

Duplo – Ein virtuelles Lego-Spiel

Peter Brandl, Elisabeth Gugerbauer, Michael Haller, Andreas Reichinger, Judith Wimmer, Dieter Zirinig

Fachhochschule Hagenberg, Medientechnik und –design, Hagenberg, AUSTRIA,
[peter.brandl | elisabeth.gugerbauer | michael.haller | andreas.reichinger
judith.wimmer | dieter.zirinig] @ fhs-hagenberg.ac.at

Zusammenfassung: *Virtual Reality ist in der heutigen Zeit zwar immer noch eher Spezialisten vorbehalten, aber über kurz oder lang wird sie sich wohl in immer mehr Gebieten etablieren und auch in den Consumer-Markt Einzug nehmen. Grund genug, sich einmal näher mit dem Gebiet zu beschäftigen. Mit einem Trackingsystem und einer halbdurchsichtigen VR-Brille entwickelten wir, im Rahmen eines Semesterprojektes ein virtuelles Lego-Spiel. Damit kann man auf einer realen Oberfläche mittels selbstgebautes 3D-Joystick virtuelle Lego-Steine nehmen, in virtuellen Farbtöpfen färben, auf dem Spielfeld setzen und wieder nehmen, in einem Abfalleimer wegwerfen, oder einfach wegschießen. Außerdem entwickelten wir eine Möglichkeit, die VR-Brille optimal an den Benutzer anzupassen, damit virtuelles Bild mit der Realität vollkommen übereinstimmt.*

1. Einleitung

Dieses Projekt entstand im vierten Semester als einsemestrige Arbeit im Rahmen des Faches „Anwendungsbezogene Seminare und Projekte“ an der Fachhochschule Hagenberg, Studienzweig „Medientechnik und -design“.

Prinzipiell war das Projekt eher als eine Machbarkeitsstudie vorgesehen, ob und wie sich die an der Fachhochschule befindlichen Virtual-Reality-Geräte am Besten einsetzen lassen. Dabei fanden wir es besonders spannend, ein virtuelles Lego-Spiel zu entwickeln.

2. Das Setup

Es standen uns ein magnetisches Trackingsystem (FASTRAK) der Marke „Polhemus“ mit vier Receivern und eine halbtransparente 3D-Brille der Marke „IGlasses“ zur Verfügung [1, 2].

Als Eingabegerät bauten wir in einen alten Joystick einen Tracker ein. In der virtuellen Szene schwebt dann ein 3D-Cursor in einem bestimmten Abstand vor dem Joystick. Die beiden Tasten des Sticks wurden über ein Kabel mit den Tasten einer Maus verbunden, die dann einen Tastendruck am Joystick wie einen Mausklick an den PC weiterleitet.



Abbildung 1: Die Hardware von Duplo umfasst einen FASTRAK-Tracker, einen Joystick mit integriertem Tracker-Receiver und die 3D-Brillen.

Auch auf den IGlasses ist ein Tracker befestigt. Anhand der Daten dieses Trackers wird auf die Augenposition und Blickrichtung zurückgerechnet, um die virtuelle Kamera

den realen Gegebenheiten entsprechend auszurichten. Mit diesen IGlasses ist es möglich die virtuelle Szene kombiniert mit der Realität zu sehen. Es ist, als würden die virtuellen Lego-Steine leicht transparent in der Wirklichkeit herumstehen (siehe Abbildung 5). Zusätzlich zu diesem rein visuellen, künstlerischen Aspekt der Transparenz ist es auch sehr wichtig, nicht vollständig von der Realität abgeschottet zu sein, um nicht das Gleichgewicht zu verlieren, oder mit diversen Gegenständen zu kollidieren. Außerdem erleichtert sie den Umgang mit dem Joystick. Man muss nicht erst den 3D-Cursor suchen, er schwebt ohnehin immer vor dem real sichtbaren Joystick.

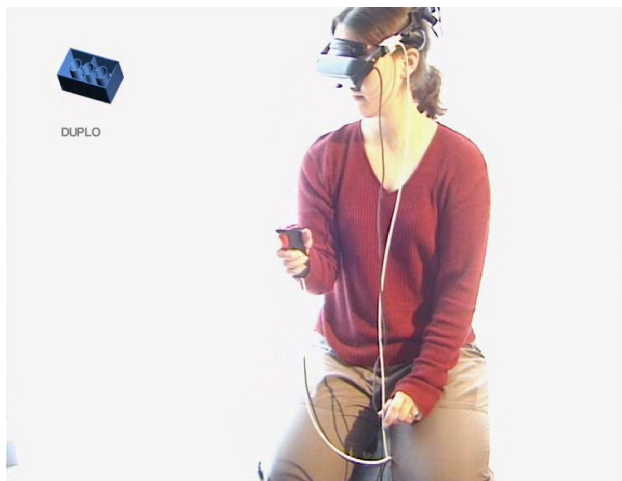


Abbildung 2: Der „verkabelte Benutzer“.

Als zusätzliches „Eingabegerät“ verwendeten wir den Tracking-Sender, der für gewöhnlich fix montiert wird. Durch ein Bewegen des Referenzpunktes wird ja die gesamte Szene verschoben. Für unsere Lego-Applikation finden wir gerade diese Metapher für sehr sinnvoll, da ja auch beim realen Lego einmal auf dem Tisch gespielt und dann auf dem Boden weitergebaut wird.

Das Programm läuft auf einem handelsüblichen PC unter Microsoft Windows. Für eine flüssige Performance wird aber eine schnelle OpenGL-fähige Grafikkarte und für die Soundunterstützung natürlich eine Soundkarte benötigt.

3. Das Spiel

3.1 Szenenaufbau

Grundüberlegung war, dass sich der Polhemus Tracker vor dem am Boden sitzenden Spieler befindet (vgl. Abbildung 3). Ausgehend von der Annahme, dass die meisten Spieler Rechtshänder sind und somit den Joystick (das Interaktionsinstrument) mit der rechten Hand führen

würden, sind die Steine und Farbtöpfe, die am häufigsten verwendet werden, rechts vom Spieler innerhalb seiner Reichweite platziert. Links vom Spieler befindet sich ein Mistkübel, der dazu dient, Steine zu verwerfen. Direkt vor dem Spieler ist die Spielfläche auf der er tatsächlich bauen kann. Das heißt Steine, die direkt auf der Spielfläche gesetzt werden, dürfen nur in diesem Bereich liegen.

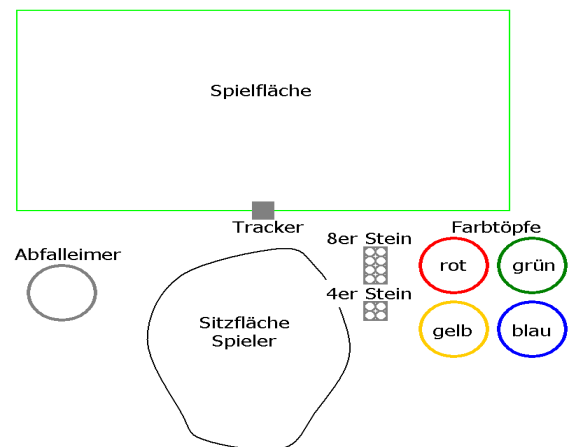


Abbildung 3: Schematischer Szenenaufbau.

3.2 Interaktion

Man kann Lego-Bausteine aufnehmen, indem man den 3D-Cursor direkt in den Stein bewegt und klickt. Dabei kann man entweder einen der beiden Grundsteine nehmen (befinden sich fix in der Szene neben den Kübeln und bleiben immer an der Stelle, sodass man immer wieder einen der Steine nehmen kann). Oder man nimmt einen frei zugänglichen Stein aus der bereits gebauten Szene. Hat man den Stein „in der Hand“, kann man ihn mit einem weiteren Klick auf der Spielfläche platzieren; aber nur dort, wo er im realen Lego-Spiel auch platziert werden könnte. Die Setzmöglichkeit wird einem durch ein Snappen auf die Stelle signalisiert. Durch Eintauchen der Bausteine in einen der vier Farbtopfmodelle werden die Steine mit der entsprechenden Farbe gefärbt. Ist man mit dem Selbstgebauten nicht zufrieden, hat man die Möglichkeit, Steine wegzunehmen und in einem Abfalleimer zu entsorgen. Für eine bessere Bedienbarkeit wird durch bestimmte Geräusche dem Benutzer signalisiert, ob er eine Aktion erfolgreich ausgeführt hat.

Als lustiger Alternative zum Abfalleimer kann man den Stein auch wegschießen. Außerdem besteht die Möglichkeit die Szene größer oder kleiner zu machen, und im Navigationsmodus sich in der beispielsweise vergrößerten Szene zu bewegen.

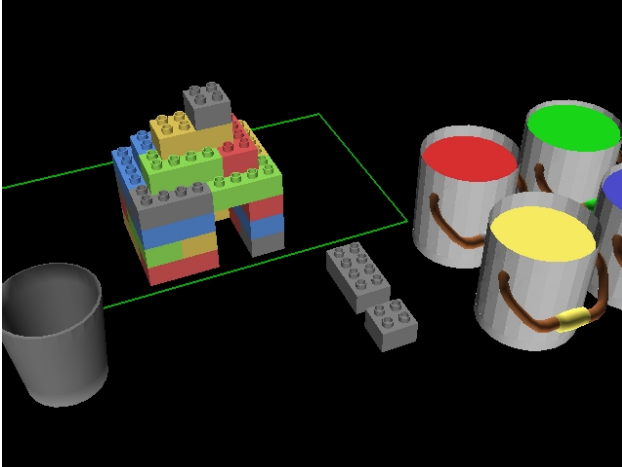


Abbildung 4: Szene aus der Sicht des Benutzers ohne Realität.

3.3 Kalibrierung der IGlasses

Da man immer eine Kombination aus Realität und virtuellem Bild sieht, ist es sehr wichtig, dass beide Bilder dabei einander genau entsprechen. Ist dies nicht der Fall, ist der 3D-Cursor abhängig von Betrachtungswinkel nicht in konstantem Abstand zum Joystick, und die virtuelle Szene verschiebt sich bei jeder Kopfbewegung. Dies ist einerseits störend für die Interaktion, andererseits kann man auch „seekrank“ werden [5]. Es sollten also für jeden Benutzer die IGlasses kalibriert werden.



Abbildung 5: Der Benutzer kann in der realen Welt einen virtuellen Lego-Turm bauen.

Bei der Kalibrierung muss man für jedes Auge nacheinander durch vier Punkte am Display eine Art Zielvorrichtung anvisieren, und dann den Joystickknopf drücken. Über sehr aufwändige Berechnungen werden dann die nötigen Offsets der Augenposition berechnet.

4. Implementierung

Sämtliche Objekte (Steine, Farbtöpfe, Abfalleimer) wurden mit Kinetix 3D Studio Max modelliert. Bei den Maßen wurde auf wirklichkeitsgetreue Proportionen geachtet, die tatsächliche Objektgröße (deutlich größer als wirkliche Steine) ergab sich aus Aspekten der Benutzerfreundlichkeit. Es wurden bewusst keine Texturen verwendet, um einerseits die Performance zu steigern und andererseits den echten „Lego-Steinen“ besser zu entsprechen. Diese Objekte wurden mit dem Programm 3D Exploration von Right Hemisphere [3] in OBJ-Files konvertiert, die dann mit einem entsprechenden Loader (GLM Loader von GLUT) in OpenGL integriert werden.

Generell wurde das Programm in C/C++ mit den OpenGL-Libraries geschrieben. Für den Sound wurde OpenAL verwendet, da diese API das unkomprimierte Format WAV unterstützt [4]. Das hat im Vergleich zu dem Format MP3 den Vorteil, dass das File beim Abspielen nicht decodiert werden muss, sondern direkt wiedergegeben werden kann.

Es entstanden dabei relativ einfach zu bedienende Klassen für die Verwendung von IGlasses (mit Kalibrierung), für das Polhemus Tracking-System, für Microsoft Maus-Buttons und für legoähnliche Bausteine, die leicht wiederverwendbar sind.

Für das Dokumentationsvideo war es nötig die Bewegungen aufzeichnen zu können, und später aus verschiedenen Perspektiven wieder abspielen zu können. Auch gibt es noch die Möglichkeit einen Tracker auf einer Kamera zu montieren, um die virtuelle Szene aus der Sicht einer realen Kamera zu rendern.

5. Resümee

Benutzertests haben gezeigt, dass die Verwendung des 3D-Joysticks sehr einfach und intuitiv ist. Anfängliche Probleme haben die meisten Leute allerdings mit dem 3D-Cursor, denn auf den gewohnten 2D-Oberflächen reicht es bereits, wenn man **über** einem Objekt klickt. Im 3D-Raum muss man **in** das Objekt hineinfahren. Auch gibt es zu Beginn Schwierigkeiten sich im 3D-Raum zurechtzufinden und Tiefen abschätzen zu können – obgleich die Applikation Stereographik unterstützt. Für viele ist es ungewohnt, dass die IGlasses nur einen kleinen Sichtbereich haben, und man den ganzen Kopf und nicht nur die Augen bewegen muss, um alles zu sehen. Brillenträger sind hier eher bereits daran gewöhnt. Generell gilt aber, dass sich fast alle Testpersonen sich

bereits nach wenigen Minuten gut mit dem System zurecht gefunden haben.

Problematisch ist immer noch die Hardware. Sie ist zwar recht teuer, ist aber sehr eingeschränkt! Bei den (vergleichsweise billigen) IGlasses ist der Sichtbereich und die Auflösung sehr klein. Außerdem muss die Umgebungshelligkeit stimmen, denn in zu hellen Räumen sieht man nichts vom virtuellen, in zu dunklen Räumen nichts vom realen Bild.

Das Trackingsystem hat zwar laut Hersteller einen Aktionsradius von 3 Metern, in der Praxis sollte man sich allerdings nur etwa 50 cm vom Sender entfernen, sonst fängt das Bild zu zittern an, und die Koordinaten werden immer verfälschter. Außerdem muss man sich von Metall fernhalten. Das heißt, dass man auf den meisten Tischen und auf „Büroböden“ magnetische Tracker nicht verwenden kann.

Bleibt nur zu hoffen, dass die Systeme bald besser und vor allem erschwinglicher werden.

6. Referenzen

[1] Polhemus, *Fastrak User's Manual, Revision*, November 1993, Technischer Report, OPM3609-002C, <http://www.polhemus.com>

[2] I-O Display Systems, *Personal Display System*, Technischer Report, MA2276B, <http://www.i-glasses.com/>

[3] Right Hemisphere, *What is 3D Exploration?* http://www.us.righthemisphere.com/3dexploration/3d_exploration_home.htm, Juni 2001

[4] John Grantham, *OpenAL, Announcing a new open source platform for all-purpose high-fidelity sound*, Loki Entertainment Software, 2000, <http://www.openal.org>

[5] Joseph J. LaViola Jr., *A Discussion of Cybersickness in Virtual Environments*, SIGGCHI Bulletin, Volume 32, Number 1, 47-56, Januar 2000.