Tools for Creating Interactive Teaching Environments

Yusuf Pisan
University of Technology, Sydney
Australia
ypisan@it.uts.edu.au

Chi Wai Wong
Macquarie University
Australia

Abstract: Although learning-by-doing is one of the most effective teaching and learning methodology, its application to some disciplines, such as history, is difficult even when using complex simulation environments. One possible solution is to combine learning-by-doing principles with gaming elements to create immersive and interesting virtual environments. Unfortunately, the level of programming required to create such environments makes it prohibitive for most educators. We describe the Graphical Interactive Fiction Toolkit, GIFT, system we have created that allows educators and writers to easily create interactive stories. While programmers will still be required for complex interactive environments, we believe GIFT can lower the entry bar and allow much richer set of interactive environments to be produced.

1. Interactive Fiction

Traditional books present a single, linear storyline. The only choice available to the reader is to continue reading or not. Interactive Fiction (IF) is a computer-medium narrative that allows the user to interact with the story. The roots of IF are in *choose your own adventure* series of books that started in 1979. Unlike traditional books, the reader has to make a choice at the end of each chapter, skip to the chapter based on her choice and thereby influence how the plot unfolds. Using a computer, we can create truly interactive and much more complex stories. By changing the level of user interaction, we can create systems that are more like *games* where the user has to constantly make decisions or closer to the original choose your own adventure series.

Interactive fiction involves creating interesting and consistent variations of the story. It is not just a matter of inserting random variations at critical junctures in the story (Crawford 1999). Any variation must be dramatically interesting and consistent with the previous development of the story. Interactive fiction focuses on the plot as the main element. Both OZ Project (Bates 1992) and Hayes-Roth et al (1995) focus on character improvisations and often do not have an underlying story. DEFACTO by Sgouros (1999) uses a 'plot manager' to shape an interactive plot into the structure of Aristotelian drama by modelling the rising conflict between the characters following certain rules. Mateas and Stern's Façade (2000) is a short, one act interactive story that combines autonomous agents with plot generation; however, their focus is in strong integration of a single plot-line autonomous characters rather than branching storylines as often found in interactive fiction.

Our interest is in using IF for teaching about dynamic social systems by making the connections between choices and consequences clear. For example, an interactive system built around World War I, would allow the user to take on the role of a head of state, face the same decision those leaders faced at the time and see how the alternative history unfolds. This system could be used to present the inevitability of the war or how the war could be prevented or delayed through some key decisions. The system can be construed as a single-person system or a group exercise where different groups act as different leaders jointly determining the outcome of the events. Learning by doing, even when the activity is through a simulation, is often superior to route memorisation of facts (Schank & Cleary 1995).

Although the computer technology has progressed much beyond what it was when the original choose your own adventure books came out, tools for writers to create interactive fiction are severely limited or in most cases non-existent. Most tools such as, Hugo, Inform and Erasmatron (Crawford 1999) are intended for programmers. While a programmer will always be required for complex interactive systems, visual systems can provide entry points for writers and can provide a basic framework to expand. We present a Graphical Interactive Fiction Toolkit, GIFT, for creating interactive fiction. Authors can draw various objects in the story such as characters and locations, fill in the details of each object, and then generate the source code for the interactive fiction corresponding to the narrative elements

defined on the system. The aim of GIFT is to solve the problem of programming complexity by hiding the programming details of interactive fiction as much as possible from the authors. By hiding the complexity of coding the narrative elements, it allows non-programmers who have creative talent in narrative to create interactive fiction without the difficulties of programming.

Section 2 describes the architecture and implementation of GIFT; Section 3 discusses our preliminary evaluation, and in Section 4 we present our conclusions and future work.

2. GIFT

The main difficulty in creating interactive fiction is due to the high level of programming requirements. In comparison hypertext stories presented as a series of web pages has been blooming due to the availability of easy to use authoring tools. The current hypertext stories are adaptations of the original choose your own adventures to the web. While the main interaction remains the same, it has been possible to create much larger collaborative stories using hypertext. We describe the interface to GIFT, provide some screenshots to demonstrate how it is used by writers.

Writers interact with GIFT by creating a series of locations connected by paths using the location panel. The paths allow the user as well as the other autonomous characters to travel between locations. The set of items that the user can interact with is entered via the item panel and he set of characters entered via the character panel. GIFT generates source code based on the objects created which can then be modified and expanded by a programmer. We currently generate Inform source code as it is one of the more popular languages for creating interactive fiction and it has an extensive set of libraries, but GIFT could easily be modified to produce another language.
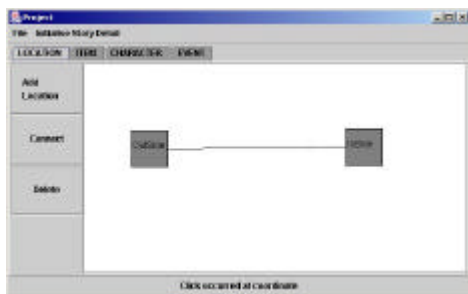

Figure 1: Location Panel

Figure 1 show the location panel with two locations. The locations can be double-clicked to modify their properties, such as name and description. The paths between locations provide the initial connectivity graph among locations and can further be modified to be one directional or to specify conditions, such as holding a key, as a prerequisite to travelling through paths. By default paths are crossed instantaneously as one leaves a room and enters another, but it is also possible to represent paths as a series of locations where alternative actions and even turning back might be possible.
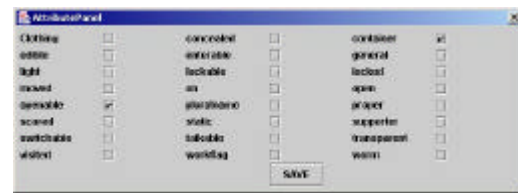

Figure 2: Object Properties

Objects can have a large number of properties specifying what types of interaction with them is possible. Figure 2 shows the 24 properties that are possible for *container* objects. A separate description for different conditions of the object can be added through the interface.
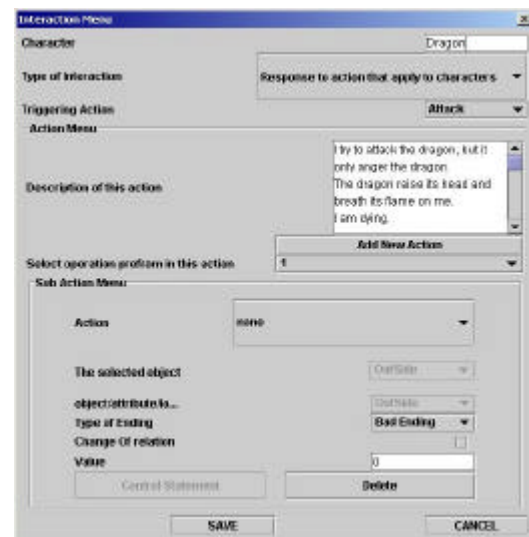

Figure 3: Interaction Menu

The interaction panel allows the writer to specify the interaction of a character inside the story. Interaction is critical to IF, as it provides the hook to branching storylines that the user can explore. The writer can assign an interaction to a character by connecting a character node and an interaction node, and defining the interaction further by

filling in details in the interaction menu (Figure 3). The interaction menu allows the write to specify the character that performs this interaction, the type of interaction, the action of the player that triggers this interaction, the text description of the interaction, and a number of background actions that affect the story world but are not noticeable to the player.

The event panel provides a similar interface to the interaction panel with the exception that event are generated by the system even when the user takes no action. The events provide a greater flexibility as they can be used to model the passing of time as well as background events taking place. User is immersed in a world that is alive, where inaction is a valid choice with its own consequences.

3. Evaluation

The aim GIFT is to hide the complexity of creating interactive fiction, so the writer can focus on the storytelling rather than the technology used for creating and delivering the story. We have completed a preliminary evaluation that will feed back into the design process of GIFT. Five users were asked to test GIFT. Two users had little programming experience, another two have some programming experience, and the final user was an experienced programmer. All users were interviewed separately focusing on 1) GIFT's ease of use; 2) overall system performance on assisting author to create the stories they wanted; and 3) whether they would choose GIFT for creating IF in the future.

All users found GIFT's user interface is self-explanatory and had no problems getting around the interface. In creating their own stories, users who were not familiar with the Inform language found the number of properties for each object and the choices for interactions to be overwhelming. This is not surprising as the properties were not documented beyond their text labels and the interaction among properties can be daunting even to experienced programmers. All but one of the users preferred GIFT to writing a story in Inform. The main reason given by the user who chose programming was the flexibility provided by having access to the full underlying language. All the users agreed that for anything other than very small stories using GIFT would be much quicker.

4. Conclusion

The computer provides a powerful medium for telling interactive stories. Until now, the lack of available tools has made it prohibitive for most educators to create interactive stories that can be used in teaching and learning. GIFT is a first step in creating flexible tools. While a programmer may still be needed for complex stories, GIFT makes it possible for educators to create simple stories through a graphical interface. The initial evaluation of GIFT has been positive and provided with feedback for improving the user interface. We will be expanding GIFT and using it to create actual and imaginary storylines around World War I for teaching high-school students.

References

Bates, J (1992). "The Nature of Characters in Interactive Worlds and The Oz Project", Technical Report CS92 -200, School of Computer Science, Carnegie Mellon University.

Crawford, C. (1999). "Assumptions underlying the Erasmatron interactive storytelling engine" in Proceedings of the AAAI Fall Symposium on Narrative Intelligence.

Hayes-Roth, B., Brownston, L, and Sincoff, E. (1995) "Directed Improvisation by Computer Characters" Stanford Knowledge Systems Laboratory Report KSL-95-04.

Hugo. An Interactive Fiction Design System, Retrieved November 10, 2003 from http://www.generalcoffee.com/hugo.html

Inform. A Design System for Interactive Fiction. Retrieved November 10, 2003 from http://www.inform-fiction.org/

Mateas, M. and Stern, A. (2000) "Towards Integrating Plot and Character for Interactive Drama" in *Working notes of the Social Intelligent Agents: The Human in the Loop Symposium*. AAAI Fall Symposium Series. Menlo Park, CA: AAAI Press. 2000.

Packard, E. (1979). The Cave of Time. Bantam Books.

Rousseau D., and Hayes-Roth, B. (1997) "Interacting with Personality-Rich Characters", Stanford Knowledge Systems Laboratory Report KSL-97-06.

Schank, R.and Cleary, C. (1995) Engines for Educators, Lawrence Erlbaum.

Sgouros, N. M. (1999). Dynamic generation, management and resolution of interactive plots. *Artificial Intelligence*, 107(1),.29-62.

TADS. Text Adventure Development System, Retrieved November 10, 2003 from http://www.tads.org/