# Fundamentals of Real-Time Camera Design

**Mark Haigh-Hutchinson**
**Senior Software Engineer**
**Retro Studios Inc.**
mhh@computer.org
www.realtimecameras.com

# 1. Introduction

Camera design influences every aspect of interactive game applications. As the main avenue through which the player interacts with the game, the quality of the camera system and its effectiveness at presenting the game world to the player has a major influence on player satisfaction and enjoyment of the game. A poorly implemented camera system will cripple the game design and no amount of excellence in graphical presentation or game play mechanics will overcome it. A brief perusal of game reviews will certainly confirm this. On the other hand, a well designed camera system - one that presents the game in an unobtrusive and easy to understand manner - will elevate a good design to a great one.

In light of this, it is somewhat surprising that there is so little information available to the designers and implementers of game camera systems.

*Fundamentals of Real Time Camera Design* is aimed at expanding the knowledge of designers and programmers with regards to this crucial yet often overlooked topic. While there is common ground between traditional cinematography techniques and game camera systems (especially, of course, with respect to non-interactive cinematic sequences), the interactive nature of games and their requirements of dynamically changing game play demands different approaches and solutions. Interactive applications lag behind traditional film in that the underlying principles of real time camera design are still in their formative process. However, enough information is now available to establish a set of ground rules that will apply regardless of the game genre or presentation style.

An ideal camera system, regardless of the genre, is notable by the lack of attention given to it by the player. This is actually a difficult goal to achieve in practice, but certainly worth striving towards.

Camera design is so fundamental to the game design process that there needs to be production resources dedicated solely to this aspect from the initial design, through prototyping and finally implementation of all interactive game applications. This lecture aims to increase awareness of the importance of this aspect of game development, and to raise the bar for camera system quality.

# 2. Camera System Overview

The camera system is responsible for management of all camera-related features of the game. This encompasses both cinematic sequences and real-time game play. It defines how the game world is presented to the player under a variety of different situations.

The camera system provides information to other game systems including rendering of the current world-view, mapping of controller input to player motion, and so forth. Additionally, the camera system must provide mechanisms by which the designers may dynamically alter camera behavior. This is often required by changes to the player character, environmental changes such as confined spaces, and so forth. Altering the camera behavior dynamically would also require that the camera system manage prioritization of competing camera behavior requests.

To this end the camera system will often provide facilities to allow designer specified scripting of camera motion, positioning, orientation, and other properties at a game wide level combined with scripted areas that specifically override or change behaviors. Considering the close links between cameras and player control (as shown later in this paper), it is common to also allow the camera system to dictate the relationship between controller input and player motion.

A successful camera system is dependent upon collaboration between designers, artists, and programmers. Each discipline is required to participate in the presentation of the game to satisfy the competing constraints of game play design, aesthetics and technical limitations.

# 3. Camera Fundamentals

A game camera defines the way in which we present a view of the game world to the player. As expected, it must specify a number of properties regarding the position and orientation of the viewpoint within the game world. More importantly, the *presentation style* (see below) dictates how the game world is projected (or *rendered*) onto the display device. Other rendering properties might include field of view, aspect ratio, and so forth. We may also consider the ways in which the camera reacts to player motion and other game play elements, commonly referred to as the *camera type*.

Several defining features can be used to differentiate the myriad of camera types available for use in games. The simplest of these would depend on whether the player is able to control their character, i.e. whether the camera is **interactive** or **cinematic**.

*Cinematic cameras* within games share many characteristics with their real world counterparts. Essentially, any behavior exhibited by a real world cinematic camera is usually replicated in the game version. The same rules of cinematography apply, with the added bonus that game cameras may move or behave in ways that real cameras may not (although this should be handled with care since viewers are not very familiar with such behaviors). In the same manner, cinematic game cameras display their scenes in the same way every time they are used. Often the aspect ratio or other ancillary presentation style effects are changed when using cinematic cameras as an additional cue to the player that they no longer have control over their character, as well as emulating the form of traditional film cameras. As display devices change over to using a widescreen aspect ratio, this distinguishing feature may no longer apply.

*Interactive cameras* main distinguishing characteristic is that they change their behavior in response to real time game events, whether dictated by the player's actions or some other element of the game. Interactive cameras have a more difficult task in that the actions performed by the player are infinitely variable, as are the situations in which they occur.

While both kinds of camera share many properties with regards to their design and implementation, this paper is primarily concerned with interactive cameras. References to cinematic cameras are for completeness only.

## 3.1. Camera Presentation Styles

We can consider the presentation of the game world as a major distinguishing characteristic of different camera types, since it will greatly affect both camera motion and orientation determination. Naturally, it will also affect any calculations performed to transform coordinates from world space to screen space and vice versa. This presentation style is usually restricted to either an orthographic or perspective rendering of the game world, and may be referred to as **2D** or **3D** cameras for our convenience. A hybrid presentation form is also possible, where either the game world or game objects are rendered as a perspective view, and the other elements are presented as two-dimensional. This presentation style is referred to here as **2.5D** cameras.

Technically all of these presentation styles can be implemented in the same generic manner, since only the viewing transformation varies.

## 3.2. Camera Types

Regardless of the presentation style, there are three main types of game cameras: **First Person, Third Person**, and **Cinematic** cameras. Note that hybrid forms of these cameras are becoming more prevalent in games, where the camera will change according to either game play demands or player desire. These hybrid cameras combine elements of first person camera control with third person camera positioning (or vice versa) and can be useful when game play requirements include long range weapon aiming using free look or similar control schemes. They have other desirable properties including a more immersive experience than pure third person approaches.

### 3.2.1. First Person / Point Of View

Popularized by a number of games such as DOOM *et al*, the **First Person Camera** represents a view of the game world directly through the eyes of the protagonist. For this reason it is also sometimes referred to as a **point of view** camera. Indeed a whole sub genre of game is associated with this particular camera type, the so-called *First-Person Shooter*, or **FPS**. However, there is no restriction as to the type of game made using this type of camera.

Because of the limited view that is offered to the player by first person cameras, we often have to introduce new elements into the game in order to further enhance the representation of the player's physical presence in the world. Obvious examples would include rendering the player's weapon, helmet (including, perhaps, a *Heads-Up Display* or **HUD**), hands/arms, and even sometimes their feet. Invariably these devices rarely add much to the sensation of being within the world, and oftentimes prove to be a distraction. Other attempts would include displacement of the view whilst moving to simulate head-bobbing (which may be easily overdone, inadvertently causing nausea in some players), camera shaking in response to game events, etc. Additionally there are audio cues that may be applied especially with regards to surround sound positioning of sound effects, reverb, footsteps, and so forth.

This view has often been likened to wearing "blinders"; that is to say, the horizontal view of the world is usually very limited. If the field of view is reasonably large, say 90 degrees or so, then a "fish-eye" view of the world is presented with other undesirable properties (e.g. motion sickness due to exaggerated rotation, distortion of vertically aligned objects, etc.). These problems mainly relate to the actual display mechanism itself – a 2D projection of a 3D environment.

When implementing a first person camera it is advisable to separate the camera from the head of the character, even though its position is dependant upon the latter. The reason for this is because there should be some lag in terms of vertical axial motion, to dampen the effect of the player walking over small obstructions in the game world. Otherwise, the camera would react to every small bump, just as the player does, which would be jarring and result in a discontinuous camera. Additionally, it is necessary to allow reorientation of the camera independent of the player character orientation in order to provide automated pitch control or player-controlled use of *free-look* (q.v.).

First-person cameras are also used in vehicle simulators, although usually in conjunction with a variety of selectable third person camera viewpoints. The main view in this case may even be considered a third person view even though it represents the player’s viewpoint from piloting the vehicle. One major difference is that roll around the forward axis of the camera is permitted and usually required.

### 3.2.2. Third Person / Detached camera view

A third person camera is one in which we can (usually) see the player character from an external position. Ideally, we can view both the player, and the environment in which they interact, in a clear and easy to understand fashion. Third person cameras are the catchall category for all camera types but we can enumerate a number of specific examples:

- Behind the character: Either with a fixed position with respect to the character orientation, or with rotational lag around the vertical axis of the character.
- The camera is positioned so as to give an over the shoulder view, typically with little or no orientation lag.
- Medium distances away from the character such that animation, etc. is still clearly visible and a good view of the surrounding area is possible. This includes being able to observe the character's feet, which is important in order to judge jumping when player control over this aspect is desired. Automated control schemes that cope with situations such as jumping alleviate the need for this particular viewpoint as the timing of the player action occurs precisely as required without player intervention.
- The camera is located far behind the player character; this is greatly dependent upon the environment, and is a common solution in vehicle games.
- The camera maintains a fixed position with respect to the character position (usually used for replay cameras). Examples include rear facing (flight simulator), fender (racing), etc.

### 3.2.3. Cinematic Cameras

Although open to interpretation, in this paper we are going to adopt the convention that a cinematic camera by definition is one in which the view of the world is displayed in a *non-interactive* form. Although the underlying camera may use exactly the same technology as the other types of cameras, the main difference is that the view of the world presented is outside of player control. Note, that it is possible to have "in-game" or "real-time" cinematic events that allow for player control, but that these present their own unique problems. In addition, it's entirely possible (and often used in the movie industry to varying effect) to have a first person cinematic camera.

The reader is referred to *Grammar of the Film Language* (see *Further Reading* section) for extensive information concerning cinematographic techniques, as that subject is not the focus of this paper.

# 4. Camera Design Principles

Camera design like many aspects of game development is both somewhat subjective and dependent upon the game genre or requirements of game play. Some design choices depend upon the presentation style (see above); others according to the game genre (e.g. sports vs. adventure games, etc.). Nonetheless, it has proven to be the case that there are general rules that may apply irrespective of the game genre. Some of these rules are dependent upon the presentation style, and some must give way to greater design requirements. Keep an open mind and use judgment as to which works best for the particulars of your game.

It is unlikely that a case will be encountered that breaks this set of rules, though it can sometimes be necessary to adapt to the specifics of the game play. These "rules" are intended more as guidelines, since game play should dictate camera requirements whenever possible:

## 4.1. *Always keep the player character in view.*

Total occlusion by geometry or other game objects (or alternatively, view frustum culling) of the main player character will disorient and confuse the player, yet surprisingly few games pay attention to this essential point. Note, this does not necessarily mean focus on the player character directly, nor does it mean that partial or temporary occlusion is not permissible. The determination of the look at position of the camera, and how this pertains to the player character is too large a topic for this paper (see *Further Reading* section below).

## 4.2. *Prevent the camera passing through (or close to) game objects or physical environmental features.*

If the near plane of the camera view frustum intersects render geometry, unwanted visual artifacts will be produced. These will certainly detract from the graphical quality of the game, and at best seem very unprofessional. This problem is completely avoidable; it should be considered as simply unacceptable in modern camera systems. A passable solution to avoid this problem is to apply transparency effects to the geometry in question. By effectively removing the geometry (and indeed actually doing so according to camera proximity), the camera may be allowed to pass through without creating distracting visual artifacts.

## 4.3. *Do not require the player to manipulate the camera simply to play the game – unless it is a design requirement.*

For third person presentations, the camera system should be able to choose the most appropriate solution automatically, even in the most complex of situations. It is permissible to allow the player to control the camera when it is absolutely necessary or desirable to do so, but it should certainly not be required. Many games adopt a

laissez-faire attitude to camera manipulation, which is unfortunate given its importance. If camera manipulation is allowed (or specified as part of the game design) then care must be taken to ensure the player cannot position the camera outside of the world geometry or into a position occluding the player character. The range of motion and method of manipulation must be carefully matched to the game design and technical requirements. An obvious exception to this rule would be first person games, where camera manipulation is an important part of regular game play. Even in this case, there are times where the camera orientation may be automatically adjusted to aid the player, without overriding player control. Usually this is a subtle assistance to the player in clearly defined situations. Motion up or down steep inclines may induce some vertical pitching of the camera to assist in aiming or navigation of the world, for example.

## 4.4. *Allow camera manipulation when possible, or dictated by game design requirements.*

Certain game play situations will disallow camera manipulation, but certainly denying the player this control can often seem restrictive. Naturally, we should strive to present a view that doesn't require this manipulation, but there are certainly cases where the design would demand the player to examine their environment in detail. It is also true that it can be difficult to judge player intent and camera manipulation allows a greater sense of control and determinism for the player. There can be nothing more frustrating to a player than being prevented from seeing the world in a manner relevant to their current situation or intended action. The problem, however, is that the camera designer should not abdicate this responsibility as a solution in itself. Additionally, restrictions to camera manipulation after such control has been allowed may be confusing and frustrating to the player.

## 4.5. *Minimize camera motion whenever possible.*

This is especially true of cinematic sequences, but it is true to say that camera motion should be avoided unless it would either result in the camera being left behind by the player character, or the camera interpenetrating the player character. Slight or unintentional camera motion caused by reactions of the player character to environmental factors (or noise from player controller inputs) should be avoided. Similarly, if camera motion is directly linked to that of the player, it often results in an "unnatural" or "stiff" feeling to the camera.

## *4.6. Ensure camera motion is smooth.*

Smooth, frame-coherent motion of the camera is necessary to avoid disorienting or distracting the player. Of course, smooth motion is normally achieved via velocity dampening which has the adverse effect of allowing the target object to either accelerate away from the camera or worse, overtake and interpenetrate the camera itself. Nonetheless, smooth camera motion is of great importance and there are techniques available to assist the camera in cases where the player character is subject to rapid acceleration. Low-pass filters can help smooth out irregularities in movement especially when the camera motion is tied directly to that of the player character, where noise from unwanted player input may cause slight motion.

## 4.7. *Limit the reorientation speed of the camera.*

Unless it is under direct player control, rapid or discontinuous reorientation of the camera is disorienting and confusing. Reorientation of the camera causes the entire rendered view of the world to be redrawn; since the direction in which camera is facing ("through the lens" as it is often called) determines the part of the world to be drawn.

By limiting the velocity of this reorientation, this effect can be minimized. In third person cameras, this may be at the cost of losing sight of the target object for a short period. Instantaneous reorientation is permissible when the *cut* is made in an obvious fashion such as in combination with a repositioning of the camera (see below), but only when the new orientation is retained for a sufficient period to allow the player to understand the new situation. Retention of the player *control reference frame* (see section 7) can assist in this regard. Regardless, instant reorientation should occur infrequently, and never in quick succession. Reorientation to a new desired heading should always move in the shortest angular direction.

## 4.8. *Limited roll should be allowed in regular game cameras.*

That is, no rotation should normally occur around the forward axis of the camera. Again, this is distracting and disorienting during regular game play, but within cinematic sequences, it is permissible. Some games have used this effect to intentionally disorient the player, but it should be used sparingly. Flight simulators and their ilk are an exception to this rule, as they usually present a view of the game world from the point of view of the vehicle. Even so, many people react poorly to extreme roll, so in external vehicle views it may prove wiser to only allow a limited amount of roll to emphasize vehicle banking, etc.

There are cases where roll can be used to a limited degree in order to emphasize differences in game play or to underscore an emotional state of the player character. Clearly, roll is perfectly allowed during cinematic sequences.

## 4.9. *Do not allow the camera to pass outside the game world.*

In the vast majority of cases, the camera is required to remain within the visible geometry of the world to prevent the player from seeing the construction of the game environment and thus destroy the illusion of the game play. There are limited cases where it is necessary to position the camera outside of the main game environment but only when care has been taken to hide construction details from the player.

## 4.10. *Do not focus directly on the player character when it is moving.*

This pertains to third person cameras that are often made to look directly at the player character, i.e. the position of the player character does not vary in screen space. Whilst this might seem initially to be correct, and it does present the player character well, it is actually undesirable in many cases. Typically, we need to be looking ahead of the character motion in order to evaluate the position of the character within the environment and to anticipate future actions. However, the rules governing the amount of look ahead are complex and beyond the scope of this paper.

## 4.11. *Retain control reference frames after rapid or instantaneous camera motion.*

This topic is covered in detail in section 7.2.

# 5. Game Genre Cameras

The variety of camera types to be found in games is greatly dependent upon the genre and presentation styles chosen, although it is becoming more accepted to combine multiple genres and thus camera types within a single game. Nonetheless, certain camera types are more amenable to particular game play types, and some of the more common camera behaviors will be enumerated here.
There is nothing to say that a specific camera type must be used for a certain genre of game, of course. Experimentation in this regard is certainly encouraged. A completely new perspective (literally) can sometimes radically change a player's perception and enjoyment of a game.
Choosing the most desirable camera types according to the requirements and aesthetics of a particular game genre is a difficult process. A full description is beyond the scope of this paper and the reader is referred to the further reading section for more information. A brief summary of the most salient features for some of the more commonly used game genres follows:

## 5.1. Action Adventure

- Motion and orientation lag are important to retain a "loose" feeling
- Elevated position above character to allow a good view of the environment
- Look-at depends on action; usually ahead of the character as it is moving
- Generalized solution difficult due to environmental complexity and game play demands
- Minimize reorientation and repositioning of the camera as the character turns toward the camera

## 5.2. 3D Platform

- Elevated position above character to allow a good view of the environment
- Often moves along a pre-defined path or towards a stationary vantage point
- Environment view is important to allow player judgment of distance, jumping etc.
- User control of camera positioning and orientation is important unless a good view can be guaranteed

## 5.3. Flight Simulation

- Requires multiple viewpoints to represent different views from outside and inside the craft
- Roll is permissible when view from within the craft; in external views this can sometimes be distracting to novice players
- Look-at position is normally ahead of the craft depending on maneuver
- Replay cameras are often required to review game play or illustrate mission completion, etc.

## 5.4. Racing

- Multiple viewpoints around or inside the vehicle
- Defined camera paths are usually available since the movement of the vehicle is restricted
- Motion and orientation lag add to a better feel for the responsiveness of the vehicle
- Replay cameras are a major component and can be automated or under player control

## 5.5. Fighting

- Must frame multiple characters, often requiring rapid camera motion or zoom effects
- Focal point between enemies rather than one specific character; usually allows a better view of both characters
- Open environments help with the camera motion necessary for framing
- Cinematic approach to non-interactive sequences so as “throw” moves and so forth where the movement of one or player characters may be pre-calculated.

## 5.6. Role Playing Games (RPG/MMOG)

- Elevated, distant position to show a larger section of the environment
- Often entirely user controlled due to multiple enemies and environment complexity
- Some RPGs use fixed vantage points during combat
- Navigation difficulties often prevent intelligent camera control in such complex situations

## 5.7. Sports

- Matches TV presentation style to enhance player familiarity
- Multiple viewpoints are often allowed for grandstand views as well as individual players
- Elevated positions mostly although sometimes from the perspective of a player
- Replay cameras are a large component to once again emphasize the comparison to a “real” sport

# 6. Camera Design Process

Camera design is a combination of aesthetic sensibilities tempered by the requirements of game play. As with most design skills, observation and analysis of existing systems is perhaps one of the most important elements to practice. When playing any new game, intentionally take some time to manipulate the player character in ways that tax (or break) the camera system. Observe how the system handles difficult cases (such as fast motion or confined environments) as well as simply noting how it behaves with respect to regular character motion and orientation changes. Determine how your camera system would deal with similar cases.

When approaching the design of camera behavior we must look at general camera behavior as well as cases that are particular to a specific area or section of the game.

## 6.1. Design Process Overview

- Examine high level design goals; look at the overall scope of the game and how that impacts camera design on a global scale. What commonality is present in the differing parts of the game?
- Evaluate player character abilities; How the player character moves directly influences camera design. Moreover, do the player abilities change over time?
- Determine scope of environments; the camera must work within the confines of the game environment. Consider the implications of the environmental design upon camera placement and motion.
- Define base camera behavior(s); Determine the fundamental desired camera behavior used in the majority of the game. This will be the basis of all camera behaviors in the game.
- Determine area-specific requirements; Each area of the game should be analyzed for changes to the base camera behavior (e.g. confined tunnels). More details on this process are found in section 6.2.
- Technical evaluation of cameras; Once the desired camera behavior has been determined, the technical feasibility of the approach must be verified in the prototyping stage.

## 6.2. Area-specific camera requirements

For each unique area of the game where a general camera solution cannot work successfully, we need to examine the specifics of that area in terms of both the environment and the actual game play elements. An example might be a case where the player's motion is restricted within, say, confined tunnels and we wish to present a view of the tunnels from a distant or elevated position.

We would start by analyzing the particulars of how the player moves, where the camera needs to be positioned during that movement, where enemy AI objects may be placed and so forth. This should occur before any firm commitment had been made to the physical layout of the environment, art resources, and so forth. Game play requirements such as these will normally have a higher priority than aesthetic requirements, but they are bound by the limitations of available technology. In many cases aesthetics overlap with game play in the sense that many

aspects of game play rely on a visually pleasing and understandable view of the game world, but this doesn't mean that cinematography rules may be applied, for example.

After designing camera systems and behaviors for some time, this process can be reasonably standardized. Consider the following questions so that a full understanding of any special game play situation can been gained before starting its camera design:

## 6.3. Camera Design Questions

- How does the player character move?
- Has character motion and abilities been finalized? Changes to either of these components may greatly influence camera requirements.
- Where should the camera be located compared to the character?
- What is important for the player to see?
- How should the player character be controlled?
- Does the situation demand a different kind of control scheme from regular game play? (Such as world-relative, for example).
- Within what kind of environment is the character moving?
- Does the environment influence camera positioning or orientation? For example, the character could be moving through the surface of a water plane, thus requiring the camera to avoid interpenetration of its near plane. A confined, closed space will require a different camera solution compared to free form flight over a wide-open space.
- Does the camera need to be position outside of the environment and if so, how will that be presented to the player? How will the camera transition between regular game play and this special case?
- Does the camera need to avoid complex geometry or large game objects?
- Are there changes that can be made to the geometry to simplify camera design or implementation?
- What kind of presentation would work best; does the player character have differing abilities that would demand wildly different camera behaviors?
- Can interpolation methods be guaranteed to work given the environment?
- If the camera is forced to pass through game objects or geometry, can their rendering be altered to make them transparent or otherwise unobtrusive without adversely affecting game play?
- Is it possible for the camera to be prevented from moving to its desired position, and if so, how should this be dealt with?
- Do we wish to allow camera manipulation including repositioning? To what extent will this be allowed?
- Is the most dynamic camera choice the best for game play clarity? Remember that camera motion should be minimized where possible.

This is by no means an exhaustive list, but it should provide enough information for the initial pass at a potential camera solution. It's advisable to create a document that outlines camera capabilities and how the game design can work within those constraints. This document will be a useful crib sheet for both level and camera designers in order to ensure that level construction take account of technical limitations.

## 6.4. Environmental concerns

No camera system is likely to be able to dynamically cope with the arbitrary complexities of game environments without either predetermined motion paths or extensive analysis (and storage) of possible camera positions. That is, the camera system has to both navigate the environment and present a view of the world that satisfies both aesthetic and game play requirements. There will undoubtedly be cases where the physical nature of the environment precludes camera placement to satisfy either (or both) of these conditions. This should obviously be avoided whenever possible, and is a good case for prototyping of camera placement and motion before committing to a particular environment. Attempting to find camera solutions after artwork has been completed can lead to both designer frustration and additional work for the artists. Design the environment around game play and camera limitations if possible, but there will always be compromises in the process.

## 6.5. Technical limitations

Just as any other game system, the processor and memory requirements for a camera system must be considered when determining a solution for particular game play scenarios. If a custom camera solution is required for a particular game play situation, the engineering effort required for that solution as well as the technical requirements must also be considered. Often an engineering solution may be avoided by small modifications either to the game play requirements or to the environment itself. The easiest camera problems to solve are the ones that you don't have; which is another way of saying that design to avoid problems whenever possible. Good communication between the three disciplines of programming, design and art is as important with camera system design and implementation as any other major game system.
Therefore, we need to continually assess the amount of system performance dedicated to the camera system. Running multiple cameras simultaneously will naturally affect this, as will the actual camera types involved. There are four things to consider with regards to technical limitations of the camera system:

- Processor usage. Camera systems are notoriously expensive when they have to test or explore the environment, as this usually entails casting rays through the world to determine line of sight amongst other things. Some systems call for multiple rays to be cast every frame, and this is often an area of optimization through processor amortization.
- Memory usage. Often camera systems require additional information about the environment in order to adjust camera behavior, avoid collisions or player occlusion, navigate complex geometry, and so forth. Separate collision data may be too expensive depending on the complexity of the environment and amount of detail required.
- Scripting capabilities. How much control can be given to designers to modify camera behavior according to game play requirements? Alternatively, must additional functionality always be programmed by hand? Data driven solutions offer greater flexibility at a potentially high cost in maintenance and learning curves for the designers.
- Debugging capabilities. Understanding the reasons behind camera behavior is key to effective control of the camera system. Are we able to expose or reveal internal workings of the camera system in an understandable fashion? Are there methods to manipulate a free form camera to observe game events outside of the regular camera system?

## 6.6. Production Pipeline

An important aspect of camera system design is working together with other disciplines in the production of the final data to be used in the game. There are a number of different, yet equally important stages, which define the production pipeline. A camera system designer needs to be aware of all of the production stages and to stay involved in all steps of the production pipeline.

Game development is a difficult, expensive, and lengthy process; thus, it is necessary for camera system designers to remain vigilant throughout the process with regards to the demands placed on their camera system. Constraints upon usage of the camera system should be determined early in the project, and enforced with few exceptions. Naturally, the designer cannot anticipate all future camera requirements and should be prepared to add additional functionality as needed. However, be wary of implementing major changes later in the project, as with any major game system. Relatively simple game play changes may have inadvertent consequences on a game-wide scale, and this is certainly true with regards to camera systems. Something as innocuous as changing the amount of time that a door stays open, for example, can result in cameras being separated from their target object.

However, changing fundamental game play requirements late in a project has been the death knell for many games. Good communication between the different disciplines involved in the production of the game is essential to minimize surprises.

# 7. Player Control

The topic of player control is involved and genre specific. For the purposes of this paper we will concentrate on a limited subset pertaining to third person cameras, that is the notion of the *control reference frame*.

## 7.1. Control Reference Frame

This dictates the mapping of player controller values into desired motion of the player character. Whilst this may seem an obvious concept, many games fail to ensure that player controls work consistently under varying game play situations especially where the camera orientation changes with respect to the player character.
There are five commonly used control reference schemes:

- **Character-relative**: This is most often used in First Person games though occasionally in third person games. The direction of the character, and thus the control reference frame, are not tied to the camera view.
- **Screen-relative:** This uses the orientation of the current camera in the world to dictate the directions of character motion. It would mean that left and right controls would move the character parallel to the plane of the view of the world. Side-scrolling games are an obvious example of this scheme.
- **Camera-relative**: This reference frame looks at the relationship in space between the character and the camera, and uses that as a basis for controls. In this situation, left or right controls would inscribe an arc of motion around the camera position (if it were stationary).
- **World-relative**: For this scheme, an arbitrary reference plane can be specified in the world coordinate space. Examples would include half-pipes or confined spaces where the player needs a specific sense of direction irrespective of the character motion.
- **Object-relative**: where the control mapping depends on the position of a different game object relative to that of the player character.

Control reference frames are often calculated in both 2D and 3D terms. If the player character retains a consistent orientation with respect to the world up-axis (as is usually the case with humanoid characters, for example), then the 2D reference frame is most frequently used to dictate the control mapping.

One of the main problems encountered by many third person games, is changing the control reference frame in an arbitrary manner. Most often, this occurs when the camera cuts to a new position, and the new position is on the opposite "side" of the player character from its previous position. For camera-relative or character-relative control schemes, this will often cause the player's current joystick position to force the character to move back towards their previous location. This is ultimately very confusing and disorienting for the player and is counter to basic game design principles.

To put it simply: never instantaneously change the control reference frame. Either interpolate the control frame to match the new position, or allow the previous control reference frame to remain active until the player is reoriented. Sometimes this involves waiting for the character to move in a different orientation to the previous one, or it may be time-based.

If external forces are moving the player, it is even more important to maintain a consistent reference frame so that the player is able to correctly assess how to move relative to the forced motion.

## 7.2. Player intent

The most important part of the control reference frame is that it should always reflect the intended action of the player. Changes to the frame should be minimized or controlled using one of the methods outlined above. With character-relative control schemes, the position of the camera relative to the player character can dramatically change the player perception of the control reference frame, even though technically it hasn't changed. Camera positioning in this case has more influence than the actual control reference frame.

Use of a character-relative control scheme combined with a fixed camera viewpoint is an example of problematic camera design. In the past, this approach has been taken when a dynamically positioned camera was not feasible due to rendering limitations. Such situations change the player's perception of how the character is controlled in a non-intuitive fashion. Perhaps more important than the usage of the reference frame in cases such as these, is that of *control consistency*. Preserving the intended movement of the character under control reduces the disorientation that many players feel. This is also true of the case where the player character moves to a position in close proximity to the camera.

# 8. Conclusions

Clearly, camera design greatly impacts game play as it directly controls the presentation of the game world to the player. Moreover, since we are only able to present a limited view of the world at any given time, the determination of what to display is of paramount importance to the player. This aspect of game design cannot be left to chance. Designers must make a concerted effort to ensure that players are adequately able to understand the positioning of their character within the game world. Additionally, the control of the player character can be greatly dependent upon the control methodology used with respect to the game camera system. This is especially true of third person cameras where camera positioning has an even greater influence over player perception.

In order to successfully present the game world we need to allocate dedicated resources to camera design. This encompasses determining that the environment is constructed in a suitable fashion, that predefined or variable camera behaviors are used appropriately to illustrate specific game play elements, and that player camera manipulation works as expected. Consistency of presentation is also an important factor in ensuring player satisfaction; dramatically differing views of similar game play situations will confuse and frustrate the player.

It is also necessary to allow adequate production time for quality control over camera usage. It is noted that there is no one single camera solution, and that time is required for experimentation in this regard.

We have also recognized that camera design is an essential part of the overall game design task, as should be scheduled as such. In order to minimize production problems camera systems should be prototyped early in the development cycle before committing to the final environmental construction.

Finally, we have come to understand that successful camera systems are an aid to the player and should not adversely influence game play choices. Our goal is to provide a system that presents the best view of game play without causing player frustration. Additionally we wish to allow direct manipulation of the camera by the player as required by the constraints of the overall game design.

Should we succeed in our goals, the player will not even be aware that there is a camera system at all.

# 9. Further Reading

At the time of writing, there are limited amounts of resources currently available with respect to real time camera solutions. A large body of work is available concerning cinematographic techniques and the reader is encouraged to review that material. Additionally, there are some notable articles concerning the more interactive nature of game cameras to be found in the following references. Some also contain discussions of player control, particularly the Game Programming Gems series:

*Game Programming Gems Series*

- Published by Charles River Media
- *Volume 1:*
    - 4.3 Camera Control Techniques – Dante Treglia II
- *Volume 2:*
    - 4.11 Classic Super Mario 64 Third-Person Control and Animation – Steve Rabin
- *Volume 4:*
    - 4.1 Third Person Camera Navigation – Jonathan Stone

*AI Game Programming Wisdom Series*

- Published by Charles River Media
- *Volume 1:*
    - Camera AI for Replays – Sandeep Kharkar
- *Volume 2:*
    - An AI Approach to Creating an Intelligent Camera System – Phil Carlisle
    - A Modular Camera Architecture for Intelligent Control – Saneep Kharkar

*Real-Time Cinematography for Games*

- Published by Charles River Media
- Written by Brian Hawkins

*GDC 2004 Proceedings*

- Full Spectrum Warrior Camera System
- Written by John Giors

*Grammar of the Film Language*

- Written by Daniel Arijon

A more thorough discussion of the topics raised in this paper, as well as many other pertinent issues for the design and implementation of real time camera systems will be found in the following forthcoming title:

*Real-Time Cameras*

- Published by Morgan Kaufmann
- Series in Interactive 3D Technology
- Written by Mark Haigh-Hutchinson
- www.mkp.com
- www.realtimecameras.com

# 10. Thanks

The author wishes to thank *Nintendo* and *Retro Studios Inc*. for allowing publication of this paper, and the GDC 2005 Selection committee for their selection of this presentation.
*John Giors, Ryan Harris, Akintunde Omitowoju,* and *Kynan Pearson* are also thanked for their help in reviewing both the paper and the presentation.