Augmented Reality Kanji Learning

Daniel Wagner   Istvan Barakonyi

Vienna University of Technology
Favoritenstr. 9-11/188/2
A1040 Vienna, Austria
{wagner|bara}@ ims.tuwien.ac.at

Abstract

ARToolKit programmers are familiar with the kanji symbols supplied with the distribution. Most of them have do not know what these kanji symbols mean. We propose a piece of educational software that uses collaborative Augmented Reality (AR) to teach users the meaning of kanji symbols. The application is laid out as a two player Augmented Reality computer game. The novelty of our approach is that we do not use regular workstations or laptops to host the AR (Augmented Reality) application. Instead we use fully autonomous PDAs, running the application together with an optical marker-based tracking module that makes this application not only available for a broad audience but also optimally mobile.

Keywords: Augmented Reality, Game, Educational Software, Tracking, PDA

1. Introduction

Using games as a framework for educational software has a long tradition. While classical learning applications are often dry and demotivating games have the potential to keep the user focusing for longer periods of time. Augmented reality allows extending classical games such as cards to a completely new dimension.

An early attempt at using immersive environments for educational tasks has been made in the "Zengo Sayu" project [5] which taught Japanese. Augmented Reality itself already has been successfully used in games. The ARQuake [2] project extended the famous first person shooter to let people walk around in the real world and shoot virtual enemies. Other games such as Mah-Jongg, Chess or Maze have been implemented [1] as AR games too.

All those approaches use either workstations or large, bulky backpack setups to provide the necessary infrastructure. While those setups allow quick application development by using common libraries and tools from the non-portable area, they are only intermediate steps towards truly mobile setups.

We demonstrate the first fully autonomous handheld AR game that runs on an unmodified PDA equipped with a camera and WiFi. To implement this application we used our Handheld AR framework.

2. AR kanji learning

AR Kanji is a slimmed down version of the Kanji Teaching Agent by Istvan Barakonyi. The PDA-based AR Kanji game is round based. Two players sit in front of each other having a pile of cards lying on the desk between them (see Figure 1).
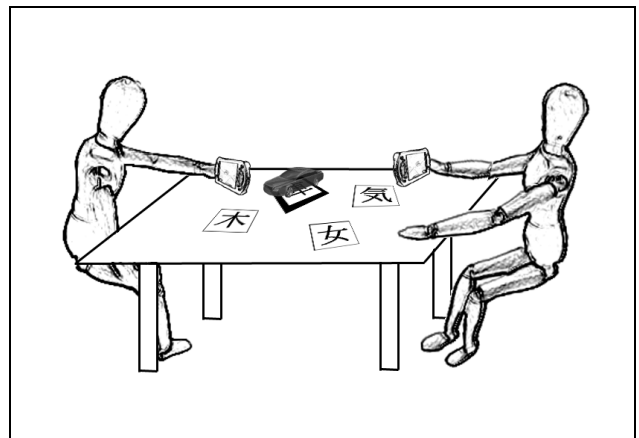
Figure 1: AR kanji learning setup showing a car.

The game contains 10 unique cards, each having a kanji symbol printed on both sides. On the front side the kanji symbol is surrounded by a black border so that they are detected by the tracking software when the front side is visible. Initially all cards are spread on the table with the back side up. The PDA shows an icon of the piece of

vocabulary to that the user must search for (such as a tree) and request the active player to find it. The PDA behaves as a magic lens: as the player turns a card, the PDA displays that card augmented with the corresponding 3D object. If the player chooses the correct card, he scores a point, keeps the card and has another go. Otherwise the player returns the card and then it is the opponents turn.

3. System Overview

The setup consists of two PDAs, each equipped with a camera and wireless connection to the opponents PDA. The system runs a ported and speed-optimized ARToolKit [4] version natively. Speed optimization was done by using the Intel Fixed Point library which is part of the Intel GPP [6].

Rendering is done with SoftGL [3], an OpenGL subset developed in-house which implements the most important 3D graphics primitives such as projection, modelview and texture matrices, lighting and video background. SoftGL is being developed with portability and performance in mind. Platform configuration is done by modifying template parameters. This way the type of floating point and the video memory layout can be configured.

The AR Kanji game is implemented as a lightweight application based on the aforementioned modules. Both PDAs retain system state which includes the undiscovered kanji symbols, the current card and the score of both players. The PDA of the active player acts as the server and determines whether the player chose the correct card or not.

4. Results

The AR Kanji game demonstrates the simplicity of building compelling applications on top of well defined basic libraries such as OpenGL and ARToolKit. By porting these libraries onto the PocketPC platform we opened a door for more upcoming games and applications on this promising platform.

Due to optimizations to the ported ARToolKit module the systems runs at interactive frame rates of 3-4 fps. The marker detection can be outsourced to a server using a wireless connection which speeds up the frame rate to ~5 fps. In this setup the overall performance is mainly limited by the CompactFlash camera, which delivers maximum 7 images per second. We expect faster hardware and multimedia accelerating PDA in the near fugure.

References

[1] Szalavári, Z., Eckstein, E., Gervautz, M.: Collaborative Gaming in Augmented Reality, *Proceedings of VRST'98, pp.195-204*, Taipei, Taiwan, November 2-5, 1998.

[2] Thomas, B., Close, B., Donoghue, J., Squires, J., de Bondi, P., Morris, M., Piekarski, W.: ARQuake: An Outdoor/Indoor Augmented Reality First Person Application, *Fourth International Symposium on Wearable Computers (ISWC'00)*, October 18 - 21, 2000

[3] Wagner, D.: SoftGL, An OpenGL Subset, http://www.ims.tuwien.ac.at/research/handheld_ar/develo per/softgl.php

[4] Billinghurst, M., Kato, H., Weghorst, S. and Furness, T. A. A Mixed Reality 3D Conferencing Application. *Technical Report R-99-1 Seattle: Human Interface Technology Laboratory, University of Washington,* 1999

[5] Rose, H., Billinghurst, M.: Zengo Sayu: An Immersive Educational Environment for Learning Japanese. University of Washington, Human Interface Technology Laboratory, Report No. r-95-4, 1995

[6] Intel Graphics Performance Primitives for the Intel® PXA250 Applications Processor, http://www.intel.com/ design/pca/applicationsprocessors/swsup/gppv1.htm