

Video Games: A New Interface for Non-Professional Game Developers

Oliver Gray

Department of Computer Science
North Carolina State University
Raleigh, NC, 27695
e-mail address: ohgray@unity.ncsu.edu

R. Michael Young

Department of Computer Science
North Carolina State University
Raleigh, NC, 27695
e-mail address: young@csc.ncsu.edu

ABSTRACT

New environments and technologies allow non-professional users to produce and share media content in a variety of environments. One such environment is video games. This paper describes Virtuoso, a multi-user programming environment built using Valve's Source engine that functions as a tool to allow non-professional users to create interactive educational video games.

INTRODUCTION

New technologies have shifted media creation out of the hands of specialists and into the hands of non-professionals. For example, online video sites such as YouTube and online blog sites (e.g. Xanga, Facebook, MySpace) have allowed non-professionals to create and distribute media in ways that previously only existed for professionals. This shift in media creation also exists in the world of computer game *mods* – extensions or modifications to existing commercial computer game using a range of development tools in order to create a different version of the game.

New interfaces have been created to support both non-professional media development and viewing of the media product. However, these interfaces can be improved to elicit greater user involvement, communication, sharing, and general enjoyment when developing and viewing the product (i.e. media). This paper discusses an interface for non-professional media development based on computer games called Virtuoso, which is part of the HI-FIVES (Highly Interactive Fun Internet Virtual Environments in Science) project [1].

Virtuoso is multi-user programming environment targeted at novice non-programmers. It functions as a tool to allow such users to create interactive educational video

games. Virtuoso is based on Half-Life 2, a popular video game made by Valve [2].

Using a game environment as the foundation for an interface through which to build computer games allows for the following three important features:

- development and viewing occur in the same environment
- game environments are designed with social interaction as a driving feature
- game environments have an inherent fun factor

RELATED WORK

Alice [4] is an interactive 3-D programming environment built with the goal of making it easy for new programmers to create 3-D environments. The creators of Alice (the Stage 3 Research Group) intended it to serve as a novel interface in which new computer science students can learn the fundamentals of computer programming.

Alice and Virtuoso differ in two areas: multiplayer and scripting language. Alice does not support multiplayer while it is supported in Virtuoso. One of Alice's goals is to teach basic programming concepts to beginners. In order to accomplish this goal, Alice has a rich scripting language which contains many of the same features seen in most programming languages. Virtuoso's scripting language is still under construction, so at this point it only contains basic control structures (i.e. if/else, for, while, etc). This scripting language will be expanded in future versions.

MUPPETS (The Multi-User Programming Pedagogy for Enhancing Traditional Study) [5] is a collaborative virtual programming environment based on the Java language. Its goal is to enhance student education in the areas of programming and problem solving. The creators propose that creating a virtual world where

students can interact with study partners using avatars and visual representations of their work will achieve this goal because of the growing strength of virtual relationships.

Virtuoso and MUPPETS differ in their level of complexity. MUPPETS is designed for first and second year computer science students and expects users to have a basic understanding of JAVA and object-oriented concepts. In MUPPETS, users write actual java code in order to create/edit objects. Virtuoso is designed for users with no computer science background and takes a different approach to coding. In Virtuoso, code is generated using a template system where the user fills in the preset blanks. This allows novice users to develop code where the only possible error is a logical error (see Figure 1).

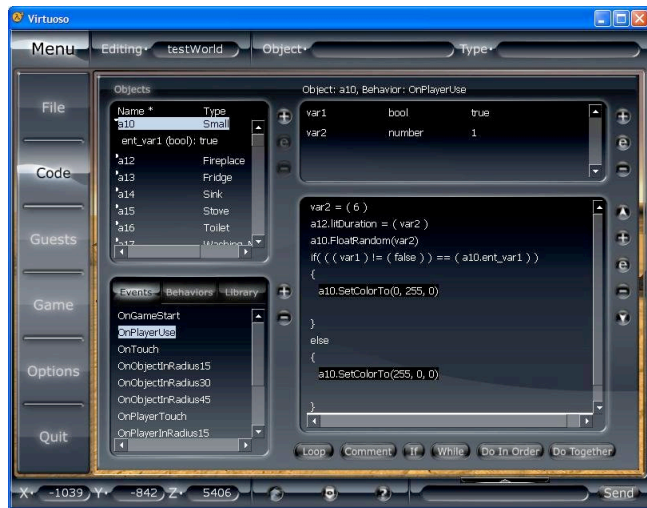


Figure 1: Programming Panel in Virtuoso

Kid's Programming Language (KPL) [6] is a simplified integrated development environment and programming language designed to capture the interest of beginner programmers and lessen the learning curve of programming. KPL's IDE mimics Visual Studio and Eclipse in order to provide a smooth transition from KPL to other programming languages and IDEs. Its largest libraries are those related to video games, as the authors have found that creating video games is one of the best ways of keeping beginner programmers interested in programming.

Virtuoso and KPL differ in their goals and interfaces. KPL's goal is to provide a medium for novice programmers that will keep them interested while also teaching them how to program. Virtuoso's goal is to provide a tool for users with no programming experience that will allow the users to create educational event-driven multiplayer video games. KPL's interface is designed to be similar to widely used IDE's interfaces such as Visual Studio and Eclipse while Virtuoso's interface is designed with the central idea of keeping things simple yet powerful enough for non-programmers to implement their ideas.

HI-FIVES AND VIRTUOSO

The goal of the HI-FIVES project is to allow middle and high school teachers and their students to develop their own video games using tools made for non-professional game developers. These teachers and students can then play those games in the context of their course curriculum. Virtuoso provides both the development environment in which the users build their games and the run-time environment in which the games are played. Virtuoso is a computer game mod based on Source, the engine running Valve's popular computer game, Half-Life 2 [2].

Using Virtuoso, users start with an empty game world, populate that world with different objects selected from a predefined library, and then specify event-driven behaviors for each added object. For example, a teacher using Virtuoso could start the program in edit mode, add a crate to his or her world, and then assign it the behavior: FloatRandom(5) for the event OnPlayerUse. When students loaded this world in play mode, the barrel would float in a random direction for 5 seconds when any of the students pressed the *use* button while facing the crate.

Virtuoso is designed to support non-professional users, specifically middle and high school teachers and students. The key factors to achieve this support are intuitive interfaces, help that is always accessible, and guaranteed syntax free coding. Interfaces are built around existing standards when possible. For example, file manipulation is handled through the File menu, as is the standard for most software products. Help messages are clear and available to the user at all times during use of Virtuoso. The interface for users to add behaviors for objects (the coding interface) is designed to prevent users from creating syntactically incorrect code by providing a GUI-driven method for building behavior logic. This allows users with varying technical experience to avoid common pitfalls of coding and instead focus on creating code that logically achieves their desired goal.

INTERFACE

Context switching between developing and viewing content interrupts the development process, which can cause the user to be less involved in the content that they are developing, decrease the speed and joy of producing content, and cause the user to be less likely to share the content. For example, when uploading a video to YouTube, there is a delay between when the user uploads the video and when the video is available to the public. During the delay between developing (uploading) and viewing (available to the public), the user is likely to lose interest in the project, feel frustrated with the production process, or forget to share the content with their friends.

In Virtuoso this context switching between development and viewing does not exist because development and viewing are done in the same environment. A user can develop content and then immediately view that content without any time or concentration lost from context switching. For example, a user can add an object and the object immediately appears in-game, or the user can add a behavior then immediately view the execution of the added behavior. This immediacy is handled through the interface by denoting two different modes: edit mode for editing games and play mode for playing games. Objects and behaviors are modified in edit mode and behaviors are executed in play mode. The user can quickly switch between play and edit mode through a mouse-click or hotkey.

One way to encourage user involvement, increase joy of producing, and inspire users to communicate and share content is through a multi-user environment. Research in the field of pair-programming (a multi-user environment) has shown that when two or more users work together, each user feels peer pressure to perform at their best for the other user [7,8]. This peer pressure causes a heightened level of user involvement. The same research shows that people feel a higher degree of enjoyment and communicate better while producing content with someone else rather than through single-user production. In a single-user environment, the user must leave the environment in order to communicate and the development process cannot be shared between users. A multi-user environment allows sharing during development and viewing and should also produce a greater level communication by allowing communication within the environment.

Virtuoso is a multi-user environment and was built to support a high degree of communication and content sharing through avatars, concurrent multi-user developing and viewing, and chat capabilities. Virtuoso games (whether in edit mode or play mode) are accessible to all Virtuoso users via a network browser interface that shows all currently running Virtuoso instances. This encourages communication and sharing by supporting users working with each other to develop games and also experience games together. Virtuoso users joining the same game share the same virtual space, have avatars that are visible to each other and communicate using in-game text chat and, when sufficient network bandwidth is provided VOIP audio chat.

User involvement and enjoyment can be increased by making the development and viewing environment fun and interesting to the user. KPL's creators [6] found that video games are such an environment and created video game libraries for KPL in order to have a greater probability of keeping user interest and joy of producing at high levels. Virtuoso intends to create the same phenomenon by using a game environment as an interface for users to develop content.

In order to heighten user involvement and enjoyment while using an interface based on video games, it is important to keep the same factors that make a video game fun in the interface. Virtuoso's design seeks to retain the elements that contribute to the fun factor of video games. Although Virtuoso modifies the game environment to support the creation and play of custom games, fun game features still exist in the interface. For example, in edit mode, the Virtuoso Heads-Up Display (HUD) is made to look like the interface of a hi-tech device one might find in a futuristic class laboratory (see Figure 2). This creates the illusion that the users are creating their worlds by interacting with different parts of the device. Within the editing interface, Virtuoso allows edit-mode users to modify the play mode HUD, selecting from a range of pre-defined "skins" (i.e. a HUD that resembles a nature theme with a timer in the lower-right corner). This modifiable HUD allows users to create a specific look and feel for each custom game.

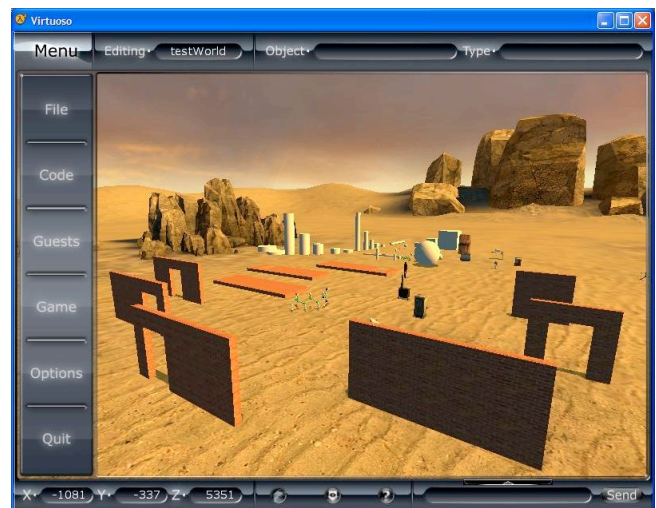


Figure 2: Virtuoso Edit Mode HUD

CONCLUSIONS

New advances in technology have allowed for media creation by non-professionals. The interface for this new technology should both motivate the user to create media as well as encourage social interaction through communication and sharing. This paper described Virtuoso, an interface for non-professional users that is conducive to a high degree of user involvement, communication, and sharing. Virtuoso's interface is based on conventions and interaction designs common to computer game environments and is unique in its immediacy, ability to encompass both development and viewing, and high level of supported communication and sharing during both development and viewing.

ACKNOWLEDGMENTS

The authors would like to thank the HI-FIVES team members for their help in the design and development of the work described here, including Len Annetta, John Park, Tim Burnett, Nick Darnell, Smith Newnam and David Tredwell. This work was supported by award 0525115 from the National Science Foundation.

REFERENCES

1. HI-FIVES. Available: <http://ced.ncsu.edu/hifives>
2. Valve. Available: <http://www.valvesoftware.com>
3. Anderson, R.E. Social impacts of computing: Codes of professional ethics. *Social Science Computing Review* 10, 2 (1992), 453-469.
4. Cooper, S., Dann, W., Pausch, R. Alice: A 3-D Tool For Introductory Programming Concepts. *JCSC 15*, May 2000.
5. Phelps, A.M., Bierre, K.J., Parks, D.M. MUPPETS: Multi-User Programming Pedagogy for Enhancing Traditional Study. *CITC4' 03*, October 16-18, 2003.
6. Schwart, J., Stagner, J., Morrison, W. Kid's Programming Language (KPL). *SIGGRAPH '06*. 2006.
7. Williams, L.A., Kessler, R.R. The Effects of "Pair-Pressure" and "Pair-Learning" on Software Engineering Education. *Conference of Software Engineering Education and Training*. 2000.
8. Williams, L. and Upchurch, R. In Support of Student Pair Programming. *SIGCSE Conference on Computer Science Education*, Charlotte, NC, pp. 327-331