

AR Karaoke: Acting in Your Favorite Scenes

Maribeth Gandy¹, Blair MacIntyre², Peter Presti², Steven Dow¹, Jay Bolter³,
Brandon Yarbrough², Nigel O'Rear³

¹*Interactive Media Technology Center*, ²*College of Computing*,

³*School of Literature, Communication and Culture*

GVU Center, Georgia Institute of Technology

Atlanta, GA 30332, USA

*{maribeth, blair, steven}@cc.gatech.edu, peter@imtc.gatech.edu,
jay.bolter@lcc.gatech.edu, {gte750u, gte055u}@mail.gatech.edu*

Abstract

In this paper we present a concept for augmented reality entertainment, called AR Karaoke, where users perform their favorite dramatic scenes with virtual actors. AR Karaoke is the acting equivalent of traditional karaoke, where the goal is to facilitate an acting experience for the user that is entertaining for both the user and audience. Prototype implementations were created to evaluate various user interfaces and design approach reveal guidelines that are relevant to the design of mixed reality applications in the domains of gaming, performance, and entertainment.

1. Introduction

Karaoke is a popular form of entertainment where a performer sings live while accompanied by pre-recorded music, usually prompted with the lyrics on a video display. Karaoke is a social activity located in a restaurant, bar, or at a party and the songs are usually well known by the audience. The goals for the performer vary from wanting to execute the best vocal performance possible to amusing the audience with a comedic show. Some performers and audiences take karaoke very seriously, while for others it is simply a fun and casual group activity. In both cases the goal of karaoke is entertainment for both the performer and the audience.

In this paper, we describe AR Karaoke (ARK), a form of AR entertainment that seeks to do for acting what traditional karaoke does for singing; enable a participant to take on an existing role from a theatrical scene and perform it with virtual actors.

The challenges in creating our ARK prototypes did not lie in developing the systems themselves. The tracking technology and display hardware that currently exist are sufficient to create working systems. We envision ARK taking place on an empty stage; the graphics would be spatially located but not precisely registered with any physical objects. A self-contained video-mixed display with an integrated tracker, of the sort

that could be made with current components, is ideal for entertainment applications and alternatives to an HMD, such as the use of projection could also work.

Our DART prototyping environment [1] allowed us to rapidly build the prototypes using informal, sketched content. The linear nature of the experience constrained the software requirements, eliminating the need for complex components such as intelligent agents. The ease of creation and deployment allowed us to focus on the interesting research questions posed by ARK— interface approaches and experience design issues—that can inform a broad class of future AR entertainment applications. In the following sections we present two of the prototypes we implemented. A simple implementation of scene from *The Wizard of Oz* and a physically and mentally demanding sword fight from *The Princess Bride*.

2. Related Work

For years, researchers in virtual and augmented reality have been exploring the use of AR/VR technologies in the theatrical arts and performance. Applications range from the use of the virtual world as a stage for a distributed performance of Shakespeare's *A Midsummer Night's Dream* in 1998 [2] to a VR system that allowed actors in remote locations to rehearse with each other [4] to a virtual kabuki theater, where the user is visually changed by the use of different avatars [3].

However, a technologically simple approach to karaoke for famous movie scenes is currently in practice in bars and restaurants in New York City. This *Movieoke* system¹ projects the DVD of a film on a screen while the performer stands on the stage in front and receives prompts from the closed captioning text included on the DVD. Users of this system express a desire to be “in” the scene, rather than in front of it.

To inform our design choices for the ARK prototypes, we analyzed the experience and interface design of a class

¹ www.movieoke.net

of successful performance or “rhythm” games (e.g. *Dance Dance Revolution* (DDR)), isolating the key components of these systems as they apply to ARK. These features include support for improvisation by the performer, replayability (easy to play hard to master), simplified interfaces that still allow for artistic expression, and feedback from not only the system but from the audience (formal or informal) as well.

3. Prototype 1: The Wizard of Oz

Our first prototype (shown in Figure 1) is based on a scene from the classic movie *The Wizard of Oz* (WOz). The user performs the role of the Cowardly Lion during his first meeting with Dorothy, the Tin Man, and the Scarecrow. The goal of this prototype was to create a proof-of-concept version that would begin to shed light on the interface and development issues. Rough sketch-based animatics² were created to represent the virtual actors, and the voices from members of our research team were used for the virtual characters.

We used a simple form of prompting in this prototype with textual stage directions for this dialog driven scene. When the user is supposed to speak, the lines appear at the bottom of the screen, along with stage directions. When it is time for the user to be done speaking the text simply disappears.

4. Prototype 2: The Princess Bride

The goal of the second prototype (shown in Figure 2) was to explore various approaches to stage direction and dialog pacing by using a scene with fast-paced dialog and choreographed action. We also wanted a “difficult” scene that a user might not master on the first try, which would have the flavor of a rhythm game. For this prototype we chose a scene from the 1987 film *The Princess Bride* (PB). This movie is a cult classic; the type of film where



Figure 1. The Wizard of Oz. (a) What the participant sees: animatic characters, movement and performance directions, and her lines. (b) She offers to fight “with one hand tied behind my back ... on one foot.”

² An animatic is a sequence of images, typically sketches, synchronized with audio. Animatics are used as a pre-visualization technique in film.

many people know the lines and enjoy quoting them. In this scene, the characters are meeting for the first time and engage in both humorous rapid dialog and a dramatic sword fight.

We used this experience to test four interface components for the heads-up-display (HUD): a *text display*, a *progress bar*, an *overhead map*, and a *direction display*. Each of these components is discussed further in the implementation section.

This prototype uses two *direction displays*. The first (the blue arrow on the left in 2(a)) shows movement commands (turn around, move forward etc.). The second (the green shield in 2(a)) is for prompting sword movements. Animated icons are used for both types of prompts. Not only does the animation draw the user’s attention, but also illustrates the required movement.

The user also has the choice of referring to the overhead map (shown on the right of 2(b)) or looking down at the floor to see where she needs to move (2(b)). We use this *ghost actor* to encapsulate the location of the “intended” actor location, and use its location to drive these two interface elements.

5. Observations and User Feedback

The process of implementing and testing these prototypes revealed guidelines and open issues that are relevant to the design of mixed reality applications in the domains of gaming, performance, and entertainment.

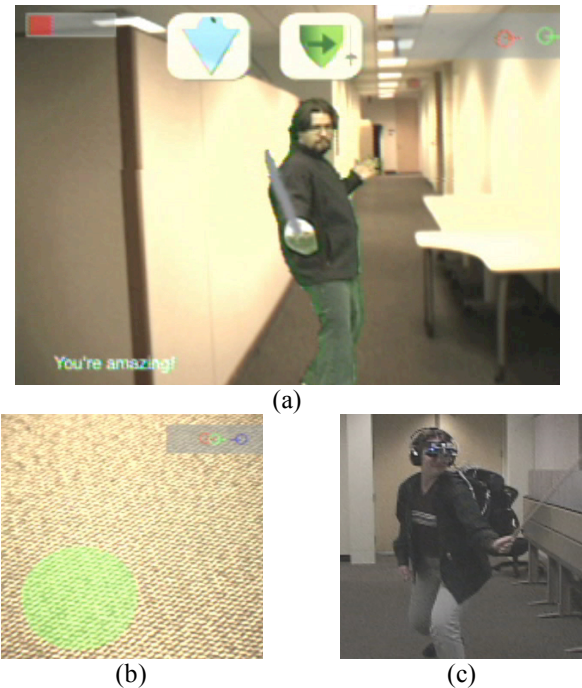


Figure 2. The Princess Bride. (a) What the participant sees: image-based animatic characters, movement and fighting icons, map showing her where she should be, and her lines with a progress bar. (b) Spot on the floor where she should be. (c) She is parrying to the left.

5.1. User Reactions

The feedback from those who “performed” during informal demonstrations of the prototypes has been very positive. The experience is engaging; users leapt from their chairs, invented grandiose fencing moves, and growled ferociously. Users were quick to grasp the “it’s like karaoke for acting” concept and understood the basic interface with little explanation. However, the biggest problem lies with the reading of the lines. Users universally have problems seeing the text, either because of the size (text was made small so as not to occlude too much of their view) or the color (text that is visible against a dark carpet is hard to read against a white wall).

Users also tended to be tentative in their movements, inhibited by the backpack with the laptop to run the system and the HMD, tracker, and headphones on their heads. They did tend to become more gregarious as they become acclimated to the equipment, however.

Our expectation that the PB scene would require replay was confirmed. The users tended to get behind in the action, become confused, move incorrectly (such as dropping their “sword” when tossing it from one hand to another), speak a line incorrectly, and so on. However, both the users and the audience enjoyed the experience, as we expected. These challenges are what make mastering such experiences enjoyable. Overall, this prototype showed us that it is possible to create entertaining ARK content using physically oriented scenes.

Our informal evaluation has not yet explored different kinds of audience experiences. Although there are always other people watching the “performance” and they clearly are entertained, we have not tested this experience in the actual type of setting we envision for ARK (e.g. a social setting like a party or bar). While they are able to see the live performer on the stage, a variety of techniques could be used to show the audience the virtual part of the scene.

5.2. Direction: Dialog

It was clear from the WOz scene that simply displaying text when it was time to deliver a line was not sufficient. The user had no warning when a line would appear, and no feedback on how long she had to say the displayed line. Therefore, we decided to provide pacing cues to the user, settling on a progress bar to indicate how long the user had to say a particular line. In the PB scene, the progress bar proved very helpful to the user, although when the lines were short and the give and take between the characters was rapid (e.g. “Thank You”, “You’re Welcome”) it was less effective.

A related problem is that, in some cases, the lines for the user were too long, filled too much of the screen, or were hard to read due to their color. Adopting a scrolling display where all text is displayed and current text highlighted, perhaps on a translucent background, could ameliorate both of these problems.

5.3. Direction: Movement and Blocking

In WOz, all stage directions are given via text (e.g. “run after Toto”). Text directions worked quite well in this prototype because it was a simple scene with slower paced dialog. However, this approach is insufficient for scenes involving complex, fast-paced choreography.

The PB scene helped us uncover several findings about the user interface needs for this type of experience, especially regarding how much explicit direction the user would require. Most interesting, perhaps, is the observation that while intuition tells us that explicit direction would be needed for a choreographed sword fight, the user gets many cues on when and how to move simply by watching and listening to the virtual actor. For example, when the virtual actor holds out his sword, this prompts the user to approach to look at it. Similarly, the movement of the actor towards you during the fight will also guide you into backing up and vice versa. While explicit cues will still be needed to help participants learn when to make proactive rather than reactive movements, it is likely that experienced participants would be able to eventually rely solely on natural cues.

Based on early feedback, the movement icons designed for the PB proved useful for simple directions that need to be done quickly (e.g. jump forward). However, it appears that the overhead map and/or ghost actor is more useful for complex direction guidance.

6. Implementation

Our prototypes are all implemented in the Designer’s Augmented Reality Toolkit (DART) [1]. DART is well suited for implementing this type of application, as its interpreted code-base is easily extended. Throughout the course of ARK development, as the experience and interface design issues were exposed, we quickly added new components to DART to support custom objects such as dialog prompts. The support for rapid prototyping techniques in DART facilitated the creation of several AR Karaoke prototypes in a relatively short time.

6.1. Interface Components

Several specialized DART components were created during the development of the ARK prototypes.

6.1.1. “Ghost” Actor. ARK applications can include a *GhostActor* to drive the target location or motion of the participant. The ghost encapsulates what the user should be doing in the scene. The movements of the ghost are driven by pre-recorded tracking data, stored in a Director cast, that can be obtained using DART’s data capture facilities while the developer is physically performing the scene herself. Since the tracker data is stored in a simple textual form, the developer is also free to modify or create data manually.

6.1.2. Heads Up Display (HUD). One approach to providing the user with direction information is via a HUD overlaid on the 3D view. Several components were created to allow the developer to customize a HUD panel for an ARK user.

Text Display: The main use of this component is for prompting the user with their lines, or for showing text stage direction.

Progress bar: This is a component that indicates how long a user has to say the current displayed line(s). The progress bar receives a “cue” that tells it how long the duration of the line is and a moving bar indicates to the user how much time is passing.

Overhead map: An overhead map can be used in ARK to display the locations of the virtual actors, the user, and the “target” actor.

Direction Display: This is a basic component that allows the developer to display images or animations in a section of the HUD. In the prototypes, these objects are used to show movement instructions along with application specific instructions for activities such as sword fighting

6.2. Architecture of an Application

Figure 3 shows the architecture of a generic ARK application. Cues are fired based on a timeline, and can also be generated by a human “wizard” when more control is needed. These cues are broadcast to the objects in the application, such as the virtual actors and the dialog prompter. The objects have lists defining what actions they should execute when a cue is received.

A live position and orientation tracker is used to track the user’s head, and pre-recorded tracker data is used to drive the movement of the ghost actors. The overhead map component monitors the location of each of the 3D components, such as the camera/user, the ghost actor, and the virtual actors to create the overhead map’s 2D display.

In the example in Figure 4, two characters are interacting in a scene; “A” is a virtual actor, and “B” is the user. The Timeline shows that “A” talks first (at 2.3 seconds). When this cue is fired the virtual actor shows an image sequence (video frames or an animatic) and starts an audio clip of A’s line. When it is time for the user, “B”, to talk the virtual actor moves forward, the text is displayed for the user to speak (“Let’s fight, you thief!”), and the progress bar starts displaying how long the user has to say the line. Before the user finishes her line, another cue will be fired, “Fight_Start.” This causes the direction display to show the user an icon indicating that she should adopt a fighting stance.

7. Future Work and Conclusion

This paper presents our initial exploration into the concept of ARK. There are many possibilities for improving the user experience. Future directions range

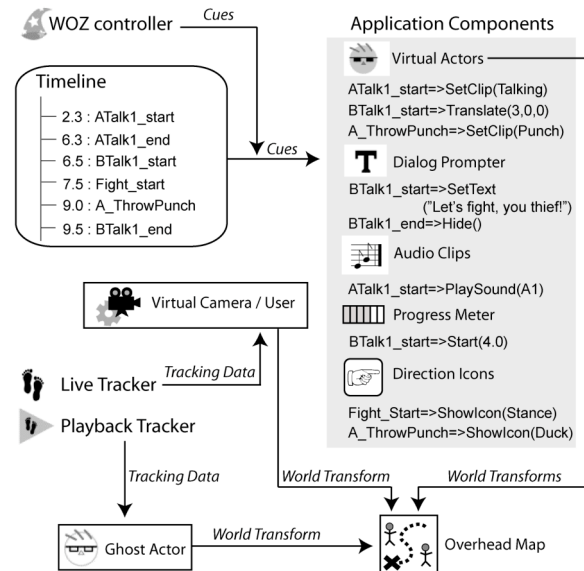


Figure 3. Architecture of a typical application.

from creating a multi-user version (suggested by some of our users) to exploring applications outside the bounds of entertainment, such as training and conflict resolution.

We have not yet performed a formal user study on these experiences. When we have polished content for one or more ARK scenes, we intend to evaluate the system not just from the performer viewpoint, but also from the viewpoint of the audience, as it is this shared experience that is the essence of karaoke.

8. Acknowledgements

We would like to thank Marleigh Norton for performing in the ARK prototypes during the making of the video and images for this paper, and the members of our lab and class for interesting discussions and demonstrations of the ARK concept. This work is supported by NSF CAREER Grant 0347712.

9. References

- [1] MacIntyre, B., M. Gandy, S. Dow, J. Bolter, “DART: A Toolkit for Rapid Design Exploration of Augmented Reality Experiences”, *Conference on User Interface Software and Technology (UIST’04)*, 2004.
- [2] Matsuba, S. N., B. Roehl, “Bottom, Thou Art Translated: The Making of VRML Dream”, *IEEE Computer Graphics and Applications*, 19 (2), March/April 1999, 45-51.
- [3] Ohya, J., K. Ebihara, J. Kurumisawa and R. Nakatsu, “Virtual Kabuki Theater: Towards the realization of human metamorphosis systems,” *5th IEEE International Workshop on Robot and Human Communication*, pp.416-421 (1996.11).
- [4] Slater, M., J. Howell, A. Steed, D-P. Pertaub, M. Garau, “Acting in virtual reality”, *Proceedings of the third international conference on Collaborative virtual environments*, 2000.