Automatic Storytelling in Comics: A Case Study on World of Warcraft

Chia-Jung Chan
Academia Sinica
Institute of Information Science
128 Academia Road, Section 2
Nankang, Taipei 115, Taiwan
apple15.c@gmail.com

Ruck Thawonmas
Ritsumeikan University
Department of Human and Computer Intelligence
1-1-1, Noji-higashi, Kusatsu
Shiga 525-8577, Japan
ruck@ci.ritsumei.ac.jp

Kuan-Ta Chen
Academia Sinica
Institute of Information Science
128 Academia Road, Section 2
Nankang, Taipei 115, Taiwan
ktchen@iis.sinica.edu.tw

Abstract
This paper presents a development of our comic generation system that automatically summarizes players' actions and interactions in the virtual world. The feature of the system is that it analyzes the log and screenshots of a game, decides which events are important and memorable, and then generates comics in a fully automatic manner. Also, the interface of our system allows users to customize their own comics. As a result, users can easily use the system to share their stories and create individual comics for archival purposes or storytelling.

Introduction
Online games are booming, as they enable people to entertain and fulfill themselves in a virtual world. In a massive multiplayer online role playing game (MMORPG), many players not only participate in the game world day and night, but also share and preserve their gaming adventures with others via blogs and forums. Currently, media such as video, screenshots, and text are the most popular formats used by players to preserve and share their gaming experiences. However, these media have some drawbacks.

The video format, for example, is not only storage-consuming but also a weak form for narration because
it records every detail about a player’s journey during a certain period. The video clips can certainly be made more appealing by film editing such as Machinima [1], but the task requires a great deal of effort by the publishers; therefore, the video format is unsuitable for daily use. On the other hand, screenshots generally do not contain sufficient context information about the player’s current status, so they are not vivid enough to narrate a story. The screenshots can be realigned into comics, which lighten storage requirements and are a pleasing way for sharing experience on the Internet. The comic strips not only provide a condensed view of a long story in a few pages, but also identify important frames by using different sizes, slants and display regions on a page. Additionally, comics can invoke rich visual expressions of vocabulary, emotions, and situations to convey a long story effectively. However, producing comics also takes a lot of time and effort. Therefore, we propose a system that utilizes the comic format to summarize and preserve a player’s notable game experiences for storytelling.

Developing an automatic comic generation system raises many challenging problems. Technically, our major concerns are how to choose important frames and create the comic layout with adequate word balloons and onomatopoeic sounds. From the user’s perspective, we would like the system to generate comics in a fully automatic manner but still we encourage users to be involved in the creation process. Those are the challenges we have in this study, and we will show our solution in this paper.

The contributions of this study are as follows. a) Our system is the first work that renders storytelling comics from games without modifying the games’ core engine. Therefore, our approach can be applied to any other real life commercial games. b) We designed an efficient comic layout algorithm and provide a user interface with interactive editing functions for users to create and customize their own materials into comics.

In the following sections, we first discuss the development and preliminary implementation of our comic generation system. We then present our case study of World of Warcraft (WoW) and the user survey of our system. The interface design is refined based on the feedback of the user survey we conducted in order to make the system more useful and enjoyable.

**Related Work**
Some studies have used the comic format to summarize online communications [2], video content [3, 4], and game events [5, 6]. In the field of computer graphics and entertainment technology, a few systems have been developed to produce comic-style summarizations automatically. For example, Shamir’s system [5] generates comics from a 3D interactive computer game, *Doom*. The drawback of most existing approaches is that the games’ core engine must be modified in order to record game events. Thus, the general public cannot play with the gadgets.

**Methodology**
Figure 1 shows an overview of the approach to our prototype. The methodology comprises the following stages. First, a set of game events and screenshots are collected during game play. Then, an interface generates comics from the collected materials. Screenshots are selected based on the importance and frequency of events and the images are rendered into comic pages using a layout optimization procedure.
Game Logs and Screenshots
To generate comics automatically in our system, a log file is created and screenshots must be prepared in advance. Many games provide mechanisms to monitor changes in a player's status and actions, and record the game events and screenshots considered to be important during game play. Status changes and interactions, such as chatting, combat, looting, zone changes, experience point changes, and trade between two players, are regarded as possible significant events and are thereby recorded. At the same time, screenshots of the game are recorded a) when a potentially significant event occurs and b) once per \( i \) seconds. The interval of \( i \) depends on how specific a user would like to be when recording and editing their game session using our system.

Frame Selection
To produce a concise summary, only the most important screenshot images are used for comic generation. In the generation process, the number of pages in the comic can be specified by users. When the player assigns the number of pages and clicks the generation button, the frame selection steps proceed as follows: a) the total number \( n \) of images required for the desired comic is estimated; b) the significance scores of all images are evaluated; and c) the estimated \( n \) images are selected from the image pool.

In the first step, we introduce a random variable into the estimation process. Thus, users can change the results of the generated comics with different layouts simply by clicking a button. Given the number of pages or \( \text{numPage} \), the total number of images or \( \text{numImgs} \) is calculated by

\[
\text{numImgs} = \text{randImgPP} \times \text{numPages},
\]

where the number of images per page \( \text{randImgPP} \) follows a normal distribution with a mean equal to 5 and standard deviation equal to 1 in order to improve the layout's appearance.

Next, the significance score of an image is evaluated. Let \( \text{sinImg} \) be an image's significance score and \( n \) be the number of event types. For an event type \( k \), let \( c_k \) denote its frequency, and \( w_k \) be the specified weight (importance) of \( k \). The values of the weights are initially given; however, they can also be changed by users through our interface. Then, the significance score of an image occurring at timestamp \( i \) is calculated by

\[
\text{sinImg}_i = \sum_{k=1}^{n} (c_k \times w_k).
\]

All the images are then sorted in decreasing order of their significance scores. Finally, \( \text{numImgs} \) images are selected from the top of the sorted image list for layout computation in the next step.

Layout Computation
After selecting the significant frames, we compute how to place them on the comic pages in two steps. First, we pack several images into a group, and decide their placement on the pages. Second, we decide each image's shape and size on a comic page based on its importance.

We design an algorithm to pack the images into groups, where the number of groups is equal to the number of pages specified by the user. Initially, the selected images are divided into page groups based on their significance scores in chronological order. Then, on each page, the algorithm arranges the appropriate image group into several rows based on their scores.
Once a page has been generated, the image set of the page, the positions and the sizes of the images on the page are fixed (see Figure 2).

Since the presentation of the comics is laid out in a 2D space, images that have been grouped on a page are placed into blocks in either column or row order. In our system, the row order approach is used in the layout procedure. To avoid packing too many images on a page, a variable is added to constrain the number of images displayed in a row. If there are too many images in a row, the process of inserting an additional row is taken and repeated until the number of images in each row satisfies the constraint.

We define a region as an image's shape and size on a page. To create variety and visual richness, regions are randomly reshaped with slants on their edges so that the images look interesting on the comic pages. After deciding the placements of the selected images, we calculate the dimensions and regions of the images based on their significance scores. Images with higher significance scores are assigned larger areas on a page; conversely, less significant images cover smaller areas.

**Rendering**
To create the appearance and feeling of a comic-book, we employ a three-layer scheme to render an image on a page. The layers are:

- the image
- the mask of the image
- the word balloons and sound effects (if any)

An example of the three-layer scheme is illustrated in Figure 3. An image on the bottom layer is processed and placed on the panel, which is the area where the image is to be placed on the comic page. An edge detection technique and cartoon-like filters are applied to the image to emulate a comic style. The image is then resized to fit the region and drawn with its center aligned on the panel. Next, a mask layer is placed over the bottom layer to crop an image's region; that is, any drawing outside the region is ignored. Finally, embellishments such as word balloons and sound effects are placed on the top layer to enrich expressions in the comic's text. In particular, by the help of the detection, we try to put the word balloons where no main characters are placed.

**A Case Study on World of Warcraft**
In the prototype, we chose World of Warcraft (WoW), an MMORPG, as the testing platform for our system for three reasons. a) WoW is the most prevalent MMORPG worldwide. According to a report published by Blizzard, the company that created WoW, the game had more than 11.5 million subscribers [7] by late 2008. b) The
A player’s interactions in the game are archived as a log file and screenshots. They are then loaded into the interface. In the Log section, users can open the original log text file and make some changes. In the Image section, users can set their own preferences for use in the later comic generation process.

Figure 4. The main window of the automatic comic generation system. In this interface, the log file is optional. If a user does not have a log file, the system will randomly assign a significance score for each image and render comic pages without text.

WoW Players tend to share their gaming experiences with each other in both real life and virtual communities. For example, stories, such as record breaking events or the victory of a team of players over an entrenched arch enemy, are often posted on weblogs. c) WoW provides a sophisticated game log scheme. Blizzard publishes a comprehensive set of game APIs that allow users to record every game event through a WoW AddOn. Thus, we create a WoW Addon to script the game events and screenshots required for comic generation without modifying the WoW core engine.

Figure 5 shows an example of a WoW comic page generated by our system. The page, which represents a partial summary of a WoW player's game play, illustrates the diversity of region sizes and visual richness, such as the slants on edges of the regions. The system also retrieved the chat messages and combat logs that occurred as the game's screenshots were being recorded. As shown in the figure, the chat messages are displayed in the word balloons, and the sound effects of combat are added by our system to make the comics more interesting.

User Survey
We conducted prototype testing and a survey of six WoW players, both leisure and power players, to find out a) what they think about the system and b) how they would like to use the interface. The players are asked to use our system to generate comics of their
own game journey in WoW. Overall, all players think the gadget is appealing and user-friendly. Still, we received useful suggestions from some of the players:

“I think the frame selection mechanism still has room for improvement. Although the automatic idea is cool, but I wish I can contribute more in the comic creating process.”

“It would be better if the interface has more setting and editing flexibility. I hope I can cut and resize some scenes.”

The design of our system and the interface is refined based on user feedback. To adapt the system to reflect users’ preferences, we added new features to the interface so that it is more flexible and enjoyable. For example, users can define their own preferences to determine the importance of the events, and refine the log and screenshots in the later comic generation process. Figure 4 manifests the interface of our system.

Future Work
We plan to extend the current system in three ways. First, we will incorporate a better object detection technique to pinpoint the location and size of game characters in screenshots so that we can crop the comic book frames and put word balloons on frames accurately. Second, we are improving the layout computation algorithm to make the generated comics more similar to hand-drawn publications. Last but not least, we are refining the interface design and adding new features to meet users’ needs. We plan to release our system for public use in the near future, and make it a platform for experience sharing and storytelling among players in the virtual community.

Acknowledgements
The authors would like to thank the anonymous reviewers for their constructive comments. This work was supported in part by Taiwan E-learning and Digital Archives Programs (TELDAP) sponsored by the National Science Council of Taiwan under the grants NSC98-2631-001-011 and NSC98-2631-001-013. It was also supported in part by the National Science Council of Taiwan under the grant NSC97-2221-E-001-009.

Reference
[1] Machinima
http://www.machinima.com/channel/view&id=5
[7] Blizzard Entertainment