SPATIAL CINEMATIC MEDIATION IN REAL-TIME ARCHITECTURAL WALKTHROUGHS

SUBMITTED: July 2005
REVISED: November 2005
PUBLISHED: May 2006 at http://itcon.org/2006/25/
EDITOR: J.M. Kamara

Carlos Calderón
School of Architecture, Planning and Landscape, University of Newcastle, UK
email: carlos.calderon@newcastle.ac.uk

Nicholas Worley
Faulkner Browns Architects, Newcastle, UK
email: n.worley@faulknerbrowns.co.uk

Karl Nyman
Northfield, New Hampshire, USA
email:chipartist@metrocast.net

SUMMARY*: This paper presents a methodological framework for the utilisation of our cinematic camera control engine in the exploration and representation of architectural designs. In this paper, we report on a fully developed modification (mod) of a real-time engine (Unreal TM): a cinematic control camera engine designed to enhance a real-time navigatable experience architectural contexts. Basically, the mod enables the use of cinematographic techniques (Tracking, Cutaway, Exposure, etc) to explore architectural designs in a 3D real-time development and testing environment (the Unreal Engine). The purpose of our investigation is to use this "spatial cinematic mediation" to improve the presentation of architectural designs by facilitating a reading of architecture in a design sense. Additionally, the mod might be use by the designer to notice design issues which otherwise might not be noticeable. Currently, there is neither software nor a structured approach which facilitates this in architectural visualisations.*

KEYWORDS: *virtual environments, navigation, camera engine, cinematography, experiential design.*

1. INTRODUCTION

Representation in architecture is the expression of a concept by visual means. Creating and visualising are indissoluble steps in the process of designing. Representation, therefore and as highlighted by design scholars (Madrazo, 2000), should not been seen as just visualisation techniques, but should be considered as a conceptual design paradigm with which the idea is created. However, identifying which are the new representation possibilities of the computer, and making an intelligent and creative use of them, is not straightforward. In fact, it is the whole heart of the matter of using computers to design in architecture.

1.1 Representation with interactive media

Since 3D technology and time-based media have emerged in architecture, representation techniques now have more options. As architectural design becomes more complex and dynamic, new forms of representation are required to explore and represent architectural designs. Traditional forms of representation techniques such as photos, drawings, and animations are, for instance, limited in conveying knowledge about how architectural structures are made, how they work, how they influence people’s perception of space and so forth. It is also widely accepted that real-time visualisation (i.e. games engines) can enhance the design in various stages as well as better present concepts because a virtual environment reduces the mental effort required to comprehend a 3D world by assigning relevant content at specific design stages and provides possibilities for user directed exploration. However, besides 3D real-time computer environments, recent developments in interactive navigation modelled environments suggest significant ease, user involvement, and effectiveness in representation can be gained in relation to other conventional modes of analysis (Saleh Uddin, 2001). Moreover, representation related to sensory perception can only take place through interactive media and motion related representation. Hence, in our case, we are investigating the development of software systems (i.e. cinematic camera control engine) and methodological frameworks (i.e. structured approach) which enable the integration of new modes of interactive navigation of 3D real-time environments in the exploration and representation of architectural designs.

1.2 Cinematic control camera engine

Most of the applications which use interactive 3D graphics as a platform are portrayed from a first person point of view; a particular character's point of view or from a free roaming mode. However, these camera modes are not always appropriate to facilitate the exploration and reading of architecture in a design sense. For example, to experience rhythm in architecture is to observe "variations on a theme within a rectilinear pattern" (Rasmussen, 1962) and, thereby, when you feel that a line is rhythmic means that by following with your eyes you have an experience that can be compared with the experience of rhythmic dancing. We are accustomed to perceiving architectural rhythm at a human perspective, that is, at walking pace and eye-level height. We believe that within a VE (virtual environment) the limitation of having to navigate from "standard" camera modes (first person point of view; a particular character's point of view or from a free roaming mode) makes architectural concepts, such as rhythm, more difficult to comprehend and communicate (see Fig. 1). A VE allows us to perform actions that are not possible as part of real-life. We argue that, with this advantage, experiencing architecture is not limited to a first-person perspective – we are able to view ourselves in relation to our surroundings. In fact, similar types of "communication" problems have been faced by cinematographers for over a century. Over the years, filmmakers have developed a set of rules and conventions that allow actions to be communicated comprehensibly and effectively. This paper addresses the problem of exploring architectural designs in 3D real-time virtual environments by proposing a camera mode which incorporates cinematic principles. It is important, however, that if the user wishes he should be able to control the camera from a first person perspective and a free roaming perspective too – thus ridding himself of the cinematic aspect and intended viewpoints of the camera. We have therefore established 3 modes of camera use which can be freely selected by simply typing a console command at any time during the walkthrough: a) architectural mode (see Fig 1); b) first person point of view and c) free roaming mode in which the user is granted the ability to fly and go through any geometry (no collision detection). It is the architectural mode which is the subject of this paper since the other two are already implemented in the real time engine used: Unreal TM (UnrealEngine2, 2004).

FIG. 1: Standard modes: first (a) and third person (b). Architectural camera mode(c) using cinematographic techniques: a tracking shot.

1.3 Integrating computers in architectural design

For an architect, designing with a computer means to work creatively within the framework of a given methodology. It is through the method that a successful dialogue between designer and machine can be achieved (Madrazo, 2000). In this paper we proposed a methodological framework for the utilisation of our cinematic camera control engine in the exploration and representation of architectural designs. This framework relies on the designer using the notion of partnering architectural concepts to cinematographic techniques to create a linear narrative structure of the story of their designs as their spatial cinematic mediation layer. This layer provides a mechanism for structuring the VE spatial content in a way which facilitates a reading of architecture in the design sense. Additionally, the cinematic camera control engine can be used as a device for focusing awareness on a design component that illustrates or embodies the concept (i.e. a colonnade embodies rhythm) to enable the designer to notice design issues. Both uses are illustrated in the example section using a real case scenario.

The paper is structured as follows: in the first section we introduce the principles and related work with regards to existing paradigms for the development of cinematic camera control engines and the use of digital environments to incorporate filmmaking techniques in architectural design. In the next section, within the experiential design framework, we explain the rationale behind our spatial cinematic mediation framework. We then describe how the system works and show how the proposed "spatial cinematic mediation" is utilised by a designer in the design of a visiting centre for The Portland Sculpture and Quarry Trust. We conclude the paper with some conclusions and future directions.

2. RELATED WORK

There are number of adjacent areas in which related work has been explored. Recent advancements in computer science have explored paradigms for automatically generating complete camera specifications in virtual 3D environments. Karp and Feiner (Karp and Feiner, 1990) developed an animation-planning system that can customise computer-generated presentations for a particular viewer or situation. Christianson et al. (Christianson et al, 1996) presented an interactive system which plans a camera sequence based on a simulated 3D animation script. All these techniques use an off-line planning approach to select the sequence of camera positions. Off-line planning techniques take a pre-existing animation path and calculate the best camera placements. In this investigation, by contrast, we are concerned with real-time camera placement as the interactively controlled action proceeds. That is, systems which concentrate on finding the best camera placement when interactive tasks are performed. These type of systems were pioneered by Drucker and Zeltzer. (Drucker and Zeltzer, 1995). A system for implementing intelligent camera (Drucker and Zeltzer, 1995), they show how to set up optimal camera positions for individual shots. In our case, we are not only interested in real-time camera control system but also in incorporating cinematographic expertise for camera placement. He et al (He et al, 1996) were the first ones to present a paradigm for automatic real-time camera control which incorporated cinematic knowledge which be used across application domains. We have used their implementation as the starting point for our system.

In the architectural realm initial steps have been taken to investigate ways in which filmmaking can be used for the development of *off-line* architectural animations (Temkin, 2003) (Alvarado and Castillo, 2003). They argue that if we are to evolve beyond the average fly-through animation new ways of seeing and composing in time, which can be used to inform the process of architectural design, ought to be developed (Alvarado and Castillo, 2003). More related to our research is the work carried out at Cambridge by Nitsche et al (Nitsche et al, 2003), their main objective was to use drama concepts to develop flexible interactive techniques that supply VE's with a coherent context and make the resulting "virtual place" available to the user in a *dramatically* engaging way. Therefore, they have used the concept of drama as the driving notion in the development of their techniques for the structuring of the content of a VE. This, therefore, gives a particular (dramatic) reading of the architecture. For instance, when the user navigates through the "Dining Hall colonnade" their objective is to engage the user in dramatic context of the space by making her remember that, i.e., 'this was the colonnade where the quarrel happened'. We do not think that this is necessarily a reading of architecture in a design sense because it is difficult to say what event might happen within a particular part of a building so, to stage a quarrel or something only gives one instance of an event occurring within that space. What we are doing is looking at the development of software systems (i.e. cinematic camera control engine) and methodological frameworks (i.e. structured approach) which enable structuring the VE spatial content in a way which facilitates a reading of architecture in the design sense. Additionally, these techniques can be used as a device for focusing awareness on a design component that illustrates or embodies the concept (i.e. a colonnade embodies rhythm) to enable the designer to notice design issues. Hence, we have used Nitsche et al's (Nitsche et al, 2003) work as a reference for our spatial cinematic framework and not as a reference for our system's development.

3. SPATIAL CINEMATIC FRAMEWORK

Our framework draws on the practice-based research structure proposed at Cambridge in which they argue that VEs lack a "natural" point of view (POV) and that mediation layers separate the user from the synthetic 3D space. Furthermore, without these mediation layers or devices a VE does not emerge from its mathematical abstraction. Nitsche et al (Nitsche et al, 2003) distinguish between three equally important constellations of mediation devices that each delivers spatial content differently. Every constellation has a distinct balance between freedom of interaction and authorial control and combining provides complementary expressive resources. The constellations are: user driven, spatially dependent and author defined.

The main mediation devices of the user driven constellation are the (virtual) cameras, fixed or mobile relative to the user avatar, and always under some degree of user control. Avatar movement can also be explicitly controlled by the user, with an input-device such as keyboard or mouse. The approach is typical of 1st person or 3rd POV camera used in games and other interactive titles. This user-driven camera strategy is predominantly concerned with the description of the avatar and its action rather than with facilitating the reading of architecture or enabling the designer to notice design issues. As stated previously, in our framework, the 1st person or 3rd person POV can be combined with our camera mode.

Within the spatially dependent constellation, the medication devices in the VE designers' arsenal are akin to those spatial elements which aid in forming the mental image of the user. For example, regions in the VE can be imbued with content to reinforce significant changes in spatio-functional patterns and add meaning to the underlying spatial topology. An example of this would the use of Norberg-Schulz's phenomelogical theories (Norberg-Schulz, 1980) in architecture to aid orientation and navigation in VEs (Charitos, 1998). Therefore, in principle, similar principles could be followed to model the virtual environment to facilitate its architectural reading. An example of this can be found in Charitos' PhD thesis (Charitos, 1998) where the author tried to establish the architectural principles of designing space in virtual environments.

Finally, in the author defined constellation, users directly control avatar movement but not the camera and the medication devices are activated by spatial, mobile and conditional triggers. These devices are specifically designed to allow the author to add "extra" meaning to the virtual space. Conceptualised and implemented as an integral part of the exploratory experience, these can interpret and shape the interactive flow. The user driven and spatially defined mediation devices are outside the scope of this paper. Our cinematic camera control engine is intended as a mediation device for the author driven constellation. This device relies on the designer using the notion of partnering architectural concepts to cinematographic techniques to create a linear narrative structure of the story of their designs as their spatial cinematic mediation layer. This layer, in turn, provides a mechanism for structuring the VE spatial content in a way which facilitates a reading of architecture in the design sense.

3.1 Linear narrative structure

Narrative theory was introduced by Brenda Laurel (Laurel, 1993) to the field of computer human interfaces, with "computers as theatre". Although widely quoted, it has not been widely applied, except for the entertainment industry. Narrative theory's principles draw on the use of human ability to create and manipulate symbolic representations is the central feature of human intelligence and imagination. In our approach, we are using a linear narrative model, whose principles were laid by Laurel (Laurel, 1993) as a preliminary vehicle for structuring the spatial content with the aim of allowing an easier reading of the architectural space. Essentially, this model follows the basic rule established through good practice in narrative productions which says that good productions have a certain build-up. It starts with an introduction, with a low tension (see Fig. 2). This tension rises to a climax, which usually coincides with the main features of the design in our case, just before the end and then declines quickly.

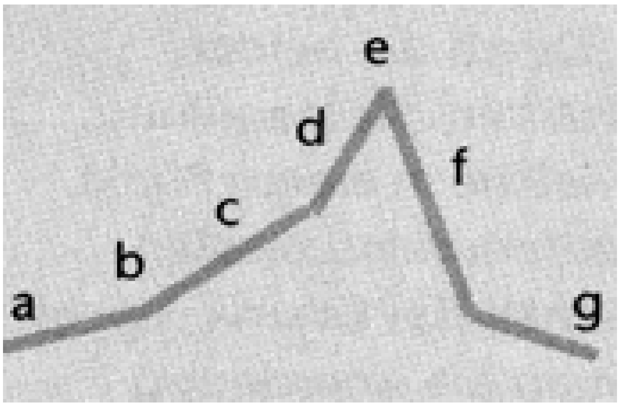

FIG. 2: Linear narrative model.

Finally, it must be noticed that in conventional cinematographic or narrative productions the architecture is merely a setting for a plot to unfold. The focus is on the unfolding events, the architecture is a secondary backdrop. But in our approach, the architecture is the event: the actions of the avatar and the cinematic moves are aimed at "unfolding" the building itself.

3.2 Encoding cinematography

As explained in the previous section, various architectural scholars (Temkin, 2003) (Alvarado and Castillo, 2003) have already identified the potential of film theory in the early stages of architectural design. From the perspective of designing a new camera mode, there is, however, a more important aspect about cinematography which other possible set of conventions (i.e. viewing modes in computer games) lack: the existence of grammars and languages (i.e. (Arijon, 1976)) which have been translated into a (film) vocabulary and a series of well known (cinematographic) techniques. Hence, existing collections of cinematographic conventions provide an initial path to map low level specifications for the camera placements to high level construction of narratives. However, any attempt to automate cinematography, in our case the creation of a new camera mode, faces a difficulty not faced by real-world filmmaking or storytelling: a description of the rules of cinematography which is explicit enough to be directly encoded as a formal language (He et al, 1996).

In our case, we have solved this problem by creating goal-oriented programmes (scripts) which enable the recreation of well known camera shots by simple assigning values to certain variables in the programme. These programmes provide camera movement along all Cartesian world-space axes (see right top corner in Fig. 3), plus an Avatar-centred rotational capability. This rotational capability includes manipulations on all Euler angles (see right top corner in Fig. 3) except roll which corresponds to the viewpoint's line-of-sight axis or the traditional camera's optical axis. However, some "special effects" like camera shake do require a roll component. Due to the limitations in the graphical engine, those effects are achieved by assigning values rather than modifying the specific "roll" function which is handled natively (i.e. C++ code) by the graphics engine.

For instance, imagine that we want to recreate a tracking shot: a tracking shot sets the camera along a perpendicular from the line of interest and then moves with the actor maintaining the same orientation (see Fig. 3). In our first implementation, the tracking shot module allowed us to modify parameters such as the speed at which the camera moves away from the actor (CamDistAdjust), the maximum distance that the camera can reach (CamDistAdjust), the speed at which the camera rotates to reach its perpendicular vector (CamRotFactor), and the direction of rotation (CamRotFlag) (see Fig.3).

In similar fashion and using the appropriate module, the following descriptions of camera shots are also recreated in the example section: establishing and releasing shot, canted framing, angle of framing and frame cropping. The first four camera shots also demonstrate the generalisation of the approach through all six degrees-of-freedom (DOF). Table 1 provides a more detailed explanation of the movements covered by those four shots and appendix A for a comprehensive description of all cinematic techniques implemented in our system.

In this section we have shown that our system is recreating a representative subset of camera shots. These can then be combined to create walkthroughs with a narrative structure, following Brenda's model, as we described in the example section.

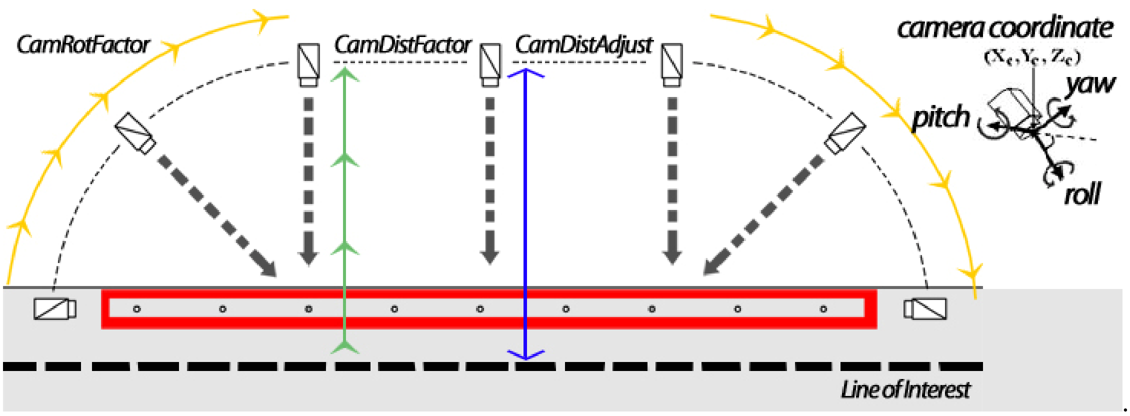

FIG. 3: Camera path in a tracking shot; variables (CamRotFactor, CamDistFactor, CamDistAdjust) to recreate a camera shot and camera DOF.

TABLE 1: List of camera shots and camera movements available in the Architectural Cinematographer. A complete description of camera shots can be found in (Yale, 2004)

Camera shot	Camera movements
Tracking shot	Transitional movement occurs either solely in the x or y or z axis while the camera yaw is altered.
Canted framing	Canted framing is a shot where the camera is positioned off centre and at an oblique angle. The camera utilises all three camera rotations: pitch, yaw and roll and a translational movement.
Establishing/Releasing shot	This shot is used to introduce the locale for a scene. It is achieved by a combination of translational movement accompanied by modification of the pitch of the camera.
Cutaway shot	This shot can be a stationary shot which is cut to, remains stationary, and cuts away to a different shot. There is no movement in any axis and no rotation involved

3.3 Architectural concepts

We shall take it, for our purposes here, that the form of a building is its internal physical structure, as described under some appropriate conceptualization. Many aspects of internal physical structure might be considered and described, but the conceptualization always describes the scope of our interest.

Alexander et al (Alexander et al, 1977) provide us with a pattern language which is extremely practical in its nature. For instance, it can be used for the generation (construction) of architectural elements (i.e. a porch) by combining different patterns. This, in turn, would create a language of, for example, a porch. Whilst Alexander's language is extremely useful to describe buildings from a technological or even functional standpoint, it is not particularly well suited for the conceptualization of buildings from an experiential point of view. Wilenski (Wilenski, 1927.) insisted that an architect's "business as artist" was with "the definition, organization and completion of his formal experience by creating a concrete object". He went on to propose that "the architect experiences, synthesizes, and creates; he experiences proportion, balance, line, recession and so on, he coordinates and organizes his experience, and he gives it definite form in a building… He is concerned from first to last with problems of formal relations". We felt, therefore, that experiential issues are more closely related to aesthetics than to technology and opted for selecting Rasmussen (Rasmussen,1962) conceptualization of architecture because, as he put it, "art should not be explained; it must be experienced". Rasmussen description of architectural concepts is an attempt to, by means of words, help others to experience architecture which is precisely our objective. The architectural concepts used in the example section consist of rhythm (see Fig. 4), proportion, scale, verticality and composition. This set of architectural concepts provides a representative sample of Rasmussen's description and a way of testing the generalisation of cinematographic techniques.

FIG. 4: Museum of Roman Civilisation, rhythmic colonnade.

4. ARCHITECTURAL CINEMATOGRAPHER

The Architectural Cinematographer's purpose is to enhance architectural walkthroughs created using UT2004 game technology with interactive camera effects. The enhanced features are implemented by a series of custom-coded actors (the goal-oriented programmes described in the previous section) that can be placed in custom designed levels (aka "maps") made in the Unreal Editor (UEd), UT2004 version. The effects are triggered by the interactive participant (the Viewer) entering a designer-specified area of the interactive environment, or "map".

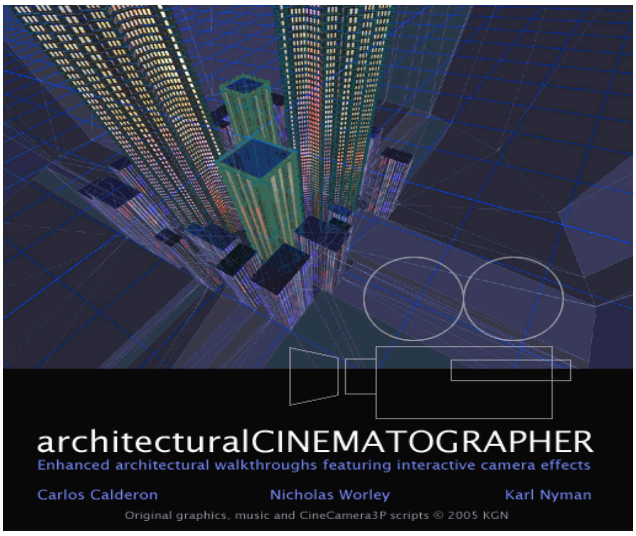

FIG. 5: The splash screen when installing the architectural cinematographer as a modification to Unreal Tournament 2004 ^{TM}.

The Architectural Cinematographer uses the UT2004 mod architecture system to create a separate directory structure for the files the mod uses. Directories for the files are automatically created by the Umod installer when the mod is installed. That is, by double-clicking on the Architectural Cinematographer software package (see Fig. 5), this self-installs seamlessly into the Unreal Engine benefiting from all the graphical user interfaces available through the main game to access designed levels, customise settings, change avatar, and so on.

The default viewer's avatar (Joe College) used in Architectural Cinematographer has been designed to more closely imitate a normal human (compared to the boosted combat lifeforms of UT2004, at least), and thus is much more limited in his movement capabilities, i.e., he has been designed to walk as the default movement mode.

4.1 System

Our system is “on-line system” camera system in which cinematographic techniques are encapsulated in modules (custom-coded actors: scripts) and related to the architectural concepts by an event-model. In other words, the system generates camera placements in real-time as the interactively controlled action proceeds according to the cinematographic technique encoded in the modules and, taking advantage of the event-model embedded in the graphical engine, these modules are assigned to specific architectural concepts of the design using volumes (a mechanism specific of Unreal ^{TM}). For instance, let us assume the modeller/designer wants to show the rhythm embedded in the colonnade using a tracking shot technique. He/she creates a “volume”, where he/she wants to create the effect –i.e. inside the colonnade- and links it to the tracking shot custom coded actor via the event-model. That is, links the tracking shot script’s (event) tag to the colonnade’s volume. Fig. 6 illustrates this.

Consequently, when running the environment and the user approaches the colonnade, upon entering, an event is triggered in the VE which is recognised by the system and the camera module (tracking shot) takes over. Fig. 7 demonstrates this graphically from the user’s standpoint. As the user approaches position 1 in Fig. 7 the act of the user entering the volume (the wire frame box in Fig. 7) is recognised by the system as an event. Window (a) in Fig. 7 depicts the user’s approaching the colonnade from a “standard” camera view: a third person point of view. Once the event has been identified, the camera moves into the tracking position shot (the line in Fig. 7 indicates the camera path and window (b) what the user sees once the camera is in position) whilst the user remains within the volume boundaries (see position 2 in Fig. 7). Finally, the user’s action of leaving the volume (see position3 in Fig.

7) is recognised by the system as an event and the system, in turn, responds by returning the camera to a third person point of view (window (c)).

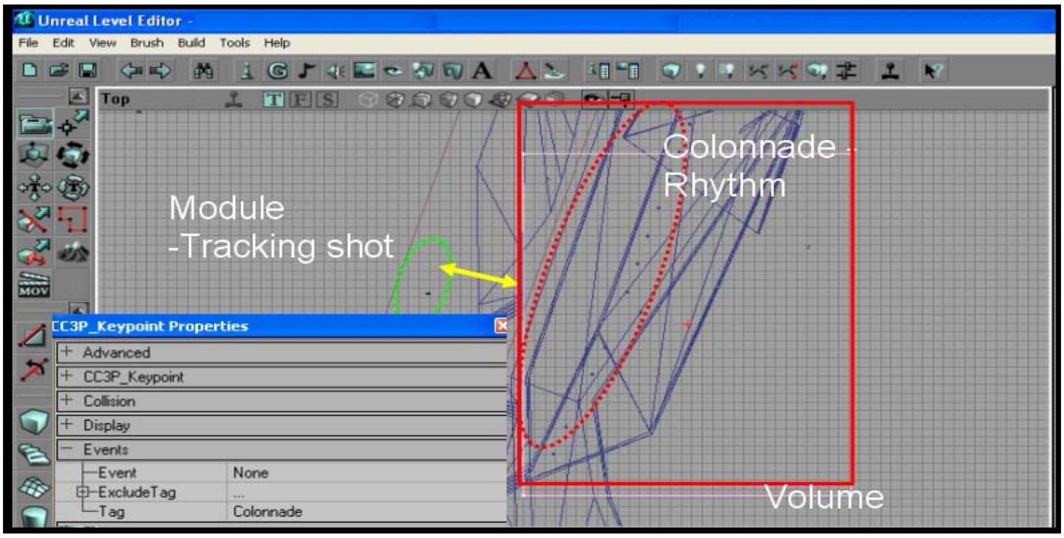

FIG. 6: The splash screen when installing the architectural cinematographer as a modification to Unreal Tournament 2004 TM.

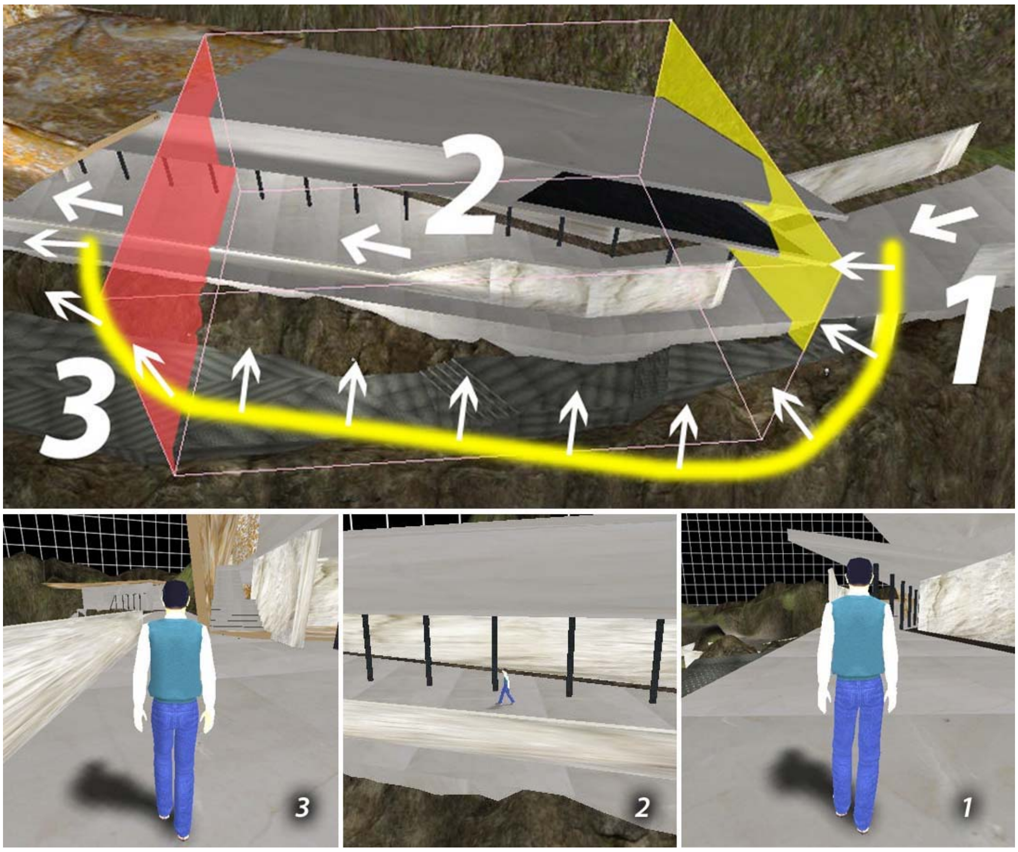

FIG. 7: User approaches the colonnade, upon entering an event is triggered in the VE which is recognised by the system and the camera module takes over.

More than one custom-coded actor can be associated with a certain Volume, so multiple effects can be triggered simultaneously and will operate concurrently. There are, of course, limits to the usefulness of some combinations. Fig. A1 (see Appendix A) shows the custom-coded actors derive from keypoint. These actors are available through the level design editor (UnrealEd) and, by dragging and dropping, they can be placed in the map at the user's will.

A complete description of all implemented actors can be found in appendix A.

4.2 Example

In the previous sections, we have proposed a methodological framework for the utilisation of our Architectural Cinematographer in the exploration and representation of architectural designs. As explained, this framework relies on the designer using the notion of partnering architectural concepts to cinematographic techniques to create a linear narrative structure of the story of their designs as their spatial cinematic mediation layer. This layer provides a mechanism for structuring the VE spatial content to facilitate a reading of architecture in the design sense. Additionally, the Architectural Cinematographer can be used as a device for focusing awareness on a design component that illustrates or embodies the concept (i.e. a colonnade embodies rhythm) to enable the designer to notice design issues. In this section, we report on the use of the Architectural Cinematographer and the proposed framework by the designer using a visiting centre set up by the Portland Sculpture and Quarry Trust as illustration.

4.2.1 Presentation: a linear narrative structure

The narrative model used in the example follows the principles laid by Brenda (see Fig. 8. narrative stages). Point (a) on the narrative scale introduces the first architectural concept (composition) through the use of the 'establishing shot' (cinematographic technique). The user remains relatively calm while progressing through the environment and gets a sense of the composition of the architecture at hand. As he nears the built form (b), his experience heightens as he is introduced to more concepts at more of a regular pace. As he reaches (c), (d), and (e) he experiences rhythm, scale, and proportion – this area is where the experience of the environment is most enveloping. Reaching points (f) and towards (g) the user is given a slow release from the environment, finally to look back over the architecture once again, taking in the architectural composition from another perspective. The correspondence between different narrative stages; architectural concepts and cinematographic techniques used is shown in Fig. 8. Fig. A2 (see Appendix B) shows a series of film strips detailing each architectural concept with its partnered cinematographic technique as it used in the architectural walkthrough. The sequential nature of the camera positions exists as a vehicle for providing the structured, experiential narrative. Deviating from this 'path' finds the user out-of-narrative but able to experience the environment, and thus the architecture, along an undetermined path. A more detailed description of the designer's reasoning behind the partnering architectural concepts and cinematographic techniques can be found in appendix B.

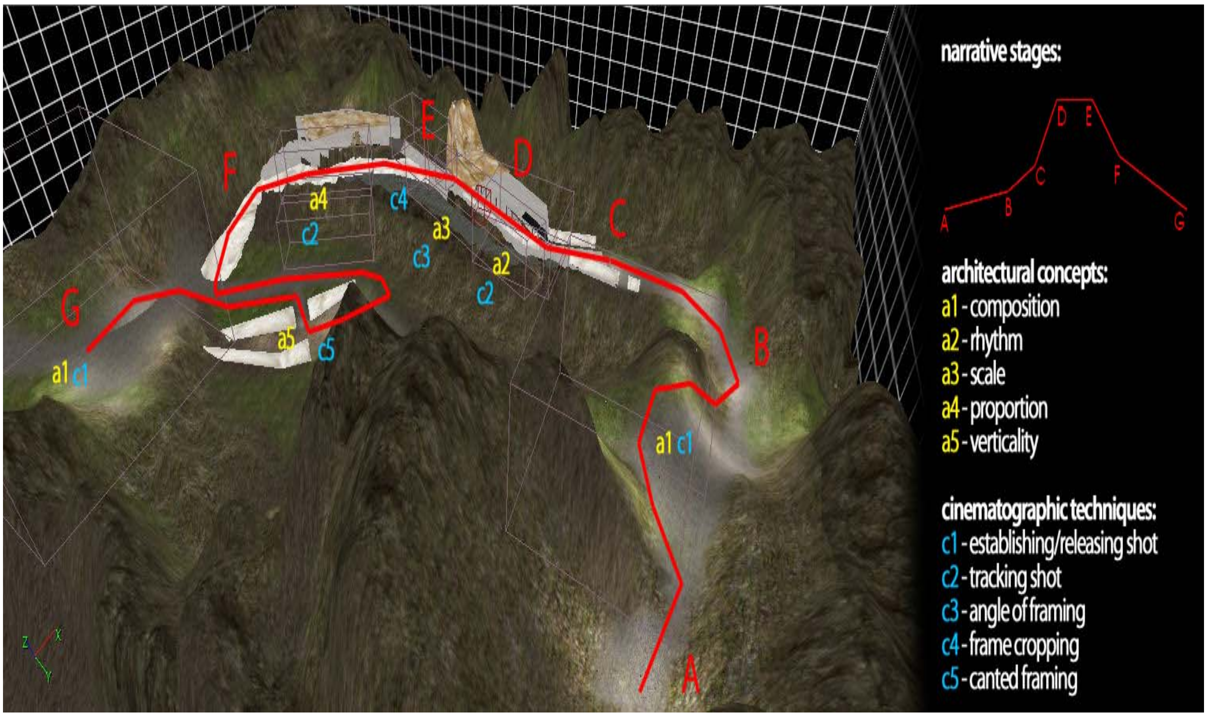

FIG. 8: Correspondence between different narrative stages; architectural concepts and cinematographic techniques.

4.2.2 Design issues

The diagrams below detail, after preliminary testing, the design issues that were noticed in the example virtual environment when using the different cinematographic techniques. No modifications have been made to the design at this stage due to lack of time allocated to the project, but each issue details improvements that may be made to rectify the issue raised.

Design issued recognised: ramp gradient. The tracking shot in this scene prompted the designer to re-evaluate the gradient of the ramp. The designer felt, after observing the avatar walk up the ramp, that the step between the ramp and the building is too large. People would find it difficult to relate to the building as they do not come into direct contact with it, rather they come into contact with a wall of raised ground (see Fig. 9). It may be that the design solution is to reduce the gradient of the ramp so that the ground remains at the same level as the building at this location and the loss in height that the ramp suffers is made up at some other point further down the ramp itself.

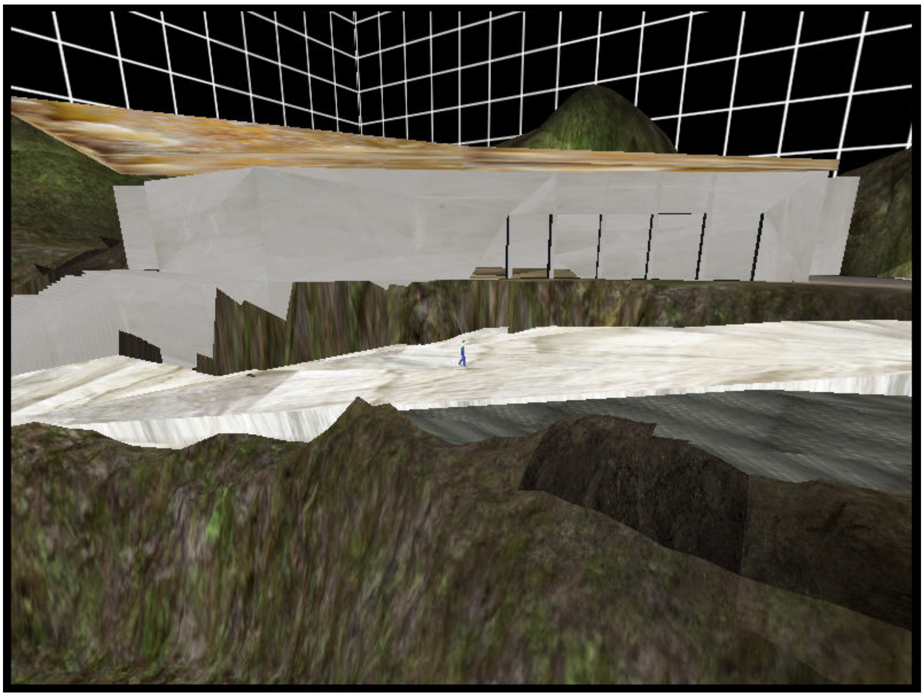

FIG. 9: The tracking shot in this scene prompted the designer to re-evaluate the gradient of the ramp.

Design issue recognised: tower scale. Using ‘angle of framing’ the designer had the intention to show the scale of the tower in relation to the user. The designer found however that the tower dwarfs the user – the difference in scale of the character to the tower was too large and the designer felt that a person would find it difficult to relate to such a high structure when standing next to it (see Fig. 10). The tower acts as the circulation pivot point for the whole scheme so the designer felt that it was important that the user was able to relate to the tower at first hand.

A possible design solution would be to bring the height of the tower down or to animate the tower at ground level – opening up the façade at a lower level would reveal the accommodation within the tower and would provide a more human scale to that portion of the scheme.

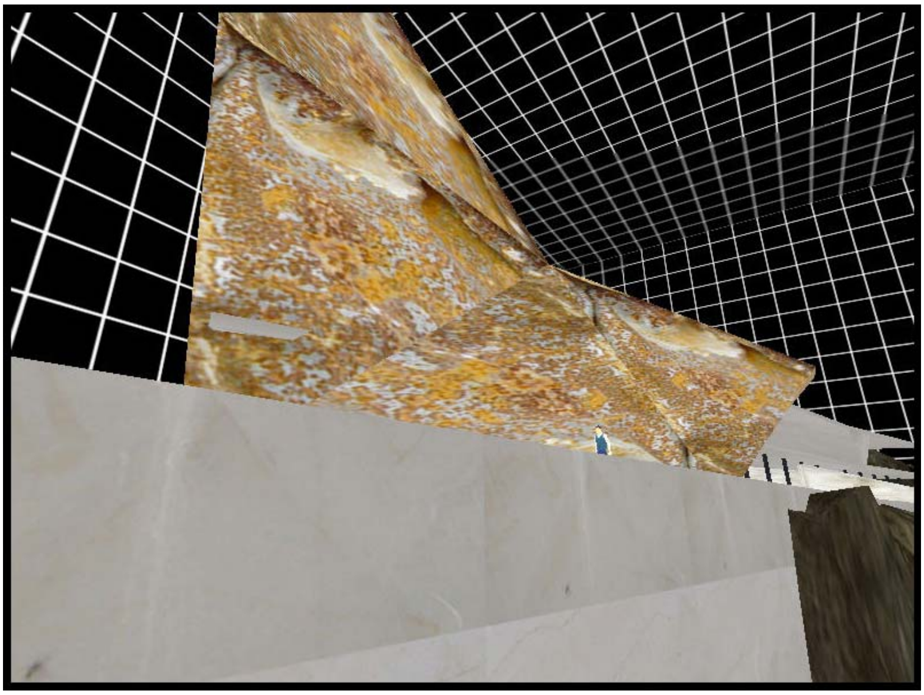

FIG. 10: Using 'angle of framing' the designer found that the scale of the tower could be improved.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have presented: a fully implemented cinematic camera control engine which is seamlessly integrated into the Unreal Engine and a structured approach for the use of the architectural cinematographer in architectural visualisations. Both the system and the approach were illustrated using a visiting centre for The Portland Sculpture and Quarry Trust as a case study.

We could be criticise for introducing third person cinematic techniques which might imply that something is going to "happen", but then nothing actually "happens". This is perhaps a side effect of the way we typically experience this kind of cinematography in films, in which the architecture is merely a setting for a plot to unfold. The focus is on the unfolding events, and the architecture is a secondary backdrop. But in our case and as stated previously, the architecture is the event - the actions of the avatar and the cinematic moves are aimed at "unfolding" the building itself. There are film precedents for this. It seems like this is not so much a limitation of our technique as it is in people's preconceptions.

There is a tendency to conceive narrative in terms of what was assumed to be the conventions operative in nine-teeth-century realist fiction –a linear development from origin to end. And, thus, any strategy that opposes the "naturalness" of these assumed conventions is thought to be "anti-narrative". In our case, in this our initial approach, we have opted for a linear strategy for simplicity as well as for being a well known and accepted way of structuring "stories". Our intention was always to use this accrued story telling knowledge as the starting point for investigating ways of structuring the VE spatial content to facilitate a reading of architecture in the design sense. At this point in time, we have not accumulated enough data to provide an informed judgement. We intend to solve this by testing the validity of our ideas in a variety of projects and collect the input given to us by the designers.

Since the architectural cinematographer was first developed, we have had the opportunity of testing it on a 13 different of projects. Currently, we are still analysing the results of our experimentation but two conclusions can already be drawn: first, it is difficult to extract factual evidence to sustain our argumentation about whether architectural concepts are understood with more ease for presentation or design purposes. Preliminary evidence suggests that it does. For instance, the following comment was extracted by an evaluation report completed by one of our users: "relationship between different elements of design is more apparent such as distances, scale, contrast". Secondly and more importantly, the use of the our system adds value to the current process as stated by all our users because our approach combines the freedom of movement and interaction provided by a VE but gives the designer a mechanism to control and divert the user's attention towards those architectural features he/she wants the user to experience.

6. REFERENCES

Alexander, C., Ishikawa, S., and Silverstein, M. (1977). A Pattern Language. Oxford University Press.

Alvarado, Rodrigo García. and Castillo, Gino Alvarez. (2003). Técnicas cinematográficas para las animaciones arquitectónicas. In proceedings of SIGraDi.

Arijon, D. (1993). “Grammar of the Film Language”. Communication Arts Books, hasting house, Publishers, New York.

Axling, T., Haridi, S, and Fahlen, L. (1996). Virtual reality programming in Oz. In Proceedings of the 3rd EUROGRAPHICS Workshop on Virtual Environments, Monte Carlo.

Charitos, D. (1998). The architectural aspect of designing space in virtual environments. PhD thesis, University of Strathclyde.

Christianson, D. B., Anderson, S. E., He, L., Cohen, M.F., Salesin, D.H., and Weld, D.S. (1996). Declarative Camera Control For Automatic Cinematography”. In proceedings of AAAI ’96, pp 148-155.

Drucker, S. and Zeltzer, D. (1995). CamDroid: A system for implementing intelligent camera control. In proceedings of the 1995 Symposium on Interactive 3D Graphics, 139-144.

He, L. Cohen, M. and Salesin, D. (1996). The virtual cinematographer: A paradigm for automatic real-time camera control and directing. In Proceedings of the ACM SIGGRAPH’96, 217-224.

Karp, P. and S.K. Feiner. (1990). Issues in the automated generation of animated presentations. Graphics Interface ’90.

Laurel, B. (1993) Computers as theatre. Addison Wesley Longman.

Madrazo, L. (2000). Computer and architectural design: going beyond the tool. Automation in Construction Vol 9 pp 5-17.

Nitsche, M., Roudavski, S., Penz, F., and Thomas, M. (2003). Drama and Context in real-time virtual environments: use of pre-scripted events as a part of an interactive spatial mediation framework. In TIDSE’03 Technologies for Interactive Digital Story-Telling and Entertainment. Darmstadt, Germany.

Norberg-Schulz, C. (1980). Genius loci: towards a phenomenology of architecture. Rizzoli, New York.

Rasmussen, S.E. (1992) Experiencing Architecture, MIT Press.

Saleh Uddin, M. (2001). Extents and limitations of 3D computer models for graphics analysis of form and space in architecture. The 19th ECAADE Conference, Helsinki, Finland..

Temkin, A. (2003). Seeing Architecture with a Filmaker’s Eyes. In proceedings of ACADIA 2003: connecting-crossroads of digital discourse.

UnrealEngine2.(2004). Internet. Available from http://udn.epicgames.com/Two/WebHome; accessed 17th of August 2004.

Yale (2004). Film Analysis Guide. http://classes.yale.edu/film-analysis

Wilenski, Reginal Howard. (1927). The Modern Movement in Art. London: Faber and Faber.

7. APPENDIX A

CC3P_CantedFraming

This actor rotates the view along the viewing axis (roll), providing a canted angle shot.

bProgressive -- if true, effect happens over a specified time period

bIsTimed -- if true, effect lasts for a specified time

ViewCantAngleDeg -- amount of roll in the camera view, specifed in degrees; positive numbers = clockwise roll; negative numbers = counter-clockwise roll default is 0 (zero) = level view.

ProgressionTimeSec -- number of seconds to complete progress of effect

EffectDurationSec -- number of seconds the effect lasts if bIsTimed = true, measured from completion of the effect's progression to full effect.

EndEffectTime -- for untimed effects (bIsTimed = false) the time an effect will last if Viewer is not actually in the triggering Volume.

CC3P_CutawayShot

Provides a viewpoint from the placed actor's position, looking at the specified look target. The camera is stationary and locked (no mouse control) and Viewer control of the pawn is disabled during the shot; cutaway shots are always timed effects.

bUseOnce -- if true, this effect can only be used once

bUseViewerFOV -- if true, FOV of effect = FOV of Viewer when actor is triggered

bUseCustomFOV -- if true, FOV of effect is specified by the LD Note: default effect FOV = default FOV of Viewer (usually 90 deg -- see CC3P_SceneInfo); range of FOV value is 2 - 150 (extreme telephoto to extreme wide angle)

ShotDurationSec -- number of seconds the cutaway lasts

ReUseDelaySec -- number of seconds before actor can be re-triggered (prevents repetitive triggering); default = 2 sec.

CustomFOV -- user-specified field of view for the effect, in degrees

LookTargetTag -- Tag of the actor the cutaway shot looks at; can be a LookTarget actor, but any actor's unique Tag can be specified.

CC3P_ZoomFOV

The field of view of the camera is modified.

bProgressive -- if true, effect happens over a specified time period

bIsTimed -- if true, effect lasts for a specified time

NewFOV -- user-sepcifed field of view; range of FOV value is 2 - 150 (extreme telephoto to extreme wide angle)

EffectDurationSec -- number of seconds the FOV remains changed

ProgressionTimeSec -- number of seconds for the FOV to change to the specified value

EndEffectTime -- for untimed effects (bIsTimed = false) the time an effect will last if Viewer is not actually in the triggering Volume.

CC3P_SceneInfo

Provides effects to start the scene with a fade in, a specified aspect ratio, and at a specified field of view (which becomes the default FOV for the Viewer in that map).

bOpenFadeIn -- if true, the level opens by fading in from black

OpeningFOV -- the field of view at level start; this becomes the Viewer's default FOV for the map.

OpenFadeInTime -- number of seconds for the opening fade from black to finish

PawnScale -- changes the size of the Viewer avatar (pawn) in the map, allowing fine adjustments to matchscale of map & pawn. Scale range is 0.477 minimum to 3.00 maximum, and is a multiplier (i.e., standard avatar size -- approx 91 Unreal units-- will bemultiplied by this factor to ge the in-map size).

CameraSetback -- Changes the distance between viewpoint camera and pawn. Can be used with PawnScale (equivalent values for both) to maintain relative "standard" distance, or can be customized to suit. The standard distance is multiplied by this factor.

CC3P_OpenAspectRatio

-- aspect ratio for the level; only the ratio of the two numbers is used, so user-specified units can be arbitrary.

ARO_FullAperture_133 -- 1:1.33

ARO_Academy_137 -- 1:1.37

ARO_Widescreen_166 -- 1:1.66

ARO_Widescreen_175 -- 1:1.75

ARO_Widescreen_185 -- 1:1.85

ARO_SixteenByNine_177 -- 9:16

ARO_Manual_XbyY -- user-specified ratio

ManualOpenAR_Width -- user-specified width factor

ManualOpenAR_Height -- user-specified height factor

CC3P_Fader

Enables a transitional fade out to black and back in; there are options for changing the Viewer's locationduring the black time, or for exiting to the map-select menu.

FadeInTime -- number of seconds for the fade-in to reach completion

FadeOutTime -- number of seconds for the fade-out to reach completion

TransFadeBlackTime -- number of seconds the fade stays at fully black screen before fading back in

bExitOnFadeOut -- if true, game will exit to the map-select menu at end of fade out of a TransitionFade

bRelocateOnFadeOut -- if true, Viewer pawn will be moved to a new location during the black time of a TransitionFade

RelocatePointTag -- Tag of the actor whose Location specifies where the Viewer ppawn is relocated when bRelocateOnFadeOut = true

CC3P_FrameCropping

Creates a reduced area of view in the form of a rectangular "window," framed in black. Size and center position of the visible window are specified by the LD.

bFadeCropping -- if true, cropping fades in & out progressively

bIsTimed -- if true, effect lasts for a specified time

FrameWindowWidth -- width of the visible screen area after cropping, specified as a decimal portion of the full-screen width

FrameWindowHeight -- height of the visible screen area after cropping, specified as a decimal portion of the full-screen height

FrameWindowCtr_W -- center of the visible screen area after cropping, specified as a decimal portion of the full-screen width

FrameWindowCtr_H -- center of the visible screen area after cropping, specified as a decimal portion of the full-screen height

CroppingDurationSec -- number of seconds the effect lasts

CropFadeTimeSec -- number of seconds for the cropping effect to fade in and fade out

EndEffectTime -- for untimed effects (bIsTimed = false) the time an effect will last if Viewer is not actually in the triggering Volume.

CC3P_TrackingShot

Repositions the camera while maintaining the pawn-centered interest, moving with the pawn as the camera changes position. Camera-to-pawn distance during the effect is determined by the pawn's distance from the placed actor's location when the effect starts, allowing for extreme changes in viewpoint.

bTimeProgressive -- if true, camera moves to new position in time specified in ProgressionTimeSec

bRateProgressive -- if true, camera moves to new position at a time specified in EffectRate; range is greater than zero to 1.0

bIsTimed -- if true, effect lasts for a specified time

bReverseRotation -- if true, rotation direction of camera is reversed; by default, camera uses shortest route to new position -- this option uses the longest route

ProgressionTimeSec -- number of seconds for the camera to move into new position & return when effect is done

EffectDurationSec -- number of seconds the effect lasts

EffectRate -- relative speed the camera moves to new position

EndEffectTime -- for untimed effects (bIsTimed = false) the time an effect will last if Viewer is not actually in the triggering Volume.

CC3P_FilterEffect

This actor implements HUD overlay effects that act asvisual filters on the scene. Currently the type of filter is limited to simple textures or specific kindsof FinalBlends that can be created using the supplied"filter templates" in the CC3P_Textures.utx file (See the FilterFX group). Sample filters are included, and the "Filter Facts" readme describes the steps forcreating custom filter materials.

bIsTimed -- if true, effect lasts for a specified time

FilterMaterial -- the Material used for the filter effect

FilterStrength -- specifies how much effect the filter has on the scene. Value range is 0 to 1.0. Visual effect of filter strength depends on the type of filter used, e.g. a full-strength simple texture filter and a full-strength FinalBlend Brighten filterwill have very different effects.

FilterFadeTime -- the time in seconds that it takes the filter effect to reach full FilterStrength, and to fade out completely when FilterDurationSec has elapsed

FilterDurationSec -- the time the filter is in effectat full FilterStrength.

EndEffectTime -- for untimed effects (bIsTimed = false) the time an effect will last if Viewer is not actually in the triggering Volume.

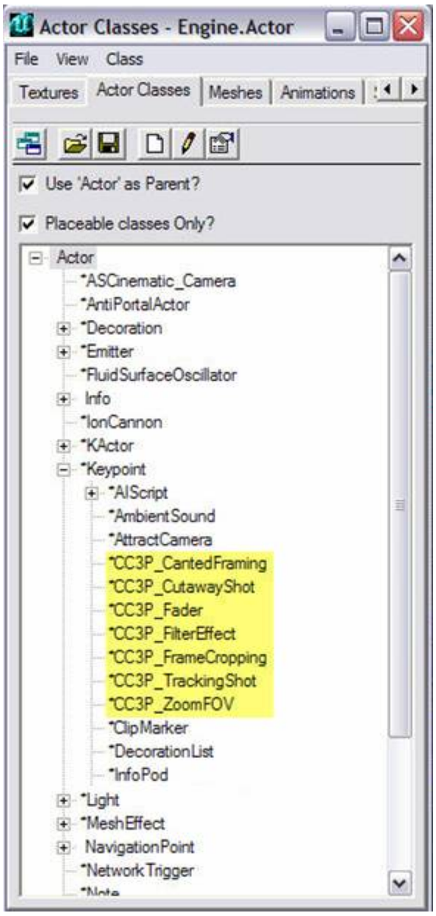

FIG. A1: Custom-coded actors derive from keypoint. These actors are available through the level design editor (UnrealEd) and, by dragging and dropping, they can be placed in the map at the user's will.

8. APPENDIX B

It is important to introduce the environment to the user in a way that makes them feel comfortable to begin navigating. The way to do this would be to give them an understanding of the context they are in. At position (A) we achieve this by using a cinematographic technique called the 'establishing shot' (C1). It is the duty of the establishing shot to introduce the scene to the viewer and to give them a sense of where they are going – rather than having the user walk around freely, hoping that they might discover the more important areas of the environment the establishing shot serves to heighten the anticipation of the user as they see areas which are immediately inaccessible but which they can search out the path for. In an architectural sense the establishing shot gives the user a sense of scale between himself and the environment, as well as the built form within the environment in relation to other built form, the composition of the scheme is apparent (A1).

The user then begins to reach the main development within the environment and the tension begins to rise, there is a greater desire to explore the building. At point (C) we introduce the 'tracking shot' (C2) to view the user as he walks alongside the colonnade. Here the intention is to give the viewer a sense of the rhythm (A2) inherent in this part of the building – viewing the avatar as he walks past the columns with a particular speed and motion allows the viewer to understand the spacing of the columns in relation to walking pace rather than as a static image.

One of the more prominent features of this scheme is the rust-steel clad tower that acts as a circulation pivot point within the environment. To bring the users attention to the height and mass of this tower a technique 'angle of framing' was utilised (C3). This shot provides a static image with no movement of the camera or avatar possible. The user pauses and observes what we as designers are representing to them. The avatar is shown stood next to the tower from a low perspective, drawing the users attention to the scale (A3) of the tower.

The user is now immersed in the environment and is achieving the highest sense of exploration and anticipation on the narrative scale (D-E).

Moving further along the narrative path we utilise 'frame cropping' (C4) to highlight the fenestration on the building that now lies in front of the user. This acts as an attractor to bring the user closer to this piece, to focus their eye on a particular area of the screen thus blocking out all context. When the user reaches this area of the map they will be inclined to investigate this area in more detail. This action is also twinned with the next shot, another tracking shot (C2) which views the user from a perpendicular perspective as he walks past this building. The tracking shot here is set at a distance which draws the users attention to the proportion (A4) of the second building of the environment. Here he also recognises the fenestration viewed in the previous shot and gains a more rounded knowledge of this particular part of the map.

At point F the tension and anticipation begins to decline as we prepare to release the user from the environment. Exploring the lower area of the VE the user encounters a structure consisting of 4 vertical planes. On entering these planes we utilise a CT named 'canted framing' (C5). Here the camera rolls as the user enters the space and the vertical planes twist as he walks. Attention is drawn to the verticality (A5) of the space as the user becomes slightly disoriented – he looks to the immediate surroundings for safe markers which enable him to understand the change in view and encounters the enclosing planes of the space. As he moves further into the structure the view returns to its normal position and the planes move in view once again. The viewer observes these vertical planes moving in virtual space.

Reaching point (G) leads on to the final technique used in the environment which mirrors the opening technique. Here we use a 'releasing shot' (C1) which performs the same motion as the establishing shot but rather than introducing the scene to the user it allows them to recapture the journey they have taken, to observe the architecture in its entirety, to once again evaluate the composition (A1) of the scheme with a deeper understanding of some of the architectural concepts imbedded within the design, and to release themselves comfortably from the environment.

Through the use of carefully placed cinematographic techniques the user is guided through the environment and is urged to follow a narrative structure of which they are not aware. The story teller is able to tell their story without being direct, the story is uncovered by the user at their own pace – this gives them a much better understanding of the environment as they make decisions to discover particular areas themselves.

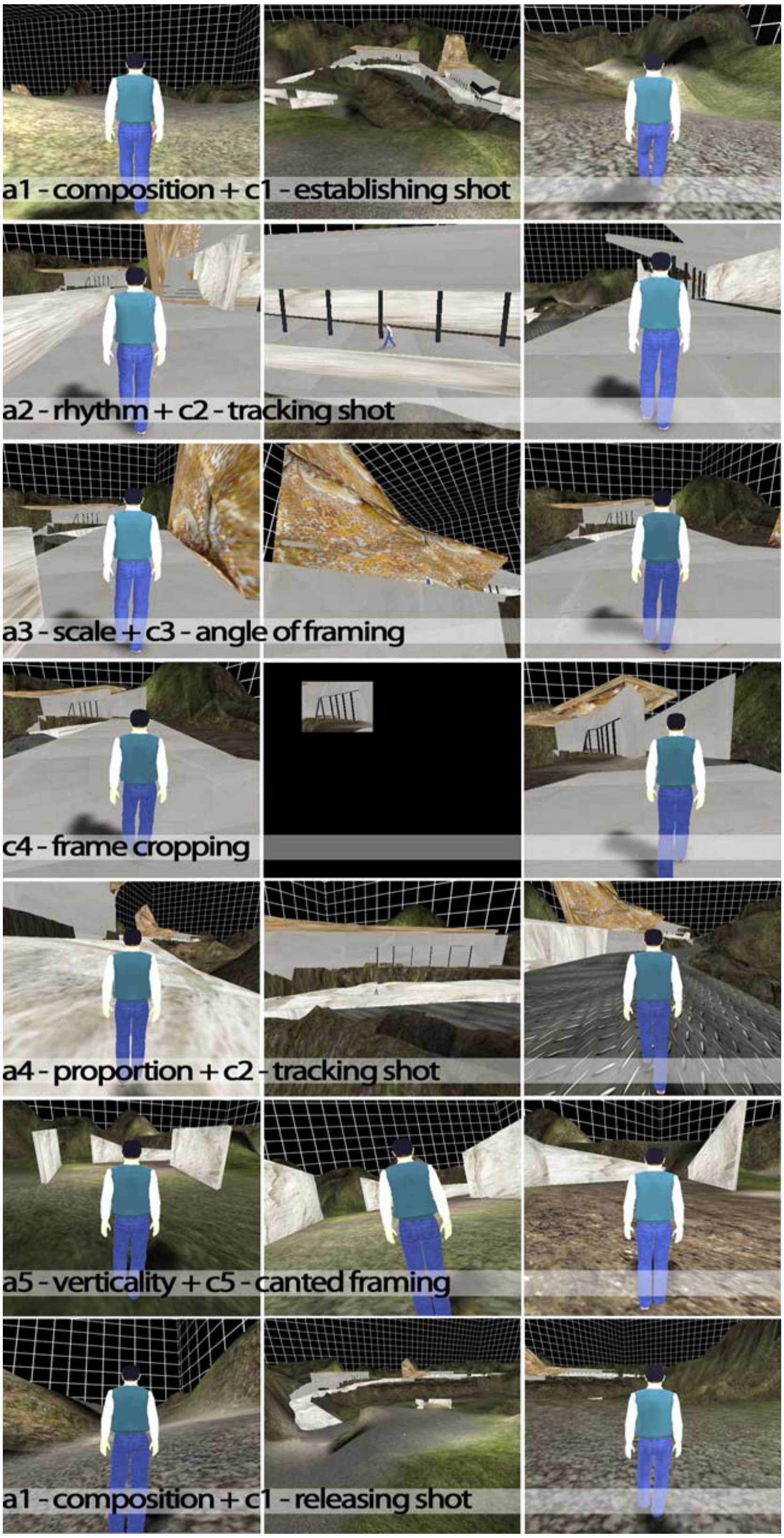

FIG. A2: Architectural concepts with its partnered cinematographic techniques as it used in the architectural walkthrough.