Prototyping Tool for Web-Based Multiuser Online Role-Playing Game

Shusuke OKAMOTO^{†a)}, Masaru KAMADA^{††}, and Tatsuhiro YONEKURA^{††}, Members

SUMMARY This letter proposes a prototyping tool for Web-based Multiuser Online Role-Playing Game (MORPG). The design goal is to make this tool simple and powerful. The tool is comprised of a GUI editor, a translator and a runtime environment. The GUI editor is used to edit state-transition diagrams, each of which defines the behavior of the fictional characters. The state-transition diagrams are translated into C program codes, which plays the role of a game engine in RPG system. The runtime environment includes PHP, JavaScript with Ajax and HTML. So the prototype system can be played on the usual Web browser, such as Firefox, Safari and IE. On a click or key press by a player, the Web browser sends it to the Web server to reflect its consequence on the screens which other players are looking at. Prospected users of this tool include programming novices and schoolchildren. The knowledge or skill of any specific programming languages is not required to create state-transition diagrams. Its structure is not only suitable for the definition of a character behavior but also intuitive to help novices understand. Therefore, the users can easily create Web-based MORPG system with the tool.

key words: MORPG, interactive animation, state-transition diagram

1. Introduction

Islay [1]–[4] is a programming tool for authoring an interactive animation. Our tool employs the modern paradigm of the object-oriented modeling and the classical state-transition diagram. An animation made by the tool is constructed as a collection of characters. Behavior of each character is specified by a state-transition diagram and pictures to express its appearance. The tool consists of GUI editors for diagrams and pictures, a generator for animation files which are written in a script language, and an interpreter for the animation files, as well as a translator for animation files into target codes. Each character moves about on the animation area. The animation is a simple video game when the area corresponds with a computer window for an animation viewer.

Because Islay is based on an object-oriented model, it can be naturally extended to a multiuser online version by introducing a mechanism to share the object state over the network. It also becomes a Web-based animation which may be used as a network communication tool, when the area is shared by the screens on the several Web browsers. This Web mode is supported by the extended translator.

Manuscript received September 8, 2007.

Manuscript revised November 29, 2007.

[†]The author is with the Faculty of Science and Technology, Seikei University, Musashino-shi, 180–8633 Japan.

^{††}The authors are with the Faculty of Engineering, Ibaraki University, Hitachi-shi, 316–8511 Japan.

a) E-mail: okam@st.seikei.ac.jp

DOI: 10.1093/ietisy/e91-d.6.1700

Once a viewer makes clicking or keying, the viewer's input is immediately sent to the Web server. And the input is reflected in the other user's screen.

As far as the definition of a character's behavior concerned, there is little difference between standalone and Web mode. But the Web mode system has a quite different capability that offers an animation with mutual user interaction, which can be indirectly done through animation characters. So an animation author can easily create Web-based interactive animations. The importance of Web application has been increasing because PC and the Internet connection became daily necessities. This prototyping tool will invite a variety of people including programming novices and schoolchildren to author their own MORPG.

2. Related Work

In [5], Huang, Fang-Tsou, and Chang discussed the 3D application for supporting the interaction of multiple participants in the Web environment. They implemented the software which runs on a Windows NT machine with an embedded Web server. In [6], Noser and Daniel proposed a rule-based animation system, which creates a real-time-structured virtual environment that both high-level autonomous humanoids and interactive users can easily share. In [7], Liu and Li discussed the design of the message protocol in multi-user virtual environment system as well as the design of the animation functions. In [8], Celentano, Nodari and Pittarello discussed an interaction adaptivity in 3D worlds. They try to anticipate the user behaviors by monitoring their interaction patterns. In [9], Boukerche, Duarte and Araujo proposed a description language based on XML for expressing VE. It can be used for the extension of VE applications, such as 3D virtual shops, customized training courses, educational and entertainment games. In [10], Presser proposed a Java Web application that enables users to explore and annotate VRML worlds collaboratively. On this system, the clients are required only to have a Web browser and a working VRML browser, and Java is not required on the client end.

3. GUI Editor

Figure 1 shows the main window to edit state-transition diagrams which are treated as classes in object-oriented programming. Each state has a picture which represents the appearance of the character at the state, and it also has several

Fig. 1 Main window of GUI editor.

Fig. 2 Attributes setting of a state.

attributes to take action. Figure 2 shows the dialog window for setting the attributes of the state. On this example, the state named 'right' has the action to move relatively right by five units. Available other actions are moving relatively, jumping to a specified position, sending a message, creating other characters, and so on. A transition from a state to another state takes place in the specified time interval. The action of a state is taken at each time interval.

If a state has more than one outgoing transition arrows, the transition from the state is conditional. Each arrow has an attribute that corresponds to an event. The events which happen to a character include the mouse clicking, the key pressing, the collision with the wall, the collision with other characters, the receipt of the specified message, and so on.

GUI editor generates an animation file. It is a text-based animation file which is written in a custom script language. It consists of pixel data of the pictures and three kind of definitions, which are the definitions of picture lists, state-transitions, and character groups.

An animation file can be played directly as an animation using the interpreter. And it can be also converted to a Java program, a JavaScript program [1], SWF Flash animation [2], or a program written in C language (for Windows, Gameboy-Advance [3], BREW-based mobile phone [4]).

4. Web-Based MORPG Sample

The RPG user can operate its own avatar on this game. An

Fig. 3 Screenshot of login window.

Fig. 4 Screenshot of a simple MORPG.

avatar can move about on the animation area. It meets other user's avatars and Non-Player Characters (NPC). And it picks up some items. The behaviors of NPCs are defined by state-transition diagrams. It is also expressed by state-transition diagrams what kind of belongings the avatar has.

Figures 3 and 4 show the screenshot of Firefox. Figure 3 is the login window. The user inputs the handle name and password, and he or she can also select the appearance of the avatar. After making a click on the start button, the game screen appears on the same window. It consists of the chatting field, the animation area and the avatar status field.

The new message text with handle name is displayed from the top of the chatting field. In addition to the mes-

sages from other users, the message from an NPC is also displayed.

The own avatar is always displayed at the center of the animation area. It can be operated with the mouse or the arrow key. A user's mouse inputs can also influence the status of an NPC. The information about the avatar movement and the clicking on an NPC are immediately sent to the Web server. The animation area is updated periodically so that the browser screen of the other users are also updated accordingly. These are implemented with the Ajax technology.

The avatar status field at the bottom displays some icon pictures which indicate the avatar's belongings. Each of belongings is associated with a state-transition diagram. An Islay's animation character can interact with others using the function of sending a message. Their behavior can be varied according to the received message. The PRG user can affect an NPC by making his avatar collide with the NPC or by clicking. When the status of the NPC is changed, the action is taken. If the action is to send a message to the avatar's belongings, the statuses of the belongings are changed.

The RPG author creates the state-transition diagram for avatar's belongings and NPCs, as well as the relationship between them. And the author can replace the appearances of avatars by preparing icon pictures. The Web-based MORPG is built through those steps.

5. Implementation

Figure 5 shows the relationship between the animation area and the display area. The animation area is the total range where avatars can move about. It is also the background picture which is partitioned into several square regions. The JavaScript code on the Web browser requests the Web server to send only the regions of the background picture to be displayed.

The system structure is shown in Fig. 6. It consists of Web server, DB server and the game engine. Its software architecture is the Model-View-Controller (MVC).

The animation file, which is made from the state-transition diagrams, is converted to the game engine codes

Fig. 5 Animation area and display area.

written in C language. This game engine operates as the model part. It accesses the data stored in the DB server by SQL codes, and it updates the statuses of avatars and NPCs periodically.

The view part is an HTML page, which includes JavaScript with Ajax. The controller part is the PHP code on the Web server. When a Web browser accesses to the server according to user input, the PHP code responds to it. The Web server manages the pictures of the characters. The DB server manages the locations of avatars and NPCs, the inputs from viewers, and the message texts.

Because of using SQL accesses, the PHP code, which is the interface for the browser, and the engine code can run on the same host machine, or they can run on the individual machines. The possibility of this choice is an advantage of MVC, and it is an important factor for balancing the load.

The stored information of avatars and NPCs are returned to a Web browser as a result of the response of the PHP code on the Web server. The JavaScript codes on the browser analyzes the information and displays the characters at the appropriate positions. Figure 7 shows the HTML <DIV> tag which is dynamically structured by the JavaScript codes. The callback function for clicking can be set on the attribute of the <DIV> tag. So the clickable character can be made on the viewer's screen.

Fig. 6 System structure of Web-based animation.

```
<div id="g1" style="position:absolute;
top:70px;left:10px" onClick="click()">
  <br>name
</div>
```

Fig. 7 Generated HTML DIV tag for a character.

6. Consideration

We have roughly examined the runtime performance of the resulting game. Several Linux PCs (AMD Athlon 64 X2, 1 GB memory) on Fast Ethernet networks are used as a server and clients machines. The game engine updates the state of ten NPCs once every three seconds. On the client side, the avatar performs the moving action once a second, which updates the database on the server, and the Web browser performs its screen update request once every three seconds, which reads the coordinates for the other avatars and NPCs from the database. The retry rates of the database transactions were 2.5% for hosting two users, 4.0% for four users, 10.0% for six users, 11.1% for eight users and 12.5% for ten users. All retries concern the transactions by game engine (C program code). No retries were caused by requests from the clients (PHP code).

It is considered that the maximum number of concurrent users is under ten users on this experimental setting. The operation of updating screen just reads the database, but it interferes with the write operation by the game engine. It seems that the write operation for an avatar move did not lead to retries since its database transaction consists of a single write, and it is done in the auto commit mode. A retry is due to failure in acquiring the exclusive lock of the database. A standard method for improvement is to divide a game field and save in a separate database. However, it may lead to changing the semantics of the game, since the updates of all NPCs are not carried out at once.

In a real-time multi-user RPG, network delay also causes a problem of response failure. In order to cope with this problem, it is possible to apply the dead-reckoning technique on the both sides of client and server. In our tool, this has not been implemented yet.

7. Conclusions

A prototyping tool for Web-based Multiuser Online RPG was reported. The Web-DB application is attractive in the modern Internet environment. However, to implement this kind of application, various knowledges such as SQL databases, Web servers, the JavaScript technique for cross-browser are needed. This disturbs the software develop-

ment. By using our tool, even the person who does not know those technologies can create a Web-based MORPG.

In this paper, only simple RPG was presented. Although our interactive animation is 2D, it can be used in various fields, such as the e-learning systems and the communication systems. Moreover, if synchronized with the other devices, the business system such as the shuttle bus pursuit system can be easily implemented.

Acknowledgments

This work was partially supported by the JSPS Grant-In-Aid no.18300027.

References

- [1] S. Okamoto, M. Kamada, and T. Nakao, "Proposal of an interactive animation authoring tool based on state-transition diagrams," *IPSJ Trans. Programming*, vol.45, SIG 1(PRO24), pp.19–27, 2005.
- [2] M. Nakagawa, S. Okamoto, M. Kamada, and T. Yonekura, "Flash movie authoring environment based on state diagram," 5th Workshop on Network & System Support for Games 2006, 2006.
- [3] M. Yamamoto, T. Yonekura, and S. Okamoto, "GBA code generation technique for the tool that creates interactive animations," *IPSJ Symposium Series*, 4, vol.2005, pp.29–30, 2005.
- [4] K. Shindome, S. Okamoto, M. Niibori, M. Kamada, and T. Yonekura, "Game engine on BREW," *IEICE Technical Report*, CW-6, 2007.
- [5] J.Y. Huang, C.T. Fang-Tsou, and J.L. Chang, "A multiuser 3d web browsing system," *IEEE Internet Comput.*, vol.2, no.5, pp.70–79, 1998.
- [6] H. Noser and D. Thalmann, "A rule-based interactive behavioral animation system for humanoids," *IEEE Trans. Vis. Comput. Graphics*, vol.5, no.4, pp.281–307, 1999.
- [7] Y.L. Liu and T.Y. Li, "A multi-user virtual environment system with extensible animations," *Proc. Eighth International Conference on 3D Web Technology*, pp.129–134, 2003.
- [8] A. Celentano, M. Nodari, and F. Pittarello, "Adaptive interaction in web3d virtual worlds," *Proc. Ninth International Conference on 3D Web Technology*, pp.41–50, 2004.
- [9] A. Boukerche, D.D. Duarte, and R.B. Araujo, "A language for building and extending 3d virtual web-based environments," *Proc. Web-Media & LA-Web 2004 Joint Conference 10th Brazilian Symposium on Multimedia and the Web 2nd Latin American Web Congress*, pp.114–116, 2004.
- [10] C.G.M. Presser, "A java web application for allowing multiuser collaboration and exploration of existing vml worlds," *Proc. Tenth International Conference on 3D Web Technology*, pp.85–92, 2005.