

ARTierchen – Augmented Reality in touch

Armin Strobl, Ansgar Wolsing, Christina Mohr, Rouven Lotze,
Eike Michael Lang, Jürgen Ziegler

Institut für Informatik und Interaktive Systeme, Universität Duisburg-Essen

Zusammenfassung

Augmented Reality (AR) ist eine Technik, welche die reale Welt um virtuelle Objekte erweitert. Unter Nutzung von mobilen Geräten und speziellen AR-Displays können auch virtuelle Spielwelten in räumliche Umgebungen eingebettet werden. Im vorliegenden Beitrag sollen am Beispiel des Augmented-Reality-Spiels „ARTierchen“ die Möglichkeiten dargestellt werden, die sich dem Entwickler durch die computergerenderte visuelle Erweiterung einer zugrunde liegenden natürlichen Spielwelt für interaktive 3D-Spiele bieten.

1 Einleitung

Augmented Reality (dt. „erweiterte Realität“) ist die Kombination virtueller und physikalischer Realität (Azuma 1997). Die physikalische Realität wird in Echtzeit um Objekte erweitert, die in ihren dreidimensionalen Kontext eingebunden werden. Azuma (1997) hebt insbesondere die Aspekte der Interaktivität und der Kombination von Informationen in Echtzeit hervor. Auch wenn die Konzentration von Forschungsarbeiten und technischen Umsetzungen von AR-Systemen auf den Bereich der Erweiterung visueller Reize eine Beschränkung suggeriert, können auch auditive, olfaktorische oder haptische Stimuli „ergänzt“ werden (vgl. z.B. Landin al. 2002).

Das „ARTierchen“ ist ein interaktives Spiel, welches das Potential von Augmented Reality nutzt. Seine Entwicklung begann 2004 als interdisziplinäres Praxisprojekt am Lehrstuhl für Interaktive Systeme und Interaktionsdesign der Abteilung für Informatik an der Universität Duisburg-Essen. Zielsetzung des Projekts waren der Entwurf und die Implementierung eines Spielkonzepts und die Einbettung der virtuellen Spielwelt in eine räumliche reale Umgebung. Während des gesamten Prozesses stand auch stets die Entwicklung eines Tangible User Interfaces (TUI) im Mittelpunkt. Dabei mussten zwei Anforderungen an ein geeignetes Eingabegerät miteinander vereinbart werden: bei einer möglichst hohen Funktionalität sollte sich die Nutzung des TUI dem Spieler möglichst leicht verständlich, intuitiv handhabbar,

eingänglich und unkompliziert gestalten (vgl. zur Interaktion mit virtuellen Objekten mittels stofflicher Gegenstände, Hornecker 2004).

2 Konzeptioneller Spielaufbau

Protagonistin des ARTierchen ist das namensgebende „Tierchen“, das animierte 3D-Modell einer Giraffe. Wie bei einem realen Haustier muss sich der Spieler um das Wohlbefinden der Giraffe kümmern, für ausreichende Ernährung sowie Spiel und Abwechslung sorgen. Dem Spieler stehen dazu unter anderem eine Frisbee, die er der Giraffe zuwerfen kann, ein Trinknapf und Süßigkeiten als virtuelle Items zur Verfügung. Zugleich muss er sein „Haustier“ vor den Gefahren schützen, die die virtuelle Spielwelt birgt: eine fleischfressende Pflanze trachtet der Giraffe nach dem Leben. Es ist Aufgabe des Spielers, diese Pflanze zu vertreiben. Als Spielprinzip haben wir uns an dem 1996 entwickelten Tamagotchi orientiert, dem wir mit Hilfe einer greifbaren Schnittstelle mehr Interaktivität verleihen wollten.



Abbildung 1: Augmented-Reality-Spielfeld

Die Implementierung der Interaktion zwischen dem Spieler und den virtuellen Objekten soll im Folgenden näher beschrieben werden.

3 Implementierung

Markerbasiertes Tracking

Die in C++ geschriebene Implementierung des *ARTierchens* basiert auf dem *ARToolkit* (von Kato et al. 2000), einer Softwarebibliothek zum Erstellen von Augmented-Reality-Applikationen. Das *ARToolkit* benutzt verschiedene Bilderkennungsalgorithmen, um die Überlagerung der realen Welt mit virtuellen Objekten umzusetzen. Es enthält eine Videotracking-Bibliothek, die die Kameraposition und die relative Orientierung in Echtzeit berechnet. Zur Verbesserung des markerbasierten Trackings wurde das *ARToolkit* um die *ARTag-*

Bibliothek (Fiala 2004) ergänzt. Diese bietet eine stabilere Marker-Erkennung und arbeitet auch bei schlechten Lichtverhältnissen zuverlässig. Es reicht schon eine unvollständige Bildinformation aus, um einen Marker zweifelsfrei zuzuordnen. Die Funktionsweise des ARTag unterscheidet sich insofern von der des ARToolkits, dass ARTag ein vorgegebenes System von 36-Bit-Schwarz-Weiß-Mustern als Marker nutzt (Fiala 2004), während für die Verwendung des ARToolkits Repräsentationen der verwendeten Marker gespeichert werden. Eine optimale Erkennung der Marker beruht auf der Maximierung der Hamming-Distanz¹ zwischen den Markern. Dies wird durch ein geeignetes Auswahlverfahren der Marker erreicht. Die von den Bibliotheken bereitgestellten Funktionen erlauben es, die Position eines Markers relativ zur Kamera auf den Zentimeter genau zu bestimmen. Die Erkennung lässt sich dabei grob in zwei Phasen unterteilen. Erst erfolgt eine unspezifische Erfassung eines Markers, anschließend wird versucht, gefundene Marker exakt zu identifizieren. Bei der Erfassung eines Markers wird zuerst die Grundstruktur des Markers erkannt. Allen Markern ist dabei ein schwarzes Quadrat auf weißem Grund als Grundfläche gemeinsam (vgl. Abbildung 2, links, unter 1.). Anschließend werden Vektormuster in den erkannten Quadraten gesucht (vgl. Abbildung 2, links, unter 2.). Da zumindest die Verzerrung des zu suchenden Musters bekannt ist, kann nun das passende Muster zugeordnet werden.

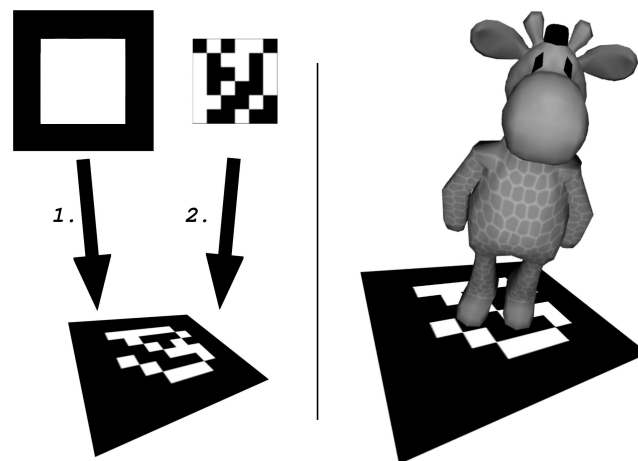


Abbildung 2: Marker dienen der Erkennung der Kameraposition. Jedem Marker sind die umgebende schwarze Fläche und ein Muster im Inneren zugeordnet. Werden Grundstruktur (1.) und das Vektormuster (2.) erkannt, so wird auf dem Marker das virtuelle Objekt gerendert.

Erst dann ist die Berechnung der Tiefe im Raum möglich. Da die Ausmaße der Marker variieren können, muss die Größe des erkannten Markers mit der ihm zugeordneten Größe verglichen werden, um seine Distanz zur Kamera zu erhalten. Die Koordinaten der Marker liegen am Ende eines Erkennungsdurchgangs in Matrixform vor, so dass deren Position und

¹ Die Hamming-Distanz ist die Anzahl der Stellen zweier Binärzahlen, die sich paarweise voneinander unterscheiden.

Rotation bekannt sind. Auf diesen Informationen basierend kann nun die Umgebung berechnet werden. Im einfachsten Fall ist dies die Projektion eines Objektes auf einen Marker.

Unsere Implementierung setzt auf der *OpenGL*-Schnittstelle des Betriebssystems auf. Zusätzlich wurde freier Quelltext (basierend auf Quake II, id Software 1997) unter der *GNU Public Licence* (Free Software Foundation, Inc. 1989) eingebunden, um die im md2-Format vorliegenden 3D-Modelle zu laden. Die Spiellogik wurde durch ein objektorientiertes C++-Klassenmodell realisiert.

Interaktion mit virtuellen Objekten

Der Spieler interagiert mit der Spielfigur über das von uns entwickelte als *Advanced Flexible Pointing Device* (AFPD) bezeichnete TUI, einer auf Vorder- und Rückseite mit zwei Interaktionsmarkern versehenen handelsüblichen Fliegenklatsche (vgl. Abbildung 3). Der sogenannte *Laufmarker* auf der Vorderseite dient dazu, die Giraffe zu einem per AFPD angezeigten Zielort zu führen. Der *Aktionsmarker* auf der Rückseite dient sämtlicher weiterer Interaktion mit der Giraffe, anderen Elementen oder der virtuellen Umgebung. Unter Verwendung des Aktionsmarkers werden Aktionen der Giraffe ausgelöst. In der praktischen Anwendung kann der Spieler der Giraffe so z.B. Kommandos geben oder sie streicheln. Zugleich kann der Spieler mit ihm die fleischfressende Pflanze zerstören. Nutzt der Spieler den Aktionsmarker, so wird auf das AFPD eine virtuelle Hand gerendert. Ist der Laufmarker aktiv, so wird die räumliche Entfernung des AFPD zur gerenderten Wiese durch die Position einer roten Kugel dargestellt, die Aktionsbereitschaft der Giraffe durch einen Kreis bildende blaue Kugeln, wobei die Anzahl der Kugeln proportional zur Aktionsbereitschaft steigt.



Abbildung 3: Interaktion des Spielers mit der Augmented-Reality-Welt

Der Kontakt des AFPD mit der Giraffe oder den interaktionsrelevanten Objekten der Spielwelt wird über eine Kollisionsabfrage festgestellt, nachfolgend werden dann die objektspezifischen Reaktionen ausgelöst. Die Abarbeitung der entstehenden Aktionen und Interaktionen wird mittels einer *Fifo-Queue*² organisiert. Auslösende Aktionen werden fortlaufend registriert und erzeugen Befehlssequenzen. Die *Fifo-Queue* wird mit diesen Sequenzen gefüllt und dann abgearbeitet. Einige Aktionen löschen alle vorhergehenden Befehle aus der Queue. Die Unterteilung einer Aktion in mehrere Einzelschritte erlaubt es, komplexere Vorgehensweisen

² First in-First out-Warteschlange

als eine Kette von übersichtlichen Funktionen auszudrücken. Zum Tragen kommt dieser Vorteil zum Beispiel bei allen Aktionen, vor deren Durchführung die Giraffe zunächst einen bestimmten Ort aufsuchen soll. Der erste Befehl beinhaltet dann das Kommando zum Laufen, ein zweiter Befehl die eigentliche Aktion.

Charakterfigur Giraffe

Die Persönlichkeit der Giraffe wird durch die Attribute und Methoden der Giraffen-Klasse abgebildet. Zur Modellierung der Persönlichkeit der Giraffe wird ein *Emotionspunktesystem* verwendet, das sich aus vier Komponenten zusammensetzt. Diese Komponenten spiegeln jeweils prozentual die Eigenschaften der Giraffe wieder: Ernährung, Zufriedenheit, Ausdauer und Gesundheit. Die Interaktion mit der Spielwelt und dem Spieler beeinflusst den emotionalen Zustand der Giraffe, der durch kontexterkennende Methoden nachgehalten wird. Zugleich beeinflussen sich die Persönlichkeitsattribute auch untereinander: Ist die Giraffe krank, was durch einen niedrigen Gesundheitswert dargestellt wird, verringert sich sukzessive ihre Zufriedenheit. Darüber hinaus ist die Interaktion mit dem Spieler durch den emotionalen Zustand der Giraffe bestimmt. Schickt der Spieler mit Hilfe des AFPD die Giraffe in das in die Spielwelt integrierte Bett, legt sie sich nur dann schlafen, wenn ihr Ausdauerwert eine bestimmte Schwelle unterschreitet. Wird dieser Wert nicht unterschritten, hüpfte sie auf ihrem Bett. Der Spieler kann den aktuellen Emotionsstatus der Giraffe jederzeit über die vier Eigenschaften repräsentierenden Statusbalken nachvollziehen und entsprechend mit Interaktionseingaben reagieren.

4 Fazit

Durch das ARTierchen wird eine Interaktion zwischen einem menschlichen Spieler und einem virtuellen Charakter ermöglicht. Der Spieler kann mittels eines TUI in die virtuelle Spielwelt eingreifen und (Re-)Aktionen der Giraffe verursachen oder einleiten. Somit ist der Spieler nicht an stereotypische Eingabegeräte gebunden, sondern kann intuitiv mit der Welt interagieren; eine vorherige Einweisung in die Peripherie ist nicht nötig. Markerbasiertes Tracking dient hierbei der Berechnung der Kameraposition; wenn die Kamera an einem Head-Mounted Display (HMD) installiert ist, indirekt auch der Kopfposition des Anwenders, der die um virtuelle Objekte erweiterte Spielwelt jederzeit aus dem korrekten, realistischen Blickwinkel registriert. Dies stellt die erste Grundbedingung für Immersion dar, also das „Eintauchen“ des Anwenders in die Augmented-Reality-Situation.

Bei unseren bisherigen Präsentationen stießen wir auf großes Interesse von Seiten der Wirtschaft und Politik. Da sich das Spiel immer noch in der Weiterentwicklung befindet, können wir momentan einen fortgeschrittenen Prototypen präsentieren. Die bisherige Resonanz lässt aber hoffen, dass ein abgeschlossenes Projekt sich durchaus großen Interesses erfreuen wird.

Literaturverzeichnis

<http://www.artierchen.de>

Azuma, R. (1997): A Survey of Augmented Reality. In: Teleoperators and Virtual Environments 6, 4 (August 1997), S. 355-385.

Fiala, M. (2004): ARTag Revision 1, A Fiducial Marker System Using Digital Techniques. In: NRC/ERB-1117. November 24, 2004.

Hornecker, E. (2004): Tangible User Interfaces als kooperationsunterstützendes Medium. Dissertation zum Dr. Ing. an der Universität Bremen.

Kato, H.; Billinghurst M.; Poupyrev, I. (2000): ARToolKit Documentation. In: <http://www.hitl.washington.edu/people/grof/SharedSpace/Download/ARToolKit2.33doc.pdf> und <http://www.hitl.washington.edu/artoolkit/>.

Landin, H.; Lundgren, S.; Prison, J. (2002): The Iron Horse – a Sound Ride, in: Proceedings of Nord-CHI 2002-The Second Nordic Conference on Human-Computer Interaction, S. 303ff. ACM Press.

Free Software Foundation, Inc. (1989): General Public License. In: <http://www.gnu.org/licenses/gpl.html>.

Quake II (1997): id Software. In: <http://www.idsoftware.com/>.