Architecture of an Authoring System to Support the Creation of Interactive Contents

Kozi Miyazaki[1,2], Yurika Nagai[1], Anne-Gwenn Bosser[1], Ryohei Nakatsu[1,2]

[1]Kwansei Gakuin University, School of Science and Technology
2-1 Gakuen, Sanda 669-1337, Japan
miyazaki@nirvana.ne.jp, annegwenn@gmail.com, nakatsu@ksc.kwansei.ac.jp
[2]Nirvana Technology
Keihanna Plaza Lab Wing
1-7 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237, Japan
htt://www.nirvana.ne.jp/

Abstract. Since three-dimensional computer graphics (3D-CG) technology and interaction technology should be applied to e-learning as well as games, people must be able to easily create interactive contents based on 3D-CG, even if they are not 3D-CG professionals. In this paper, we propose a support system for creating interactive contents that runs on MS Windows and uses Direct X as the file format. By describing a script using two kinds of script files prepared by the system, a content creator can easily create 3D-CG scenes and can also control interactions between a user and the system. As an example of content creation, we present and explain interactive content in which users can enter the virtual world of classic Japanese tales and experience the story development of various types of classic tales for the first time.

1 Introduction

Recently, extremely rapid progress has been made in computer graphics (CG) technology for game machines and computers. This development has been particularly noticeable in the expression capability and special effects technology for three-dimensional (3D) computer graphics, and now it is finally becoming possible to generate images that look highly realistic. Because of this progress, it is now possible to virtually generate various kinds of objects in the real world, and such technologies are being widely applied to video games and simulations. At the same time, however, it is becoming increasingly difficult and time-consuming to develop/create various types of software or content related to CG. For example, to create new games that include 3D-CG, first we have to prepare a virtual 3D space, 3D computer characters, and rules of various actions for this 3D world. Such tasks require a lot of effort and time. In addition, this type of work needs a wide variety of in-depth knowledge, such as sophisticated programming skills and comprehensive understanding of calculation techniques for 3D space and modeling techniques for 3D computer characters. Therefore, in this area non-professionals face various barriers to develop entire systems from scratch.

Based on the above considerations, we want to simplify the development of games and/or contents using 3D-CG, especially for non-professionals. Focusing on the creation of contents using 3D-CG, we have begun to develop an authoring system whose most significant feature is that we have prepared and installed various types of 3D-CG-handling functions in it, and we have also made it possible for users to handle these functions easily using high-level commands. In our authoring system it is possible to handle a simple script language. As a result, even if users lack the specific skills required for 3D-CG creation, with this script language they can still create 3D-CG contents and 3D characters relatively easily. Furthermore, since this system will simplify contents creation, it can be applied to various fields of interest. For example, game creators could easily develop game prototypes by this system and carry out preliminary evaluations of the game.
In the case of e-learning, several platforms are already available, but the problem of how to create new contents suitable for e-learning remains. So far, most e-learning contents are character-based, and only a few include interactive functions; thus the creation of interactive contents promises to become crucial in the near future. One purpose of the system presented in this paper is to help contents creators of e-learning create interactive contents more easily.
In this paper we describe the authoring system we are studying and developing. First, we explain the system's concept and provide an overview of it, followed by a detailed description of its architecture. Finally, we show several examples of how users can develop interactive contents with it.

2 Related Work

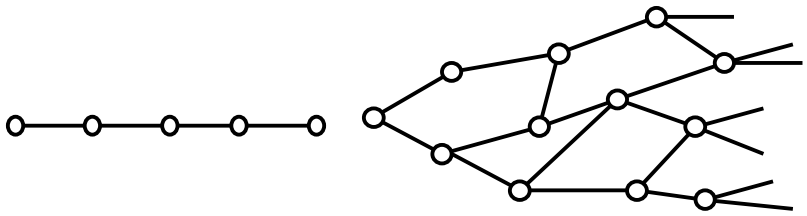
One representative interactive contents is Role Playing Games (RPGs), which is a genre of video games where users enjoy various story developments by controlling their characters. One of the most important issues for RPGs is how to control the development of interactive stories that have been actively studied under the name of "Interactive Storytelling" [1][2][3][4][5][6][7]. Although there are various kinds of small variations for story development, basically for RPGs the overall storyline is carefully designed beforehand and controlled to lead users through it. On the other hand, in the case of interactive storytelling, the key issues include the generation of autonomous interactive characters and also the emergence of storylines based on interactions between virtual characters and users and also among virtual characters. This is surely the final target of interactive storytelling. Society consists of people with autonomous capabilities, and based on interactions among people various everyday events occur: in other words, story. The problem with present interactive storytelling is that, since the generation of sophisticated autonomous characters is so complicated, it is difficult to maintain the consistency of the generated story for more than several minutes [8]. For games and e-learning users are expected to interact with the system for a certain duration, probably for hours. Therefore, for these applications, it is more realistic to develop a narrative based on a plot prepared beforehand. If a great amount of story variation must be generated, our idea is to generate a story by connecting various kinds of short stories called 'plots.' In this case there must be an

authoring system with which content creators, even non experts in computer graphics or animation techniques, could easily develop interactive contents. Based on this simple consideration, we started to develop an authoring system for the creation of interactive contents.

3 Concept and Overview of the Authoring System

3.1 Concept of Story Generation

Here the basic concept of story generation is described. A story consists of a concatenation of short stories called plots. In linear stories such as novels and movies, plots are connected based on fixed storylines. On the other hand, in interactive storytelling, a plot to be connected to a previously developed storyline is based on the interaction between users and the system (Fig. 1). If a large number of plots are prepared as a plot database and the system can watch the process of story development and extract an appropriate plot from the database, an immense number of story variations could be generated. The selection process of adequate plots and combining plots to maintain story consistency are of great importance and difficulty. In this paper we focus on the generation of storyline by connecting plots one by one, as we believe that this is the key function of an authoring system.



(a) Linear story (b) Interactive story

Fig. 1. Linear story vs. interactive story.

3.2 Concept of Authoring System

Our first prototype authoring system supplied two major benefits: one for contents viewers and one for contents creators. For contents viewers, they can enter a 3D world created by the system and walk through it. Also, they can experience various events in the 3D world through interactions with the 3D characters who inhabit it. For contents creators, by preparing 3D characters and describing scripts of the script file, they can create any kind of interactive contents or event contents that employ these 3D characters and 3D backgrounds.

We carefully considered and prepared the contents/software development environment of this system and adopted Windows as its basic OS and Microsoft Visual C++ for the software development. To generate 3D scenes and play music, we use Direct X and Direct X SDK (software development kit), respectively. The data file format for 3D models is the X file format normally used for Direct X. Since X file format is standard for Direct X, its generality makes it superior to other file formats.

3.3 System Construction

In our authoring system, there are two kinds of files: object (Object.txt) and script (Script.txt). The system generates interactive narrative by interpreting these files.
Objext.txt: A text file used for the basic definition and setting of 3D objects.
Script.txt: A text file that defines the generation of each scene, the actions of each character, and the interactions between users and the characters and also among characters.
Detailed explanations of the functions of these files will be described in section 4. In this section, the basic mechanism of story is described. The control of interactive story development is basically described and achieved by Script.txt. The file consists of a set of plot scripts each of which corresponds to a plot described in 3.1. Each plot script begins with START and concludes with END, as shown in Fig. 2, where PLOT X is the name of the plot script. Also utilizing GOTO mechanism, control is handed from one plot script to another based on interaction results. By utilizing this basic mechanism, the storyline can proceed from plot to plot, thus achieving interactive story generation.

```
START MAIN
GOSUB PLOT1
END MAIN
# Script of Plot1
START PLOT1

END PLOT1
# Script of Plot2
START PLOT2

END PLOT2
```

Fig. 2. Basic structure of “Script.txt.”

The system mainly consists of three modules: a main, an authoring, and a rendering for 3D rendering. Figure 3 illustrates the construction of the system. The main module manages the overall process and thus controls the entire application. The authoring module manages the script and the data flow. It also controls each independent object that corresponds to each character. Furthermore, that module manages interactions. The rendering module manages the information for the rendering process and also controls the creation of 3D models.

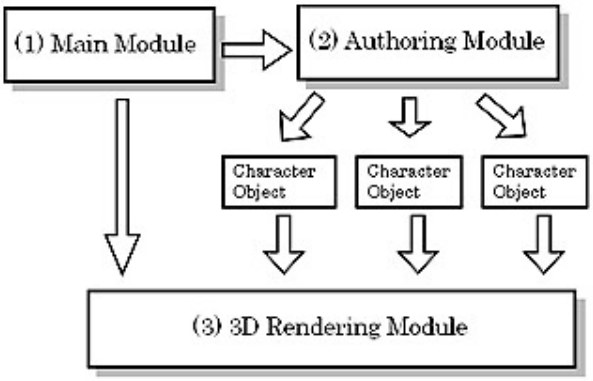


Fig. 3. System configuration.

4 Contents Creation

As described in section 3, in this authoring system, contents can be created using Object.txt and Script.txt. Now the details of the contents and functions of these files will be described.

4.1 Object.txt

```
// Countryside
BACKGLOUND_01 {
// House
LOAD_MESH( "cottage01.x" );
MESH_SCALE( 1.2 );

// Ground
LOAD_MESH( "ground01.x" );
MESH_ROTATION(    0.0,    0.0,
0.0 );
MESH_SCALE( 1.0 );
}
```

Fig. 4. A description sample of “Ojbect.txt.”

This file is used to define 3D objects. It can treat multiple X files by grouping them and treating them as one 3D object. For one X file data (mesh data), three types operations can be carried out: rotation, parallel shift, and size change. These operations can also be simultaneously performed for multiple X files. An example of the description of Object.txt is shown in Fig. 4.

In Fig. 4, X files corresponding to house and ground models are read from the data storage, and by integrating them, the background 3D object is defined. In the authoring system, this integrated object is named [Background01] and referred to at the time of script processing, 3D rendering, and so on.

4.2 Script.txt

This file is used to define the interactions, motions, and conversations of each character. Basically, a sequence of statements is interpreted and carried out according to its order. In addition to this basic control mechanism, just as for the functions of C language or the subroutines of BASIC language, it is possible to describe a script as a module and to call any module at any time. Furthermore, as one type of flow control, branch processing that uses [IF] can be carried out. When a script is described and separated by [START] and [END], it is treated as one module corresponding to a plot script, and a group of these modules controls all processing. [START MAIN] denotes an entry point, and processing starts from this module, corresponding to the "main ()" in C language. Also, several variables such as character strings, integers, and real numbers can be handled by this system, which can carry out simple numerical calculations among these variables. The character string after [#] is treated as a comment and is neither interpreted nor processed.

Table 1. Commands and their functions.

Statements	Function
GOSUB	Call another plot script.
GOTO	Jump to the specified plot script.
IF, ELSE, END_IF	Same function as C/C++
EYE_POSITION	Specify the eye position.
EYE_DIRECTION	Specify the eye direction.
RENDERING_ON	Show a 3D object.
RENDERING_OFF	Hide a 3D object.
INPUT	Input user's string.
MSG	Show a message.
STRING	Show a narration message
SET_BOUNDERING_RADIUS	Specify a radius for collision detection
SET_INTERACTION	Jump to the specified plot script based on the collision detection
MOVE	Move a 3D object according to the specified shift and rotation
ANIMATION_START	Start an animation of 3D object using the specified animation data

```
  START MAIN
  GOSUB INIT
  END MAIN
  # script of Interactive comm.
  START HUMAN_SCRIPT
  $count = $count + 1
  BACKGROUND_COLOR  128 0 0
  IF $count > 1 THEN
  @mes = $count
  @mes  =  @mes  +  "times  you
visited."
  MSG @mes
  ELSE
  MSG "Hello!(^o^)"
  END_IF
  WAIT 0.5
  END HUMAN_SCRIPT
```

Fig. 5. A description sample of "Script.txt."

Table 1 presents examples of basic commands and their functions. In the present version only a small number of basic functions have been prepared to achieve interactive contents. Future versions, however, will include many additional convenient functions and image effects.
GOSUB is a function to summon a specified plot script. On the other hand, GOTO jumps to that specified plot script. By combining these functions it is possible to describe story development based on the concatenation of plot script. Also it is possible to describe sub story development within a plot script. EYE_POSITION and EYE_DIRECTION are used to specify the position and direction of the eye's perspective or the camera. RENDERING_ON and RENDERING_OFF are used for showing a 3D object or hiding it from the display. INUT is used to obtain an input from a user. So far only key input is allowed but input based on speech/image recognition is planned. By combining the IF, GOSUB, and GO_TO functions, control can be handed to any plot script based on interaction results. STRING is a function to show a message identical to MSG. As STRING can display a message of various character sizes and colors, it is mostly used to display narration messages. MOVE is used to move a 3D object according to specified shift and rotation. ANIMATION_START is used to start animation using specified animation data. By combining these commands, any kind of animation for a 3D object is fairly easily achieved. Figure 5 illustrates an example of a description of Script.txt.

5. Example of Contents Creation

One significant characteristic of the proposed authoring system is that content creators, even schoolteachers or e-learning content creators who are not CG technique experts, can fairly easily create new interactive contents. Utilizing these contents, children can

be expected to experience the world of old tales or historical events. People used to learn how to do with other people, some of whom are not good guys, by listening to old tales told by their parents or grandparents. Also people learned the basic mechanisms of how our society is managed and how to adopt themselves to it. Therefore, we think that one good application of the authoring system is the generation of interactive tales. Below are the functions the system is expected to achieve.
*Function that generate classic tales as animation:
*Function that allows users to join the development of a tale story by interaction with the characters in the tale's world.
*Function to generate new tales based on a mixture of classic tales.
We carried out an experiment to achieve the above functions by selecting five representative classic Japanese tales. For these five stories, we carried out the following processes:
*Segmented each story into a set of short plots and generated a plot database
*Generated CG models necessary to generate scenes and objects for all plots
*Generated animation data necessary to generate actions and events that occur in each plot
The details of each process will be described below.

5.1 Plot Database Generation

Each of the five classic Japanese tales was analyzed and segmented into a series of short stories called plots. Each tale consisted of about 30 to 40 plots, and the total number of plots that consisted of five tales is 180. Then the description of each plot was abstracted. For example, "Momotaro left his home to kill monsters" became "A left B to do C." By doing this, it became possible to merge similar plots. Also this allows plots from two different tales to be combined. After combining similar plots based on this abstract process, the total number of plots was halved to 90. Each of these plots was then converted into a plot script including wide variations expected to occur based on interactions.

5.2 CG Database Generation

Then we investigated the number and types of CG models necessary to generate each of the plots in the plot database. There are basically two types of CG models to be prepared. One is characters, animals, and sometimes static objects that are expected to interact with users. The other type is static objects that do not interact with users and that are considered background. By combining similar objects and backgrounds, the total number of CG models is reduced to one third of the original numbers. In the experiments carried out here, we only prepared the fewest number of CG models. In such real applications as education software, however, it would be preferable to prepare as many variations as possible for one CG object to increase the software's appeal.

5.3 Evaluation of authoring system

After the above preparation processes, it is necessary to carry out story generation. As each plot is expressed abstractly, each user is asked to make correspondences between each character, object, and CG model. The following system function was confirmed in experiments.

Generation of animation
It was confirmed that all five classic Japanese tales were generated as an animated movie by concatenating plots according to the original story. It was found that adding a small variation to the animation gave users a fresh feeling each time they observed the animation. Figure 6 illustrates one variation created by the system.

Interactive story development
Our system still lacks a function that automatically selects a plot to be connected to the already developed storyline. This time, therefore, we carefully prepared various plot concatenations beforehand, and by monitoring story development, decided which plot to select. Automatic plot selection and connection is a future research theme. At the same time real-time manual development of interactive story, as in the movie "Truman Show," is also an interesting research theme.

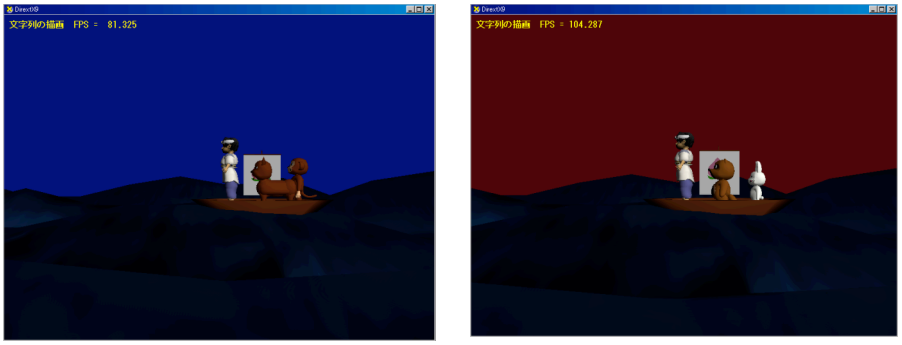


Fig. 6. An example of a scene generation.

6 Conclusion

In this paper we proposed an authoring system to help people create new contents and described the details of the system's construction and functions. Although in such applications as games and e-learning, various types of 3D interactive contents are required, to date the creation of interactive contents continues to consume much time and effort. The system we have proposed and developed is the first prototype with the potential to reduce the time and labor required for producing interactive content. As a

result, even users who lack sufficient 3D-CG knowledge or techniques will be able to create 3D-CG interactive contents using this system.
This system features potential for use in a variety of applications. In history, for example, education using only textbooks is often boring and difficult to provide students with a realistic sense of historical events. This system, however, will enable teachers themselves to create interactive contents for history lessons. Another merit of this system is that, since it adopts Direct X as a computer graphics module, we can use the latest 3D rendering environment.
Several issues require further study. The first is the addition of input devices, visual effects, and sound effects. In the present version, interaction between computer characters and users only operates by using character input/output. To make the system more realistic, it is desirable to adopt speech recognition/synthesis for interaction between computer characters and users. Also, utilizing a camera image of a user, carrying out image processing, and then inputting that image processing result into the system would improve the user interface. Furthermore, to make the interaction in virtual space more immersive, various image and sound effects must be added to the system. It is also necessary to evaluate the system. We are planning to create several historical interactive contents that allow user interaction with the system and obtain feedback using questionnaires. Such evaluation would clarify the weaknesses of our system.

References

1. Swartout, W. et al., 2001. Toward the Holodeck: Integrating Graphics, Sound, Character and Story. Proceedings of the Autonomous Agents 2001 Conference.
2. Hayes-Roth, B., van Gent, R. and Huber, D., 1997. Acting in Character. In R. Trappel and P. Petta (Eds.), Creating Personalities for Synthetic Actors. Berlin, New York, Springer.
3. Bickmore, T., and Cassell, J., 1999. Small Talk and Conversational Storytelling in Embodied Interface Agents. AAAI Fall Symposium. “Narrarive Intelligence,” November 5-7, Cape Cod, MA.
4. Mateas, M. and Stern, A., 2000. Socially Intelligent Agents: The Human in the Loop. AAAI Fall Symposium.
5. Young, R.M., 2000. Creating Interactive Narrative Structures: The Potential for AI Approaches. AAAI Spring Symposium in Artificial Intelligence and Interactive Entertainment, AAAI Press.
6. Young, R.M., 2001. An Overview of the Mimesis Architecture: Integrating Narrative Control into a Gaming Environment. AAAI Spring Symposium in Artificial Intelligence and Interactive Entertainment, AAAI Press.
7. Cavazza, M. Charles, F. and Mead, S.J., 2001. AI-based Animation for Interactive Storytelling. Proceedings of Computer Animation, IEEE Computer Society Press.
8. Charles, F. and Cavazza, M., 2004. Exploring Scalability of Character-based Storytelling. Proceedings of ACM AAMAS’04.