

Dynamische, interaktive Virtuelle Umgebungen

Kristopher J. Blom

interactive media · virtual environments
Dept. Informatik, Universität Hamburg
blom@informatik.uni-hamburg.de

Abstract: Virtuelle Umgebungen sind Bestandteil zahlreicher aktueller Anwendungssysteme. Die Spannweite der Anwendungsbereiche reicht vom Ingenieurwesen und der klinischen Psychologie bis zur Unterhaltungsindustrie. Dabei ist es oft wichtig, erfahrbare und glaubhafte Umgebungen zu schaffen. Da diese virtuell sind, sind der Schaffung solcher Umgebungen theoretisch ausschließlich die Grenzen der Vorstellungskraft des Designers gesetzt. Leider wird die praktische Implementierung solcher Umgebungen stark durch eine mangelhafte Unterstützung bei der Programmierung beschränkt. Dieser Artikel fasst die Arbeit des Autors an einer Klasse solcher erfahrbarer Umgebungen, den Dynamic Interactive Virtual Environments (DIVEs), zusammen. Dynamisch bedeutet hier die dynamische Veränderung der virtuellen Welt, interaktiv die echtzeitfähige Einflussnahme auf die Welt. Zunächst wird der Gestaltungsraum von DIVEs untersucht. Basierend auf den hierbei gewonnenen Erkenntnissen wird ein neues, physiknahes System zur Unterstützung der DIVE Entwicklung vorgeschlagen und implementiert. Das hieraus hervorgegangene System, Functional Reactive Virtual Reality, bietet eine neuartige Unterstützung von Designern und Programmierern bei der Erschaffung dynamischer, reichhaltiger, glaubhafter virtueller Umgebungen.

1 Einführung

Virtuelle Umgebungen ermöglichen die Erstellung vielseitiger Welten, die theoretisch nur durch die Vorstellungskraft ihrer Autoren beschränkt sind. Diese Welten können mittels spezialisierter Software erschaffen werden, welche wiederum nur durch das Können der Entwickler limitiert ist. Um statisch modellierte Virtuelle Umgebungen lebendig gestalten zu können, stellt allerdings das benötigte programmiertechnische Können eine große Hürde dar. Trotzdem wird die Möglichkeit, solche Umgebungen zu schaffen, oft gewünscht und allgemein als notwendig angesehen. Der Erfolg und die Effektivität vieler Anwendungsgebiete hängen entscheidend von der Erzeugung interessanter, fesselnder und affektiver Virtueller Umgebungen ab.

Die Schaffung solcher Umgebungen, die reichhaltige Erfahrungen bieten, hängt von vielen Faktoren ab. Einen nicht unerheblichen Anteil hieran hat die Kreativität des Designers. Doch aufgrund der gegenwärtig verfügbaren Systeme wird der kreative Aspekt leider nicht selten von den technischen Schwierigkeiten bei der Programmierung solcher Umgebungen in den Hintergrund gedrängt. Das Hauptanliegen der hier präsentierten Arbeit besteht folglich darin, die Unterstützung für die Gestaltung dieser erfahrbaren Umgebungen zu

verbessern.

Um dies zu erreichen, muss man sich zunächst darüber im Klaren sein, an welchen Stellen Unterstützung zur Verfügung gestellt werden kann und was genau unterstützt werden soll. Es existieren zwei Hauptbestandteile in virtueller Echtzeit-Umgebungen, die für einen programmierungsseitigen Support in Frage kommen: Dynamiken und Interaktionen. Vereinfacht ausgedrückt, sind Dynamiken Veränderungen der Umgebungen, ihrer Darstellung oder ihrer Bestandteile, die sich über einen Zeitraum hinweg ereignen. Dynamiken ziehen die Aufmerksamkeit des Nutzers auf sich und machen Umgebungen interessant, vor allem über längere Zeiträume hinweg. Die Interaktion mit der Umgebung vermittelt ein Gefühl von "Agency", das dem Nutzer hilft, sich als Teil der Umgebung zu fühlen. Umgebungen, die sich aus diesen Komponenten zusammensetzen, bezeichnen wir als Dynamic Interactive Virtual Environments (DIVEs), zu deutsch Dynamische interaktive virtuelle Umgebungen.

Dieser Artikel fasst die Dissertation des Autors zum Thema DIVE zusammen [Blo09]. Die Arbeit gliedert sich in zwei Hauptbereiche. Der erste definiert DIVEs und analysiert ihre Natur. Diese kritische Untersuchung vermittelt uns ein Verständnis von dem, was sich kreieren lässt und bildet so die Grundlage für die Entwicklung von Werkzeugen, die die Gestaltung solcher Umgebungen unterstützen. Der zweite Teil bezieht sich auf die zweite Phase der Untersuchung – die Entwicklung eines Supportsystems auf Basis der Anforderungen, die sich aus der Analyse des Gestaltungsraums und der Untersuchung bereits bestehender Ansätze ergaben. Darüber hinaus stellen wir unsere Herangehensweise und das hieraus resultierende System, Functional Reactive Virtual Reality (Funktionale reaktive Virtuelle Realität) vor.

2 Gestaltungsraum von DIVEs

Aus dynamischen und interaktiven Komponenten bestehende Virtuelle Umgebungen stehen im Mittelpunkt dieser Untersuchung. Es wurden zahlreiche Systeme zur Schaffung von Virtuellen Umgebungen und Interaktionen entwickelt. Natürlich liegen einigen dieser Arbeiten ähnliche Beweggründe und bis zu einem gewissen Grad ähnliche Ziele zugrunde, doch interessanterweise wird in keiner dieser unmittelbar relevanten Arbeiten präzise definiert, was genau unterstützt werden soll. Ebenso wenig werden Gestaltungsräume bestimmt. In diesem Abschnitt erläutern wir unsere Herangehensweise an Definition und Bestimmung der angestrebten Umgebungen und präsentieren Auszüge aus den Ergebnissen.

Da für solche Umgebungen keine präzise Definition existiert, bestand die erste Aufgabe darin, das Interessengebiet zu definieren. Dies hatten wir zum Teil bereits getan, indem wir den Schwerpunkt auf Dynamiken und Interaktionen legten. Für diese wählten wir eine sehr großzügige Definition, um unsere Forschung nicht künstlich einzuschränken. Betrachtet man die kombinierte Bedeutung von Dynamiken und Interaktionen, ergeben sich drei interessante neue Forschungsrichtungen. Dynamische Interaktionen sind Dynamiken, die allein durch die Interaktionen des Nutzers entstehen. Beispielhaft seien hier gängige

Interaktionen in Virtuellen Umgebungen wie die Fortbewegung und „direkte Manipulationen“ genannt. Interaktive Dynamiken sind Interaktionen, bei denen der „Gegenstand“ einer Interaktion eine Dynamik ist. Hierbei ist es wichtig hervorzuheben, dass der „Gegenstand“ einer Interaktion nicht ein Objekt sein muss, sondern auch etwas wesentlich Abstrakteres sein kann, wie z.B. eine Handlung. Der dritte zu betrachtende Bereich ist die Kombination aus den beiden zuvor genannten Kombinationen, den wir als Dynamische Interaktionen mit Dynamiken bezeichnen. Dynamische Interaktionen mit Dynamiken sind sämtliche dynamischen Interaktionen, bei denen der „Gegenstand“ einer Interaktion selbst eine Dynamik ist. Ein Beispiel hierfür wäre das Herunterstoßen eines Pakets von einem sich bewegenden Laufband.

Der nächste Schritt bestand darin, ein Verständnis von jedem Gestaltungsraum der jeweils identifizierten Komponenten zu erlangen. Anfänglich trugen wir einfach im Rahmen mehrerer Brainstorming-Sessions mit Fachleuten alle Möglichkeiten zusammen. Die so ermittelte Datenmenge war extrem groß, insbesondere für den Gestaltungsraum von Dynamiken. Aus diesem Grund beschlossen wir, hierarchische Klassifizierungen für die Gestaltungsräume zu schaffen. Um die Gestaltungsräume zu visualisieren, verwendeten wir Baumstrukturen im Stil von Mind Maps. Ein Beispiel hierfür ist in Abbildung 1 zu sehen. Die Klassifizierung basierte anfangs darauf, wie der Nutzer die Komponenten wahrnimmt.

Der Gestaltungsraum von Dynamiken ist sehr umfangreich. Selbst in komprimierter, klassifizierter Form ist er zu weitreichend, um ihn hier vorzustellen. Die Interaktion ist das einzige Gebiet, welches bereits von anderen untersucht wurde. Für unsere Zwecke zogen wir für diesen Gestaltungsraum die vorliegende Arbeit von Bowman heran [BH99]. Der Gestaltungsraum dynamischer Interaktionen besteht überwiegend aus Interaktionen, die in der Community bekannt sind. Einem Großteil von ihnen wird jedoch kaum ausgiebig Beachtung geschenkt und die dynamische Natur dieser Interaktionen wurde meistens ignoriert. Der Gestaltungsraum interaktiver Dynamiken ist unten im Rahmen eines Beispiels aus unserer Arbeit kurz beschrieben. Dynamische Interaktionen mit Dynamiken sind sehr interessant, weil sie am wenigsten verstanden werden und am komplexesten sind. Dieses Untersuchungsfeld ist noch weitgehend unerforscht - wahrscheinlich aufgrund der Schwierigkeiten bei der Schaffung dieses Interaktionstyps. Selbst auf der konzeptionellen Ebene sind die meisten Möglichkeiten nicht trivial. Ohne die richtigen Werkzeuge ist es extrem schwer, mit ihnen zu experimentieren. Die derzeit verfügbaren Systeme bieten einfach keine Unterstützung für diesen Interaktionstyp. Das von uns entwickelte Supportsystem, welches wir im nächsten Abschnitt vorstellen, liefert erste Werkzeuge für solche Experimente.

2.1 Interaktive Dynamiken

Konzeptionell sind Interaktive Dynamiken recht unkompliziert. Es handelt sich bei ihnen einfach um Interaktionen mit einer Dynamik. Nichtsdestotrotz werden sie weder wirklich verstanden noch eingehend untersucht. Beispiele für Interaktive Dynamiken lassen sich im wirklichen Leben finden. Bei vielen Sportarten stehen solche Interaktionen im Mittelpunkt, z.B. das Treten eines rollenden Balls. Hierin besteht oft ein Großteil der Heraus-

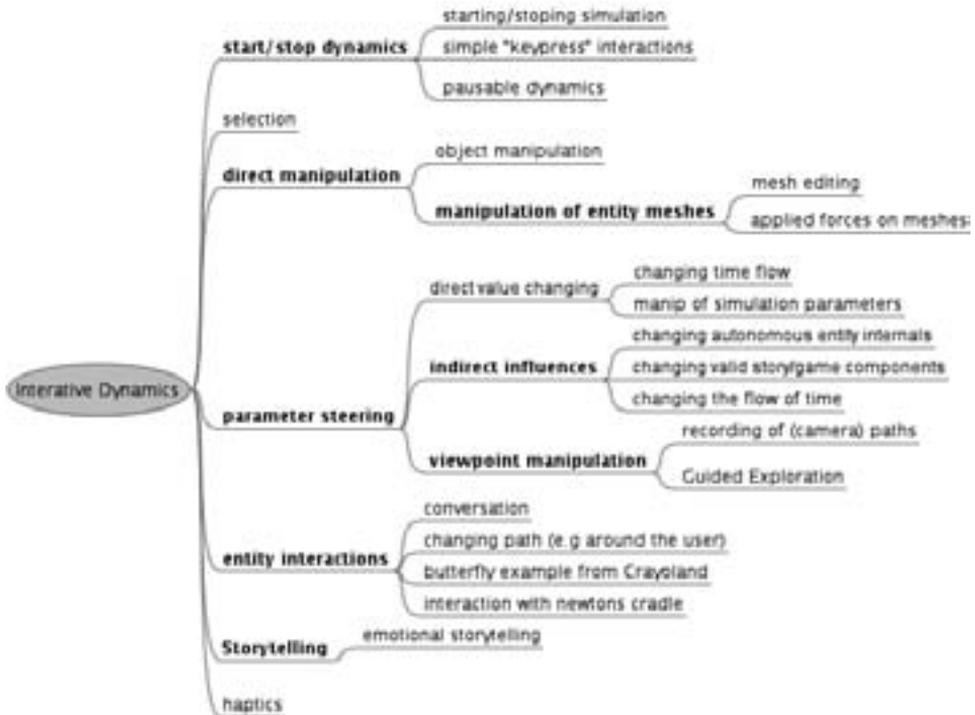


Abbildung 1: Der Gestaltungsraum interaktiver Dynamiken in Form einer Taxonomie.

forderung dieser Sportart, besonders für Jugendliche und Amateure. Doch obwohl nicht jeder aktiv Sport treibt, ist eine andere Klasse Interaktiver Dynamiken sehr verbreitet. Der „einfache“ Vorgang, ein Gespräch zu führen basiert auf Interaktionen, die in diese Kategorie fallen. Der Gestaltungsraum interaktiver Dynamiken ist in Abbildung 1 dargestellt. Der Gestaltungsraum ist scheinbar klein, obwohl die meisten Kategorien eine Vielzahl von Potenzialen einschließen. In den folgenden Absätzen erläutern wir die Möglichkeiten, die wir bei der Klassifizierung in Abbildung 1 hervorgehoben haben.

Eine Klasse Interaktiver Dynamiken ist die Interaktion mit sich bewegenden Objekten. Während wir uns leicht vorstellen können, dass wir mit Dynamiken interagieren werden, ist es schwieriger, es tatsächlich zu tun oder es auch nur in Betracht zu ziehen und zu planen. Typischerweise geschieht dies häufig in Computerspielen, da hier auf diese Weise die Herausforderung für den Spieler erhöht wird. Die Schwierigkeiten liegen in der Regel in der Implementierung. Aufgrund von Implementierungsproblemen ereignen sich Interaktive Dynamiken mit Objekten normalerweise unmittelbar. Allerdings gibt es eine ganze Klasse von Interaktionen, die andauern (d.h. Dynamische Interaktionen), die potenziell interessanter sind als die nur „unmittelbaren“ Interaktionen des Schießens und draufspringen, die klassischen Aktivitäten in Computerspielen. Diese sich überschneidenden Bereiche sind diejenigen, die wir als Dynamische Interaktion mit Dynamiken bezeichnen. Die Haptik bildet hier eine Ausnahme, da sie sich bereits seit langem mit dieser Art der In-

teraktion mit Dynamiken beschäftigt. Die Haptik stellt jedoch einen Sonderfall dar, weil haptische Lösungen immer noch unüblich sind.

Die meisten anderen Dynamischen Interaktionen beziehen sich auf solche, die sich eher auf der Metaebene abspielen. Beispielsweise erfordern die meisten sinnvollen Interaktionen mit einer „Entität“ (z.B. Avatar, Person, Tier) ein gewisses Maß an Abstraktion. Das Führen eines Gesprächs setzt voraus, dass ein dynamischer Vorgang den inneren Zustand des Gesprächsteilnehmers bestimmt. Ebenso schließen Interaktionen in einer interaktiven Applikation mit Handlung ein, dass die Interaktionen des Nutzers in einem gewissen Umfang die Dynamik dieser Handlung beeinflussen. Obgleich dies auf die Programmierung nicht zutreffen muss, vermitteln erfolgreiche Anwendungen die Illusion einer derartigen Interaktion.

3 Unterstützung der Erstellung von DIVEs

Der zweite Teil der hier präsentierten Arbeit besteht aus der Entwicklung und Implementierung einer Methode zur Unterstützung der Erschaffung von DIVEs. Im ersten Unterabschnitt geben wir einen Einblick in die Anforderungsentwicklung und fassen zusammen, wie gut existierende Virtuelle Umgebungssysteme sowie bestehende zeitbasierte Ansätze aus der Informatik diese Anforderungen erfüllen. Im zweiten Unterabschnitt beschreiben wir kurz das hieraus entstandene System Functional Reactive Virtual Reality.

3.1 Systemdesign

Im Rahmen der im vorangegangenen Abschnitt behandelten Untersuchung einigten wir uns auf Definitionen verschiedener Typen gewünschter Umgebungen. Die entwickelten Klassifizierungen sind ideal, um Gruppen von Anwendungsfällen zu erstellen. Dort kategorisierten wir die Möglichkeiten danach, wie die Anwender sie wahrnehmen. Indem wir die gesammelten Möglichkeiten auf Basis eines anderen Faktors klassifizierten, gewannen wir einen neuen Blickwinkel und konnten so eine neue Gruppe von Anforderungen erschließen. Der Schlüssel hierzu war festzustellen, dass Zeit der ausschlaggebende Aspekt ist (Interaktion bildet eine leichte Ausnahme, wobei die meisten identifizierten Interaktionen zur Untergruppe Dynamischer Interaktionen zählen). Abbildung 2 zeigt ein Beispiel für diese Klassifizierung Dynamischer Interaktionen. Die erforderlichen Darstellungen von Zeit sind in dieser Form gut erkennbar.

Eine abschließende Gruppe von Anforderungen wurde anhand der zeitbasierten Analyse, der Anwendungsfälle und der Einzelanalyse jedes DIVE-Bereichs entwickelt. Nachfolgend ist eine Auswahl der wichtigsten Anforderungen aufgeführt:

- Implementierung des Support unter Verwendung zeitlicher Darstellungen
 - andauernd
 - andauernd in Intervallen
 - diskrete Zeit (Ereignisse)

- geordnete Gruppe (andauernder) Intervalle
- geordnete Gruppe diskreter Ereignisse
- Unterstützung laufzeitabhängiger Sortierungen von Gruppen
- Unterstützung laufzeitabhängiger Veränderungen der Virtuellen Umgebung
- Fähigkeit, Zeit zu manipulieren, einschließlich der Fähigkeit
 - Zeit zum Stillstand zu bringen
 - sich selbst zu verändern, d.h. die “Geschwindigkeit” der Zeit
 - verschiedene “Geschwindigkeiten” der Zeit zu definieren, möglicherweise je Dynamik
 - Rückgängig machen für alle zeitlichen Darstellungen
- Unterstützung der Ableitung von zeitlichen Funktionen höherer Ordnung bei andauerndem Input, z.B. Geschwindigkeit und Beschleunigung
- Unterstützung mehrerer verschiedener Ausgabemodalitäten mit Aktualisierungen in unterschiedlichen Abständen
- Unterstützung hierarchischer Ebenen von Dynamischen Interaktionen

Weitere Anforderungen an ein Supportsystem für die Erstellung von DIVEs ergaben sich aus den ursprünglichen Beweggründen für das Projekt und der Softwaretechnikpraxis. Diese Anforderungen waren darauf ausgelegt, das System für einen großen Personenkreis nutzbar zu machen. Beispielsweise muss eine nicht-monolithische Architektur vorhanden sein, die in bestehenden Virtual Reality-Systemen sowie plattformübergreifend eingesetzt werden kann. Außerdem übernahmen wir zusätzlich die Anforderung, dass das System eine Darstellung der Dynamiken aufweisen muss, die dem menschlichen Zeitverständnis besser gerecht wird. Ein wesentlicher Bestandteil hiervon besteht darin, eine Mischform der Zeitdarstellung einzusetzen, d.h. sowohl andauernde als auch diskrete Ereignisse als inhärent Teile der Software. Eine weitere Anforderung war, dass der Entwickler nicht direkt mit einer framebasierten Zeitänderung (d.h. δt) zu tun haben sollte.

3.1.1 Evaluierung existierender Systeme

Eine erste Aufgabe bestand darin, die existierenden Systeme in Hinblick auf die festgestellten Anforderungen zu evaluieren. Keines der bestehenden Systeme bot Unterstützung für diese Anforderungen. Dynamiken sind hierbei als besonders wichtige Komponente hervorzuheben, für die der Support fehlt. Von einigen wenigen Ausnahmen abgesehen, behandeln sämtliche existierenden Systeme Zeit als unmittelbar und etwas, das sich nur während der Aktualisierung von Einzelbildern abspielt. Die meisten Systeme stellen nur die Programmiersprache selbst und einen „Timer“ zur Verfügung. Dies ist beispielsweise typisch für die große Klasse der Datenfluss-Systeme, zu denen die Mehrzahl der Systeme zählt. Normalerweise steht Support für andauernde Dynamiken – sofern vorhanden – nur im Rahmen vorab geplanter “keyframed” Animationen bereit. Leider schließt diese Methode die meisten Interaktionen mit der Dynamik aus. Die wenigen Systeme mit echtem Support für andauernde Dynamiken, wie das auf Constraints basierende DLoVe-System [Del00], schränken Interaktionen mit vorprogrammierten diskreten Interaktionen ein. Einige interessante Ansätze wurden bei Systemkontrollen auf höheren Ebenen gefunden, besonders beim HCSM-System [CKP95]. Bedauerlicherweise boten solche Systeme

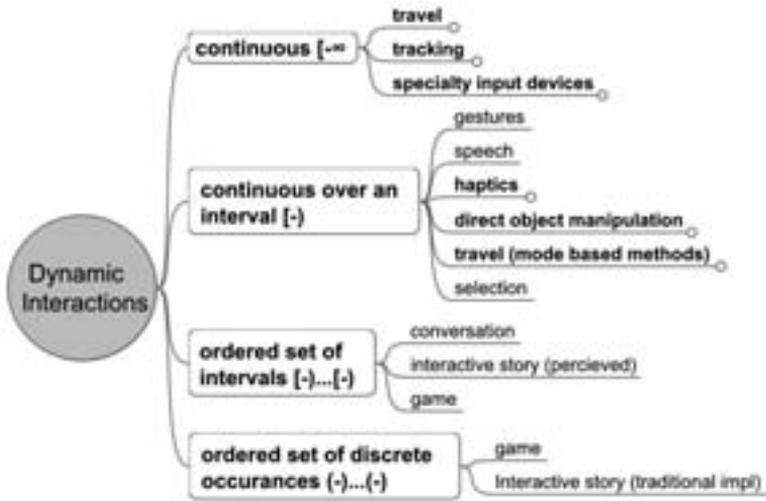


Abbildung 2: Der Gestaltungsraum Dynamischer Interaktionen klassifiziert auf Basis der Darstellung von Zeit.

kaum Unterstützung für die eigentliche Implementierung von Dynamiken und Interaktionen. Schließlich waren viele der interessantesten Systeme monolithisch und stark an veraltete Rendering-Systeme gekoppelt, wodurch eine Weiterentwicklung dieser Ansätze nicht sinnvoll erscheint. Bewertungen der unterschiedlichen Systeme sind nachzulesen unter [BB07c, Blo09].

3.1.2 Evaluierung existierender Ansätze

Da keiner der in den existierenden Softwaresystemen für Virtuelle Umgebungen realisierten Ansätze die festgelegten Anforderungen zufriedenstellend erfüllen konnte, war eine neue Herangehensweise gefragt. Ein fehlender Schlüsselaspekt war die Frage, wie Zeit behandelt wird. Die wenigen Ausnahmen, die den Zeitaspekt gut gelöst hatten, besaßen nicht die Fähigkeit oder Flexibilität, um den Anforderungen gemäß zu interagieren. Deshalb wurde eine grundlegende Auswertung der unterschiedlichen zeitbasierten Ansätze, die sich in diversen Literaturquellen finden, vorgenommen. Viele der am relevantesten erscheinenden Ansätze waren unzureichend, entweder in Hinblick auf ihre Verständlichkeit für den menschlichen Nutzer oder in Bezug auf zeitliche Aspekte beziehungsweise Interaktionsfähigkeiten. Die Suche ergab eine einzige Methode, die versprach, alle notwendigen Anforderungen zu erfüllen: Functional Reactive Programming [ESYAE94, EH97, Eli98, WH00].

Functional Reactive Programming (FRP) ist interessant, weil dieser Ansatz eindeutig für einen ähnlichen Zweck entwickelt wurde: die Definition von Dynamiken (in ihren terminologischen Funktionen). Dabei sind sie gleichzeitig in der Lage, auf Nutzereingaben zu reagieren. Die bislang fortschrittlichste Implementierung von FRP ist AFRP (Arrowized FRP) [WH00]. AFRP wurde in der rein funktionalen Sprache Haskell implementiert und

verwendet fortgeschrittene Fähigkeiten von Arrows, einem mathematischen Konstrukt aus der Kategorientheorie. Durch den Einsatz von Arrows schafft AFRP ein System, in dem Zeit eine Komponente der erster Klasse für die Beschreibung der Dynamik bildet und so die Möglichkeit bietet, ausgesprochen hoch entwickelte Funktionalitäten zu kreieren. Obgleich AFRP aufgrund eines unbekannteren Programmierstils und der Komplexität des Arrows-Konzepts schwierig zu lernen ist, verwendet die jüngste Version eine Syntax auf mathematischer Grundlage. Glücklicherweise bleiben die meisten Komplexitäten dem Durchschnittsanwender verborgen, da nur Systementwickler auf dieser Ebene mit der Sprache arbeiten müssen.

3.2 Functional Reactive Virtual Reality

Wir entwickelten ein System auf Basis des gewählten FRP-Ansatzes. Bei dem implementierten System handelt es sich um Functional Reactive Virtual Reality (FRVR) [BB07a, BB07b, BB07c]. FRVR stellt eine Erweiterung des ursprünglichen FRP-Modells zum Zweck der Erstellung von DIVE dar. FRVR wurde außerdem konzipiert, um die stärker an Bedingungen gekoppelten Anforderungen der Virtuellen Realität (VR) zu erfüllen. Außerdem ist es dafür ausgelegt, existierende VR-Systeme ebenso wie multimodale Systeme zu ergänzen, die mehrere Aktualisierungs-Loops mit verschiedenen Geschwindigkeiten durchlaufen müssen. Hierfür wurden spezielle Systeme für die Verbindung mit AFRP-basierten Modulen entwickelt. Der hieraus resultierende, lose gekoppelte Systementwurf wurde in zwei VR-Systeme integriert, denen mehrere verschiedene Ausgabesystemimplementierungen zugrunde liegen.

FRVR adaptiert und erweitert die vorhandenen Fähigkeiten von AFRP. Es bringt diverse zusätzliche Fähigkeiten für traditionelle Anwendungsgebiete ein, z.B. für Animationen, die mit Schlüsselbildern arbeiten. Das reaktive FRP-System gestattet ein Umschalten zwischen „Verhaltensweisen“, und zwar auch für erst entstehendes und nicht lineares Verhalten des Systems. FRVR bietet mittels verschiedener Funktionalitäten zusätzlich Support für die Erkundung Dynamischer Interaktionen. Sogar Zeit wird flexibel im FRVR-System gehandhabt und lässt sich für alle Verhaltensweisen manipulieren, entweder einzeln je Verhalten oder gemeinsam für eine Gruppe von Verhaltensweisen. Interaktion ist sogar möglich, während die Zeit angehalten wird. Die „Rückgängig machen“-Funktion, die FRVR als einziges System bietet, erlaubt sogar ein Zurückgehen in der Zeit. Wie bei vielen dieser Fähigkeiten, erfordert auch das Rückgängig machen lediglich das Hinzufügen einer einzigen Funktion.

4 Fazit

Dieser Artikel hat die wichtigsten Inhalte der Dissertation des Autors zusammenfasst [Blo09]. Er gewährte einen Einblick in zwei wichtige Beiträge, die hierdurch geleistet wurden: die Definition und Analyse von Dynamic, Interactive Virtual Environments (DIVEs) und die

Entwicklung eines Systems, das die Schaffung von DIVEs unterstützt. DIVEs sind diejenige Klasse Virtueller Umgebungen, welche die benötigten technischen Komponenten enthalten, um hoch erfahrbare Virtuelle Umgebungen zu gestalten. Unsere Untersuchungen der Natur von DIVEs hat den Fokus auf interessante Gebiete gelenkt, die bis jetzt noch wenig erforscht sind, wie beispielsweise die Interaktiven Dynamiken. Der zweite große Teil des Forschungsansatzes besteht darin, einen Ansatz für die Unterstützung der DIVE-Entwicklung sowie für die Entwicklung unseres Systems, Functional Reactive Virtual Reality (FRVR) herauszuarbeiten. FRVR stellt Funktionalitäten bereit, die zuvor kein anderes System bot, wie ein einfaches, zeitbasiertes Rückgängig machen und ermöglicht das Experimentieren mit sämtlichen Aspekten von DIVEs. Die Open Source FRVR-Middleware wird inzwischen von verschiedenen Forschungsgruppen eingesetzt.

Literatur

- [BB07a] Kristopher J. Blom und Steffi Beckhaus. Functional Reactive Virtual Reality. In *IPT/EGVE '07: Short Paper Proceedings of the IPT/EuroGraphics workshop on Virtual Environments*, Seiten 295–302. EuroGraphics Association, June 2007.
- [BB07b] Kristopher J. Blom und Steffi Beckhaus. Integrating Functional Reactive Programming in a High-Level VR Framework. In *Proceedings of the "Virtuelle und Erweiterte Realität" the 4th Workshop of the GI working group VR/AR*, Seiten 189–196, 2007.
- [BB07c] Kristopher J. Blom und Steffi Beckhaus. Supporting the Creation of Dynamic, Interactive Virtual Environments. In *VRST '07: Proceedings of the 2007 ACM symposium on Virtual Reality Software and Technology*, Seiten 51–54. ACM Press, 2007.
- [BH99] Doug A. Bowman und Larry Hodges. Formalizing the Design, Evaluation, and Application of Interaction Techniques for Immersive Virtual Environments. In *Journal of Visual Languages and Computing*, Seiten 37–53. Academic Press, 10 1999.
- [Blo09] Kristopher J. Blom. *Dynamic, Interactive Virtual Environments*. Sierke Verlag, 2009.
- [CKP95] James Cremer, Joseph Kearney und Yiannis Papelis. HCSM: a framework for behavior and scenario control in virtual environments. *ACM Transactions on Modelling and Computer Simulation*, 5(3):242–267, 1995.
- [Del00] Leonidas Deligiannidis. *DLoVe: A specification paradigm for designing distributed VR applications for single or multiple users*. Dissertation, Tufts University, 2000.
- [EH97] Conal Elliott und Paul Hudak. Functional Reactive Animation. In *International Conference on Functional Programming*, Seiten 196–203, 1997.
- [Eli98] Conal Elliott. Functional Implementations of Continuous Modeled Animation. In *Proceedings of PLILP/ALP '98*, Seiten 284–299, 1998.
- [ESYAE94] Conal Elliott, Greg Schechter, Ricky Yeung und Salim Abi-Ezzi. TBAG: A High Level Framework for Interactive, Animated 3D Graphics Applications. *Computer Graphics*, 28:421–434, 1994.
- [WH00] Zhanyong Wan und Paul Hudak. Functional reactive programming from first principles. In *Conference on Programming Language Design and Implementation (PLDI '00)*, Seiten 242–252. ACM SIGPLAN, ACM Press, 2000.



Kristopher J. Blom hatte während seines Bachelor Studiums in einem Forschungsprojekt bei den Argonne National Laboratories zum ersten mal Kontakt mit Virtual Reality. Nach dem Erhalt des Bachelor of Science in Computer Engineering von der University of Evansville (USA) hat er seine Forschung im Gebiet der Virtual Reality bei Dr. Carolina Cruz-Neira an der Iowa State University (USA) fortgesetzt und dabei den Abschluß Master of Science in Computer Engineering erhalten. Während des Studiums hat er bei weiteren wissenschaftlichen Projekten bei den Argonne National Laboratories und den Naval Research Laboratories mitgearbeitet. Nach seinem Master Studium hat er ein Jahr als Wissenschaftlicher Mitarbeiter bei der Virtual Reality Gruppe des Fraunhofer Instituts für Medienkommunikation (IMK) in St. Augustin verbracht. 2004 hat er die *interactive media · virtual environments* Gruppe um Dr. Steffi Beckhaus an der Universität Hamburg mitgegründet. Seine Aufgaben als wissenschaftlicher Mitarbeiter beinhalteten den Aufbau eines Virtual Reality Labors, Lehre und Forschung. Nachdem er 2009 seine Dissertation an der Universität Hamburg abgeschlossen hat, hat er seine Arbeit als post-doktoraler Mitarbeiter bei der *interactive media · virtual environments* Gruppe fortgesetzt.