

# TNM Camera System Documentation

## Table of Contents

1	Camera Properties.....	1
1.1	bPlayerVisible.....	1
1.2	cmd.....	1
1.3	Value, eventName and InterpolationTarget .....	1
1.4	timeSmooth and timeWaitPost.....	1
1.5	bParallel (Removed).....	2
1.6	bRandom, randomCount and postRandomNum.....	2
1.7	SequenceNum and Tag.....	2
2	Camera Types and Behaviors.....	3
2.1	CAMCMD_MOVE.....	3
2.2	CAMCMD_MOVETOPLAYER.....	3
2.3	CAMCMD_PUSH .....	3
2.4	CAMCMD_FOV.....	3
2.5	CAMCMD_TILT.....	3
2.6	CAMCMD_PAN.....	3
2.7	CAMCMD_ROLL .....	3
2.8	CAMCMD_TRIGGER .....	3
2.9	CAMCMD_FADEOUT .....	3
2.10	CAMCMD_FADEIN .....	3
2.11	CAMCMD_INTERPOLATE.....	3
2.12	CAMCMD_END.....	4
3	Triggering and Ending Camera Chains .....	4

## 1 Camera Properties

### 1.1 bPlayerVisible

When this camera is triggered the player's visibility is set based on this property. If bPlayerVisible is true then when this camera is triggered the player will be visible until another camera point is triggered. Similarly if bPlayerVisible is false then when this camera is triggered the player will become invisible until another camera point is triggered. In this context the visibility of the player is the visibility to the camera as opposed to visibility in the world. It basically determines whether the camera is viewing from a first person or third person view. If the player is made to be 'invisible' pawns can still see him, his reflection will still show in mirrors etc.

### 1.2 cmd

This is an enumeration determining what type of camera this is. Camera types and behavior are covered in section 1.2.

### 1.3 Value, eventName and InterpolationTarget

Value is a floating point number that affects the behavior of the camera by acting as a parameter to the camera type. EventName is a name property used by CAMCMD\_Trigger cameras. InterpolationTarget is used exclusively by CAMCMD\_Interpolate cameras. Camera types and behavior are covered in section 1.2

### 1.4 timeSmooth and timeWaitPost

timeSmooth is then number of seconds camera should take to perform its actions. timeWaitPost is the number of seconds a camera should wait once it has completed its actions.

As an example:

Camera A's purpose is to change FOV from 75 to 10. When camera A is triggered the player's view will begin to change to 10. This change will take the number of seconds specified in timeSmooth. Once the player's FOV has reached 10 the camera system will pause for the number of seconds specified in timeWaitPost before triggering camera B (The next camera in the sequence). The total time taken between camera A being triggered and Camera B being triggered will be timeSmooth + timeWaitPost seconds.

It is perfectly acceptable to have one or both values set to zero. If a camera action should occur instantly its timeSmooth value should be 0. If a camera's action should be immediately followed by the action of the next camera in the sequence then its timeWaitPost should be 0.

## 1.5 bParallel (Removed)

At best this property should be considered experimental. Its purpose is to allow multiple cameras actions to be executed simultaneously. If Camera A has its bParallel property set to true then when Camera A is triggered it will immediately trigger camera B (the next camera in the sequence). However while this will result in Camera B running, Camera A will also run.

Example:

Camera A changes the FOV

Camera B rotates the camera

Camera A is parallel.

Camera B follows camera A in the camera sequence.

When camera A is executed it will immediately execute camera B. The resulting action is that the player's view will both rotate *and* change FOV for the duration of the cameras. Note that each camera retains its own timeSmooth. Camera A will run for its entire duration regardless of Camera B's activities. Similarly camera B can continue running after Camera A has finished or stop before camera A has finished. One must use caution when setting timeSmooths on parallel cameras as if the camera B has a shorter timeSmooth the camera sequence will continue without waiting for camera A.

Note that only the first camera in the pair must have its bParallel property set. If camera A and camera B both have their bParallel properties set then when camera A is triggered and triggers camera B, camera B will trigger camera C.

timeWaitPost is irrelevant for any camera that is mark as parallel.

## 1.6 bRandom, randomCount and postRandomNum

This property is experimental. Cameras with the same sequence number marked as random will be picked at random and executed randomCount number of times. randomCount should be the same for all points with the same sequence number. Following the execution the camera with sequence number matching that of postRandomNum will be executed. postRandomNum should be the same for all points with the same sequence number

Example:

Camera X, camera Y and camera Z all have the same sequence number and are all marked as random.

RandomCount is 5 for all three cameras.

postRandomNum is 23 for all three cameras.

When the cameras are triggered one will be selected at random and triggered. Once this camera has completed its execution it will pick another from X, Y and Z (possibly itself) and trigger it. This process will repeat until 5 cameras have been triggered and completed. At this point the camera with sequence number 23 can be triggered.

## 1.7 SequenceNum and Tag

These properties define camera point chains. All camera points with the same tag are considered part of the same chain. Camera points with lower sequence numbers are executed first in any given chain. The camera point with sequence number 0 is the start of the chain and the only camera point in the chain that can be triggered externally. Cameras should be numbered in the order in which they occur such that every sequence number, from 0 to the last

number used, corresponds to one camera (unless other properties define multiple cameras with the same sequenceNums are acceptable e.g. bParallel).

## 2 Camera Types and Behaviors

### 2.1 CAMCMD\_MOVE

A move camera will translate and rotate the view over the time specified in timeSmooth. The starting location will be the location of the previous camera in the chain. The end location will be the location of the move camera.

### 2.2 CAMCMD\_MOVETOPLAYER

A move to player camera functions exactly as a move camera except that its position and rotation are not significant and instead the destination is the origin of the player's view. This can be used to pan the camera into the players head such that his view does not change when the chain ends. It can also be used with a fade in camera to allow a fade back into the players view. Bear in mind that when the camera gets inside the players head, the back of the player's face is visible so users should consider making the player invisible for fade in transitions of this nature.

### 2.3 CAMCMD\_PUSH

A push camera will translate the view over the time specified in timeSmooth in the direction the view is facing by a distance determined by the value property. The position and rotation of a push camera is not significant.

### 2.4 CAMCMD\_FOV

A FOV camera will change the FOV over the time specified in timeSmooth to the value stored in the value property. The position and rotation of a FOV camera is not significant.

### 2.5 CAMCMD\_TILT

A tilt camera will change the pitch of the view over the time specified in timeSmooth by the value stored in the value property. The position and rotation of this camera type is not significant.

### 2.6 CAMCMD\_PAN

A pan camera will change the yaw of the view over the time specified in timeSmooth by the value specified in the value property. The position and rotation of this camera type is not significant.

### 2.7 CAMCMD\_ROLL

A roll camera will change the roll of the view over the time specified in timeSmooth by the value specified in the roll property. The position and rotation of this camera type is not significant.

### 2.8 CAMCMD\_TRIGGER

A trigger camera will cause a trigger event to occur. The event will be determined by the eventName property. This can be used to trigger other camera chains however that would make this camera the equivalent of an end camera. The position and rotation of this camera type is not significant. timeSmooth should be 0 for this camera type.

### 2.9 CAMCMD\_FADEOUT

A fade out camera will fade the players view out to black over the duration specified by timeSmooth. The position and rotation of this camera type is not significant.

### 2.10 CAMCMD\_FADEIN

A fade in camera will fade the players view out to visible over the duration specified by timeSmooth. The position and rotation of this camera type is not significant.

### 2.11 CAMCMD\_INTERPOLATE

An interpolate camera will cause the players view to travel along the interpolation path specified by InterpolationTarget. This should be the tag of the interpolation path to follow. The position and rotation of this

camera type is not significant. The value of timeSmooth and timeWaitPost are very important for this camera type. Their sum must be as close to the total time taken to travel the interpolation path, erring on the side of too short. The camera will continue to travel the path until the time runs out. In the event that the path ends the camera will jump to the start of the path and begin its path again.

## **2.12 CAMCMD\_END**

An end camera simply aborts the chain and reverts view and control back to the player.

Usage of this camera type is described in section 1.3

# **3 Triggering and Ending Camera Chains**

To trigger a camera chain from within a level simply call a trigger event using the camera chain's tag.

To signal the end of a camera chain an end camera must be triggered. This restores control and view to the player.

This should always be the last camera in a chain as not cameras after it will be triggered. All camera chains should have an end camera unless they are to be overridden by another camera chain.

A camera chain is overridden when another camera chain is triggered while it is executing. The resulting behavior is that the current camera chain will stop execution and the new camera chain will execute. Unexpected results will occur when a parallel camera is overridden so this should be avoided.