



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Matthias Wanner Institut für Kartografie ETH Zürich

Konzipierung einer interaktiven Steuerung für den Schweizer Weltatlas

Bachelorarbeit

Frühjahressemester 2008

Adrian Weber
Prof. Lorenz Hurni
Institut für Kartografie
ETH Zürich

Vorwort

Bei der Wahl des Themas meiner Bachelorarbeit war mir wichtig, dass es einerseits praxisorientiert und andererseits innovativ und neu ist. Deshalb habe ich mich für das Thema: "Konzipierung einer interaktiven Steuerung für den Schweizer Weltatlas" entschieden. Der Hauptteil dieser Arbeit setzte sich aus Programmieren am Computer und dem Arbeiten mit modernen Technologien zusammen. Dies empfand ich als sehr spannend. Besonders freut mich, dass das Ergebnis dieser Arbeit in die Entwicklung des „Schweizer Weltatlas“ interaktiv einfließt.

An dieser Stelle möchte ich mich beim Institut für Kartografie, insbesondere bei Adrian Weber, für die tolle Unterstützung bedanken.

Ein Dank geht auch an die Entwickler diverser Open-Source Projekte, welche diese Arbeit durch ihre zahlreichen und kostenlos zur Verfügung gestellten Pakete erst ermöglicht haben.

Zusammenfassung

Für die interaktive Version des Schulatlas „Schweizer Weltatlas“ wird ein virtueller Globus entwickelt. Im Fokus dieser Bachelorarbeit steht die Programmierung einer Fernsteuerung für den virtuellen Globus des „Schweizer Weltatlas“. Ziel dieser Arbeit ist, dass die Lehrpersonen in Zukunft einfach und intuitiv Karten präsentieren können, ohne dabei direkt an einem Computer sitzen zu müssen. Ergebnis dieser Arbeit ist eine Java Bibliothek, welche einerseits die Steuerung mit einem bestimmten Gamepad und Joystick erlaubt und andererseits die Möglichkeit offen lässt, beliebig viele unterschiedliche Controller zu implementieren.

Abstract

For the interactive version of the school atlas “Swiss World Atlas” a virtual globe is developed. This Bachelor thesis focuses on programming a remote control for the virtual globe of the “Swiss World Atlas”. Its goal is that teachers can present maps simply and intuitively in the future, without having to sit directly at a computer. Result of this work is a Java library, which provides the control with a certain gamepad and joystick as well as the possibility of implementing as many as desired different controllers.

Inhaltsverzeichnis

VORWORT	2
ZUSAMMENFASSUNG	3
ABSTRACT	3
1 EINLEITUNG	6
1.1 Einführung in die Problemsituation	6
1.2 Konzept und Inhaltsübersicht der Arbeit.....	6
2 GRUNDLAGEN, METHODE UND VORGEHEN.....	7
2.1 Ausgangslage	7
2.1.1 Nasa World Wind	7
2.1.2 Globus des „Schweizer Weltatlas“	8
2.2 Technische Grundlage	9
2.2.1 Schnittstellen.....	9
2.2.2 Hardwareschnittstelle	9
2.2.3 Mensch-Maschinen-Schnittstelle	12
2.3 Arbeitsmittel.....	16
2.3.1 Programmiersprache.....	16
2.3.2 Weitere Hilfsmittel	16
2.4 Gewählte Methode	17
2.4.1 Anforderungen an das Eingabegerät.....	17
2.4.2 Realisierbare Eingabegeräte.....	17
2.4.3 Gewähltes Eingabegerät.....	18
2.5 Vorgehen	18
2.5.1 Programmierung eines einfachen Programms mit Jinput.....	18

3	ERGEBNISSE	23
3.1	Word Wind Remote Control	23
3.1.1	Steuerung	23
3.1.2	Programm.....	25
3.1.3	Funktionsweise.....	27
3.2	SWA Anbindung	27
3.3	Evaluation	28
3.3.1	Beschreibung.....	28
3.3.2	Erster Teil.....	28
3.3.3	Zweiter Teil.....	29
3.3.4	Diskussion der Befragung	29
4	AUSBLICK	30
5	LITERATURVERZEICHNIS	31
6	ABBILDUNGSVERZEICHNIS	32

1 Einleitung

1.1 Einführung in die Problemsituation

Der „Schweizer Weltatlas“ ist ein in der Schweiz weitverbreiteter Schulatlas für die Mittelschule. Die gedruckte Version wird alle fünf Jahre erneuert und erscheint in den drei Schweizer Landessprachen Deutsch, Italienisch und Französisch. Der Atlas enthält zahlreiche physische und thematische Karten mit den geografischen Schwerpunkten Schweiz und Europa (Marty 2007).

Im Moment wird für diesen Atlas eine interaktive Version entwickelt, welche unter anderem ein Programm beinhaltet, das dem Lehrer erlaubt, einen Globus mit Weltkarten interaktiv zu steuern. Dieser Globus kann mittels Beamer an eine Wand projiziert werden, um die Karten im Unterricht den Schülern präsentieren zu können. Bisher funktionierte die Steuerung mit Maus und Tastatur und geschah statisch an einem Arbeitsplatz. Neu soll die Steuerung nicht an einen Platz gebunden, sondern von anderen Standorten aus möglich sein. Um diese Fernsteuerung zu entwickeln und interaktiv zu gestalten, wurde diese Bachelorarbeit geschrieben.

1.2 Konzept und Inhaltsübersicht der Arbeit

In erster Linie ist das Ziel dieser Arbeit, eine optimale Fernsteuerung für den interaktiven Globus zu erhalten. Dies zu erreichen, bedingt eine umfassende Internetrecherche und ein Einarbeiten in die Programmiersprache Java.

Die Arbeit ist folgendermassen aufgebaut: Im ersten Teil werden die allgemeinen Eingabegeräte eines Computers und deren Schnittstellen erläutert. Darauf fällt im zweiten Teil die Auswahl auf einen bestimmten Joystick respektive Gamepad. Diese Eingabemethoden werden dann mittels der Programmiersprache Java im interaktiven Globus NASA World Wind implementiert. Das entstehende Produkt lässt sich wiederum im interaktiven Globus des „Schweizer Weltatlas“ als Paket integrieren. Daraus folgt das Ergebnis einer Fernsteuerung (engl. Remote Controller) für diesen Globus. Im letzten Teil wird die Fernsteuerung anhand einer kleinen Evaluation geprüft und diskutiert.

2 Grundlagen, Methode und Vorgehen

2.1 Ausgangslage

2.1.1 Nasa World Wind

Die dreidimensionale Visualisierung ist eine sehr beliebte Methode, um die Topografie einer Landschaft darzustellen. Gegenüber einer zweidimensionalen Karte repräsentiert das Relief einer solchen Visualisierung das Terrain sehr gut und die Realität lässt sich damit relativ genau abschätzen.

Google hat eine solche Visualisierung erstellt, welche jedem Computerbenutzer mit Internetverbindung erlaubt, die Welt in einem virtuellen Modell zu betrachten (Google 2008).

Die NASA ihrerseits hat im Jahre 2004 ebenfalls eine solche Software entwickelt. Im Grunde ist NASA World Wind ein Programm, welches laufend Fernerkundungs-Bilddaten vom Internet abrufen kann und diese dann auf einen Globus projiziert.



Abbildung 1 NASA World Wind .Net Windows Anwendung

Die Performance von NASA World Wind kommt am besten im Gebiet der USA zu Geltung, da dort hochauflösende Satellitenbilder vorhanden sind. Die Darstellung der restlichen Welt ist eher grob aufgelöst und eignet sich weniger zur detaillierten Ansicht, als vielmehr zur allgemeinen Übersicht. Das Höhenmodell der NASA ist um einiges mächtiger als dasjenige von Google Earth, da es noch zusätzliche Optionen, wie zum Beispiel das stereografische 3D-Sehen, ermöglicht. Durch die Auswahl verschiedener Satellitenbilder ergibt sich die Möglichkeit zur Betrachtung von anderen Planeten (Reibold, 2005).

Der wohl wichtigste Punkt ist, dass die NASA eine Java Development Version anbietet, welche dem Benutzer eine Einbindung des Programms für eigene Zwecke ermöglicht. Dies ist darauf zurückzuführen, dass die installierbare .Net Anwendung nur für Windows entwickelt ist und nicht auf anderen Plattformen läuft. Durch diese gratis verfügbare Java Bibliothek sind die Entwicklungsmöglichkeiten sehr vielseitig und der Globus inklusive Kartenmaterial kann in jedes dazu vorgesehene Programm implementiert werden.

Als Fazit ist zu sagen, dass mit NASA World Wind eine wissenschaftliche Version von Google Earth geschaffen wurde, die wesentlich mehr Einstellungen und Differenzierungen erlaubt und sich somit für diverse Geo-Applikationen eignet (WorldWind, 2008).

2.1.2 Globus des „Schweizer Weltatlas“

Die NASA bietet den Programmiercode des Globus gratis an. Somit lassen sich auf der NASA Java Development Version beliebig viele „Planeten“ mit beliebig vielen eigenen Karten oder Bildern programmieren. Dies ermöglicht, statt wie bisher alle Karten des „Schweizer Weltatlas“ in Buchform herauszubringen, diese anhand einer virtuellen Erdkugel darzustellen und so eine ganz andere Perspektive zu erhalten. Das Ziel dieses Projekts ist eine Version zu erarbeiten, die je nach Auswahl ökonomische, topografische oder klimatische Welt- und Kontinentalkarten auf den Globus abbildet. Momentan befindet sich der Globus des „Schweizer Weltatlas“ noch als Prototyp in Entwicklung.

2.2 Technische Grundlage

2.2.1 Schnittstellen

Eine Schnittstelle ist eine Verbindungsstelle zweier unterschiedlich arbeitender Teilsysteme. Sie ermöglicht den Austausch von Daten, Signalen und Ähnlichem. Allgemein unterscheidet man zwischen drei unterschiedlichen Schnittstellen:

- Hardwareschnittstelle (Steckverbindungen, Verbindungskabel)
- Softwareschnittstelle (Kommunikation zwischen zwei Programmen)
- Mensch-Maschinen-Schnittstelle (Grafische Benutzeroberflächen und Eingabegeräte wie Maus und Tastatur)

In den folgenden Kapiteln wird auf die Hardwareschnittstelle, welche den Verbindungskabeln oder Verbindungsarten der unterschiedlichen Eingabemethoden entspricht, eingegangen. Im Weiteren werden die Eingabegeräte der Mensch-Maschinen-Schnittstelle diskutiert (MeyersOnline 2008).

2.2.2 Hardwareschnittstelle

Gameport

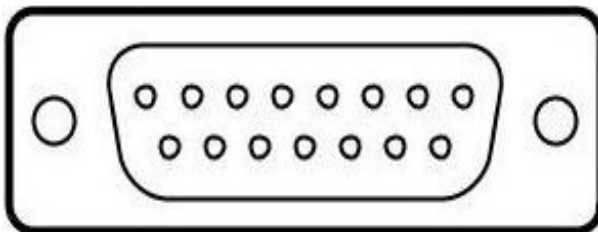


Abbildung 2 Gameport Anschluss

Der Gameport war früher der Hauptanschluss für Joysticks oder Gamepads. Die Konzeption erlaubt es, seriell vier Achsen und vier Buttons abzufragen. Da dies bei neueren Geräten nicht mehr ausreicht, wird dieser Anschluss in Zukunft nicht mehr

vorhanden sein. Neuere Betriebssysteme wie Windows Vista unterstützen diesen Anschluss nicht mehr (Electronics57 2008).

USB

Der Universal Serial Bus (USB) ist ein standardisierter vier-poliger I/O-Bus, der als serielle Schnittstelle für verschiedene Geräte dient.

An einer einzigen USB-Schnittstelle können mittels HUB bis zu 127 Geräte angeschlossen werden. Die Übertragungsrate hat sich seit der ersten Version deutlich verändert.

- USB 1.1 Low-Speed-Version (1.5 Mbit/s → Anschluss von: Tastatur, Maus, Modem)
- USB 1.1 Standard (12 Mbit/s → Anschluss von: Drucker)
- USB 2.0 (480 Mbit/s → Anschluss von externen Festplatten, Scanner, Memory Sticks, Digitalkameras)



Abbildung 3 USB Steckerarten

Die Konzeption dieses Anschlusses erlaubt es, am laufenden System Geräte anzuschliessen und wieder zu entfernen (Plug and Play).

Die Datenübertragung erfolgt paketweise, wobei der PC oder der HUB die Steuerung und Zuordnung der einzelnen Pakete übernimmt.

Das USB Verbindungskabel ist ein ungeschirmtes (für die Low-Speed-Version) oder ein abgeschirmtes (für USB 1.1 und 2.0) Kabel, in welchem vier

Adern verlaufen. Zwei davon dienen der Übertragung von Differenzsignalen, die anderen der Stromversorgung.

Grundsätzlich sind drei verschiedene Steckerarten vorgesehen (Abbildung 3).

Zukünftige Entwicklungen sollen neben einem Wireless USB, welches mit einer Übertragungsrate von 480 Mbit/s 10 Meter weit senden kann, auch die Version 3.0 hervorbringen, die eine Übertragungsrate von 5 Gbit/s erreicht (USB 3.0).

USB ist heutzutage der Standard für alle verkabelten Eingabegeräte. Tastatur und Maus, welche früher mit dem PS/2 Anschluss über eine eigene Schnittstelle verfügten, werden über USB betrieben (IT-Wissen 2008).

Bluetooth

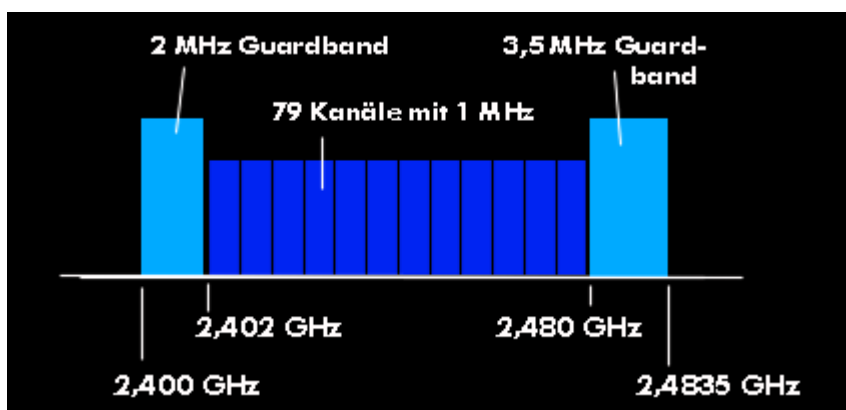
Das Bluetooth Protokoll, benannt nach dem Wikinger Blauzahn, der durch seine ausserordentliche Kommunikationsfähigkeit die zerstrittenen nordischen Völker vereinigte, ist ein

Standard für die Kommunikation in Gebäuden mittels Funk. Die 1998 von diversen grossen Informatik Firmen wie Ericsson, IBM, Toshiba und Intel entwickelte Schnittstelle, hatte das Ziel, die Kurzstrecken-Kommunikation zwischen Endgeräten zu revolutionieren.

Bluetooth wird in drei Klassen unterschieden. Je mehr Reichweite das Gerät hat, desto grösser wird auch dessen Stromverbrauch.

- Klasse 1 Sendeleistung 100mW Reichweite 100m
- Klasse 2 Sendeleistung 2.5mW Reichweite 50m
- Klasse 3 Sendeleistung 1mW Reichweite 10m

Bluetooth arbeitet im Mikrowellenbereich mit den Frequenzen ab 2.402 GHz bis und mit



2.480 GHz. Die Kanalbreite der 79 Kanäle entspricht 1 MHz, wobei sie bis zu 1'600mal pro Sekunde zwischen den 79 Frequenzen springen kann. Die Bluetooth Technologie stellt einen breitbandigen Verbindungs-

Abbildung 4 Bluetooth Frequenzbereich

kanal für die Übertragung von Daten und gesprochener Sprache zur Verfügung. Es lassen sich synchron bis zu drei Sprachkanäle mit einer Übertragungsrate von je 64 kbit/s betreiben. Für den Datenverkehr gibt es eine asymmetrische Lösung mit bis zu 723.2 kbits/s. Die neueste Generation des Bluetooth (Version 2.0) erlaubt eine Übertragungsrate von ungefähr 2.1 Mbit/s.

Speziell zu erwähnen ist noch, dass die beiden kommunizierenden Geräte untereinander abgeglichen werden müssen. Zu diesem Zweck wird von jedem Gerät ein 48bit langes Master Synchronisierungssignal gesendet, das eine eindeutige Identifizierung des Bluetooth-Geräts ermöglicht.

Viele Eingabegeräte, welche eine drahtlose Verbindung mit dem PC nutzen, laufen über die Bluetooth Schnittstelle (IT-Wissen 2008).

Infrarot

Eine weitere Alternative zu den gängigen Drahtlos-Techniken ist die Nutzung von Licht im Infraroten Bereich. Die Infrarot Technologie dient mehr der Steuerung von Geräten wie Fernseher als dem eigentlichen Datentransfer. Die Übertragungsgeschwindigkeiten sind sehr langsam. Ein grosser Nachteil der Technik ist, dass zwischen Sender und Empfänger eine Sichtverbindung bestehen muss (IT-Wissen 2008).

2.2.3 Mensch-Maschinen-Schnittstelle

Tastatur

Die Tastatur ist eines der wichtigsten Eingabegeräte für die Übermittlung von Buchstaben, Ziffern, Zeichen und Funktionen. Die Tastatur ist standardisiert und umfasst sowohl einen numerischen wie auch einen alphanumerischen Teil. Im weiteren gibt es noch einige Steuer-tasten.

Mit einer sogenannten QWERTZ-Tastatur können der gesamte ASCII-Zeichensatz und zusätzlich noch spezielle Zeichen übergeben werden. Die Tastatur kennt nur das An/Aus Prinzip, was zur Folge hat, dass die Tastatur rein digital arbeitet. Das Eingabegerät kann also nur Befehle weitergeben, wenn eine Taste gedrückt wird. Somit sind nur zwei Zustände möglich, entweder „Taste gedrückt“ oder „Taste nicht gedrückt“. Der Anschluss einer Tastatur erfolgt heutzutage über USB oder Funk, während früher vor allem die PS/2 Schnittstelle verwendet wurde (IT-Wissen 2008).

Maus

Der Durchbruch der Maus gelang, als grafische Benutzeroberflächen zum Einsatz kamen. Sie wurde konzipiert, um Befehlseingaben zu vereinfachen und kann im zweidimensionalen Raum geführt werden, um zum Beispiel einen Cursor auf dem Bildschirm zu bewegen.

In der Maus befindet sich eine Rollkugel, welche zwei Walzen in horizontaler und vertikaler Richtung dreht. Die Walzen wiederum geben relative oder absolute Positionsangaben an den PC weiter.

Dadurch ist es auch möglich, Beschleunigungsmessungen durchzuführen, was gerade bei der Steuerung von interaktiven Globen von Bedeutung ist. Neben diesen Positionssensoren befinden sich noch diverse Tasten auf der Maus, die zur Auswahl oder Einschränkung von Icons oder zum Funktionsaufruf genutzt werden können. Die neueren Mäuse verfügen über einen optischen Sensor, der die klassische Kugel ersetzt. Der grosse Vorteil daran ist, dass die Maus auf fast allen Oberflächen gesteuert werden kann und die Auflösung merklich verbessert wird. Dabei tastet eine Leuchtdiode den Untergrund ab und wandelt Bewegungen in elektrische Impulse um, welche an das System weitergeleitet werden.

Die Maus ist über die USB Schnittstelle oder über Funk mit dem Rechner verbunden. Die früher gebräuchlichsten Anschlüsse waren die serielle oder die PS/2-Verbindung (IT-Wissen 2008).

Joystick



Abbildung 5 Logitech Attack3 Joystick

Der Joystick hat seine Herkunft den Flugzeugen zu verdanken, die seit Anbeginn ihres Bestehens auf die bewährte Steuerung dieser Art von Knüppel zählten.

Mit der Optik eines Automobil-Gangschalthebels hatte der traditionelle Joystick zwei analoge Achsen und eine Vielzahl (abhängig vom Markentyp) an Buttons. Heute wird gerade für Flugzeugspiele zusätzlich ein analoger Regler eingebaut, der zur Kontrolle des Schubmechanismus dient.

In den beiden Bewegungsachsen sind Drehwiderstände eingebaut, welche die entsprechende Richtungsänderung durch Kontaktbildung an den

Computer weitergeben. Die dritte Achse kommt ohne Gegendruck aus und lässt sich wie ein Mousrad bedienen. Die Verbindung mit dem Computer war zu Beginn mittels des Gameports geregelt, wurde dann aber vom USB Anschluss abgelöst. Das Gerät gilt als das wohl meistgenutzte Spieleingabeinstrument (IT-Wissen 2008).

Gamepad

Der Gamepad hat seinen Ursprung in der Welt der Konsolen, wo er zum ersten Mal bei Nintendo auftauchte. Der Vorteil eines solchen Gerätes ist die Möglichkeit, es komplett in beiden Händen zu halten und sich damit, sofern es die Schnittstelle zwischen Gerät und Computer erlaubt, frei im Raum zu bewegen. In der ersten Version des vor allem für die Spielkategorie Jump'and'Runs gedachten Eingabegeräts, war neben Buttons ein Steuerkreuz vorhanden, welches jeweils in eine Richtung gedrückt werden konnte und somit nur von digitaler Eingabe-Natur war. Dies war ein Nachteil für Simulationsspiele, die eine analoge Eingabe bevorzugen (Flugsimulator), jedoch gerade für Sport und Actionspiele, welche auf der Konsole äusserst beliebt waren, sehr geeignet. Erst mit der Einführung der Spielkonsole Nintendo 64 bekam ein zusätzlicher analoger Stick Platz auf dem Pad.

Analog bedeutet, dass nicht nur die zwei Zustände An und Aus übergeben werden können, wie dies bei der Tastatur der Fall ist. Je nach Druckstärke in eine Stickrichtung wird ein Wert von „keine Eingabe“ stufenlos bis zu „volle Eingabe“ weitergegeben. Diese stufenlose Eingabe ermöglicht eine viel genauere Kontrolle von Bewegungen. Ein Beispiel wäre etwa die Steuerung einer Automobillenkung, welche nicht nur die Befehle „Räder gerade“ und „Räder voll eingeschlagen“ kennt, sondern auch alle Zwischenwerte. Die Kombination von digitaler Eingabe mittels Button und dem analogen Stick setzte sich durch und ist heute nicht mehr wegzudenken.

Konventionelle Gamepads verfügen über bis zu vierzehn Buttons, ein digitales Steuerkreuz und zwei analoge Sticks mit insgesamt vier analogen Achsen.

War früher die Verbindung über den Gameport geregelt, wurde sie ähnlich dem Joystick vom USB-Port abgelöst. Die neusten Generationen sind Wireless verbunden und nutzen die Bluetooth Funk Technologie.

Spezielle Eingabegeräte

- *Wii-Remote-Controller*



Abbildung 6 Wii Remote

Die Wii-Fernbedienung ist das Eingabegerät der Wii-Konsole von Nintendo. Der grösste Unterschied zu konventionellen Gamepads besteht darin, dass die Wii-Remote die Armbewegung des Spielers im Raum registriert und anhand dieser die Eingaben im Spiel berechnet.

Mit Hilfe zweier Referenzpunkte in einer Sensorleiste, welche auf dem Fernseher platziert wird, und einer Infrarotkamera an der Vorderseite des Controllers, wird die Position und Lage des Eingabegeräts relativ zum Bildschirm bestimmt. Zusätzlich enthält der Controller einen Beschleunigungssensor. Dieser ermöglicht Bewegungen in allen drei Dimensionen zu erfassen.

Die Übergabe der Befehle an die Konsole ist mit Bluetooth geregelt. Den Wii-Controller gibt es nur für die Wii-Konsole und er ist für gängige PCs nicht erhältlich (Wii.Nintendo 2008).

- *Datenhandschuh*

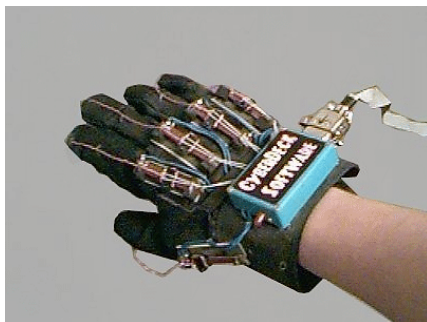


Abbildung 7 Datenhandschuh

Um eine Handbewegung zu erfassen und in Steuersignale umzusetzen sind viele Sensoren nötig, die alle Bewegungs-, Beugungs- und Positionsänderungen detektieren. Die exakte Position der Hand wird bestimmt und auf den 3D-Raum des PCs projiziert.

Datenhandschuhe sind weniger für die Spielindustrie, als viel mehr für die Medizin oder Robotik geeignet (IT-Wissen 2008).

2.3 Arbeitsmittel

2.3.1 Programmiersprache

Die Programmiersprache, die für diese Arbeit verwendet wird, ist Java der Firma Sun Microsystems in der Version 1.6.0. Folgende Punkte wirken auf diese Wahl ein:

- Plattformunabhängig
- Objektorientiertes Programmieren
- NASA World Wind in dieser Sprache geschrieben

2.3.2 Weitere Hilfsmittel

Netbeans

Als Entwicklungsplattform wird Netbeans in der Version 6.0.1 gewählt, da es primär eine Java-Entwicklungsumgebung ist (Netbeans2008).

Bibliotheken

Für das Interaktive Steuerungsprogramm sind noch zwei weitere Bibliotheken zu Hilfe genommen:

- WorldWind
WorldWind ist die von der NASA zur Verfügung gestellte Open-Source Java-Bibliothek. Sie beinhaltet Klassen und Methoden die zur Programmierung und Steuerung eines Globus in der Art des NASA World Wind Programms nötig sind und mit der auch der Globus des Schweizer Weltatlas arbeitet (WorldWind 2008).
- Jinput
Jinput ist ebenfalls eine Open-Source Bibliothek, die es ermöglicht, Eingaben von verschiedenen Eingabegeräten abzufragen und diese in vordefinierten Zeitabständen dem System zu übergeben (Jinput 2008).

2.4 Gewählte Methode

2.4.1 Anforderungen an das Eingabegerät

Um den Anforderungen der Aufgabe zu genügen, muss das Gerät gewisse Funktionen unterstützen, die direkt mit der Globus-Steuerung im Einklang stehen. Bisher wurden die vier wichtigsten Elemente mit der Maus gesteuert.

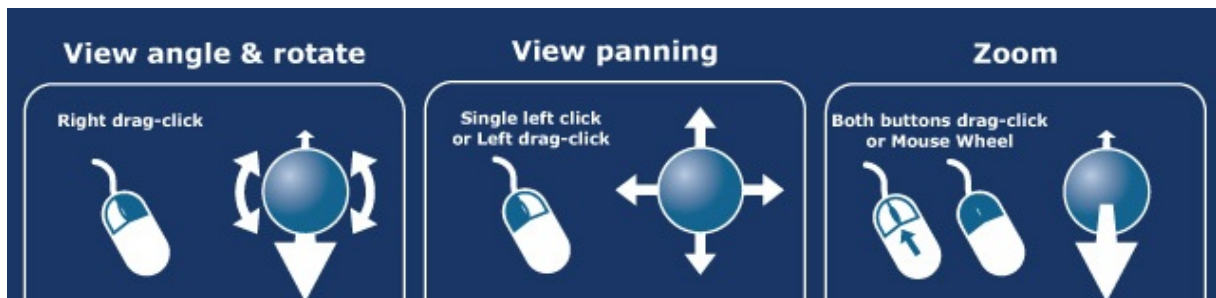


Abbildung 8 Steuerungsmöglichkeiten von NASA World Wind

Es sind dies im Groben die Funktion des Panning (Bewegen), die des Pitch (Kippen), des Zooms und des Heading (Rotieren).

Weiter ist zu nennen, dass gerade Rotation und Zoom analog übergeben werden sollten, da eine stufenlose Eingabe möglich sein soll.

2.4.2 Realisierbare Eingabegeräte

Um den Anforderungen gerecht zu werden, muss zwischen der Vielzahl von Eingabegeräten ausgewählt werden. Folgende drei Geräte entsprechen den Anforderungen:

- Joystick (drei analoge Achsen, viele Buttons)
- Gamepad (vier analoge Achsen, viele Buttons)
- Datenhandschuh (viele analoge Achsen, keine Buttons)

2.4.3 Gewähltes Eingabegerät



Abbildung 9 SpeedLink GamePad

Die Wahl fiel auf das Gamepad der Firma Speed Link, welches den Anforderungen sehr gut entspricht. Die vier analogen Achsen der zwei Sticks erlauben eine vielfältig programmierbare Steuerung der Grundfunktionen. Die beiden Sticks können ihre Werte unabhängig voneinander dem System weitergeben. Eine Vielzahl von Buttons ist ebenfalls vorhanden und die Verbindung über das USB Kabel ermöglicht einen plattformunabhängigen Einsatz. Das Gamepad entspricht

vom Design einem *SonyPlaystation2* Controller und ist durch seine Kompaktheit sehr handlich und portabel. Im Verlauf der Arbeit wird als weiteres Eingabegerät noch ein Logitech Attack3 Joystick implementiert (vgl. Abbildung 5).

2.5 Vorgehen

2.5.1 Programmierung eines einfachen Programms mit Jinput

In einem ersten Teil geht es darum, sich mit der Bibliothek Jinput vertraut zu machen. Dies wird anhand eines Programms gemacht, welches in erster Linie die Befehle des Gamepads herausliest und sie in Form eines Frames grafisch darstellt.

Erste Schritte

Die Jinput Homepage stellt dem Benutzer neben den eigentlichen Bibliotheken auch drei Programme zur Verfügung, welche erste Informationen über die angeschlossenen Eingabegeräte liefern. Eines dieser Programme, der *ControllerReadTest*, liefert als Output eine Darstellung der einzelnen Gamepad-Komponenten und deren momentanen Eigenschaften. In der folgenden Abbildung ist zu sehen, wie der Output in der vorgegebenen Situation aussieht.

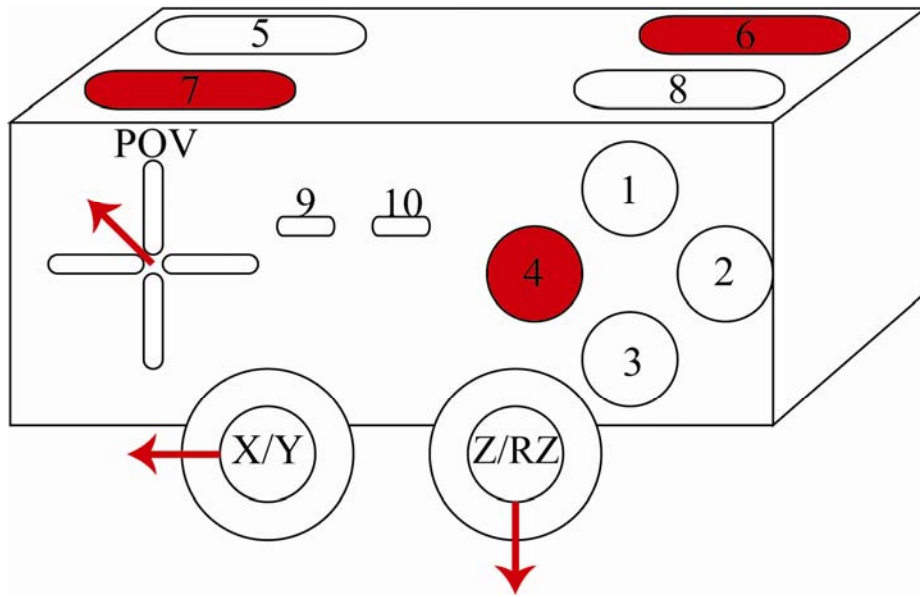


Abbildung 10 Schematische Darstellung des Gamepads in einer bestimmten Situation

Generic USB Joystick			
Z-Rotation(rz)	Z-Achse(z)	Z-Achse(z)	Y-Achse(y)
1.0	-1.5258789E-5	-1.0	-1.5258789E-5
X-Achse(x)	Mehrwegeschalter(pov)	Taste 0(0)	Taste 1(1)
-1.0	UP+LEFT	OFF	OFF
Taste 2(2)	Taste 3(3)	Taste 4(4)	Taste 5(5)
OFF	ON	OFF	ON
Taste 6(6)	Taste 7(7)	Taste 8(8)	Taste 9(9)
ON	OFF	OFF	OFF
Taste 10(10)	Taste 11(11)		
OFF	OFF		

Abbildung 11 Output zu Abbildung 10

Diese Darstellung hilft bei der Definition der Achsen und Knöpfe und gibt eine erste Information, wie der Output der Achsen aussieht.

- X-, Y-, Z-, und RZ- Achse geben Werte von -1 bis +1 aus, wobei sie in der Grundstellung auf dem Wert 0 oder beinahe 0 verharren. Wird der Stick nur zu circa 50% in eine Richtung bewegt, so entspricht die Ausgabe +/- 0.5.
- Die Buttons null bis elf sind als Boolean-Variablen definiert und zeigen entweder die Information richtig oder falsch an.
- Das Steuerkreuz (POV) kann acht verschiedene Richtungen anzeigen.

Programm

Im Buch von *Andrew Davidson (2007)* ist ein Beispiel aus vier Java Klassen gegeben, das einen weiteren Einblick in Jinput liefert. Die Klassen sind folgendermassen benannt:

- *ButtonsPanel.java*
- *CompassPanel.java*
- *GamePadController.java*
- *GamePadViewer.java*

Die Klasse *GamePadController* ist sehr nützlich, da sie einen Code beinhaltet, der automatisch den Gamepad erkennt. Des Weiteren werden die Buttons, das Steuerkreuz und die beiden analogen Sticks erkannt und definiert. Die Klasse konnte mit leichter Abänderung für das nachfolgende Programm verwendet werden.

Das Grundprogramm beinhaltet insgesamt 3 Klassen.

- *GamePadController.java*

Die *GamePadController* Klasse ist wie schon oben genannt in erster Linie für die Definition des Gamepads verantwortlich. Sie erkennt grundlegende Informationen über die verschiedenen Buttons, Sticks und die Schnittstelle und gibt diese auf dem Bildschirm aus. Für das SpeedLink Gamepad sieht dieser Output folgendermassen aus:

```
Game pad index: 2
Game controller: Generic USB Joystick , Stick
Num. Components: 18
Found X-Achse; index: 4
```

```
Found Y-Achse; index: 3
Found Z-Achse; index: 1
Found Z-Rotation; index: 0
Found Mehrwegeschalter; index: 5
Found Taste 0; index: 6
Found Taste 1; index: 7
Found Taste 2; index: 8
Found Taste 3; index: 9
Found Taste 4; index: 10
Found Taste 5; index: 11
Found Taste 6; index: 12
Found Taste 7; index: 13
Found Taste 8; index: 14
Found Taste 9; index: 15
Found Taste 10; index: 16
Found Taste 11; index: 17
```

Im zweiten Teil der Klasse wird die Funktion *poll* definiert, welche den Gamepad in einem vordefiniertem Zeitintervall abfragt. Die Methoden *getStickDir* fragen die Position der jeweiligen Achse ab. Zuletzt werden noch die Buttons abgefragt, um von jedem Button die Boolean-Variable zurückzubekommen.

- *makeGui.java*

Die Klasse *makeGui* ist verantwortlich für das Graphical Interface. Neben der Definition des Frame Rasters und der Initialisierung der einzelnen Textfelder, transformiert sie Zahlen, welche von der Klasse *Beispielprogramm* kommen, in Prozentzahlen. Je nach Stärke des Neigungswinkels wird in eine bestimmte Stick-Richtung eine äquivalente Prozentzahl angezeigt. Im untersten Fenster wird noch dargestellt, welcher Knopf im Moment gedrückt wird.

- *Beispielprogramm.java*

Die *Beispielprogramm* Klasse, welche auch die *main* Methode beinhaltet, initialisiert den Gamepad, startet den *pollTimer* für die zyklische Abfrage, gibt die Befehle an die anderen Klassen weiter und initialisiert das Graphical Interface. In der folgenden Abbildung ist ein Beispiel des Frames zu sehen, wobei im linken Teil der gegebene Input des Gamepads veranschaulicht ist. Der Stick der X/Y Achse wird dazu ganz in die linke obere Ecke gedrückt, der linke Z/RZ Stick dagegen mit etwa halber Kraft in die rechte untere Ecke. Zusätzlich wird

noch der Button 5 gedrückt. In der rechten Abbildung ist der Output zu sehen, welcher genau der beschriebenen Eingabe entspricht.

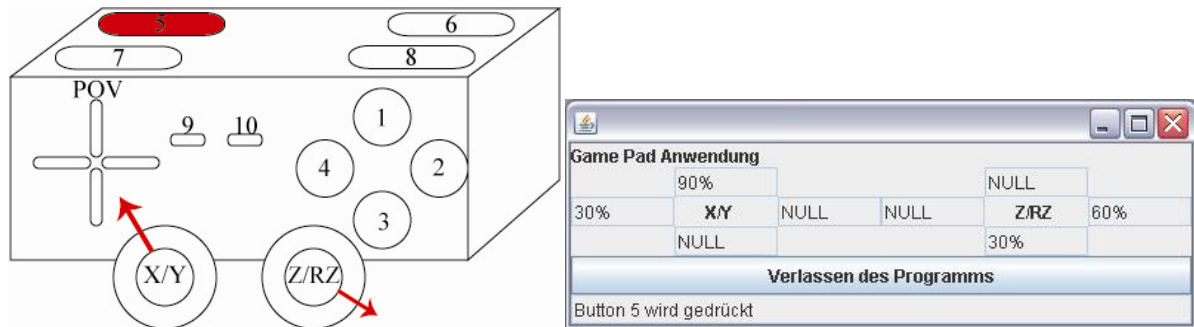


Abbildung 12 Eingabe und Ausgabe im Beispielprogramm

Programmablauf

In der folgenden Abbildung ist ersichtlich, wie das Programm funktioniert. Das Gamepad wird mit dem *pollTimer* abgefragt, welcher in der Abbildung mit „Abfrage ob Veränderung“ bezeichnet wird. In diesem Intervall von jeweils 40 Millisekunden wird das Gamepad auf Veränderungen überprüft. Falls der ursprüngliche Zustand nicht verändert wurde, so passiert nichts und der Zyklus beginnt von neuem. Kann jedoch eine Veränderung registriert werden, so wird die Eingabe abgefragt, die Werte werden transformiert, dem grafischen Interface weitergegeben und danach auf dem Bildschirm ausgegeben.

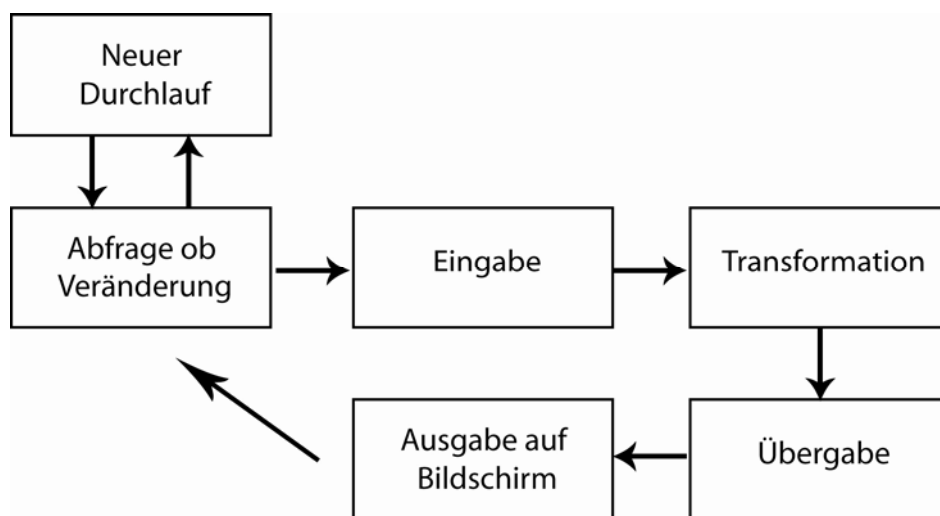


Abbildung 13 Schematischer Programmablauf

3 Ergebnisse

Das Globussteuerungs-Programm welches im Kapitel 3.1 und 3.2 präsentiert wird, stellt das eigentliche Ergebnis der Arbeit dar. In Kapitel 3.3 wird zusätzlich noch eine Evaluation erläutert, die im Rahmen dieser Bachelorarbeit durchgeführt wurde.

3.1 Word Wind Remote Control

In den ersten Schritten wurde eine einfache Version des Globus mit Standardeinstellungen initialisiert, um einen Eindruck über die vorhandenen Steuermethoden und Funktionsweisen zu erhalten.

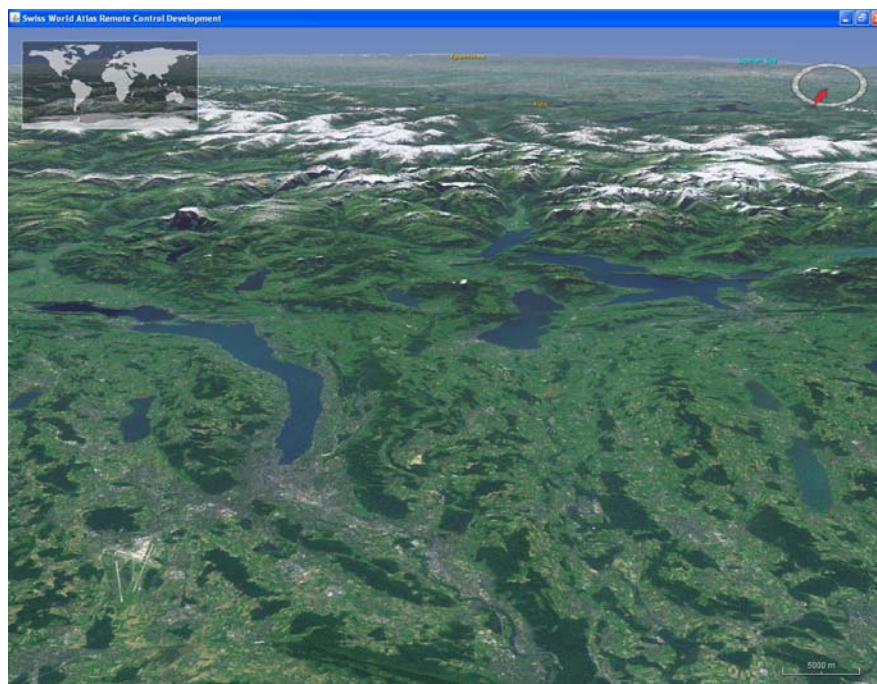


Abbildung 14 Einfache Version von NASA World Wind

3.1.1 Steuerung

Die Steuerung wird im Voraus definiert, um die Tastenbelegung zu klären. Der Einfachheit halber wurde auf den linken Stick die *Panning* Funktion gelegt, welche dem Benutzer die Bewegung mit dem linken Daumen ermöglichen soll. Die Funktion *Zoom* und *Heading* werden auf den rechten Stick gelegt. Die *Pitch* Funktion wird mit den Tasten 5 und 7 bezie-

ungsweise 6 und 8 ausgeführt. Mit dieser Tastenbelegung können alle Funktionen gleichzeitig mit zwei Händen bedient werden und es ist nicht mehr nötig, direkt vor dem Computerbildschirm zu sitzen. Ein weiterer Vorteil ist, dass alle Funktionen unabhängig voneinander bedient werden und somit mehrere Steuerungsfunktionen gleichzeitig an den Globus weitergegeben werden können. Zusätzlich sind die Buttons 1 und 2 mit unterschiedlichen *Reset* Funktionen ausgestattet, welche das Heading beziehungsweise das Heading und den Pitch zurückstellen. In der folgenden Abbildung ist der schematische Aufbau zu sehen.

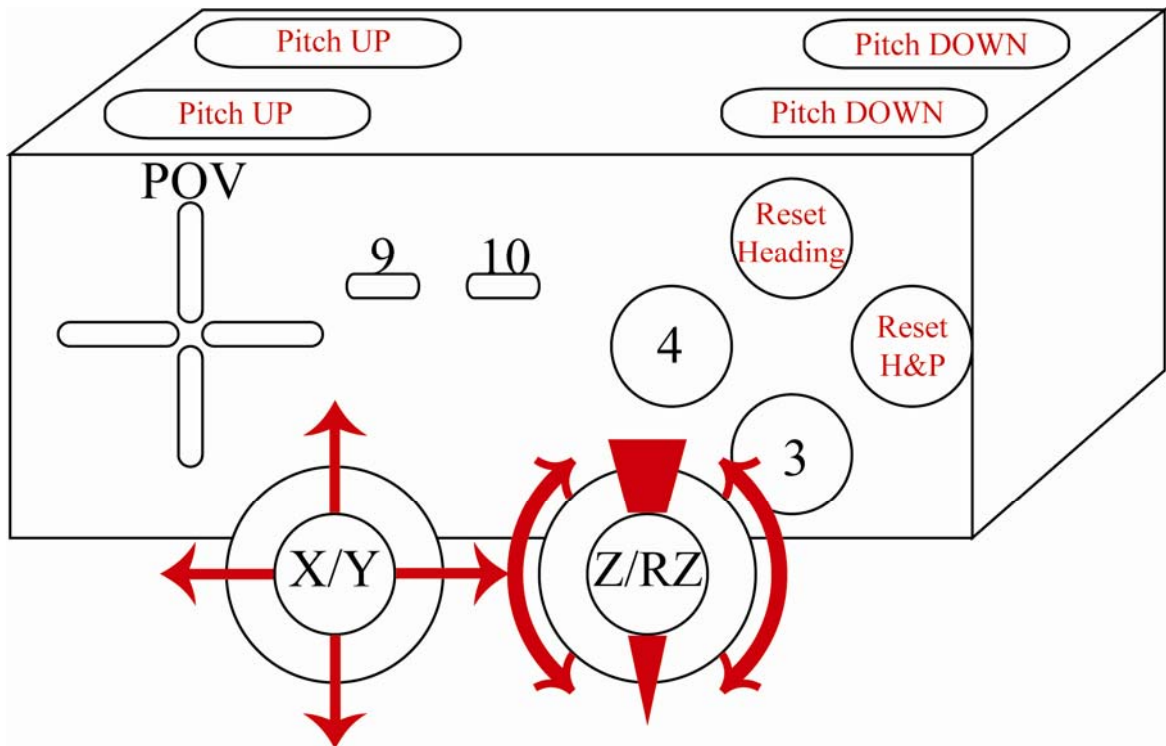


Abbildung 15 Schematische Darstellung der Steuerung des World Wind Remote Control

3.1.2 Programm

Anhand des im vorherigen Abschnitt beschriebenen Programms, wird auch die Implementierung des Remote Controllers realisiert. Dazu werden folgende Klassen erstellt:

- *ControllerFactory.java*
- *GamePadController.java*
- *NewJFrame.java*
- *PadInputHandler.java*
- *PadOrbitViewInputBroker.java*
- *RemoteController.java*
- *JoystickController.java*

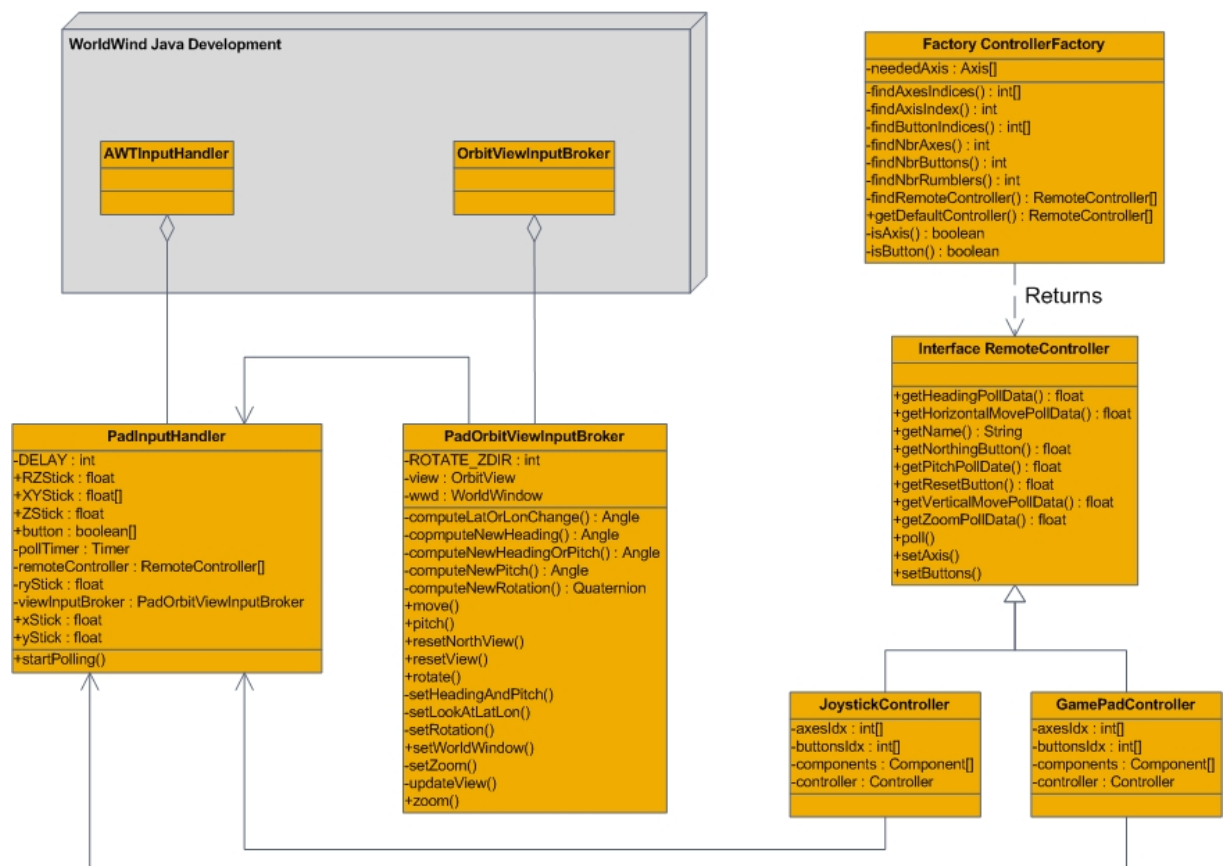


Abbildung 16 UML Diagramm des Remote Controller Programm

ControllerFactory

Die Klasse mit dem Namen *ControllerFactory.java* ist zuständig für die Erkennung des angeschlossenen Geräts und für die Auflistung der gerätespezifischen Eigenschaften. Schlussendlich wird anhand der analog verfügbaren Achsen entschieden, um was für eine Art Remote Controller es sich handelt. Danach wird ein Objekt derjenige Klasse instanziiert, welche dem Controller entspricht. Anhand dieser Factory lassen sich beliebig viele Remote Controller initialisieren, die dann unabhängig voneinander angesprochen werden können.

RemoteController

Das Interface *RemoteController* definiert die Standards und die zu implementierenden Funktionen, welche für jeden Controller definiert werden müssen. Jede Controller Klasse ist demnach eine Ableitung dieses Interfaces. Diese Art von Klasse ermöglicht beliebiges Programmieren von beliebig vielen Controllern, welche sich danach sehr einfach einbinden lassen. In dieser Arbeit sind dazu ein Gamepad und ein Joystick eingebunden.

GamePadController

Die Klasse *GamePadController* implementiert das Interface *ControllerFactory* und ist die Klasse, die die Steuerung des Globus mittels eines Gamepads erlaubt. Die Funktionen der Klasse geben dem *PadInputHandler* die gewünschten Werte weiter.

JoystickController

Diese Klasse ist ebenfalls eine Implementierung des Interface und ermöglicht somit die Programmierung der gleichen Funktionen, wie diese schon in der *GamePadController* Klasse gemacht wurden.

NewJFrame

Die *main* Methode und die Initialisierung des Frames sind der Inhalt dieser Klasse. Im weiteren wird die *PadInputHandler* Klasse dem zuvor initialisierten *Canvas* übergeben und das Model definiert.

PadInputHandler

Die *InputHandler* Klasse ist von der *AWTInputHandler* Klasse der World Wind Bibliothek abgeleitet und erweitert diese mit den Funktionen, die eine Steuerung mit einem alternativen Eingabegerät erlauben. Dabei bleiben die Grundsteuerungsmöglichkeiten mit Maus und Tastatur erhalten.

Die Klasse fragt den Controller in einem vordefinierten Intervall nach Werten ab und gibt diese an den *OrbitViewInputBroker* weiter.

PadOrbitViewInputBroker

Die Broker Klasse ist eine Ableitung der *OrbitViewInputBroker* Klasse der World Wind Bibliothek und verarbeitet die vom *InputHandler* gesendeten Werte. Mittels verschiedener Funktionen wird der Globus der Eingabe entsprechend neu gezeichnet.

3.1.3 Funktionsweise

Das Programm liest im vordefinierten Zeit-Abstand des *pollTimers*¹ die Werte des Eingabegerätes aus. Diese Werte werden ähnlich dem Programm „Beispielprogramm“ nur ausgelesen, wenn auch eine Veränderung des Ursprungszustandes des Eingabegerätes auftritt. Ist eine solche Veränderung registriert, so werden die Werte in ein Format transformiert, welches dem des NASA World Wind Globus entspricht. Sie werden jedoch nicht grafisch dargestellt, sondern direkt an den Globus weitergegeben. Danach verändert dieser seine Abbildung des Globus den Eingaben entsprechend.

3.2 SWA Anbindung

Aus dem bisherigen Programm wird ein Java Archive File (*.jar) generiert, welches danach als Bibliothek im Globus Projekt eingebunden wird. Somit sind sämtliche Funktionen auch im Globus des „Schweizer Weltatlas“ vorhanden.

¹ Vergleiche Kapitel 2.5.1; Unterkapitel Programmablauf

3.3 Evaluation

3.3.1 Beschreibung

Mit insgesamt fünf Probanden, wobei es sich um zwei Studenten und um drei Mitarbeiter des Instituts für Kartografie handelt, wurde eine kleiner Test mit Befragung zu der Handhabung des Gamepads zur Globussteuerung durchgeführt. Das Ziel dieser Befragung war abzuklären, ob diese neue Steuerungsmöglichkeit intuitiv anwendbar ist.

3.3.2 Erster Teil

Die Evaluation war in zwei Teilen aufgeteilt. Im ersten Teil setzte sich der Teilnehmer, ohne vorher eingeführt worden zu sein, an den Computer und hatte circa eine Minute Zeit sich mit der Steuerung vertraut zu machen. Dabei sollte er nach dem Test erklären, welche Funktionen auf welche Tasten gelegt wurden. Die Aufgabe wurde gut gelöst, wie in der folgenden Abbildung zu sehen ist.

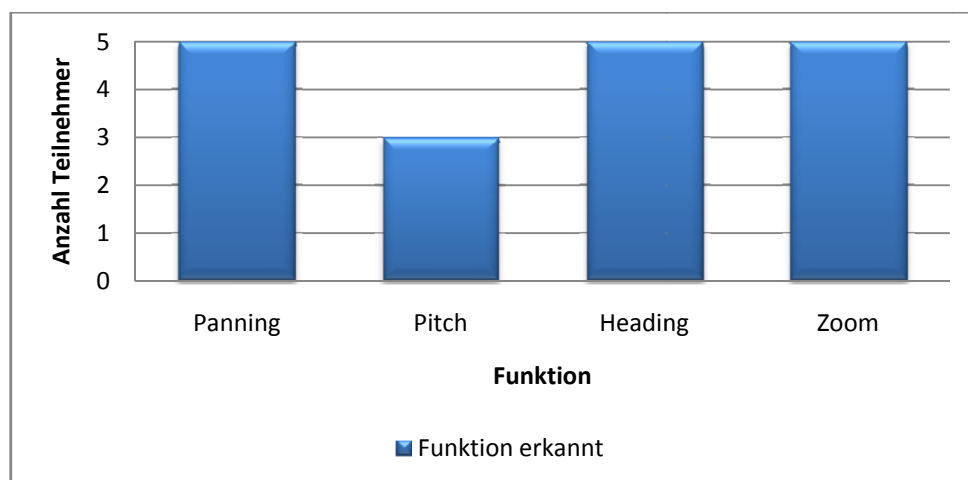


Abbildung 17 Übersicht der ersten Frage

Die drei Funktionen Panning, Zoom und Heading wurden von allen Teilnehmern erkannt und auf den richtigen Tasten eingeordnet. Einzig die Pitch Funktionen, welche bekanntlich auf den hinteren Tasten liegt, ist bei einigen Teilnehmern nicht auf Anhieb erkannt worden. Dies ist darauf zurückzuführen, dass diejenigen bisher nicht in Kontakt mit einem Gamepad gekommen sind und die Tasten gar nicht bemerkt haben.

3.3.3 Zweiter Teil

Im zweiten Teil ging es darum, von der Standard Einstellung des Globus auf die eigene Wohnung zu zoomen und dabei mit Panning die gleiche Ansicht zu erlangen, welche der Aussicht aus dem Wohnzimmer entspricht. Die Teilnehmer hatten dazu eine kurze Einführung, in der ihnen alle Funktionen und die Auswirkungen auf den Globus erklärt wurden.

Die Aufgabe wurde von allen Teilnehmern korrekt gelöst.

3.3.4 Diskussion der Befragung

Die Befragung vermittelt einerseits, dass die Steuerung intuitiv erlernbar ist, jedoch Personen, die selten oder gar nie in Kontakt mit solchen Remote Controller kommen, Schwierigkeiten bereiten kann. Der erfreuliche Schluss daraus ist aber, dass der Globus nach kurzer Eingewöhnungszeit von jedem Teilnehmer gesteuert werden konnte. Dabei war der Blick der Probanden nach gewisser Zeit nur noch auf den Bildschirm gerichtet und die Bedienung des Gamepads erfolgte nur noch mit den Händen, ohne dabei direkt Blickkontakt mit dem Eingabegerät halten zu müssen.

4 Ausblick

Um diese Arbeit in die finale Version des Globus des „Schweizer Weltatlas“ einzubringen, bedarf es noch einer grossen Testserie. Bei dieser muss zum einen die absolute Plattform und Versionenunabhängigkeit überprüft werden. Zum anderen müssen alle momentan auf dem Markt verfügbaren Joystick und Gamepads implementiert und dokumentiert werden.

Es würde sich auch eine Steuerung im Stile des *Nintendo Wii* anbieten, welche aber nur mit sehr grossem Aufwand zu realisieren wäre, da noch keine Verbindung mit einem handelsüblichen Computer möglich ist. Allerdings besteht durchaus die Möglichkeit, dass in naher Zukunft ein *Wii* ähnlicher Controller für den PC erscheint. Es wäre sehr interessant, diesen Controller in einem Programm einzubinden.

5 Literaturverzeichnis

- Davidson, A. 2007: Pro Java 6 3D Game Development,
USA: Berkley Apress, 498 S.
- Electronics57 2008: Gameport Beschreibung,
mysite.du.edu/~etuttle/electron/elect57.htm (Zugriff 22. Mai
2008)
- Google 2008: Homepage von Google Earth
earth.google.de (Zugriff 22. Mai 2008)
- IT-Wissen 2008: Online Computer Lexikon, Suchbegriffe: Universal Serial Bus,
Bluetooth, Infrarot, Tastatur, Maus, Joystick, Datenhandschuh;
www.itwissen.info (Zugriff 22. Mai 2008)
- Jinput 2008: Homepage von Jinput,
jinput.dev.java.net (Zugriff 22. Mai 2008)
- MeyersOnline 2008: Online Lexikon,
lexikon.meyers.de/meyers/Schnittstelle (Zugriff 22. Mai 2008)
- Marty, P 2007: "Swiss World Atlas – Interactive" – Analysis of user needs and
concept of a new school atlas; Co Autoren: Häberling ,C.; Hurni,
L.; Institut für Kartografie; ETH Zürich
- Netbeans 2008: Homepage der Java Entwicklungsumgebung Netbeans,
www.netbeans.org (Zugriff 22. Mai 2008)
- Reibold, H 2008: NASA World Wind Kompakt,
Deutschland: Bomots Verlag, 116 S.
- WorldWind 2008: HomePage von WorldWind,
worldwind.arc.nasa.gov (Zugriff 22. Mai 2008)
- Wi.Nintendo 2008: Homepage Nintendo Wii,
www.nintendo.de/NOE/de_DE/wii_54.html (Zugriff 22. Mai
2008)

6 Abbildungsverzeichnis

Abbildung 1 NASA World Wind .Net Windows Anwendung
Screenshot

Abbildung 2 Gameport Anschluss
Quelle: upload.wikimedia.org/wikipedia/commons/4/4d/DA-15_DSubminiatures2.svg (Zugriff 22. Mai 2008)

Abbildung 3 USB Steckerarten
Quelle: www.netzwerk-online.de/netzwerkonline/img/artikel/Image/Usbm2.jpg (Zugriff 22. Mai 2008)

Abbildung 4 Bluetooth Frequenzbereich
Quelle: www.itwissen.info/media/lex_pics/bf15g16.png (Zugriff 22. Mai 2008)

Abbildung 5 Logitech Attack3 Joystick
Quelle: www.simlog.com/images/logitech-attack-3.jpg (Zugriff 22. Mai 2008)

Abbildung 6 Nintendo Wii Remote
Quelle: www.foxnews.com/images/300224/0_61_wii_remote.jpg (Zugriff: 22. Mai 2008)

Abbildung 7 Datenhandschuh
Quelle: cgi.zdnet.de/glossar/i/small/id16f14_t.png (Zugriff 22, Mai 2008)

Abbildung 8 Steuerungsmöglichkeiten von NASA World Wind
Quelle: www.worldwindcentral.com/wiki/images/a/a4/Keychart.gif (Zugriff 22. Mai 2008)

Abbildung 9 SpeedLink GamePad
Quelle: cache2.evendi.de/pics/large/80734.jpg (Zugriff 22. Mai 2008)

Abbildung 10 Schematische Darstellung des Gamepads in einer bestimmten Situation
Screenshot

Abbildung 11 Output zu Abbildung 10
Screenshot

Abbildung 12 Eingabe und Ausgabe im Beispielprogramm
Screenshot

Abbildung 13 Screenshot	Schematischer Programmablauf
Abbildung 14 Screenshot	Einfache Version von NASA World Wind
Abbildung 15 Screenshot	Schematische Darstellung der Steuerung des World Wind Remote Control
Abbildung 16 Screenshot	UML Diagramm des Remote Controller Programm
Abbildung 17 Screenshot	Diagramm erstellt aus den Daten der Evaluation