Hybrid Interface Design for Distinct Creative Practices in Real-Time 3D Filmmaking

Michael Nitsche
Georgia Institute of Technology
Atlanta, GA, USA
michael.nitsche@gatech.edu

Friedrich Kirschner
Hochschule fuer Schauspielkunst Ernst-Busch
Berlin, Germany
f.kirschner@hfs-berlin.de

ABSTRACT

TUIs have become part of a larger digital ecology. Thus, hybrid interface design that combines tangible with other interaction options is ever more important. This paper argues for the importance of such hybrid approaches for creative practices that use divergent collaborative processes. It presents the design, evolution, and implementation of such a hybrid interface for machinima film production. Finally, it provides initial reflection on the use and preliminary evaluation of the current system.

Author Keywords

Hybrid interface, tangible, machinima, filmmaking.

ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation]: User Interfaces---*input devices and strategies, interaction styles*; J.5 [Arts and Humanities]: *Performing arts*. K.8 [Personal Computing]: *Games*.

General Terms

Design, Experimentation.

INTRODUCTION

Tangible interfaces have become widely accepted, next to more traditional systems. As a consequence, interaction design has become a multi-facetted task where the question is not so much whether the interface includes a tangible part or not, but which kind of tangible and non-tangible interfaces have to be combined to optimize creative engagement. With increasing interface cultures, these investigations have to spread across *digital ecologies* that are defined not by single creative activities undertaken with a specialized interface but of shared creativity where “collaboration is distinctively concerned with the articulation of fragmented interaction” [4]. The design has not only to support the distinct interaction but also to facilitate the merging of them to creative collaborations.

This paper presents Cinematic Interfaces, a project that combines object-based interaction and 3D tracking on a tabletop-like interface. The assemblage of these different approaches did arise from the individual creative tasks at hand: real-time CGI filmmaking. This form of real-time animation has been known as machinima. Filmmaking and machinima are good examples for creative interface ecologies because there is already considerable expertise in both, but the practices are often calling for different designs. For example, set design uses different methods than camera control or actors’ performances. The goal of Cinematic Interfaces was to combine different creative practices of filmmaking in such a way as to allow each practice to use its own “distinct” control schemas and still shape the complete output in real-time collaboration.

CINEMATIC INTERFACES

First, the paper briefly outlines the culture of machinima and why it calls for hybrid interface design. Then, it presents a prototype design and its implementation. Finally, it provides first qualitative expert feedback on the individual parts and an outlook for future development.

Machinima as Creative Practice

Machinima has been described as “animated filmmaking within a real-time virtual 3D environment” [6]. It is a digital art practice that emerged from gaming via hacking and expert playing into an own sub culture of animation [7]. It is largely based on real-time graphic rendering done with the help of game systems. Along the way, the different practices of these communities have shaped the tools and production methods in machinima. Machinima includes many creative practices from filmmaking: from camera control to acting, to editing, to set design, and directing, but it realizes them in the 3D video game engines available today. For machinima, these practices remain connected to the culture of gaming. The question, then, is how to build tools that balance the machinima-typical approaches and existing practices of filmmaking without extinguishing the particular quality of machinima as a game-based cinematic format. How can we support creativity across multiple disciplines that need to collaborate seamlessly in order to perform the particular art of machinima? As a particular limitation, machinima is a digital art form performed and developed by non-experts. Thus, the interfaces have to be accessible (affordable) as well as usable for non-experts.

Existent Approaches

A range of commercial video games ship with tools for machinima creation. These tools usually allow the player community to script and trigger particular cinematic

TEI 2013, Feb 10-13, 2013, Barcelona, Spain.
Copyright 2013 ACM 978-1-4503-1898-3/13/02....$15.00.

sequences, cutscenes, as well as import 3D objects via plug ins from content creation suites like Autodesk's Maya. By and large, this approach mirrors the production pipeline of game companies. The resulting tools are often in-house editors for the creation of game content that are released to players in order to encourage them to produce new content. Valve's Source Filmmaker, the content editor for their game engine, is one example for an engine-based tool that follows game production conventions. The approach focuses on game-specific practices from the scripting of animations and events to control of 3D geometry and camera performance.

However, production practices in the film industry have established their own traditions and preferences that are not always compatible with the gaming world. For example, cameras in film production are controlled by tactile means and not in a disembodied fly-through way as seen in games. That is why specialized commercial interfaces emerged for this niche market. For example, the Gamecaster system (www.gamecaster.com) is a specialized interface to control cameras in virtual environments through 3D tracking. It is geared to a professional market and game-related TV programs. However, its high price makes it unfeasible for the machinima community that largely consists of enthusiastic but notoriously underfunded producers.

Systems like Moviestorm or iClone are software packages for real-time animation and offer an own blend of tools and interfaces. These real-time render engines lack regular gameplay features but focus purely on machinima production. While they function as machinima production studios, they resemble more a low cost version of 3D modeling and animation tools, such as Maya, that have been optimized for machinima. The playful approach through direct interaction and the context of the game world is sidelined in these engines. Non-game-based approaches using TUIs often focus on limited aspects of virtual filmmaking, such as scanning and performing objects [9] or multi-touch based camera positioning [10]. Our open system aims to cover a range of creative areas simultaneously.

Designing the Interface

This paper proposes that advanced machinima tools need to balance the cinematic and the game-based practices to further this new digital art form. Otherwise, the production practices will be shaped too much by either side and weaken machinima as an own digital art infused by multiple influences. To facilitate such an open approach, our system is modular and open sourced. It can (and has been) extended with additional input and output devices. Here, we will focus on the design and evolution of two core elements: set control and camera control.

Camera Control: Design

Camera control and optimization has been a relevant area of research since the rise of widely accessible 3D environments and particularly with the rise of 3D video games. Foundations and key metaphors for such a camera control were established already in the mid 80s to early 90s [8]. They introduced basic concepts such as the "eyeball in hand" or "scene in hand" approaches. On the hardware level, new technologies have become accessible to consumer level audiences with interfaces such as Nintendo's Wii and Microsoft's Kinect, leading to new hybrid interfaces for camera control (e.g. [2]). In contrast to these implementations, game-based interfaces often support more playful and less pre-planned engagement. In fact, we have shown in earlier work, that the playful interactive visual exploration of the scene is a typical feature of machinima practice [5]. Multi-touch based approaches to camera placement [10] lack necessary degrees of freedom when placing virtual cameras in a 3D scene. Thus, a playful yet flexible input mechanism was the goal.

Camera Control: Evolution

Our initial approach to camera control focused on the positioning of a virtual camera in physical space, simulating the affordances of traditional physical camera placement. A first prototype consisted of a GameTrak input device providing 3D positional data fed into the Unreal Tournament 2004 game engine. We extended this initial approach using an Ultra Mobile PC, reading rotational data from a 3 DOF sensor. This allowed direct rotational control of a virtual camera by rotating a screen that mimicked the virtual camera's perspective. This was implemented in Unity. Not unlike the Gamecaster model discussed above, these prototypes translated physical camera-placement affordances into virtual space that were easy to understand and to use for camera operators. Virtual camera handling was closely related to real world handling of our interface. But this direct mimicking of real film production techniques bypassed the machinima specific qualities we aimed for.

Thus, we employed a common visual tool for camera decision-making, namely the floor plan, but applied it in a new way. We aimed to visualize multiple camera positions within a given scene and combine this with the tangible direct control provided by new interfaces. This approach has multiple advantages. Because multiple cameras can be represented at the same time, their relation to each other becomes clear, which is an important aspect for continuity editing [3]. The scene becomes condensed into a manageable physical interaction space that is easy to traverse. Information about non-camera objects such as actors and props is easily communicated. Yet, control itself remains "in the hands" of the operator at all times. To support the direct physical input method and allow a high degree of camera manipulation, we decided to track a user's hand and finger as representation of a selected camera. This is an extension of the "eyeball in hand" approach. At the same time, it adds a simple and playful element without restricting the expressive range of the virtual camera work.

Stage Control: Design

So far, the virtual stage in machinima productions is created and largely controlled by purely virtual means. Either the filmmakers re-use the existing game levels provided by the game developer or they create own virtual environments using available tools. With very few exception (e.g. www.previzart.com), no direct control over sets is possible in machinima production. While virtual stage design has become more important in today's TV and film production techniques, it does not reflect the more tangible way that many set designers apply when creating physical stages. This process is often compared with "sculpting" [1] sets and needs direct access to the stage and the décor. As much as 3D modeling programs provide metaphors for this process, they remain digitally translated versions of this sculpting.

Stage Control: Evolution

Initially, our system used pre-fabricated 3D scenes in the Unreal and Unity installations. These systems provide different import options for virtual models from other sources as well as internal modeling options. Once again, this mirrored only existent technology in game practices. Thus, we stepped away from the pre-fabricated scenery approach and instead aimed for a more direct control mechanism that allows users to design and manipulate virtual stages through direct object interaction. Such a design reflects the playful character of game-based creativity as well as the direct sculpting processes of more traditional set design approaches. This led to an object-tracking interface for real-time set production. Unlike pre-fabricated objects that could be tracked (e.g. using RFID or visual markers), the system aimed to allow for direct use of any object for a real-time set creation and control.

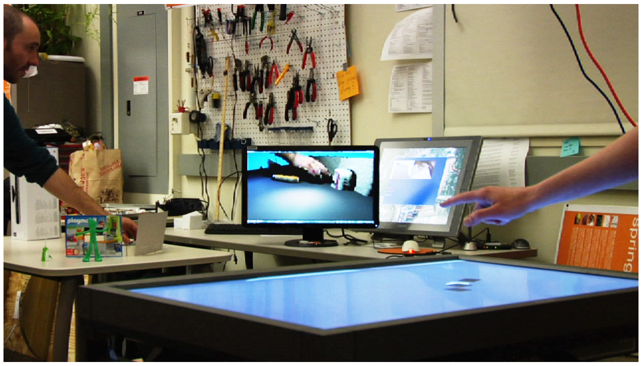
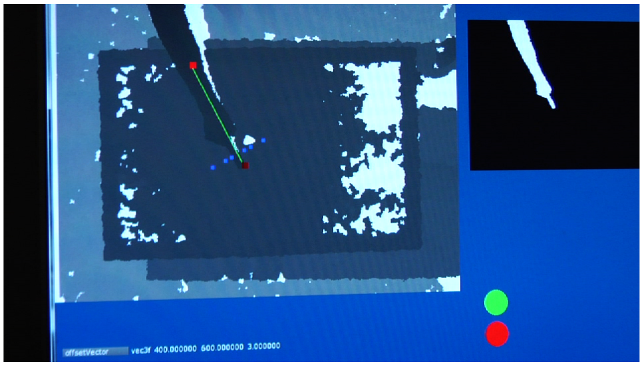


Figure 1. Cinematic Interfaces: (front) camera position and orientation are controlled by the finger hovering above the screen; (back) live camera view rendered on separate monitor and stage control using objects.

Implementation

Following a modular software approach, we created two independent programs and a set of commands transmitted via the Open Sound Control Protocol to interface with an existing open-source 3D real-time renderer. We chose the MovieSandbox engine because of our familiarity with the codebase, its open-source nature, and its open architecture. Thus, only rudimentary additions had to be implemented to adapt it to the specific needs of our architecture. Our physical setup consists of a horizontally placed monitor for the Processing application that controls the floor plan. It visualizes the camera positions and placement conventions as a top-down view. A ceiling-mounted Microsoft Kinect sensor is used for hand and gesture recognition and another computer screen for the virtual camera's point-of-view representation. A second Kinect is used to track objects on a small physical model stage. We use a single Windows computer for our setup, although the modular networked nature of our software would allow the individual tasks to be split among multiple computers. The system ran on an Intel Core2Duo Processor at 3.2 GHz with 8GB of RAM and a Nvidia GeForce 570GTX graphics card, remaining affordable for machinima producers.



offsetVector vec3f 400.000000 600.000000 3.000000

Figure 2. finger tracking for camera control.

Camera Control: Implementation

To track the hand as user input, we use a Microsoft Kinect image sensor and use image analysis that converts hand input to five degrees of freedom, providing both three-dimensional positional data as well as pitch and yaw for camera manipulation. A user can move a selected camera by extending the index finger and release it from direct control again by retracting it. Once selected, the visual representation of the camera follows the user's hand in real-time visualized on the schematic tabletop view where a selected camera follows the controlling hand/ finger on the flat surface. The camera's point of view is additionally represented in the virtual environment using a 3D real-time rendering engine on a second screen next to the interaction space (see center fig. 1). This puts fast, direct manipulation and iterative adjustments of multiple cameras quite literally in the hands of the filmmaker.

Stage Control: Implementation

A second Kinect camera faces the physical model stage. It uses the depth perception of the Kinect to create a point cloud model of the sensed objects on this stage in real-time. This model is rendered in MovieSandbox and textured with the image data also recorded from the Kinect. The result is a real-time tracking of objects on the visible stage that represents them as virtual objects in the 3D environment. Its granularity depends on the real-time performance. As we focused on real-time feedback, we chose higher response rate over higher granularity. Because the objects are digital props, they are affected by light or other stage elements in the render engine, allowing for fine-tuned adjustments if needed.

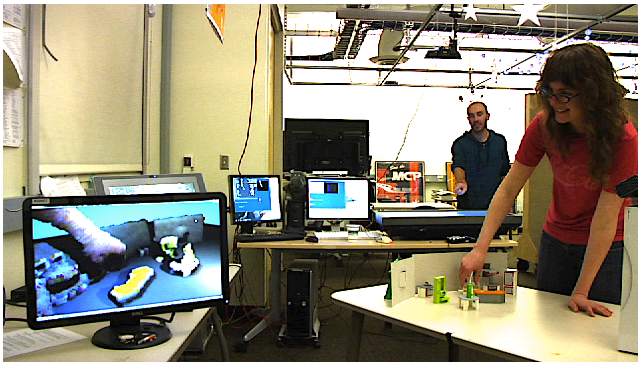


Figure 3. physical manipulation of the model stage and real-time effect on the virtual set.

Discussion

To evaluate the quality of our hybrid interface design for creative collaboration in real-time filmmaking, we have used the system at multiple workshops, such as the Play 2011 Festival for Creative Gaming in Potsdam or the International Festival for Animated Film 2012 in Stuttgart. Participants included non-expert users (aged 16-18) as well as experts in camera, set design, and puppetry. Its main interaction design is immediately understood and stimulates the kind of playful and explorative engagement that was the goal of the project. Its modular approach facilitates integration with other real-time engines and production environments. In fact, we have already combined the system with a real-time puppeteering input and it also contains a supportive AI component that assists in the camera selection and positioning. Thus, the system has proven its technical merits as part of a larger real-time production framework.

We have not yet conducted a full quantitative evaluation but the first step of acceptance of the interface appears to be successful. This is based on feedback from non-experts, but maybe more importantly, also from experts of different filmmaking domains, where we expected more skepticism. One director of photography noted on an early camera control test that s/he "feel[s] that it is much closer to reality. It is very easy to find good positions and it gives you the freedom to go for the extreme angles." Thus, the playfulness in the interaction design supported professional creative needs. At the same time, a set designer noted that "[t]he fact that I could just create my set using cardboard and paper with instant visual feedback was amazing. I am much more used to working with my hands than to learn a software program." This validates the kind of creative empowerment, which was the goal of our hybrid interaction design.

Conclusion

We presented a sustainable paradigm for a hybrid interface for camera and set control in real-time CGI animation. It is sustainable because it allows for simple yet exhaustive modes of distinct interaction that were derived from a combination of film traditions and game-based approaches. The system combines a balanced machinima specific approach that is driven by playful exploration of the virtual world through innovative interfaces. It provides an effective cinematic camera and tangible stage control that relate to already existing work practices in film and TV production. Key contributions of our work are: to argue for a hybrid approach to support the dual origins of machinima's creative practice and to present an example implementation. The evolution of the project shows the design as well as technical developments needed to provide a reference for other projects that attempt a comparable hybrid interface design for similar creative collaboration.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under Grant No. IIS-1002748 and benefitted from our collaboration with Ashton Grosz, Mark Riedl and the Entertainment Intelligence Lab at Georgia Tech

REFERENCES

1. Affron, C. and Affron, M. J. (1995). *Sets in Motion. Art Direction and Film Narrative*. Rutgers University Press, New Brunswick.

2. Benford, S., Greenhalgh, C., Craven, M., Walker, G., Regan, T., Morphett, J. and Wyver, J. (2000). Broadcasting Interaction from within Collaborative Virtual Environments. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 7, 4, 510-547.

3. Bordwell, D. and Thompson, K. (2004). *Film Art. An Introduction*. McGraw-Hill, Boston.

4. Crabtree, A. and Rodden, T. (2008). Hybrid ecologies: understanding cooperative interaction in emerging physical-digital environments. *Personal and Ubiquitous Computing*, 12, 7, 481-493.

5. Davis, N., Li, B., O'Neill, B., Riedl, M. and Nitsche, M. (2011). Distributed creative cognition in digital filmmaking. *Proceedings of the 8th ACM Conference on Creativity and Cognition*. ACM, NY, 207-216.

6. Marino, P. (2004). *3D Game-Based Filmmaking: The Art of Machinima*. Paraglyph Press, Scottsdale, AZ.

7. Salen, K. (2002). Telefragging Monster Movies. *Game On. The History and Culture of Videogames*. Ed. L. King, Laurence King Publ., London, 98-112.

8. Ware, C. and Osborne, S. (1990). Exploration and virtual camera control in virtual three dimensional environments *SIGGRAPH Comput. Graph.*, 24, 2, 175-183.

9. Held, R., Gupta, A., Curless, B., and Agrawala, M. (2012). 3D Puppetry: A Kinect-based Interface for 3D Animation. *UIST 2012*. ACM, NY, 423-434.

10. Herder, J., Brosda, C., Djuderija, S., Drochtert, D., Genc, Ö., Joeres, S., Kellerberg, P., Looschen, S., Geiger, C. and Wöldecke, B. (2011). TouchPlanVS - A Tangible Multitouch Planning System for Virtual TV Studio Productions. *Proceedings 3DUI 2011* (IEEE Symp. on 3D User Interfaces), Singapore, March, 2011.