



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# **Bachelorarbeit**

Mariusz Baldowski

Szenariobasierte Visualisierung  
von Geodaten

**Mariusz Baldowski**

**Szenariobasierte Visualisierung von Geodaten**

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Angewandte Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Thomas Thiel-Clemen  
Zweitgutachter: Prof. Dr. Stefan Sarstedt

Abgegeben am 30.08.2010

---

**Mariusz Baldowski**

**Thema der Bachelorarbeit**

Szenariobasierte Visualisierung von Geodaten

**Stichworte**

Szenario, Geodaten, Projektionen, Streaming, Nebenläufigkeit

**Kurzzusammenfassung**

Diese Arbeit beschreibt die Entwicklung eines Prototyps zur Generierung von Szenarien. Dazu werden Geodaten verwendet, um eine realitätsnahe Erde als Basis der Simulation zu erschaffen.

In dieser Arbeit geht es um die verschiedenen Geodaten, deren Formate und Projektionen, und gängige Werkzeuge zur Bearbeitung und Konvertierung. Diese Geodaten werden dann genutzt, um in einem passenden Framework einen Prototyp zu entwickeln, welcher skalier- und erweiterbar ist.

Aufgrund des hohen Speicherverbrauchs der Geodaten wird das Streamen von Daten implementiert und beschrieben. Dabei wird auf Nebenläufigkeit in einem sequentiellen Ablauf Wert gelegt.

**Mariusz Baldowski**

**Title of the paper**

Scenario-based visualization of geological data

**Keywords**

Scenario, Geological data, Projections, Streaming, Parallel processing

**Abstract**

This thesis describes the development of a prototype which can generate scenarios. Geological Data is used in order to create a close-to-reality earth as a basis of a simulation.

This thesis is about the different geological data, their formats and projections and the usual tools for the treatment and conversion. This geological data is used in order to develop a prototype in a suitable framework. The prototype will be scalable and expandable.

Due to the high memory consumption of the geological data, streaming of data will be implemented and described. Great emphasis is placed on parallel processing in a sequential process.

---

# Inhaltsverzeichnis

	Inhaltsverzeichnis .....	4
1	Einführung .....	7
1.1	Motivation .....	7
1.2	Zielsetzung .....	8
1.3	Abgrenzung .....	8
1.4	Inhaltlicher Aufbau .....	8
2	Grundlagen .....	10
2.1	Geoinformationssystem .....	10
2.1.1	Geoportale .....	11
2.2	Digitale Geodaten .....	11
2.2.1	Datenerfassung und Formate .....	12
2.2.1.1	Primärdaten von Geoinformationen .....	12
2.2.1.2	Sekundärdaten von Geoinformationen .....	13
2.2.2	Projektionen .....	13
2.2.2.1	Quadratische Plattkarte .....	14
2.2.2.2	Mercator-Projektion .....	14
2.2.3	Geodatenbanken und Server .....	15
2.2.4	Kachelserver .....	15
2.3	Szenarien .....	16
2.4	Zusammenfassung .....	17
3	Analyse .....	18
3.1	Ist-Analyse .....	18
3.1.1	Vorhandene Geoinformationssysteme .....	18

# INHALTSVERZEICHNIS

---

3.1.2	Geodatenquellen .....	19
3.1.3	Szenario .....	20
3.2	Anforderungsanalyse .....	20
3.3	Lastenheft .....	21
4	Design.....	22
4.1	Schichtenarchitektur.....	22
4.1.1	Informationsschicht.....	23
4.1.2	Zusätzliche Texturen .....	23
4.1.3	Topographie .....	23
4.1.4	Höheninformation .....	24
4.2	Quadtree als Datenstruktur .....	25
4.3	Zusammenfassung.....	25
5	Werkzeuge und Geodatenquellen .....	26
5.1	Geodatenquellen .....	26
5.1.1	SRTM Höheninformationen .....	26
5.1.2	NASA Blue Marble - Next Generation .....	27
5.1.3	NASA Earth Observations .....	29
5.1.4	Google Static Maps .....	30
5.1.5	Weitere Texturen .....	32
5.2	Verwendete Werkzeuge .....	32
5.2.1	Earth Sculptor.....	32
5.2.2	Global Mapper .....	33
5.3	Zusammenfassung.....	34
6	Implementation .....	36
6.1	Einarbeitung .....	36
6.1.1	Geländeerstellung .....	36
6.1.2	Framework.....	38
6.1.3	Probleme .....	39
6.2	Realisierung .....	40
6.2.1	Stadium 1 .....	40

## INHALTSVERZEICHNIS

---

6.2.2	Stadium 2 .....	42
6.2.3	Stadium 3 .....	47
6.2.4	Stadium 4 .....	48
7	Auswertung.....	49
8	Diskussion und Ausblick .....	51
	Literaturverzeichnis.....	52
	Abbildungsverzeichnis .....	56
	Anhang .....	58

# Kapitel 1

## Einführung

Diese Arbeit befasst sich mit der grafischen Visualisierung von Geodaten. Ziel dieser Visualisierung ist es, bestimmte Szenarien innerhalb dieses Systems zu simulieren, welche für das reale System zu komplex und dynamisch sind. Durch die Analyse der Simulation soll das Problem besser verstanden und anschaulich gemacht werden.

Problematisch bei der Entwicklung und der Darstellung im Zielsystem sind die unterschiedlichen Formate und Projektionen der Geodaten. Diese müssen für das Zielsystem erst interpretiert, ausgewertet und portiert werden. Dabei muss eine einfache einheitliche Schnittstelle zwischen unterschiedlichen Geodaten gefunden werden.

### 1.1 Motivation

Den Rahmen für diese Arbeit liefert ein Projektmodul an der Hochschule für Angewandte Wissenschaften Hamburg. Parallel dazu bot Prof. Dr. Stefan Sarstedt ein weiteres Projektmodul im gleichen Semester an der HAW Hamburg an, welches sich mit der Generierung eines speziellen Szenarios beschäftigte. Dabei wurde ein Logistiksystem entwickelt (HAW Logistics System, kurz HLS), welches die Grundlage für das in dieser Arbeit entwickelte Geoinformationssystem bot.

### 1.2 Zielsetzung

In dieser Arbeit werden Geodaten, welche in verschiedenen Formaten und Projektionen vorliegen, vorgestellt. Dabei werden passende Werkzeuge und Methoden verwendet, um die Daten aufzubereiten und für das Szenario basierte Modell zu portieren.

Es wird ein Geoinformationssystem entwickelt, welches unterschiedliche aufbereitete Geodaten visualisiert und Schnittstellen für die Szenario Generierung innerhalb dieses Modells bereitstellt. Ein Prototyp des Geoinformationssystems ist das Ergebnis dieser Arbeit und dient als Grundlage für die weitere Auseinandersetzung dieser Thematik.

### 1.3 Abgrenzung

Die Entwicklung und Integrierung eines komplexen und realitätsnahen Szenarios ist eine Aufgabe, die aufgrund der Recherche und Aufbereitung der nötigen Daten viele Jahre in Anspruch nehmen kann. Das Szenario muss sorgfältig geprüft, mit Experten beratschlagt und für das Zielsystem aufbereitet werden. Im Anschluss müssen die ausgewerteten Daten auf Plausibilität getestet werden, um die nötigen Rückschlüsse aus dem Szenario zu ziehen.

Aufgrund dieser Tatsache und der begrenzten Zeit, die für die Erstellung dieser Arbeit zur Verfügung steht, wird nur ein Prototyp entwickelt. Das entwickelte Design wird der Grundbaustein des Prototyps werden, welcher für die zukünftige Szenario Generierung bereit stehen wird. Die Entwicklung des Prototyps wird sich jedoch an das entwickelte Szenario des HLS orientieren.

### 1.4 Inhaltlicher Aufbau

Der Aufbau dieser Arbeit wurde so gewählt, dass die Entwicklung des Prototyps nachvollzogen werden kann. Im zweiten Kapitel werden die Grundlagen angesprochen, die für diese Arbeit und für den Prototyp die Basis bieten. Dabei werden Szenarien, verschiedene Geodaten und deren Projektionen näher beschrieben.

Im dritten Kapitel wird das zu entwickelte System analysiert. Eine Analyse

von vorhandenen Systemen und Geodatenquellen hilft der Eingrenzung der Anforderungen. Im vierten Kapitel wird das zu verwendete Design näher beschrieben. In Anbetracht der vorangegangenen Kapitel wird hier eine geeignete Architektur gewählt. Im fünften Kapitel werden die für den Prototyp benutzten Werkzeuge und Geodatenquellen vorgestellt und näher beschrieben.

Das sechste Kapitel beschreibt die Implementierung des Prototyps. Dabei werden die Einarbeitung und die verschiedenen Stadien zusammengefasst. Eine Auswertung findet in Kapitel 7 statt. Die Diskussion und ein Ausblick für potentielle zukünftige Aufgabenbereiche des Prototyps sind in Kapitel 8 zu finden.

## Kapitel 2

### Grundlagen

In diesem Kapitel werden die grundlegenden Elemente näher beschrieben. Dabei geht es um das Verständnis von Geoinformationssystemen und Geodaten, die vorliegenden Formate und die gängigsten Projektionen. Des Weiteren wird das Problem der Portierung von Rohdaten erläutert und der Begriff der aufbereiteten Geodaten definiert.

#### 2.1 Geoinformationssystem

Geoinformationssysteme sind Informationssysteme zur Erfassung, Bearbeitung, Analyse und Präsentation geografischer Daten. Geoinformationssysteme schließen die benötigte Hard- und Software, die Daten und Anwendungen mit ein. Geoinformationssysteme werden unter anderem in der Umweltforschung, Archäologie, Marketing, Kartographie, Städteplanung, und in der Logistik verwendet.

Geoinformationssysteme erweitern die Nutzungsmöglichkeiten der klassischen Landkarte. Die primäre Funktion ist die Visualisierung der Daten, jedoch unterstützen moderne Informationssysteme weitere Funktionen zur Analyse der Geodaten. Dazu gehören die Datenbearbeitung und Konvertierung, die Koordinatentransformation und die generelle Datenverwaltung. In den folgenden Abschnitten werden einige dieser Funktionen näher beschrieben.

### 2.1.1 Geoportale

Mithilfe von verteilten, dienstbasierten Architekturen ist eine einfache und kostengünstige Geodatendistribution möglich. Die meisten Geoinformationssysteme unterstützen den Zugriff auf webbasierte standardisierte Karten- und Geodatendienste. Entwicklungen in diesem Bereich zeigen eine verstärkte Bedeutung von Geoinformationssystemen im Internet auf. Geoportale sind eine spezifische Ausprägung von Geoinformationssystemen, die Webportale für die Suche nach und den Zugriff auf geographische Informationen und Dienste mithilfe eines Webbrowsers ermöglichen. [BP97]

## 2.2 Digitale Geodaten

Eine Information über ein geographisches Phänomen, welche direkt oder indirekt mit einer bestimmten Position auf der Erde verbunden ist, nennt man Geoinformation. Werden diese durch Zeichen codiert und für den Computer lesbar gemacht, nennt man diese Informationen digitale Geodaten. Es können unterschiedliche Inhalte in Geoinformationen gespeichert werden. Geodaten können unmittelbar gewonnene Geoinformationen (Primärdaten) oder weiter bearbeitete Sekundärdaten sein.

Unter Primärdaten versteht man die Informationen, die unmittelbar gewonnen werden. Aus diesen Primärdaten, besser bekannt als Rohdaten, können Daten abgeleitet und aufbereitet werden, die so genannten Sekundärdaten oder aufbereitete Daten.

Unter Georeferenzierung, Geokodierung oder Verortung versteht man die Zuweisung raumbezogener Referenzinformationen zu einem Datensatz. Der Raumbezug muss erst durch Transformationen und Konvertierungen hergestellt werden. Dazu gehören die Eliminierung geometrischer Verzerrungen oder die Einpassung der Daten in ein gewähltes Koordinatensystem. Rohdaten sind im Gegensatz zu den aufbereiteten Sekundärdaten nicht georeferenziert.

Daten, die durch Modellier- oder Verarbeitungsschritte erstellt werden, nennt man Sekundärdaten (oder aufbereitete Daten). Man benutzt die gesammelten Primärdaten, verarbeitet und interpretiert diese, um Sekundärdaten zu erhalten, welche für die weiteren Aufgaben verwendet werden können.

Primärdaten können dementsprechend nicht verwendet werden, da sie abhängig von der Interpretation der Daten verständlich gemacht werden müs-

sen. Jedoch können diese Daten nicht korrekt interpretiert werden, da sie nicht georeferenziert sind. Um nicht von dem Einsatzgebiet abhängig zu sein, versucht man, Primärdaten in ein Format zu konvertieren, welches in vielen Zielsystemen verwendet werden kann.

### 2.2.1 Datenerfassung und Formate

Da Primärdaten nicht direkt verwendet werden können, müssen diese erst in Sekundärdaten konvertiert werden. Im folgenden Abschnitt werden standardisierte Formate vorgestellt, die für die Entwicklung des Geoinformationssystem von Bedeutung waren. Um die Primärdaten zu konvertieren, werden passende Werkzeuge benötigt, welche die Rohdaten in das gewünschte standardisierte Format konvertieren. Werkzeuge, die für diese Arbeit ausgewählt und benutzt wurden, werden in Kapitel [5.2](#) näher erläutert.

#### 2.2.1.1 Primärdaten von Geoinformationen

Während der Shuttle Radar Topography Mission (SRTM) im Februar 2000 wurden Fernerkundungsdaten aufgezeichnet. Mit diesen Daten ist es möglich, ein hochauflösendes digitales Geländemodell der Erdoberfläche zu erstellen. Die SRTM Daten liegen im HGT Format vor. Das HGT Format speichert die Rohdaten der SRTM Daten. [\[RBS00\]](#) Die aufbereiteten SRTM Daten werden den grundlegenden Baustein des Prototypen ausmachen und in Kapitel [5.1.1](#) weiter erläutert.

Eine andere Möglichkeit, Rohdaten zu sammeln, ist mithilfe eines GPS-Empfängers. Dadurch können gelaufene oder gefahrene Strecken aufgezeichnet werden. Sogenannte GPS-Tracks dienen als Rohdaten für die Erstellung von Straßennetzen. Der GPS-Empfänger zeichnet nur die Wege auf, ohne genau zu wissen, wie die Straßen und Wege heißen, weshalb diese Rohdaten noch interpretiert und weiter bearbeitet werden müssen. Sinnvoll ist, die Sekundärdaten in einem XML-Format zu speichern, da die GPS-Tracks überwiegend durch Attribute beschrieben werden können.

### 2.2.1.2 Sekundärdaten von Geoinformationen

Sekundärdaten sind georeferenziert, was heißt, dass man zu einer gewählten Position die nötigen Informationen aus den Sekundärdaten auslesen kann. Da das XML Format standardisiert ist und die Schnittstellen öffentlich einsehbar und verständlich sind, eignet es sich sehr gut für Sekundärdaten. Mithilfe von Attributen wird im XML-Format gespeichert, an welcher Position sich welche Information befindet.

Um Rohdaten zu visualisieren, zeichnet man zu den zugehörigen Positionen Bildpunkte. Wenn die Daten in eine Bilddatei visualisiert werden, sind sie einfach für mehrere Zielsysteme zu interpretieren. Bilddateien werden von den meisten Systemen unterstützt. Die Formatierung in eine Bilddatei erschafft den Vorteil, auch andere Formate mit den visuellen Repräsentationen zu verknüpfen.

### 2.2.2 Projektionen

Für diese Arbeit werden die Geodaten der Erde verwendet. Problematisch bei der Verwendung von Geodaten der Erde sind die unterschiedlichen Kartenprojektionen (auch Kartenabbildung genannt), die je nach Anwendung vorteilhafter sind.

Die Erde ist eine dreidimensionale Kugel. Um sie nun auf eine zweidimensionale Karte zu projizieren, müssen unterschiedliche Abbildungsvorschriften benutzt werden, welche man mathematisch ausdrücken kann. Um die Erde in eine geeignete Kartenabbildung abzuleiten, sind drei Schritte notwendig:

- Modellauswahl für die Form der Erde (Kugel oder Ellipsoid)
- Geographische Koordinaten (Länge und Breite) werden in ein kartesisches Koordinatensystem umgewandelt (Richtungsachsen stehen orthogonal aufeinander, Koordinatenlinien sind Geraden in konstantem Abstand voneinander)
- Skalierung der Karte

Diese Koordinatentransformation ist eine zentrale Funktion von Geoinformationssystemen. Die Koordinatentransformation kann im laufenden Betrieb, oder in einem eigenen Arbeitsschritt erfolgen. Es gibt unzählige Projektionen, die Vor- und Nachteile besitzen. Die zwei Projektionen, die für diese Arbeit verwendet wurden, werden im Folgenden näher vorgestellt.

### 2.2.2.1 Quadratische Plattkarte

Die quadratische Plattkarte (auch Equirectangular Projektion genannt) ist eine abstandstreue Zylinderkarte, die einen Berührungskreis am Äquator besitzt. Bei der quadratischen Plattkarte ist der Abstand von zwei benachbarten Längen- und Breitengraden identisch. Dies führt zu einer Verzerrung der Breiten.

Aufgrund der Verzerrung wird diese Projektion selten verwendet. Sie ist nicht zur Navigation zu gebrauchen. In modernen Geoinformationssystemen wird diese Projektion verwendet, weil die geographischen Koordinaten direkt in die Karte eingetragen und ausgelesen werden können.

Plattkarten sind Ergebnisse von Satelliten, die nur um den Äquator kreisen.

### 2.2.2.2 Mercator-Projektion

Die Mercatorprojektion ist eine Zylinderprojektion, die in Nord-Süd-Richtung geeignet verzerrt ist, um eine winkeltreue Abbildung zu erreichen. Eine winkeltreue Abbildung beschreibt eine Abbildung, die keine Winkelverzerrung besitzt. Alle Winkel entsprechen dem des Urbildes.

Bei einer winkeltreuen Abbildung können Richtungen und Distanzen korrekt eingetragen werden, ohne die Abbildung zu ändern. Aufgrund dieser Eigenschaft eignet sich die Mercator-Projektion hervorragend für die Navigation.

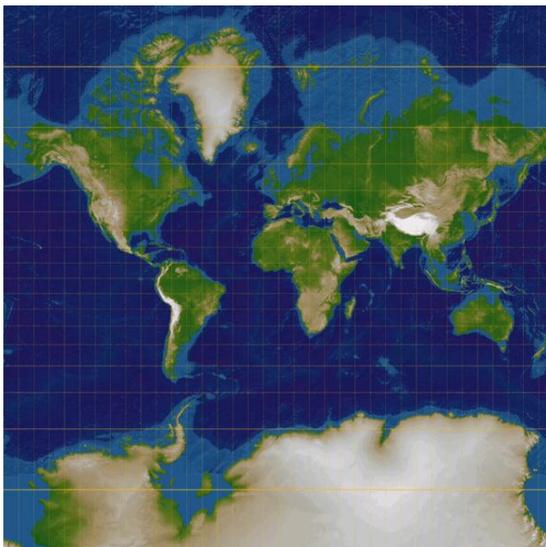


Abbildung 2.2.1:  
Mercator-Projektion [Oea03]

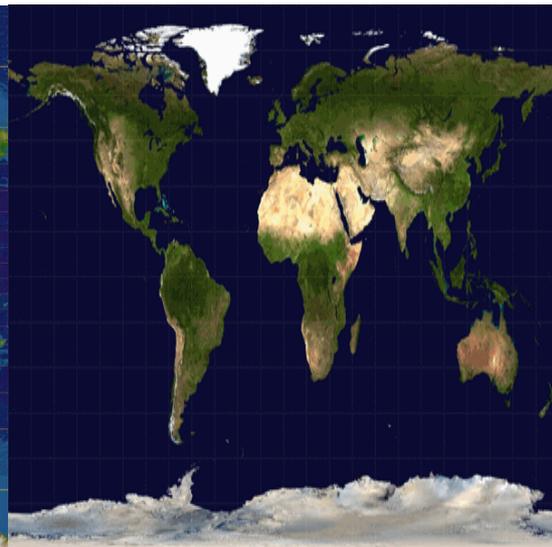


Abbildung 2.2.2:  
Quadratische Plattkarte [Oea03]

### 2.2.3 Geodatenbanken und Server

Für die Speicherung der Sach- und Geometriedaten hat sich die Nutzung von marktgängigen relationalen beziehungsweise objektrelationalen Datenbanksystemen für die Geodatenverwaltung durchgesetzt. Geodaten können nicht durch konventionelle Datenbanken effizient verwaltet werden. Es gibt für viele Datenbanken Erweiterungen für die Verwaltung von Geodaten, wie zum Beispiel PostGIS. [PGI00]

Die Geodaten in der Datenbank können in unterschiedlichen Formaten und Projektionen vorliegen. Das System, welches Geodaten anfragen möchte, muss diese Anfragen an die Server schicken, welche die Datenbanken verwalten. Es gibt Server, die die Konvertierungen der Formate und Portierungen der Projektionen anbieten. Diese Server sind aber an bereits vorhandene Geoinformationssysteme gekoppelt, weshalb nicht auf diese Funktionalität zurückgegriffen werden kann.

### 2.2.4 Kachelserver

Ein Kachelserver (Tile-Server) ist ein Geodatenserver, welcher ein Verfahren einsetzt, um Karten durch Zerschneiden in kleinere Kacheln zu unterteilen. Dazu wird die Karte in ein gewähltes Gitterkoordinatensystem unterteilt. Die geringste Stufe zeigt die gesamte Weltkarte. Mit jeder Erhöhung der Zoomstufe verdoppelt sich die Anzahl der Tiles, über die sich die Karte erstreckt. Somit kann die Auflösung der Karte gesteigert werden. Da die Unterteilung bekannt ist, kann zu einer gegebenen Position und Zoomstufe nur die benötigte Kachel der Karte zu laden. Der Service von [Google Static Maps](#), welcher in dieser Arbeit verwendet wurde, nutzt dieses Verfahren. Die Kacheln werden dann über http Anfragen geladen.

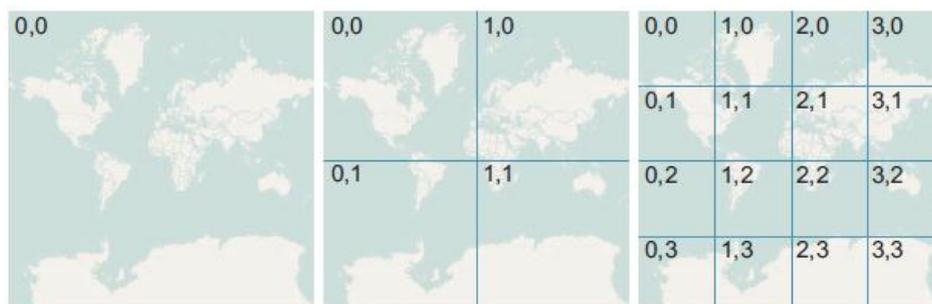


Abbildung 2.2.3: Eine Karte bei verschiedenen Zoomstufen [Kai08]

## 2.3 Szenarien

Ein Szenario ist die Beschreibung einer möglichen zukünftigen Situation. Jedes Szenario lässt andere Schlussfolgerungen und Ergebnisse zu. Dabei geht es in dieser Arbeit um Szenarien, die auf die Erde angewandt werden können. Die Szenarien können unterschiedliche Thematiken simulieren, wie zum Beispiel bestimmte Umweltkatastrophen oder Evakuierungsmissionen in Gebäuden.

Das Ziel dieser Arbeit ist, eine Grundlage für Simulationen zu erschaffen, um solche realitätsnahen Szenarien zu testen und auszuwerten. Durch die Entwicklung eines Geoinformationssystems, welches zur Modellierung die Geodaten der Erde verwendet, wird eine passende Komponente zur Visualisierung und Simulation des Szenarios bereitgestellt. Gängige Geoinformationssysteme sind zwar in der Lage, die georelevanten Daten zu visualisieren, jedoch fehlt in diesen Systemen eine Integrationsmöglichkeit für das Szenario, da das Szenario stark abhängig von der Interpretation und Implementierung ist. In den meisten Fällen wird das Grundgerüst des Szenarios unabhängig von dem Geoinformationssystem erstellt, weshalb eine Anbindung zu gängigen Geoinformationssystemen schwer bis unmöglich ist. Diese Arbeit befasst sich mit der Entwicklung eines Geoinformationssystems, welches dafür ausgelegt ist, ausgekoppelte Szenarien in der modellierten Welt zu simulieren.

## 2.4 Zusammenfassung

Geoinformationssysteme visualisieren und interpretieren Geodaten. Sie besitzen einige Funktionen, die die Nutzungsmöglichkeiten der klassischen Karte erweitern. Dabei muss darauf geachtet werden, dass die Geodaten in korrekten Projektionen liegen, um eine einheitliche Georeferenzierung zu ermöglichen. Geoinformationssysteme können auf Geoportale und Server mit Geodatenbanken zugreifen, um die nötigen Daten zur Visualisierung zu erhalten. Dabei ersetzen die meisten Geoinformationssysteme die Funktionalität der Server, und benötigen nur eine direkte Anbindung zu den Geodatenbanken. Die Geoinformationssysteme werden unabhängig von bestimmten Szenarien entwickelt, weshalb eine Anbindung und Simulation erschwert wird. Ein Geoinformationssystem mit einer Schnittstelle zu dem Szenario System kann durch nachfolgende fachliche Architektur beschrieben werden.

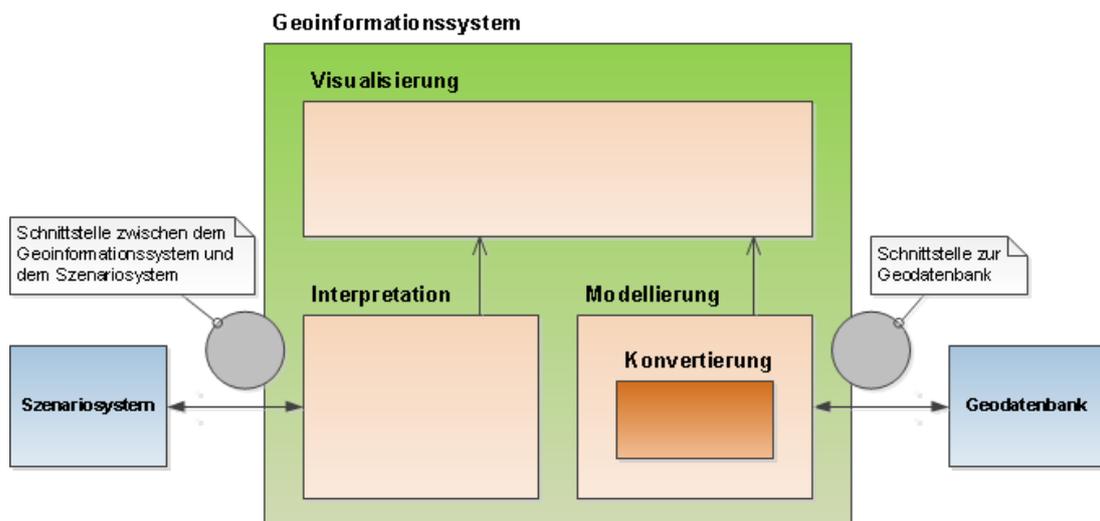


Abbildung 2.4.1: Fachliche Architektur eines Geoinformationssystems

## Kapitel 3

### Analyse

In diesem Kapitel wird erläutert und zusammengetragen, welche Anforderungen das Geoinformationssystem aufweisen muss, um eine Ausgangsversion zur Erstellung von Szenarien zu erschaffen. Dabei sollte die Skalierbarkeit des Systems im Vordergrund stehen. Eine Ist-Analyse fasst zusammen, welche Komponenten und Daten bereits verfü- und benutzbar sind, und bietet die Grundlage für die Anforderungsanalyse.

#### 3.1 Ist-Analyse

In diesem Abschnitt werden vorhandene Geoinformationssysteme untersucht und ausgewertet. Des Weiteren wird nach vorhandenen Geoinformationen gesucht, die verwendet werden können. Auch das verwendete Szenario von HLS wird vorgestellt.

##### 3.1.1 Vorhandene Geoinformationssysteme

Es gibt zwei größere Geoinformationssysteme, die Geodaten visualisieren. Mit NASA World Wind [\[NAS08\]](#) bietet die NASA ein Programm an, welches als Basis die von der NASA selbst erstellten Geodaten nimmt und die Welt modelliert. Die einzelnen Datensätze können als Schichten ausgetauscht und angezeigt werden. Die Quellen sind öffentlich, Java wird als Programmiersprache verwendet. Leider fehlt eine Dokumentation über dieses Programm und des Designs. Erweiterungen sind schwer zu implementieren, da dieses Projekt eine unüberschaubare Größe angenommen hat. Dadurch können

keine speziellen Szenarien eingebunden werden, es fehlt, wie in [Abbildung 2.4.1](#) zu sehen, die Schnittstelle für das Szenario System.

Jedoch eignet sich dieses Programm, um spezielle Problemlösungen zu finden, wie zum Beispiel die Realisierung der einzelnen Projektionen der Welt. Die benutzten Lösungsstrukturen werden im Kapitel [6.2.2](#) näher erläutert.

Google bietet mit Google Earth ein ähnliches System wie das der NASA an, mit dem Unterschied, dass Google Earth verschiedene Geodatenquellen verwendet. Google bedient sich der Quellen der NASA und anderen größeren Satellitenbildern für die niedrigauflösende Darstellung der Welt. Dazu reichert Google hochauflösende Straßen- und Gebietsnetze mit ihrem eignen Dienst, Google Maps, an. Außerdem können in Google Earth Schichten hinzugefügt werden, wenn sie in dem für Google Earth konzipierten KML Format vorliegen.

Auch Google Earth bietet keine Schnittstellen nach außen an. Es ist ein geschlossenes System ohne eine Szenario Generierung.

### 3.1.2 Geodatenquellen

Die Geodatenquellen werden hier aufgrund der Ist-Analyse genannt und in Kapitel [5](#) näher erläutert. Zur Ist-Analyse ist vorerst nur wichtig, dass diese Datensätze verfügbar und für den Prototyp nutzbar sind.

Auf der Suche nach Geodaten zur Modellierung der Welt stößt man auf die von der NASA angebotenen Daten. [\[Ret05\]](#) Durch Satelliten hat NASA die komplette Erde als Bilddatei gespeichert. Die NASA Datensätze sind qualitativ hochwertig, da sie eine sehr feine Auflösung haben, sehr aktuell und wolkenlos sind. Des Weiteren sind sie allgemein zugängliche Quellen, das heißt, sie sind technisch dafür geeignet und auch bestimmt für die Allgemeinheit zugänglich zu sein.

Mithilfe der Geodaten der NASA ist eine Modellierung der Welt möglich. Es fehlen aber noch detailliertere Karten und Karten von Straßennetzen.

Google hat in den letzten Jahren Aufsehen mit dem Service Google Maps erregt. Die hochauflösenden Bilder der Straßen und Gebiete eignen sich hervorragend als Informationsquelle neben den Geodaten der NASA. Mit der [Google Static Maps API](#) [\[Goo10a\]](#) wird ein Service angeboten, der Bilddateien anhand einer URL mit Parametern zurückschickt. Die Datensätze von

Google bringen die nötige Granularität, um realitätsnahe Simulationen zu entwickeln.

### 3.1.3 Szenario

Das HAW Logistics Systems ist ein Logistiksystem, welches im Rahmen eines Projektmoduls an der HAW Hamburg unter der Leitung von Prof. Dr. Stefan Sarstedt entwickelt wurde. In diesem Szenario wird eine fiktive Firma, HAW Logistics, als neuer Spediteur auf dem Markt gegründet. Dabei erhält HAW Logistics Kundenaufträge für Transportprozesse und übernimmt die komplette Transportplanung bis hin zur Abrechnung. HAW Logistics agiert auf verschiedenen Verkehrszweigen, daher kommen sowohl Land, als auch Luft und See für Transporte in Frage.

Dieses Szenario eignet sich als Ausgangslage für die Schnittstelle zwischen dem Geoinformationssystem und dem Szenario System, da die interne Logik innerhalb des Szenario Systems stattfindet, während das Geoinformationssystem das Szenario visualisiert. Das HLS benötigt Transportwege, daher visualisierte Straßen- und Verkehrskarten, und Transporteinheiten wie Autos, Flugzeuge und Schiffe, um das Szenario zu simulieren. Das entwickelte Lastenheft kann unter [\[HLS10\]](#) nachgelesen werden.

## 3.2 Anforderungsanalyse

Anhand der Ist-Analyse kann man feststellen, dass genug Geoinformationen zur Verfügung stehen, um den Prototyp damit anzureichern. Die explizite Sammlung von neuen Geodaten ist nicht notwendig. Die Kombination aus den NASA Datensätzen zur Modellierung der Welt und dem Service von Google, der das Streamen von Daten erlaubt, eignet sich zur Erstellung des Prototypen für die realitätsnahe Szenario Generierung.

Es gibt viele verschiedene Geoinformationen in unterschiedlichen Formaten. Das Geoinformationssystem sollte die gängigsten Sekundärdaten interpretieren und visualisieren können, um nicht abhängig von einem Format zu sein. Dadurch erreicht man die notwendige Skalierbarkeit.

In den Geodatensätzen der Welt gibt es keine Geoinformation, die alle Bereiche zur Modellierung der Erde zeitgleich in sich trägt. Die einzelnen Bereiche, wie das Kartennetz oder das dreidimensionale Modell der Welt, sind separat gespeichert. Dadurch muss das Geoinformationssystem unterschiedliche Geodaten zeitgleich verwenden können, um eine realitätsnahe Welt zu mo-

dellieren.

Da die Geodatenquellen der NASA und von Google verwendet werden können, wird das Geoinformationssystem primär für die Interpretation von Bilddateien entwickelt, da die Geodaten in Bildformat vorliegen.

Das Geoinformationssystem sollte ohne größere Veränderungen verschiedene Geodaten anzeigen können. Da es unterschiedliche Projektionen gibt, muss das Geoinformationssystem feststellen können, in welcher Projektion die Geoinformation vorliegt und diese entsprechend umwandeln, um die Datensätze visualisieren zu können.

Das Geoinformationssystem soll eine realitätsnahe Visualisierung besitzen. Im Zusammenhang mit einem Planeten, in diesem Falle unserer Erde, sollte der Prototyp die Erde als dreidimensionale Kugel visualisieren können, ohne Informationsverlust. Dies heißt, dass eine Abbildungsvorschrift gefunden werden muss, wo Positionen von zweidimensionalen Karten auf einer dreidimensionalen Projektion äquivalent sind.

Die gefundenen Anforderungen bieten die Grundlage für das Design und die Implementierung des Prototyps.

### 3.3 Lastenheft

Aus den gesammelten Anforderungen in der Anforderungsanalyse kann ein Lastenheft erstellt werden. Dies fasst die Anforderungen kurz zusammen und dient als Nachschlagewerk.

#### Anforderungen

- A1** Das System kann Bilddateien zur Modellierung der Welt interpretieren.
- A2** Das System kann Geodaten aus unterschiedlichen Geodatenquellen zeitgleich anzeigen.
- A3** Das System kann die Projektion der Geodaten speichern.
- A4** Das System kann die Projektion von Geodaten zwecks korrekter Interpretierung intern verändern.
- A5** Das System kann Positionen im dreidimensionalen Raum korrekt referenzieren (In Bezug auf die zweidimensionale Karte).

#### Ausgrenzungen

- G1** Es werden keine neuen Geodaten explizit gesammelt.
- G2** Es wird kein neues Szenario entwickelt.

# Kapitel 4

## Design

In diesem Kapitel wird das Design erläutert, welches der entwickelte Prototyp haben wird. Die gefundenen Anforderungen in Kapitel 3.2 dienen als Grundlage für das Design.

### 4.1 Schichtenarchitektur

Aus Anforderung A2 geht hervor, dass der Prototyp unterschiedliche Geodatenquellen benutzen wird. Dies ist notwendig, da nicht alle relevanten Informationen in einer einzelnen Geoinformation zu finden sind.

Eine typische Modellierung der Welt unterteilt sich hauptsächlich in 4 Schichten, die zum Teil aus mehreren Ebenen bestehen. Diese Beobachtung wurde während der Einarbeitungsphase bestätigt, da Programme, die Gelände entwerfen konnten, die Informationen über das Gelände in verschiedenen Schichten unterteilt haben. Näheres zu dieser Beobachtung ist in Kapitel 6 in dieser Arbeit zu finden.

Auf der untersten Schicht sind die Informationen über das Relief der Welt (Höheninformation) gespeichert, da diese Ebene den Grundbaustein der dreidimensionalen Modellierung bildet.

Zu jedem Relief gehören die Informationen über Topographie, welche in Kapitel 4.1.3 näher beschrieben werden. Da es unterschiedliche Datenquellen gibt, ist es sinnvoll, hier mehrere auswechselbare Ebenen zu implementieren.

Die Modellierung der Welt ist somit abgeschlossen. Zusätzliche Texturen dienen zur realitätsnahen Visualisierung und sind für die Szenario Generierung nicht essentiell.

Die Informationsschicht dient zur Verwaltung der Daten, die für das Szenario relevant sind. Hier könnten als Beispiel Informationen über Gebäude gespeichert sein, oder klimatische Beobachtungen. In Bezug auf das ausgewählte Szenario des HLS werden hier die Transportinformationen gespeichert.

Diese Schichtenarchitektur bietet den Vorteil, Positionsrelevante Informationen über mehrere Geodatenquellen zusammenzufassen, solange die Projektionen abgestimmt sind. Dies ist für die Skalierbarkeit des Geoinformationssystems wichtig.

### **4.1.1 Informationsschicht**

In der Informationsschicht sind die für das Szenario relevanten Geodaten gespeichert, welche nicht bereits in den unteren Schichten abgedeckt werden. Höheninformation oder Topographie sind allgemein zugängliche Geodaten, die die Grundlage für jedes Szenario bilden.

Diese Schicht kann von der Implementierung her recht offen gestaltet werden, da die Informationen unterschiedlich gespeichert werden. In dieser Schicht könnte ein XML-Parser eingesetzt werden, um Informationen aus XML-Dokumenten auszulesen. Relevant ist, dass die Schicht über eine Schnittstelle verfügt, über welche man die nötigen Informationen zu einem bestimmten Punkt erhalten kann.

### **4.1.2 Zusätzliche Texturen**

Zusätzliche Texturen besitzen keine nennenswerten Informationen und dienen lediglich der Visualisierung von zusätzlichen Ebenen. Beispiele hierfür sind Schatteneffekte oder ein transparenter Wolkenhimmel. Die zusätzlichen Texturen können, müssen aber nicht über eine Schnittstelle bezüglich der korrekten Georeferenzierung besitzen.

### **4.1.3 Topographie**

Die Topographie beschreibt die Erdoberfläche, ohne Informationen von den darauf befindlichen Objekten. Hier wird das Gelände grafisch beschrieben, auch Straßenkarten sind möglich. Diese Schicht ist die „Haut“ über den Hö-

heninformationen.

Anzumerken ist, dass aufgrund von Anforderung [A2](#) der Prototyp abhängig von den Einstellungen verschiedene Topographien aus unterschiedlichen Geodatenquellen visualisieren muss. Dies ist dann nötig, wenn von einer Geodatenquelle nicht alle Informationen erhalten werden können, oder eine andere Geodatenquelle ab einer bestimmten Konfiguration bessere Resultate liefert.

### 4.1.4 Höheninformation

Die Grundlage einer dreidimensionalen Welt sind die Höheninformationen zu gegebenen Punkten. Da dies die Basis für die auf dieser Ebene liegenden Schichten ist, muss hier die Projektionsart gewählt werden. Die darüber liegenden Schichten müssen dann entsprechend der gewählten Projektion die Geoinformationen korrekt abbilden können.

Die Bathymetrie, also die Höheninformationen der Ozeane, kann als zusätzliche Ebene hinzugefügt werden.

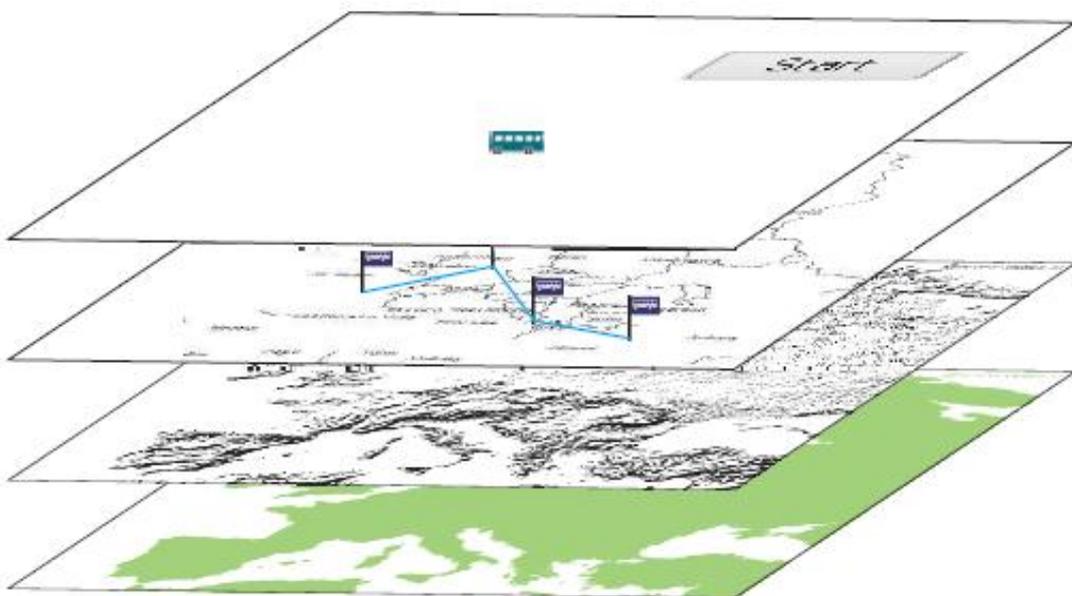


Abbildung 4.1.1: Zusammensetzen eines Kartenausschnittes aus mehreren Schichten [[Kai08](#)]

## 4.2 Quadtree als Datenstruktur

Um mit regulären Gitterstrukturen verschiedene Stufen eines [Kachelservers](#) darstellen zu können, muss eine hierarchische Repräsentation gefunden werden, welche Teile der Karte schrittweise verfeinern kann. Eine solche Multiresolution Repräsentation kann mit einer Quadtree-Struktur erreicht werden. In einer Quadtree-Struktur wird eine quadratische Fläche gleichmäßig in vier Quadranten aufgeteilt. Jeder dieser Quadranten kann dann wiederum in vier weitere Quadranten aufgeteilt werden.

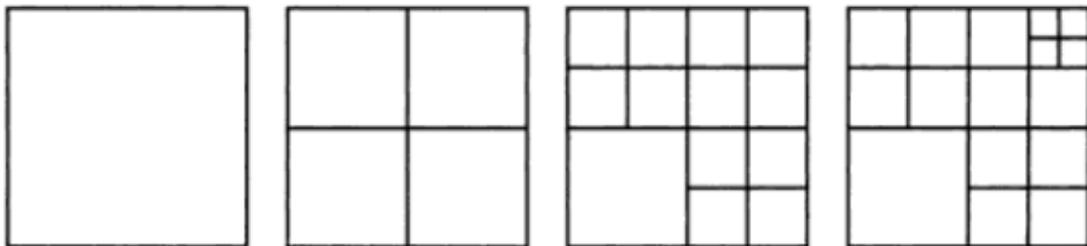


Abbildung 4.2.1: Rekursive Verfeinerung eines Quadtrees [\[Ste09\]](#)

## 4.3 Zusammenfassung

Die in Kapitel 2 und in Kapitel 3 gefundenen Grundlagen und Anforderungen dienen als Grundlage für das Design. Es wird eine Schichtenarchitektur verwendet, um die nötige Skalierbarkeit des Geoinformationssystems zu gewährleisten. Der Unterschied zu den vorhandenen Geoinformationssystemen zeigt sich in der hinzugefügten Informationsschicht, die für die relevanten Szenario Informationen des abgekapselten Szenario Systems bereits steht. Diese Architektur erlaubt es der Informationsschicht, Informationen aus den darunter liegenden Schichten zu benutzen. Des Weiteren erfüllt diese Architektur die Anforderung [A5](#), da dadurch Positionsäquivalenz der Informationen in allen Schichten gewährleistet wird. Es wird eine Quadtree als Datenstruktur benötigt, um mit den Kartenteilen eines [Kachelservers](#) zu Recht zu kommen.

## Kapitel 5

# Werkzeuge und Geodatenquellen

In [Digitale Geodaten](#) wurden die Geodaten näher beschrieben. Dieses Kapitel beschäftigt sich mit der Interpretation von Roh- und aufbereiteten Geodaten, die durch Werkzeuge visualisiert und bearbeitet werden können. Des Weiteren werden die in dieser Arbeit verwendeten Quellen der Geodaten vorgestellt und näher beschrieben. Auch deren Projektionen und Probleme werden erläutert.

## 5.1 Geodatenquellen

Im folgenden Abschnitt werden konkrete Geodatenquellen genannt. Der Prototyp wird diese Daten nutzen, um die Welt zu modellieren. In dem Abschnitt

Schichtenarchitektur wurde bereits festgelegt, dass zur Modellierung der Welt mindestens vier Schichten notwendig sind. Zu jeder Geodatenquelle wird hinzugefügt, in welche Schicht die Datenquelle zugeordnet wird.

### 5.1.1 SRTM Höheninformationen

Die Shuttle Radar Topography Mission (SRTM) war eine Mission der NASA im Weltall in Februar 2010. Das Ziel dieser Mission war es, hochauflösende Höheninformationen von der Oberfläche der Welt zu sammeln. [\[NAS10a\]](#) Während der SRTM wurden 80% von der Oberfläche der Erde gewonnen. Die Daten können in zwei Detailstufen aufgerufen werden: 1:90 Meter pro Pixel für die Erde und 1:30 Meter pro Pixel für die USA. Die erste Version der

Daten war sehr ungenau und unklar. Des Weiteren hatte dieser Datensatz viele Löcher, beziehungsweise fehlende Pixel, die jedoch mit passenden Werkzeugen ausgefüllt werden konnten. Des Weiteren mussten Küstenlinien nachgezeichnet werden.

In 2005 wurde eine fertige Version mit weltweiter Abdeckung fertiggestellt. Diese Geodaten sind sehr detailgenau und haben kaum Lücken. Mit den Geodaten von United States Geological Survey (USGS) [USG05] können die Lücken aufgefüllt werden, womit man eine weltweite Abdeckung der Höheninformationen ohne Lücken erhält. [YU05]

Die Geodaten im Rohformat HGT sind unter [NAS09a] zu finden. Es bedarf an speziellen Werkzeugen, welche in Kapitel 5.2 vorgestellt werden, um diese Dateien zu interpretieren und weiterzuverarbeiten. Jedoch sind bereits interpretierte Sekundärdaten der SRTM bei der NASA zu finden [NAS09b], wodurch das portieren nicht mehr nötig ist.

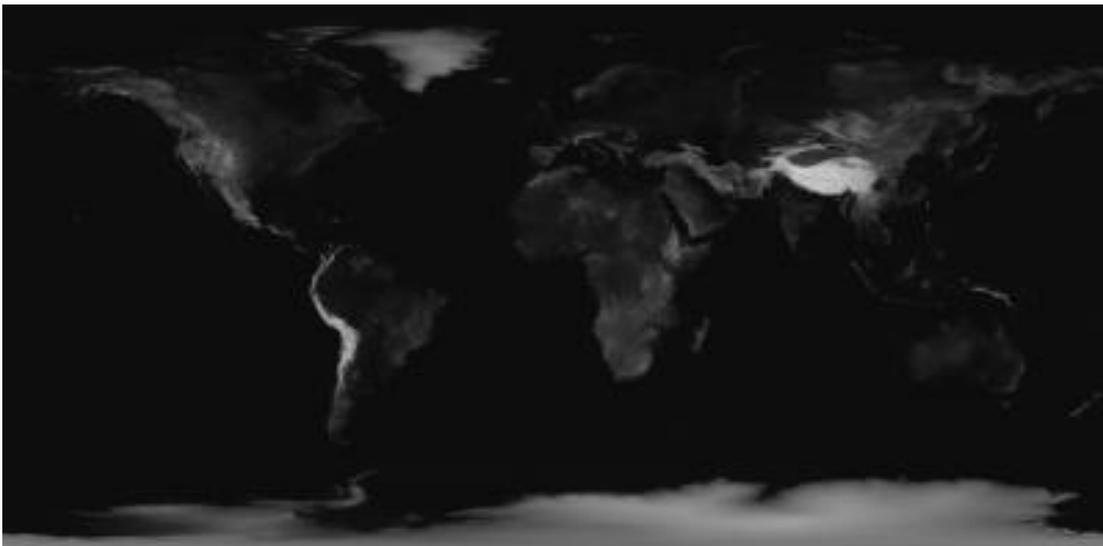


Abbildung 5.1.1: SRTM-Sekundärdaten [NAS00]

Diese Geodaten bieten die Grundlage zur Modellierung der Welt und werden in den Prototypen verwendet.

### 5.1.2 NASA Blue Marble - Next Generation

Die National Aeronautics and Space Administration (NASA) wurde am 29. Juli 1958 gegründet und ist die US-Bundesbehörde für Luft- und Raumfahrt. Die NASA untersucht nicht nur das Weltall, sondern betreibt auch Forschung

## WERKZEUGE UND GEODATENQUELLEN

---

zu unserer Erde. Die ersten von der NASA gesammelten Bilder der Erde waren schwarz weiß. Schon hier erkannte die NASA, dass aus den Beobachtungen der Erde von weiter Ferne Rückschlüsse gezogen werden konnten, die zum Beispiel die Umwelt oder das Wetter betrafen.

Auf den Bildern der Astronauten, die an den Apollo Missionen teilgenommen haben, ist die Erde erstmals in Farbe zu sehen. 1972 hat die Crew der Apollo 17 das bisher atemberaubendste Foto der Erde eingefangen.

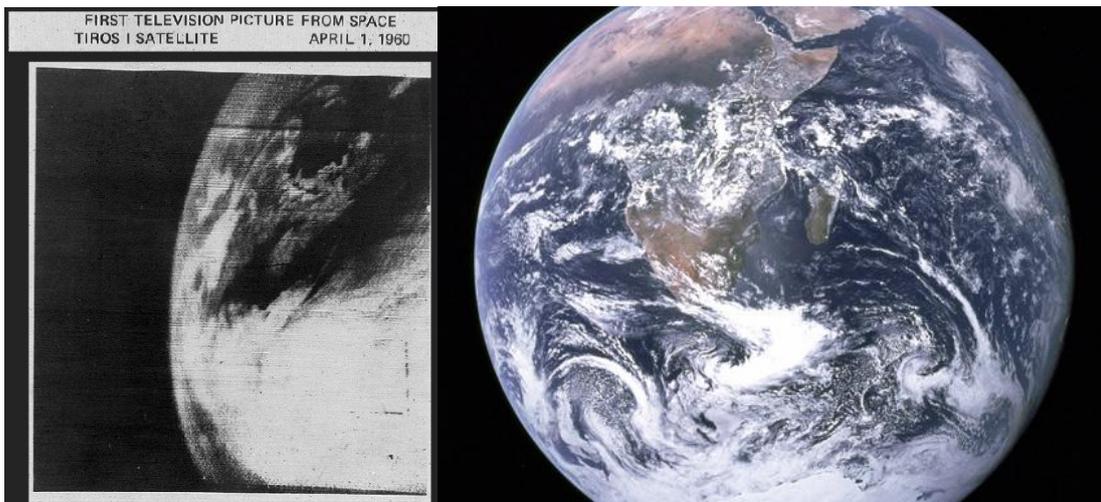


Abbildung 5.1.3:  
Bild der Erde von TIROS1  
[\[NAS60\]](#)

Abbildung 5.1.2:  
Die Erde aus Sicht der Crew von  
Apollo 17 [\[NAS72\]](#)

Blue Marble, ein von der NASA am meisten detaillierte und farbige Fotosatz der Erde, wurde in 2002 produziert. Diese Bilder hatten eine Auflösung von ein Quadratkilometer pro Pixel. Des Weiteren waren die Bilder wolkenlos. In Oktober 2005 haben die Forscher der bisher veröffentlichten Blue Marble Datensätze eine neue Version mit einer doppelt so hohen Auflösung (500 Quadratmeter pro Pixel) und monatlichen Bildern des Jahres 2004 herausgebracht. Diese neue Kollektion wird Blue Marble: Next Generation (BMNG) genannt.

Die BMNG Bilder können als kompletter Satz oder Teile der Erde von der NASA erworben werden. Des Weiteren bietet die NASA die Datensätze in unterschiedlicher Qualität an. Diese Daten sind allgemein zugänglich. [\[NAS05\]](#) Wie bereits bei den [SRTM Höheninformationen](#) sind bereits interpretierte Sekundärdaten bei der NASA zu finden [\[NAS09b\]](#).

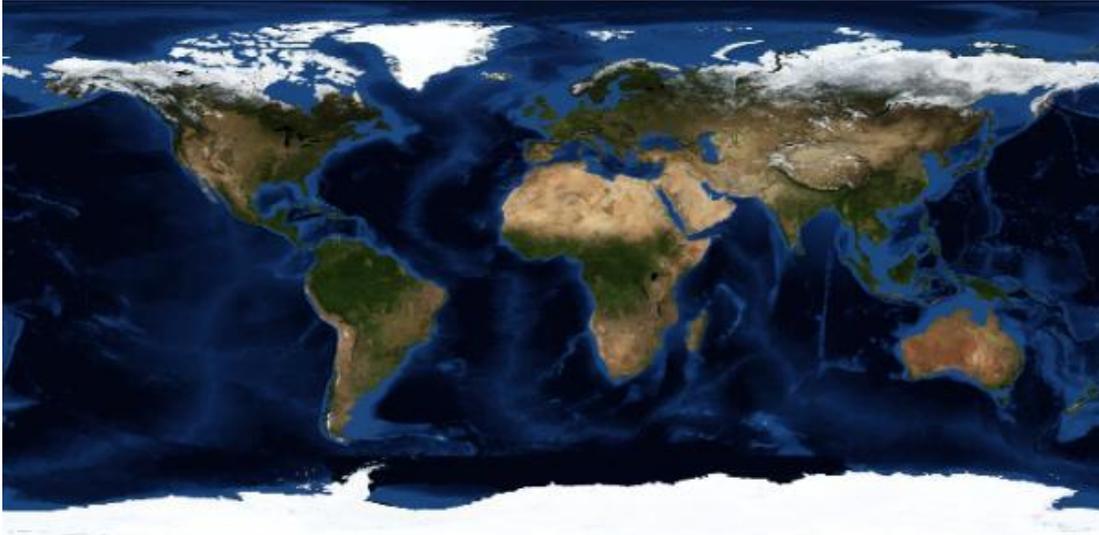


Abbildung 5.1.4: BMNG Sekundärdaten [NAS10b]

Die BMNG Sekundärdaten werden als [Topographie](#) für diese Arbeit verwendet. Sie bieten eine Auflösung von 500 Quadratmeter pro Pixel und sind von unterschiedlicher Qualität, wodurch eine Feinjustierung bei den zu ladenden Datenmengen entstehen kann. Die BMNG Datensätze sind, wie auch die [SRTM Höheninformationen](#), von der Projektion her eine [Quadratische Plattkarte](#).

### 5.1.3 NASA Earth Observations

Bei NASA Earth Observations (NEO) handelt es sich um eine Datenbank der NASA, welche speziell für klimatische und umweltbedingte Einflüsse der Welt eingeführt worden ist. Ziel dieses Services ist, Personen und Programmen den Zugriff auf solche Daten zu ermöglichen, damit diese analysiert und bezüglich des Aufgabenkontextes interpretiert werden können. NEO ist ein allgemein zugänglicher Service.

Die von NEO generierten Bilddaten eignen sich hervorragend für die Verwendung in der [Informationsschicht](#), da die Geodaten georeferenziert sind und Informationen liefern, die nach Interpretation sich für ausgewählte Szenarien eignen. Die Datensätze können auch als zusätzliche Texturen verwendet werden. [Sal06] Auch wenn in dieser Arbeit das Szenario von dem HLS verwendet wird, dient diese Geodatenquelle für weitere informative Untersuchungen.

Die NEO Geodaten sind von der Projektion her eine Quadratische Plattkarte.

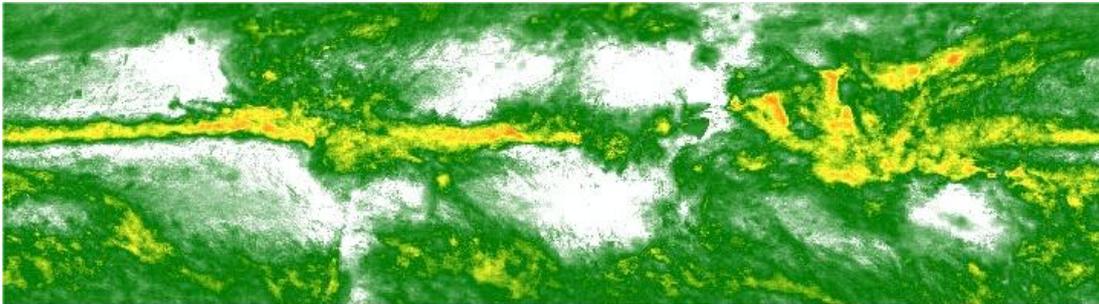


Abbildung 5.1.5: NEO Totaler Niederschlag [NEO10a]



Abbildung 5.1.6: NEO Aktive Brände [NEO10b]

### 5.1.4 Google Static Maps

Mithilfe der Google Maps API [Goo10b] können die Bilder von Google Maps anhand von Parametern angefordert und empfangen werden. Das API bietet einige Hilfsprogramme zur Bearbeitung von Karten und zum Hinzufügen von Inhalten. Die Maps-API ist ein von Google kostenloser Service.

Google Static Maps erstellt die Karte auf Grundlage von Parametern, die über eine URL an Google gesendet werden. Google Static Maps-API-URL haben die folgende Form:

<http://maps.google.com/staticmap?parameter>

Es gibt erforderliche und optionale Parameter. Die Kartenbilder werden über folgende Parameter definiert:

## WERKZEUGE UND GEODATENQUELLEN

---

- **center**: Definiert die Kartenmitte mithilfe geografischer Breiten- und Längenangaben.
- **zoom**: Definiert die Zoomstufe der Karte.
- **size**: Definiert die Abmessungen des Kartenbildes in Pixeln.
- **format**: Legt das Format des Bildes (GIF, JPEG oder PNG) fest.
- **maptype**: Definiert den Typ der zu erzeugenden Karte (satellite, terrain, hybrid, mobile).

Die Google Maps Daten sind akkurat und haben eine sehr feine Auflösung (Bis zu 0.1 Meter pro Pixel [Wik10]). Man kann auf einigen Bildern sogar Gesichter erkennen. Google Static Maps benutzt die [Mercator-Projektion](#).

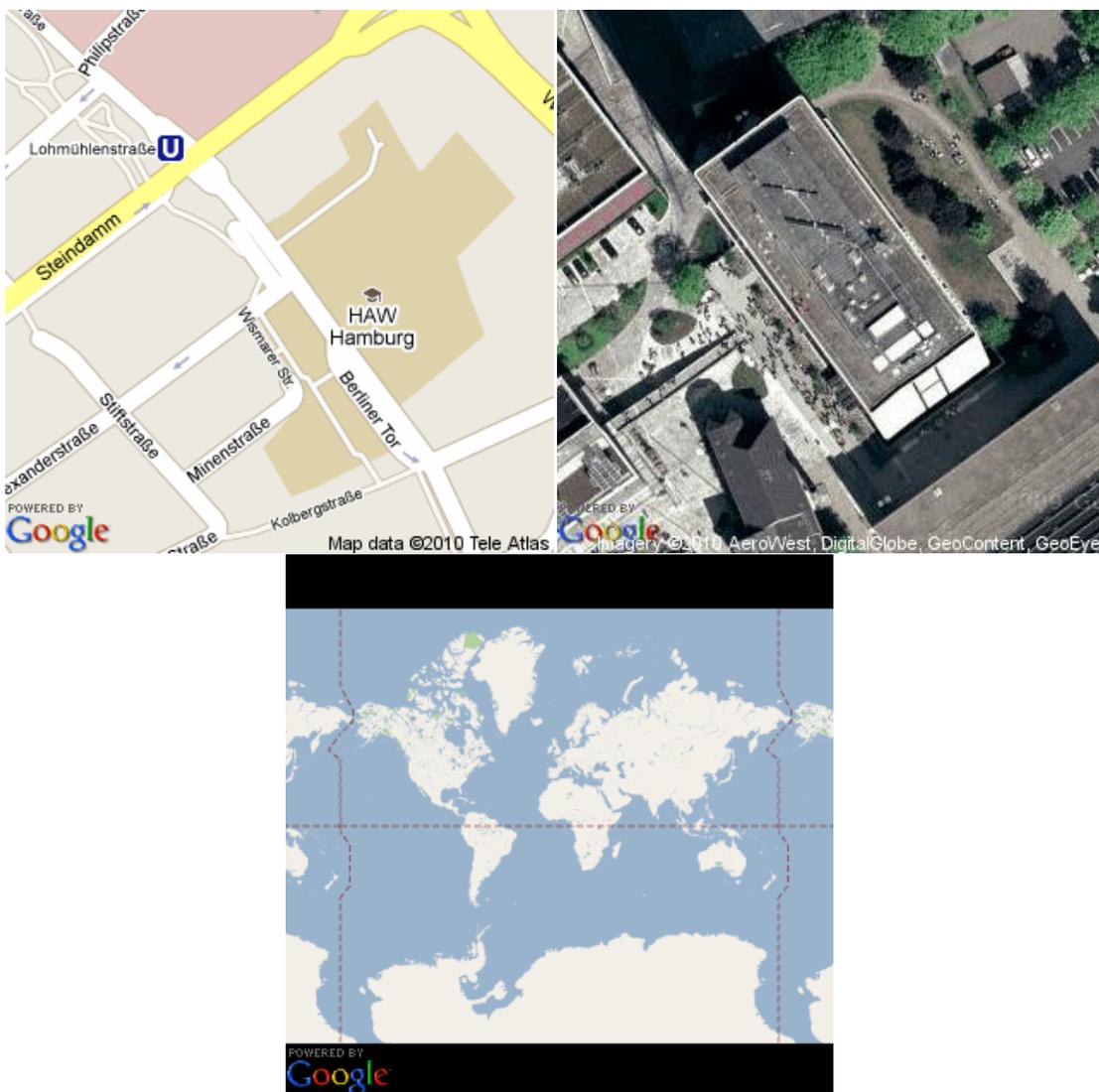


Abbildung 5.1.7: Google Maps Aufnahmen

### 5.1.5 Weitere Texturen

Wie bereits in Kapitel 4.1.2 beschrieben besitzen weitere Texturen kaum nennenswerte Informationen. Sie dienen der verbesserten Darstellung und zeigen, dass der Prototyp auch weitere Texturen anzeigen lassen kann. Sinnvolle Texturen, die die Erde haben kann, welche nicht in den Geodaten enthalten sind, sind die Wolken oder Politische Linien.

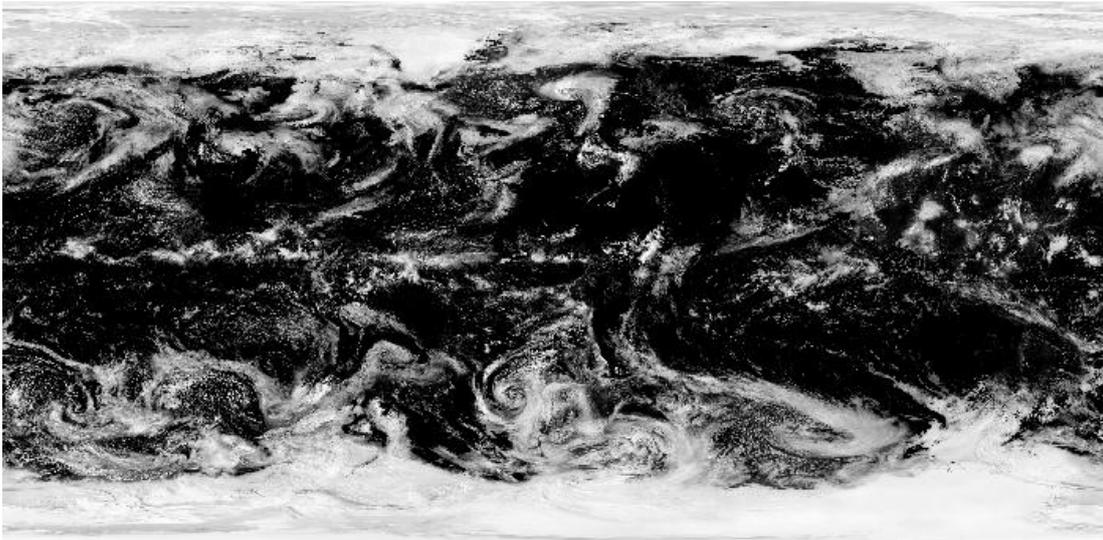


Abbildung 5.1.8: Wolkenbild der Erde [Oea03]

## 5.2 Verwendete Werkzeuge

In diesem Abschnitt werden die in dieser Arbeit verwendeten Werkzeuge zur Interpretation von Geoinformationen. Einige dieser Werkzeuge unterstützten das Verständnis über die Thematik, andere dienen zur Bearbeitung und Konvertierung von Geodaten.

### 5.2.1 Earth Sculptor

Earth Sculptor [Ern04] ist ein Werkzeug, um ein Gelände zu kreieren und es mit verschiedenen Farben und Texturen auszustatten. Dieses Werkzeug wird in Kunstprojekten, Geographischen Visualisierungen und im Spieleentwicklungsbereich verwendet. Mithilfe von Earth Sculptor kann man ein Gelände von Grund auf erstellen, oder man lädt als Bilddateien gespeicherte Höhenin-

## WERKZEUGE UND GEODATENQUELLEN

---

formation als Grundlage für das Relief. Earth Sculptor speichert die erstellten Gelände wieder als Bilddateien ab, welche dann weiter verarbeitet werden können.

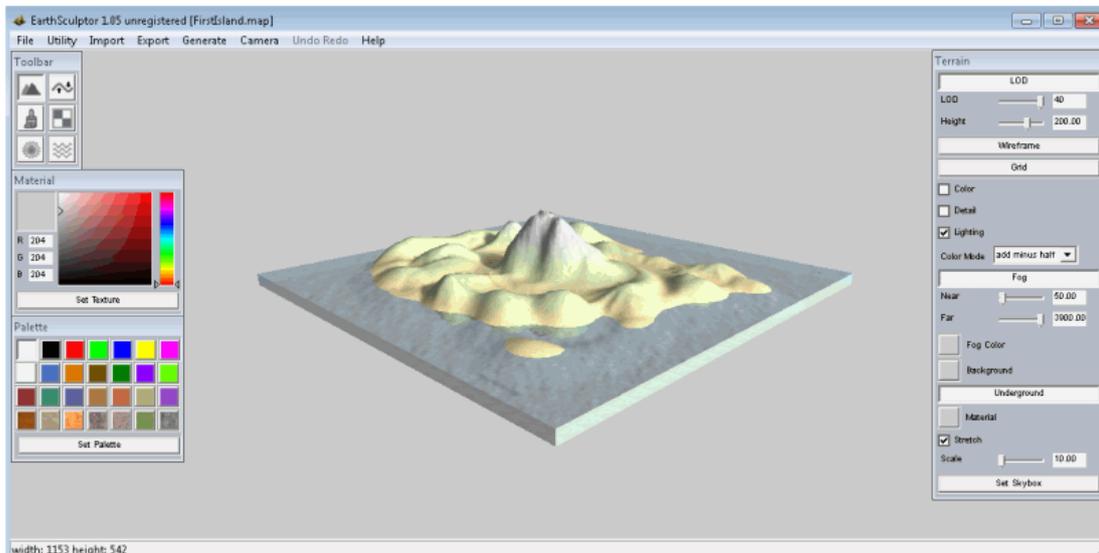


Abbildung 5.2.1: Oberfläche von Earth Sculptor

In dieser Arbeit wurde Earth Sculptor am Anfang benutzt. Dieses Werkzeug diente als Einarbeitung in das Framework und der Thematik. Durch Earth Sculptor wurde das Verständnis von [Höheninformation](#), Texturen und Geländeerstellung vertieft. Dadurch konnte eine geeignete [Schichtenarchitektur](#) gefunden werden, welche als Basis für die Entwicklung des Geoinformationssystems dient.

### 5.2.2 Global Mapper

Global Mapper ist ein sehr mächtiges Werkzeug zur Anzeige, Bearbeitung und Konvertierung von Raster-, Höhen- und Vektordaten. Global Mapper kann verschiedene Ebenen laden und gleichzeitig anzeigen, wie zum Beispiel Höheninformationen und topographische Karten. Das Ergebnis kann wiederum in ein geeignetes Format exportiert werden. Des Weiteren verfügt Global Mapper über eine Anbindung zu verschiedenen Servern, die Geodaten anbieten. Somit muss nicht erst nach Geodaten gesucht werden – man sucht sich die benötigten Geodaten aus dem Menü aus. Flächen- und Entfernungsberechnungen werden unterstützt.

Unglücklicherweise können keine spezielle Szenarien oder Simulationen ein-

gebaut werden, weshalb Global Mapper sich nicht als Lösung zur Szenario basierten Visualisierung von Geodaten eignet, aber grundlegende Funktionen zur Bearbeitung und Konvertierung bereitstellt.

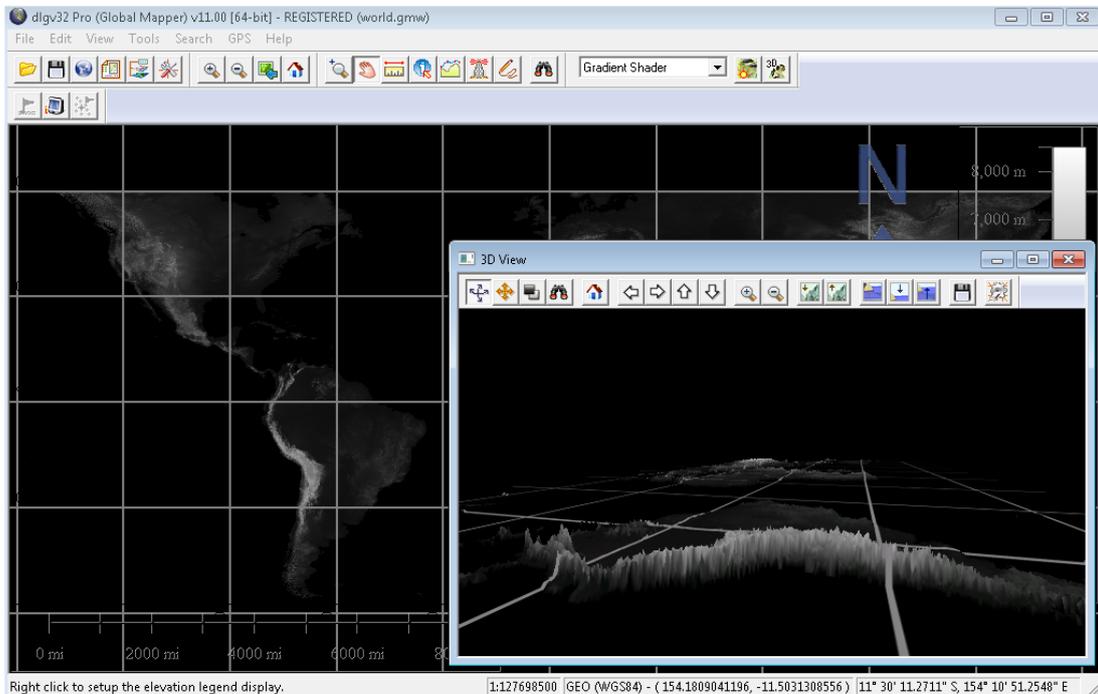


Abbildung 5.2.2: Oberfläche und 3D Ansicht von Global Mapper

### 5.3 Zusammenfassung

Aus der Anforderung [A2](#) und den Erkenntnissen aus Kapitel [5.1.2](#) kann man schließen, dass es sinnvoll ist, NASA Blue Marble - Next Generation Geodaten und Google Static Maps zeitgleich zu verwenden. Die NASA Geodaten bieten eine maximale Auflösung von 500 Quadratmeter, während Google eine Auflösung von bis zu 0.1 Meter pro Pixel bietet. Optimal wäre es dementsprechend, wenn sich die Geodaten abhängig von der internen Entfernung zur modellierten Erde ändern würden.

Problematisch wird es sein, bei unterschiedlichen Projektionen eine korrekte Georeferenzierung zu erzielen, welche aber für die [Schichtenarchitektur](#) essentiell ist. Im Kapitel [Projektionen](#) wurde bereits darauf hingewiesen.

Geodaten gibt es in hochauflösender Qualität. Sollte es zu Performanz Problemen aufgrund der Qualität der Geodaten kommen, so muss der Prototyp eine Lösung anbieten, dass die Geodaten intern so skaliert werden, damit

## *WERKZEUGE UND GEODATENQUELLEN*

---

der Prototyp flüssig läuft. Des Weiteren verbrauchen die hochauflösenden Bilder viel Speicherplatz – bei Performanz Einbußen muss nach einer Streaming Lösung gesucht werden.

Zusammenfassend lässt sich in Anbetracht der vorangegangenen Kapitel sagen, dass Geodatenquellen verfügbar sind, und das Geoinformationssystem sich an dem Szenario von HLS orientieren wird. Nun fehlt noch ein geeignetes Framework, welches die Modellierung einer Welt anhand von Bilddateien zur Verfügung stellt, und dabei die Möglichkeit bietet, das HLS Szenario zu simulieren. Des Weiteren sollte das entwickelte Geoinformationssystem mit hochauflösenden Geodaten keine Performanz Probleme hervorrufen.

## Kapitel 6

### Implementation

In den vorangegangenen Kapiteln wurden die Grundlagen, die Analyse, das Design und die verwendeten Geodatenquellen und Werkzeuge vorgestellt. Dieses Kapitel beschäftigt sich um die konkrete Implementation des Prototyps, welcher in der Analyse und dem Design definiert wurde. Dabei wird der Werdegang, von der Einarbeitung bis zum fertigen Prototypen, beschrieben. Des Weiteren werden in diesem Kapitel Probleme beschrieben und gelöst, die sich während der Entwicklung mit dem verwendeten Framework aufgezeigt haben und nicht in der Analyse oder Design ermittelt werden konnten.

#### 6.1 Einarbeitung

Bevor mit der Implementierung des Prototypens angefangen werden konnte, fand eine Einarbeitung in die Materie statt. Dabei wurde, nachdem das Verständnis über Geodatenquellen ausgebaut war, mit dem Werkzeug [Earth Sculptor](#) erste Gelände entworfen.

##### 6.1.1 Geländeerstellung

Mithilfe der Exportfunktion konnten die generierten Gelände in Formate gebracht werden, die von anderen Programmen verwendet werden können. So kann man das erschaffene Gelände in verschiedene Bilddateien exportieren. Es entsteht jeweils eine Bilddatei für die Höheninformationen, die Textur und das Licht. Aus dem in Kapitel [5.2.1](#) gezeigtem Gelände entstehen folgende Bilddateien:

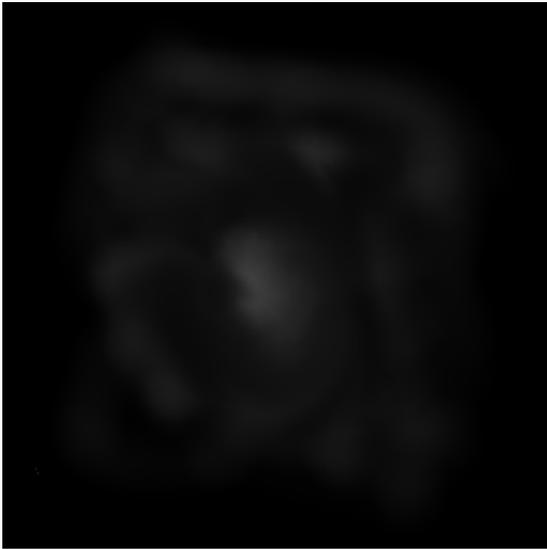


Abbildung 6.1.2:  
Exportierte Höheninformationen

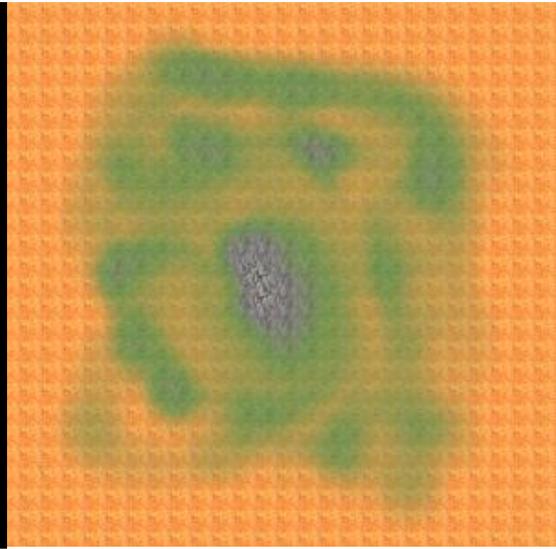


Abbildung 6.1.1:  
Exportierte Textur

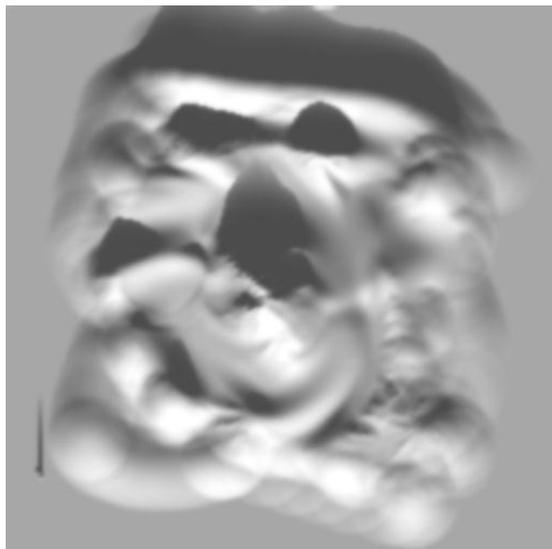


Abbildung 6.1.3:  
Exportiertes Licht

Zu diesem frühen Zeitpunkt wurde klar, dass ein Gelände aus mehreren Schichten besteht. Einige dieser Schichten sind essentiell, andere optional. Anzumerken ist, dass in diesen Schichten keinerlei Informationen enthalten sind, die dem Gelände einen Kontext geben. Es fehlt noch eine **Informationsschicht**. Aufgrund dieser Erkenntnisse konnte eine **Schichtenarchitektur** für den Prototyp entwickelt werden, welches bereits in Kapitel 4 erläutert wurde.

### 6.1.2 Framework

Das Design und die Analyse geben Aufschluss über das Framework, welches verwendet werden sollte. Das Framework sollte die Erstellung eines Geländes mithilfe von Höheninformationen in Bildformat anbieten können. Dabei muss das Framework das Modell dreidimensional darstellen und mit Texturen versehen können.

Die jMonkey Engine (jME) ist eine performante szenegraphbasierte dreidimensionale Grafik API, welche in und für Java geschrieben wurde. Eine szenegraphbasierte Architektur erlaubt die Organisation von relevanten Daten in einer Baumstruktur. Dabei werden die Blätter des Baumes auf den Bildschirm visualisiert, die Elternknoten speichern relevante interne Informationen und erlauben ein hierarchisches Abstraktionsmodell.

Mithilfe der eingebauten Kamera werden nur die Objekte gezeichnet, die auch im Sichtbereich des Betrachters sind. Somit wird eine schnelle Projektion der dreidimensionalen Objekte geboten.

Geometrische Objekte werden als Polygone modelliert. Polygone sind Vielecke, welche bei einem dreidimensionalen Framework aus Dreiecken bestehen.

jME bietet Klassen an, die ein Gelände erstellen können. Dabei muss zwischen zwei Komponenten unterschieden werden: Zum einen jene, die die interne Struktur des Geländes laden und speichern kann, und die Geometrie, welche dann von jME auf den Bildschirm gezeichnet werden. Die Geometrie bedient sich der Informationen, die sie von der Abstrakten Geländekomponente erhält. Vorteilhaft an dieser Unterteilung ist, dass die Geometrie unabhängig von der internen Information verändert und bearbeitet werden kann. Außerdem trennt man Aufgabenbereiche, um nur die relevanten Objekte auf den Bildschirm zu zeichnen.

Jedes Geometrieobjekt hat intern Informationen über die Art und Weise, wie es gezeichnet werden soll. Die Informationen wie Material oder Reflektion der Geometrie wird über sogenannte Renderer (Zeichner) definiert. Jedes Geometrieobjekt besitzt mehrere Renderer, und jeder Renderer ist für eine bestimmte Aufgabe vorgesehen. Somit gibt es auch Grafikrenderer, die Informationen über Texturen oder Farbe der Geometrie speichern.

Um Texturen oder Bilder in das Framework zu laden, bedarf es einen sogenannten Manager. Für jede Art von Laden gibt es einen Manager, wie zum Beispielen einen Objektmanager oder einen Texturemanager. Diese Manager verwalten das Laden und Speichern von externen Daten und sind essen-

## IMPLEMENTATION

tiell bei der Verwendung von Bildern.

Die Abhängigkeiten und den Verlauf von Geodatenquellen zu Geometrieobjekten mit Texturen werden in den nachfolgenden Komponentendiagrammen verständlich gemacht.

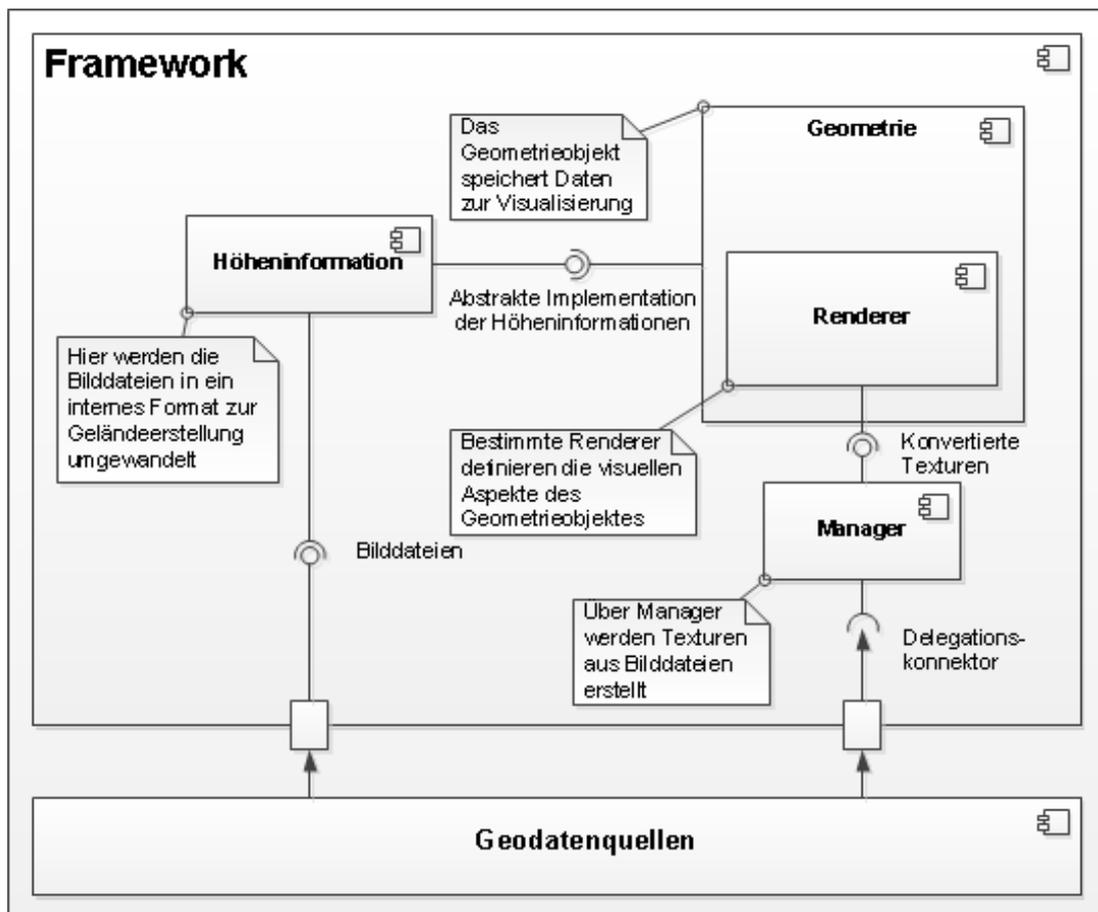


Abbildung 6.1.4: Ablauf der Konvertierung von Geodatenquellen zu Geometrieobjekten im Framework

### 6.1.3 Probleme

In Kapitel 5.3 wurde beschrieben, dass verschiedene Geodatenquellen abhängig von der Entfernung des Betrachters im dreidimensionalen Raum angezeigt werden sollten. Konkret für die Implementierung heißt dies, dass wir auf einen Datentyp oder eine Klasse zurückgreifen müssen, welcher zwischen verschiedenen Geometrieobjekten wechseln kann. Des Weiteren muss eine Lösung gefunden werden, dass nicht zu viele Dreiecke gleichzei-

tig angezeigt werden, um einen performanten Ablauf des Programms zu gewährleisten. Die hochauflösenden Höheninformationen werden zu diesem Problem führen. Abschließend lassen sich folgende noch zu lösende Probleme zusammenfassen:

- **Projektionsproblem:** Unterschiedliche Projektionen in den Geodaten müssen von dem Prototypen erkannt und behandelt werden. Siehe Kapitel 2.2.2.
- **Verschiedene Geodaten:** Abhängig von der Entfernung des Betrachters unterschiedliche Geodaten. Siehe Kapitel 5.3.
- **Hochauflösende Geodaten:** Bei Performanz Einbußen muss eine Lösung gefunden werden, damit der Prototyp stabil läuft. Siehe Kapitel 5.3.

## 6.2 Realisierung

In diesem Abschnitt wird die konkrete Implementierung des Prototyps näher beschrieben. Des Weiteren werden Problemstellungen, die während der Entwicklung des Prototyps aufgetreten und bereits angesprochen wurden behandelt und gelöst.

### 6.2.1 Stadium 1

Im ersten Stadium ging es um die Erstellung des Geländes anhand von Höheninformationen. In Kapitel 6.1.2 wurde erwähnt, dass dazu die Geodaten in eine Bilddatei konvertiert werden müssen (wenn sie noch nicht als Bilddatei vorliegen), um eine `ImageBasedHeightMap` zu generieren. Mit dieser konkreten `ImageBasedHeightMap` wird dann eine `TerrainPage` erstellt, die in Blöcke aufgeteilt wird. Dadurch wird erreicht, dass nur die relevanten sichtbaren Blöcke auf den Bildschirm gezeichnet werden. Des Weiteren sollte eine einfache Textur über das Geometrieobjekt gelegt werden. Als Geodaten wurden für die Höheninformationen die [SRTM Höheninformationen](#) und für die Textur die [NASA Blue Marble - Next Generation](#) Datensätze verwendet.

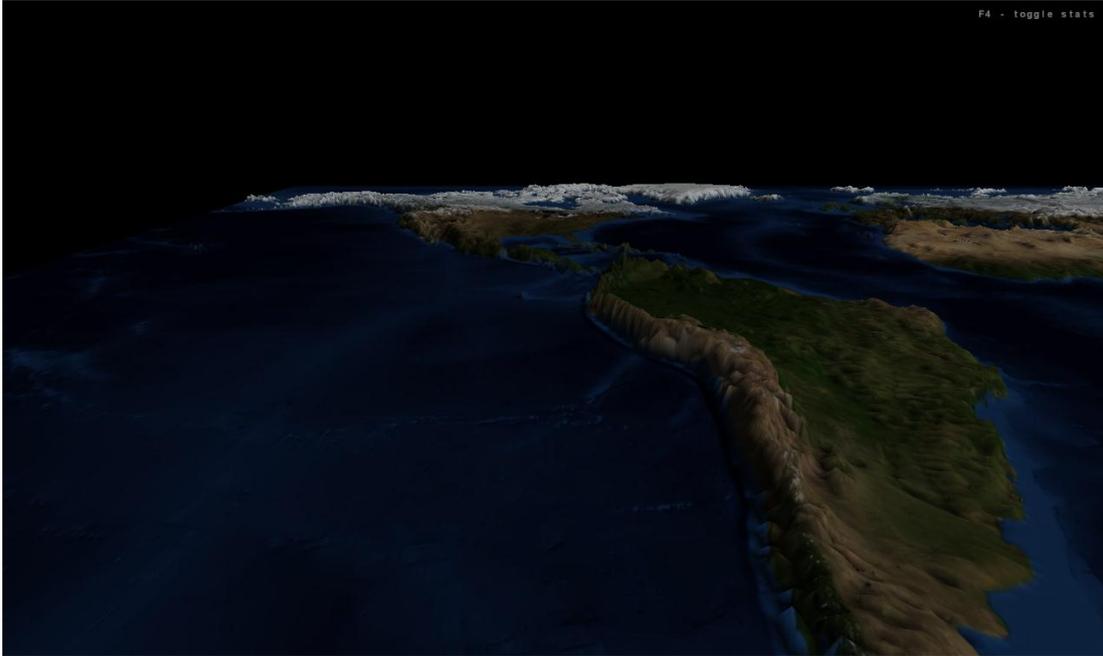


Abbildung 6.2.1: Bild einer flachen dreidimensionalen Erde, Amerika



Abbildung 6.2.2: Bild einer flachen dreidimensionalen Erde, Eurasien

### 6.2.2 Stadium 2

Im zweiten Stadium wurde das Problem der Hochauflösenden Geodaten behandelt. Des Weiteren wurde eine Abbildungsvorschrift für die Umwandlung von Projektionen gefunden. Man muss zwischen zwei Teilproblemen unterscheiden:

- Hochauflösende Höheninformationen und
- Hochauflösende Geodaten, die als Textur verwendet werden.

Der Unterschied liegt darin, dass die Höheninformationen als Grundgerüst des Geländes von jME verwendet, die Texturen aber nur über die Objekte gelegt werden. Problematisch bei den Höheninformationen ist, je hochauflösender das Grundgerüst, desto mehr Dreiecke werden bei dem Geometrieobjekt verwendet, und desto schlechter ist die Performanz. Ziel ist es, möglichst wenige Dreiecke zu zeichnen, wobei die Qualität nicht zu stark darunter leiden sollte. Dieses Verhalten nennt sich Level Of Detail. Bei den Texturen ist problematisch, dass zu viel Speicher verbraucht wird, da die Sekundärdaten Gigabytegröße besitzen.

Die Lösung zu dem ersten Problem hat den Nebeneffekt, dass man unterschiedliche Geometrieobjekte für die unterschiedlichen Levels erschaffen muss. Da jedes Geometrieobjekt über seine eigene Textur verfügt, können somit verschiedene Geodaten abhängig von der Entfernung des Betrachters verwendet werden.

jME bietet für genau dieses Problem die passende Lösung: Die DistanceSwitchModel Klasse. Bei dieser Klasse handelt es sich um einen Container, welcher verschiedene Objekte speichern kann. Dabei kann man angeben, ab welcher Entfernung der Kamera zu dem Container welches Objekt angezeigt werden soll. Genau dieses Verhalten wird für die Realisierung von unterschiedlichen Geodaten benötigt.

Das Problem der unterschiedlichen Projektionen musste nun gelöst werden, da unterschiedliche Geodaten verwendet werden. Die [SRTM Höheninformationen](#) und die [NASA Blue Marble - Next Generation](#) Geodaten besitzen die [Quadratische Plattkarte](#) als Projektion, die Geodaten von

[Google Static Maps](#) aber sind in der [Mercator-Projektion](#). Man benötigt eine Umwandlung von der einen Projektion in die andere. Die Umwandlung von

## IMPLEMENTATION

---

Mercator-Projektion zur Quadratische Plattkarte kann durch folgende Abbildungsformel beschrieben werden:

$$\rho(v) = \log \tan \left( \frac{\pi}{4} + \frac{v}{2} \right)$$

Jeder Bildpunkt der Mercator-Projektion wird mithilfe dieser Abbildungsfunktion in die Projektion der Quadratische Plattkarte gebracht. [Wol08] Das Ergebnis dieser Abbildung ist auf den nachfolgenden Bildern zu sehen.

```
public BufferedImage plateToMercator(BufferedImage img, double minlat, double maxlat) {
    BufferedImage mercatorImg = new BufferedImage(5400, 5400, Transparency.OPAQUE);
    for (int x = 0; x < img.getWidth(); x++) {
        for (double y = minlat; y < maxlat; y += 0.003) {
            int newX = x;
            double newY = Math.log(Math.tan(Math.PI / 4 + Math.toRadians(y) / 2));
            double consta = 2700 / Math.PI;
            int rgb = img.getRGB(x, (int) (Math.toRadians(y) * consta) + 2700 / 2);
            mercatorImg.setRGB(newX, (int) (newY * consta) + 2700, rgb);
        }
    }
    return mercatorImg;
}
```

Abbildung 6.2.3: Implementation in Java

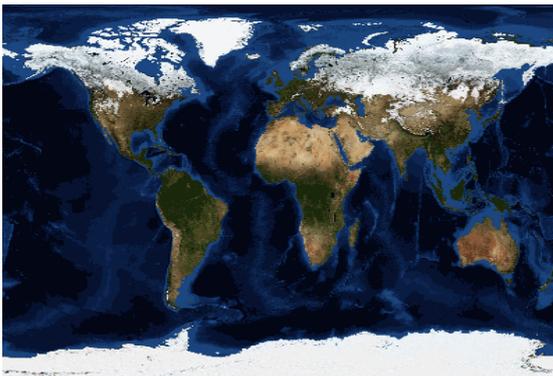


Abbildung 6.2.5:  
Quadratische Plattkarte



Abbildung 6.2.4:  
Mercator Projektion



Abbildung 6.2.7: Level 0 des Prototypen,  
Höheninformationen und Google Maps (Satellite)

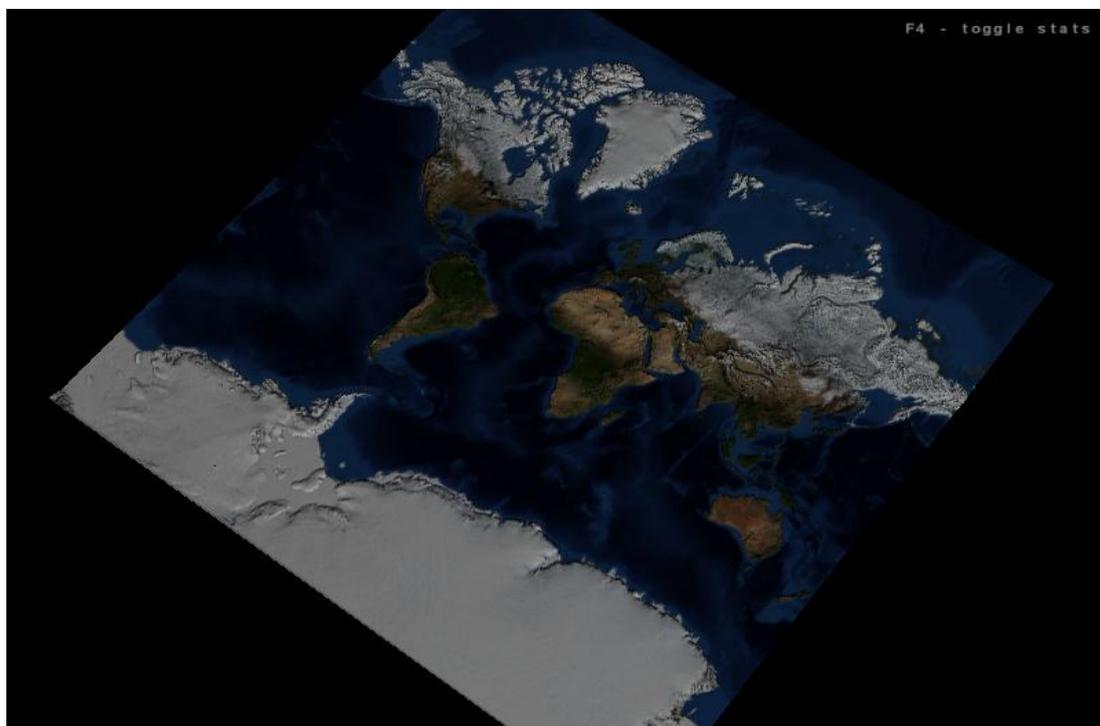


Abbildung 6.2.6: Level 1 des Prototypen,  
Höheninformationen und BMNG

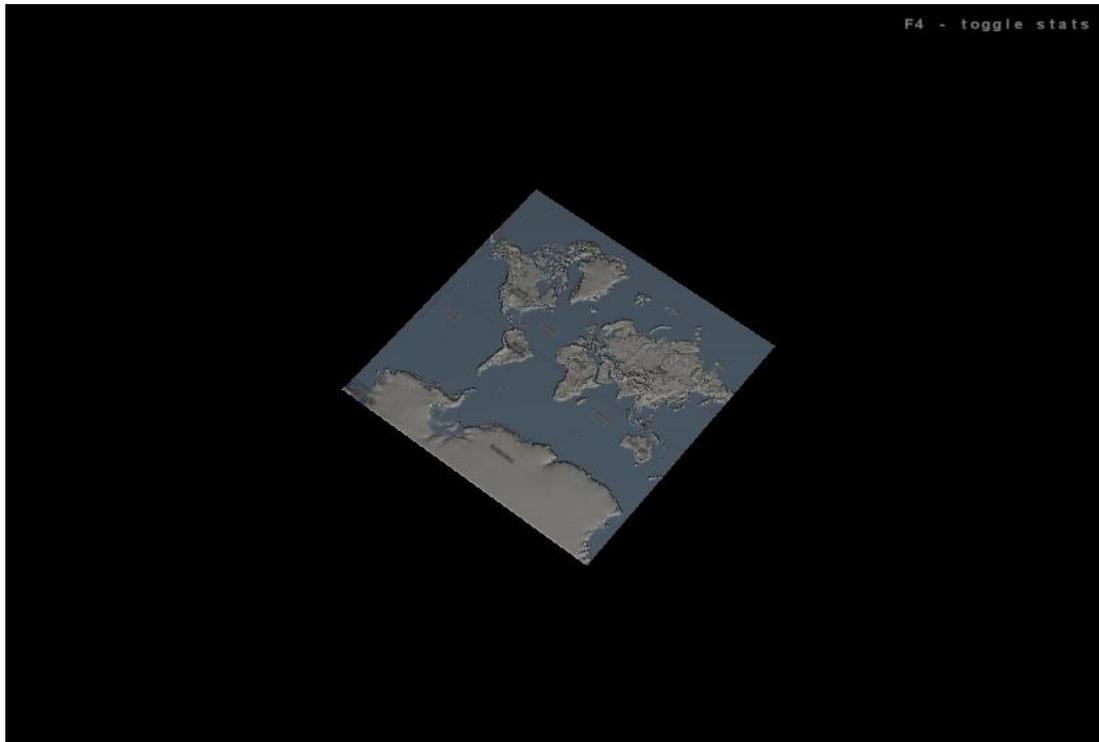


Abbildung 6.2.8: Level 2 des Prototypen, Höheninformationen und Google Maps (Street)

Bei dem zweiten Problem muss eine Lösung gefunden werden, wie das Geoinformationssystem sich bei großen Datenmengen verhält. Es wurde auf zwei Mechanismen zurückgegriffen. Zum einen werden die Sekundärdaten im Vorwege herunterskaliert, damit generell nicht viel Speicher verbraucht wird. Zum anderen wird das Streamen von Geodaten implementiert. Der entscheidende Vorteil beim Streamen ist, dass die Daten nicht beim Systemstart komplett geladen werden, sondern während der Laufzeit auf die Geometrieobjekte gelegt werden. Dabei ist es wichtig, dass die Geodaten in kleinere Bilddateien unterteilt werden, damit der Prototyp nicht das komplette Paket, sondern nur die relevanten Geodaten verwendet. Dieser Nebeneffekt bei der Implementation des Streamen erlaubt es außerdem, Geodaten von [Google Static Maps](#) zu nutzen, da diese Geodaten, wenn die richtigen Parameter gewählt wurden, bereits unterteilt sind. Das Geoinformationssystem muss zu den vorhandenen TerrainBlocks die passenden Texturen Anfragen können. Die Datenstruktur der TerrainPages wird durch eine Quadtree-Struktur (Siehe [Quadtree als Datenstruktur](#)) definiert, was eine solche Aufteilung in einzelne Kacheln erlaubt. Eine Hilfsklasse erstellt die nötigen URL-Anfragen, die Anfragen an den [Google Static Maps](#) Service senden.

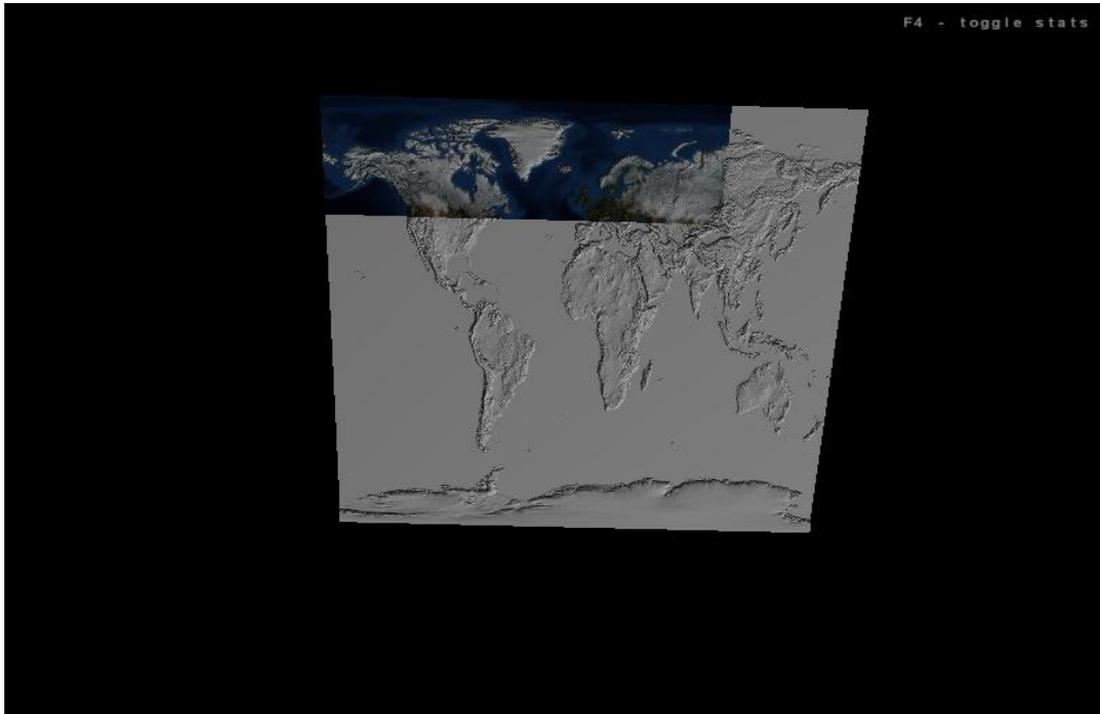


Abbildung 6.2.9: Systemstart, Streaming der einzelnen Kacheln

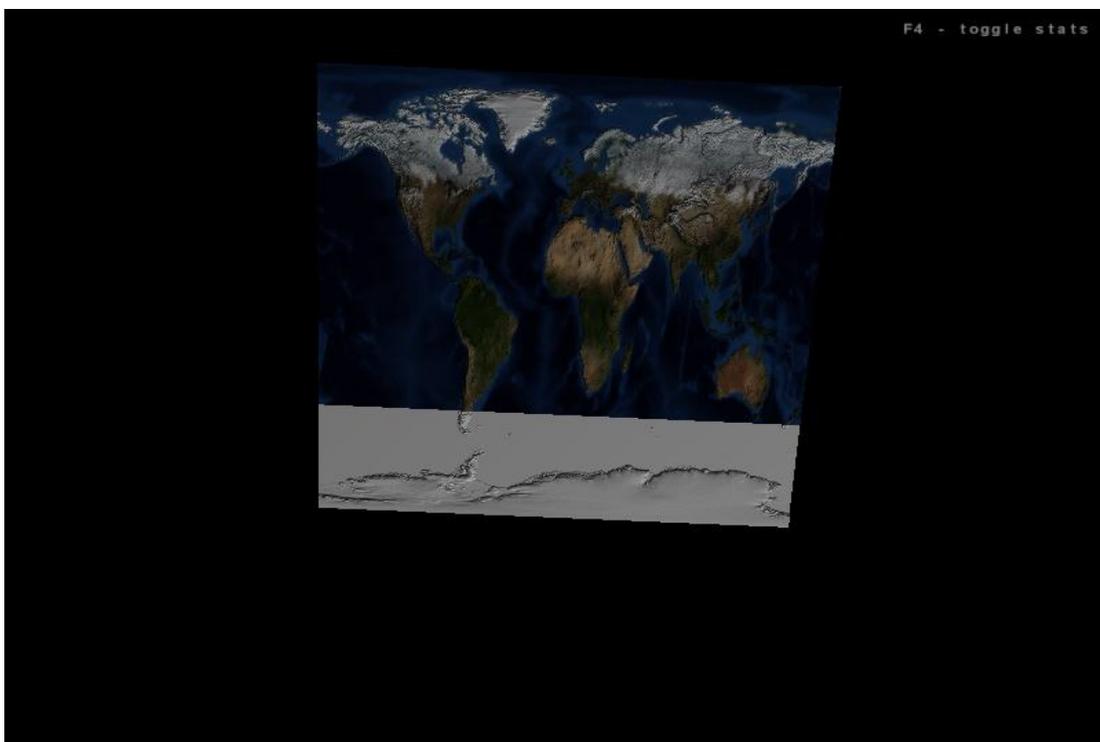


Abbildung 6.2.10: Vervollständigung des Streaming  
Prozesses der einzelnen Kacheln

### 6.2.3 Stadium 3

Die Implementation der Schichtenarchitektur ist in den vorherigen Stadien abgeschlossen worden. In diesem Stadium soll eine Möglichkeit gefunden werden, das flache dreidimensionale Gelände in eine kugelförmige Erde zu formen. jME bietet dies nicht von Grund auf an, weshalb nach einer Lösung gesucht werden musste. In den jME Foren gab es einen interessanten Beitrag [\[jME06\]](#), welcher dann auf eine Abbildungsfunktion hinauslief.

Jeder Punkt auf dem flachen Gelände wird in ein kugelförmiges Modell abgebildet. Dabei muss die Position des Punktes  $(x, y)$  die Projektion einer Kugel entsprechen [\[Dut98\]](#). Der  $z$ -Wert (die Höhe) wird dann nur noch auf den ausgerechneten Wert addiert. Folgende Formeln lassen sich für einen Punkt  $(x, y)$  aufstellen:

$$p(x) = \cos(x) \sin(y)$$

$$p(y) = \sin(x)$$

$$p(z) = \cos(x) \cos(y) + z$$

Die Implementation der TerrainBlocks-Klasse von jME erlaubt es, Koordinaten zu lesen und neu zu setzen. In diesem Fall genügt es nun, jeden Koordinatenpunkt zu besuchen und diese Formeln anzuwenden, um den neuen Punkt zu generieren.



Abbildung 6.2.11: Ergebnis nach der Umwandlung in eine Kugel

### 6.2.4 Stadium 4

Im letzten Stadium wurde die flache dreidimensionale Erde in eine kugelförmige Erdkugel umgewandelt. Dies war der letzte Schritt, um den Prototypen in eine für diese Arbeit ausreichende lauffähige Version zu entwickeln. In Stadium 4 wurde das Streaming der Daten verbessert. Durch einen Threadpool und im Hintergrund laufende Aufgaben kann die Performanz des Prototyps noch weiter verbessert werden.

jME ist ein sequentielles Framework und erlaubt keine Nebenläufigkeit von Aufgaben. Aufgrund dessen kann nicht erst gewartet werden, bis die Geodaten aus dem Internet geladen worden sind, da diese sehr groß ausfallen und eine lange Ladezeit haben. Konkret wurde deshalb auf Future Objekte und Threadpools zurückgegriffen. Durch die Future Interfaces kann eine Aufgabe in einen Threadpool gelegt werden. Der Stärke dieses Interfaces liegt darin, dass der Job zwar asynchron erledigt wird, man aber abfragen kann, ob der Job bereits erledigt worden ist, und wenn ja, sich dann das Ergebnis holen lässt.

```
if(noch_keine_Textur_auf_dem_Geometrieobjekt) {  
    // Schaue, ob die Textur bereits im Threadpool ist.  
    if(Textur_im_Threadpool) {  
        // Schaue, ob der Job bereits fertig ist.  
        if(job.isDone?) {  
            // Lege Textur auf Geometrieobjekt.  
            gObjekt.setTextur(job.getResult());  
        }  
    } else { // Die Textur ist noch nicht im Threadpool.  
        // Erstelle ein Futureobjekte und lege es in den Threadpool.  
        Future<BufferedImage> worker = bild_aus_url(url);  
        threadpool.füge_worker_hinzu(worker);  
    }  
}
```

Abbildung 6.2.12: Pseudocode der generierten Nebenläufigkeit im sequentiellen Ablauf

## Kapitel 7

### Auswertung

Moderne Geoinformationssysteme erweitern die Nutzungsmöglichkeiten der klassischen Landkarte. Die Visualisierung der Daten ist die primäre Funktion dieser Geoinformationssysteme, weitere Funktionen, wie die Konvertierung und Portierung der Daten, dienen zur Analyse. Geoinformationssysteme benutzen Geodaten der Erde, um die Erde in einem Programm zu modellieren und zu visualisieren. Jedes Geoinformationssystem muss sich mit der Problematik der unterschiedlichen Projektionen und Formate der Geodaten auseinandersetzen. Des Weiteren sollte ein Geoinformationssystem mit den Datenmengen der Geodaten zu Recht kommen, um eine flüssig laufende Anwendung zu gewährleisten. Optional können verschiedene Abbildungsvorschriften angewandt werden, die die Erde von einer flachen Karte in ein dreidimensionales Modell transformiert.

Das Ziel dieser Arbeit war es, ein Geoinformationssystem bereit zu stellen, welches nicht nur die Erde visualisiert, sondern in diesem erstellten Modell ein ausgewähltes Szenario simuliert. Eine Simulation von bestimmten Ereignissen ist notwendig, wenn das Ausgangssystem zu komplex, unüberschaubar oder der Ausgang des Ereignisses nicht reproduzierbar ist. Des Weiteren gibt es viele verschiedene Szenarien, weshalb eine einfache Anbindung nicht möglich ist. In den meisten Fällen werden Geoinformationssysteme unabhängig von Szenarien erstellt, weshalb die gängigen Geoinformationssysteme keine Schnittstellen für Szenarien bieten. In dieser Arbeit wurde ein Geoinformationssystem erstellt, welches als Grundlage für die Schnittstelle zu Szenarien Systemen ein von der HAW Hamburg erstelltes LogistikszENARIO verwendet. Dadurch konnte ein Geoinformationssystem erstellt werden, welches sich nicht nur auf die Visualisierungs- und Bearbeitungskomponente beschränkte, sondern bereits im Design eine Anbindung zu diesen Szenario

Systemen bot, um eine Grundlage für ein Geoinformationssystem mit Szenario Generierung zu schaffen.

Grundlegende Funktionen, die ein modernes Geoinformationssystem haben sollte, wurden implementiert. Somit kann das entwickelte Geoinformationssystem zwischen unterschiedlichen Formaten und Projektionen unterschieden und gegebenenfalls Umwandlungen und Abbildungsvorschriften nutzen, um eine korrekte Visualisierung zu gewährleisten.

Das entwickelte Geoinformationssystem unterstützt eine Schichtenarchitektur, um verschiedene Schichten der Erde gleichzeitig zu visualisieren. Dadurch können eine Vielzahl von Datensets nahtlos zu einem Globus zusammengefügt und dargestellt werden. Im Gegensatz zu gängigen Geoinformationssystemen besitzt diese Schichtenarchitektur über eine Informationsschicht, welche die Informationen über die darunter liegenden Schichten verwendet, um die nötigen Informationen an den gegebenen Positionen zusammenfasst. Die einzelnen Schichten werden zusammengefasst und visualisiert.

Während der Entwicklung wurde das Problem mit den großen Datenmengen der Geodaten erkannt. Hochauflösende Geodaten verbrauchen viel Speicher und benötigen dementsprechend viel Zeit, wenn sie von einem Geoserver aus dem Internet geladen werden. Es wurden verschiedene Mechanismen und Lösungen angewandt, um dieses Problem zu lösen. Dabei wurde auf Future Objekte und auf eine [Quadtree als Datenstruktur](#) zurückgegriffen. Dadurch wurde das Streaming von [Kachelservern](#) ermöglicht, wodurch verschiedene Geodatenquellen zeitgleich verwendet werden konnten.

Das Geoinformationssystem visualisiert die Erde nicht nur als zweidimensionale Karte, sondern erlaubt eine kugelförmige Modellierung als Projektionsart.

Der in den Bildern in Kapitel 6 gezeigte Prototyp, welcher als Ergebnis des hier entwickelten Geoinformationssystem anzusehen ist, ist im Anhang zu finden.

## **Kapitel 8**

### **Diskussion und Ausblick**

Das hier entwickelte Geoinformationssystem dient als Basis für weitere Arbeiten und Forschungen. Es wurden aufgrund der begrenzt verfügbaren Zeit bestimmte Punkte nicht behandelt, die für die weitere Bearbeitung von Bedeutung wären.

jME unterstützt das Culling von einigen Objekten nicht. Unter Culling wird eine Technik in der Computergrafik bezeichnet, die nicht sichtbare Polygone beziehungsweise Dreiecke von der zu zeichnenden Szene entfernt. Dadurch erhöht sich die Darstellungsgeschwindigkeit. Dies ist bei der großen Datenmenge der Geodaten von Nöten, da sonst Daten angefordert werden, die (noch) nicht im Sichtbereich des Benutzers sind. Da die Erde eine kugelförmige Projektion besitzt, stellt sich hier die Herausforderung, nur die Objekte zu zeichnen, die nicht durch andere Objekte verdeckt werden (Occlusion Culling).

Das Geoinformationssystem ist noch nicht dafür ausgelegt, weder mit Objekten zu agieren, noch sie zu interpretieren. Es fehlt eine entsprechende Architektur, die Objekte in das vorhandene System einpflegt. Unter Objekten sind Fahrzeuge, Gebäude und Lebewesen zu verstehen.

Das hier verwendete Logistikscenario diente als Grundlage für die Entwicklung, wurde jedoch nicht in den Prototypen eingebunden, da das Szenario noch nicht ausgereift genug war, um relevante Informationen dem Geoinformationssystem mitzuteilen. Wünschenswert wäre es, den Prototypen mit einem Szenario anzureichern, um das hier entwickelte Konzept zu testen.

## Literaturverzeichnis

- [BP97] Brandon, Plewe: *GIS Online - Information retrieval, mapping and the Internet*. OnWord Press, 1997. ISBN 1-56690-137-5.
- [Dut98] Dutch, Steven, University of Wisconsin: *Spherical Projections*. <http://www.uwgb.edu/dutchs/structge/sphproj.htm>. Version 1998.
- [Ern04] Szoka, Ernest: *EarthSculptor. Terrain Editor*. <http://www.earthsculptor.com/index.htm>. Version 2004.
- [Goo10a] Google: *Google Static Maps API*. <http://code.google.com/apis/maps/documentation/staticmaps/>. Version 2010.
- [Goo10b] Google: *Google Maps. Google Maps API*. <http://code.google.com/intl/de/apis/maps/documentation/staticmaps/>. Version 2010.
- [HLS10] Prof. Dr. Stefan Sarstedt, Prof. Dr. Thomas Thiel-Clemen: *HAW Logistics System. Lastenheft HAW Logistics System*. <http://www.elearning.ls.haw-hamburg.de/mod/resource/view.php?inpopup=true&id=26281>. Version 2010.
- [jME06] jME Forum: *Random Mapped Sphere?* <http://www.jmonkeyengine.com/forum/index.php?PHPSESSID=7ee090d26514e796810bbc81fd20f9b7&topic=3625.0>. Version 2006.
- [Kai08] Winter, Kai, Fachhochschule Wiesbaden: *QMapControl – Ein Qtopia-Widget für Kartenanwendungen auf mobilen Geräten*. <http://medieninf.de/qmapcontrol/download/arbeit.pdf>. Version 2008.

## LITERATURVERZEICHNIS

---

- [NAS00] NASA: *NASA SRTM. SRTM Topography.*  
[http://dds.cr.usgs.gov/srtm/version2\\_1/Documentation/SRTM\\_Topo.pdf](http://dds.cr.usgs.gov/srtm/version2_1/Documentation/SRTM_Topo.pdf). Version 2000.
- [NAS02] NASA: *Equirectangular Projection.* Version 2002.
- [NAS05] NASA: *NASA BMNG. NASA Unveils its Blue Marble: Next Generation.*  
[http://www.nasa.gov/vision/earth/features/blue\\_marble.html](http://www.nasa.gov/vision/earth/features/blue_marble.html). Version 2005.
- [NAS08] NASA: *NASA World Wind.* <http://worldwind.arc.nasa.gov/java/>.  
Version 2008.
- [NAS09a] NASA: *NASA SRTM. CGIAR CSI SRTM.*  
<http://srtm.csi.cgiar.org/>. Version 2009.
- [NAS09b] NASA: *NASA Visible Earth. NASA Datasets.*  
<http://mirrors.arsc.edu/nasa/>;  
[http://visibleearth.nasa.gov/view\\_rec.php?id=8391](http://visibleearth.nasa.gov/view_rec.php?id=8391). Version 2009.
- [NAS09c] NASA: *SRTM.* [http://dds.cr.usgs.gov/srtm/version2\\_1/SRTM3/](http://dds.cr.usgs.gov/srtm/version2_1/SRTM3/).  
Version 2009.
- [NAS10a] NASA: *NASA SRTM. SRTM Elevation Data.*  
<http://www2.jpl.nasa.gov/srtm/>. Version 2000.
- [NAS10b] NASA: *World Topography. SRTM RAMP2.*  
[http://mirrors.arsc.edu/nasa/topography/srtm\\_ramp2.world.5400x2700.jpg](http://mirrors.arsc.edu/nasa/topography/srtm_ramp2.world.5400x2700.jpg). Version 2010.
- [NAS60] NASA: *Image of the Earth from TIROS1.*  
[http://www.nasa.gov/vision/earth/features/bm\\_gallery\\_3.html](http://www.nasa.gov/vision/earth/features/bm_gallery_3.html).  
Version 1960.
- [NAS72] NASA: *Apollo 17. Image of the Earth from Apollo 17.*  
[http://www.nasa.gov/vision/earth/features/bm\\_gallery\\_1.html](http://www.nasa.gov/vision/earth/features/bm_gallery_1.html).  
Version 1972.
- [NEO10a] NASA: *NEO - Total Rainfall (1 month - TRMM).*  
<http://neo.sci.gsfc.nasa.gov/Search.html?group=39>. Version 2010.

## LITERATURVERZEICHNIS

---

- [NEO10b] NASA: *NEO - Active Fires (1 month - Terra/MODIS)*.  
<http://neo.sci.gsfc.nasa.gov/Search.html?group=8>. Version 2010.
- [Oea03] Oera: *Texture maps of Earth and Planet - Gallery*.  
<http://www.oera.net/How2/TextureMaps2.htm>. Version 2003.
- [PGI10] Refractions Research: *PostGIS - PostGIS Home*.  
<http://postgis.refractions.net/>. Version 2010.
- [Roh] Rohwedder, Lars H: *Normal Mercator projection*.
- [RBS00] Dr.-Ing. Böhm, Rolf: *Reliefschummerung - Quellen für Höhenmodelle*.  
[http://www.reliefschummerung.de/rs\\_in\\_01\\_dem\\_datenquellen\\_x0022.html](http://www.reliefschummerung.de/rs_in_01_dem_datenquellen_x0022.html). Version: 2000.
- [Ret05] Stöckli, Reto; Vermote, Eric; Saleous, Nazmi; Simmon, Robert; Herring, David: *The Blue Marble Next Generation - A true color earth dataset including seasonal dynamics from MODIS. Blue Marble Next Generation*.  
<http://earthobservatory.nasa.gov/Features/BlueMarble/bmng.pdf>.  
Version 2005.
- [Ste09] Holm, Stefan, University of Zürich: *Data Management for Terrain Rendering*.  
<http://www.ifi.uzh.ch/pax/uploads/pdf/publication/1010/Bachelorarbeit.pdf>. Version 2009.
- [Sal06] Salomonson, Vince; Ward, Kevin: *NEO - NASA Earth Observations. About NEO*.  
<http://neo.sci.gsfc.nasa.gov/View.html?page=AboutNEO>. Version 2006.
- [USG05] USGS: *Welcome to USGS*. <http://www.usgs.gov/>. Version 2005.
- [Wik10] Wikipedia; Google: *Google Earth*.  
[http://en.wikipedia.org/wiki/Google\\_Earth](http://en.wikipedia.org/wiki/Google_Earth). Version 2010.
- [Wol08] Wolfgang, St. Technische Universität Wien: *Gekrümmte Erdkugel – Flache Landkarte, Geometrie und Kartenentwürfe*.  
<http://www.geometrie.tuwien.ac.at/havlicek/slides/strobl2008.pdf>.  
Version 2008.

## LITERATURVERZEICHNIS

---

- [Wen 10] Wensing, Matthew: *How can I get latitude, longitude from x, y on a Mercator map (JPEG)?*  
<http://stackoverflow.com/questions/1166059/how-can-i-get-latitude-longitude-from-x-y-on-a-mercator-map-jpeg>. Version 2010.
- [YU05] University, Yale: *The Center for Earth Observation. Obtaining and Importing SRTM Elevation Data.*  
[http://www.yale.edu/ceo/Documentation/SRTM\\_Elevation\\_Data.pdf](http://www.yale.edu/ceo/Documentation/SRTM_Elevation_Data.pdf). Version 2005.

## Abbildungsverzeichnis

Abbildung 2.2.1: Mercator-Projektion [Oea03] .....	14
Abbildung 2.2.2: Quadratische Plattkarte [Oea03] .....	14
Abbildung 2.2.3: Eine Karte bei verschiedenen Zoomstufen [Kai08].....	15
Abbildung 2.4.1: Fachliche Architektur eines Geoinformationssystems .....	17
Abbildung 4.1.1: Zusammensetzen eines Kartenausschnittes aus mehreren.. Schichten [Kai08] .....	24
Abbildung 4.2.1: Rekursive Verfeinerung eines Quadrees [Ste09] .....	25
Abbildung 5.1.2: SRTM-Sekundärdaten [NAS00] .....	27
Abbildung 5.1.3: Die Erde aus Sicht der Crew von Apollo 17 [NAS72] .....	28
Abbildung 5.1.4: Bild der Erde von TIROS1 [NAS60].....	28
Abbildung 5.1.5: BMNG Sekundärdaten [NAS10b] .....	29
Abbildung 5.1.6: NEO Totaler Niederschlag [NEO10a].....	30
Abbildung 5.1.7: NEO Aktive Brände [NEO10b].....	30
Abbildung 5.1.8: Google Maps Aufnahmen.....	31
Abbildung 5.1.9: Wolkenbild der Erde [Oea03] .....	32
Abbildung 5.2.1: Oberfläche von Earth Sculptor .....	33
Abbildung 5.2.2: Oberfläche und 3D Ansicht von Global Mapper .....	34
Abbildung 6.1.1: Exportierte Textur.....	37
Abbildung 6.1.2: Exportierte Höheninformationen.....	37
Abbildung 6.1.3: Exportiertes Licht.....	37
Abbildung 6.1.4: Ablauf der Konvertierung von Geodatenquellen zu .....	
Geometrieobjekten im Framework .....	39
Abbildung 6.2.1: Bild einer flachen dreidimensionalen Erde, Amerika .....	41

## ABBILDUNGSVERZEICHNIS

---

Abbildung 6.2.2: Bild einer flachen dreidimensionalen Erde, Eurasien .....	41
Abbildung 6.2.3: Implementation in Java .....	43
Abbildung 6.2.4: Mercator Projektion .....	43
Abbildung 6.2.5: Quadratische Plattkarte .....	43
Abbildung 6.2.7: Level 1 des Prototypen, Höheninformationen und BMNG .	44
Abbildung 6.2.6: Level 0 des Prototypen, Höheninformationen und Google .... Maps (Satellite) .....	44
Abbildung 6.2.8: Level 2 des Prototypen, Höheninformationen und Google ... Maps (Street) .....	45
Abbildung 6.2.9: Systemstart, Streaming der einzelnen Kacheln .....	46
Abbildung 6.2.10: Vervollständigung des Streaming Prozesses der einzelnen Kacheln .....	46
Abbildung 6.2.11: Ergebnis nach der Umwandlung in eine Kugel .....	47
Abbildung 6.2.12: Pseudocode der generierten Nebenläufigkeit im .....	
sequentiellen Ablauf .....	48

# Anhang

## Inhalt der DVD

Auf der beigelegten DVD befinden sich in den folgenden Ordnern:

<b>Bachelorarbeit</b>	Die Bachelorarbeit als PDF-Datei
<b>Quellen</b>	Die als PDF verfügbaren Dokumente
<b>Quellcode</b>	Der Sourcecode des Prototypen

---

## **Versicherung über Selbstständigkeit**

*Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §22(4) bzw. §24(4) ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.*

Hamburg, 16.08.2010

---

Baldowski Mariusz