CameraBots: Cinematography for Games with Non-Player Characters as Camera Operators

James Kneafsey
Institute of Technology Blanchardstown,
Dublin 15,
Ireland.
353 1 8851139
james.kneafsey@itb.ie

Hugh McCabe
Institute of Technology Blanchardstown,
Dublin 15,
Ireland.
353 1 8851089
hugh.mccabe@itb.ie

ABSTRACT

Cinematography describes principles and techniques pertaining to the effective use of cameras to film live action. The correct application of these principles and techniques produces filmed content that is more engaging, compelling and absorbing for the viewer. 3D computer games employ *virtual cameras* in order to provide the player with an appropriate view of the game world. These virtual cameras can simulate all of the functionality of their real-world counterparts yet little effort is usually made to incorporate cinematographic techniques and principles into their operation. We introduce *CameraBots*, autonomous camera operators modelled closely on the *non-player characters* (NPCs) or *bots* already present in many games. CameraBots can perform a larger set of operations than their real-world counterparts since they are not subject to the same physical restraints. Thus, cinematographic principles can be applied to camera work with relative ease by reusing bot program code already present. Our system contains a *director* module and a *cinematographer* module which together are responsible for coordinating the CameraBots in a manner consistent with cinematography rules and practice. It is designed in a modular manner such that it may be applied to numerous computer games with little modification.

Keywords

Virtual Camera, Virtual Cinematography, Non-Player Characters, Bots

1. INTRODUCTION

Cinematography can be defined as the art of film making [3]. It includes guidelines for lighting arrangements, the positioning of actors and props and also for operating the camera. By using the extensive set of camera operations such as close-ups, pans, tilts and zooms afforded by cinematography, a game has the potential to: add dramatic emphasis where necessary; elucidate unclear events in the plot; ensure that the player does not become disorientated; and even evoke

Proceedings of DiGRA 2005 Conference: Changing Views – Worlds in Play.

emotional responses in the same way as a motion picture does. A motion picture may have one or a number of cameras operating at any given time [16]. The footage from the cameras is edited after filming to produce a series of moving pictures from a number of different angles. We aim to achieve the same effect in a game by placing a number of CameraBots in the virtual world.

We see applications for this work above and beyond simply providing a more cinematic view of proceedings for single player games. Of increasing interest is the idea of games as a spectator activity. A virtual cinematography system such as the one we are developing could be extremely useful in this context also, particularly as the constraint of having to provide views consistent with the demands of actually playing the game no longer exists. Similar considerations apply in the case of generating playbacks of a game sequence and in the case of the emerging field of *machinima* [15].

In the following section we describe some basic principles of cinematography that are relevant to 3D computer games. Next we discuss some related work and examine the camera work used currently in games. Following this we consider the relevance of bots to cinematography and outline our virtual cinematography model. Finally we conclude and outline our future work.

2. CINEMATOGRAPHY

Using cinematography as a guide for filming subject matter should result in a coherent depiction of the action and often includes both static and moving viewpoints [3, 16]. Knowing whether to use a static or moving camera and where to place it relative to the subject at each instant are the principal issues that cinematography addresses. Some basic definitions are necessary. A *scene* is a single setting and its depiction. A scene will typically include *props* and one or more *characters* whose actions are of interest. A *shot* is a continuous view filmed by one camera without interruption. The scene depiction will normally comprise of a sequence of shots and a *cut* is the action of switching from one shot to another. Given the arrangement of characters and props in the scene, camera work can portray the action in a number of different ways by using particular types of shots and cut between them at different times. On a film set, it is director who chooses the events to film and the *cinematographer* who decides the style of filming and coordinates the camera operators.

Different types of shots have different functions. *Establishing shots* introduce the viewer to a scene by filming it from a distance and at an elevated height. *Character shots* film characters, for example one type of character shot, the *full shot*, depicts characters in their entirety and the *medium shot* frames characters from the thighs up. *Close-ups* are particularly important for dramatic emphasis and to film a character's reaction to the events occurring in the scene. These frame a character from the waist, or higher, up. The camera can move gradually closer to the action from establishing shots to close-ups such that the views become more subjective and the audience tends to identify more with the characters. As such, dramatic emphasis can be added to certain events and the emotional state of the characters portrayed. The greatest subjectivity is achieved with first-person perspective. This is when the camera simulates the viewpoint of one of the characters and so the viewer sees the scene through that character's eyes.

The previous paragraph discussed outlines the basics, but a cinematographer also has at her/his disposal an arsenal of camera tricks and movements which can be employed in order to achieve particular effects. These include *zooms* where a portion of the scene is gradually magnified;

tracking shots where the camera follows a parallel path to a moving character; *Dutch tilt* where a camera whose horizontal axis is not parallel to the ground is used to signify illness, confusion, or even approaching death on the part of the character; and even shaky hand-held cameras, often used to reflect chaotic events.

A single camera or multiple cameras might be used to film particular events [16]. Where the events are planned in advance and controllable, a single camera can be used: The set can be arranged and the actors can take position for each shot. The resulting footage will be edited to give the illusion of continuous events. Multiple cameras are usually used where the action is not controllable or is unplanned. A number of cameras film the events from a number of different angles and again, the resulting footage is edited. In both cases, the events are depicted from a number of different angles, providing varying levels of subjectivity, and it appears that the events are continuous whether or not they were filmed that way.

3. VIRTUAL CAMERAS IN GAMES

Recently released computer games generally depict the game play with only a subset of the options available from cinematography and with only one virtual camera. Since the action during game play is unplanned, cinematography dictates that multiple cameras should be used. With the synchronisation of multiple cameras, the view can switch between multiple camera angles thereby exploiting a fundamental feature of cinematography.

Two camera configurations used in a large number of games are:

- *First-Person Camera:* The camera adopts the viewpoint of the player's avatar and so the player sees the game world through her/his avatar's eyes as in *Quake III Arena* [11].

- *Over-The-Shoulder Camera:* The camera keeps a fixed distance above and behind the avatar so as to position it in the lower portion of the view such that the player can see their avatar's animations in addition to other parts of the scene. This view is used in *Tomb Raider III* [5].

Some games also use pre-scripted camera operations that are triggered by certain events but these scripts have to be written off-line [19]. We concern ourselves only with camera work that is improvised in real-time. Most computer games do not take the concept of controllable action into account, whereas we believe the ability to take control away from the user for short periods of time in order to provide camera work that advances the story line or highlights certain events is important in a single player game. One computer game that does control the action in favour of cinematic camera work is Prince of Persia: The Sands of Time [21].

Often when the term "cinematic" is used with relation to the camera work in games, this means that they contain *cut-scenes*. These are non-interactive pre-rendered scenes inserted at intervals throughout the game play where more elaborate camera work is used. However, our focus is on the camera work during the interactive game play.

A number of researchers have considered the issue of automatic camera work for virtual environments other than games such as interactive narratives [1] and interactive learning environments [2]. Nieuwenhuisen & Overmars [17] present a camera system that plans paths through a virtual environment. Christianson et al. [4] generate camera configurations off-line for pre-animated non-interactive scenes. He et al. present their Virtual Cinematographer for virtual party settings [10]. Other works consider real-time automatic camera work for games but with a single camera rather than multiple cameras [9, 20]. Our approach differs from these in that we concentrate on the use of multiple cameras to film the action in games and in particular, we advocate controlling the action to an extent to allow for more appropriate cinematic camera work.

4. CAMERA BOTS

The bots [12, 13, 14, 17] in many recently released 3D computer games can act as enemies or team members [14] to the player and their behaviour is complex enough for them to be convincing in these roles. Lately more resources have been granted to the AI (artificial intelligence) portion of games including more CPU cycles [22]. We argue that these resources can be harnessed to provide the camera work for the game.

Bots are required to navigate the complex geography of game worlds by using spatial reasoning, path planning and path following routines [7, 8]. Common sense reasoning allows them to make decisions relevant to the task at hand. These traits are all that is needed for a bot to position itself relative to some characters; to move towards and away from the locality of some events; to follow a character; and to frame a particular area of a setting, i.e. the requirements of a camera operator in cinema.

Multiple cameras are used by placing a number of *CameraBots*, i.e. bots acting as camera operators, in the game. The CameraBots do not have visible representations like the character bots in games. A number of different types can be made available. These can range from a simple bot which merely simulates first-person perspective, to sophisticated path-finding bots whose job it is to follow either the player or indeed some other participant around the game environment. The system then has the potential to present the action from the point of view of any of these bots and to switch between the multiple views on offer at appropriate moments.

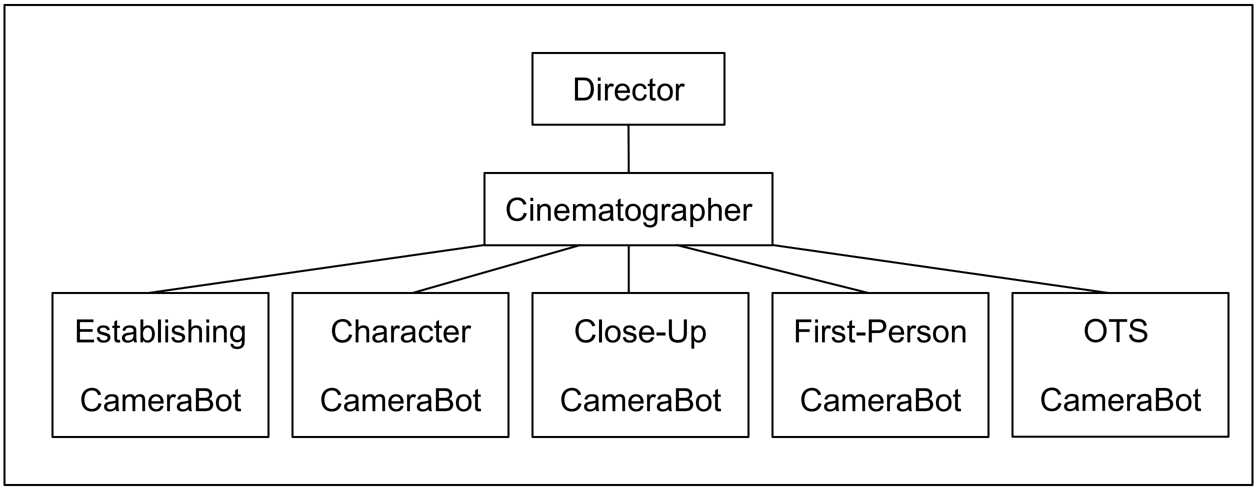

Figure 1: Virtual Cinematography Framework

Our CameraBots are part of an operational framework (figure 1) similar to that on a real-world film set. A *director* module decides what characters are to be filmed, a *cinematographer* module decides they are to be filmed, ensures that the necessary CameraBots are in the game world and provides them with instructions. There are a number of different CameraBots that may be selected and each specialises in providing certain views:

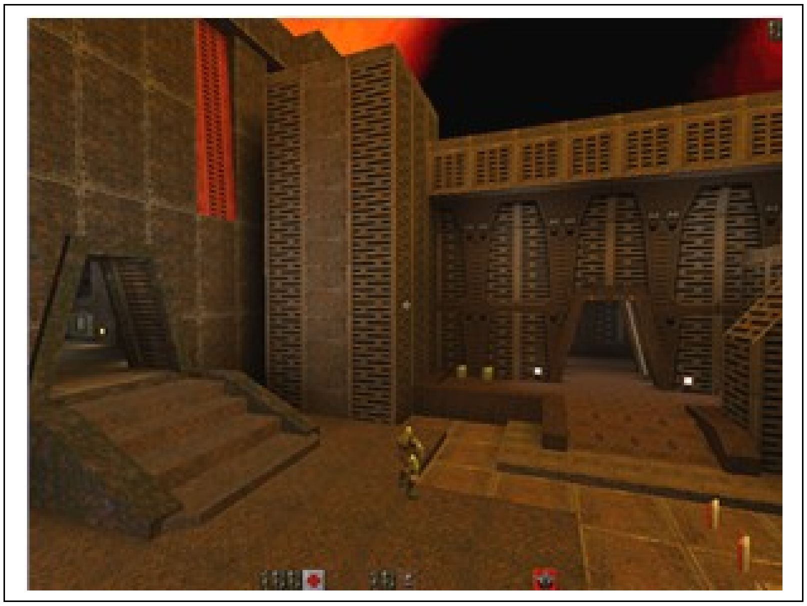

Figure 2: Establishing CameraBot view

- *Establishing CameraBot (figure 2)*: An Establishing CameraBot shoots establishing shots as mentioned in the Cinematography section.

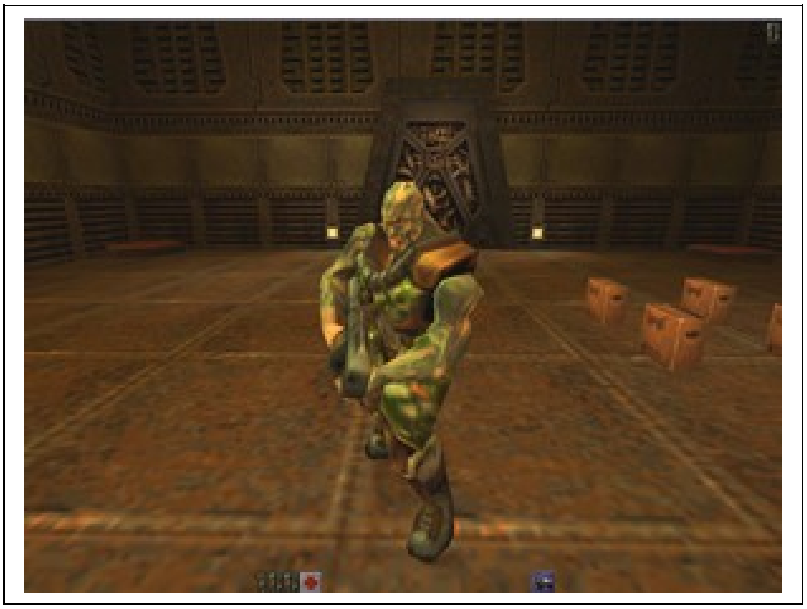

Figure 3: Character CameraBot view

- *Character CameraBot (figure 3):* A Character CameraBot shoots long shots and medium shots but not close-ups.

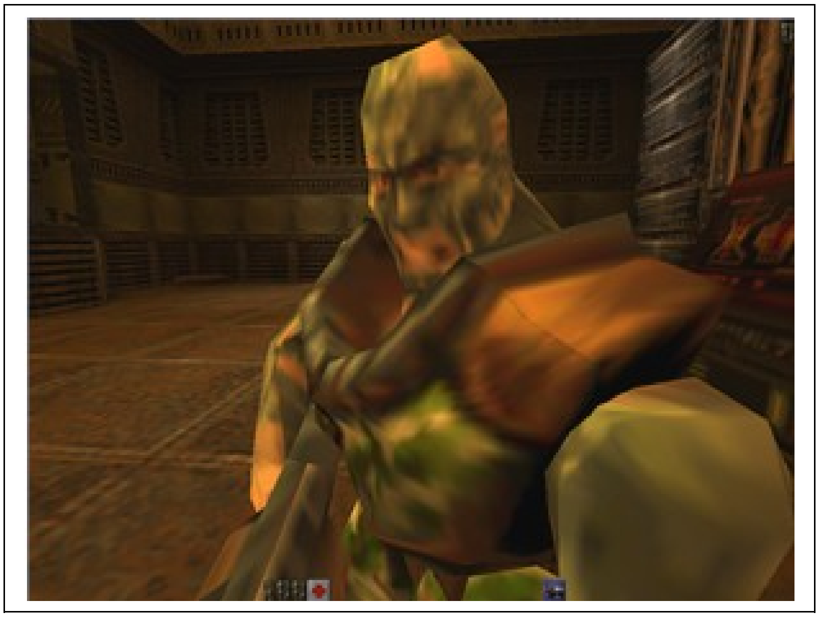

Figure 4: Close-Up CameraBot view

- *Close-Up CameraBot (figure 4):* As mentioned, close-ups have the special role of providing dramatic effects and, as such, there is a specialist CameraBot to film them.

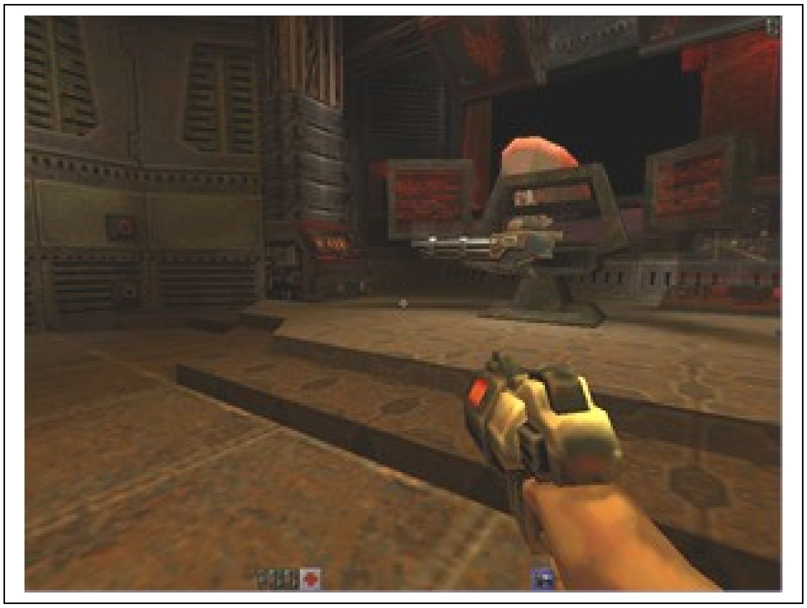

Figure 5: First-Person CameraBot view

- *First-Person CameraBot (figure 5):* The first-person CameraBot is used where accuracy is important.

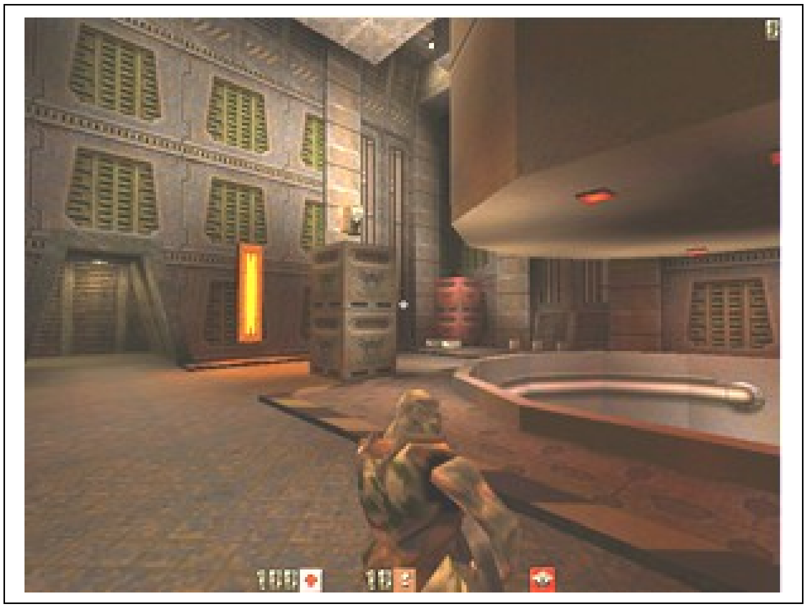

Figure 6: OTS CameraBot view

- *OTS CameraBot (figure 6):* An Over-The-Shoulder CameraBot is used for, among other things, navigation of the game world.

Simplified pseudo code for the operation of our virtual film crew for a 3D action game is as follows:

```
Director update (this occurs a number of times each second)

Evaluate scene
If action has changed enough
        Choose new characters/props and principal character to film
        Tell Cinematographer to film these
End if
```

```
Cinematographer update – given characters/props and principal character to film

If avatar is to be filmed and is a principal character
        If avatar is attacking
                Apply First-Person CameraBot to the avatar's view (for attacking accuracy)
        Else if avatar is moving (but not attacking)
                Apply OTS CameraBot to avatar (for navigation)
        Else (avatar is stationary and not attacking)
        Apply Establishing CameraBot to scene (for a general view of the scene)
    End if/else
Else (avatar is not a principal character)
    If principal character is attacking
        Apply Establishing CameraBot to scene (for a general view of their attack)
    Else if principal character is moving (but not attacking)
        Apply Character CameraBot in tracking mode to principal character
    Else (principal character is stationary and not attacking)
        Apply Close-Up CameraBot to principal character
End if/else
```

Our modular approach means that new types of CameraBots can be added to our virtual cinematography system with ease. Real-time editing of the footage provided by the bots is performed by the cinematographer. What the player sees is a number of static and moving views from different angles suited to the events occurring.

CONCLUSION

The use of multiple camera operators in the form of CameraBots provides for a more engaging gaming experience with less user disorientation arising from inappropriate camera work. By using the AI portion of a game engine, i.e. the bot program code, resources and code are reused. Our virtual cinematography system can be applied to both the player's view and to spectator views allowing others to watch a more engaging depiction of the game play. By making it possible to control the action in the player's view in a single player game we can employ more useful cinematic camera work.

Presently, we are implementing our virtual cinematography system as a modification to the Quake II game engine in C++. The code is written in a modular way and communicates with the game engine through an API. It can be ported to other game engines by providing the implementation for this API. In the future we plan to add new types of CameraBots to our framework and increase the functionality of the existing ones to allow for a wider range of cinematic camera work. We will have a number of people test our implementation from the point of view of playability and visual immersion.

REFERENCES

1. Amerson, D. & Kime, S. (2001). Real-Time Cinematic Camera Control for Interactive Narratives. In The Working Notes of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment, Stanford, CA.
2. Bares, W. H., Zettlemoyer, L. S., Rodriguez, D. W. & Lester, J. C. (1997). Task-sensitive cinematography interfaces for interactive 3D learning environments. In proceedings of the 3rd international conference on Intelligent user interfaces, pp. 81 – 88.
3. Brown, B. (2002). Cinematography: Image Making for Cinematographers, Directors and Videographers. Oxford: Focal.

4. Christianson, D. B., Anderson, S. E., He, L., Salesin, D. H., Weld, D. S. & Cohen, M. F. (1996). Declarative Camera Control for Automatic Cinematography. In Proceedings of the Thirteenth National Conference on Artificial Intelligence, 148-155.
5. Eidos Interactive. (1998). Tomb Raider III, computer game for IBM compatible PCs, released by Eidos Interactive, San Francisco, CA.
6. Fairclough, C., Fagan, M., Mac Namee, B. and Cunningham, P. (2001). Research Directions for AI in Computer Games. Proceedings of the Twelfth Irish Conference on Artificial Intelligence and Cognitive Science pp. 333 – 344.
7. Graham, R., McCabe, H. & Sheridan, S. (2003) Pathfinding in Computer Games. ITB Journal Issue Number 8, December 2003.
8. Graham, R., McCabe, H. & Sheridan, S. (2004) Neural Networks for Real-time Pathfinding in Computer Games. ITB Journal Issue Number 9, December 2004.
9. Halper, N., Helbing, R. & Strothotte, T. (2001). A Camera Engine for Computer Games: Managing the Trade-Off Between Constraint Satisfaction and Frame Coherence. In Proceedings of Eurographics, 174–183.
10. He, L., Cohen, M. F. & Salesin, D. H. (1996). The Virtual Cinematographer: A Paradigm for Real-time Camera Control and Directing. In Proceedings of SIGGRAPH 1996.
11. id Software. (1999). Quake III Arena computer game for IBM compatible PCs, released by id Software, Mesquite, TX.
12. Laird, J. E. (2000). It Knows What You're Going To Do: Adding anticipation to a QuakeBot. AAAI 2000 Spring Symposium on Artificial Intelligence and Interactive Entertainment. AAAI Technical Report SS00–02. Menlo Park, CA: AAAI Press.
13. Laird, J.E. and Duchi, J.C. (2000). Creating Human-Like Synthetic Characters with Multiple Skill Levels: A Case Study Using the Soar Quakebot. AAAI tech. report, SS-00-03, AAAI Press, Menlo Park, Calif.,
14. Laird, J. and van Lent, M. (2001). Human Level AI's killer application: Interactive computer games. AI Magazine, Volume 22, Issue 2, Summer.
15. Marino, P. (2004). 3D Game-Based Filmmaking: The Art of Machinima. Paraglyph.
16. Mascelli, J. V. (1965). The Five C's of Cinematography. Los Angeles: Silman-James Press.
17. Nieuwenhuisen, D. & Overmars, M. H. (2003). Motion Planning for Camera Movements in Virtual Environments.
18. Reynolds, C. (1999). Steering Behaviors For Autonomous Characters. Game Developers Conference 1999.
19. Sánchez-Crespo Dalmau, D. (2004). Core Techniques and Algorithms in Game Programming. Indianapolis, IN : New Riders Publishing.
20. Ting-Chieh, L., Zen-Chung, S. & Yu-Ting, T. (2004). Cinematic Camera Control in 3D Computer Games. In Proceedings WSCG 2004.
21. Ubisoft (2003) Prince of Persia: The Sands of Time, computer game for IBM compatible PCs, released by Ubisoft, Austin, TX.
22. Woodcock, S. Game AI: The State of the Industry. Game Developer Magazine August 2000.