

Thema:
**Konzeption und Umsetzung eines ortsbezogenen
Gamification-Ansatzes
für regionale Dienstleister**

Masterarbeit

**im Studiengang Wirtschaftsinformatik
der Fakultät Wirtschaftsinformatik
und Angewandte Informatik
der Otto-Friedrich-Universität Bamberg**

Verfasser:

Denis Hamann Matr.-Nr. 1684873

Gutachter:

Prof. Dr. Schlieder

Danksagung

An erster Stelle möchte ich meinen Eltern für die langjährige Unterstützung in meinem Studium danken.

Darüber hinaus möchte ich mich vor allem für die anregenden Gespräche, konstruktive Kritik, wertvolle Hinweise bei meinen Betreuern Dominik und Klaus bedanken. Diese haben nicht nur mein Interesse für die GeoGame Thematik geweckt sondern auch einen Blick über die Thematik hinaus ermöglicht.

An dieser Stelle möchte ich mich auch bei Olga bedanken, für die Möglichkeit die ESRI EMEAUC besuchen zu können, die es mir nicht nur ermöglicht hat einen Einblick in die kommerzielle Verwendung von GIS zu erhalten, sondern auch über die aktuellen Themen und Entwicklungen.

Ein weiterer Dank gilt an dieser Stelle der Ruby User Group Bamberg bzw. Govinda, für die diverse Workshops und angeregten Ruby on Rails Abenden. Zum Schluss möchte ich mich noch bei all denjenigen bedanken, die diese Arbeit Korrektur gelesen haben und auch auf deren inhaltliche Aspekte eingegangen sind.

Abstract

Thema Text

Inhaltsverzeichnis

Abbildungsverzeichnis	VI
Tabellenverzeichnis	VII
1. Einleitung: Ortsbezogene Gamification	1
2. Problemstellung	3
3. Forschungsstand	5
3.1. Gamification	5
3.2. Geogames	9
3.3. Relokalisierungsansätze	11
3.4. Verwendung offener Geodaten	15
3.5. Bewertung von Spielfeldern	17
4. Lösungsansatz	18
4.1. Mögliche Lösungen	18
4.2. Gewählter Lösungsansatz	29
5. Umsetzung	34
5.1. Erläuterung des Softwaretechnischen Entwurfs	34
5.2. Bewertung der Technologien und Werkzeuge	47
5.3. Implementierung des Geogameframeworks	51
6. Evaluierung	61
6.1. Qualität der Lösung	61
6.2. Qualität der Spielfelder	64
7. Diskussion	65
7.1. Einordnung der Ergebnisse	65
7.2. Relokalisierbarkeit geobasierter Gamification-Ansätze	65
7.3. Einfache Wiederverwertbarkeit durch Relokalisierung	65
7.4. Ausblick	65

A. Anhang	66
A.1. Section	66
B. Glossar	67
Literatur	68

Abbildungsverzeichnis

3.1. Gamification nach Deterding et al. [Det+11]	7
3.2. Flow Zustand nach nach Csikszentmihalyi [Csi91]	9
3.3. Spielfeld Verteilung nach Kiefer et al. [KMS07a]	14
3.4. OSM Elemente	16
4.1. Spielertypen nach Bartle [Bar04]	19
4.2. Globale Mobile Betriebssysteme in Anlehnung an Gartner [Gar13] - Bild: [Wik14]	26
5.1. Prozess: Von OSM zum Spielelement	34
5.2. Transformationsprozess: Relations, Ways, Nodes	35
5.3. OSM ID: Bitmaskenkodierung im 64bit long Wert	37
5.4. Visualisierte Schnittstellen des Frameworks	39
5.5. Spielfeld Mockup	41
5.6. Usecase Diagramm	43
5.7. Vereinfachtes Klassen Diagramm	44
5.8. Interaktives Debugging mit 'better_errors' und 'binding_of_caller'	50
5.9. Grundansicht - Spielfeld	54
6.1. Antwortzeit der GameAPI für den Tag public_transport=stop_area	63

Tabellenverzeichnis

3.1. Literaturübersicht zur Definition von Gamification 6

1. Einleitung: Ortsbezogene Gamification

Im Zuge der Digitalisierung steht der regionale Einzelhandel vor der Herausforderung für seine Kunden weiterhin interessant zu sein, bei gleichzeitig steigender Konkurrenz durch das Internet. Aktuelle Zahlen des Statistischen Bundesamtes [Nac12] und der GfK belegen einen stagnierenden bzw. teilweise einen rückläufigen Markt. Hierbei stellt sich die Frage in welcher Art und Weise die regionalen Händler bestehende Kunden binden können aber auch neue Kunden mit deren Angebot vertraut gemacht werden, welche dieses noch nicht im Detail kennen. Klassische Marketing Ansätze sind verbreitet und werden entsprechend genutzt. Ein Ansatz der in den letzten Jahren immer mehr an Bedeutung gewonnen hat, stellt die Gamification dar. Durch diese wird versucht eine extrinsische Motivation für bestimmte Handlungen zu erzeugen. In diesem Zusammenhang soll durch Gamification eine Kundenbindung und -neugewinnung erzielt werden.

Um ein interessantes Spielkonzept dem Kunden bieten zu können wird hierbei auf Geogames zurück gegriffen. Ein solches nutzt die physikalische Fortbewegung der Spieler als Interaktion mit dem Spiel. Durch diesen Modus kann erreicht werden, dass die Spieler geografisch mit entsprechenden Orten interagieren und im konkreten Fall regionale Anbieter aufsuchen.

Der Hauptaugenmerk dieser Arbeit soll sich vor allem auf die konkrete Erstellung der Spielfelder eines Geogames beziehen. Hierzu soll zunächst untersucht werden, inwiefern bestehende öffentliche Datenbanken wie z.B. Openstreetmaps (OSM) genutzt werden können. Auf der anderen Seite müssen diese Daten entsprechend aufbereitet werden und dem Spiel zur Verfügung gestellt werden.

Daher fokussiert sich die Fragestellung der Arbeit wie folgt:

Wie können freie Geobasisdaten zur Konfiguration von Spielfeldern genutzt werden?

Es wird untersucht, in welcher Art und Weise die Daten vorliegen und entsprechend transformiert werden müssen. Darüber hinaus muss untersucht werden, welche Spielfelder besser für eine Verwendung geeignet sind und welche schlechter. Ziel ist es

einem Spiel Designer, den Aufwand für das Staging eines Spiels zu reduzieren und gleichzeitig entsprechende lokale Gewerbe in das Spiel einbinden zu können. Dadurch können die Spieler wiederum auf Geschäfte oder andere Dienstleister aufmerksam gemacht werden.

Die konkrete Umsetzung soll anhand eines Beispielspiels erfolgen. Hierfür wird ein entsprechender Entwurf erstellt, der dies ermöglichen soll.

2. Problemstellung

Möglichkeiten ortsbezogener Gamification

Ziel der Thesis ist es, durch ortsbezogene Gamification neue oder bestehende Kunden an die lokalen Dienstleister zu binden. Hierzu muss untersucht werden, welche Möglichkeiten und Vorgehensweisen zu empfehlen sind. Hierbei sollen die Kunden durch extrinsische Motivation zu bestimmten Interaktionen mit den einzelnen Geschäften animieren werden. Durch die zusätzlich Interaktion soll entsprechend potentiell mehr Umsatz gemacht werden.

Location based Games als Mittel für Gamification

Im Zug der Arbeit gilt es zu überprüfen wie Location based Games am besten eingesetzt werden können und wie diese gestaltet werden müssen um den Ansprüchen der Gamification zu genügen. Hierbei sollen gleichzeitig die erwünschten Effekte (konkret mehr Umsatz für den Händler) eintreten. Um diesen gerecht zu werden muss das Design des Spieles bezüglich Content aber auch Spielmechanik entsprechend ausgerichtet werden.

Anforderungen an ein Geogameframework

Die Anforderung an ein entsprechendes Framework muss untersucht werden. Es muss festgestellt werden, wie einem Entwickler bzw. einem Spielveranstalter möglichst viel Arbeit abgenommen werden kann und gleichzeitig trotzdem entsprechende Konfigurationsmöglichkeiten zur Verfügung gestellt werden.

Relokalisierbarkeit von ortsbezogenen Spielen

Ein wichtiger Aspekt bei ortsbezogenen Spielen stellt der Game Content bzw. der Geogamecontent dar. Hierbei handelt es sich um georeferenzierte Spielelemente mit denen der Spiele über das Spielfeld interagiert. Die Herausforderung ist es, dass ein Spielfeld nicht nur in einem fest definierten Bereich, z.B. in einer speziellen Stadt gespielt werden kann, sondern auch an anderen Orten. Hierfür gibt es unterschiedliche Ansätze die es zu beleuchten gibt und zu entscheiden ist, welcher dieser am sinnvollsten ist.

Freie Geobasisdaten und Möglichkeiten der kommerziellen Nutzung

Es gibt viele kommerzielle und nicht kommerzielle Geodaten die heutzutage bezogen werden können. Es gilt zu untersuchen, welche ob öffentliche bzw. freie Geodaten heutzutage verwenden werden können und wie diese in Verbindung mit einem kommerziellen Projekt genutzt werden können.

3. Forschungsstand

3.1. Gamification

Der Begriff Gamification geht auf Nick Pelling 2002 zurück. Er beschreibt den Prozess bei dem Spielmechaniken auf bestehende Aspekte angewendet werden um eine extrinsische Motivation zu erzeugen.[Mar13] Erste Gamification Ansätze gab es bereits zu Beginn des 20. Jahrhunderts z.B. durch Stempelkarten an der Eisdiele. Später wurden ähnliche Konzepte in Vielfliegerprogrammen aufgegriffen.

In der Literatur gibt es unterschiedliche Definitionen der Gamification. In der nachfolgenden Tabelle 3.1 sind verschiedene Autoren so wie deren Einordnung des Gamification Begriffs zu sehen. Es lässt sich zunächst feststellen, dass ein gemeinsamer Konsens in der Literatur darüber herrscht, dass Gamification eine Nutzung von Spielmechaniken darstellt. Vergleicht man jedoch die einzelnen, so lässt sich feststellen, dass [ZC11] und [Kap12] eine Übereinstimmung bezüglich der Nutzung von Gamification als Motivation finden, sowie als Mittel zur Lösung von Problemen. [ZC11] definiert hier die Gamification als Mittel um extrinsische Motivation zu erzeugen, welche entsprechend Einfluss auf die Handlungen des Einzelnen hat. [Det+11], [Mar11] und [Oxf13] grenzen dazu im Vergleich die Gamification von normalen Spielen explizit ab. Diese legen Wert darauf, dass explizit keine Spiele selbst als Gamification verstanden werden, sondern als Basis des Gamification immer eine normale Tätigkeit steht. [Kap12] geht im Vergleich zu den restlichen Autoren hier weiter und ergänzt die Nutzung von Gamification als Lehrmittel, aber stellt diese auch als Motivator für Personen dar. Speziell die Nutzung von Gamification im Zusammenhang der Lehre lässt sich in der aktuellen Literatur ebenfalls verfolgen (Quelle!). Gamification kann auch genutzt werden um ein Empowerment bei den partizipierenden Spieler zu erzeugen [Jea03]. Ein Beispiel für die Nutzung von Gamification stellt das Sammeln von Geinformationen mithilfe einer App dar. [Dra13] Eine Einordnung und Abgrenzung der Terminologie ist in 3.1 zu sehen.

Nutzung von Spielmechanik	Zichermann/Cunningham [ZC11]	Deterding et al. [Det+11]	Markus [Mar11]	Oxford [Oxf13]	Kapp [Kap12]
Motivation	X	X	X	X	X
Problemlösung	X				X
Spielferner Kontext		X	X	X	
Verhaltensbeeinflussung					
Lernförderung			X		X
Anregung zum Handeln					X

Tabelle 3.1.: Literaturübersicht zur Definition von Gamification



Abbildung 3.1.: Gamification nach Deterding et al. [Det+11]

In der Literatur werden immer wieder die Elemente Points, Badges und Leaderboards (PBL) angesprochen. Diese dienen als Mittel um eine Gamification durchführen zu können. Points stellen Punkte dar, die verwendet werden um einen Fortschritt des einzelnen Spielers darzustellen. Dies sind zum Beispiel Meilen in Vielfliegerprogrammen oder Statuspunkte bei Bahn Bonus. Bei Badges handelt es sich um Abzeichen, welche für bestimmte Erungenschaften an den Spieler vergeben werden. Ein Beispiel hierfür ist das Trainspotter Badget bei Foursquare, welches ausgestellt wird, wenn der Spieler in eine gewisse Anzahl von Bahnhöfen eingchecked hat. Die Badges sollen einen gewissen Status gegenüber den restlichen Spielern suggestieren. Leaderboards sind klassische Ranglisten. Diese dienen dazu einen Wettbewerb unter den Spielern zu erzeugen. Hierbei wird jedoch empfohlen nicht auf die klassische Top10 Liste, wie bei vielen Spielhallen Automaten zurückzugreifen. Stattdessen soll der Spieler zwischen anderen platziert werden, vorzugsweise sind die Spieler über und unter dem aktueller Spieler dessen Freunde (vgl. Foursquare). Dies verhindert, dass der Spieler von überhöhten Punktzahlen abgeschreckt wird.

[ZC11] erweitern das Modell in dem es um weitere Aspekte ergänzen und mehr

Struktur geben. Sie pflegen den Begriff SAPS. Dieser unterteilt sich in Status, Access, Power und Stuff (SAPS). Das bekannte PBL der Literatur wird unter Status zusammengefasst wie in nachfolgender Aufzählung zu sehen.

- Status (Badges, Levels, Leaderboards)
- Access (early Access)
- Power (give power, e.g. modicum control over other players)
- Stuff (give a reward, try to prevent that the price gets known)

Bei Access handelt es sich um SZugriffßu exklusiven Dingen, welche man dem Spieler gewährt. Ein Beispiel hier für ist die Lufthansa Senator Lounge oder die DB Lounge. Es kann sich aber auch um einen zeitlich verfrühten Zugriff auf ein Produkt oder Funktionen handeln.

Unter Power sind Mechaniken zu verstehen, welche es dem Spieler erlauben Einfluss - Macht - auf andere Spieler aus zu üben. Dies kann z.B. durch Moderationsrechte ab einem bestimmten Level realisiert werden. Foursquare realisiert dies durch Superuser.

Der letzte Punkt ist Stuff. Hierbei handelt es sich um Belohnungen die dem Spieler zuteilwerden. Klassischerweise handelte es sich hierbei z.B. um ein zusätzlich kostenloses Eis. Ziel ist es dem Spieler möglichst nicht einen konkreten monetären Gegenwert sehen zu lassen. D.h. dem Spieler soll es nicht ersichtlich sein wie viel der Reward wert ist. Ziel sollte es daher auch nicht sein einfach etwas kostenlos dem Spieler zu geben, sondern viel mehr etwas, was wiederum seinen Status unterstreicht.

Im Zuge der Gamification wird gerne der Begriff des *Flow*-Zustandes aufgegriffen. Hierbei handelt es sich um einen von [Csi91] eingeführten Begriff, bei dem es darum geht den Spieler zwischen einem optimalen Zustand zwischen Anspannung und Langeweile zu halten. In dem Flow Modell wird angenommen, dass der Mensch sich in einer Situation jeweils seiner Handlungsmöglichkeiten als auch seiner Fähigkeiten bewusst ist. Übersteigt der Umfang der Aufgaben die Fähigkeiten, so stellt sich der Zustand oberhalb des flow Zustande ein, wie in Abbildung 3.3 zu sehen. D.h. Sorge bzw. Sorge. Bei einer Unterforderung oder Einschränkung der Handlungsmöglichkeiten stellt sich schnell Langeweile ein. Das Ziel ist es daher immer den optimalen Zustand für den Spieler zu finden. Viele Spiele arbeiten unter anderem mit dynamischen Schwierigkeitsstufen (vgl. Gummi Band KI/Mario Kart).

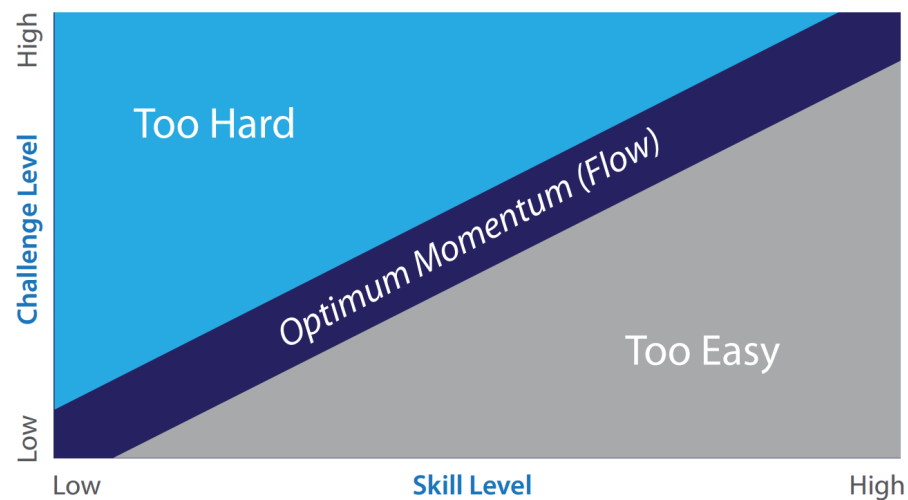


Abbildung 3.2.: Flow Zustand nach nach Csikszentmihalyi [Csi91]

3.2. Geogames

Spiel

Um den Geogames Begriff zu klären muss zunächst abgegrenzt werden, was unter dem Begriff Spiele zu verstehen ist. In der Literatur gibt es hierfür viele Definitionen. In diesem Kontext soll die Definition analog zu [SZ10] verwendet werden, welche ein Spiel als Abgrenzung zum normalen Alltag darstellt. Hierbei wird ein Spiel innerhalb eines sogenannten Magic Circles durchgeführt. Der magische Kreis dient als Regelraum in welchem das Spiel selbst durchgeführt wird.

Mobilegames

Unter Mobilegames sind Spiele aller Art zu verstehen, die unterwegs gespielt werden. Diese werden auf mobilen Endgeräten gespielt. [Bel+06] Unter Mobile Endgeräte fallen klassische Handheld-Konsolen wie z.B. Nintendo Game Boy, aber auch Smartphones.

Location based Games

Geogames sind Spiele welche in einem Geokontext gespielt werden. Hierbei wird die aktuelle Position des Spielers als Kontrollelement verwendet [SKM06]. Durch dieses kann der Spieler mit den Spiel interagieren. Geogames selbst sind nicht begrenzt auf digitale Spiele, sondern haben Ihren Ursprung in Treasure-Hunt Games. Ein späterer Nachfolger stellt z.B. das Geocaching dar. Ziel der Geogames ist die Interaktion des

Spielers mit der Umgebung. Dies unterscheidet sich von den klassischen Konsolen Spielen, bei denen der Spieler die Steuerung über einen Controller welchen er per Hand steuert, bedient. Grenzt man diese motorische Steuerung ab, so gibt es die Zwischenstufe des Vistaspaces. Im Vistaspace steuert der Spieler das Spiel nicht mehr mit seinen Händen, sondern mit motorischen Bewegungen. Beispiele hierfür sind die Nintendo Wii und die Xbox Kinect. Bei diesen werden durch Lagesensoren und Infrarot Kameras die Bewegungen des Spielers erfasst und in entsprechende Spielsituationen eingebunden. Verlässt der Spieler die eigenen vier Wände und hält sich nun im sogenannten environmental space. Hierbei findet die Steuerung der Spiels durch Locomotion, d.h. der Fortbewegung des Spielers statt. [Ben+03; KMS07a]

In Tabelle 3.2 ist eine Gegenüberstellung der einzelnen Bereiche zu sehen.

–Vergleich Vista Space, Environmental Space etc. Tabelle 3.2 Tabelle nach Schlieder/M...

In der aktuellen Literatur werden vermehrt Spiele entwickelt und untersucht, die auf die Nutzung von Smartphones richten. [Ras+06b] Durch Integration von GPS-Modulen, den fallenden Preisen für die mobile Datenübertragung und der Vereinfachung der Entwicklung entsprechender Spiele stehen diesen einer wachsenden Zielgruppe gegenüber.

Eine Spezifizierung der location based games stellen die sogenannten Geogames dar. Dieser Begriff wird vor allem von (Schlieder, Kiefer, May..) gepflegt. Hierbei handelt es sich vorzugsweise um klassische Brettspiele, deren Spielkonzept auf location based games übertragen werden. Die Grundidee ist es, die strategischen Reize der Brettspiele mit den ortsbezogenen Affordanzen der Echtzeit Situation von location based games zu verbinden. In diesem Zug werden rundenbasierte Spielmechanismen ausgetauscht gegen die Locomotion des Spielers. Die damit verbundenen Probleme und Schwierigkeiten werden in [SKM06] beschrieben.

Pervasive Games

Unter Pervasive Games sind Spiele zu verstehen, welche den in diesem Kapitel beschriebenen Magic Circle in seinen beschriebenen Dimensionen erweitern. Konkret werden die definierten Grenzen typischer Spiele überschritten. [Mon05] Hierbei geht es um die Expandierung der ortsspezifischen, zeitlichen und sozialen Grenzen. [MSW09] Darüberhinaus gibt es in der Literatur bei Nieuwdorp(TODO) und Björk(TODO) eine weitere Dimension, welche als 'ambiguity of interaction or interface' definiert wird. Hierbei geht es um Unklarheit bzw. Eindeutigkeit der Interaktion. Eine Vielzahl von Pervasive Games wurde in der Literatur behandelt und entsprechend untersucht. Beispiele hierfür sind Can You See Me Now (Flintham et al., 2003a), GeoTicTacToe, CityPoker, Neocartographer von (Schlieder et al. 2005a,

Schlieder 2005b), Human Pacman (Choetek et al. 2004) aber auch Feed my Yoshi. Anhand dieser Spieler wurden Erkenntnisse in der Praxis gewonnen, welche sich auch in Teilen mit der Gamification Literatur in Kapitel ?? decken. Eine Sammlung interessanter Spielkonzepte stellt die Sammlung von (Hinske et al., 2007) dar.

Aber auch außerhalb der Literatur werden zunehmend Pervasive Games entwickelt. Beispiele hierfür sind Parallel Kingdom oder Ingress vom Google Startup Niantic Project.

Unterscheidung zwischen LBG, AR MR etc?

3.3. Relokalisierungsansätze

Ein wichtiger Aspekt im Zuge von Pervasive Games ist der Gamecontent. Soll ein Spiel außerhalb eines fest definierten Geografischen Raums durchgeführt werden, so ist es notwendig entsprechende fremde Umgebungen mit Inhalt zu füllen. [Mon05] Bei der ortsbezogenen Relokation von Spielinhalten gibt es unterschiedliche Ansätze. Zunächst müssen die ortsbezogenen Affordanzen beachtet werden. Hierbei handelt es sich um die lokalen Gegebenheiten, welche einen Einfluss auf das Spielgeschehen haben.

So muss bei der Durchführung des GeoTicTacToe Spiels in Bamberg überlegt werden, inwiefern die Anordnung der 9 Spielpunkte einen Einfluss auf das Spielgeschehen hat. Brücken, Gebäude und Wege sind nicht strikt linear oder in Quadraten wie in vielen amerikanischen Städten oder z.B. in der Mannheimer Innenstadt. Da eine perfekte Ausgeglichenheit der Spielfelder dazu führen würde, dass jegliche Informationen über Entfernungen, Fußgängerampeln Steigung des Wegs, körperliche Verfassung des jeweiligen Spielers, sowie dessen spatiale Fähigkeiten vorhanden sein müssten, würde ein extrem komplexes Modell ohne Ideale Lösung entstehen. Daher werden die ortsbezogenen Affordanzen meist als gegeben hingenommen bzw. in das Spiel als Herausforderung bzw. Spielelement integriert.

Generell gibt es drei Ansätze zur Relokation der Spielelemente auf einer Karte.

- Keine Anpassung - Spiel nur an einem Ort möglich
- Komplette Anpassung - Spiel an jedem Ort möglich
- Hybride/teilweise Anpassung - Spiel durch Eingriffe spielbar

In der Literatur werden die Probleme von Spielen wie dem Rexplorer[Bal+07] angesprochen, allerdings keine Lösungsansätze vorgeschlagen. Rexplorer enthält keine Anpassungen und ist nur in der Stadt Regensburg spielbar. Neben dem zugeschnittenen Geolocation Content ist auch der Controller explizit auf die Umgebung angepasst. Hierbei handelt es sich um einen Art Zauberstab mit dem der Spieler mit

den Elementen in der Umgebung interagieren kann. Dieser selbst stellt ein Gehäuse dar, in dem ein Nokia E65 sowie in GPS Empfänger verbaut wurde.

Dazu gibt es im Gegensatz Spiele welche komplett Ortsunabhängig gespielt werden können. Dies kann entweder durch ein Algorithmus bewerkstelligt werden oder durch die Tatsache, dass die Spielelemente keinen direkte Anpassung benötigen. Spiele wie Feed my Yoshi, welche keine direkte Anpassung benötigen, haben deutliche Unterschiede im Hinblick ihrer Spielbarkeit abhängig von ihrer Umgebung. Die Autoren stellten fest, dass die Korrelation zwischen Bevölkerungsdichte und Anzahl der Wifis nicht nur den unterschiedlichen Teams unterschiedliche Vorteile verschaffte, sondern dies auch die Motivation der anderen Teams beeinflusst, aufgrund des Vorteils der gegnerischen Teams.

Die letzte Möglichkeit stellt ein hybrider Ansatz dar. Bei diesem werden bestehende Spielfelder von einem gewissen geografischen Kontext auf eine andere Location übertragen. Hierbei wird unter optionaler Hilfenahme von Algorithmen ein Transfer der bestehenden Daten auf ein neues Spielfeld bewerkstelligt. Sofern

Konkrete Lösungsansätze sind in der Literatur mit der Ausnahme von [KMS07a] nicht zu finden. Der von [KMS07a] gewählte Ansatz zielt darauf ab ein Vergleich von verschiedenen Spielfeldern herzustellen um Spieler von verschiedener Herkunft gegeneinander antreten können. Ziel ist es den Aufwand und die Kosten des stagings dieser Spiele zu reduzieren. [KMS07a] identifiziert drei Quellen die zu einer Heterogenität der Spielfelder führen:

- spatial scale - Unterschied in geografische Größe
- static structure - Unterschied in geografischer Struktur (Straßen, Höhe)
- dynamic conditions - Verändernde Gegebenheiten (Wetter, GPS/GSM-Empfang, Verkehr)

Eine gewisse Heterogenität macht die Herausforderung für die Spieler interessanter, allerdings führen zu große Unterschiede zu einem unfairen und damit weniger gutem Spielerlebnis. Generell gibt es zwei Arten von location-based Games. Zum einen örtlich begrenzte (spatial discrete) Spiele und zum anderen örtlich fortsetzende (spatial continuous) Spiele. Bei ersterem handelt es sich um Spiele die auf einem abgegrenzte Spielfeld durchgeführt werden und die Position des Geocontents fest auf der Karte definiert ist. Letztere sind Spielfelder, welche unbegrenzte Spielfelder haben und die Spielinteraktion zu jeder Position stattfinden kann. Im Falle der spatial continuous Spiele ist ein bijectives Mapping der Ort nötig. Dies funktioniert gut bei offenen Flächen, bei der Verwendung von Straßen und innerhalb von Städten führt dies aber zu einer starken Verzerrung der Spielfelder und zu einem Mismatch zwischen Spielerfortbewegung und Lokomotion im Spiel selbst. Dies und die Tatsache, dass

vorwiegend Geogames untersucht wurden führte dazu, dass spatial discrete Spiele im Detail untersucht wurden. Konkret wurde das Spiel CityPoker[KMS05] in zwei verschiedenen Städten gleichzeitig gespielt. Dabei konnten beide Team der jeweiligen Städte über vordefinierte POIs miteinander interagieren. Untersucht wurde dabei die optimale Gestaltung der Spielfelder, damit diese zwischen den Städten den kleinsten Unterschied zueinander darstellen. Zunächst wurden die POIs manuell gesetzt für beide Städte. Im Anschluss auf die Spielsitzung wurde untersucht, wie die unterschiedlichen Spielfelder optimal gelegt werden müssten um ein möglichst optimales Feld zu erhalten. Hierbei ist zu beachten, dass die Reihenfolge der POIs bei Citypoker eine Rolle spielt. Am Punkt 1 können nur Karten getauscht werden, welche beim anderen Team ebenfalls auf 1 liegen. [KMS07a] stellt für den Vergleich eine Entfernungsmatrix auf, welche er über ein Ähnlichkeitsmaß gegenüber gestellt werden. Hierzu wird nachfolgendes Ähnlichkeitsmaß angewandt:

$$similarity = \frac{1}{n} \sum_{row=1}^n \sqrt{\sum_{col=1}^n (c_{1,row,col} - c_{2,row,col})^2} \quad (3.1)$$

Hierbei stellen c_1 und c_2 jeweils die Entfernungsmatrizen der jeweiligen Spielfelder dar. Zu beachten ist, dass die Entfernungen als direkte euklidische Luftlinie gemessen werden. Etwaige Höhenunterschiede, sowie örtliche Gegebenheiten werden hierbei aus Gründen der Vereinfachung nicht beachtet. Anschließend wird das arithmetische Mittel der Durchschnittswerte der einzelnen Reihen gebildet. Als Ergebnis wird angenommen, dass spatial discrete Spiele eine einfache re Konfiguration erlauben wie unter anderem [BML05] angemerkt hat.



	1	2	3	4	5
1	0	762	775	987	1217
2	762	0	578	393	487
3	775	578	0	409	719
4	987	393	409	0	315
5	1217	487	719	315	0

Abbildung 3.3.: Spielfeld Verteilung nach Kiefer et al. [KMS07a]

[BML05] identifizieren mehrere Herausforderungen im Bezug auf spatial continuous games.

- Hefting domains
- Configuration
- Orchestration

Hefting domains stellen die Problematik dar, dass Spielelemente in Computerspielen fokussiert sind auf deren virtuellen Spielwelt, daher müssen bei pervasive games besonderen Wert auf die Designentscheidungen bezüglich der virtuellen, reellen und hybride Spielelementen gelegt werden. Unter der Configuration ist die Adaption von pervasive games an verschiedene lokale Gegebenheiten zu sehen, d.h. die (generierte) Erstellung von Spielfeldern an anderen Orten. Die Orchestration stellt das Management des Spiels während der Laufzeit dar. Hierbei soll sichergestellt werden, dass ein Eingriff in das Spielgeschehen zu Gunsten der Sicherheit der Spieler als auch des Spielerlebnisses möglich ist.

Erste Ansätze für Relokalisierung von Spielfeldern lassen sich in der Literatur finden. Allerdings behandeln diese nur erste Ansätze, wie z.B. [Man12a], welcher eine DSL entwirft zur Nutzung von OSM Daten für das Auffinden gleicher Spielelemente (Uni Campi).

Daher lässt sich abschließend feststellen, dass in der Literatur keine konkreten Lösungen für die (teil-)automatisierte Erstellung von Spielfeldern existiert. An dieser Stelle soll die Arbeit des Autors ansetzen und eine Lösungsmöglichkeit präsentieren.

3.4. Verwendung offener Geodaten

Zunächst ist der Begriff offene (Geo-)Daten zu definieren. Unter offenen Daten sollen im folgenden Daten zu verstehen sein, welche unter freier Lizenz zur Verfügung stehen und somit ohne Lizenzgebühren verwendet werden können. Hierbei soll im Idealfall sowohl eine kommerzielle Nutzung als auch eine private Verwendung erfolgen können. Es lassen sich generell zwei verschiedene Quellen von öffentlichen Geodaten identifizieren. Einerseits gibt es Daten von öffentlichen Behörden. Hierbei gibt es aktuell im Zuge der Open Data Bewegung [Ore07] den Anspruch Daten dieser Behörden zu den Bürgern zur Verfügung zu stellen, da diese durch Steuern gesammelt erstellt wurden. Erste Ansätze lassen sich sowohl in Großstädten wie Wien [Wie14], Hamburg [Ham14] und Berlin [Ber14] finden, als auch für ganze Länder wie Dänemark [Dig14].¹ Die Art, Qualität, sowie Umfang der Daten unterscheiden sich jedoch erheblich. Eine weitere Option sind offene (Geo-)Datenbanken, welche von privaten Personen durch Mapping oder externe lizenzierte Quellen zusammen getragen werden. Beispiele entsprechender Datenbanken sind OpenStreetMap (OSM)² und Wikimapia³. Hierbei stellt sich vor allem die Frage der Qualität der Daten im Vergleich zu kommerziellen bzw. Daten von Behörden.

//TODO: Kurze generelle Info zum Aufbau von OSM - Bild von Relations Ways und Nodes?

¹Eine Übersicht ist unter <http://www.engagedata.eu/opendatasites> (Abgerufen am 11.02.2014) zu finden.

²<http://openstreetmap.org>

³<http://wikimapia.org>



Abbildung 3.4.: OSM Elemente

In der Literatur haben sich bereits viele Autoren mit der Qualität von OSM beschäftigt. [Hak10], [FM08] und [Goo07] beschreiben, die Motivation der Personen und Problematiken, die in solchen Datenbanken entstehen. Es wird angemerkt, dass je nach Ziel des Mappers eine unterschiedliche Qualitätsstufe erreicht wird. [GT10] beschreibt in einem Vergleich von französischer Daten, dass die Qualität der Daten von OSM seit dem Start in 2004 enorm zugenommen hat, je nach Land gibt es unterschiedliche Abdeckungsraten. Auch haben Naturkatastrophen, einen Einfluss auf die Qualität der Daten [Zoo+10]. Zur Zeit gibt es bei OSM ca. 25.000 aktive Mapper [Ope13b], dass Mapping in Städten ist deutlich genauer wie in den ländlichen Regionen. Generell gibt es eine Korrelation zwischen Einwohnerdichte und Datenqualität. Die durchschnittliche Abweichung beim Vergleich von OSM zu Kartenherstellern beträgt zwischen 1 bis 30 Metern. Darüber hinaus gibt es auch eine Ungenauigkeit der Namen einzelner Objekte. Diese entstehen zum einen durch die Nutzung unterschiedlicher Sprachen, aber auch durch lokale Gegebenheiten. Die Einfachheit von OSM durch die Reduktion der Daten auf Nodes, Ways und Relations mit den dazugehörigen Tags hat den Nachteil, dass im Modell selbst keine logische Konsistenz sichergestellt wird. Dies muss in der Verarbeitung der Daten stets berücksichtigt werden. [HKH13] beschreibt eine relativ geringe Coverage im Vergleich zu professionellen Daten gibt 10%. Dies wird auch von [Pfo+13] angemerkt, jedoch wird darauf hingewiesen, dass OSM eine hohe Klassifikationsrate besitzt. Zwar stellt der Autor eine relativ hohe Fehlerrate von 23% fest, jedoch ist diese zu für den vernachlässigen. Für OSM gibt es im Vergleich zu Wikmapia ausgereifte Schnittstellen. Zunächst gibt es zwei Schnittstellen direkt von OSM. Die OSM API ermöglicht lediglich den Export der Geoinformationen bezogen auf eine Bounding Box. Während unter dem Namen OSM XAPI (Extended Api), es möglich ist Abfragen in Verbindung mit den zugehörigen Tags zu erstellen [MTS13]. Die Ergebnisse der Abfrage werden in einer XML Datei zusammengefasst.

3.5. Bewertung von Spielfeldern

Kapitel wird passend zum Evaluationskapitel gefüllt

4. Lösungsansatz

4.1. Mögliche Lösungen

Gamification

Im Hinblick auf die Problemstellung in Kapitel 2, sowie der in der Literatur beschriebenen Ansätze in Kapitel 3.1, ergeben sich mehrere Möglichkeiten für eine Lösung. Ziel ist es die Kunden zum einen zu motivieren neue Geschäfte aufzusuchen und auf der anderen Seite eine Bindung zu bestehenden Geschäften zu erreichen. Da es sich hierbei um einen Prozess handelt, der für den Kunden im ersten Schritt nur einen geringen Benefit darstellt, ist dieser ideal für Gamification geeignet [Lei12]. Ziel ist es somit die Besuche des Spielers durch eine Gamification interessant zu machen. Dies kann durch die verschiedenen Aspekte wie das Vergeben von Punkten für einen Einkauf (siehe Miles & More [Wag05], Payback [Rös05]) erreicht werden. Allerdings sind reine, einfache Punkteprogramme weit verbreitet und deren Nutzen umstritten [Sch01]. Daher ist es notwendig, das komplette Spektrum der Gamification zu betrachten. Über das reine Punkte System hinaus, manifestiert sich in der Literatur die Grundmenge "Points, Badges und Leaderboards". Konkret bedeutet dies, neben der einfachen Ansammlung von Punkten, gibt es darüber hinaus Badges/Auszeichnungen welche das Engagement des einzelnen Nutzers widerspiegeln. Ein weiterer Aspekt sind die sogenannten Leaderboards bzw. Bestenlisten. Über diese wird ein Ansporn unter den jeweiligen Spielern erzeugt die anderen Spieler zu übertrumpfen. Vergleicht man diese Ansätze mit denen der typischen Spieler Profile nach [Bar04], wie in Abbildung 4.1 zu sehen, so erkennt man, dass nicht alle dieser mit den Standard Elementen bedient werden.

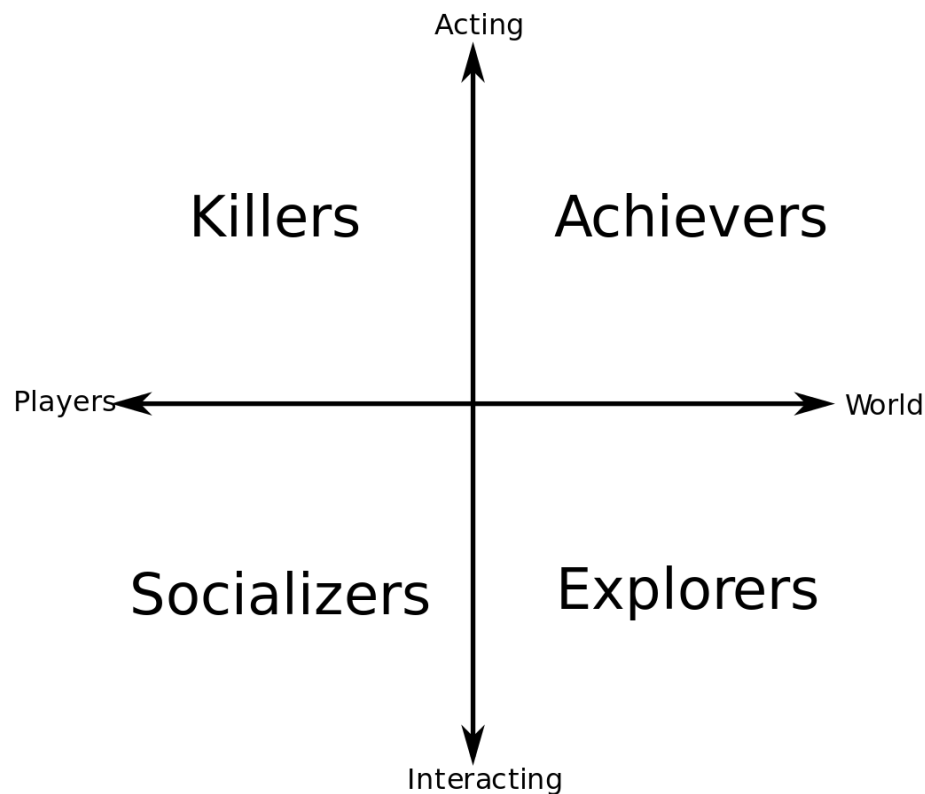


Abbildung 4.1.: Spielertypen nach Bartle [Bar04]

Die rudimentären Elemente der Gamification richten sich vor allem an Achievers und Killers. Die Frage stellt sich hierbei, wie Socializers und Explorers erreicht werden können. Hierzu kann das von [ZC11] in Kapitel 3.1 erweiterte Modell SAPS genutzt werden, welches auch auf die Interaktion unter den Menschen eingeht. Im konkreten Beispiel schlägt Zichermann vor, den Spielern die Möglichkeit zu geben bestimmte Positionen innerhalb eines Spiels einzunehmen. Durch diese sollen einzelne Spieler eine höhere Position erhalten als andere und damit auch eine Interaktion untereinander fördern. Damit kann können zumindest zu einem gewissen Teil die Socializers abgedeckt werden. Es ist jedoch zu beachten, dass für die vollständige Abdeckung aller Spielertypen konkret eine Spielmechanik erarbeitet werden muss, welche die übrigen zwei Spielertypen explizit bedient. Möglichkeiten sind hierbei eine Interaktion zwischen den Spielern zu schaffen, stellen z.B. das Erstellen von Teams/Gilden dar, welche gemeinsam aktiv handeln. Gemeinsame Ziele können so einfacher verfolgt werden aber auch speziell durch das Spiel ein Zusammenhalt explizit gefördert werden. Dies kann z.B. anhand von Bonuspunkten oder aber von Handlungen die explizit die Interaktion von mehreren Spielern erfordern. Ein Beispiel hierfür ist das Spiel Ingress von Google welches für die Erstellung von Portalen im späteren Spiel umso mehr Spieler benötigt, je höher dessen Level sein soll. [Cel13] Andere Ansätze sind kooperative Quests, Handel von Spielitems, sowie das Fördern von Kommuni-

kation zwischen den einzelnen Spielern. [CMAK13] Ein weiterer Aspekt sind neue Bekanntschaften, die durch ortsbezogene Spiele gemacht werden. In einer Umfrage von 6400 Ingress Spielern gaben 84% an neue Leute kennen gelernt zu haben und zählten dies zu einem wichtigen Faktor des Spiels [Lui14]. Explorer untersuchen Ihre Umgebung und möchten neue Dinge entdecken. Sie fragen nach zusätzlichen Informationen. Für diese Spieler ist es z.B. interessant n neue unbekannte Orte innerhalb des Spiels zu gelangen.

Dass die einzelnen Spielertypen nicht zu vernachlässigen sind wird in [Bro+13] deutlich. Hierbei stellt der Autor am Beispiel von GoPets fest, dass durch die Orientierung an konkrete Spielertypen ein beachtliche Umsatzsteigerung erzielt werden kann. [Bet07] Allerdings ist es wichtig für den Erfolg eines Spieles, dass alle Spielertypen adressiert werden.

Nachdem die (Spiel-)Elemente der Gamification selbst identifiziert wurden, muss untersucht werden, wie die Interaktion der Spieler mit dem lokalen Einzelhandel aussehen kann. Der Unterschied des Einzelhandels im Vergleich zum Onlinehandel ist der geografische bzw. physikalische Besuch. Der Kunde muss physikalisch vor Ort sein, um eine Dienstleistung wahrzunehmen oder aber ein Produkt zu kaufen. Daher ist es für den jeweiligen Händler von besonderem Interesse, dass ein potentieller Kunde sein Geschäft betritt. Ziel ist es somit den Spieler durch die Spielmechanik in das jeweilige Geschäft zu bringen. Hierbei gibt es zwei Mögliche Ansätze für ein ortsbezogenes Spiel. In der ersten und einfachsten Möglichkeit tritt das Geschäft selbst als festes Spielelement auf, dass direkt zum Spielerfolg beiträgt. In diesem Fall repräsentiert das Geschäft zum Beispiel ein TicTacToe Feld. Eine andere Möglichkeit ist es die Spieler indirekt zu unterstützen. In diesem Fall tritt das Geschäft z.B. als virtueller Laden im Spiel auf, über den die Spieler entsprechende Items kaufen können. Welche der beiden Möglichkeiten in Frage kommt, hängt von der Intention des einzelnen Händlers ab. Steht der Fokus des Spiels auf einem reinen Kontakt mit dem Spielelement, so muss sicher gestellt werden, dass die Spieler nicht einfach nur den jeweiligen Laden einfach nur betreten und dann wieder direkt verlassen. Im schlimmsten Fall kann dies sogar dazu führen, dass normale Kunden die nicht am Spiel selbst teilnehmen sich durch die Spieler gestört fühlen. Da es hierfür keinerlei Erwartungswerte gibt, ist der Aspekt schwer abzuschätzen, allerdings wäre eine mögliche Gegenmaßnahme, die Verweildauer der Spieler zu erhöhen, in dem Sie z.B. ein kleines Rätsel lösen müssen oder für eine gewisse Zeit im entsprechenden Raum aufhalten müssen. Des weiteren kann dem Händler auch die Möglichkeit gegeben werden, wiederkehrende Spieler bevorzugt zu behandeln. Ein Konzept welche z.B. bei Foursquare verwendet wird ist der Mayor einer Location. Der Mayor stellt dabei die Person,da welche die Lokalität innerhalb der letzten 60 Tage am häufigsten be-

sucht hat. Diesem kann ein Händler über Foursquare besonderes Angebot machen, wie zum Beispiel ein Kaffee zum halben Preis.[Lin+11] Ähnliche Konzepte können hier analog ebenfalls verwendet werden.

Da das Hauptaugenmerk darauf liegt, dass der Spieler mit dem jeweiligen Geschäft interagiert, muss sichergestellt werden, dass dieser physikalisch vor Ort ist. Um dies sicherzustellen, gibt es mehrere Möglichkeiten. Die einfachste Variante ist es einfach die GPS-Koordinaten des Systems zu verwenden. Der Vorteil liegt darin, dass bei den meisten mobilen Betriebssystemen eine A-GPS Implementierung vorliegt, die die Nachteile von reinem GPS unter Zuhilfenahme einer bestehenden Mobilfunkverbindung kompensiert [DR01]. Allerdings könnte es sein, dass der Empfang in einzelnen Gebäuden eingeschränkt ist und weder eine Verbindung zu GPS-Satelliten noch zum Mobilfunknetz hergestellt werden kann. Darüber hinaus besteht die Problematik, dass die Genauigkeit nicht ausreicht um festzustellen, ob die Person vor dem Geschäft oder im Geschäft selbst ist. Es muss daher sichergestellt werden, dass der Spieler wirklich im Geschäft selbst ist. Hier zu könnte man QR-Code verwenden die entsprechend abgescannt werden und darüber eine Interaktion über eine URL oder App initiiert werden. Eine andere neue Möglichkeit stellt NFC dar [Ras+06a]. Über diese kann im Bereich von bis zu 20cm eine Interaktion stattfinden. [CRB06] passive NFC Elemente arbeiten in diesem zusammenhang analog zu QR Codes was den Datenaustausch angeht. Der Unterschied liegt darin, dass die Daten nicht per Kamera sondern per NFC erfasst werden. Diese Ansätze breiten sich aktuell im Handel immer mehr aus [Hei14]. In Seoul gibt es z.b. virtuelle Stores, bei denen es sich um Werbeposterflächen handelt, die ein Regal mit QR-Code darstellen [Tes12]. Diese werden vor allem in besonders gerne während des Feierabendverkehrs genutzt. Eine neue Technologie die neben den QR-Codes und NFC zum Einsatz kommen könnte sind Bluetooth Low Energy Devices (BLE). Apple setzt bereits diese in den Appstores ein und einige Startups sehen in der Technologie eine Möglichkeit den lokalen Einzelhandel mit dem Online Geschäft zu verbinden. Konkret handelt es sich dabei um passive BLE Geräte welche bis zu 30 Meter haben können und deren Entfernung über eine Approximierung von RSSI und der übermittelten Referenzsignalstärke angegeben wird. [DG13] Durch den Einsatz von mehreren sogenannter Beacons kann analog zur GPS Lateration eine Indoor Positionierung erreicht werden.¹ Somit könnte der konkrete Aufenthaltsort in einem speziellen Bereich z.B. der Herrenmode belohnt werden.

Ein Aspekt der nicht außer Acht gelassen werden darf, ist dass bei Spielen, bei denen es darum geht mit anderen Spielern verglichen zu werden, immer zwangs-

¹Ein entsprechendes System wurde vom Autor entwickelt und evaluiert

läufig zum Cheating kommt. Consalvo [Con05] Ein besonderer Augenmerk gilt hier dem Multiplayer Cheating. Dabei geht es konkret um eine Vorteilsnahme gegenüber den anderen teilnehmenden Spielern. Yan/Randell [YR05] identifiziert mehrere Möglichkeiten des Cheatings. Für diese gibt es jeweils entsprechende Gegenmaßnahmen. Als Grundsatz kristallisiert sich daher heraus, dass der Schutz eines Spiels analog zum investierten Aufwand gegen entsprechende Versuche der Manipulation verhält und dass kein System sicher gegen eine Manipulation ist. Jeder Spielebetreiber muss basierend auf seinen verfügbaren Ressourcen entscheiden, welche Methoden dieser einsetzen möchte. Auch ortsbezogene Spiele bleiben von Manipulationen nicht verschont wie He et al. [HLR11] untersucht. In diesem Fall ist ein erster Ansatzpunkt die maximale Fortbewegungsgeschwindigkeit. Kombiniert man diese mit der Spielmechanik wie z.b. einer Pflicht Interaktionsdauer von 30 Sekunden pro POI, so reduziert sich auch der "Vorteil" durch das Cheaten.

Anforderungen an ein Geogameframework

Für die Umsetzung eines Geogameframeworks muss zunächst untersucht werden welche Anforderungen an ein solches gestellt werden. Im Hinblick auf die in Kapitel 2 beschriebene Problemstellung ist es zunächst sinnvoll die Anforderungen zu identifizieren und im Anschluss zu analysieren. Werden diese abstrahiert auf die Aspekte eines Gameframeworks so ergeben sich daraus entsprechende Vorgehensweisen.

Ziel ist es ein ortbezogenes Spiel zu entwickeln, welches dazu führt, dass die Spieler lokale Dienstleister/Einzelhändler vermehrt aufsuchen. Ein Gameframework muss in diesem Fall einem Spielleiter es ermöglichen unter möglichst wenig Aufwand ein entsprechendes Spiel durchzuführen. Dem Spielleiter sollen die Arbeitsschritte für die Selektion der Spielelemente anhand von Geodaten als auch die Durchführung abgenommen werden. Ziel ist es, dass der Spielleiter selbst mit möglichst geringen Technischen Kenntnissen ein entsprechendes Spiel durchführen kann. Daher halten wir als ersten Aspekt das einfache Staging der Spiele fest.

Für die Steigerung der Interaktion von potentieller Kunden wurde die Gamification und letztendlich die konkrete Spielmechanik in Kapitel 3.1 identifiziert. Um diese entsprechend umsetzen zu können, muss ein entsprechendes Framework nicht nur entsprechende Logik, sondern auch die entsprechenden Daten inne halten. Abhängig wie hoch der Flexibilität Anspruch des Frameworks selbst ist, macht es Sinn Spiel-Logik und Spiel-Daten weiter zu trennen. Soll das Framework nur für eine geringe Zahl ähnlicher Spiele verwendet werden, so kann es Sinn machen die Funktionen und Datenstruktur direkt in das Framework einzuplanen. Ist es allerdings der Plan

ein Framework zu entwickeln, welches möglichst generisch arbeitet und später eine Vielzahl an unterschiedlichen Spielen unterstützen soll, so ist eine entsprechende Modularisierung der einzelnen Funktionen unerlässlich.

Bei der Durchführung eines ortsbezogenen Spiels muss ein Framework mit Geodaten arbeiten. Dabei ist es unerlässlich, dass ein entsprechendes System nicht mit einem einfachen kartesischen Koordinaten System arbeitet, sondern ein entsprechendes Referenzsystem wie WGS 84 [GO98] und das UTM Koordinaten System [Gra95]. Für das jeweilige zu nutzende System müssen entsprechende Bibliotheken integriert werden, damit entsprechende Operationen wie zum Beispiel Distanzmessung auf den Geodaten durchgeführt werden können.

Das Framework muss zudem die Integration vom externen GeoDaten ermöglichen. Bei diesen wird es sich aufgrund der Ergebnisse in Kapitel 3.4 um OSM handeln. Das Framework muss auf Basis der OSM Daten entsprechende Spielfelder aufbereiten. Hierzu müssen die bestehenden Objekttypen (Nodes, Ways, Relations) in Spielelemente transferiert werden. Hierfür gibt es mehrere Möglichkeiten. Zunächst gibt es Nodes. Diese können direkt als Spielelement dargestellt werden, da sie analog zu den Spielelementen ein Objekt repräsentieren und ihnen eine direkte Koordinate auf dem Spielfeld zugeordnet werden kann. Nodes die jedoch Bestandteil eines Ways oder einer Relation sind sollten nicht als einzelnes Spielelement betrachtet werden. Gibt es ein Objekt, welches als Way oder Relation in OSM repräsentiert wird, so muss dieses zu einem Spielelement mit nur einer Koordinate transformiert werden. Hierzu müssen alle Nodes welche innerhalb eines Ways liegen zu einem "virtuellen"Node zusammengefasst werden. Wichtig ist hierbei eine Adressierung der virtuellen Nodes zu ermöglichen. Damit diese entsprechend wieder gefunden werden können. Für die Transformation mehrere Nodes zu einem Element gibt es unterschiedliche Ansätze. Der erste und einfachste stellt ein einfaches Arithmetisches Mittel auf die äußersten Kanten des Ways dar. Konkret bedeutet das, dass um alle Nodes eines Ways quasi eine Bounding Box gebaut wird. Die beiden Koordinaten die die Bounding Box begrenzen werden anschließend mit einem arithmetischen Mittel miteinander kombiniert zu einem einzelnen Punkt. Diese Methode ist besonders einfach und der Punkt liegt immer innerhalb der Bounding Box, allerdings spiegelt er nicht die Verteilung der einzelnen Objekte wieder bzw. nimmt auf die konkrete geografische Struktur des Ways keine Rücksicht:

$$x_{bb} = \frac{\max_{i>0; i \leq N}(x_i) + \min_{i>0; i \leq N}(x_i)}{2} \quad (4.1)$$

$$y_{bb} = \frac{\max_{i>0; i \leq N}(y_i) + \min_{y>0; i \leq N}(y_i)}{2} \quad (4.2)$$

Eine weitere Methode die diese Unzulänglichkeit behebt stellt der geometrische Schwerpunkt dar. Bei diesem wird der Massenmittelpunkt des Körpers bestimmt[Pap06]. Hierfür ist zunächst die Berechnung der Fläche nötig:

$$A = \frac{1}{2} \sum_{i=0}^{N-1} (x_i y_{i+1} - x_{i+1} y_i) \quad (4.3)$$

Dieses kann wiederum verwendet werden um das geometrische Gewicht zu ermitteln:

$$x_s = \frac{1}{6A} \sum_{i=0}^{N-1} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (4.4)$$

$$y_s = \frac{1}{6A} \sum_{i=0}^{N-1} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (4.5)$$

Die jeweiligen x- und y-Koordinaten liefern damit die entsprechenden Koordinaten auf dem Spielfeld.

Der Gedanke eines Frameworks zielt auf eine einfache Erweiterbarkeit, Pflege sowie Wiederverwendbarkeit ab. Daher ist die Erweiterbarkeit für weitere Spiele und Spielmechaniken ein wichtiger Aspekt. Ziel ist es daher ein Grundgerüst zu entwickeln, welches durch entsprechende Anpassung einfach angepasst werden kann. Je nach Fokus auf Flexibilität, welche das Framework bieten soll und Umfang, müssen entsprechende Funktionen und Methoden generischer oder spezifischer implementiert werden.

Der in Kapitel 1 angestrebte Mehrwert des Frameworks, neue Kunden und bestehende Kunden an lokale Einzelhändler zu binden stellt ein wichtiger Aspekt dar. Generell gibt es verschiedene Möglichkeiten wie Lokalitäten in ein Spiel integriert werden können [Nel02]. Zunächst gibt es die Möglichkeit Werbebanner direkt in Spiele einzubinden [NKY04]. Diese Werbung gleicht der Printwerbung mit dem Unterschied, dass die Sichtbarkeit im Vergleich zu klassischen Medien oder Filmen deutlich erhöht ist und sich analog zur durchschnittlichen Spielzeit verhält. Spiele die soziale Aspekte betrachten, haben im Schnitt höhere Spielzeiten [Duc+06], bei älteren Spielern sind bis zu 30 Stunden pro Spiel der Durchschnitt.

Eine andere Möglichkeit stellen so genannte Advergames dar. Bei diesen handelt es sich um Spiele die rein auf dem zu bewerbenden Produkt bzw. der Dienstleistung selbst fokussiert sind. Diese haben einen deutlich größeren Einfluss auf die entspre-

chende Zielgruppe. Allerdings werden hierbei vor allem Personen angesprochen die Spielen gegenüber offen sind [TB06]. Chen/Ringel [CR01] und Dahl et al. [DEB09] beschreiben, dass jüngere Personen bis zu 100 mal mehr spielen als ältere und der Kunde die Marke durch Advertgames deutlich mehr wahrnimmt im Vergleich zu anderer Werbung. Die Möglichkeiten für die Integration von Produkten sind vielfältig. Neben der passiven Einbindung des Produktes in die Spielumgebung, kann das Produkt auch selbst als Spielelement dienen (vgl. Colgate 1992: Harald Hårtand - kampen om de rene tænder). Darüber hinaus könnte im Fall eines ortsbezogenen Spieles der Händler/Dienstleister im Spiel selbst als Pendant seines Händlers in der Realität auftreten. D.h. ein Händler stellt im Spiel selbst einen Händler für Spielitems dar. Somit lässt sich die Einbindung generell in zwei Arten unterteilen. Einmal als aktiv für den Spielausgang entscheidendes Element (z.B: als POI für TicTacToe) oder aber als Vendor Element welches der Spieler aufsuchen kann. Im Hinblick auf die Spieleitems sollten diese vorzugsweise im Zusammenhang mit dem Händler und der Spielthematik stehen. Je besser sich diese in das Spiel integrieren, desto eher werden sie vom Spieler als plausibel wahrgenommen. Dies wiederum hat einen positiven Effekt für die Wahrnehmung. Ein Beispiel für die Integration könnte z.B. ein Supermarkt sein, welcher Essen anbietet. Dieses kann wiederum vom Spieler zur Auffrischung der Energie verwendet werden. Ein anderes Item könnte eine Schaufel darstellen, welche beim lokalen Baumarkt gekauft und im Spiel zum Bauen von eigenen Gebäuden verwendet werden kann.

Abschließend muss das Framework entsprechend modular aufgebaut werden sein. Dies soll vor allem der Erweiterbarkeit und der Portabilität dienen. Da ein ortsbezogenes Spiel auf entsprechenden Endgeräten gespielt wird. Konkret Smartphones ist es darüber hinaus wichtig entsprechend die Spielelemente für den Spieler aufzubereiten. Hierzu gibt es verschiedene Möglichkeiten. Generell gibt es unterschiedliche Smartphone Betriebssysteme deren Marktanteile sich je nach Geografischer Location unterscheiden. Schaut man sich die globale aktuelle Verteilung der mobilen Betriebssysteme an, wie in Abbildung 4.2 zu sehen.



Abbildung 4.2.: Globale Mobile Betriebssysteme in Anlehnung an Gartner [Gar13] - Bild: [Wik14]

Die aktuelle Verbreitung von ca. 80% Android, 18% iOS und 3% Windows Phone bei Smartphones stellt die Frage, für welche der besagten Plattformen man Anwendungen erstellen möchte. Die erste Möglichkeit sind native Apps, zweitens sind Web Apps [CL11]. Der Unterschied beider liegt im Aufwand der Implementierung, der Ausführungsgeschwindigkeit, Zugriff auf die Hardware und dem unterschiedlichen Deployment. Native Apps erreichen in den entsprechenden Appstores und bieten Zugriff auf allerlei Hardware Features. Konkret für das Geogameframework sind zweierlei Dinge wichtig.

- Möglichst wenig Aufwand für das Deployment
- Zugriff auf GPS Koordinaten

Dies kann sowohl über native Apps als auch über eine Web Applikation bewerkstelligt werden. Entsprechende SDKs der jeweiligen Hersteller bieten per API Zugriff auf die GPS Schnittstelle. Für WebApps kann im Zuge von HTML5 ebenfalls auf eine GPS Ortung zurückgegriffen werden [Hol11]. Um den Entwicklungsaufwand für Webapps zu reduzieren kann es sinnvoll sein auf entsprechende Webframeworks aufzusetzen, damit nicht grundlegende Funktionalitäten erneut entwickelt werden müssen. Eine entsprechende Entscheidung ist in diesem Fall abhängig von den Anforderungen. Eine entsprechende Auswahl wird in Kapitel 5 getroffen.

Abschließend lassen sich die einzelnen Anforderungen des Gameframeworks wie folgt zusammenfassen:

- einfaches Staging der Spiele
- Modularisierung der Funktionen für Gamification/Spielmechanik
- Verarbeitung von Geodaten
- OSM-Daten Transformation/Aufbereitung
- Einfache Erweiterbarkeit der Spielmechanik
- Einbindung der lokalen Händler
- Modularer Aufbau

Relokalisierbarkeit von ortsbezogenen Spielen

Für die Relokalisierbarkeit von Spielfeldern wurden in Kapitel 3.3 erste Ansätze in der Literatur identifiziert. Diese beziehen sich allerdings speziell auf spatial discrete Spiele. Generell erfolgt die Unterteilung in die zuvor identifizierten Möglichkeiten. Es kann entweder keine Anpassung, eine vollständige Anpassung oder ein Hybrid Ansatz erfolgen. Der ideale Fall stellt eine vollständige Anpassung und damit Relokalisierbarkeit eines Spiels ohne das manuelle Anpassen eines Spielleiters dar. Hier für gibt es in der Literatur allerdings keine konkreten Ansätze. Vielmehr werden konkrete Spiele wie REXplorer [Bal+07] oder bestimmte Geogames [KMS07a] adaptiert durch eine vorgegebene Liste an POIs zu einem optimalen Spielfeld in einer anderen Stadt von der ebenfalls eine Liste von POIs existiert. Einen evaluierten nutzbaren Ansatz für die automatisierte Relokalisierung gibt es daher nicht. Ein Ansatz der sich aus der Hypothese herausbildet, ist der Versuch Aufgrund bestehender Geodatenbanken - konkret Openstreetmaps - eine Relokalisierung zu erreichen. Hierbei müssen allerdings neben der bereits angesprochenen Transformation der Relationen, Ways und Nodes auch eine Auswahl der entsprechenden Spielelemente stattfinden. Denkbar sind hierbei zwei Ansätze. Um ein "gutes Spielfeld" zu identifizieren und somit eine Relokalisierung für gut zu befinden muss entsprechend ein Evaluationsansatz existieren. Ansätze zur Evaluation von Spielfeldern sollen im Kapitel 6 näher betrachtet werden. Generell kann zwischen einer Echtzeit und einer prestaging Evaluation unterschieden werden. Je Form hat ihre eigenen Vor- und Nachteile. Bei einer Echzeitevaluation der Spielfelder liegt der Vorteil in völligen Automatisierung der Auswahl der Spielelemente und weniger Aufwand für den entsprechenden der ein Spiel ausrichten möchte. Allerdings muss gleichzeitig der zu verwendende Algorithmus in Echtzeit zu einem entsprechenden Ergebnis für die Auswahl der

Spielelemente kommen. Da die ortbezogenen Pervasive Games selbst in Echtzeit laufen kann ein entsprechender Algorithmus keine größere Komplexität aufweisen und wird damit in der möglichen Qualität der Spielfelder eingeschränkt.

Im Gegensatz dazu steht eine Evaluation und Auswahl der Spielelemente vor dem Staging. Die Vorteile sind die mögliche Komplexität des Algorithmus, sowie die damit verbundene Qualität der Spielfelder. Ein Nachteil ist allerdings, dass eine Evaluation der Spielfelder im Voraus erfolgen muss. Da bei einem Pervasive Game allerdings das Spielfeld die ganze Welt darstellt, ist eine Laufzeit stark abhängig vom Algorithmus sowie der Datenbasis.

Eine andere Möglichkeit stellt ein hybrider Ansatz dar. Dabei wird ein Teil des Evaluation vorausgelagert und ein weniger komplexer Teil für die Echtzeitberechnung übrig. Eine Möglichkeit wäre es z.B. Im Voraus eine Selektion der Spielelementtypen und die konkrete Auswahl in diesem Subset an Daten während der Laufzeit zu machen. Beispielsweise mit OSM könnte im Voraus zunächst eine Auswahl der Kategorisierung anhand eines key-value Paares erfolgen (highway=bustop) und im Anschluss wird während der Laufzeit entschieden ob entsprechende Objekte zu dicht beieinander liegen. Dann könnten die Elemente entsprechend außer Acht gelassen werden oder bei zu geringer Verbreitung entsprechende dummy-Elemente eingefügt werden.

Freie Geobasisdaten und Möglichkeiten der kommerziellen Nutzung

In Kapitel 3.4 wurden entsprechende öffentliche Geodatenbanken identifiziert welche Geodaten enthalten, die für eine Nutzung für ein ortsbezogenes Spiel in Frage kommen. Von diesen hat sich lediglich OSM als brauchbar und mit entsprechender Daten-Qualität erwiesen. Untersucht man Openstreetmap bezüglich seiner Lizenz für die Verwendung in kommerziellen Projekten, so lässt sich zunächst der Wechsel der Lizenz im Jahre 2012 auf ODbL erkennen [Ram12].

Vergleicht man die Anforderungen eines Gameframeworks an die entsprechenden Geodaten so ist der Punkt wichtig, dass im Vergleich mit anderen Open Source Lizenzen eine Kommerzielle Nutzung möglich ist, ohne den Quellcode des Programmes oder zusätzlicher externen Datenquellen selbst zu veröffentlichen. Die ODbL in Version 1.0 [Com14] beschreibt hierbei, dass ein Auszug aus der Datenbank selbst als Derivative Database anzusehen ist. D.h. konkret in Verbindung mit OSM, dass das Ergebnis einer Abfrage an den OSM Server als Datenbank anzusehen ist, welche nach ODbL zu behandeln ist. Der Nutzer, hier: Betreiber, ist verpflichtet, wenn die OSM Daten selbst modifiziert werden oder in irgend einer Weise aufgewertet/aufbereitet und der Öffentlichkeit (über das Internet) zur Verfügung gestellt werden,

diese ebenfalls öffentlich zu machen. Eine private Verwendung muss nicht veröffentlicht werden. Gleiches gilt für die zusätzlichen externen Datenquellen welche mit den OSM Daten vermengt werden. Dies können zum Beispiel kommerziell eingekaufte Höhenprofile oder Landschaftsaufnahmen sein. Diese bleiben somit davon unberührt. Vergleicht man nun die Pflichten die durch die Nutzung der ODbL in Verbindung mit OSM entstehen mit den Anforderungen des Gameframeworks, so lässt sich feststellen, dass eine Nutzung der OSM Daten juristisch unbedenklich ist, allerdings die abgerufenen Daten über eine dokumentierte Schnittstelle wieder zur Verfügung gestellt werden sollten. Wie eine Dokumentation und zur Verfügungsstellung der Daten im Detail aussieht ist nicht näher spezifiziert. Hier eignet es sich eine bestehende Schnittstelle, welche im Entwurf de Frameworks geplant ist entsprechend so zu erweitern, dass diese entsprechend die Daten zur Verfügung stellt. Idealerweise stellt dies eine Schnittstelle dar, die analog zur Datenaufbereitung für das Spielfeld dient. Alternativ könnte man, um die Performance des Spiels nicht zu beeinträchtigen auch einen separaten kleineren Server dafür abstellen, welcher die gleichen Funktionen bietet, aber verhindert, dass durch die Abfrage von größeren Datenmengen nicht das laufende Spiel beeinträchtigt wird.

4.2. Gewählter Lösungsansatz

Betrachtet man die in Kapitel 1 beschriebene Forschungsfrage, so ist der Fokus der Arbeit vor allem auf der Untersuchung einer Möglichkeit zur möglichst einfachen Aufbereitung von Spielfelder zu sehen, bei denen der Spielveranstalter so wenig wie möglich sich selbst einbringen muss. Darüber hinaus sollen die lokalen Händler/Dienstleister in das Spiel eingebunden werden, damit diese entsprechend einen Mehrwert durch Kundenbindung erfahren. Da der Fokus auf die Aufbereitung und Evaluation der Spielfelder gelegt wurde, werden für das Beispielspiel nur ein teil der Aspekte umgesetzt.

Für die Gamification selbst ist es nach Analyse der Möglichkeiten am sinnvollsten sich überwiegend auf die Basis Elemente Points Badges und Leaderboards zu beschränken. Spätere Elemente die Zichermann/Cunningham [ZC11] herausgearbeitet haben, sollten entsprechend einfach erweiterbar sein, aber nicht den Fokus darstellen. Um die einzelnen Aspekte besser zu verdeutlichen, werden die verwendeten Aspekte am Beispielspiel erläutert.

Das Beispielspiel selbst stellt eine abgewandelte Variante des beliebten Capture The Flag Spielmodus [AWC99]. Es gibt verschiedene Flaggen auf dem Spielfeld. Diese Flaggen symbolisieren entsprechende POIs, welche auf Basis des Gameframeworks unter der Nutzung von OSM generiert wurden. Dem Spieler stehen entsprechende

Aktionspunkte zur Verfügung, welche er einsetzen kann um diese Flaggen zu besetzen. Jeder eingesetzte Aktionspunkt, führt zu "Prestige" auf der Flagge. Der Spieler kann Flaggen, welche von fremden Spielern bereits besetzt wurden übernehmen, indem er diese durch den Einsatz seiner Aktionspunkte um deren Prestige verringert bis diese neutral sind. Neutrale Flaggen kann der Spieler wiederum einnehmen. Dem Spieler stehen pro Tag 24 Aktionspunkte zur Verfügung. Jede Stunde erhält der Spieler einen Aktionspunkt, sofern er noch nicht mindestens 24 Aktionspunkte hat. Durch den Kauf von Items über Händler unter dem Einsatz Aktionspunkte, kann der Spieler seine Fähigkeiten verbessern. Der aktuelle Punktestand, welcher sich aus dem Prestige aller Flaggen die dem Spieler angehören berechnet, ist dem Spieler ersichtlich. Über diesen hinaus kann sich der Spieler mit Spielern aus seiner geografischen Nähe, sowie seinen Mitspielern in der Rangliste unter und über ihm messen.

Die Wahl für ein der Spielmechanik nach unkompliziertes Spiel fiel, da der Fokus auf der Generierung der Geodaten bzw. Spielelemente lag und darüber hinaus durch die Repräsentation der Spielelemente durch einfache POIs die Verwendung des Geogameframeworks generisch genug ist um sowohl für Pervasive Games als auch für Geogames eingesetzt werden zu können. Die Spielmechanik, welche an Capture the Flag erinnert, ist besonders beliebt bei Spielern und hat sich auch in bestehenden Pervasive Games bewährt [Bel+06; Lui14]. Darüberhinaus erfüllt es in der genannten Abwandlung bereits einige Aspekte der Gamification Elemente. So wird der Status des Spielers bzw. sein "Prestige" direkt anhand der Flaggen auf dem Spielfeld sichtbar. Er kann sich auch mit anderen Spielern messen und seine Interaktion mit dem Spielfeld wird für die anderen Spieler direkt in Echtzeit ersichtlich. Die Einführung der Aktionspunkte für die Interaktion mit der Umgebung hat zwei Intentionen. Zum einen soll sichergestellt werden, dass die ortsbezogenen Affordancen für die einzelnen Spieler normalisiert werden. Hiermit wird versucht ein quasi Handicap für besonders aktive und sich schnell fortbewegende Spieler zu erstellen. Es wird darüber hinaus verhindert, dass der Spieler sich nur kurz in bestimmten Regionen aufhält und das Spiel somit nicht zu einem Rennspiel sich ausbildet [SKM05]. Ein weiterer Nebeneffekt ist die Reduzierung von möglichen Betrugsversuchen, da selbst eine Automatisierung der Spielerclients durch z.B. Bots keinen entscheidenden Vorteil bringt [GD05]. Durch die Limitierung der Aktionspunkte auf 24 pro Tag wird der Spieler nichtsdestotrotz animiert jeden Tag im Spiel aktiv zu sein, da ansonsten seine Aktionen "verfallen". Items, welche der Spieler von Händlern kaufen kann, sollen als Gegenstand dem Spieler repräsentiert werden, der nicht nur einfach beim Händler kaufbar und dann im Inventar verwendbar ist, sondern auch mit anderen Spielern teilbar. Die Idee dahinter spielt auf die beliebten Rollenspiel Elemente ab und soll

dazu dienen, dass Spieler miteinander z.B. durch Handeln physikalisch interagieren. Speziell die Spielertypen Socializer werden hier angesprochen. Aber auch Explorer kommen hier zum Zug, da die Items bzw. Itemtypen nicht starr festgelegt sind und somit jede Region oder jeder Händler individuelle Items besitzen kann. Die Händler/Dienstleister selbst werden hier als nicht gewinnbringendes Spielelement aktiv. Die Entscheidung dass die Händler nicht selbst als POIs auftreten, auf die eine Flagge gesetzt werden kann hat zwei Überlegungen. Zum einen sollen die Händler nicht von Spielern "besetzt" werden, damit keine Assoziation zwischen "Besitz" der Händler stattfindet, zum anderen soll eine gewisse Modularisierung stattfinden. D.h. sofern sich die Bedingungen bzw. die Auswahl der POIs verändert, können die Händler unabhängig vom generierten Spielfeld am Spiel teilnehmen. Somit kann das Framework auch in einem anderen Zusammenhang verwendet werden und es kann darüber hinaus eine feine Selektion der teilnehmenden Händlern stattfinden. Die Integration eines entsprechenden Leaderboards ermöglicht den Spielern sich mit geografischen und punkte nahen Spielern zu messen. Dies ist vor allem für die Spielertypen Achiever und Killer relevant. Letztere wird darüber hinaus die Möglichkeit gegeben gezielt Spielern zu schaden, wenn sie explizit Flaggen eines bestimmten Spielers angreifen.

Für die Anforderungen des Geogameframeworks wurde entschieden, dass zunächst das Framework dazu eingesetzt werden soll um entsprechende Spielfelder zu generieren, unabhängig des darauf aufbauenden Spieles. Das Beispielspiel wird somit als Proof of Concept des darunterliegenden Frameworks agieren. Es soll ein entsprechendes Webframework verwendet werden, welches die Daten über eine entsprechende Schnittstelle dem Spiel zur Verfügung stellt und damit bestehende Technologien nutzt um den Implementierungsaufwand auf das Framework zu konzentrieren und nicht auf die Tools für die Konfiguration und Darstellung. Für die Basis der Geodaten wird auf OSM zurückgegriffen werden, und die damit verbundenen Lizenzanforderung der Verfügbarkeit der verwendeten Daten analog über eine Schnittstelle erfolgen. Die Daten selbst werden nicht über die OSM API sondern die Overpass API ausgelesen werden, da diese geospezifische Abfragen besser erlaubt. Für die Auswahl der Spielelemente wird ein Hybrid Ansatz gewählt. Bei diesem wird eine Teilselektion der Spielelemente ausgelagert. Die Idee ist es dabei in Echtzeit während des Spielverlaufs Spielfelder erzeugen zu können und gleichzeitig eine entsprechende Evaluation zu bieten. Hierzu wird demjenigen der ein Spiel durchführen möchte ein Tool an die Hand gegeben, welches OSM key-value Paare evaluiert für bestimmte Regionen. Durch die Festlegung auf ein spezielles Tag für die Selektion der Daten kann der Veranstalter wiederum durch eine einfache Konfiguration eines Parameters, sicherstellen, dass die Spielfelder entsprechend homogen über verschiedene Orte funktionieren. Im Framework selbst wird der entsprechende Tag dann für

die Extraktion der Spielelemente verwendet und wiederum intern durch die in Kapitel 4.1 beschriebene Transformation umgewandelt. Bei der Transformation wird jedoch das einfache arithmetischen Mittel der umfassenden Bounding Box gewählt, da dieses auch bei größeren Datenmengen zu einem zufrieden stellendem schnellen Ergebnis führt. Hierzu werden wie beschrieben Relation und Ways in virtuelle Nodes aufgelöst. Da Relationen selbst wiederum Relationen, Ways und Nodes enthalten können müssen diese entsprechend rekursiv aufgelöst werden. Hierzu werden Ways zu einer Liste von Nodes aufgelöst und entsprechend zu einem virtuellen Node aggregiert. Dies wurde durch entsprechende Tests evaluiert und basierend auf den Laufzeiten entschieden. Hierbei ist anzumerken, dass zwar die Ladezeit des Spielfeldes auf einem mobilen Client hauptsächlich das Laden der Maptiles ausmacht. Eine Verzögerung von mehr als 400ms werden vom Benutzer wahrgenommen und als störend empfunden [Gut+04]. Daher ist es unerlässlich eine schnelle Generierung der Spielfelder sicherzustellen. Eine Möglichkeit um dieses Problem zu umgehen wäre das Caching der Ergebnisse. Gegen dieses wurde sich jedoch bewusst entgegen entschieden. Aus dem Grund, dass dann für jedes Objekt welches von OSM ausgelesen wurde, eine entsprechende Persistierung in der lokalen Datenbank stattfinden müsste, auch wenn keinerlei Interaktion mit diesem stattfindet. Ein weiterer Grund ist die Tatsache, dass eventuelle Änderungen der Daten bei OSM mit einem entsprechenden Aktualisierungs-Mechanismus übertragen werden müssten. Das Framework selbst soll mittels Webbasiertes Framework erfolgen. Zwar reduziert sich dadurch der Einsatz möglicher Hardware Sensoren auf Seiten der Smartphones, aber der Entwicklungsaufwand um auf mehreren Plattformen präsent zu sein wird erheblich reduziert.

Für die Interaktion mit dem Händler ist vorgesehen, dass der Spieler geografisch in die Nähe kommt. Er soll die jeweiligen Läden der Händler explizit besuchen. Da durch die Nutzung einer webbasierten Lösung die Hardware Funktionen wie Bluetooth, NFC oder Kamera des Smartphones nicht zu Verfügung stehen, muss auf die GPS Daten zurückgegriffen werden. Hierbei stellt sich allerdings die Herausforderung, dass der GPS Empfang in Gebäuden je nachdem nur schlecht oder gar nicht vorhanden ist. Gleiches gilt für die mobile Datenverbindung. Daher bietet es sich als Option mit Coupon Code zu arbeiten. Der Spieler erhält diesen vom Händler, welchen er dann später wenn er wieder mobilen Datenempfang hat in das Spiel eingeben kann und entsprechende Items erhält. Zusätzlich funktioniert diese Methode auch Problem mit älteren Smartphones. Die Couponcodes kann der Händler gegen eine Gebühr vom Spielbetreiber erhalten. Vorstellbar wäre z.B. 100€ für 100 Interaktionen. Vergleicht man diese Preise mit den üblichen Kosten für Cost-Per-Click Werbung wie z.B. bei Google Adwords, so bietet sich dem Spielbetreiber hier eine

entsprechende kostengünstige und effektive Möglichkeit die Spielinfrastruktur und den operativen Betrieb zu refinanzieren [GSS10]. Für den Begriff Supermarkt im Raum Hamburg sind CPC von 1-2€ normal. Auch im Raum Bamberg fallen Kosten von 0,5€ an ². Im Gegenzug erhält der Händler physikalische Interaktion, die um vielen wertvoller ist, wie der Aufruf seiner Homepage. Da der direkte physikalische Kontakt des Spielers mit dem Point of Sales viel mehr zu einem Kauf führen wird wie der Besuch einer Homepage. Hier wird auch an der Problematik der Verknüpfung von Online und Offline bewusst aufgegriffen. Der offline Händler kann somit aktiv vom Internet profitieren und dem Internet affinen Nutzer hier einen Mehrwert bieten. Vorstellbar wäre, dass der Händler die Coupon Codes z.B. auf dem Kassenzetteln unterbringt, dann wäre es für den Händler möglich auch direkte Käufer zu belohnen. Für das Beispielspiel werden die Itemtypen nicht näher festgelegt. Testweise soll ein Itemtyp eingeführt werden, welcher die Maximale Aktionspunkte um 10 AP anhebt. Dies soll dem Spieler ein taktisches Vorgehen ermöglichen.

²Abruf 05.03.2014 - Google Adwords Keyword Planer

5. Umsetzung

5.1. Erläuterung des Softwaretechnischen Entwurfs

Für die Softwaretechnische Umsetzung wurden zunächst die Anforderungen an das Geogameframework in Kapitel 4.1 sowie die gewählte Lösung aus Kapitel 4.2 im Detail evaluiert. Für die Umsetzung des Entwurfs wurde zunächst ein Entwurf des Prozesses identifiziert, welcher die Geodaten aus OSM bis hin zur Darstellung im Beispielspiel darstellt. Eine Visualisierung ist in Abbildung 5.1 zu sehen.

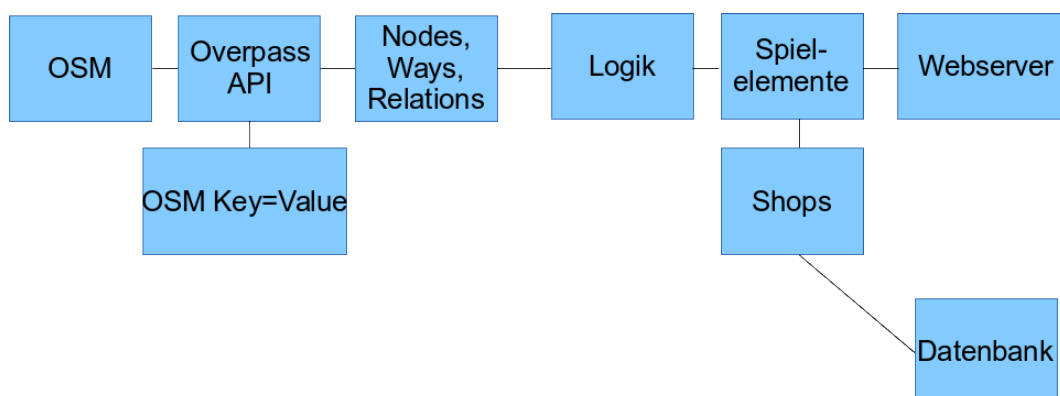


Abbildung 5.1.: Prozess: Von OSM zum Spielelement

OSM und Overpass API

Zunächst steht zu Beginn des Prozess als Datengrundlage Openstreetmaps. Die Daten werden allerdings nicht direkt von OSM über die OSM API abgerufen, sondern über Overpass. Das liegt daran, dass die OSM API selbst nur sehr rudimentäre Abfragen erlaubt. Stattdessen wird die Overpass API verwendet, da diese geografische Abfragen erlaubt [MTS13]. Diese Abfragen sind notwendig, um die zuvor deklarierte Anforderung, die Spielelemente basierend auf einem OSM key-value Paars zu bewerkstelligen. Darüber hinaus wird die Transformation der Relations und Ways in Nodes einfacher ermöglicht. Durch die Verwendung der Overpass QL-Abfrage Sprache (OQL) ist es möglich sich nicht nur die jeweiligen Relations, Ways und Nodes

eines tags zu erhalten, sondern auch zusätzlich alle rekursiv enthaltenen Elemente. Dies ermöglicht es den kompletten Datensatz mit einer Abfrage zu erhalten der für die spätere Transformation benötigt wird. Der Vorteil liegt darin, dass nicht mehrere Abfragen gestartet werden müssen und somit die Zeit bis alle Daten zur Verfügung stehen erheblich reduziert wird. Die Overpass API selbst bietet diverse Ausgabe Formate wie XML und JSON [Olb]. Für eine konkrete Umsetzung wurde sich bewusst für JSON entschieden, da zum einen die Performance bei der Verarbeitung von JSON Dokumenten beachtlich höher ist im Vergleich zu XML Dokumenten [Nur+09]. Zum anderen soll die Daten später als GeoJSON aufbereitet werden. Der Aufruf der OverpassAPI erfolgt mittels einfacher REST-Abfrage:

`http://overpass-api.de/api/interpreter?data=OQL_BEFEHL`

Über den Parameter "data" wird der jeweilige OQL Befehl abgesetzt. Für die Weiterverarbeitung der Daten wird im Anschluss das JSON Ergebnis vom Framework geparkt und weiterverarbeitet.

Transformations Logik

Die Transformation der Relations und Ways wird wie in Kapitel 4.2 umgesetzt. Eine Visualisierung ist in Abbildung 5.2 zu sehen.

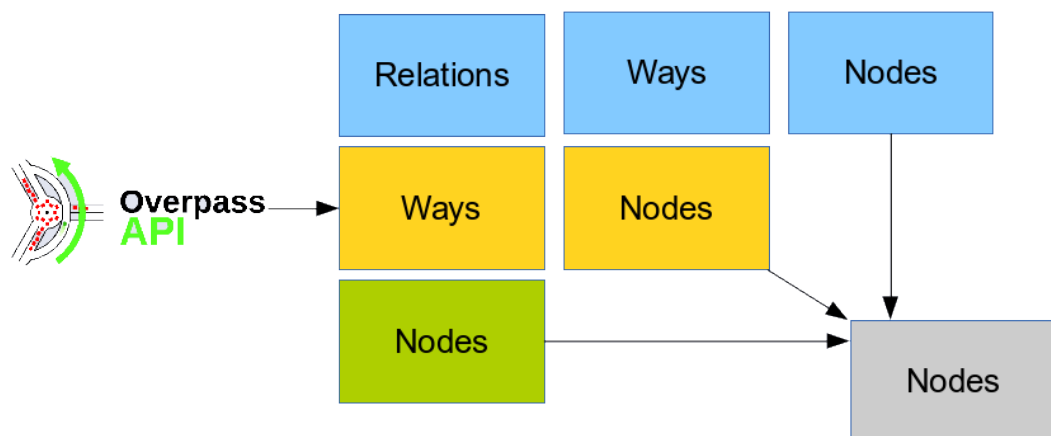


Abbildung 5.2.: Transformationsprozess: Relations, Ways, Nodes

Auf der linken Seite sind vertikal die Ausgangstypen angeordnet. Hierbei handelt es sich um die bereits angesprochenen Elemente Relations, Ways und Nodes. Im zweiten Schritt, nach dem Aufbereiten der Daten von der Overpass API wird wie folgt vorgegangen. Zunächst werden alle Relations behandelt, im Anschluss darauf die Ways und zum Schluss die Nodes. Die Idee dahinter ist es zunächst alle Ways

und Nodes zu identifizieren welche direkt einer Relation angehören und keine separaten Spielelemente darstellen. Die Vorgehensweise ist damit begründet, dass durch die Reduzierung der Anfrage auf ein einzelnen Request die Rückgabe alle Nodes enthält. Sowohl die Nodes einer Relation, als auch die eines Ways, welche selbst nicht eigenständig sind. Daher müssen die Elemente Ebene für Ebene wie in der Abbildung zu sehen, abgearbeitet werden.

Zunächst werden alle Relationen identifiziert. Für jede Relation werden nun die beinhalteten Ways und Nodes identifiziert. Diese wiederum werden dann als "geclaimed" markiert. Relations selbst werden nicht rekursiv aufgelöst, da das Ziel nicht ist möglichst wenige Spielelemente zu haben, sondern Relationen zu transformieren. Sollten Relations mehreren Unterrelations haben so sollen diese als Eigenständige Elemente betrachtet werden. Sofern dieser Ansatz sich als Nachteilig in der Evaluation herausstellt muss er entsprechend modifiziert werden. Für die jeweilige Iteration eines Relation Elements werden nun alle beteiligten Ways und Nodes zu einer Liste von Nodes zusammengefasst, Diese Liste wird wiederum durch das in Kapitel 4.2 beschriebene Verfahren einer Bounding Box dessen Mittelpunkt berechnet wird reduziert auf ein virtuelles Node, welches die Relation repräsentiert. Dieses virtuelle Node hat eine Koordinate, sowie eine ID welche eindeutig identifizierbar ist. Hierzu wird die Relations ID von OSM verwendet.

Im nächsten Schritt werden die Ways abgearbeitet. Bereits als "geclaimed" markierte Ways, die somit bereits in einer Relation enthalten sind, werden ignoriert. Alle anderen Ways werden entsprechend jeweils wiederum zu einer Liste von Nodes und anschließend zu einem virtuellen Node transformiert. Hierbei entsteht wieder eine Koordinate und als ID wird die OSM Way ID verwendet.

Die Elemente, welche direkt als Node zurückgegeben werden analog übernommen. Sie werden in Spielelemente, hier durch das graue Nodes Element symbolisiert, transferiert. Hierzu wird die Koordinate, sowie die OSM ID der Nodes übernommen.

Eine Problematik stellt sich allerdings noch in der eindeutigen Identifizierung der virtuellen Nodes. Da diese von unterschiedlichen Elementen (Relations, Ways, Nodes) abgeleitet wurden muss sichergestellt werden, dass anhand der ID eine eindeutige OSM Zuordnung möglich ist. Entweder es wird zusätzlich der abgeleitete Typ des Spielelements explizit gespeichert oder aber, man transcodiert die Information mit in die ID. Wenn man sich zunächst die OSM IDs für Nodes anschaut, stellt man fest, dass diese die 32bit signed Integer Grenze überschritten haben (Februar 2013). Im OSM Wiki wird daher empfohlen den Datentyp long zu verwenden, welcher in den Standard-Implementierungen ein 64bit Wert ist [Ope13a]. Eine Methode um den Typ des Spielelements zu übertragen könnte mit einer Bitmaske der ID funktionieren. Hierzu könnte man die den 2. und 3. Bit Bits des 64bit Wertes verwenden. Der 1. it

wird nicht verwendet um negative Zahlen weiterhin zu erlauben. Der 2. Bit würde als Identifikator für Relations dienen und der 3. für Ways als Ursprungselement. Ein Beispiel für den transformierten Way mit der OSM ID 1 ist in Abbildung 5.3 zu sehen.

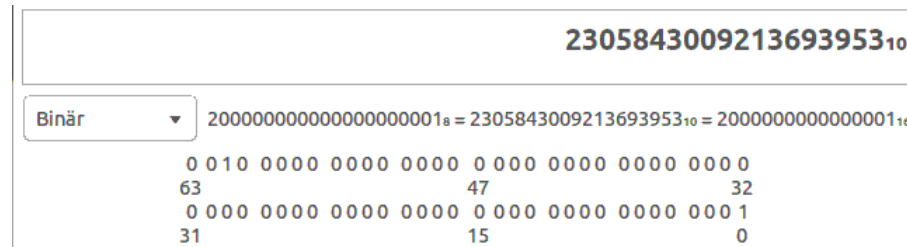


Abbildung 5.3.: OSM ID: Bitmaskenkodierung im 64bit long Wert

Wenn man die Reduzierung von 63bit (signed) auf 61bit(signed) mit der aktuellen Mapping Geschwindigkeit bei OSM vergleicht, so lässt sich feststellen, dass die Reduzierung um 2 bit aus Paradigmensicht zwar unsauber erscheint, aber eine Überschreitung der ID 2^{61} in ferner Zukunft liegt. Zudem könnte im Bedarfsfall wenn OSM auf 128bit IDs umsteigt die Bitmaske entsprechend angepasst werden.

Händler Integration

Nachdem die virtuellen Nodes für die Spielelemente alle erstellt wurden, müssen die Elemente um die Händlern ergänzt werden. In Kapitel 4.2 wurde erläutert, dass die Händler nicht als aktives Spielzielelement verwendet werden sollen, sondern eine Integration der Händler und Dienstleister im Spiel als Repräsentation ihrer Selbst. Die Händler fungieren in diesem Zusammenhang als Anbieter für Items und andere Gegenstände. Für die spätere Darstellung auf der Karte müssen diese ebenfalls mit einer Koordinate versehen werden und separat behandelt werden. D.h. die Spielpunkte der Händler kommen stattdessen aus einer lokalen Datenbank. Dabei wird bewusst auf die Verwendung von OSM Als Basis verzichtet. Zwar könnte man in einer erweiterten Version des Frameworks dem Nutzer unterstützen mit automatischen Vorschlägen basierend auf OSM, jedoch ist dies in der Grundfunktion nicht notwendig. Hier soll der Händler sich beliebig frei auf der Karte Positionieren können und entsprechende Parameter seines virtuellen Geschäfts festlegen. Im Anschluss soll er entsprechende Items in seinem virtuellen Shop hinterlegen können. Da die Items eventuell vom Händler gegen einen Betrag vom Spielleiter erkauft werden, hat der Spielleiter ein gewisses Interesse, dass er zum einen die Items im Spiel möglichst an die Spieler bringt, während er auf der anderen Seite das Spiel nicht in einen nicht balancierten Zustand bringt. D.h. dass eventuell zu viele Spieler durch die Nähe eines Händlers mit einem sehr nützlichen Item bevorzugt werden. Daher sollte es im Ermessen des

Spielleiters liegen, dass dieser die Art und Verwendung der Items selbst definiert und entsprechende Vorschläge für die Händler parat hat. Dass der Händler selbst sich mit Spielmechaniken und Balancing auseinander setzt ist höchst unwahrscheinlich und würde auch nur zu extra Koordinationsaufwand führen. Daher ist es am sinnvollsten dem Händler gewisse Items mit standard Eigenschaften anzubieten, von denen er einen Typ selbst auswählt und eine entsprechende Menge seinem virtuellen Shop zuordnet. Für die Itemtypen kann der Spielleiter aller Voraussicht nach auf gewisse Grunderfahrungen zurückgreifen. Darüber hinaus sollte das Framework ihm zu einem späteren Zeitpunkt in einer erweiterten Ausbauphase auch entsprechendes Feedback über die Verwendung und Nutzung der Items aufzeigen und der Spielleiter somit aufgrund dieser Information Rückschlüsse auf das Balancing machen kann.

Persistenz

Ein wichtiger Aspekt des Frameworks stellt die Persistenz dar. Es müssen die Spielelemente, sowie der Spielzustand selbst gespeichert werden. Die Daten für die Spielelemente stammen aus OSM und von der lokalen Datenbank. Da die OSM Daten entsprechend transformiert werden stellt sich die Frage ob dieser Prozess beschleunigt werden kann, wenn die Daten entsprechend in der Datenbank lokal zwischengespeichert werden. In Kapitel 4.2 ist bereits auf diesen Aspekt eingegangen worden. Die Problematik die sich durch eine Zwischenspeicherung stellt ist zum einen die Aktualisierung der Daten. Die lokalen Daten müssten mit einem Zeitstempel versehen werden und gehalten werden, bis diese "verfallen". Darüber hinaus müssten diese nach dem Verfallszeitpunkt entsprechend gelöscht oder aktualisiert werden oder belassen werden. Ein Caching kann hier Sinn sofern das Framework für ein Spiel mit einer kritischen Masse an Spielern verwendet wird. Allerdings ist eine Evaluation und Untersuchung des Frameworks auf Hochskalierbarkeit nicht Bestandteil der ersten Ausbaustufe. Ein weiterer Aspekt stellt die Datenmenge dar. Sofern im Framework alle jemals abgefragten OSM Daten in virtuellen Nodes/Spielelementen in der Datenbank hinterlegt werden ohne dass diese eine Zustandsveränderung erfahren haben ist dies zwar Modelltechnisch korrekt, allerdings aus Performance und Platzgründen nicht zu empfehlen. Speziell in der Hinsicht, dass das Framework dem Spielleiter so wenig Aufwand wie möglich machen soll, sollte verhindert werden, dass der Spielleiter sich um Datenbank und Speicherplatz Probleme kümmern muss. Ein gutes Beispiel hierfür stellt auch das OSM Projekt selbst dar. Die bekannte Kartendarstellung verwendet zur Anzeige entsprechende Tiles. Diese Tiles werden auf Basis der OSM Daten gerendert. Ein erster Ansatz wäre es alle Tiles entsprechen dzu Rendern. Allerdings dauert der Renderprozess dann Tage und Aktualisierungen auf der Karte würden immer nur mit mehreren Tagen Verzögerung angezeigt. Hinzu

kommt die Tatsache, dass nur ein Bruchteil der Kartendaten auch tatsächlich angeschaut wird ($<2\%$ - OSM Quelle?). Daher werden die Karten-Bilddaten in Echtzeit gerendert und nach einer gewissen Zeit wieder verworfen. Daher ist es auch hier nicht sinnvoll alle Daten zu speichern sondern nur die Spielelemente mit denen ein Spieler aktiv interagiert hat. Die Daten der Händler werden separat gespeichert. Sie befinden sich ebenfalls in einer Datenbank, besitzen allerdings im Vergleich zu den Spielelementen eine Persistenz unabhängig von ihrer Interaktion.

Schnittstellen

Ein Framework benötigt entsprechende Schnittstellen über die es Funktionen und Daten nach Außen hin zur Verfügung stellt. Zunächst muss geklärt werden, wie die Daten verwendet werden sollen. Für das Beispiel-Spiel ist die Darstellung der Karte über eine Website vorgesehen. Unabhängig von der später verwendeten Technologie müssen in diesem Fall sowohl die Spielelemente, als auch die Händlerdaten vom Framework zur Verfügung gestellt werden.

Zur Verdeutlichung wird an dieser Stelle die Entscheidung für eine Technologie in Abbildung 5.4 auf das nachfolgende Kapitel verlegt.

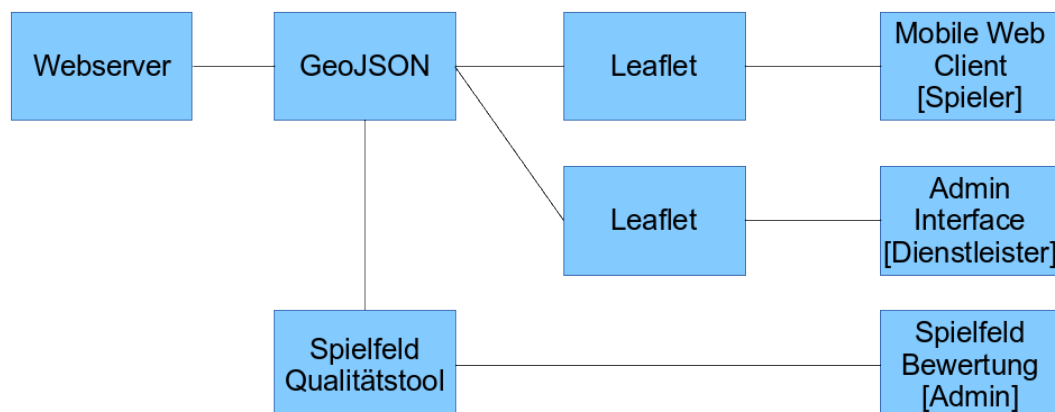


Abbildung 5.4.: Visualisierte Schnittstellen des Frameworks

Dadurch, dass das Framework die Daten von OSM/Overpass als JSON erhält und das Format bestens geeignet ist für den Austausch, da eine Vielzahl der aktuellen Frameworks und Tools dieses unterstützen ist die Wahl auf den Export der Spielelemente und Händlerdaten auf GeoJSON gefallen. Während das offene Format WKT für die reine Repräsentation von Geodaten dient [Sto03], bietet das GeoJSON Format zusätzlich die Möglichkeit entsprechende Properties an ein Geo Objekt zu speichern [But+08]. Über diese kann wiederum das Framework Informationen wie z.B. die codierte OSM ID und Informationen zum Spielelement übertragen.

Generell werden die Informationen des Frameworks für drei verschiedene Module benötigt. Zunächst einmal gibt es den Spielclient bzw. dessen Oberfläche. Dieser benötigt die Daten für das Staging des Spiels selbst. Über die Spieloberfläche interagiert der Spieler mit dem Spiel. Er sieht die aktuelle Karte, sowie die darauf platzierten Objekte. In diesem Fall sind dies die Spielelemente sowie die einzelnen Händler. Die Spielelemente selbst stellen im Beispielspiel die sogenannten Prestige Flaggen dar. Ein weiteres Modul stellt die Adminstartionsoberfläche dar. Diese dient dazu dem Spielleiter sowie dem jeweiligen Händler eine Konfiguration des Spiels vorzunehmen. Im Detail kann der Spielleiter die entsprechenden Spielitems definieren, neue Händler anlegen und diese auf der Karte positionieren. Der Händler kann seine Metadaten pflegen und entsprechende Items, welche er in seinem virtuellen Laden anbieten möchte selektieren und anbieten. Bei Fehlern oder Aktualisierungen kann der jeweilige Administrator (Spielleiter oder Händler) diese problemlos anpassen. Für die Positionierung der virtuellen Läden auf der Karte wird ebenfalls analog zum Spielfeld eine GeoJSON Schnittstelle zum Einsatz kommen. Das letzte Modul stellt die Evaluierung der Spielfelder dar. Hier wird dem Spielleiter durch die Verwendung der gleichen Schnittstelle, wie für die Aufbereitung des Spielfeldes, die Möglichkeit geboten Daten in ein entsprechendes Evaluationstool mit entsprechendem Algorithmus zu importieren. Dieses Tool wird entsprechend in Kapitel 6.2 in vorgestellt. Über dieses wird das ideale Key-Value Paar für die OSM Tag Selektion der Daten evaluiert. Dies soll dem Spielleiter die Möglichkeit einer Objektiven Bewertung der Spielfelder ermöglichen.

Darstellung

Da das Framework für ortsbezogene Spiele verwendet werden soll, ist es daher unerlässlich dem Spieler, sowie den Administratoren eine entsprechende Visualisierung zu bieten. Zunächst gibt es das Spielfeld. Das Spielfeld verwendet im Hintergrund Kartenmaterial von OSM, auf das die einzelnen Elemente platziert werden. Die Karte selbst wird initial auf die (GPS-)Koordinaten der Spielerposition zentriert. Dadurch ist es dem Spieler möglich sich direkt von seiner Position aus zu Orientieren. Die auf der Karte eingezeichneten Elemente wie Flaggen und Händler kann der Spieler somit problemlos identifizieren und zu Fuß aufsuchen. In Abbildung 5.5 ist ein Mockup der Kartenoberfläche zu sehen. Je nach Flaggestatus werden die Flaggen unterschiedlich farbig dargestellt. Das Ziel ist es Neutrale Flaggen grau, feindliche Rot und eigene grün darzustellen. Damit erhält der Spieler automatisch einen Überblick über die Situation und kann auch durch das Herauszoomen der Karte seine weiteren Spielzüge entsprechend planen. Über die Elemente hinaus werden dem Spieler sein aktueller Punktestand. Dieser wird im Beispielspiel durch das einfache Addieren der

zigartige Items zu enthalten und sicherzustellen, dass ein Item jeweils auch immer als solches im System behandelt wird. Über die Liste gibt es die Option die bestehenden Items zu modifizieren oder zu entfernen. Neue Items können über eine Schaltfläche entsprechend angelegt werden. Dabei wird dem Benutzer eine entsprechende Oberfläche präsentiert und über Metadaten das Item genauer spezifiziert. Beispielsweise der Name, Itemtyp und der dafür zu zahlende Preis.

Möchte der Benutzer dahingegen Händler pflegen, so erhält er zunächst analog zu den Items eine Übersicht der einzelnen Händler. Diese kann er analog zu den Items entsprechend modifizieren und löschen. Auch das Anlegen eines neuen Händlers ist analog zu den Items. Der Unterschied liegt jedoch darin, dass es sich bei den Händlern nicht um einfache Formularfelder handelt, sondern auch zusätzlich eine Georepräsentation stattfinden muss. D.h. es muss die Position des Händlers auf einer Karte erfolgen. Hierzu wird eine Art "Picker" verwendet. Auf einer OSM Karte soll der Benutzer die Position des Händlers definieren. Für eine Korrektur der Position reicht es aus, wenn der Benutzer einfach den Marker auf der Karte mit der Maus ergreift und per drag n drop auf seine neue Position bewegt.

Technischer Entwurf

Nachdem die Anforderungen beschrieben wurden, muss der Softwaretechnische Entwurf konkretisiert werden. Um die Software entsprechend abzubilden zu können muss zunächst ein Modell entworfen werden, welche die Software abbildet. Die Prozesse des Frameworks wurden bereits in Abbildung 5.1 sowie 5.4 visualisiert.

Zu Beginn einer Systementwicklung müssen die Usecases und die beteiligten Akteure identifiziert werden. Hierfür wird zu Beginn überlegt, welche Personen Zugang zum System haben und welche Aufgaben diese am System erfüllen werden. Dadurch wird sichergestellt, dass alle Aspekte behandelt werden, nicht nur die im initialen Lösungsansatz. Diese sind wichtig für den späteren Entwurf des Systems, sowie deren Umsetzung. Die Usecases lassen sich als Diagramm wie in Abbildung 5.6 sichtbar definieren.

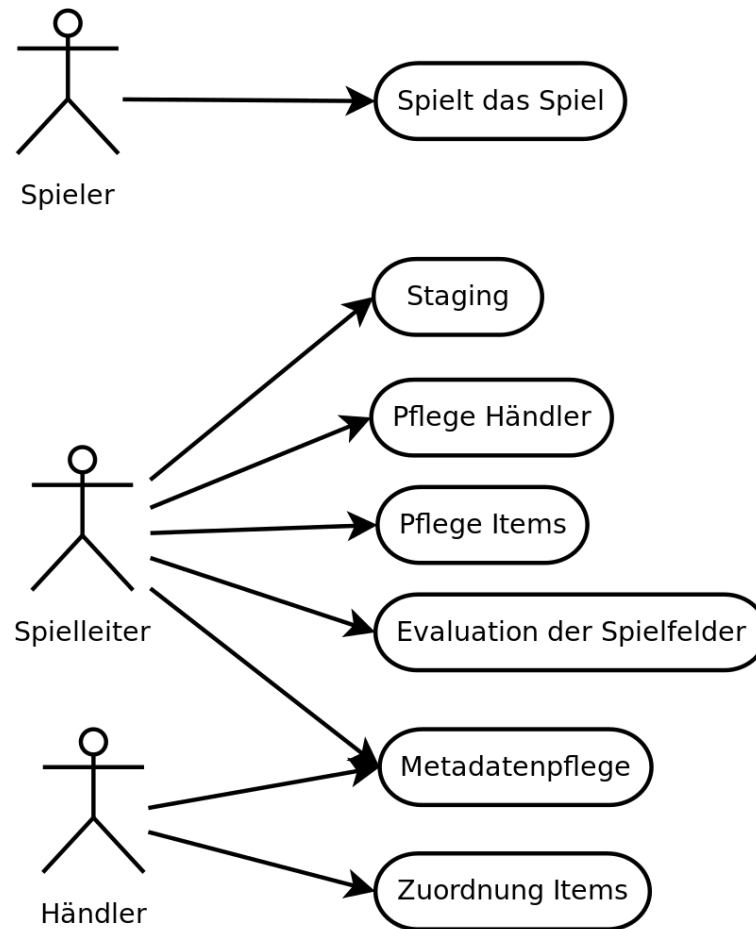


Abbildung 5.6.: Usecase Diagramm

Zunächst lässt sich feststellen, dass es drei Akteure gibt. Diese decken sich soweit mit dem Lösungsansatz. Der Spieler hat das Ziel das Spiel zu spielen. Der Spielleiter hingegen sieht es vor entsprechend das Staging des Spiels mit dem Framework zu bewerkstelligen. Hier muss er die Händler als auch Items pflegen. Darüber hinaus teilt er sich mit dem Händler die Metadatenpflege, da je nach Situation der Spielleiter mehr oder weniger in die individuelle Pflege der Händlerdaten eingebunden ist. Der Händler möchte zusätzlich seine Items, die er vom Spielleiter zugewiesen bekommen hat entsprechend auf seine virtuellen Läden verteilen. In der Grundfunktion des Frameworks sind nur diese Benutzer vorgesehen. In einem Ausbau des Frameworks macht es Sinn explizite Nutzerrollen einzuführen um eine detailliertere Rechtezuweisung zu ermöglichen.

Durch die Verwendung einer objektorientierten Programmiersprache ist es notwendig entsprechende Klassen zu definieren. Diese dienen dazu, Objekte abzuleiten und die jeweiligen Methoden von diesen zu nutzen. Es muss sichergestellt werden, dass

die Beziehungen zwischen den Klassen modular sind, damit ein einfacher Zugriff und eine Austauschbarkeit gegeben ist. Um ein Einblick in die Struktur des Frameworks zu erhalten wird in diesem Abschnitt kurz auf die wichtigsten Aspekte eingegangen.

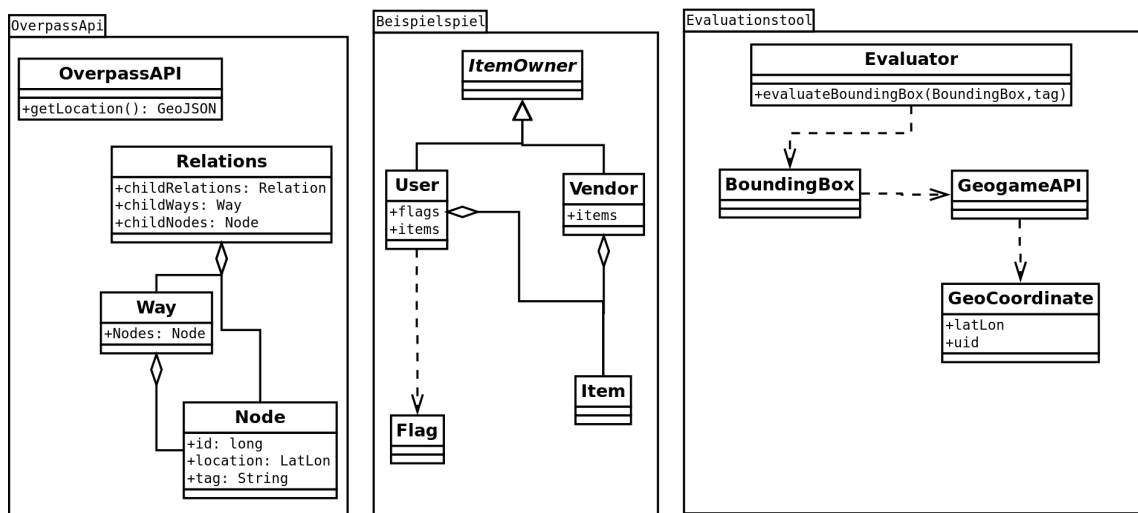


Abbildung 5.7.: Vereinfachtes Klassen Diagramm

In Abbildung 5.7 ist ein vereinfachtes Klassendiagramm des Frameworks zu sehen. Das Framework als solches besteht aus drei Modulen. Zunächst gibt es den Bereich *OverpassApi*. Hierbei handelt es sich nicht um die OverpassApi selbst, da Overpass selbst nur eine Webschnittstelle ist, sondern um die Implementierung einer entsprechenden Schnittstelle im Framework selbst, welches das Ergebnis der OverpassApi im Web transferiert auf die einzelnen Elemente Relations, Ways und Nodes. Diese werden wiederum durch den bereits beschriebenen Ansatz entsprechend transformiert in dem eine umcodierung in virtuelle Nodes erfolgt. Im gleichen Zug wird überprüft, ob es persistierte Daten für die virtuellen Nodes in der lokalen Datenbank gibt. Ist dem der Fall, so werden die virtuellen Nodes um die entsprechenden Properties ergänzt und anschließend als JSON bzw. GeoJSON gerendert. Im Anschluss werden diese über die Webschnittstelle des eigenen Frameworks ausgegeben. Für die Händler gibt es eine separate Schnittstelle im Framework, welche analog zur Overpass Implementierung fungiert, aber hierbei nicht die Daten von extern (OSM/Overpass) bezieht, sondern direkt in der fertigen Form aus der lokalen Datenbank auslesen kann. Hierbei ist zudem auch keine entsprechende Transformation notwendig, sondern diese liegen bereits als fertige Nodes vor. Dies ist möglich, da jeder Händler nur als einziger Punkt im Spiel repräsentiert wird.

Das nächste Modul ist das *Beispielspiel*. Das Beispielspiel selbst wurde einfach gehalten, da es zum einen als Proof of Concept des Frameworks dienen soll, zum anderen der Fokus auf die Verteilung der Spielelemente auf dem Spielbrett untersucht werden

soll. Generell gibt es 4 Objekte. *Spieler*, *Händler*, *Flaggen* und *Items*. Spieler und Händler stehen in einer polymorphen Verbindung zu Item. Ausgelöst, leiten beide Klassen von einer abstrakten Klasse *ItemOwner* ab, welche entsprechende Methoden vorhält die für die Interaktion als ItemOwner essentiell sind. Beispielsweise der Verkauf oder die Benutzung eines Items. Ein Item kann nur einem Spieler oder einem Händler angehören, niemals aber beiden gleichzeitig. Des weiteren gibt es auch die Möglichkeit, dass ein Item ohne ItemOwner existiert. Ein Beispiel könnte es sein, dass der Spielleiter für ein Event bestimmte Items auf der Karte ablegt oder lokale zwei Spieler Items manuell tauschen möchten.

Das letzte Modul stellt das sogenannte *Evaluationstool* dar. Mit diesem sollen die erzeugten Spielfelder analysiert und bewertet werden. Eine genauere Erläuterung ist in Kapitel 6.2 zu finden. Das Evaluationstool greift zunächst auf die GeoJSON Schnittstelle des Frameworks zu. Dieses liefert bei entsprechender Abfrage mittels Bounding Box und entsprechendem OSM Tags die Spielelemente zurück. Die Boundingbox wird auf Basis einer initialen Koordinate berechnet. Somit kann der Spielleiter einfach zwei Koordinaten definieren zu denen er gerne eine entsprechende Auswertung interessanter Tags erhalten möchte. Das Evaluationstool startet im Anschluss den Vorgang und wandelt die Spielelemente in vereinfachte Objekte mit ID und Geo-Koordinaten um. Diese wiederum werden der jeweiligen Evaluationsmethode als Liste übergeben und deren Rückgabewert beschreibt die Kombination aus Boundingbox und OSM Tag.

Datenbank

Aufgrund der vorherigen Analysen wird normalerweise ein Entity Relationship Model (ERM) erstellt. Allerdings soll der Einsatz eines Webframeworks erfolgen welches ein mindestens ein Objektrelationales Mapping unterstützt und somit die Manuelle Erstellung der Datenbankstruktur nicht mehr vorgesehen ist. Dies wird unter der Annahme gemacht, dass das Framework später im Produktiv Betrieb für die Datenspeicherung optional auf eine NoSQL Lösung umgestellt werden kann. Diese bieten eine bessere Performance bei einfachen Abfragen. Dies geschieht auch vor dem Hintergrund den in der Literatur häufig kritisierten Mismatch zwischen Objektorientierter Programmierung und der Verwendung von Relationalen Datenbanken um Objekte welche von Klassen abgeleitet wurden zu speichern [Cat91].

Die geringe Verbreitung der objektorientierten Datenbanken liegt darin, dass nicht versucht wurde die bestehenden Datenbanken zu ersetzen. Viel mehr sind die bereits existierenden Datenbanken in Unternehmen nicht in neue objektorientierte Datenbanken transferiert worden. Der Grund hierfür liegt in den historisch gewachsenen Applikationen und Datenbanken, dessen Umstellung einen sehr hohen Aufwand und

Kosten darstellen würde [Bur94].

Ein weiterer Aspekt ist die Tatsache, dass die objektorientierten Datenbanken nicht in jedem Aspekt besser sind als relationale Datenbanken. Es gibt Situationen in denen haben Objektorientierte Datenbank Management Systeme klare Vorteile gegenüber den relationalen Datenbanken besitzen. Die Objektorientierten Datenbanken spielen Ihre Vorteile speziell bei der Abbildung von Beziehungen von Objekten untereinander aus. Nicht nur bei Vererbungen sondern auch wenn die Objekte diverse Methoden besitzen. Bei einer Sequentiellen Abfrage mehrerer Datensätze sind jedoch die relationalen im Vorteil [VZKB06]. Hier ist es notwendig den Zweck der zu speichernden Daten bzw. deren Verwendung zu untersuchen. Je nachdem kann der Einsatz von objektorientierten Datenbanken oder relationalen Datenbanken Sinn machen.

Vergleicht man die Verbreitung von objektorientierten Datenbanken in gewissen Einsatzgebieten, so lässt sich feststellen, dass diese trotz der in Summe geringen Verbreitung ein berechtigtes Dasein haben. Speziell in Geoinformationssystemen spielen objektorientierte Datenbank Management Systeme eine wichtige Rolle,[Bri05] da Sie deutlich einfacher ermöglichen komplexe Verbindungen zwischen den unterschiedlichen Objekten herzustellen und eine Veränderung dieser Beziehungen in einem Bruchteil der Zeit ermöglichen, welcher eine relationale Datenbank dafür aufwenden müsste. Die umfangreiche Literatur zum Thema Geoinformationssysteme gibt einen Ausblick für Möglichkeiten sich durch die Nutzung dieser ergeben können.

Weitere Aspekte

Ein weiterer Aspekt stellt die Optimierung der Anfragen an die Overpass Api Schnittstelle des Frameworks seitens des Spielfeldes dar. In der Grundvariante des Frameworks, stellt das Spielfeld jeweils eine Anfrage als Bounding Box an die Schnittstelle um die aktuellen Spielelemente für den jeweiligen Kartenausschnitt zu erhalten. Eine Optimierung dessen wäre es mit dynamischen erweiterten gecachten Bounding Boxen zu arbeiten. Das bedeutet, dass bei einer Abfrage zunächst die Boundingbox am Rand um jeweils 30% erweitert wird. Die zusätzlichen abgefragten Spielelemente werden allerdings nicht direkt angezeigt. Bewegt der Spieler nun das Spielfeld wird überprüft ob der aktuelle Kartenausschnitt sich noch innerhalb der dynamisch erweiterten Bounding Box befindet. Ist dies der Fall, so spart sich der das Spiel einen zweiten Request. Dies hat zweierlei Vorteile der erste stellt eine Reduzierung der HTTP Request auf der Clientseite dar. Speziell bei ortsbezogenen Spielen ist der Empfang beim Spielen öfters eingeschränkt und nur GPRS oder EDGE seitens Mobilfunkanbieter verfügbar. Durch die Reduzierung der Requests wird die Spielperformance verbessert. Ein anderer Aspekt ist die Reduzierung der Anfragen auf der Server Seite. Es verbessert deutlich die Skalierbarkeit und ermöglicht somit mehr

Spieler mit einem einzigen Server bedienen zu können.

5.2. Bewertung der Technologien und Werkzeuge

Nachdem der Softwaretechnische Entwurf erstellt wurde, muss untersucht werden welche Technologien und Werkzeuge für die Umsetzung des Frameworks am besten geeignet sind. Zunächst stellt sich die Frage in welcher Programmiersprache das Framework umgesetzt werden soll. Diese Frage lässt sich anhand der Anforderungen eingrenzen. Zunächst muss eine Website erstellt werden, welche ein Staging des Beispielspiel erlaubt und gleichzeitig die Spieldaten über eine Webschnittstelle exportieren kann. Hiermit lässt die Auswahl auf entsprechende Sprachen reduzieren, welche eine Erstellung dynamischer Webseiten erlauben. Die nachfolgende Aufzählung nennt die aktuell verbreitetsten Sprachen [WWW14; DD08]:

- PHP (81.8%)
- ASP.NET (17.8%)
- Java (2.7%)
- ColdFusion (0.8%)
- Perl (0.6%)
- Ruby (0.5%)
- Python (0.2%)
- JavaScript (0.1%)

Das Framework kann auf allen der genannten Sprachen umgesetzt werden. Das Ziel ist es aber zum bevorzugt auf OpenSource Technologien zurückzugreifen, da diese ohne Lizenzkosten sind und meist guter Dokumentation bieten können. Damit fallen ASP.NET und ColdFusion aus der engeren Auswahl. Ein weiteres Kriterium stellt die Verwendung eines Web Frameworks dar. Ziel ist es das Framework zu implementieren und den Aufwand für andere Aspekte auf einem Minimum zu halten. Darüber hinaus reduzieren Web Frameworks auch die Gefahren im Hinblick auf Sicherheit [LE07] und reduzieren den Implementierungsaufwand [Sch+01]. Für die restlichen Sprachen gibt es eine Vielzahl an entsprechender Web Frameworks [Wei+11]. Eine komplette Analyse über alle Sprachen hinweg, sowie deren Vor- und Nachteile ist nicht Bestandteil der Arbeit, daher wird ein Framework ausgewählt, welches dem Autor vertraut ist und eine möglichst effiziente und schnelle Umsetzung des Frameworks ermöglicht.

In diesem Fall wurde sich daher für 'Ruby on Rails' entschieden, einem Webframework welches auf der Sprache Ruby basiert. Ruby bietet darüber hinaus eine Paketverwaltung analog zu den bekannten Paketverwaltungssystemen in etablierten Linux Distributionen [BK07]. Durch eine Versionskoppelung der Pakete kann sichergestellt werden, dass ein Projekt problemlos auf allen Rechnern problemlos funktioniert, da fehlende Bibliotheken mit einem Befehl problemlos nachgeladen werden. Ein weiterer Vorteil ist die feste Verwendung des Model View Controller-Patterns [TH06]. Durch dieses wird sichergestellt, dass das Modell komplett unabhängig von der Darstellung ist. Dies ist auch für das Game Framework wichtig. Speziell um konkrete Funktionen über Schnittstellen über und Dialoge zur Verfügung zu stellen. Allerdings ist zu beachten, dass es sich in Webframeworks, in diesem Fall auch bei Rails um eine abgewandelte Form des MVC namens *Model2* handelt [Qiu02]. Der Controller wird mit dem Seitenaufruf angestoßen und interagiert mit dem Model. Im Anschluss verwendet der View die Ergebnisse/Daten des Controllers und rendert entsprechend die Webseite.

Nachdem die Wahl des Webframeworks auf Ruby on Rails gefallen ist wurden entsprechende Bibliotheken für die Verwendung gesucht, welche Funktionen die für das Framework benötigt werden. Die Vorteile fertiger Bibliotheken liegen darin, dass der Entwickler selbst nicht nur Zeit bei der Entwicklung spart, sondern auch durch die Reduzierung seines Codeumfangs auch eine Reduzierung von möglichen Fehlerquellen erreicht.

Modere Bibliotheken wie JQuery für JavaScript und Compass für Sass bzw. CSS werden mit Rails direkt unterstützt. Auch die Unterstützung von Coffee Script und Sass sind bereits integriert.

Das Game Framework muss auf auf spatiale Operationen zurückgreifen. Hierfür wird auf das Gem 'rgeo' zurückgegriffen werden. Es handelt sich dabei um eine weit verbreitete Bibliothek die nicht nur mit geografischen Objekten umgehen kann, sondern auch direkt eine Anbindung von GIS-Datenbanken wie PostGIS, Spatialite und MySQL-Spatial erlaubt.¹

Ein weiterer Aspekt stellt die Kartendarstellung dar. Für die Kartendarstellung selbst gibt es mehrere Ansätze. Die am weitesten verbreiteten sind die Google Maps API, Openlayers und leaflet [Der13]. Mit der Google Maps API kann man zwar die Openstreetmap Tiles als Layer laden, jedoch sind der kommerziellen Nutzung gewisse Einschränkungen unterworfen und der Quellcode nicht frei verfügbar. Da sich ein potentieller Spielleiter nicht mit den rechtlichen Problematiken und Lizenzvereinbarungen auseinander setzen sollte sollte diese vorzugsweise vermieden werden.

¹<http://dazuma.github.io/rgeo/>

Openlayers hat im Vergleich zu Leaflet eine längere Versionsgeschichte [Ohl14]. Und bietet deutlich mehr Funktionen. Allerdings ist OpenLayers mit 800KB deutlich größer wie Leaflet mit 120KB. Da es das Ziel ist, das Framework speziell auch im Zusammenhang mit Smartphones zu nutzen ist es wichtig, dass nicht nur die Dateigröße minimal ist, sondern auch eine möglichst gute Funktionsweise auf Smartphones sichergestellt wird. Hier ist Leaflet deutlich moderner und besser angepasst. Beide Bibliotheken unterstützen GeoJSON und ermöglichen somit die einfache Einbindung von Geo-Objekten. Die Entscheidung aufgrund der Anforderungen und ausreichenden Reife auf Leaflet. Auch für dieses gibt es bereits für Ruby ein entsprechendes Gem, welches Leaflet direkt für Rails integriert 'leaflet-rails'.

Neben der Kartendarstellung ist es auch wichtig, alle Spieler einzeln zu identifizieren und für jeden Spieler eine eigene SSession zu haben. Für die Benutzerverwaltung gibt es für Rails ebenfalls bereits fertige Lösungen. Diese kann man entsprechend für seine Bedürfnisse erweitern, muss trotz allem nicht komplett die Logik und Datenbankzugriffe für das Erstellen, Anlegen und Ändern von Benutzerdaten kümmern. Auch Funktionen, wie das zurücksetzen eines Passworts sind bereits integriert. An dieser Stelle wurde sich für das Gem 'Clearance' entschieden, da dies ausgereift ist und eine einfache Anpassung der Seitenbenutzer um zusätzliche Attribute ermöglicht, welche im Zuge des Frameworks entsprechend hinterlegt werden sollen.

Für die Schnittstellen welche das Gameframework anbieten soll, werden keine explizite Bibliotheken benötigt. Rails bietet automatisch die Möglichkeit für bestimmte Routen die passenden Dateiformate zu hinterlegen. Bei Routen handelt es sich bei Rails um Seitenpfade. Durch das Anlegen der Route '/flag/show/id' wird nach dem Aufruf des FlagControllers mit der Methode show die View show aufgerufen. Durch die Defaulteinstellung werden automatisch zuerst die html Files gerendert sofern der Request nichts anderes Fordert. Der View show.html.erb erhält somit nach dem Aufruf der show Methode ein entsprechendes Objekt mit der angegebenen Id aus der Datenbank zurück. Möchte man allerdings das Element nicht als Webseite darstellen sondern als JSON Objekt, so legt man lediglich eine entsprechende show.json.erb Datei an und kann direkt über die vorhandene JSON Bibliothek das Objekt entsprechend als JSON serialisieren. Hier zeigt sich die Flexibilität und Einfachheit die sich durch die Kombination von Ruby und Rails ergibt. Dies macht das Webframework ideal für die Nutzung mit dem zu erstellendem Game Framework.

Für die Erstellung des Quellcodes kommen entsprechende IDE zum Einsatz sowie eine Softwareversionskontrolle. Dies ist hilfreich um durch die Verwendung von Code Snippets sowie einer visualisierte Fehler-Erkennung und Lösung sicherzustellen. Für das Schreiben des Java-Quellcodes des Evaluationstools wurde auf Eclipse zurückgegriffen. Eclipse ist ein bewährtes Tool und dem Autor bestens vertraut und bietet auch eine ausreichende Modularität durch die Installation von Erweiterungen

über eine integrierte Verwaltung. Für die Entwicklung des Ruby on Rails-Codes wurde der grafische Text-Editor Geany verwendet. Da ein reiner Texteditor mit Syntax Highlighting und Code Completion keine Debugging Möglichkeit bietet wurde auf spezielle Rails Gems zurückgegriffen. Zunächst einmal wurde das bewerte 'Pry' Gem verwendet um ein einfaches Binding an entsprechender Code Stelle zu ermöglichen. Eine Parallele Rails Console ermöglicht das Überprüfen von entsprechender Active Record Abfragen, wie z.B. das Erfassen aller Punkte eines Spielers für die Status Übersicht. Für das direkte Debuggen von Fehlern zur Laufzeit wurde das Gem 'better_errors' in Verbindung mit 'binding_of_caller' verwendet. Dies ermöglicht es quasi Analog zum Debugmodus in Eclipse direkt an der Stelle an der ein unbehandelter Fehler auftritt in den Code einzusteigen. Das bedeutet es kann nicht nur einfach untersucht werden, an welcher Stelle und mit welchen Werten der Fehler auftritt, sondern direkt dort an der Stelle mit einer interaktiven Konsole aktiv debuggt werden. Eine Darstellung ist in Abbildung 5.8 zu sehen. Dadurch ist es möglich ohne den Rails Prozess zu stoppen direkt hintereinander diverse Befehle zu testen und somit schneller die Ursache des Fehlers aufzufinden. Dies war vor allem im Zusammenhang der in Kapitel 5.3 erläuterten Probleme bei der Entwicklung besonders hilfreich.

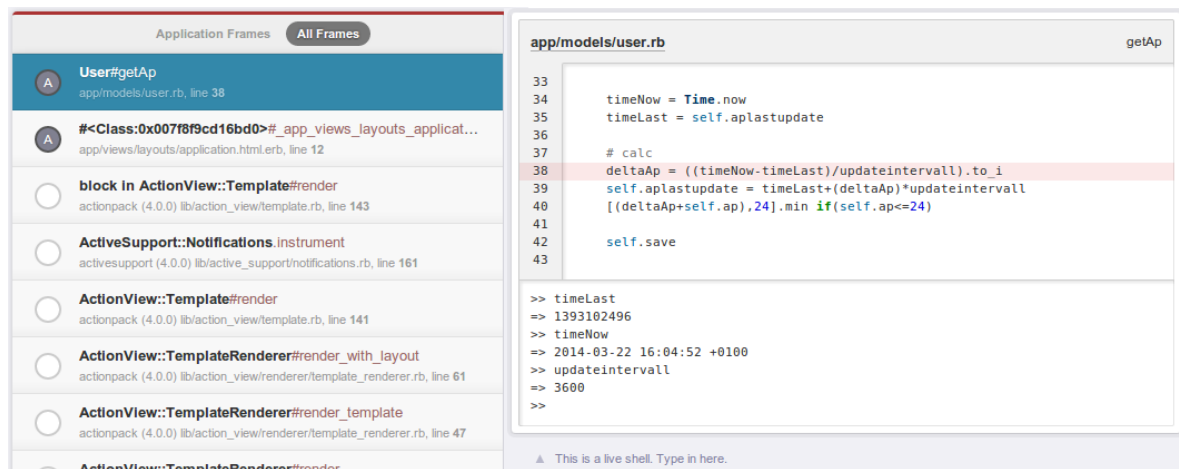


Abbildung 5.8.: Interaktives Debugging mit 'better_errors' und 'binding_of_caller'

Für die Entwicklung von Software ist es essentiell beliebig den Code wiederherstellen zu können und einfach neue Funktionen auszuprobieren. Diese Anforderung bedienen Software Systeme zur Versionsverwaltung. Die Auswahl wurde auf ein Open-Source System gelegt. Am weitesten verbreitet sind zur Zeit Subversion und Git. Letzteres bietet Möglichkeit einer nicht linearen Entwicklung [Bir+09]. Die Wahl fiel speziell auf Subversion, da es im Gegenzug zu CVS das Versionsschema nicht auf einzelne Dateien sondern auf das ganze Projekt bezieht. Das hat den Vorteil, dass das Hinzufügen einer neuen Funktion nicht in der Hauptklasse Version 50 und in der

Methodenklasse Version 70 gespeichert wird, sondern in einer gemeinsamen Version. Somit ist es für den Entwickler möglich den Zusammenhang zwischen den einzelnen Dateien direkt zu erkennen, da die neue Funktion z.B. in der Revision 60 in beiden Dateien erkennbar ist. Die Verwendung von Git wurde in Erwägung gezogen, aber Aufgrund der Tatsache, dass das Projekt nur einen Entwickler besitzt, wurde dies als unproblematisch angesehen. Ein späterer Transfer in ein Git-Projekt ist problemlos mit 'git-svn clone' und weiteren Anpassungen möglich. Der Transfer ist in jedem Fall zu empfehlen, gerade wenn das Framework später mit mehreren Entwicklern weiterentwickelt wird.

5.3. Implementierung des Geogameframeworks

Realisierung

Nachdem alle Anforderungen, Schnittstellen und Werkzeuge des Frameworks festgelegt wurden, konnte mit der Umsetzung begonnen werden. Wie bereits in Kapitel XX beschrieben lässt sich das Framework in drei Module unterteilen auf die im nachfolgenden jeweils entsprechend eingegangen werden soll.

- GameAPI (Overpass API)
- Beispielspiel
- Evaluationstool

GameAPI

Die GameAPI stellt die API des GameFrameworks dar. Sie wird dazu verwendet um die Spielfelder entsprechend aufzubauen. Generell gibt es X unterschiedliche Funktionen die zur Verfügung gestellt werden. Es wird unterschieden zwischen den Locations (im Beispielspiel Flaggen) sowie den lokalen Händlern. Der Aufruf der Schnittstelle erfolgt mittels eines RestFULL Request mit folgenden Parametern:

```
SITE_URL/overpass_api/getLocation.json?s=49&w=10&n=50&e=11&tag=highway=bus_stop
```

Der Aufruf kann sowohl als GET als auch als POST Request durchgeführt werden. Die Parameter s,n,w,e sind essentiell und der Parameter tag ist optional. Die Parameter sind zwei Werte. Die ersten vier Parameter beschreiben die angefragte Bounding Box ab. S steht hierbei für South (Bounding Box Minimum Latitude), N

für North (Bounding Box Maximum Latitude), W für West (Bounding Box Minimum Longitude) und E für East (Bounding Box maximum Longitude). Der Letzte Parameter tag beschreibt den zu verwendenden OSM Tag. Wird der entsprechende Tag nicht angegeben so wird der im Framework hinterlegte Default tag verwendet. Ziel ist es hierbei den Standard Wert für das darauf aufbauende Beispielspiel zu verwenden und den optionalen Parameter tag für die Evaluation einzelner Tags - konkret der jeweiligen key-value-Paare zu ermöglichen. Nachdem der entsprechende Aufruf erfolgt ist. Wird das Ergebnis als GeoJSON mit entsprechender Properties zurückgeliefert.

```
1 {  
2   "type": "FeatureCollection",  
3   "features": [  
4     {  
5       "type": "Feature",  
6       "geometry": {  
7         "type": "Point",  
8         "coordinates": [  
9           10.8748794,  
10          49.9002723  
11        ]  
12      },  
13      "properties": {  
14        "popupContent": "Test",  
15        "id": "301967628",  
16        "user_id": "neutral",  
17        "prestige": 0  
18      }  
19    }  
20  ]  
21 }
```

Code 5.1: GeoJSON Response Location (Reduziert)

Wie in Code 5.1 zu sehen ist die Antwort der API analog zur GeoJSON Spezifikation [But+08]. In jedem Fall enthält ein GeoJSON ein Objekt. Enthält das GeoJSON mehr als ein Objekt so sind diese in einer FeatureCollection gesammelt. Diese wiederum enthält entsprechende Objekte. Jedes Objekt nimmt einen der nachfolgenden Objekttypen an: "Point", "MultiPoint", "LineString", "MultiLineString", "Polygon", "MultiPolygon" oder "FeatureCollection". Für die Implementierung des Frameworks wird allerdings nur Point und FeatureCollection genutzt. Reduzierung auf Points ist der Tatsache geschuldet, dass durch die Transformation in Virtuelle Nodes lediglich feste einzelne Spielelemente existieren die auf einen Punkt reduziert wurden. Daher ist das passendste Element im GeoJSON ebenfalls der Point. Neben den Koordinaten des Punktes können zusätzlich weitere Attribute frei definiert werden. Diese für das Gameframework unter anderem id, user_id und prestige. Ersteres

beschreibt die virtuelle Node Id des OSM Objekts von dem das Node abstammt. In diesem Fall lässt sich anhand der Zahl erkennen, dass es sich um ein Node handelt und kein Way oder Relation von dem der Punkt abstammt. Das Attribut `user_id` beschreibt den Besitzer des Elements. Für das Beispielspiel wurde hier keine konkreten Ids sondern die Stati "neutral", "owner", "foe" bestimmt. Mit einer einfachen Anpassung einer Zeile im Framework können hier auch direkt die Userid ausgegeben werden. Sofern die Schnittstelle ohne Session von extern aufgerufen wird, gibt es nur den Zustand "neutral" oder "foe", da ein Anonymer Zugriff keinem eingeloggtem User zugeordnet wird. Ist hingegen ein User authentifiziert über das Beispielspiel so erhält er zusätzlich die Information ob die aktuelle Flagge in seinem Besitz ist. Das Attribut `Prestige` gibt ganz normal den aktuellen Prestige Wert der Flagge zurück. In diesem Fall handelt es sich um ein neutrales nicht bisher eingenommene Flagge die daher automatisch den Wert 0 hat und noch nicht in der Datenbank persistiert wurde. Eine Aussage ob ein Objekt entsprechend persistiert wurde oder nicht kann derjenige der die API verwendet nicht erkennen. Dies ist aber auch nicht notwendig, da die Persistierung transparent² vor dem Spieler und gegenüber dem Schnittstellenbenutzer ist.

Die nächste Schnittstelle stellt die Händlerschnittstelle dar. Über diese können die Händler in einer vorgegebenen Bounding Box abgefragt werden, analog zu den Flaggen.

`SITE_URL/vendors/getVendors.json?s=49&w=10&n=50&e=11`

Im Gegensatz zu einem Spielelement enthält das GeoJSON des Händlers allerdings weniger Attribute, wie in Codebeispiel 5.2 zu sehen.

```
1 {  
2     "type": "Feature",  
3     "geometry": {  
4         "type": "Point",  
5         "coordinates": [10.869845151901245, 49.902191491264695]  
6     },  
7     "properties": {  
8         "popupContent": "Insel 11"  
9     },  
10    "id": 2  
11 }
```

Code 5.2: GeoJSON Response Vendor (Reduziert)

Die Standardattribute sind gleich, jedoch wurden die Properties reduziert. 'popupContent' beschreibt den Inhalt des Popup-Fensters. In diesem Fall werden hier die

²im englischen Sinne

Karte selbst sind per Layer die entsprechenden Spielelemente, sowie Händler eingebunden. Bewegt der Spieler den Kartenbereich oder bewegt er sich physikalisch fort, so werden die Daten entsprechend nachgeladen. Die Daten werden mittels GameAPI über das Framework ausgelesen und als GeoJSON eingebunden. Möchte nun der Spieler mit den entsprechenden Spielelementen interagieren, so muss er nur auf das entsprechende Element klicken. Im Beispielspiel öffnet sich dann je nach Elementtyp entweder die Übersicht der jeweiligen Flagge oder aber der Händler. Durch die Interaktion kann sich der Spieler über den aktuellen Prestige-Stand informieren, sowie die Flagge angreifen. Dies kann der Spieler allerdings nur, wenn er sich im Umkreis von 40 Metern zur Flagge befindet. Beim AJAX Request, welcher der Spieler bei einem Angriff entsprechend durchführt, wird dieses sichergestellt. Der Aktionsradius von 40 Metern soll sicherstellen, dass auch bei einer höheren GPS- Ungenauigkeit der Spieler trotzdem mit der Umgebung interagieren kann. Beim Händler erhält der Spieler eine Übersicht über die verfügbaren Items. Im Gegensatz zu den Flaggen werden die Informationen über die jeweiligen Händler nicht direkt bei dem Abruf aller Händler mitgeteilt, sondern explizit für den jeweiligen Händler angefordert. Somit wird sichergestellt, dass die Synchronisierung möglichst zeitnah ist und der Spieler eine aktuelle Übersicht über das Inventar des virtuellen Händlers erhält. Den aktuellen Punktestand kann der Spieler von der oberen Statusleiste entnehmen. Dieser berechnet sich anhand aller Flaggen die der Spieler eingesammelt hat. Dank Active Record in Verbindung mit Rails kann die Abfrage stark vereinfacht werden:

```
1 points = Flag.sum(:prestige, :conditions => ['user_id = ?', self.id])
```

Code 5.3: Ruby - Abfrage der Spielerpunkte

Direkt daneben ist die Anzeige für die verbliebenen Aktionspunkte. Diese werden alle 60min um eins erhöht bis diese 24 Aktionspunkte erreichen. Der letzte Punkt stellt die Item-Übersicht dar. Über diese kann der Spieler seine aktuellen Items anzeigen lassen und je nach Bedarf diese auch entsprechend verwenden. Die Items werden erst mit dem Klick auf das Inventar explizit per AJAX nachgeladen. Zuvor kann der Spieler jedoch die Anzahl seiner Items in der Statusleiste sehen.

Im Gegenzug dazu gibt es die Oberfläche für die Pflege der einzelnen Händler. Hierfür werden als Basis, durch Scaffolding generierte Formulare verwendet. Diese wurden entsprechend um zusätzliche Funktionen wie einem Leaflet map picker, sowie der dazugehörige JavaScript Code ergänzt. Zunächst kann der Spielleiter sich eine Übersicht der Händler über die nachfolgende URL aufrufen:

`SITE_URL/vendors`

In der Übersicht kann er bestehende Händler direkt löschen, Neue anlegen oder Bestehende bearbeiten. Legt der Spielleiter einen Händler an, so wird ihm nicht nur eine Liste an Attributen angezeigt, sondern er erhält auch eine Leaflet Karte, die entsprechend auf der aktuellen Position zentriert ist. Über diese kann er frei auf der Karte einen Marker für die Position des Händlers setzen. Intern werden die Koordinaten des Markers entsprechend gespeichert und in der Datenbank hinterlegt. Möchte der Spielleiter nun einen der Händler bearbeiten wird analog dazu das Formular wieder aufgerufen und entsprechend mit den Daten aus der Datenbank gefüllt. Gleiches gilt auch für die Leaflet Karte auf der der zuvor gespeicherte Marker hinterlegt wurde. In diesem Modul hat der Spielleiter zudem die Möglichkeit Items direkt dem Händler zuzuweisen die dann später zum Verkauf stehen. Für die Einfachheit werden entsprechend alle freien noch nicht zugewiesenen Items entsprechend angezeigt und können mit einem einfachen Klick hinzugefügt werden. Hierfür sind entsprechende Controller für die Händler Klasse erstellt worden, die das Kaufen und Zuweisen von Items ermöglichen.

Über die nachfolgende Unterseite, findet die Pflege der Items statt:

`SITE_URL/items`

Auf dieser Seite erstellt der Spielleiter alle Items die den Händlern zur Verfügung stehen sollen. Hierbei handelt es sich ebenfalls um ein nach typischen Scaffolding erzeugtem Formularmuster zur Pflege der einzelnen Daten. Jedes Item ist dabei explizit als Objekt angelegt und kann einem Itemtyp angehören. Eine Anpassung der Items ist nach der Erstellung möglich. Im Zuge des Beispielspiels wurde auf eine umfangreiche Rollen und Rechtevergabe verzichtet, da der Fokus auf den Spielelemente lag. Nichtsdestotrotz können diese entsprechend erweitert werden, je nachdem für welches Spiel das Framework eingesetzt werden soll.

Evaluationstool

Das Evaluationstool verwendet die zuvor in der GameAPI beschriebene Schnittstelle um Spielfelder zu bewerten bzw. zu evaluieren. Hierfür wird die zusätzliche Möglichkeit genutzt konkrete tags zu einer Bounding Box abzufragen. Für eine Evaluation verwendet das Tool eine vorgegebene Liste an Key-Value Paaren und eine Geo-Koordinate, die zu einer Bounding Box erweitert wird. Basierend auf dem im Kapitel 6.2 beschriebenen Ansätzen werden die einzelnen Spielfelder jeweils evaluiert. Für die Evaluation der Spielfelder müssen die Entfernung zwischen den einzelnen Punkten berechnet werden. Hierbei werden nicht die euklidische Distanz zwi-

schen den Geo-Koordinaten berechnet sondern es muss die reale Netzwerkdistanz auf dem Straßennetz berechnet werden. Da der Spieler nicht nur per Auto sondern bevorzugt auch per Fahrrad und im besten Fall zu Fuß das Spiel nutzt, ist ein Fußgängerrouting notwendig. Das bedeutet Wege die nur für Fußgänger zugänglich sind, müssen ebenfalls beachtet werden. Da Openstreetmap im Vergleich zu Google Maps im Hinblick auf den Datenumfang an dieser Stelle einen Vorteil hat, ist es sinnvoll hierfür auf OSM zurückzugreifen. Durch den Umfang und Anzahl der Abfragen die sich durch die Evaluationsmethoden ergeben, ist es sinnvoll die Anfragen nicht an einen Onlinedienst zu stellen sondern diese offline mit einem dedizierten Routing durchzuführen. Für das offline Routing mit OSM gibt es diverse Bibliotheken, aufgrund der Anforderung möglichst viele Abfragen zeitnah durchzuführen wurde sich für das Tool GraphHopper entschieden. Dieses bietet ein vollständiges Offline Routing auf Basis von OSM Rohdaten [Kar14]. Hierfür erstellt GraphHopper zunächst entsprechende Indexdateien auf denen dann später das Routing stattfindet. Diese enthalten in binärer Form die aggregierten Pfade des Netzwerkes.

Mit Hilfe von GraphHopper ist es nun Möglich entsprechend schnelle Abfragen durchzuführen. Da die GraphHopper Bibliothek selbst nicht für Parallelisierung ausgelegt ist, wurde das Evaluationstool entsprechend optimiert um die Ressourcen eines Rechners/Servers vollständig auszunutzen. Dadurch kann die Laufzeit bei mehreren Tags je nach Anzahl der vorhandenen CPUs auf $\frac{1}{lc}$ verkürzt werden. Wobei lc die Anzahl der logischen CPUs darstellt. Bei einer Laufzeit von $\mathcal{O}(n^2)$ ist dies unerlässlich. Da GraphHopper in Java realisiert wurde, wurde das Evaluationstool analog dazu ebenfalls in Java entwickelt.

Tests

Da das Framework später entsprechend für verschiedene Spiele genutzt werden soll, muss dieses entsprechend getestet werden. Hierbei wird unterschieden zwischen Low-Level-Tests und High-Level-Tests[PTVV02], Unter Low-Level-Tests sind solche Test zu verstehen, die während der Implementierung an Teilen des Systems stattfinden. Bei High-Level-Tests wird das komplette System getestet. Einer der Low-Level-Tests ist der Modultest. Bei diesem werden einzelne Module im Programm getestet.

Für das Gameframework wurden die einzelnen Module unabhängig voneinander getestet. Zunächst wurden speziell die Schnittstellen zu OSM und Overpass getestet. Hierbei wurde vor allem kontrolliert, dass die Übergabeparameter, sowie das Datenformat korrekt ist und die Interpretation der Daten korrekt vorgenommen wird. Nachdem die die GameAPI mit ihrer Transformation der OSM Elemente in Spielelemente implementiert wurde, ist das Beispielspiel entsprechend darauf aufgebaut worden. Hierzu wurde die korrekte Transformation der Elemente anhand von spezi-

ellen Tags manuell überprüft. Hierzu wurde <http://taginfo.openstreetmap.org> verwendet. Diese Seite bietet einen statistischen Überblick aller Tags in OSM sowie die Verteilung auf die Elemente Relation, Way und Nodes. Der Test erfolge anhand von Key-Value Paaren die jeweils explizit nur als einer der drei Typen bevorzugt gemappt werden.

Das Spiel selbst musste ebenfalls getestet werden. Da ein ortsbezogenes Spiel als Grundlage die Position des Spielers verwendet und ein Debugging unterwegs sich als schwierig gestaltet, ist es am besten die GPS-Koordinate zu simulieren. Hierfür gibt es entsprechende Plugins für die am weitesten verbreiteten Browser, wie Chrome oder Firefox. Mit Hilfe von diesen kann die Position, welche die HTML5 Geo API zurückliefert, entsprechend verändert werden. Darüber hinaus ist es auch möglich die Genauigkeit der GPS Position, sowie das Bewegungs-Event zu triggern. Mit Hilfe von diesem können alle ortsbezogenen Funktionen des Beispielspiels auch direkt während der Entwicklung getestet werden. Für die Nutzung des Spiels auf Smartphones eignet sich darüberhinaus der Einsatz von entsprechenden Emulatoren. Für Android und FirefoxOS sind diese frei verfügbar. Für iOS fallen jährliche Gebühren an und der Emulator inkl. SDK ist nur unter der neuesten Mac OS X Version verfügbar.

Tests für das Evaluationstool wurden vorwiegend für die Performance und den Werten durchgeführt. Konkrete Unit-Tests wurden für das Framework aus Zeitgründen nicht entwickelt, sollten aber als nächster wichtiger Punkt entsprechend implementiert werden.

Als High-Level bzw. Blackbox Test wurden die in den Usecase Diagramm beschriebenen Anwendungsfälle getestet. D.h. die Funktionen, die den jeweiligen Akteuren zur Verfügung stehen, wurden ohne Betrachtung.

Probleme

Während der Umsetzung sind auf einige Besonderheiten gestoßen, welche eine spezielle Anpassung oder Überlegung erfordert haben. Im nachfolgenden soll kurz auf diese eingegangen werden, um die Erkenntnisse festzuhalten. Idealerweise können diese in zukünftigen Untersuchungen und Implementierungen entsprechend vermieden bzw. umgangen werden.

Ein erster Aspekt der von Relevanz ist die Spezifikation der GeoJSON. Standardmäßig werden Koordinaten in der Reihenfolge Latitude, Longitude angegeben [Sch02; Bar+96; Mal91]. Die GeoJSON Spezifikation hingegen sieht im Kontrast zum de facto Standard vor, dass zuerst die Longitude und dann die Latitude genannt wird [But+08]. Ist dies nicht bekannt, kann dies zu einigem Aufwand führen, der vermieden werden kann.

Ein weiteres Problem, die im Zuge des Transformationsprozesses der Relations

und Ways zu virtuellen Nodes mit einer mit Bitmasken codierten Eigenschaft, führte dazu, dass im Zuge der Verwendung mit Javascript Probleme auftraten. Dies äußerte sich darin, dass die Ids nach der Interpretation plötzlich auf unerklärlicher Weise veränderte Werte annahmen. So kamen Abweichungen von bis zu 100 zu Stande. Nach längeren Debug Aufwänden wurde herausgefunden, dass die implizite Typisierung in JavaScript den Ganzzahlenwert der Id intern als Gleitkommazahl transferiert und dadurch unbewussterweise die 51Bit überschreitet nach denen die Mantisse der Gleitkommazahl beginnt. Durch diese Verschiebung wurden die hinteren Bits beim Auslesen des long Wertes vernachlässigt. Um dieses Problem zu umgehen gibt es zwei Möglichkeiten. Entweder man teilt die Zahl auf Basis des 64bit long Wertes in zwei 32bit Integer Werte und legt diese in zwei Zahlen in JS ab. Sollten keine arithmetischen Operationen auf der Id durchgeführt werden sollen im JavaScript Code, dann empfiehlt es sich die Id explizit als String auszugeben, damit JavaScript diese nicht als Zahl interpretiert und versucht entsprechend umzuwandeln.

Ein weiterer Punkt stellt Turbolinks dar. Turbolinks ist ein Gem, welches Standardmäßig in Rails aktiv ist. Es sorgt dafür, dass bei einer Interaktion mit der Seite nicht die komplette Seite neu geladen wird, sondern nur die Teile des Html Codes, welcher sich geändert hat [GCAB13]. Die Problematik die damit einhergeht sind Ajax Requests, welche über normale href links angestoßen werden. Eine typische Verwendung ist z.B.:

```
1 <a href='#' onClick="saveItem(1);">Save</a>
```

Code 5.4: a href HTML Code

Allerdings führt der Klick auf SSave in diesem Fall zu einem Neuladen der Seite durch Turbolinks. Befindet sich an dieser Stelle aber ein Javascriptcode mit Ajax request, passiert es, dass die Seite neu geladen wird anstatt den Code auszuführen, welcher z.B. eine Aktualisierung einer Zahl auf der Seite zur Folge hätte. Um diese zu vermeiden muss Turbolinks explizit angewiesen werden, bei a href Links nicht aktiv zu werden. Dies kann man pro Link individuell setzen:

```
1 <a href='#' data-no-turbolink onClick="saveItem(1);">Save</a>
```

Code 5.5: a href HTML Code - Turbolinks deaktiviert

Dadurch wird sichergestellt, dass der entsprechende JavaScript Code ausgeführt wird und nicht einfach die Seite neugeladen wird.

Ein letzter Punkt, der zu Problemen führen kann ist der SZurückButton im Browser. Durch diesen wird die vorherige Seite wieder aufgerufen, allerdings deren Javascriptcode nicht noch einmal 1:1 getriggert wie beim Laden der Seite. D.h. diverse Event-Listener welche auf `ÖnDocumentReady` z.B. mittels JQuery warten, werden nicht noch einmal aufgerufen. Wird dies nicht entsprechend in der Entwicklung berück-

sichtigt, kann es dazu führen, dass die Seite sich in einem nicht definierten Zustand befindet.

6. Evaluierung

6.1. Qualität der Lösung

Für eine Evaluation der in Kapitel 4 und 5 vorgestellten Lösung muss zunächst die Evaluationsmethode definiert werden. Für die Evaluation soll die Lösung mit der in Kapitel 2 definierten Problemstellung gegenübergestellt werden. Es wird daher untersucht, inwiefern das Framework, sowie das Beispielspiel zur einer Lösung der Problemstellung beitragen. Das Ergebnis der Evaluation soll entsprechend weitere Handlungsempfehlungen bzw. Verbesserungsvorschläge aufzeigen, sofern diese aufgrund der Evaluation notwendig sein sollten. Zunächst sollen die nachfolgenden Punkte analysiert werden im Hinblick auf die dargestellte Lösung:

- Gamification unter Einbezug der Händler
- Pervasive Games
- Relokalisierbarkeit
- Verwendbarkeit von OSM Daten

Die Evaluation der Spielfelder, welche durch das Framework erzeugt werden und somit eine Bewertung der Relokalisierbarkeit wurde als eigenständiger Part in Kapitel 6.2 separat behandelt.

Gamification

Im Bezug auf die in Kapitel 1 aufgestellte Forschungsfrage soll die Lösung mittels Gamification den regionalen Händlern zu neuen Kunden führen können und bestehende entsprechend halten. Ziel ist es nicht zu überprüfen, ob und wie hoch der entsprechende Neukunden und Bestandskundenanteil durch die Lösung beeinflusst wird, sondern inwiefern entsprechende Elemente aus der Literatur umgesetzt wurden. Diese wiederum sollen die Möglichkeit für einen besseren Umsatz der Händler bieten. In der Literatur wurden zunächst die rudimentären Elemente eines Gamification Prozesses herauskristalisiert. Für die softwaretechnische Umsetzung wurden

entsprechende Möglichkeiten geschaffen und entsprechend integriert, damit die typischen Elemente Points, Badges und Leaderboards problemlos umgesetzt werden können. Zudem wurden die Aspekte der klassischen Spieltheorie aufgegriffen und dadurch die Gamificationen Ansätze erweitert. Es wurden mehreren Optionen aufgezeigt, welche in Spielen umgesetzt werden können. Darüber hinaus wurden diese Aspekte mit entsprechenden Technologien wie NFC, oder Bluetooth Low Energy in Verbindung gebracht, wie eine Integration der Spielelemente vor Ort bei den Händlern gestaltet werden könnte. Das Framework und das Beispielspiel bilden diese nicht ab, sind aber entsprechend in diese Richtung problemlos erweiterbar. Durch eine entsprechende Kombination der Gamification Elemente und der Tatsache, dass das Framework auf Basis von Informationen aus der Umgebung Spielelemente aufbereitet und lokale Händler integriert wird auch das Maß der Immersion entsprechend für die Spieler erhöht und der Effekt der Gamification verstärkt. Für den Aspekt der Gamification lässt sich daher abschließend festhalten, dass die aufgezeigte Lösung einen Beitrag zur Lösung der Problematik der Gamification von Neukundenbesuchen bei regionalen Händlern liefern kann.

Pervasive Games - Anforderungen an ein Gameframework

Im Zuge der Erstellung eines Frameworks zum Staging von Pervasive Games müssen mehrere Aspekte behandelt werden. Zunächst muss sichergestellt werden, dass die definierten Grenzen des Magic Circle überschritten werden können. Im Zuge dieser Arbeit ging es darum die Dimension Ort und Zeit zu lösen. Das bedeutet, dass Spiel soll sowohl ortsunabhängig als auch zeitunabhängig gespielt werden kann. Die Ortsunabhängigkeit der vorgestellten Lösung wird durch die Relokalisierung mit Hilfe von OSM-Daten sichergestellt, auf die später im Detail eingegangen wird. Die zeitliche Unabhängigkeit ist dadurch sichergestellt, dass jeder Spieler sich individuell am Beispielspiel anmelden kann und es keine fest definierten Uhrzeiten oder Termine gibt um das Spiel zu spielen. Im Vergleich zu typischen Geogames ist es daher möglich auch über feste Uhrzeiten und Geografische Gebiete hinaus das Spiel zu spielen. Durch die entsprechende Formulierung der Anforderungen in Kapitel 2 und 4.1 und der Ausrichtung des Frameworks an diesen wurde sichergestellt, dass das Framework diese Anforderungen erfüllt. Da es sich in der Problemstellung um ein Echtzeitspiel handelt muss auch entsprechend im Framework sichergestellt werden, dass die Interaktion mit dem Spiel in Echtzeit erfolgen kann. Durch die Modularisierung und Auskopplung der Evaluation der OSM Tags, ist es möglich geworden die zeitintensiven Bewertungs-Funktionalitäten unabhängig von der Spielfeldgenerierung durchzuführen. Dadurch ist es ermöglicht, lediglich die relativ einfachen Prozesse zur Transformation der OSM-Elemente während der Laufzeit durchzuführen.

ren. Vergleicht man die Antwortzeit der Requests, welche abhängig sind von der Anzahl der OSM-Elemente in der Bouding Box und der Antwortzeit des Overpass API Servers, so liegen diese im Schnitt bei ca. 400ms. Die Varianz ist wie in Abbildung 6.1 erkenntlich zu vernachlässigen. Die Evaluationsfunktion hingegen liegt bei 2-3 Minuten für ein Tag mit ca. 200 Elementen. Für ein Feld mit 1200 Elementen liegt diese bereits bei über 30 Minuten.

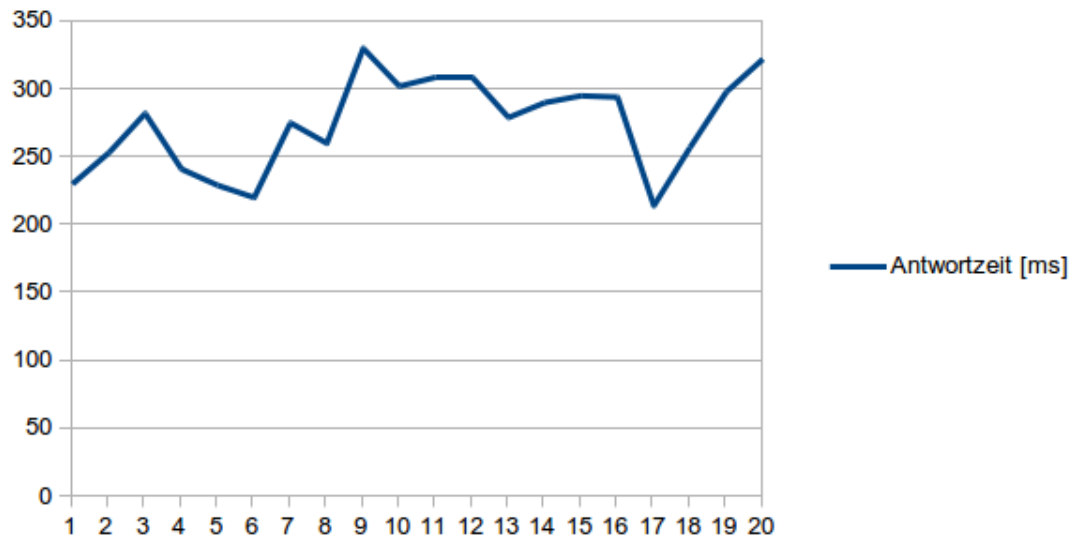


Abbildung 6.1.: Antwortzeit der GameAPI für den Tag
public_transport=stop_area

Abweichungen der Antwortzeit sind weniger bedeutend, da das Spielfeld sich bereits vorher für den Spieler aufbaut und die Spielelemente mittels Ajax Request geladen werden. Darüber hinaus wird durch die dynamische Erweiterung der Bounding Box sichergestellt, dass alle Elemente am Rand der Karte bereits geladen sind. Das Nachladen der Elemente beim Fortbewegen kann daher auch minimal länger dauern, da das Spielfeld immer zentriert auf den Spieler ist.

Durch die Kapselung der einzelnen Module wird auch sichergestellt, dass eine Erweiterbarkeit der Spielmechnaik ohne größeren Aufwand möglich ist. Das entsprechende Beispielspiel kann problemlos erweitert werden oder aber durch ein beliebiges anderes Spiel ersetzt werden. Hierbei zeigt sich, dass die Anforderung für die Flexibilität des Frameworks hinsichtlich seiner Erweiterbarkeit und Austauschbarkeit gegeben ist.

Relokalisierbarkeit

Der nächste essentielle Aspekt der Problemstellung war die Relokalisierbarkeit. Ziel war es im Zuge des einfachen Stagings dem Spielleiter entsprechend eine Möglichkeit an die Hand zu geben, welche im auch ein Spielen außerhalb eines festen Bereiches ermöglicht. Hierzu wurden entsprechende OSM-Daten verwendet die durch eine Transformation zu Spielelementen umgewandelt wurden. Durch die entsprechende Evaluation der OSM Tags im Voraus wird sichergestellt, dass der jeweilige beste Tag für die getesteten Umgebungen ausgewählt wurde.

-Verarbeitung von Geodaten -Einbindung der lokalen Händlern

Verwendbarkeit von OSM Daten

-OSM-Daten Transformation/Aufbereitung

-vergleich zu den anforderungen

Für den Spielleiter lässt sich daher festhalten, dass die Anforderung für ein einfaches Staging erfüllt wird. Darüberhinaus findet auch die geforderte Modularisierung der einzelnene Funktionen statt.

Was ist noch zu tun?

6.2. Qualität der Spielfelder

7. Diskussion

7.1. Einordnung der Ergebnisse

7.2. Relokalisierbarkeit geobasierter Gamification-Ansätze

7.3. Einfache Wiederverwertbarkeit durch Relokalisierung

7.4. Ausblick

-Evaluation Spielfelder ausbauen -Framework untersuchen/optimieren auf Hochskalierbarkeit -reales Spielfeedback, iterativ

A. Anhang

A.1. Section

The contents...

B. Glossar

API	Die API stellt eine dokumentierte Software-Schnittstelle dar, die von anderen Programmen aus genutzt werden kann.
Basismaschine	Eine Basismaschine stellt Datenobjekte und Operatoren bereit, auf deren Grundlage die Datenobjekte und Operatoren der Nutzermaschine realisiert werden.
CLI	Command Line Interface - Kommandozeile. Die Kommandozeile ist ein Eingabebereich für die Steuerung einer Software, die typischerweise im Textmodus abläuft.
DNS	Ermöglicht es Klarnamen in numerische IP Adressen (z.B. google-public-dns-a.google.com in 8.8.8.8 umzuwandeln).
GUI	Hierbei handelt es sich um die grafische Benutzeroberfläche.
OQL	Overpass Query Language.
RFC	RFCs sind eine Reihe von technischen und organisatorischen Dokumenten zum Internet, die sie zu einem Standard entwickelt haben.
POI	Pointn of Interest. Sehenswürdigkeit
PBL	Points, Badgets, Leaderboards
Shell	Eingabe-Schnittstelle zwischen Computer und Benutzer
SQL	Eine deskriptive Abfragesprache von Datenbanken.

Literatur

- [AWC99] Marc S Atkin, David L Westbrook und Paul R Cohen. “Capture the Flag: Military simulation meets computer games”. In: *Proceedings of AAAI Spring Symposium Series on AI and Computer Games*. 1999, S. 1–5.
- [Bal+07] Rafael A Ballagas, Sven G Kratz, Jan Borchers, Eugen Yu, Steffen P Walz, Claudia O Fuhr, Ludger Hovestadt und Martin Tann. “REXplorer: a mobile, pervasive spell-casting game for tourists”. In: *CHI’07 extended abstracts on Human factors in computing systems*. ACM. 2007, S. 1929–1934.
- [Bar04] Richard A Bartle. *Designing virtual worlds*. New Riders, 2004.
- [Bar+96] Farhad Barzegar, Irwin Gerszberg, Martin J McGowan III und Robert E Schroeder. *Wireless information system for acquiring location related information*. US Patent 5,559,520. Sep. 1996.
- [Bel+06] M. Bell, Chalmers M., Barkhuus L., Hall M. und Sherwood. “Interweaving mobile games with everyday life: In, Conference on Human Factors in Computing Systems, 22-27 April 2006, pages pp. 417-426, Montreal, Canada.” In: (2006), S. 417–426.
- [Ben+03] Steve Benford, Adam Drozd, Andy Crabtree, Rob Anastasi, Martin Flintham, Ju Row-Farr, Chris Greenhalgh, Matt Adams und Nick Tandavanitj. “Coping with uncertainty in a location-based game”. In: *IEEE pervasive computing* 2.3 (2003), S. 34–41.
- [Ber14] Stadt Berlin. *Berlin Open Data*. abgerufen am: 11.02.2014. 2014. URL: <http://daten.berlin.de/>.
- [Bet07] E. Bethke. “MMO goal structures as a panacea”. In: *Proceedings of the Austin Game Developers Conference 2007*. Austin, Texas, 2007.
- [Bir+09] Christian Bird, Peter C Rigby, Earl T Barr, David J Hamilton, Daniel M German und Prem Devanbu. “The promises and perils of mining git”. In: *Mining Software Repositories, 2009. MSR’09. 6th IEEE International Working Conference on*. IEEE. 2009, S. 1–10.

- [BK07] Michael Bächle und Paul Kirchberg. “Ruby on Rails.” In: *IEEE Software* 24.6 (2007), S. 105–108.
- [BML05] Steve Benford, Carsten Magerkurth und Peter Ljungstrand. “Bridging the physical and digital in pervasive gaming”. In: *Communications of the ACM* 48.3 (2005), S. 54–57.
- [Bri05] Thomas Brinkhoff. *Geodatenbanksysteme in Theorie und Praxis*. Wichmann Heidelberg, 2005.
- [Bro+13] Steve Bromley, Pejman Mirza-Babaei, Graham McAllister und Jonathan Napier. “14 Playing to Win?” In: *Multiplayer: The Social Aspects of Digital Gaming* (2013), S. 172.
- [Bur94] Donald Burleson. “OODBMSs gaining MIS ground but RDBMSs still own the road”. In: *Software Magazine* 14.11 (1994), S. 63–68.
- [But+08] Howard Butler, Martin Daly, Allan Doyle, Sean Gillies, Tim Schaub und Christopher Schmidt. *The GeoJSON Format Specification*. 2008.
- [Car07] Carsten Röcker Carsten Magerkurth. *Concepts and Technologies for Pervasive Games: A reader for pervasive gaming research vol. 2*. Bd. 1. A reader for pervasive gaming research. Aachen: Shaker, 2007. ISBN: 3832262237.
- [Cat91] RGG Cattell. “Next-generation database systems”. In: *Communications of the ACM* 34.10 (1991), S. 30–33.
- [Cel10] E. Celtek. “Mobile advergames in tourism marketing”. In: *Journal of Vacation Marketing* 16.4 (2010), S. 267–281.
- [Cel13] Irene Celino. “Location-Based Games for Citizen Computation”. English. In: *Handbook of Human Computation*. Hrsg. von Pietro Michelucci. Springer New York, 2013, S. 297–316. ISBN: 978-1-4614-8805-7. DOI: 10.1007/978-1-4614-8806-4_25. URL: http://dx.doi.org/10.1007/978-1-4614-8806-4_25.
- [Che10] Adrian David Cheok. *Art and technology of entertainment computing and communication: Advances in interactive new media for entertainment computing*. London und New York: Springer, 2010. ISBN: 1849961379.
- [CL11] Andre Charland und Brian Leroux. “Mobile application development: web vs. native”. In: *Communications of the ACM* 54.5 (2011), S. 49–53.

- [CMAK13] Maurizio Caon, Elena Mugellini und Omar Abou Khaled. “A Pervasive Game to Promote Social Offline Interaction”. In: *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*. UbiComp '13 Adjunct. Zurich, Switzerland: ACM, 2013, S. 1381–1384. ISBN: 978-1-4503-2215-7. DOI: 10.1145/2494091.2497359. URL: <http://doi.acm.org/10.1145/2494091.2497359>.
- [Com14] Open Data Commons. *Open Database License (ODbL) v1.0*. 2014. URL: <http://opendatacommons.org/licenses/odbl/1.0/>.
- [Con05] Mia Consalvo. “Gaining Advantage: How Videogame Players Define and Negotiate Cheating.” In: *DIGRA Conf.* 2005.
- [CR01] J. Chen und M. Ringel. *Can Advergaming be the Future of Interactive Advertising?* 2001.
- [CRB06] Paul Coulton, Omer Rashid und Will Bamford. “Experiencing ‘touch’ in mobile mixed reality games”. In: *International Conference in Computer Game Design and Technology*. 2006.
- [Csi91] M Csikszentmihalyi. *Flow, The Psychology of Optimal Experience, Steps towards enhancing the quality of life*. Harper&Row, Publishers, 1991.
- [DD08] Paul Deitel und Harvey Deitel. *Deitel's #174; Developer Series Ajax, Rich Internet Applications, and Web Development for Programmers*. First. Upper Saddle River, NJ, USA: Prentice Hall Press, 2008. ISBN: 9780137142309.
- [DEB09] Stephan Dahl, Lynne Eagle und Carlos Báez. “Analyzing advergaming: active diversions or actually deception. An exploratory study of online advergaming content”. In: *Young Consumers: Insight and Ideas for Responsible Marketers* 10.1 (2009), S. 46–59. DOI: doi : 10 . 1108 / 17473610910940783. URL: <http://www.ingentaconnect.com/content/mcb/yc/2009/00000010/00000001/art00004>.
- [Der13] Jonathan Derrough. *Instant Interactive Map Designs with Leaflet Javascript Library How-To*. Packt Publishing, 2013.
- [Det+11] Sebastian Deterding, Dan Dixon, Rilla Khaled und Lennart Nacke. “From game design elements to gamefulness: defining gamification”. In: *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*. ACM. 2011, S. 9–15.
- [DG13] Trinh Minh Tri Do und Daniel Gatica-Perez. “Human interaction discovery in smartphone proximity networks”. In: *Personal and Ubiquitous Computing* 17.3 (2013), S. 413–431.

- [Dig14] Center for Digitization. *Denmark Open Data*. abgerufen am: 11.02.2014. 2014. URL: <http://digitaliser.dk/ressourcer>.
- [DR01] Goran M Djuknic und Robert E Richton. "Geolocation and assisted GPS". In: *Computer* 34.2 (2001), S. 123–125.
- [Dra13] Damir MEDAK Mario MILER Dražen Odobasic. "Gamification of Geographic Data Collection". In: (2013).
- [Duc+06] Nicolas Ducheneaut, Nicholas Yee, Eric Nickell und Robert J Moore. "Alone together?: exploring the social dynamics of massively multiplayer online games". In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM. 2006, S. 407–416.
- [FGG09] Chris Forman, Anindya Ghose und Avi Goldfarb. "Competition between local and electronic markets: How the benefit of buying online depends on where you live". In: *Management Science* 55.1 (2009), S. 47–57.
- [FM08] Andrew J Flanagin und Miriam J Metzger. "The credibility of volunteered geographic information". In: *GeoJournal* 72.3-4 (2008), S. 137–148.
- [FOS13] FOSSGIS. "Anwenderkonferenz für Freie und Open Source Software für Geoinformationssysteme". In: (2013).
- [Gar13] Gartner. *Gartner Says Annual Smartphone Sales Surpassed Sales of Feature Phones for the First Time in 2013*. Abgerufen am: 05.03.2014. 2013. URL: <http://www.gartner.com/newsroom/id/2665715>.
- [GCAB13] Adam Gamble, Cloves Carneiro und Rida Al Barazi. "JavaScript and CSS". In: *Beginning Rails 4*. Springer, 2013, S. 191–201.
- [GD05] Philippe Golle und Nicolas Ducheneaut. "Preventing bots from playing online games". In: *Computers in Entertainment (CIE)* 3.3 (2005), S. 3–3.
- [GO98] E Grafarend und F Okeke. "Transformation of conformai coordinates of type Mercator from a global datum (WGS 84) to a local datum (Regional, national)". In: *Marine Geodesy* 21.3 (1998), S. 169–180.
- [Goo07] Michael F Goodchild. "Citizens as sensors: the world of volunteered geography". In: *GeoJournal* 69.4 (2007), S. 211–221.

- [Gra95] E. Grafarend. “The Optimal Universal Transverse Mercator Projection”. English. In: *Geodetic Theory Today*. Hrsg. von Fernando Sansò. Bd. 114. International Association of Geodesy Symposia. Springer Berlin Heidelberg, 1995, S. 51–51. ISBN: 978-3-540-59421-5. DOI: 10.1007/978-3-642-79824-5_13. URL: http://dx.doi.org/10.1007/978-3-642-79824-5_13.
- [GSS10] Marco Guerini, Carlo Strapparava und Oliviero Stock. “Evaluation Metrics for Persuasive NLP with Google AdWords.” In: *LREC*. 2010.
- [GT10] Jean-François Girres und Guillaume Touya. “Quality Assessment of the French OpenStreetMap Dataset”. In: *Transactions in GIS* 14.4 (2010), S. 435–459.
- [Gut+04] Carl Gutwin, Steve Benford, Jeff Dyck, Mike Fraser, Ivan Vaghi und Chris Greenhalgh. “Revealing delay in collaborative environments”. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM. 2004, S. 503–510.
- [Hak10] Mordechai Haklay. “How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. Environment and planning. B, Planning & design 37.4 (2010): 682.” In: (2010).
- [Ham14] Stadt Hamburg. *Hamburg Open Data*. abgerufen am: 11.02.2014. 2014. URL: <http://daten.hamburg.de/>.
- [Hei14] Gerrit Heinemann. “Location Based Services als Basisfaktor Nr. 2 des SoLoMo”. German. In: *SoLoMo - Always-on im Handel*. Springer Fachmedien Wiesbaden, 2014, S. 65–118. ISBN: 978-3-658-03967-7. DOI: 10.1007/978-3-658-03968-4_3. URL: http://dx.doi.org/10.1007/978-3-658-03968-4_3.
- [HKH13] Robert Hecht, Carola Kunze und Stefan Hahmann. “Measuring Completeness of Building Footprints in OpenStreetMap over Space and Time”. In: *ISPRS International Journal of Geo-Information* 2.4 (2013), S. 1066–1091.
- [HLR11] Wenbo He, Xue Liu und Mai Ren. “Location cheating: A security challenge to location-based social network services”. In: *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*. IEEE. 2011, S. 740–749.
- [Hol11] Anthony T Holdener. *HTML5 Geolocation*. O’Reilly Media, Inc., 2011.
- [Jea03] Marc Jeannerod. “The mechanism of self-recognition in humans”. In: *Behavioural brain research* 142.1 (2003), S. 1–15.

- [Kap12] Karl Kapp. *The Gamification of Learning and Instruction. Game-Based Methods and Strategies for Training and Education*. Pfeiffer, 2012.
- [Kar14] Peter Karich. “GraphHopper Maps: Fast Road Routing in 100-Percent Java”. In: *Java Magazone* 1/2014 (2014), S. 71–73.
- [Ker13] Johanna Schockemöhle Kerstin Neeb Ulrike Ohl. *Hochschullehre in der Geographiedidaktik: Wie kann die Ausbildung zukünftiger Lehrerinnen und Lehrer optimiert werden?* Hrsg. von Kerstin Neeb. Bd. 7. Gießener geographische Manuskripte. Aachen: Shaker, 2013. ISBN: 9783844020120.
- [KM05] Peter Kiefer und Sebastian Matyas. “THE GEOGAMES TOOL: BALANCING SPATIO-TEMPORAL DESIGN PARAMETERS IN LOCATION-BASED GAMES”. In: (2005).
- [KMS05] Peter Kiefer, Sebastian Matyas und Christoph Schlieder. “State space analysis as a tool in the design of a smart opponent for a location-based game”. In: *Proceedings of the Games Convention Developer Conference “Computer Science and Magic”, Leipzig, Germany*. 2005.
- [KMS06] Peter Kiefer, Sebastian Matyas und Christoph Schlieder. “Systematically Exploring the Design Space of Location-based Games”. In: (2006).
- [KMS07a] Peter Kiefer, Sebastian Matyas und Christoph Schlieder. “Playing Location-based Games on Geographically Distributed Game Boards”. In: (2007).
- [KMS07b] Peter Kiefer, Sebastian Matyas und Christoph Schlieder. “Playing on a line: Location-based games for linear trips”. In: (2007).
- [LE07] Benjamin Livshits und Úlfar Erlingsson. “Using web application construction frameworks to protect against code injection attacks”. In: *Proceedings of the 2007 workshop on Programming languages and analysis for security*. ACM. 2007, S. 95–104.
- [Lei12] Steven Leigh. *Smart Insurers turn to Gamification as a way to Change Agent Behavior*. abgerufen am: 11.02.2014. 2012. URL: <http://www.infosysbpo.com/offerings/industries/insurance/Documents/insurance-journal-2012.pdf>.
- [Lin+11] Janne Lindqvist, Justin Cranshaw, Jason Wiese, Jason Hong und John Zimmerman. “I’m the mayor of my house: examining why people use foursquare-a social-driven location sharing application”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2011, S. 2409–2418.
- [Lui14] Lui. *Ingress Survey 2014*. Abgerufen am: 04.03.2014. 2014. URL: <http://goo.gl/ljQkqx>.

- [Mag07] Carsten Magerkurth. *Pervasive gaming applications*. Bd. 2. A reader for pervasive gaming research. Aachen: Shaker, 2007. ISBN: 9783832262242.
- [Mal91] DH Maling. "Coordinate systems and map projections for GIS". In: *Geographical Information Systems: Principles and Applications*. John Wiley & sons (1991), S. 135–146.
- [Man12a] Andrea Mannara. "Location-based games and the use of GIS information: Design of a DSL for (re) locating a pervasive game. Diss. Norwegian University of Science and Technology, 2012." In: (2012).
- [Man12b] Claudia Manns. *Gamification zur Steigerung der Mitarbeitermotivation*. 2012.
- [Mar06] Matthew Chalmers Louise Barkhuus Malcolm Hall Scott Sherwood Paul Tennent Barry Brown Duncan Rowland Steve Benford Alastair Hampshire Mauricio Capra Marek Bell. "Interweaving Mobile Games With Everyday Life". In: (2006).
- [Mar11] Breuer Markus. "Was ist Gamification?" In: (2011). URL: <http://intelligent-gamification.de/2011/05/11/was-ist-gamification/>.
- [Mar13] A. Marczewski. *Gamification: A Simple Introduction*. Andrzej Marczewski, 2013. ISBN: 9781471798665. URL: <http://books.google.de/books?id=I0u9kPjldYC>.
- [Mat11] Sebastian Matyas. "Gemeinschaftliche Qualitätsgesicherte Erhebung und Semantische Integration von Raumbezogenen Daten". Diss. 2011.
- [May10] Frans Mayra. *An introduction to game studies: Games in culture*. Repr. Los Angeles: SAGE, 2010. ISBN: 9781412934466.
- [MM07] Victoria Mallinckrodt und Dick Mizerski. "The effects of playing an advergame on young children's perceptions, preferences, and requests". In: *Journal of Advertising* 36.2 (2007), S. 87–100.
- [Mon05] Markus Montola. "Exploring the edge of the magic circle: Defining pervasive games". In: *Proceedings of DAC*. 2005, S. 103.
- [MSW05] Mark Maybury, Oliviero Stock und Wolfgang Wahlster. *Intelligent technologies for interactive entertainment: First international conference, INTETAIN 2005, Madonna di Campaglio, Italy, November 30 - December 2, 2005 ; proceedings*. Bd. 3814. Lecture notes in computer science Lecture notes in artificial intelligence. Berlin: Springer, 2005. ISBN: 3540305092. URL: <http://www.springerlink.com/openurl.asp?genre=issue&issn=0302-9743&volume=3814>.

- [MSW09] Markus Montola, Jaakko Stenros und Annika Waern. *Pervasive games: Theory and design ; [experiences on the boundary between life and play]*. Morgan Kaufmann game design books. Amsterdam: Elsevier/Morgan Kaufmann, 2009. ISBN: 978-0-123-74853-9. URL: <http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10365016>.
- [MTS13] Torben Meyer, Matthias Trojahn und Steffen Strassburger. "Using crowd-sourced geographic information from OpenStreetMap for discrete event simulation of logistic systems". In: *Proceedings of the 46th Annual Simulation Symposium*. Society for Computer Simulation International. 2013, S. 2.
- [Nac12] Deutsche Wirtschafts Nachrichten. *Einzelhandel mit stärkstem Umsatzeinbruch seit vier Jahren*. abgerufen am: 11.02.2014. 2012. URL: <http://deutsche-wirtschafts-nachrichten.de/2012/11/30/einzelhandel-mit-staerkstem-umsatzeinbruch-seit-vier-jahren-2/>.
- [Nel02] Michelle R Nelson. "Recall of brand placements in computer/video games". In: *Journal of advertising research* 42.2 (2002), S. 80–92.
- [NKY04] Michelle R Nelson, Heejo Keum und Ronald A Yaros. "Advertainment or adcreep game players' attitudes toward advertising and product placements in computer games". In: *Journal of Interactive Advertising* 5.1 (2004), S. 3–21.
- [Nur+09] Nurzhan Nurseitov, Michael Paulson, Randall Reynolds und Clemente Izurieta. "Comparison of JSON and XML Data Interchange Formats: A Case Study." In: *Caine* 9 (2009), S. 157–162.
- [Ohl14] Ohloh. *Comparision of OpenLayers and Leaflet*. Abgerufen am: 20.03.2014. 2014. URL: http://www.ohloh.net/p/compare?project_0=OpenLayers&project_1=Leaflet.
- [Olb] Roland Olbricht. *Overpass API Output Formats*. Abgerufen am: 18.03.2014. URL: http://overpass-api.de/output_formats.html.
- [Ope13a] Openstreemaps. *64-bit Identifiers*. Abgerufen am: 03.03.2014. 2013. URL: http://wiki.openstreetmap.org/wiki/64-bit_Identifiers.
- [Ope13b] Openstreemaps. *Active Contributors until 2013-12*. 2013. URL: http://wiki.openstreetmap.org/wiki/File:Active_contributors_month_201312.png.
- [Ore07] Tim O'reilly. "What is Web 2.0: Design patterns and business models for the next generation of software." In: *Communications & strategies* 65 (2007).

- [Oxf13] Oxford. "Oxford Dictionary 2013". In: (2013).
- [Pap06] Lothar Papula. *Mathematische Formelsammlung für Ingenieure und Naturwissenschaftler*. Bd. 7. Springer, 2006.
- [Pfo+13] Dieter Pfoser, Agnès Voisard, Jacinto Estima und Marco Painho. "Exploratory analysis of OpenStreetMap for land use classification". In: *the Second ACM SIGSPATIAL International Workshop*. 2013, S. 39–46.
- [PNZ11] Dennis Zielstra Pascal Neis und Alexander Zipf. "The Street Network Evolution of Crowdsourced Maps: OpenStreetMap in Germany 2007–2011". In: *Future Internet; Volume 4; Issue 1; Pages 1-21* (2011).
- [Pos11] Stefan Poslad. *Ubiquitous Computing: Smart Devices, Environments and Interactions*. 2. Aufl. s.l: Wiley, 2011. ISBN: 9780470035603. URL: http://ebooks.ciando.com/book/index.cfm/bok_id/875100.
- [PTVV02] Martin Pol, Ruud Teunissen und Erik Van Veenendaal. *Software testing: a guide to the TMap approach*. Pearson Education, 2002.
- [Qiu02] HC Qiuhui. "Study on mvc model2 and struts framework". In: *Computer Engineering* 6 (2002), S. 109.
- [Ram12] Roland Ramthun. "Offene Geodaten durch OpenStreetMap". In: *Open Initiatives: Offenheit in der digitalen Welt und Wissenschaft* (2012), S. 159.
- [Ras+06a] Omer Rashid, Will Bamford, Paul Coulton, Reuben Edwards und Jurgen Scheible. "PAC-LAN: Mixed-reality Gaming with RFID-enabled Mobile Phones". In: *Comput. Entertain.* 4.4 (Okt. 2006). ISSN: 1544-3574. DOI: 10.1145/1178418.1178425. URL: <http://doi.acm.org/10.1145/1178418.1178425>.
- [Ras+06b] Omer Rashid, Mullins Ian, Coulton Paul und Edwards Reuben. "Extending Cyberspace:Location Based Games Using Cellular Phones". In: *(2006) Extending Cyberspace:Location Based Games Using Cellular Phones. Computers in Entertainment (CIE), 4 (1). pp. 1-18. ISSN 1544-3574* (2006).
- [Rös05] Gerhard Rösl. "Regionalwährungen in Deutschland". German. In: *Wirtschaftsdienst* 85.3 (2005), S. 182–190. ISSN: 0043-6275. DOI: 10.1007/s10273-005-0354-2. URL: <http://dx.doi.org/10.1007/s10273-005-0354-2>.
- [Sal12] Simon Salt. *Social Location Marketing: Erreichen Sie Ihre Kunden mit Lokalisierungsdiensten*. Always learning. München u.a: Addison Wesley, 2012. ISBN: 9783827331083.

- [Sch+01] D. Schwabe, L. Esmeraldo, Gustavo Rossi und F. Lyardet. “Engineering Web applications for reuse”. In: *MultiMedia, IEEE* 8.1 (Jan. 2001), S. 20–31. ISSN: 1070-986X. DOI: 10.1109/93.923950.
- [Sch01] Christina Schmitt. “Chancen für Loyalitätsprogramme durch das Internet: das Beispiel Lufthansa Miles & More”. German. In: *Effektives Customer Relationship Management*. Hrsg. von Stefan Helmke und Wilhelm Dangelmaier. Gabler Verlag, 2001, S. 85–99. ISBN: 978-3-409-11767-8. DOI: 10.1007/978-3-322-82348-9_4. URL: http://dx.doi.org/10.1007/978-3-322-82348-9_4.
- [Sch02] Philip J Schoeneberger. *Field Book for Describing and Sampling Soils, Version 3.0*. Government Printing Office, 2002.
- [Sch13] Christoph Schlieder. “Geogames – Gestaltungsaufgaben und geoinformatische Lösungsansätze: unveröffentlichtes Manuskript, Universität Bamberg”. In: (2013).
- [Seb08] Christian Matyas Christoph Schlieder Peter Kiefer Sebastian Matyas. “CityExplorer - A Geogame Extending the Magic Circle”. In: (2008).
- [SKM05] Christoph Schlieder, Peter Kiefer und Sebastian Matyas. “Geogames: A Conceptual Framework and Tool for the Design of Location-Based Games from Classic Board Games”. In: (2005).
- [SKM06] Christoph Schlieder, Peter Kiefer und Sebastian Matyas. “Geogames: Designing Location-Based Game Geogames: Designing Location-Based Games from Classic Board Location-Based Ga from Classic Board Games from Cl Games”. In: (2006).
- [Sto03] Knut Stolze. “SQL/MM Spatial-The Standard to Manage Spatial Data in a Relational Database System.” In: *BTW*. Bd. 2003. 2003, S. 247–264.
- [SZ10] Katie Salen und Eric Zimmerman. *Rules of play: Game design fundamentals*. [Nachdr.] Cambridge und Mass: The MIT Press, 2010. ISBN: 0-262-24045-9.
- [TB06] Winkler Tina und Kathy Buckner. “Receptiveness of gamers to embedded brand messages in advergames: Attitudes towards product placement”. In: *Journal of Interactive Advertising* 7.1 (2006), S. 3–32.
- [Tes12] Tesco. *Tesco Homeplus expands number of virtual stores*. Abgerufen: 05.03.2014. 2012. URL: <http://www.tescopl.com/index.asp?pageid=17&newsid=593>.

- [TH06] Bruce A Tate und Curt Hibbs. *Ruby on Rails: Up and Running: Up and Running*. O'Reilly Media, Inc., 2006.
- [VZKB06] Pieter Van Zyl, Derrick G Kourie und Andrew Boake. "Comparing the performance of object databases and ORM tools". In: *Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*. South African Institute for Computer Scientists und Information Technologists. 2006, S. 1–11.
- [Wag05] Oliver Wagner. "Kundenbindung: Miles & More — Kundenbindung in der Luft". In: *Handbuch Kundenzufriedenheit*. Springer Berlin Heidelberg, 2005, S. 135–153. ISBN: 978-3-540-21144-0. DOI: 10.1007/3-540-27050-7_8. URL: http://dx.doi.org/10.1007/3-540-27050-7_8.
- [Wei+11] Joel Weinberger, Prateek Saxena, Devdatta Akhawe, Matthew Finifter, Richard Shin und Dawn Song. "A systematic analysis of xss sanitization in web application frameworks". In: *Computer Security—ESORICS 2011*. Springer, 2011, S. 150–171.
- [Wie14] Stadt Wien. *Wien Open Data*. abgerufen am: 11.02.2014. 2014. URL: <https://open.wien.at>.
- [Wik14] Wikipedia. *World Wide Smartphone Sales Share - Diagram based on Gartner*. Abgerufen am: 05.03.2014. 2014. URL: http://en.wikipedia.org/wiki/File:World_Wide_Smartphone_Sales_Share.png.
- [WWW14] WWWtechs. *Usage of server-side programming languages for websites*. Abgerufen am: 20.03.2014. 2014. URL: http://w3techs.com/technologies/overview/programming_language/all.
- [YR05] Jeff Yan und Brian Randell. "A systematic classification of cheating in online games". In: *Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games*. ACM. 2005, S. 1–9.
- [ZC11] Gabe Zichermann und Christopher Cunningham. *Gamification by design: Implementing game mechanics in web and mobile apps. - Publisher from cover*. S.l: O'Reilly, 2011. ISBN: 978-1-449-39767-8.
- [ZL13] Gabe Zichermann und Joselin Linder. *The gamification revolution: How leaders leverage game mechanics to crush the competition*. New York und NY: McGraw Hill Education, 2013. ISBN: 9780071808316.
- [Zoo+10] Matthew Zook, Mark Graham, Taylor Shelton und Sean Gorman. "Volunteered geographic information and crowdsourcing disaster relief: a case study of the Haitian earthquake". In: *World Medical & Health Policy* 2.2 (2010), S. 7–33.

Eidestattliche Erklärung

Ich erkläre hiermit gemäß §17 Abs. 2 APO, dass ich die vorstehende Masterarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

29. März 2014

(Datum)

(Unterschrift)