



**HTWM**

Hochschule für Technik und  
Wirtschaft Mittweida

Technikumsplatz 17  
09648 Mittweida

# Praktikumsbericht

(Thema: Erstellung eines Onlinespiels)

Erstellt von:  
Isabel Drost (if99wP1)

[SendMail@isabel-drost.de](mailto:SendMail@isabel-drost.de)  
<http://www.isabel-drost.de>



## **Inhaltsverzeichnis**

<b>1</b>	<b>Einführung.....</b>	<b>5</b>
1.1	Aufgabenstellung.....	5
1.2	Ziel der Aufgabe.....	5
1.2.1	Warum ein Spiel zur ausschließlichen Benutzung im Internet entwickeln? .....	5
1.2.2	Besonderheiten bei der Onlinespiel - Entwicklung .....	6
1.2.2.1	Layout.....	6
1.2.2.2	Technische Probleme.....	6
1.2.3	Grundsätzliche Eigenschaften von Unigen.....	6
1.2.3.1	Spielkatogorie.....	6
1.2.3.2	Gründe für die Positionierung in eine ferne Zukunft.....	6
1.2.3.3	Spielablauf aus Sicht eines Nutzer.....	7
<b>2</b>	<b>Die Bearbeitung der Aufgabe.....</b>	<b>8</b>
2.1	Vor Beginn des Praktikums vorbereitete Ergebnisse .....	8
2.1.1	Aufgabenstellungen, die schon vor Beginn des Praktikums bearbeitet wurden .....	8
2.1.2	Tools, die bei der Vorbereitung des Praktikums Verwendung fanden .....	8
2.2	Entwicklung während des Praktikums .....	9
2.2.1	Zu bearbeitende Aufgabenstellungen während des Praktikums .....	9
2.2.2	Zeitplan bei der Entwicklung.....	9
2.2.3	Einarbeitungszeiten .....	10
2.2.3.1	Die erste Entwicklungsphase .....	10
2.2.3.2	Die zweite Entwicklungsphase .....	10
2.2.3.3	Die dritte Entwicklungsphase – die Testphase .....	10
2.2.4	Die verwendete Software.....	11
2.2.4.1	Warum MySQL als Serverdatenbank? .....	11
2.2.4.2	PHP als Scriptsprache? Warum nicht Perl? .....	11
2.2.4.3	Gründe für die Verwendung von C für die Auswertungsfunktion.....	12
2.2.4.4	Entwicklungsumgebung auf dem zur Verfügung gestellten Serversystem.....	12
2.2.5	Bei der Entwicklung verwendete Hardware .....	12
2.2.6	Aussagen zu Qualitätsvorschriften .....	12
2.3	Entwicklung nach Abschluß des Praktikums.....	13
<b>3</b>	<b>Kurze Aussagen zur Struktur des Spiels selbst.....</b>	<b>13</b>
3.1	Vertikale Gliederung der Entwicklung in zwei Teilbereiche .....	13
3.2	Horizontale Gliederung des Spieles in Spielzugarten .....	14
3.3	ERM – das Rückgrat von Unigen .....	14
3.3.1	Grundprinzipien bei der Erstellung des ERM.....	14
3.3.2	Zuordnung der Tabellen zu den einzelnen Kategorien: .....	14
<b>4</b>	<b>Anhang .....</b>	<b>15</b>
4.1	Das ERM von Unigen.....	15

## Abbildungsverzeichnis

Abbildung 1-1-1: Kommunikationsradius einer Kolonie.....	7
Abbildung 2-2: Timeline.....	9
Abbildung 2-3: Warum MySQL? .....	11
Abbildung 2-4: Auswertung von Formulardaten mit PHP.....	11
Abbildung 2-5: Für Unigen zum Einsatz kommendes Serversystem.....	12
Abbildung 3-1: Vertikale Zweiteilung des Spiels.....	13
Abbildung 3-2: Horizontale Gliederung.....	14
Abbildung 3-3: Vertikale Gliederung der Datenbank .....	14
Abbildung 3-4: Gliederung in Nutzerabhängige und -unabhängige Daten, sowie Horizontale Gliederung der Tabellen.....	15



# 1 Einführung

## 1.1 Aufgabenstellung



Wie der Titel des Praktikumsberichts schon andeutet, bestand die Aufgabenstellung im Design, der Entwicklung und dem Test eines Onlinespiels. Unter dem Begriff „Onlinespiel“ ist hierbei keine der üblichen Flashanimationen oder Mini – Javaapplets zu verstehen, die sich zu hauf im Internet finden lassen, sondern die Umsetzung eines Strategie- und Aufbauspieles rein auf der Basis von PHP, C und MySQL. Dabei sollen die zukünftigen Spieler über ein Webinterface, dass mittels der Sprache PHP generiert wird, ihre Spielzüge einstellen können. Diese werden dann pro Tag einmal von einem Auswertungsprogramm berechnet und ausgeführt.

Die Aufgabenbereiche des Praktikums lassen sich dabei in drei Gebiete einteilen: Einmal in die konzeptionelle Entwicklung des Softwareproduktes. Dazu gehörte einmal die Festlegung von konkreten Spielregeln, die Konzeptionierung einer Datenbank, mit der sich die Spielregeln einfach umsetzen lassen sowie eine grobe Erstellung eines Entwurfs des späteren Quellcodes. Dies fand größtenteils schon im Vorfeld des Praktikums statt, um die spätere Arbeit zu erleichtern.

Die zweite Stufe war die Entwicklung des Produktes selbst. Dazu gehört einerseits die Erstellung eines Nutzerinterfaces, mit dem gespielt werden kann. Vorerst handelt es sich dabei lediglich um ein funktionales Interface, mit dem Testdaten generiert werden können. Sobald das Spiel in die Betaphase geht, wird eine Oberfläche erstellt werden, die für die zukünftigen Nutzer ergonomischer und übersichtlicher ist. Andererseits gehörte zur zweiten Phase aber auch die Erstellung eines Programms in der Programmiersprache C, dass die Daten, die die User generieren, auswertet.

Die dritte Stufe umfasst das Testen sowohl der Scripte als auch des Auswertungsprogrammes. Um späteren freien Alphatestern die Arbeit zu erleichtern wurde natürlich entsprechend dem V-Modell schon während der Entwicklung der Scripte und der Auswertung mittels ausgewählter Testdaten die Richtigkeit des bisher Erstellten geprüft. Aufgrund des Umfangs des Projektes kann dies aber durch eine einzelne Person allein nicht bewältigt werden. Daher wurden während der letzten 6 Wochen umfangreiche Tests durchgeführt, bei denen sich ausgewählte Nutzer beteiligen konnten.

Die Aufgabenstellung wurde vom Praktikanten im genannten Umfang allein bearbeitet. Die Idee dazu stammte allerdings von 5 weiteren Personen, die sich bereits einige Jahre früher eine Story sowie einige grundlegende Spielregeln überlegt hatten. Allerdings hatten sie frühzeitig während der Konzeptionierung abgebrochen, als klar wurde, wie umfangreich das Projekt werden würde.

## 1.2 Ziel der Aufgabe

Ziel war die Erstellung einer Rahmenumgebung für ein Onlinespiel. Onlinespiel heißt hier, dass das Spiel nicht nur eine Erweiterung innehat, mit der Nutzer auch im Netz damit spielen können, sondern dass es komplett im Internet gespielt wird und im besten Falle außer einem Browser und einem Internetzugang keine weiteren Voraussetzungen gegeben sein müssen, um teilnehmen zu können.

Aufgrund der Tatsache, dass einem Praktikanten allein das Thema zur Bearbeitung überlassen wurde und kaum konkrete Softwareentwürfe vorlagen, konnte im Vorhinein kein konkreter Entwicklungsstatus angegeben werden, mit dem das Projekt beendet wird.

### 1.2.1 Warum ein Spiel zur ausschließlichen Benutzung im Internet entwickeln?

In der Gemeinde der Jugendlichen, die von Computerspielen begeistert sind, sind schon seit längerer Zeit Spiele beliebt, mit denen man über Netzwerk gegen Freunde und Bekannte antreten kann. In der Vergangenheit hat sich gezeigt, dass auch die Möglichkeit, über das Internet gegen vorerst unbekannte Mitspieler anzutreten, rege genutzt wird. Dies macht es einerseits einfach, Gleichgesinnte kennenzulernen, andererseits kann man so aber auch einfach Tips zur eigenen Spielweise erhalten.

Ein Spiel rein als Onlinespiel zu entwickeln, stellt, je nach Detailgrad der Story, die nachgespielt werden soll, ein Angebot dar, was dem kommerzieller Softwarefirmen durchaus nahe kommen kann. Allerdings steigt natürlich mit wachsendem Detailgrad auch die Belastung an den Server, der dann eine um so umfangreichere Datenbasis verwalten muß.

### 1.2.2 Besonderheiten bei der Onlinespiel - Entwicklung

#### 1.2.2.1 Layout

Einen Nutzer für ein Onlinespiel zu interessieren, stellt den Entwickler vor Probleme, die grundsätzlich denen vergleichbar sind, vor denen auch Anbieter anderer Onlineangebote stehen:

- Derjenige, der die Seite aufruft, muß mit Mitteln des Layouts überzeugt werden, dass das Angebot für ihn oder sie interessant sein könnte. Es muß sich also um ein anspruchsvolles, themengerechtes Layout handeln.
- Die Bedienung des Spiels muß intuitiv und ohne das Lesen von Hilfetexten möglich sein. Allerdings muß sie gleichzeitig dem Thema des Spiels gerecht werden.
- Das Layout soll so gestaltet werden, dass es mit im Idealfall plattformunabhängig auf allen gängigen Browsern (Opera, Konqueror, Netscape 4.7x, IE) lauffähig ist. Alternativ dazu soll mehr als ein Layout entwickelt werden, wobei je nach Browser das passende angezeigt wird.

#### 1.2.2.2 Technische Probleme

- Das Ziel eines solchen Angebots besteht natürlich darin, möglichst viele Nutzer zu begeistern und sie vom Mitspielen zu überzeugen. Allerdings muß auf der anderen Seite für jeden Nutzer eine möglichst hohe Geschwindigkeit bei der Bearbeitung seiner Anfragen garantiert werden können.
- Aufgrund der Tatsache, dass davon auszugehen ist, dass viele Spieler an einem solchen Projekt teilnehmen werden, die entsprechend viel Zeitaufwand in das Spiel stecken, ist eine möglichst hohe Verfügbarkeit des Servers sicher zu stellen. Weiterhin ist darauf zu achten, dass alle Daten der Spieler vertraulich behandelt und vor Angriffen von außen gesichert werden.

### 1.2.3 Grundsätzliche Eigenschaften von Unigen

#### 1.2.3.1 Spielkategorie

Unigen ist vornehmlich als rundenbasiertes Aufbau-Strategiespiel zu betrachten, bei dem Elemente aus Rollenspielen eingearbeitet wurden, um die Kommunikation zwischen den Mitspielern zu erhöhen. Eine Spielrunde verläuft dabei über 24 Stunden, in denen die Spieler ihre Züge einstellen können. Nachts werden diese dann ausgewertet und somit tatsächlich ausgeführt.

Das Ziel des Spiels ist es, einen Planeten mit einer bestimmten Art von Wesen (mensenähnlichen, naturgebundenen, auf die Nutzung von Energie oder auf Kriegsführung spezialisierte Kreaturen) zu besiedeln. Weiteres Ziel ist es, mit anderen Kolonien auf dem eigenen Planeten sowie mit anderen Planeten zu kommunizieren und zu handeln.

Dabei werden diverse soziale Fähigkeiten des Spielers trainiert. Er ist gezwungen, seine Umgebung zu erkunden und mit anderen Mitspielern zu kommunizieren. Weiterhin wird er dazu gezwungen werden, Konflikte zu lösen und unter Umständen Kompromisse einzugehen.

#### 1.2.3.2 Gründe für die Positionierung in eine ferne Zukunft

- Spieler können in der Anfangsphase leicht räumlich voneinander isoliert werden. Dies verhindert einerseits, dass ein neuer Nutzer von zu vielen neuen Eindrücken (Spieler, Bündnisse und deren Relationen untereinander) überrascht wird – er kann sich einfacher und schneller zurechtfinden. Auf der anderen Seite wird es so aber möglich, dass neue Nutzer zuerst im kleinen Kreis an den im Spiel üblichen Umgangston gewöhnt werden, ehe sie mit einem Fehltritt gleich bei allen Mitspielern negativ auffallen.
- Zur Wahl stand außerdem die Positionierung in eine mittelalterliche Umgebung, die allerdings die Möglichkeiten im Spiel historisch begrenzt. Bei der Wahl eines Themas aus dem Fantasybereich sind die Grenzen ebenfalls relativ eng gesetzt (einerseits durch die Erwartungen der Spieler an ein solches Thema, andererseits durch diverse Publikationen zu diesem Thema). Im Gegensatz dazu sind die Spielräume zur Entwicklung von Umgebungen bei der Wahl eines Grundthemas aus dem Science – Fiction Bereich relativ groß.
- Das Thema Science Fiction ist neben Fantasy derzeit relativ beliebt unter Jugendlichen – egal, ob in Form von Geschichten in den Printmedien, als Kinofilm oder als Hintergrundstory zu einem Spiel. Daher sollte die Wahl eines der beiden Bereiche als Grundthema für ein Spiel genügend potentielle Spieler begeistern und anlocken.

### 1.2.3.3 Spielablauf aus Sicht eines Nutzer

Der potentielle Nutzer des Spiels Unigen meldet sich zu Beginn an und wählt grundlegende Parameter seiner zukünftigen Kolonie. Dazu gehört z.B. die Rasse der Bewohner. Mit Beginn des Spieles hat der Spieler keine Möglichkeit mit anderen Kolonien in Verbindung zu treten – er ist vollkommen isoliert. Allerdings ist er in diesem Stadium auch noch verhältnismäßig gut geschützt: Er kann von anderen Mitspielern nur schwer gefunden werden – somit ist es auch schwer, einen neuen Spieler zu demotivieren, indem man ihn schon in einer frühen Phase mit Angriffen zermüht.

In der Anfangsphase des Spiels sollte das Ziel des Nutzers in einer zuverlässigen eigenen Rohstoffproduktion liegen. Dazu müssen Technologien erforscht und schließlich Gebäude aufgebaut werden, in denen unter Nutzung mehr oder minder hoher Technologien eine bestimmte Menge an Rohstoffen produziert wird.

Diese Rohstoffe können dann z.B. über eine spielinterne Auktion gegen andere Rohstoffe getauscht werden. Dabei übernimmt im einfachsten Fall das Auswertungsprogramm die Aufgabe, einen der Bieter für die angebotenen Rohstoffe als Handelspartner auszuwählen.

Parallel dazu können die Kommunikationsmöglichkeiten ausgebaut werden. Hier gibt es grundsätzlich zwei Bereiche, die es gilt, auszubauen:

- „Horchen“ – der Spieler stellt auf der Oberfläche seines Planeten Stationen auf, mit denen er Signale aus dem ihn umgebenden All empfängt. Ausgesendet werden diese von Kolonien, die vergleichbare, sendende Stationen aufgebaut haben. Ob Signale empfangen werden können, hängt davon ab, wie weit die sie aussendende Station entfernt ist und wie stark die empfangende Station ist. In den Spieldaten spiegelt sich das in der Entfernung der Planeten und in der Ausbaustufe der Sende- und Empfangsstationen wieder.
- „Senden“ – Der Spieler baut Sendestationen, mit deren Hilfe er selbst Signale aussenden kann. Die Reichweite seiner Signale wird dabei von der Reichweite der Sendestationen bestimmt.



**Abbildung1-1-1: Kommunikationsradius einer Kolonie**

Bei der Kommunikation wird im Grunde auf das System von Foren zurückgegriffen:

Sobald ein Spieler einen Thread eröffnet, wird dieser im spielinternen Forum der Mitspieler sichtbar, die eine ausreichend starke Empfangsstation besitzen, sichtbar. Sie können dann antworten, wenn sie wiederum eine Sendestation haben, die stark genug ist, den Initiator des Threads zu erreichen.

Die letzte Variante von Spielzügen beschäftigt sich mit dem Aufbau einer Flotte von Raumgleitern. Diese können dann zum direkten Handel mit anderen Kolonien eingesetzt werden oder aber mit Waffen bestückt in den Krieg ziehen.

## 2 Die Bearbeitung der Aufgabe

### 2.1 Vor Beginn des Praktikums vorbereitete Ergebnisse

#### 2.1.1 Aufgabenstellungen, die schon vor Beginn des Praktikums bearbeitet wurden

Um eine zügige Arbeit während des Praktikums zu gewährleisten, wurde schon ca. ein Semester vor Beginn des Praktikums mit den Vorbereitungen begonnen.

An erster Stelle stand hier eine Bestandsaufnahme. Als Grundlage für das Praktikum diente vorerst nur eine lose Sammlung von Ideen zum Thema „Onlinespiel, Weltall, Aufbausimulation“. Diese Ideen waren ca. zwar Jahre zuvor in einem Webforum gesammelt wurden, wobei das ursprüngliche Ziel, die Implementierung des Spiels, nie angegangen wurde.

Anschließend begann die Entwicklung eines brauchbaren Datenbankschemas. Parallel dazu wurden erste Spielregeln aufgestellt, die sich vor allem damit auseinandersetzten, was ein Spieler alles tun können soll. Das heißt, es wurde festgelegt, dass es Handelsbeziehungen zwischen den Spielern geben soll, dass Kriege möglich sein sollen, die bis zur Invasion eines fremden Planeten gehen.

Anfangs wurde an dem ERM vor allem mit Block und Stift gearbeitet. Als es jedoch Ausmaße annahm, die so nicht mehr zu erfassen waren, griff ich zu dem ersten Tool, das mir zum Thema ERM-Entwicklung einfiel: Case von Microtool. Während der Softwaretechnik – Vorlesung war dieses das erste Mal vorgestellt worden. Allerdings stand schon bald fest, dass dieses Tool für eine effektive Weiterentwicklung nicht in Frage kommen würde. Es ist vor allem zur Entwicklung von komplexen Softwaresystemen geeignet, aber weniger auf die Datenbankentwicklung spezialisiert. Außerdem konnte während der Entwicklung vor dem Praktikum lediglich eine Trialversion zum Einsatz kommen, die auch auf meinem eigenen Rechner lauffähig ist. Dort ist aber zum Beispiel die Anzahl der Tabellen zu stark eingeschränkt.

#### 2.1.2 Tools, die bei der Vorbereitung des Praktikums Verwendung fanden

- ERwin

Hierbei handelt es sich um ein Tool zur Erstellung von ERM. Das Tool wurde von Frau Rupf für die Entwicklung komplexer Datenbankmodelle empfohlen. Es zeichnet sich einerseits durch seine intuitive Bedienung aus, andererseits verfügt es über einen für die Aufgabenstellung mehr als umfangreichen Funktionsumfang auch in der Testversion.

Es war hiermit möglich, die Basis für das Spiel (ein möglichst geschwindigkeitsoptimiertes Datenbankmodell in erster und zweiter Normalform) zu entwickeln und im Prozess des Entwurfs der Spielsoftware zu verfeinern.



- Case5/0



Zur Analyse der Aufgaben des Software sowie zum Entwurf des grundlegenden Aubaues des Spieles wurde Case verwendet. Dabei kamen vor allem die Erfahrungen zum Tragen, die im Fach Softwaretechnik gemacht wurden. Bei der Entwicklung des Spiels sollte vor allem ein Fehler vermieden werden, der anscheinend bei anderen Projekten dieser Art häufig gemacht wird: Das Nachbessern und Erweitern des

Softwareproduktes im Nachhinein sollte zwar einfach möglich sein, allerdings so selten wie möglich nötig werden. Aufgrund der Größe der Aufgabenstellung und der eben genannten Zielstellung ergibt sich, dass eine Straight – forward – Lösung des Projektes nicht möglich war. Somit stützte sich die Entwicklung auf verschiedene Softwareentwicklungstools wie ERwin und das eben genannte Case.

- A3 – Block, Bleistift und diverse Buntstifte zu ersten Entwürfen von Gebäuden, Faktoren und des ersten ERM



## 2.2 Entwicklung während des Praktikums

### 2.2.1 Zu bearbeitende Aufgabenstellungen während des Praktikums

Mit Beginn des Praktikums konnte, gestützt auf die Ergebnisse der Vorbereitungsphase, die Erstellung der die Spieldaten beinhaltenden Datenbank begonnen werden.

Dabei wurde die Entwicklung aufgeteilt in verschiedenen Bereiche:

1. Verfeinerung des Datenbankmodells im Zuge der Entwicklung der Scripte, die das Userinterface generieren.
2. Entwicklung eines Auswertungsprogrammes, dass die von den Usern angeordneten Züge berechnet.
3. Test aller Komponenten



Abbildung 2-2: Timeline

### 2.2.2 Zeitplan bei der Entwicklung

Zeitraum	Aufgabenstellung
<b>11.03.2002</b>	Entwicklung eines ersten Interfaces
-	Darunter:
<b>05.04.2002</b>	<ul style="list-style-type: none"> <li>• 11.03.-05.04 Anmeldescript sowie erste Tabellen in der Datenbank</li> <li>• 13.03.-19.03 Script zum Anzeigen aller baubaren Gebäude sowie Script zum administrativen Einfügen neuer Gebäude ins Spiel, Debugging dieser</li> <li>• 20.03.-25.03 Scripte, die den Handel zwischen den Kolonien abwickeln</li> <li>• 26.03.-05.04 Scripte zur Kommunikation unter den Spielern, Scripte, die die Kriegsführung betreffen</li> </ul>
<b>08.04.2002</b>	Entwicklung des Auswertungsprogrammes
-	Darunter:
<b>08.06.2002</b>	<ul style="list-style-type: none"> <li>• 1. Woche: Bau von Gebäuden und Auswertung ihrer Einflüsse auf die Rohstoffe</li> <li>• 2., 3. und 4. Woche: Handel zwischen den Spielern</li> <li>• 5., 6. und 7. Woche: Entwicklung aller nötigen Angriffsfunktionen. Dabei kam es immer wieder zu grundlegenden Umstellungen des Kampfprinzips. Dies ist vor allem darin begründet, dass die Kampffunktionalität der einzige Bereich war, der von vornherein im Entwurf vernachlässigt worden war, dabei allerdings den größten Teil der Auswertung ausmacht.</li> <li>• 8. Woche: Einbringen diverser Verbesserungen in der Gebäudestruktur und Erstellung erster Gebäude, mit denen die Alphatester spielen können</li> <li>• 9. Woche: Verbesserungen im Interface, das jetzt darauf abgestimmt wurde, Bilder zur Darstellung der Gebäude sowie deren Beschreibung aus einer extra Tabelle in der Datenbank zu entnehmen. Somit wird eine Internationalisierung des Spieles vereinfacht.</li> </ul>

## 2.2.3 Einarbeitungszeiten

### 2.2.3.1 Die erste Entwicklungsphase



In der ersten Phase der Entwicklung wurden, wie oben schon erwähnt, mittels PHP Scripte geschrieben, die das Userinterface generieren und die Aktionen, die die Nutzer ausführen möchten, in der Datenbank hinterlegen.

Hier war vor allem eine Einarbeitung in das Sessionhandling von PHP4 nötig. Damit ist es einfach möglich, einen User, wenn er sich einmal eingeloggt hat, während seiner Aktionen immer wiederzuerkennen. Dabei wird einfach je eingeloggtem User ein Sessioncookie serverseitig erstellt, in dem verschiedene Eigenschaften des Nutzers gespeichert werden können (z.B. die Zeit, zu der er das letztmal ein Script aufgerufen hat oder seine Nutzer ID). Diese Informationen können dann innerhalb der Scripte einfach genutzt, verändert kurz: verarbeitet werden.

Weiterhin war eine kurze Einarbeitung in die MySQL-Funktionalität von PHP nötig. Nachdem der grundlegende Zugriff auf MySQL klar war, wurden mit einfachen Programmierprinzipien die Scripte portabel gemacht. Das heißt, es wurden Wrapperfunktionen um die eigentlichen mysql-Funktionen geschaffen. So wurde zum Beispiel die Fehlerbehandlung innerhalb der Scripte, die die Interfaces generieren, deutlich verkürzt und vereinfacht. Außerdem wurde so die Verbindung zum Datenbankserver, die Usernamen, Paßwort und URL der Datenbank benötigt, auf eine allgemeine Funktion beschränkt, die aus allen anderen Interfacescripten aufgerufen werden kann.

Bei der Einarbeitung hilfreich war vor allem das Buch „Grundlagen und Profiwissen – PHP4“ aus dem Hanser-Verlag.

### 2.2.3.2 Die zweite Entwicklungsphase

In der zweiten Entwicklungsphase ging es an die Bearbeitung des Auswertungsprogramms. Die einzige entscheidende Wahl, vor der ich zu Beginn der Entwicklung dieses Moduls gestellt wurde, bestand darin, eine passende Bibliothek für den Zugriff auf die Datenbank auszuwählen.

Dabei standen eine objektorientierter Ansatz einem strukturiertem (von MySQL mitgeliefert) gegenüber. Die Entscheidung fiel letztendlich auf die Bibliothek, die von den Datenbankentwicklern von MySQL mitgeliefert wurde. Gründe dafür waren zum Beispiel, dass der Funktionsumfang dieser Bibliothek den Ansprüchen genügte. Dabei war weniger die Performance entscheidend, die dieses Funktionen sicherlich auch auszeichnet, sondern vielmehr eine gewisse Sicherheit bei der Ausführung der einzelnen Abfragen. Ziel ist schließlich, dass dieses Programm letztendlich unbeaufsichtigt jeden Tag um eine bestimmte Uhrzeit startet und dann möglichst stabil seine Aufgaben beenden muß.

### 2.2.3.3 Die dritte Entwicklungsphase – die Testphase

Schon während der Entwicklung wurden natürlich sämtliche Scripte und Auswertungsmodule auf Syntax- und Semantikfehler hin untersucht. Dabei wurde versucht, beim Test die Vorgehensweise des V-Modells einzuhalten.

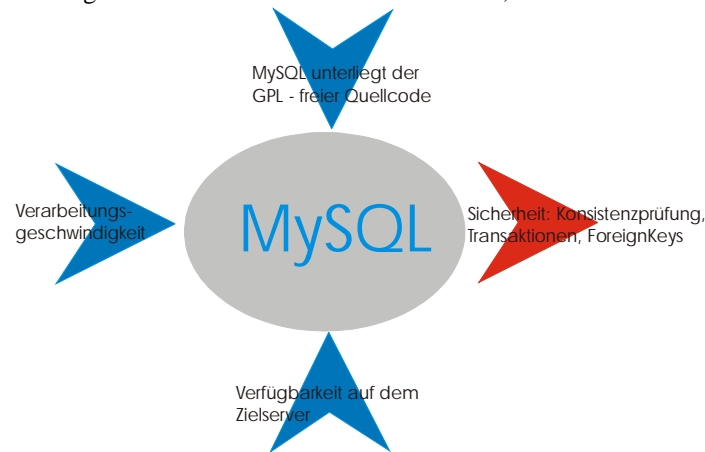
Für die erste Entwicklungsphase bedeutet dies, dass nach jedem fertiggestellten Script intensive Tests folgten, die sicherstellen sollen, dass in diesem keine Fehler mehr enthalten sind. In der zweiten Entwicklungsphase wurden dann mithilfe der bereits fertiggestellten Scripte Testdaten erzeugt und jeweils ein Modul auf Richtigkeit überprüft. Dabei konnte lediglich das Modul, dass die Angriffe berechnet nicht vollständig getestet werden. Deshalb schloß sich hier eine dritte Phase an.

Nachdem die Entwicklung der PHP-Scripte und des Auswertungsprogrammes als abgeschlossen definiert wurden, erfolgte die Suche nach geeigneten Alphatestern, die das gesamte Projekt nochmals auf seine korrekte Funktion überprüfen sollten. Eine zweite Aufgabe, die ihnen zukam, war der Test auf die Sicherheit des Projekts. Das heißt, es sollte nahezu unmöglich werden, falsche Daten in der Datenbank abzulegen oder im Spiel Informationen abzurufen, auf die der Spieler eigentlich keinen Zugriff hat. Die dritte Aufgabe bestand letztendlich in der Überprüfung der Spielregeln und Verbesserung z.B. der Gebäudedefinitionen.

## 2.2.4 Die verwendete Software

### 2.2.4.1 Warum MySQL als Serverdatenbank?

Beginnt man mit der Entwicklung einer Anwendung, bei der eine Datenbank sich um die Speicherung und Verwaltung der zu erwartenden Daten kümmert, stehen dem Entwickler viele Wege offen. Er kann sich für ein professionelles System wie Oracle entscheiden. Hier bekommt er zwar allen Komfort, den man sich nur wünschen kann, dies aber unter Umständen zu finanziellen Unkosten, die den Vorteilen des Systems in keiner Weise in einem akzeptablen Verhältnis gegenüberstehen.



Auf der anderen Seite stehen Tools wie PostgreSQL oder MySQL, die zwar frei verfügbar und erweiterbar sind, aber dafür im Vergleich zu kommerziellen Tools nur mit einem begrenzten Funktionsumfang aufwarten können.

Im Falle von Unigen war einerseits entscheidend, dass das DBMS kostengünstig ist. Auf der anderen Seite aber auch möglichst offen, um die Entwicklung der Scripte und

**Abbildung 2-3: Warum MySQL?**

des Auswertungsprogrammes zu erleichtern. Dies wurde erkaufte zum Preis der Sicherheit, die kommerzielle Lösungen bieten. MySQL ist als eine sehr schnelle Datenbank zu bezeichnen, die aber dafür auf diverse Sicherheitsmechanismen oder komfortabel Selects verzichtet.

So unterstützt zum Beispiel die für Unigen verwendete Version von MySQL noch keine Subselects. Dafür ist sie als verhältnismäßig stabil anzusehen, da sie schon seit geraumer Zeit auf diversen Webservern läuft.

Der letztendlich ausschlaggebende Grund ist aber auch hier, welche Datenbank auf dem Zielsystem verfügbar ist – im Falle Unigen handelte es sich dabei um MySQL.

### 2.2.4.2 PHP als Scriptsprache? Warum nicht Perl?

PHP ist noch relativ neu und von daher auch häufiger in den Schlagzeilen bezüglich gefundener Sicherheitslücken. Außerdem ist diese Sprache nicht als besonders schnell einzustufen – schon allein darin begründet, dass sie lediglich interpretiert wird.

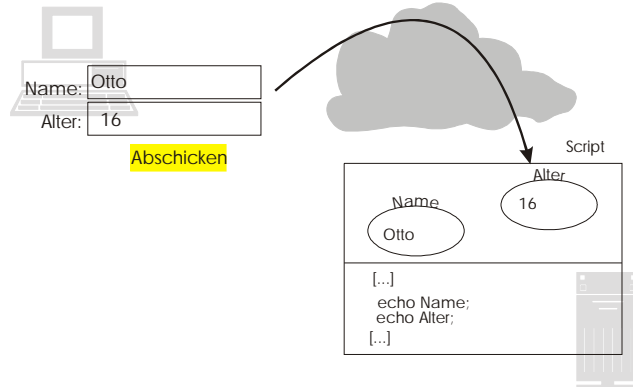
Dennoch fiel für den Aufbau der Interfaces die Wahl auf PHP. Der Grund liegt einerseits in der Einfachheit der Sprache.

PHP unterstützt den Web – Programmierer schon in vielerlei Hinsicht. So werden die vom User eingegebenen Daten, die der Browser übermittelt, wenn ein Formular abgeschickt wird, PHPseitig bereits auf Variablen verteilt. Diese können dann einfach im das Formular verarbeitenden Script einfach genutzt werden. In Perl muß sich der Programmierer selbst darum kümmern, die vom Browser übermittelten Daten auszuwerten, zu parsen und auf Variablen zu verteilen.

Andererseits ist PHP als Programmiersprache zum Generieren von Webseiten konzipiert worden:

Alles, was an einer HTML - Seite dynamisch sein soll, wird einfach in den bestehenden HTML – Code mit einfachen PHP-Anweisungen eingefügt. Diese Anweisungen müssen lediglich für den Parser auf dem Webserver gekennzeichnet werden. So können zum Beispiel Tabellengerüste einfach mit täglich veränderten Inhalt gefüllt werden.

Im Gegensatz dazu muß in Perl immernoch die gesamte Webseite generiert werden. Das heißt, auch einfachste, statische HTML-Tags müssen mittels des Befehls print an den Client geschickt werden – diese Stufe entfällt bei PHP.



**Abbildung 2-4: Auswertung von Formulardaten mit PHP**

### 2.2.4.3 Gründe für die Verwendung von C für die Auswertungsfunktion

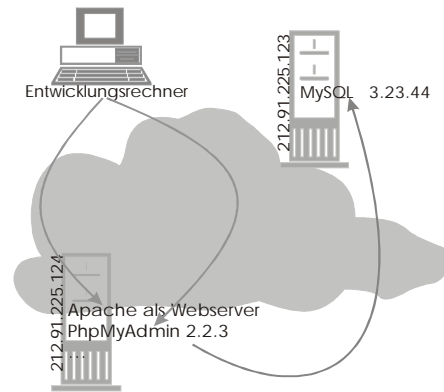
Die Auswertungsfunktion soll mit keinem User kommunizieren können, dafür aber sicher und verhältnismäßig schnell laufen. Da C nicht wie PHP oder Perl interpretiert wird, ist der Geschwindigkeitsvorteil diesen beiden Sprachen gegenüber gesichert. Weiterhin ist es mit C sehr einfach, passende, fehlerbehandelte Routinen zu schreiben.

### 2.2.4.4 Entwicklungsumgebung auf dem zur Verfügung gestellten Serversystem

Auf den Server, auf dem Unigen letztendlich laufen sollte, war lediglich der Zugriff via ftp möglich. Auch auf die zum Einsatz kommende Datenbank konnte nicht direkt zugegriffen werden. Dies ist aus Sicherheitsgründen sehr sinnvoll, bedeutet aber für den Entwickler einen gewissen Mehraufwand. So müssen z.B. die Scripte lokal auf dem Entwicklungsrechner editiert, auf den Server kopiert und dann dort getestet werden.

Um wenigstens den Zugriff auf die Datenbank zu ermöglichen, wurde die Umgebung phpMyAdmin auf dem Webserver installiert, über die die Inhalte und die Struktur der Datenbank komfortabel verändert werden können. Doch schon bald stellte sich das als recht unpraktikabel und langsam heraus, so dass vor allem bei der Programmierung des Auswertungsprogramms auf eine lokale Lösung ausgewichen wurde.

Das heißt, die bereits auf dem Zielsystem entwickelte Datenbank wurde in das DBMS MySQL unter Linux eingefügt und mit Daten gefüllt. Anschließend konnte das Auswertungsprogramm lokal entwickelt und getestet werden. Im Zuge dieser Änderung wurden auch die schon fertigen PHP-Scripte so umgeschrieben, dass sie zwar einerseits auf der lokalen Umgebung funktionieren, andererseits aber so portabel sind, dass es genügt, wenige Zeilen zu editieren, um sie für den Einsatz auf dem Zielsystem vorzubereiten. Nach erfolgreichem Abschluß aller Alphatests werden Scripte und Auswertung dann auf dem Zielsystem integriert.



**Abbildung 2-5: Für Unigen zum Einsatz kommendes Serversystem**

### 2.2.5 Bei der Entwicklung verwendete Hardware

- Als Entwicklungssystem konnte der Laptop des Praktikanten verwendet werden, wodurch eine gewohnte Arbeitsumgebung gesichert war.
- Als Hostsystem für das Spiel diente ein Cobalt RaQ2.0 (Cobalt OS 4.0) mit 128 MB Ram.

### 2.2.6 Aussagen zu Qualitätsvorschriften

Qualitätsvorschriften spielten sowohl in der allgemeinen Arbeit von Network Nation als auch bei der Bearbeitung der Praktikumsaufgabe keine Rolle. Dies liegt schon in der Größe und dem eigentlichen Aufgabenbereich der Firma begründet. Bei Network Nation handelt es sich um ein Familienunternehmen, das von zwei Brüdern geführt wird. Die alltäglichen Aufgaben bestehen aus dem Webhosting, der Planung, Installation und Administration lokaler Netzwerke von Unternehmen.

Daher ist die Realisierung der Praktikumsaufgabe als ein vom allgemeinen Geschehen bei Network Nation losgelöstes Projekt anzusehen. Die Firma hatte sich lediglich bereit erklärt, das Ergebnis auch über die Praktikumszeit hinaus kostenfrei zu hosten und bei Bedarf bei der Entwicklung zu helfen.



## 2.3 Entwicklung nach Abschluß des Praktikums

Nach Abschluß des Praktikums wird die Entwicklung von Unigen wenn möglich fortgesetzt. Derzeit ist z.B. lediglich eine Rasse verwendbar – die der Menschen. Für alle anderen wurden noch keine Gebäude, Waffen oder Raumgleiter definiert. Das Ziel nach dem Praktikum wird also sein, die Umgebung auch der anderen drei Rassen auszuarbeiten. Dabei können alle Spieler, die sich als Alphatester angemeldet haben, natürlich mitwirken und ihre Ideen einbringen.

Außerdem sollen für alle Bauwerke Grafiken erstellt und die Ergonomie der Interfaces verbessert werden. Als eine dritte Aufgabe kann die Erweiterung des Funktionsumfanges um die Invasion fremder Planeten angesehen werden, die aus Zeitgründen weder in den Scripten noch in dem Auswertungsprogramm implementiert wurde.

Der weitere Entwicklungsverlauf von Unigen wird aber einmal von der Initiative der angeworbenen Tester abhängen, auf der anderen Seite von der Zeit, die mir für das Spiel bleibt. Das endgültige Ziel wird aber bleiben, Unigen als vollwertiges Onlinespiel, lizenziert unter der GPL im Internet unter Umständen gegen Bezahlung anbieten zu können.

## 3 Kurze Aussagen zur Struktur des Spiels selbst

### 3.1 Vertikale Gliederung der Entwicklung in zwei Teilbereiche

Bei dem Spiel Unigen handelt es sich um ein rundenbasiertes Aufbaustrategiespiel, wie in der Einführung schon angedeutet. Das heißt, im Verlauf eines Tages stellen die Nutzer ihre Spielzüge ein. Dazu soll ein einfacher Browser, mit dem man die Webseite von Unigen anwählt und sich über einen Login als User identifiziert genügen.

Nach Ablauf eines Tages soll eine Auswertung die Spielzüge letztendlich berechnen und ausführen.

Diese Trennung sollte laut den ersten Entwürfen auch in den letztendlich arbeitenden Programmteilen sichtbar werden. Eine Hälfte ist für das Interface, mit dem der Nutzer spielt, verantwortlich. Diese Seite wurde in PHP geschrieben und war möglichst portabel zu halten.

Die zweite Seite ist ein Programm, das täglich einmal die Spielerdaten und Züge auswertet – da während der Auswertung der Server für Zugriffe von außen gesperrt wird, war die Zeit der Auswertung kurz zu halten. Daher wurde hier auf die Verwendung einer Scriptsprache verzichtet und stattdessen C eingesetzt.

Bei beiden Teilen erfolgt der Zugriff auf die Datenbank, in der sowohl die Spielzüge gehalten werden, als auch die Userdaten und der aktuelle Spielstand. Dazu jedoch mehr im Kapitel „ERM – das Rückgrat von Unigen“.

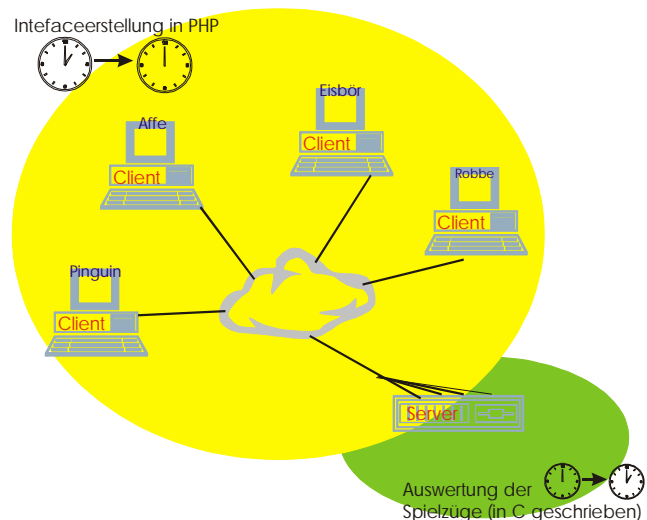


Abbildung 3-1: Vertikale Zweiteilung des Spiels

### 3.2 Horizontale Gliederung des Spieles in Spielzugarten

Die Spielzüge selbst gliedern sich grob in 4 Teilbereiche, die sich in den Modulen der Auswertung, aber auch in der Aufteilung der Aufgaben auf die Scripte widerspiegelt:

- 1) Bau von Gebäuden (Rohstoffproduzierer, Technologien, Kommunikationstationen)
- 2) Handel mit Mitspielern und Veranstaltung von programmgesteuerten Auktionen
- 3) Aufbau einer Flotte. Die Flotte ermöglicht den Transport von Gütern mit Raumgleitern. Somit ist auch ein Handel außerhalb von Auktionen einfach möglich. Aber auch die Kriegsführung und Eroberung von feindlichen Planeten setzt eine starke Flotte mit entsprechenden Waffen voraus.
- 4) Kommunikation mit Mitspielern, deren Voraussetzung der Bau von Kommunikationsstützpunkten (entweder aktiv oder passiv) ist.

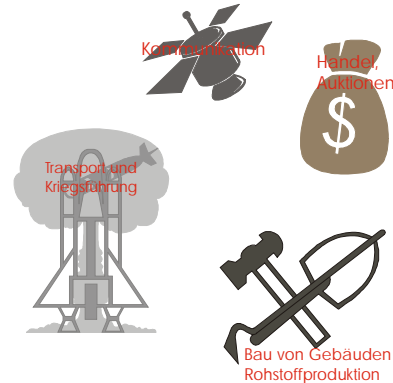


Abbildung 3-2: Horizontale Gliederung

Die Aktionen eines Spielers können in diese drei grundlegenden Gruppen untergliedert werden. Entsprechend können auch die Module der Auswertung eingeteilt werden. Weiterhin erleichtert diese Unterteilung das Verständnis des ERM.

### 3.3 ERM – das Rückgrat von Unigen

Ein genaues ERM befindet sich im Anhang zu diesem Bericht. Dieses zu verstehen ist aber auf den ersten Blick etwas schwierig. Allerdings läßt sich anhand des ERM sehr gut die Funktionsweise und der strukturelle Aufbau von Unigen rekonstruiert werden. Deshalb sollen hier noch einige Erklärungen erfolgen.

#### 3.3.1 Grundprinzipien bei der Erstellung des ERM

- Das Modell soll möglichst alle Daten des Spieles beinhalten, sodaß weder bei der Ausführung der Scripte noch bei der Entwicklung der Auswertung Ausnahmefälle entstehen, die die Wartung des Spieles erschweren.
- Das Modell soll sowohl die vertikale als auch die horizontale Gliederung des Spieles widerspiegeln. Hinzu kommt eine Gliederung in Daten, die durch Spieler und Spielzüge erzeugt werden und Daten, die das Spiel selbst ausmachen (z.B. gewisse Definitionen von Gebäudetypen).

#### 3.3.2 Zuordnung der Tabellen zu den einzelnen Kategorien:

Eine erste grobe Einteilung der Tabellen kann erfolgen, indem man zwischen Daten unterscheidet, die von Spielern erzeugt werden (z.B. die Beauftragung eines neuen Gebäudes) und Daten, die der reinen Spieldefinition dienen. Dies spiegelt sich auch in der Namensgebung der Tabellen in der Datenbank wieder: Alle Tabellen, die Daten enthalten, die von Nutzern generiert werden beginnen mit dem Vorsatz „chars\_“.

Eine weitere grobe Einteilung der Tabellen kann nach der vertikalen Gliederung des Spieles in auswertungsbe-  
troffene und auswertungsunabhängige Daten erfolgen. In der Grafik rechts wurden dabei vorerst nur die Tabellenkomplexe beachtet, die Daten enthalten, die von Nutzern verändert werden können. Es werden also in der Grafik alle die Tabellen außen vor gelassen, die z.B. Gebäudedefinitionen beinhalten.

Die vertikale Gliederung des Spiels in Auswertung und Interfacegenerierung spiegelt sich hier insofern wieder, als bestimmte Daten, die von Nutzern erzeugt werden, von der Auswertung nicht verändert werden. So spielen in der Auswertungsphase zum Beispiel alle Tabellen, die der Nutzerkommunikation im Spiel dienen, keine Rolle. Weiterhin spielen natürlich während der Auswertung Nutzerdaten, wie Adresse, Alter oder Nickname keine wesentliche Rolle.

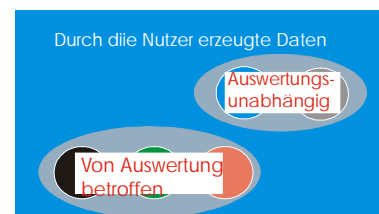
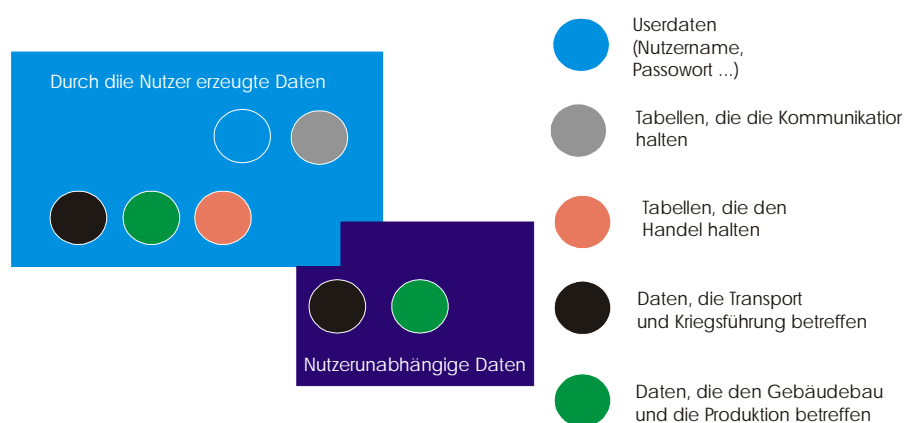


Abbildung 3-3: Vertikale Gliederung der Datenbank

Weiterhin lassen sich die auswertungs(-un-)abhängigen Daten jeweils der horizontalen Gliederung in verschiedene grundsätzliche Spielzugarten des Spieles unterwerfen. (siehe dazu auch die Abbildung unten).

Auch die horizontale Gliederung spiegelt sich in der Namensgebung im ERM wieder. Zum Beispiel enthalten alle Tabellen, die mit dem Bau und der Definition von Gebäuden beschäftigen, die Silbe „\_build\_“.

Somit spiegelt auch der Datenbankaufbau die horizontale und vertikale Gliederung wieder. Hinzu kommt hier die am Anfang des Kapitels erwähnte Einteilung in Daten, die Nutzer verändern und solche, die meist bloße Definitionen enthalten und vom Nutzerinterface nur genutzt, nicht aber verändert werden können.



**Abbildung 3-4: Gliederung in Nutzerabhängige und -unabhängige Daten, sowie horizontale Gliederung der Tabellen**

## 4 Anhang

### 4.1 Das ERM von Unigen