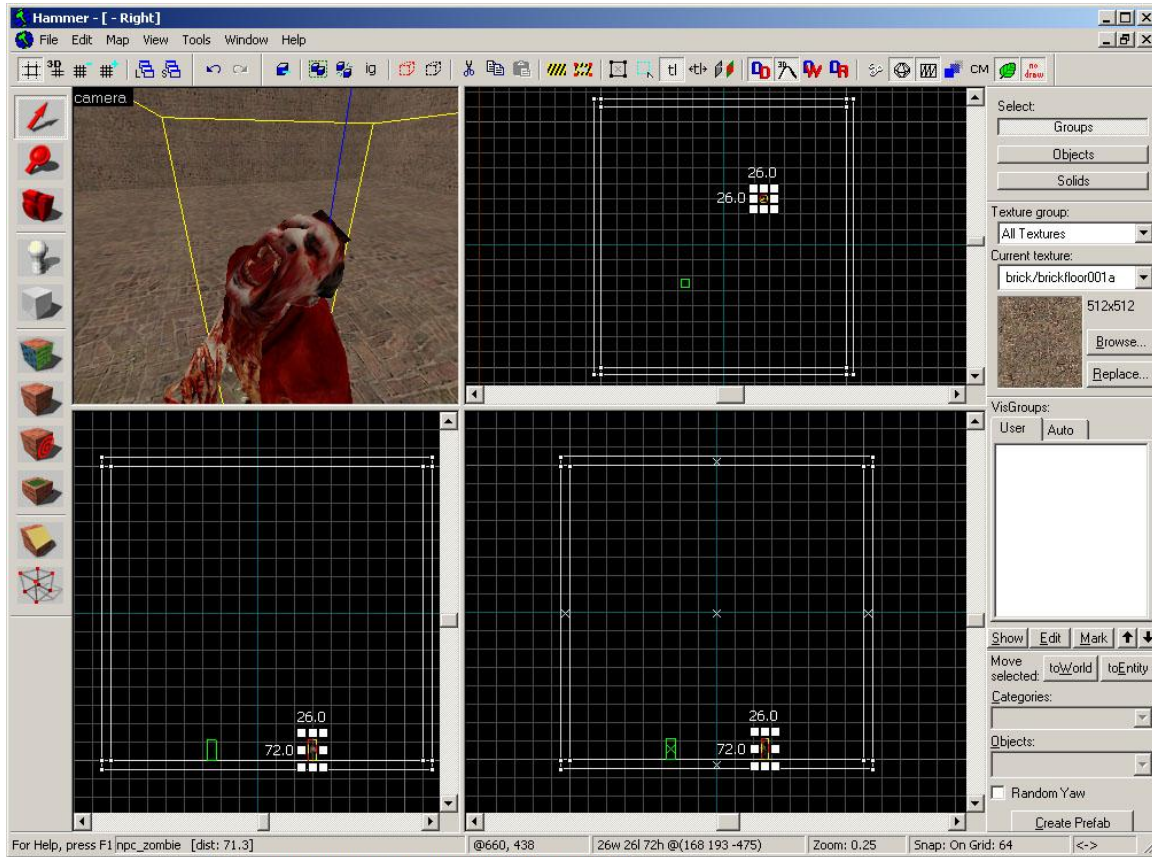


# A tutorial on scripted sequences & cutscenes creation

By Christian Clavet

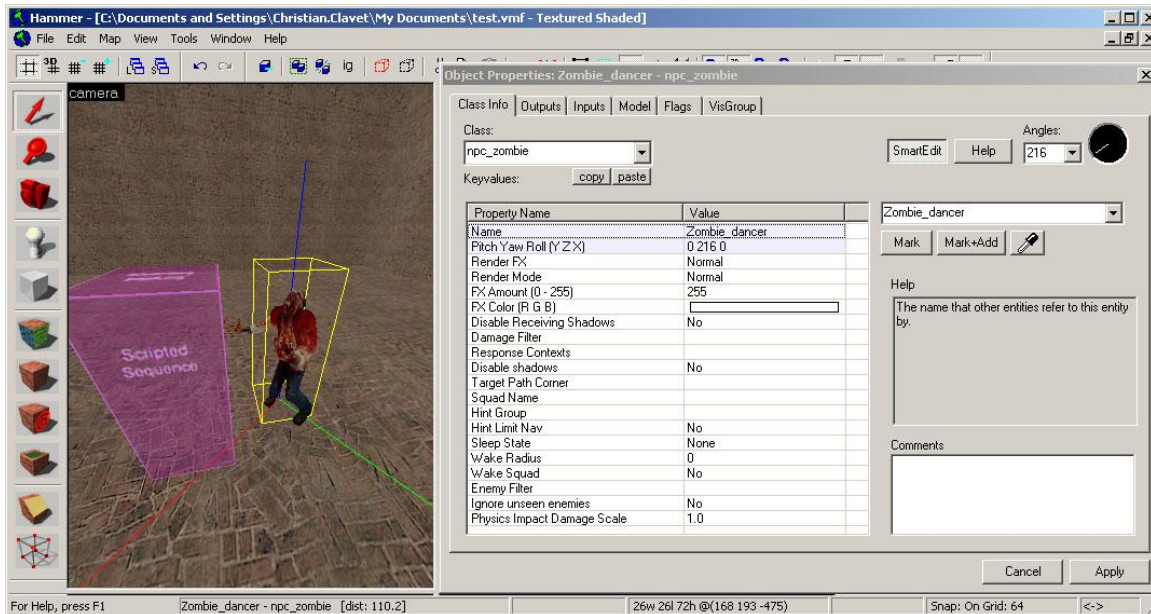
Setting up the scene



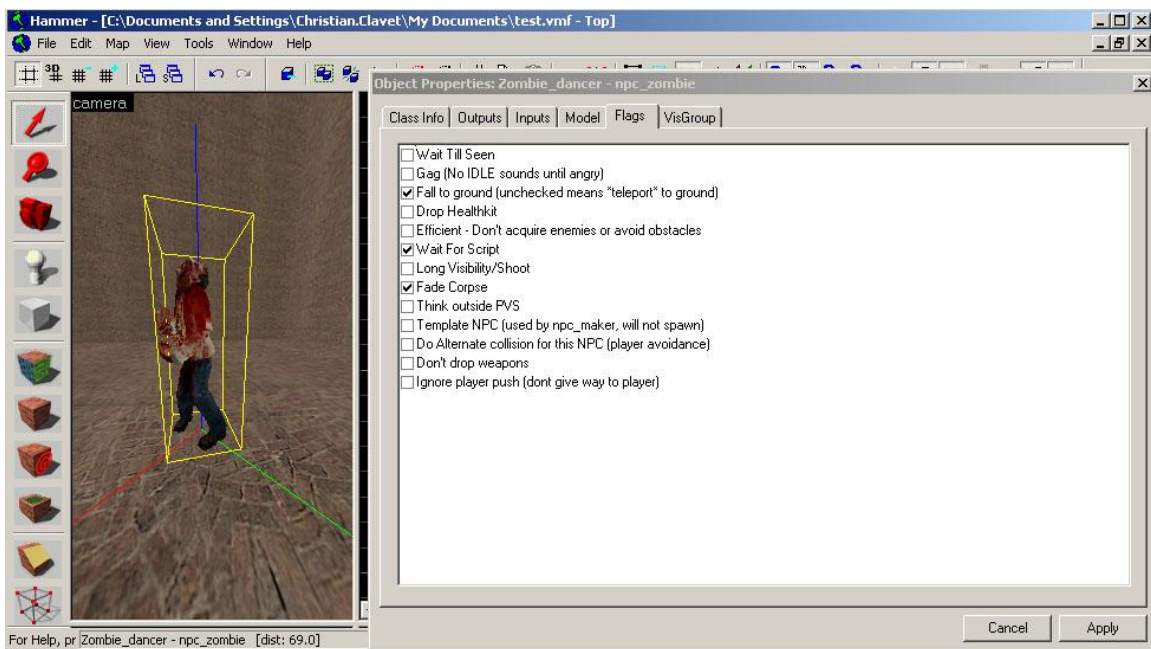
This is a quick tutorial to explain how to use the entity named : « scripted-sequence » to be able to move a character on screen. You could use this if you'd need to have a specific action done at a specific time. Mostly useful for creating cutscenes.

For this tutorial, I've created a basic « box » room, added the player and a Zombie that we'll manipulate to do a specific action. For creating the Zombie, I've added the entity named « NPC\_ZOMBIE » to the scene.

## Choosing the character to animate

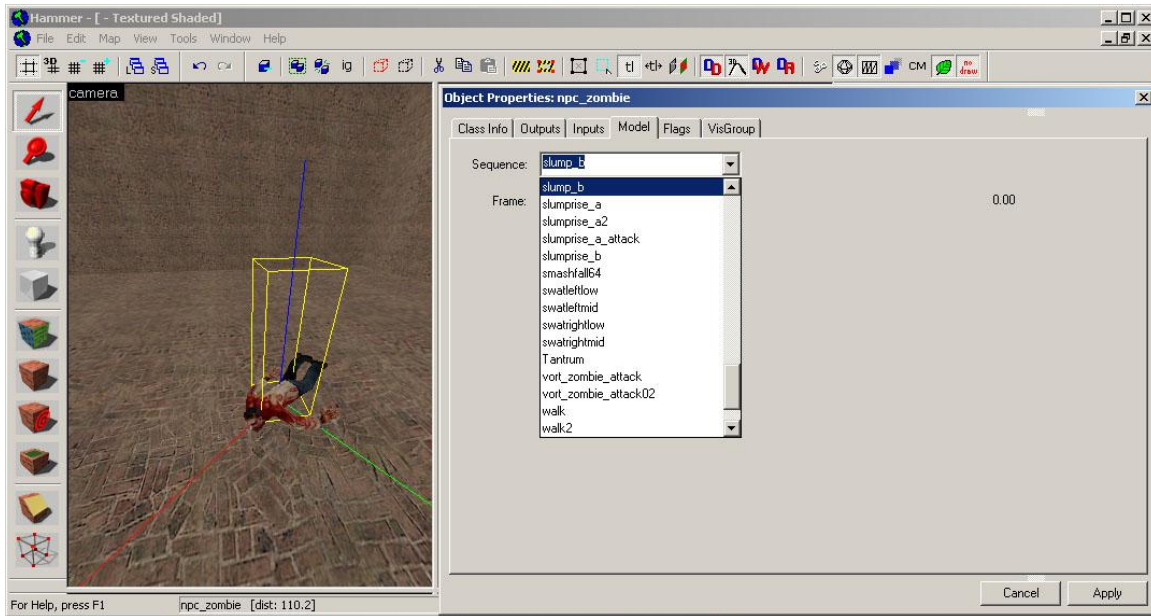


Once the Zombie is placed at the wished position, select him and open the properties. The first thing to do is to **name** the character entity. I've named him « *Zombie\_dancer* »



You'll also need to setup the flag to allow the character to use scripts. Click and activate the "Wait for Script" flag.

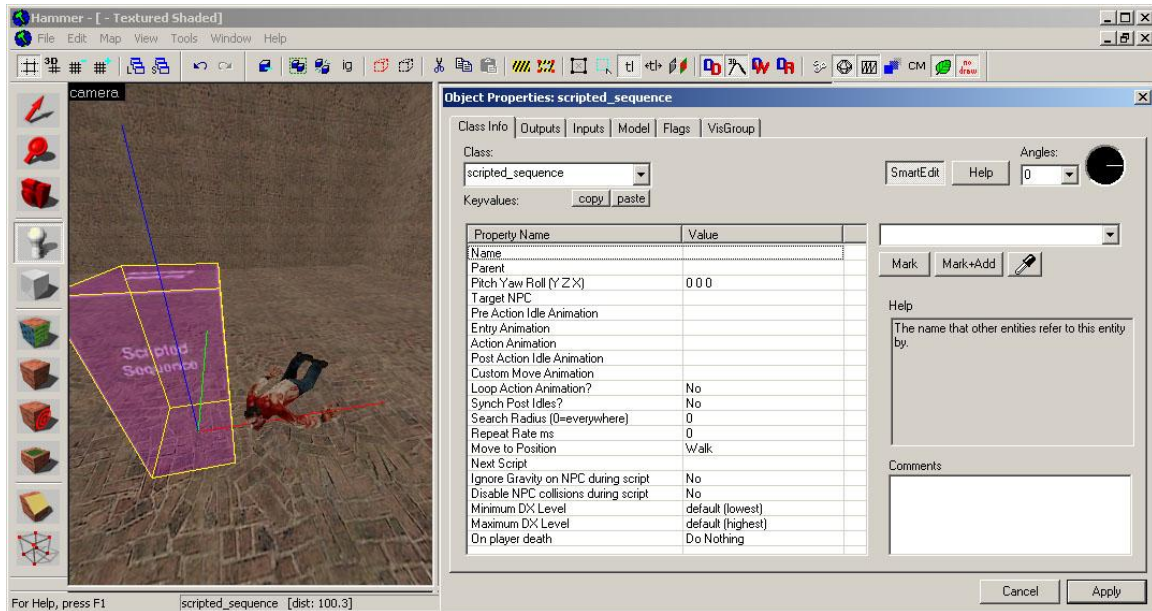
## Listing the animations



If you'd go in the « **model** » panel, you will see a list of all the available animations that this character can do. Remember this, we'll use this later.

For the scripted sequence you'll need to type in exactly the name of the animation listed here. You could use the cut & paste method to make it easier, because the text field is selectable.

## Adding the scripted sequence

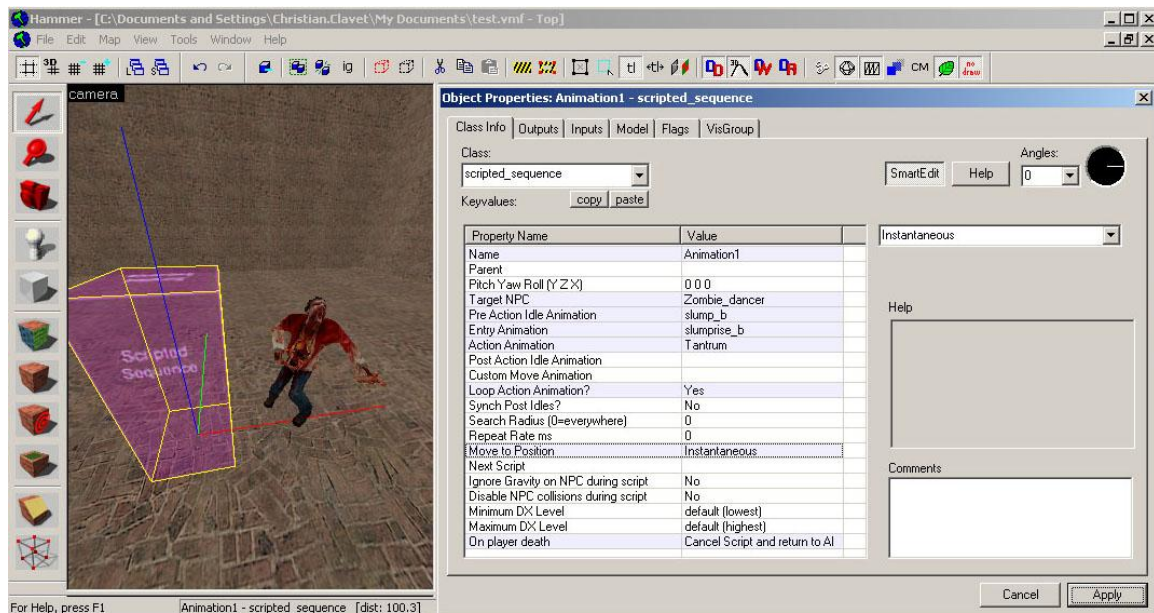


Now, we'll add the entity named « *Scripted\_Sequence* ». Just position near the zombie. For a specific action that you would like, you could place that were you'd want the character would run, or walk. For our need here, we will not use this. We'll need to add info in the Class info and in the Flags.

This scripted sequence has some limits. It cannot animate a bunch of character of same name, so you'd have to create one per character, and is limited also to one animation type per character like, « *pre-action* », « *entry-action* », « *action* » and « *post-action* ». So if you need more, you will have to create others scripted sequences for the same character.

The « *move to position* » feature is interesting because on certain case, you can have your character walk, run or even do a specific move to reach the scripted sequence box.

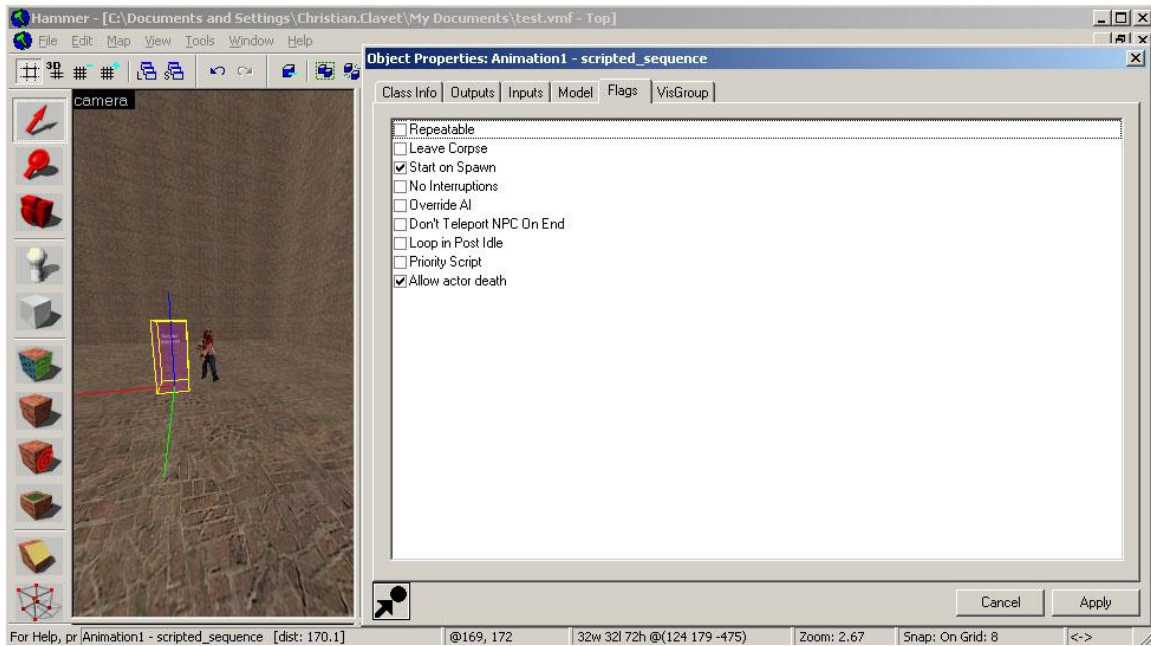
## Setting the animations of the character in the scripted sequence



Here we'll define the parameter of the scripted sequence :

- Give this scripted sequence a **name**. I've named it « *Animation1* »
- « **target NPC** », in my case « *Zombie\_dancer* »
- For the **pre-action** (before we start the sequence), I've selected « *slumb\_b* » from the list of the animation the zombie supported. Check back the list of animation supported for your character on the model tab when you select your character, the animation name must be entered manually.
- For the **Entry animation**. (First animation to play when starting the sequence) I've selected « *slumprise\_b* ».
- The **action animation** is the main animation the character will play. For this zombie, I would like to have him simulate a dance move, and chosen « *Tantrum* ». I've chosen to loop the action animation so that the character will repeat the « *tantrum* » move without end. This should give the impression that his dancing.
- **Move to position** is set to « *no* » because I don't want the character to start at the box entity position, but at the current position in his « *slumb\_b* » pre-action animation without moving there.
- On **player death** is set to Cancel Script and return to AI if I attack the character.

## Setting the flags in the scripted sequence



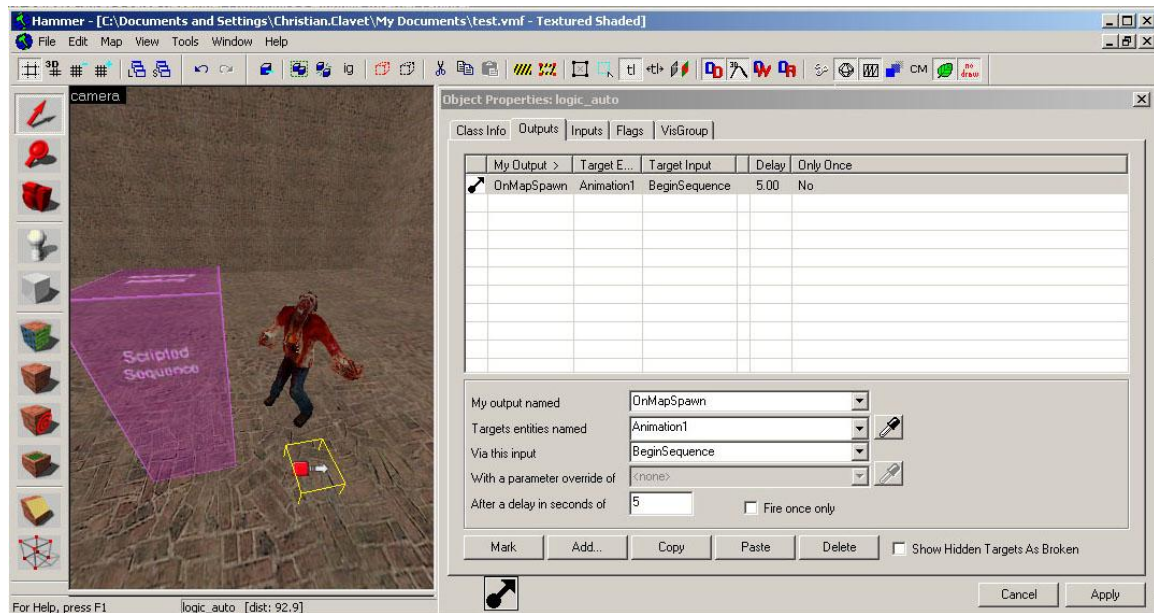
Here you would have to set these :

- **Start on spawn** : to start the pre-action
- **Override AI**, if you don't want the zombie attack and it's only for cutscene
- **Priority script** that help to override the scripts there
- **Allow actor death**, so that we can attack him

Once that is done, we can test it ingame. We should see the zombie in his SLUMB mode.



## Triggering the scripted sequence animation on game launch

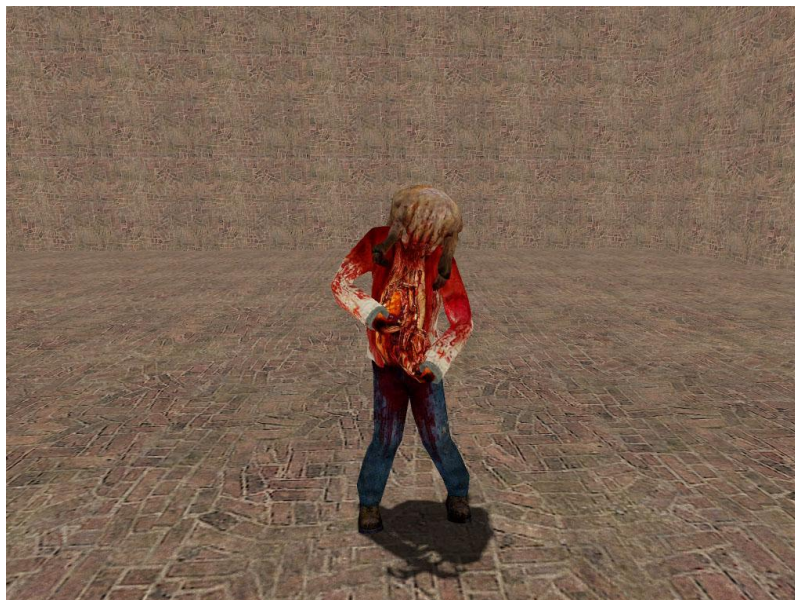


Since I'd want to use that scripted sequence like a cutscene, I could trigger it many different ways. Since, I want to start it at map spawn time, I'll use the entity named "logic\_auto".

You simply have to put this entity somewhere in your map and after set the output to trigger what you'd want to do at map spawn time.

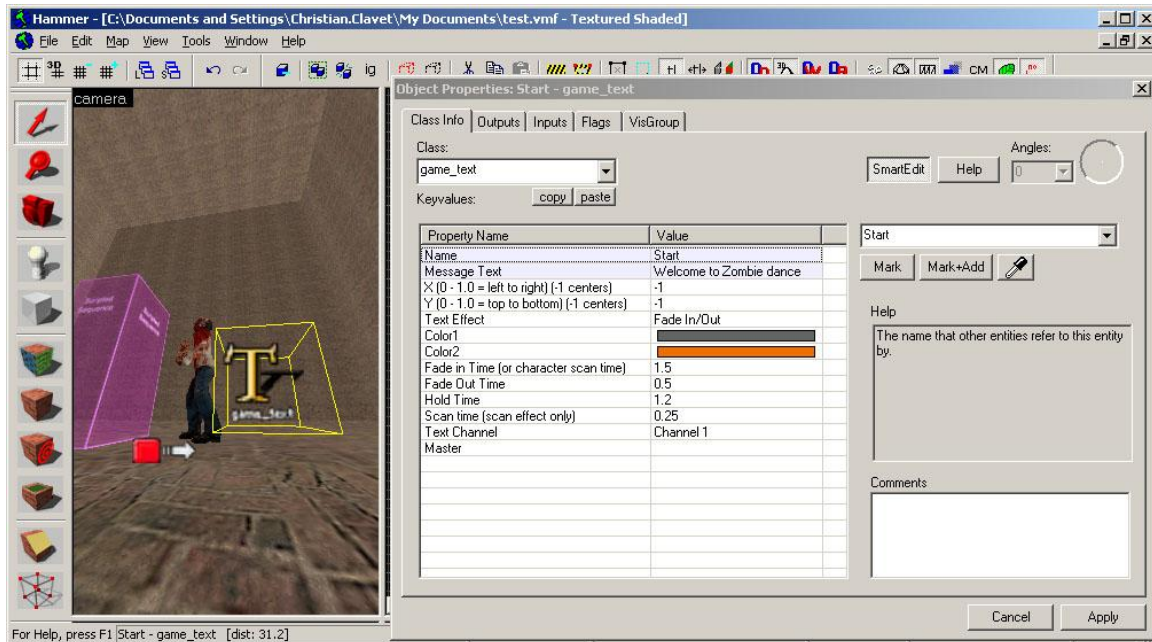
For the tutorial, I've selected the Target as "Animation1" that is my scripted sequence then, set a delay of 5 seconds to start the movement.

When you'll launch the game, the zombie, will be on the ground, rise and then "dance".



## Going further: creating a cutscene with the current scene

Creating ingame texts:



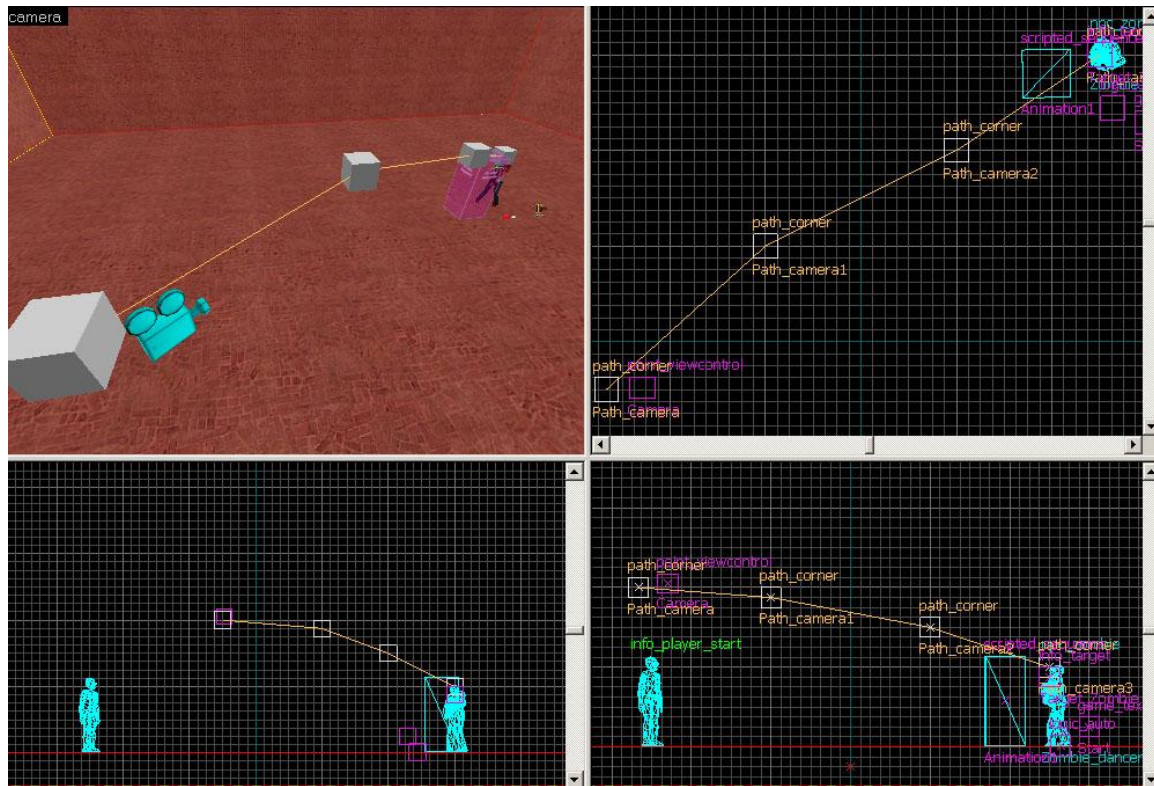
There is a nice Entity named "game\_text" that will allow you to create in-game text in your level. You need to define at least those:

- Name: give it a proper name so we can trigger it later
- Your message text, simply type a sentence there
- X and Y position, you can have centered, bottom or top
- Text effect, there is 2 effects, fade and type effect
- Hold time to specify how long you want the text to remain on screen.

The text entity can use up to four(4) channel so you could display up to 4 things at the same time on the screen with those entities.



Creating a camera and a movement for the scene:



For creating a moving camera style cutscene you will need to setup this:

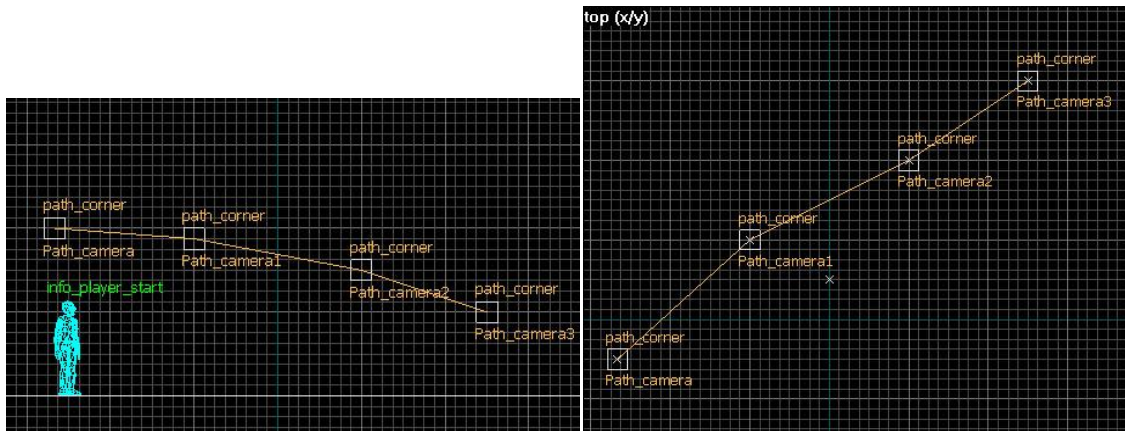
- Create a “point\_viewcontrol” entity that would create the view (camera)
- Create “path\_corner” entities that trace the path of the moving camera
- Create a “info\_target” entity to define where the camera is going to look

Once that is done, you’ll have to set some info as:

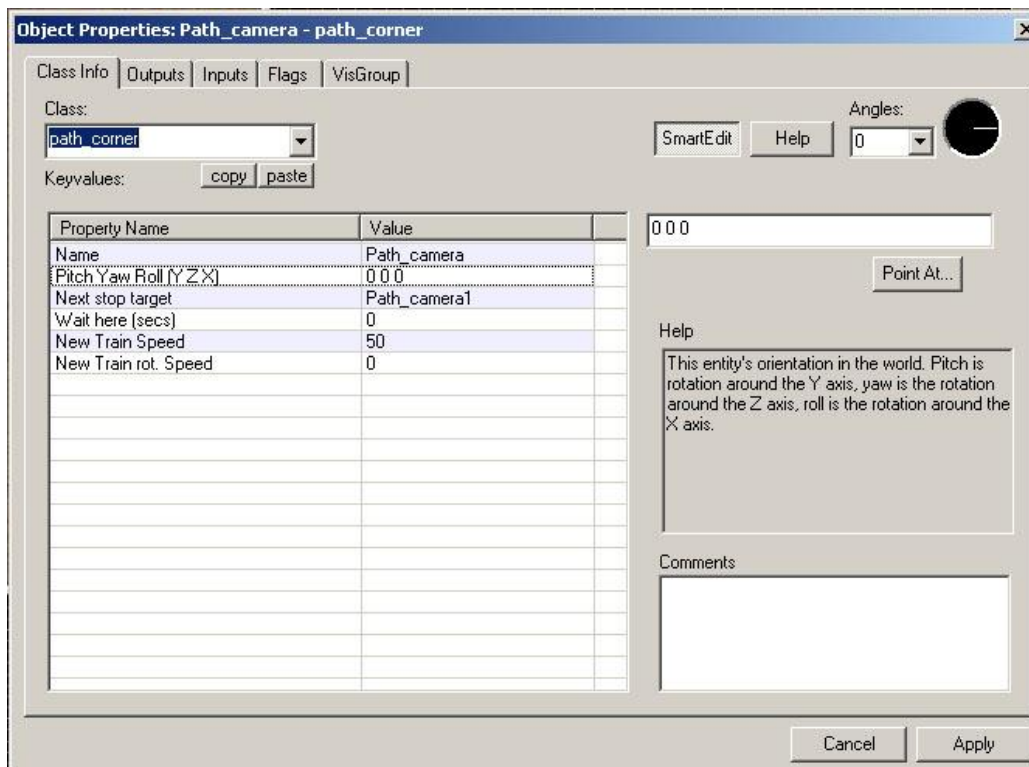
- First Path\_corner entity. Set the initial speed of the moving camera (this can be changed at each path corner entity)
- Last Path\_corner entity. Tell the camera to stop and return the control to the player.
- Use a trigger to activate the camera and removecontrols from the player for the duration of the animation.

Creating the Path corner and define the trajectory of motion.

Here is how first path corner entity fully defined:

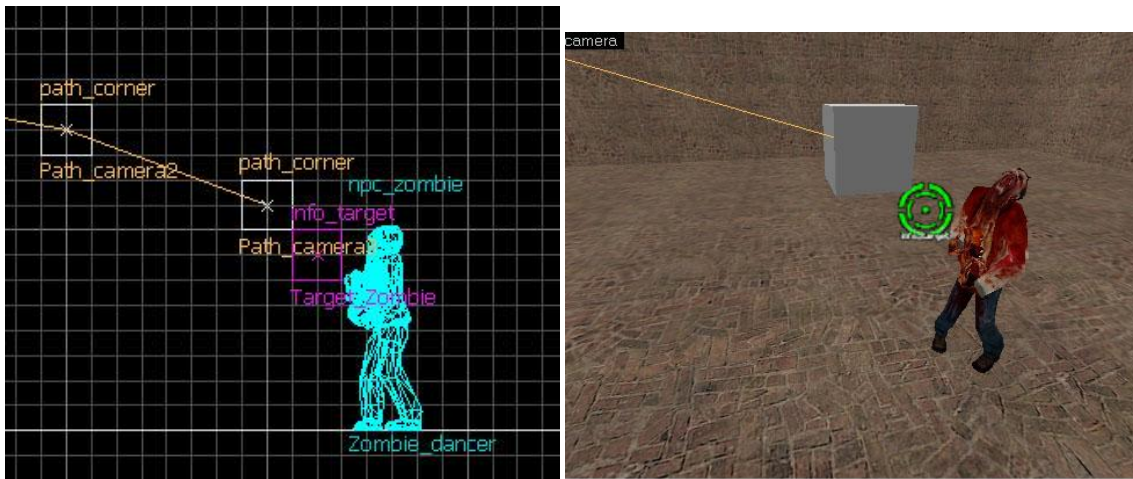


- I've placed the first entity and gave it a name.
- Then moved it using the SHIFT key and created a duplicate, Hammer then, updated the name and created a link to the next automatically. For the movement I'd was willing to make I've used four(4) path corner. The base name given was "Path\_camera"



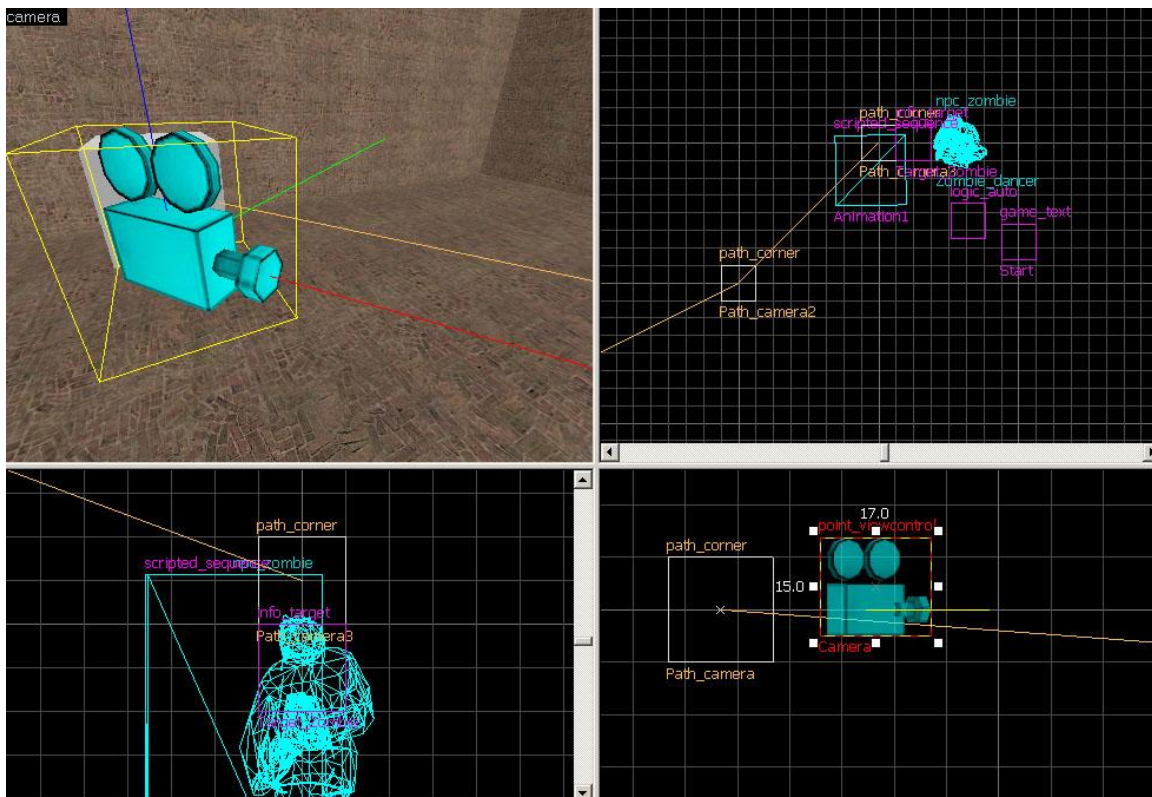
Here, I selected my first path corner and set the "NEW TRAIN SPEED" to 50. This will define the starting speed of the camera when it will use this path.

## Creating a camera target for the sequence



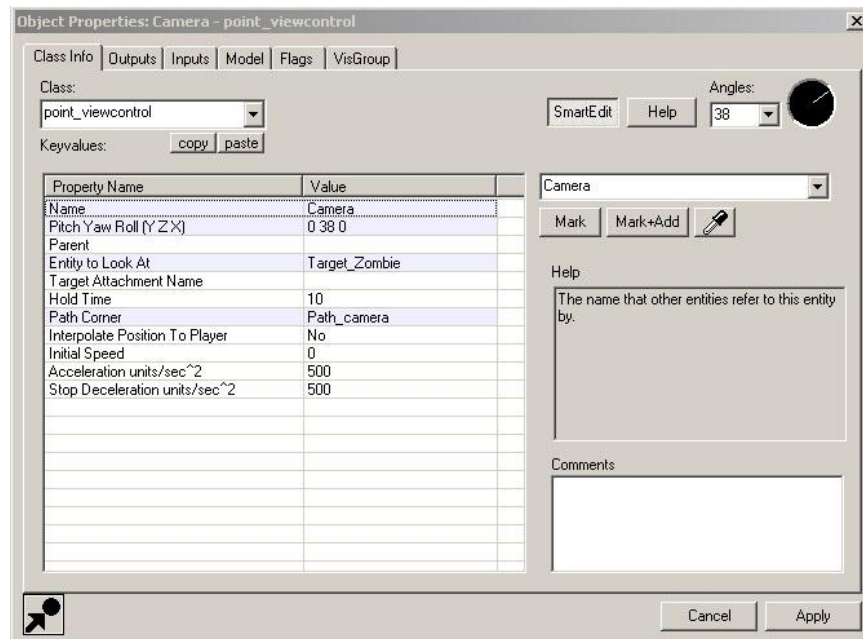
You simply add the entity “info\_target” and give it a name. For the tutorial, I’ve named this one “Target\_Zombie”.

## Creating the camera that will use the target AND the path



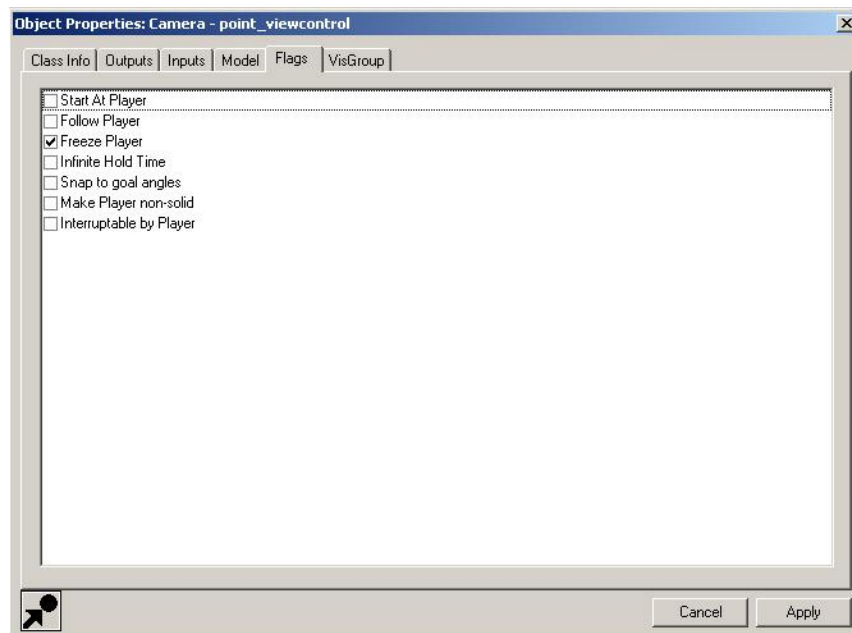
I’ve created a **point\_viewcontrol** entity, placed it near my first path corner and oriented it the direction of the path. All these steps are not required by are more easy to locate things after in the editor.

## Setting the viewcontrol entity:



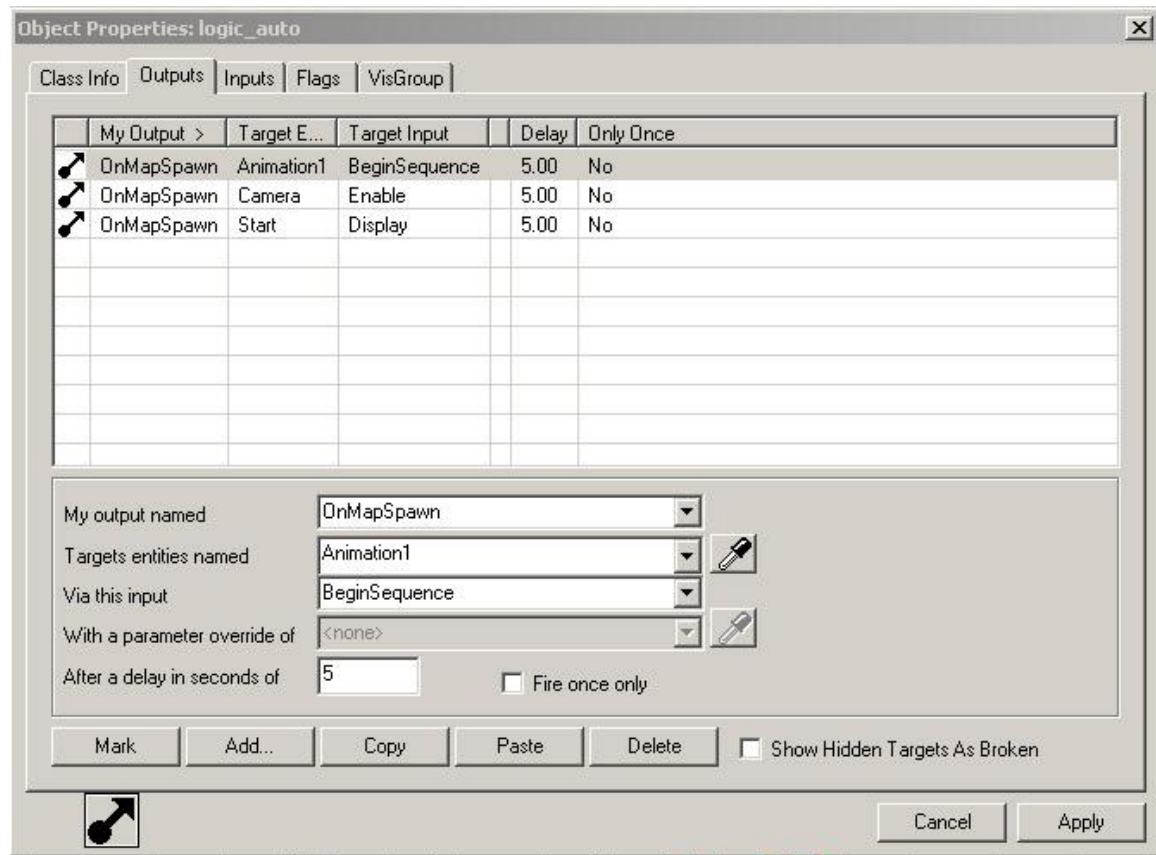
After you created the point\_viewcontrol entity, here what need to be set:

- **Name:** You define the view name, needed later for being triggered
- **Entity to look at:** This is the info\_target entity defined previously
- **Path Corner:** Required if you need to move the camera view



You also need to set a flag to “freeze” the player while it’s being used. Without this the view will not release to the new view but stay in the player control.

## Making it all work: Setting the trigger start the camera and the game text



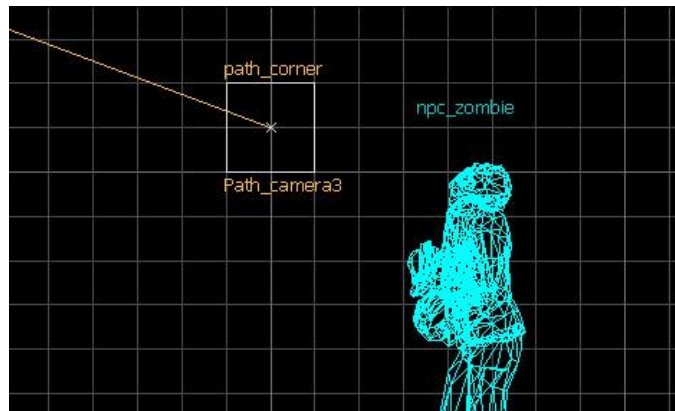
I've used for trigger the Logic\_auto entity so I could trigger these when the level is being played directly.

You select the Logic\_auto entity, go into it's properties and select the Output panel that will trigger some events.

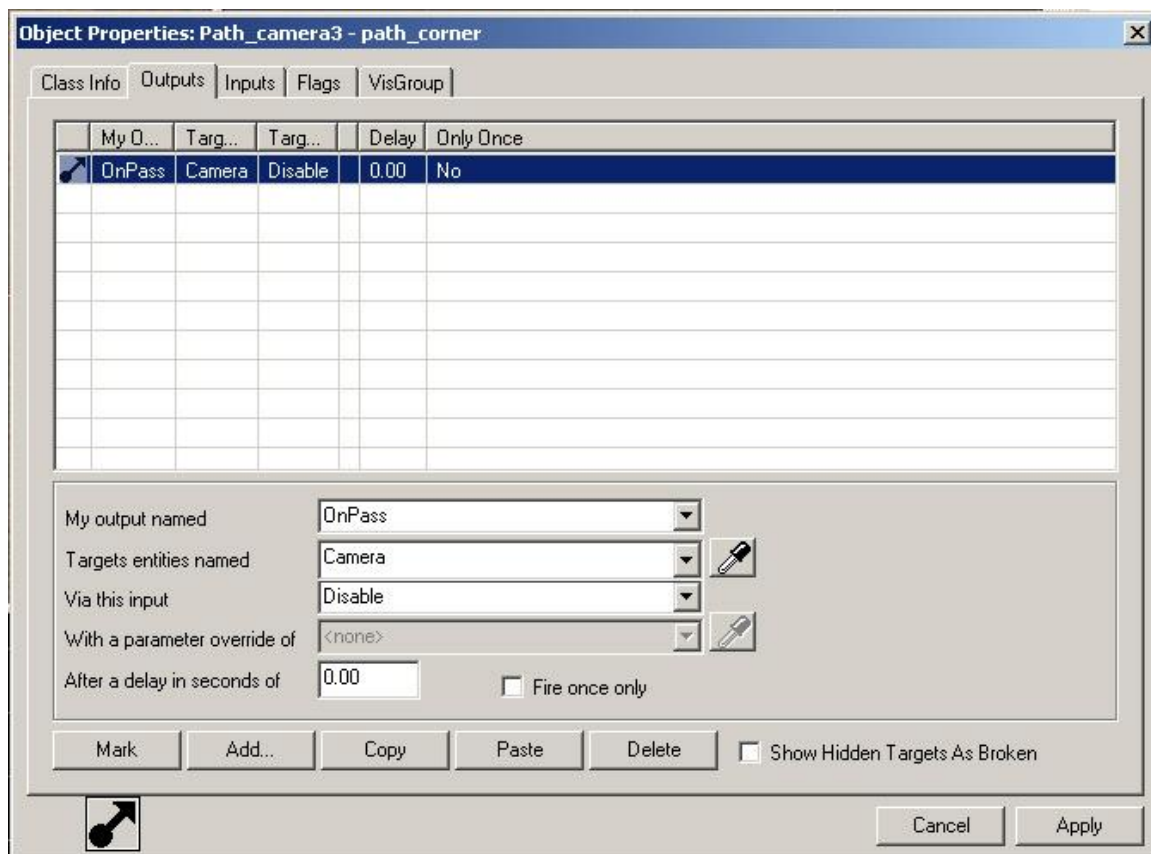
- Animation1 is the name given to the scripted\_sequence entity used previously that will trigger the animation of the character
- Camera is the name given to the point\_viewcontrol entity it will start the animation of this defined camera
- Start is the name given to the game\_text entity.

I've set a delay of 5 seconds, so after 5 seconds it will all start. The control is taken from the player and enter the cutscene. But there is still a problem! The control is not given back to the player. We will fix this next.

## Releasing the camera to the player



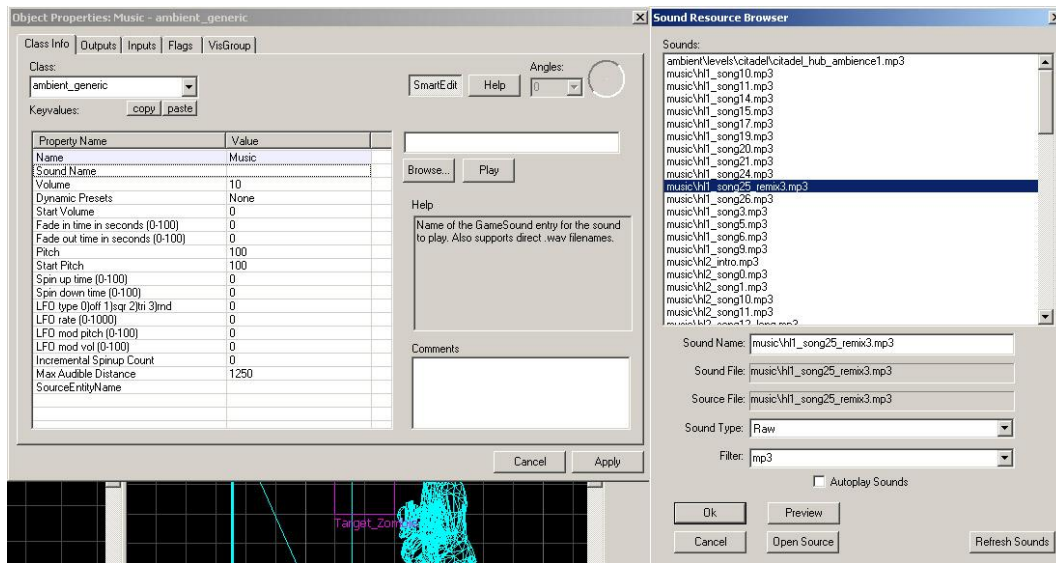
To release the camera after it's being played; you have to select the last **path\_corner** you've created and go into its properties



Select the *Output* panel, then from there you can trigger (**OnPass**), to disable the **point\_viewcontrol**.

That will release the control to the player. This can be also be used for example to open doors, when the camera is passing by them.

## What about music or voiceover for the cutscene?

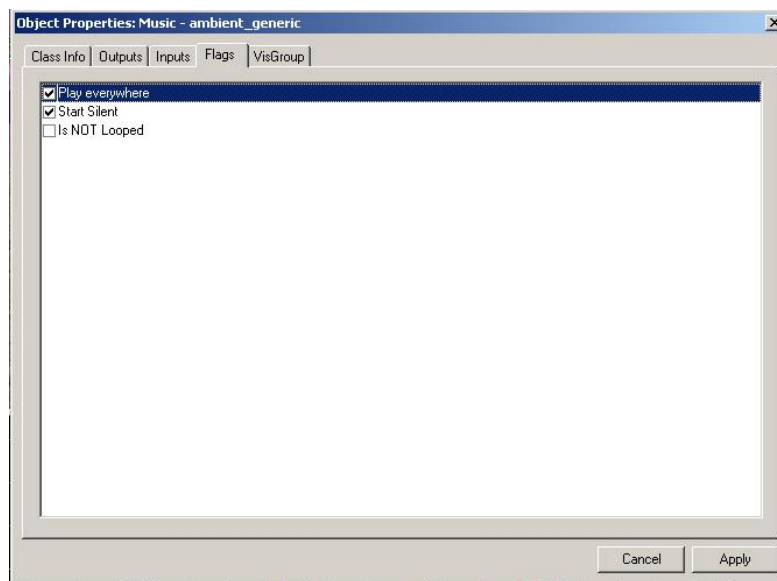


For music you can use the entity named "ambient\_generic".

For MP3 sound, these need to be triggered and cannot be previewed.

For WAVE sound, these can be previewed and they can be used directly without triggers.

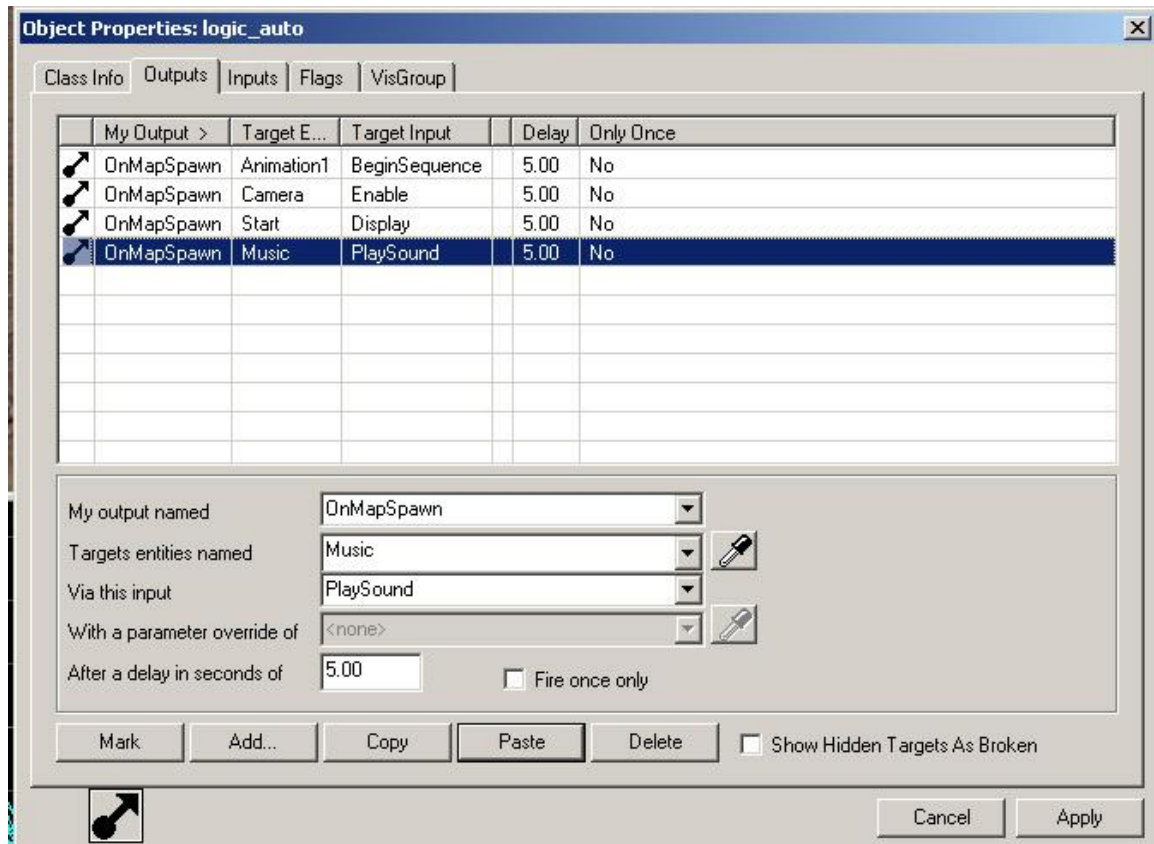
I'll use a MP3 music from half Life 2 for this tutorial. I've given the entity a name (Music)



There is some "flags" to set. Select the FLAGS panel, then set:

- Play everywhere, so the sound is hear from any location
- Start Silent so it will only start from a trigger (Mp3 need to be triggered even this is off)
- Is NOT looped, since it's a music we don't need this on. Useful for ambient sound.

Triggering the sound for a starting cutscene.



For simplicity, I used again the LOGIC\_AUTO entity. But you could use about any trigger to start the music.

In here, the ambient\_generic entity action is PlaySound to start the music.

If you've done all the steps in this tutorial, your should have a cutscene, with animation, music and text starting after 5 seconds and releasing the control after the animation of the camera is being played.