

Mobile ortsbasierte Browserspiele

Andreas Brodt, Christoph Stach
IPVS Universität Stuttgart, Universitätsstraße 38, 70569 Stuttgart
{ andreas.brodt | christoph.stach } @ ipvs.uni-stuttgart.de

Abstract: Die Verbreitung von mobilen Geräten und Positionssensoren wie GPS ermöglicht mobile ortsbasierte Spiele, in denen sich die Spieler physisch in ihrer Umwelt bewegen müssen, um Einfluss auf das Spielgeschehen zu nehmen. Klassische Computerspiele werden zunehmend als Browserspiel realisiert, d.h. der Web-Browser des Spielers wird für die Benutzerschnittstelle verwendet. Indem der Web-Browser um eine Kontextschnittstelle erweitert wird, kann einem Browserspiel Zugriff auf die aktuelle Position des Spielers gewährt werden. Dadurch wird es möglich, mobile ortsbasierte Spiele im Web-Browser zu spielen.

In diesem Papier beschäftigen wir uns mit den Eigenschaften mobiler ortsbasierter Browserspiele. Wir stellen zwei Beispiele vor, anhand derer wir untersuchen, welche Einflüsse mobile ortsbasierte Spiele auf das Spielkonzept haben und welche technischen Konsequenzen sich daraus ergeben. Abschließend präsentieren wir ein Framework zur Entwicklung mobiler ortsbasierter Spiele.

1 Einleitung

Heutige mobile Geräte, in Verbindung mit Positionssensoren wie GPS, erfüllen längst alle technischen Voraussetzungen für mobile ortsbasierte Spiele, d.h. Computerspiele, in denen die Spieler agieren, indem ihre Bewegungen durch die reale Welt in die virtuelle Spielwelt übertragen werden. Somit verbinden mobile ortsbasierte Spiele klassische Spiele im Freien und Computerspiele. Mobile Spiele binden die Umgebung eines Spielers inklusive seiner Mitmenschen in ein virtuelles Spielerlebnis ein und schaffen so ein eigenes Spielgenre, das die Realität anreichert. Das Spiel *BotFighters* aus dem Jahr 2001 ist eines der ersten kommerziellen mobilen Spiele [Sot02]. Dabei bekämpfen sich die Spieler, repräsentiert durch Roboter, gegenseitig, müssen aber physisch in "Schussreichweite" sein. *Can you see me now?* [FBA⁺03] und *Uncle Roy All Around You* [BFD⁺04] sind Spiele, bei denen Teilnehmer am PC mit Spielern in der realen Welt zusammenarbeiten müssen. *REXplorer* dient der Wissensvermittlung, indem Touristen mit Smartphones und GPS ausgestattet in der Regensburger Altstadt "merkwürdige Phänomene" aufdecken, um so mehr über die Stadtgeschichte zu lernen [BKB⁰⁷]. Neben dem ausgefeilten Entwurf des Spielkonzepts und -inhalts ist all diesen Spielen auch ein nicht unerheblicher technischer Aufwand gemeinsam, da die Programmierung mobiler Geräte auch heute noch stark plattform- und geräteabhängig ist.

Im Bereich der klassischen Computerspiele war in den letzten Jahren ein Trend hin zu Browserspielen erkennbar, d.h. zu Computerspielen, die den Web-Browser des Spielers für die Benutzerschnittstelle nutzen. Die ersten Browserspiele bestanden aus rein textba-

sierten Benutzerschnittstellen, die den Browser lediglich als komfortableren Telnet-Client verwendeten. 1995 wurde mit *SOL* [Spo95] ein Browserspiel veröffentlicht, das sowohl mehrspielerfähig ist, als auch eine persistente Spielwelt grafisch darstellen kann. Im Zuge einer zunehmenden Professionalisierung in dieser Softwaresparte erschien im Jahr 2000 mit *Planetarion* von *Fifth Season AS* eines der ersten kommerziellen MMOGs (Massively Multiplayer Online Game) als Browserspiel. Mit der stetigen Verbreitung von Flash entstand eine Vielzahl von Spielen, die nicht selten ältere Computerspiele wie Tetris oder Pacman kopierten, nun aber ohne Installation schnell und spontan spielbar waren. Das Aufkommen von AJAX erlaubte zudem ausgefeilte Kommunikations- und Visualisierungstechniken im Browser, was komplexe Browserspiele ermöglichte. Spiele wie *Travianer* [Tra08] sind in vielen Belangen mit klassischen installierten Computerspielen vergleichbar. Während sich Computerspiele im Web verbreiteten, wurden auch bestehende Webseiten auf die Möglichkeiten von Browserspielen aufmerksam. So publizierte z.B. das soziale Netzwerk *Facebook* eine Applikationsschnittstelle, über die Dritte beliebige Webanwendungen in Facebook integrieren können, was oft für Spiele genutzt wird. Diese Spiele stehen somit einem Millionenpublikum zur Verfügung und können auf die Daten und sozialen Verbindungen der Spieler zurückgreifen.

Browserspiele bieten eine besonders niedrige Einstiegshürde für den Spieler. Sie können leicht auf andere Web-Resourcen zugreifen, wie Bilder, Karten oder soziale Netzwerke, und können die vielfältigen Kommunikationsmöglichkeiten des Web leicht nutzen. Darüber hinaus sind Browserspiele plattformunabhängig und daher von jedem PC aus spielbar, der über einen hinreichend aktuellen Web-Browser verfügt. Somit kann ein Spieler an verschiedenen Rechnern und Orten den Fortlauf eines MMOGs verfolgen. Für die Spielehersteller bieten Browserspiele durch ihre Plattformunabhängigkeit Vorteile bei der Entwicklung. Das Web erleichtert die Verbreitung eines Spiels. Außerdem haben Browserspiele, die auf den Servern ihrer Hersteller ablaufen, keine Probleme mit Raubkopien; für die kostenpflichtige Nutzung müssen sich die Spieler mit entsprechenden Benutzerkennungen am Server anmelden.

Aktuelle mobile Geräte, wie das iPhone von Apple oder die Internet Tablet-Serie von Nokia sind mit einem vollständigen Web-Browser ausgestattet, der sich von dem eines PCs kaum noch unterscheidet. Neuere Entwicklungen [BNSM08, Goo08] zielen darauf ab, Kontextdaten wie die GPS-Position über den Web-Browser eines mobilen Geräts an Webseiten weiter zu geben. Dadurch wird es möglich, mobile ortsbasierte Spiele als Browserspiel zu realisieren, was die Spielgenres der mobilen Spiele und der Browserspiele verbindet. *Mobile ortsbasierte Browserspiele* bringen die Vorteile der Browserspiele in die Domäne der mobilen Spiele und eröffnen somit neue Möglichkeiten, die es zu betrachten gilt.

In diesem Papier beschäftigen wir uns mit den speziellen Merkmalen mobiler ortsbasierter Browserspiele. Wir illustrieren dies anhand zweier konkreter Beispiele und leiten Anforderungen an die konzeptionelle und technische Entwicklung solcher Spiele ab. Darauf aufbauend stellen wir ein modulares Framework vor, das die Entwicklung ortsbasierter Browserspiele erleichtert.

2 Beispiele für mobile ortsbasierte Browserspiele

Wir implementierten prototypisch zwei mobile ortsbasierte Browserspiele, anhand derer wir dieses Spielgenre veranschaulichen: TREASURECACHE und T^4 – *TicTacToe in Teams*. TREASURECACHE ist ein an Geocaching angelehntes Suchspiel, das in das soziale Netzwerk Facebook integriert ist und sich dessen Ressourcen bedient. T^4 ist eine ortsbasierte Version des Pen-and-Paper-Spiels TicTacToe, das in zwei Teams gespielt wird. Als mobiles Gerät dient bei beiden Spielen ein Nokia N810 Internet Tablet, das sowohl über einen eingebauten GPS-Empfänger verfügt, als auch über einen Mozilla-basierten Web-Browser, der nahezu mit einem Desktop-Browser vergleichbar ist. Der Web-Browser des Geräts wurde in unseren früheren Arbeiten [BNSM08] um eine Kontextschnittstelle [WHR⁺07, Bro08] erweitert, die den Spielern Zugriff auf die aktuelle GPS-Position verschafft.

2.1 TREASURECACHE

TREASURECACHE ist ein ortsbasiertes Browserspiel für Facebook. Kern des Spiels sind *Challenges*, Herausforderungen einen bestimmten Ort aufzusuchen, die ein Spieler annimmt und lösen muss. Ein Facebook-Benutzer erzeugt eine solche Herausforderung mit Hilfe des web-basierten TREASURECACHE-Editors, der in Abbildung 1 dargestellt ist. Anschließend kann der Benutzer die Herausforderung an seine Facebook-Freunde schicken, die sich entscheiden können, die Herausforderung anzunehmen. Dazu muss der Spieler lediglich mit dem Browser seines mobilen Geräts dem Weblink folgen, der in der Einladungsnachricht enthalten ist. Er gelangt dadurch auf die TREASURECACHE-Spielseite, die die GPS-Position des Spielers ausliest und zusammen mit der Herausforderung auf einer Karte anzeigt. Befindet sich der Spieler im Gebiet der Herausforderung, so beginnt das Spiel.

Um eine Herausforderung zu lösen, müssen verschiedene *Tasks* erledigt werden, die sich jeweils an einen geographischen Ort befinden. Es wird unterschieden zwischen *Position Tasks*, deren Ort einfach nur gefunden werden muss, *Duration Tasks*, die innerhalb einer bestimmten Zeitspanne erreicht werden müssen, und *Question Tasks*, an denen der Spieler eine Multiple-Choice-Frage beantworten muss. Mit Ausnahme der Zieltasks am Ende einer Herausforderung, besitzen alle Tasks ein oder mehrere Folgetasks. Dadurch ergibt sich ein Netz, in dem sich der Spieler bewegt. Die Folgetasks hängen davon ab, wie gut der Spieler den Task gelöst hat. Hat der Herausforderer die Folgetasks geschickt platziert, so wird der Spieler über einen kürzeren oder längeren Pfad zu einem Zieltask geleitet.

Der Spieler hat die Herausforderung gemeistert, wenn er einen Zieltask erreicht hat und keine der optionalen KO-Kriterien wie Zeitüberschreitung oder falsche Antworten erfüllt. Die TREASURECACHE-Spielseite stoppt die Zeit, die der Spieler für die Herausforderung benötigt. Darüber hinaus kann der Spieler Punkte sammeln, wenn er einen Task gut löst. Das erreichte Ergebnis wird im Facebook-Profil des Spielers zusammen mit einem Ranking veröffentlicht. Außerdem sind die erzeugten Herausforderungen, die ein Facebook-Benutzer erzeugt hat, in seinem Profil zu sehen.

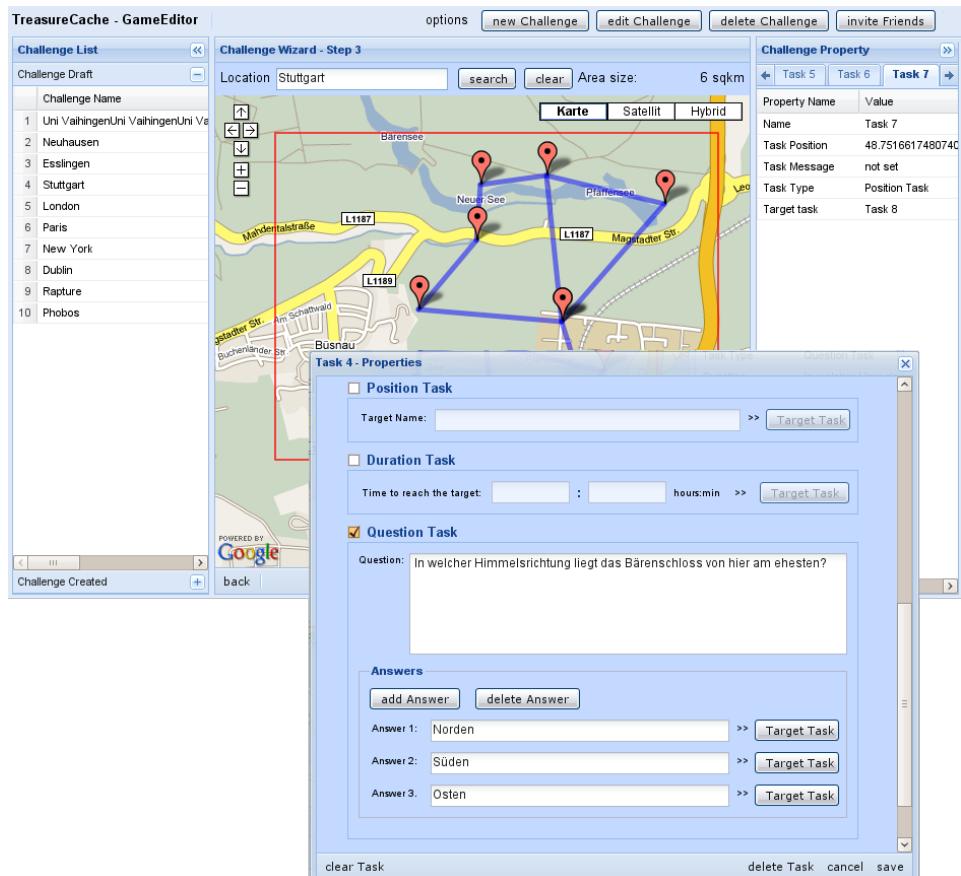


Abbildung 1: Der TREASURECACHE-Editor

Eine Grundidee des Konzepts von TREASURECACHE liegt darin, dass der Spieler sich an den Ort einer Herausforderung begeben muss, aber nicht zwingend der Herausforderer. Die Herausforderungen können im TREASURECACHE-Editor an einem beliebigen Ort platziert werden. Dadurch ist es möglich, auch weit entfernte Facebook-Freunde herauszufordern. Dies entspricht dem weit verteilten und nicht selten internationalen Charakter der sozialen Netzwerke, deren Popularität sich ja nicht zuletzt darauf begründet, dass sie es erleichtern, geographisch weit entfernte Bekanntschaften zu pflegen.

2.2 T^4 – TicTacToe in Teams

Bei dem Spiel *TicTacToe* handelt es sich um eines der bekanntesten Pen-and-Paper-Spielen der Welt. Zwei Spieler belegen abwechselnd Felder eines drei mal drei Felder großen

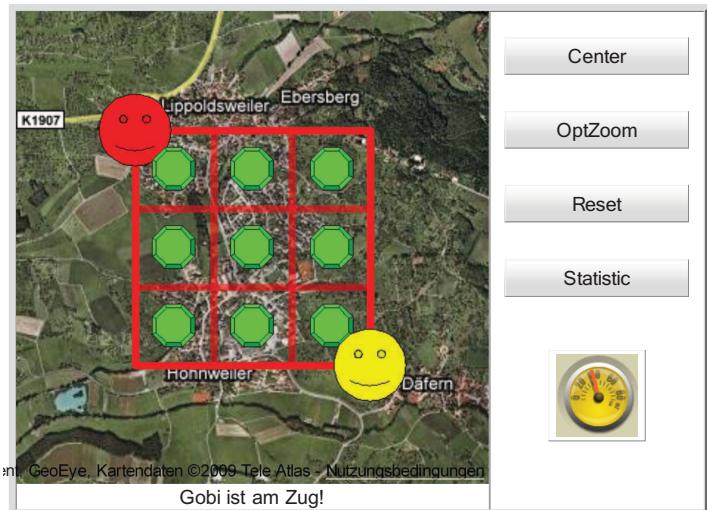


Abbildung 2: Screenshot einer Partie T^4

Spielfeldes, bis ein Spieler eine komplette Zeile, Spalte oder Diagonale besetzt hat. T^4 – *TicTacToe in Teams* setzt diese Spielidee in einem mobilen ortsbasierten Browserspiel um. Dabei benutzt T^4 die geographische Umgebung der Spieler als Spielfeld und erweitert die Spielerzahl auf acht Spieler in zwei Teams. Die Oberfläche von T^4 ist in Abbildung 2 dargestellt.

Haben sich mindestens acht Spieler auf der T^4 -Webseite für ein Spiel angemeldet, so werden diese in zwei Teams eingeteilt und ein Spielfeld wird aufgebaut. Das Zentrum des Spielfeldes wird dabei automatisch, abhängig von der Anfangsposition der Teilnehmer mittig positioniert und die restlichen Felder entsprechend darum herum verteilt. Dadurch ist es möglich, T^4 an jedem beliebigen Ort zu spielen, ohne dass im Vorfeld eine Spielwelt manuell erstellt werden muss – es sollte natürlich darauf geachtet werden, dass das Spielfeld begehbar ist. Die Startpositionen der Teams befinden sich in den entgegengesetzten Eckpunkten im Nordwesten und Südosten, die in den jeweiligen Teamfarben markiert werden. Sobald sich die Teams dort eingefunden haben, beginnt das Spiel.

Die Teams sind abwechselnd am Zug. Dabei wählen sie ein freies Feld, welches somit *aktiviert*, aber noch nicht besetzt wird. Das Team, das zuerst den Mittelpunkt des aktivierte Feldes erreicht, besetzt das Feld. D.h. wenn das gegnerische Team selbst Interesse am aktivierte Feld hat oder verhindern will, dass das Team am Zug das Feld besetzt, so kann es einen ihrer Spieler zum Feld schicken. Verlässt ein Spieler ein Feld vor Spielende, so verliert das Team dieses Feld wieder. Das Feld kann dann erneut aktiviert und besetzt werden. Ein Grund, ein Feld bewusst zu verlassen, könnte sein, dass alle Teammitglieder bereits ein Feld besetzt halten, das Team aber noch nicht gewonnen hat. Gleichsam kann sich die Spielsituation so verändert haben, dass das Feld seinen taktischen Nutzen für das Team verloren hat. Es kann aber auch durch schlechte teaminterne Absprachen zu einer versehentlichen Aufgabe eines Feldes kommen.

Wie man an diesen taktischen Möglichkeiten erkennen kann, ist es für die Teams unumgänglich, sich genau abzusprechen; sei es wenn ein Feld aktiviert wird, als auch bei der Wahl des Läufers, der das Feld besetzen soll. Die Teams müssen dieses Kommunikationsproblem selbst lösen, indem sie beispielsweise über Chat oder Mobiltelefon Kontakt zu ihren Teamkollegen halten.

3 Eigenschaften mobiler ortsbasierter Browserspiele

Mobile ortsbasierte Browserspiele wie die in Abschnitt 2 vorgestellten Beispiele haben eine Reihe spezieller Eigenschaften, die sich auf das Konzept eines Spiels auswirken, als auch technische Konsequenzen nach sich ziehen.

3.1 Einflüsse auf das Spielkonzept

Nutzung von Web-Resourcen Indem ein Spiel im Web abläuft, kann es die mannigfaltigen Resourcen des Webs benutzen. So greift TREASURECACHE neben den Facebook-Benutzern und ihren Kontakten auch auf Google Maps zurück. Bestehende Benutzerauthentifizierungssysteme können mitbenutzt werden, wie es TREASURECACHE ebenfalls praktiziert. Außerdem kann das Web zur Kommunikation genutzt werden. Ein TREASURECACHE-Spieler kann, während er eine Herausforderung löst, über den Facebook-Chat mit dem Herausforderer kommunizieren oder im Anschluss seine Erfahrung per Facebook-Nachricht mitteilen. Da ortsbasierte Browserspiele in der realen Welt ablaufen, können auch insbesondere Web-Resourcen über die Umgebung des Spielers benutzt werden, beispielsweise geographisch annotierte Bilder oder Lexikonartikel über die Umgebung. Das (nicht browserbasierte) Spiel ECHOES [DMC+08] nutzt beispielsweise das Fotoportal Flickr, um Bilder der Spielumgebung auszutauschen.

Diskontinuität Im Gegensatz zu klassischen Computerspielen kann bei mobilen ortsbasierten Browserspielen nicht davon ausgegangen werden, dass der Spieler sich über längere Zeit ohne Unterbrechung mit dem Spiel beschäftigt. Wie allgemein bei mobilen Spielen, kann der Spieler aufgrund von Umgebungseinflüssen wie z.B. dem Straßenverkehr oft gezwungen sein, seine Aufmerksamkeit vom Spiel abzuwenden. Auch können technische Faktoren wie ein leerer Akku, schlechter GPS-Empfang oder abbrechende Konnektivität das Spiel ungewollt unterbrechen oder ganz beenden. Im Vergleich zu einer auf dem mobilen Gerät installierten Anwendung ist es für eine Webseite schwer, dabei den Spielzustand zu behalten, da sie kaum Daten lokal auf dem Gerät speichern kann. Browsererweiterungen wie Gears [Goo08] könnten von technischer Seite das Problem der Diskontinuität entschärfen. Man wird aber beim Entwurf eines Spiels nicht umhin kommen, sich mit dem Problem auseinander zu setzen und das Spiel so zu gestalten, dass es auch unterbrochen werden kann und Ausfälle des Lokalisierungssystems oder der Konnektivität verkraftet. Ein Spiel kann dazu z.B. in mehrere kurze Episoden aufgeteilt werden, wie die Tasks von

TREASURECACHE. Auch kann ein geschickt gewähltes *Sicherungskonzept* als Gestaltungsmittel bei Unterbrechungen eingesetzt werden. Hierzu kommen neben der vollständigen und permanenten Möglichkeit, den Spielzustand zu sichern, auch eine Fortschrittsicherung nur an bestimmten geographischen, zeitlichen oder vom Spielzustand bestimmten Sicherungspunkten in Frage. Gleichsam kann auch beim Entwurf eines Spiels bewusst auf Sicherungen verzichtet werden.

Eingabemöglichkeiten Mobile Spiele erfordern mobile Geräte, deren Eingabemöglichkeiten trotz ständiger Verbesserung nicht mit denen eines PCs vergleichbar sind. Hinzu kommt die Ausführung im Web-Browser, was die Eingabemöglichkeiten weiter einschränkt. Beim Entwurf eines mobilen ortsbasierten Browserspiels sollte daher darauf geachtet werden, dass die Spieler mit einfachen Bedienaktionen auskommen. Gleichzeitig verfügt ein ortsbasiertes Browserspiel über die Benutzerposition und evtl. andere Kontextinformationen, und kann außerdem die Zeit mitverfolgen. Ein Spiel sollte diese Daten zur Steuerung verwenden. TREASURECACHE erfordert lediglich vom Spieler, dass er sich an die Positionen der Tasks begibt und Multiple-Choice-Fragen beantwortet, was mit sehr wenigen Klicks erledigt wird. Die Spieler von T⁴ steuern nur, indem sie Felder aktivieren und sich auf dem Spielfeld bewegen.

Kommunikationsmöglichkeiten Wie das Beispiel T⁴ zeigt, ist es vor allem bei teambasierten Spielen wichtig, den Spielern geeignete Kommunikationsmöglichkeiten zu bieten, beispielsweise einen Echtzeit-Chat. Diese Anforderung existiert auch bei nicht browserbasierten Spielen wie *Can you see me now?* Hinzu kommt, dass sich ein erfolgreiches Spiel durch eine aktive *Community* auszeichnet, die durch Interaktionsmöglichkeiten, wie einem Forum, bei Laune gehalten werden muss. Das Web bietet als Plattform besonders einfache Integration solcher Kommunikationsmittel.

Nutzergenerierte Inhalte Im Zuge des Web 2.0-Trends [O'R05] wurde das Prinzip der nutzergenerierten Inhalte im Web stark thematisiert. Engagierte Computerspieler bereicherten schon lange vorher ihre Lieblingstitel um zusätzliche Spielstufen, Charaktere oder sonstige Artefakte. Ein Spiel im Web kann es den Spielern allerdings besonders einfach machen, selbst Inhalte einzubringen und anderen Mitspielern verfügbar zu machen. Durch ihren geographischen Bezug eignen sich gerade mobile ortsbasierte Spiele sehr gut für solche nutzergenerierten Inhalte, da jeder Spieler Szenarien aus seiner persönlichen Umgebung einbringen kann. Bei einer hinreichend aktiven Community kann ein Spiel dadurch eine Abdeckung erreichen, die ein einzelnes Team von Spieleentwicklern nie stemmen könnte. Das Konzept des Spiels muss dies allerdings antizipieren [WG08]. TREASURECACHE baut beispielsweise mit seinen Herausforderungen gezielt auf nutzergenerierten Inhalten auf.

3.2 Technische Einflüsse

Kontextschnittstelle Mobile ortsbasierte Browserspiele werden erst möglich, wenn das Spiel als Web-Applikation die Position des Benutzers beziehen und verarbeiten kann. Hierzu gibt es verschiedene technische Möglichkeiten. Wir halten eine JavaScript-Schnittstelle für die sinnvollste, da dies dem AJAX-Ansatz sehr entgegen kommt und auch Szenarien mit hoher Dynamik gestattet. Zurzeit existieren mit W3C DCCI [WHR⁺07], der Geolocation API von Gears [Goo08] und anderen verschiedenen Browsererweiterungen, die Positions- und ggf. andere Kontextdaten in der JavaScript-Umgebung des Web-Browsers verfügbar machen. Bisher konnte sich keine von ihnen als allgemeiner Standard durchsetzen. Bis eine weit verbreitete (und idealerweise standardisierte) Kontextschnittstelle für Web-Applikationen auf vielen mobilen Geräten verfügbar ist, muss sich ein Spieleentwickler für eine bestimmte Lösung entscheiden, die vor allem vom den einzusetzenden mobilen Geräten abhängt. Für die in Abschnitt 2 vorgestellten Spiele benutzten wir Telar DCCI [Bro08], eine DCCI-Implementierung für Nokia Internet Tablets aus unseren früheren Arbeiten.

Spieloberfläche Wie bei den Möglichkeiten zur Benutzereingabe sind die Mittel für aufwändige graphische Oberflächengestaltung bei Browserspielen beschränkt¹. Auf mobilen Geräten, die auch heute noch über vergleichsweise wenig Rechenleistung verfügen, können resourcenintensive AJAX- und Flash-Techniken zum Problem werden [BNSM08]. Das Spiel sollte daher ohne aufwändige graphische Effekte auskommen. Wie der Erfolg einfacher Flash-Spiele zeigt, können Spiele auch ohne ausgefeiltes Grafiksystem den Spieler fesseln. Bei mobilen ortsbasierten Spielen ist die reale Welt Teil des Spiels, so dass die Graphik auf dem mobilen Gerät wenig bedeutsam ist.

Benutzerkonzept und -verwaltung Klassische installierte Computerspiele und einfache Flash-Spiele im Browser werden vom Benutzer einfach gestartet und gespielt. Ein komplexeres Browserspiel kommt dagegen nicht ohne ein Benutzerkonzept aus, über das Profildaten hinterlegt, erreichte Spiel- und Punktestände gesichert und nicht zuletzt auch Kosten abgerechnet werden können. Neben einer webtypischen Administration der Accountdaten stellt der Umgang mit Teams eine häufig verwendete Aufgabe dar. Jedes Team kann dieselben Eigenschaften eines Einzelspielers besitzen, aber darüber hinaus noch weitere Informationen über die Mitglieder und deren Rollen innerhalb des Teams. Teams sind allerdings stark dynamische Strukturen, die neue Mitglieder aufnehmen und von alten verlassen werden. Daher wird eine Benutzerverwaltung sowohl auf Benutzerebene als auch auf Teamebene benötigt.

Abhängig vom Konzept eines Spiels sollte erwogen werden, ob eine bestehende Benutzerverwaltung übernommen und ggf. erweitert werden kann. So kann beispielsweise eine Benutzeranmeldung mit OpenID [Ope07] als Authentifizierungsmechanismus Entwicklungskosten senken und die Sicherheit der Benutzerdaten erhöhen.

¹Das Spiel *Quake Live* (www.quakelive.com) ermöglicht sich mittels einer applikationsspezifischen Browsererweiterung eine aufwändige 3D-Oberfläche im Web-Browser, weicht dadurch aber vom Konzept einer plattformunabhängigen Webapplikation ab.

Einsatz von Drittsoftware Wie in Abschnitt 3.1 erläutert, ermöglicht das Web in besonderem Maße, externe Softwarekomponenten einzubinden. Beispiele sind Kommunikationsmöglichkeiten, eine Benutzerverwaltung, aber auch Flash-Multimediacomponenten wie eingebettete Minispiele oder Filmclips. Die Architektur eines Spiels muss allerdings dafür ausgelegt werden und hinreichend erweiterbar sein. Es sollte bedacht werden, dass Drittsoftware im Gegenzug tiefgreifende Auswirkungen auf die Architektur eines Spiels haben kann. TREASURECACHE bedient sich beispielsweise der Benutzerverwaltung und -authentifizierung sowie der Kommunikationsmöglichkeiten von Facebook. Dies hat zur Folge, dass TREASURECACHE nicht ohne einen Facebook-Zugang gespielt werden kann und dass bei Änderungen in der Facebook-API das Spiel angepasst werden muss.

Persistenz Bei Web-Anwendungen ist es im Browser i.d.R. nicht möglich, umfangreiche Datenobjekte lokal über die Browsersession hinaus persistent zu halten. Die Architektur des Spiels muss es vorsehen, dass entsprechend des im Spiel vorgesehenen Sicherungskonzepts (siehe Abschnitt 3.1) sämtliche Spielobjekte und -zustände vom Server geladen und wieder zurückgeschrieben werden können. Gleichsam ist es aber ratsam, große Teile dieser Objekte so weit wie möglich im Client zu halten, um das Spiel robust zu machen gegenüber temporären Unterbrechungen der Konnektivität. So lädt TREASURECACHE sofort bei Spielbeginn die Daten sämtlicher Tasks der gerade gespielten Herausforderung.

Editor Ein grafischer Editor, der es erlaubt, auf einfache Weise Spielinhalte zu erzeugen, erleichtert nicht nur die Spielentwicklung, um möglichst schnell die ersten testbaren Spielstufen erstellen zu können. Er kann auch nach Veröffentlichung eines Spiels von Bedeutung sein. Um nutzergenerierte Inhalte zu erleichtern (siehe Abschnitt 3.1) ist es ratsam, ein einfaches graphisches Werkzeug bereit zu stellen, das das Spiel um neue Inhalte bereichert. Um unnötige Hürden zu vermeiden, sollte der Editor wie auch das Spiel webbasiert sein, wie es der TREASURECACHE-Editor in Abbildung 1 zeigt.

4 Ein Framework für mobile ortsbasierte Browserspiele

Um die Entwicklung mobiler ortsbasierter Browserspiele zu erleichtern, haben wir ein Framework entworfen, das viele wiederkehrende Aufgaben übernimmt. Wir gehen von der Dreischichtarchitektur AJAX-basierter Web-Anwendungen aus: Ein Client, der mit JavaScript, HTML, CSS und ggf. Flash realisiert ist, eine serverseitige Logikschicht, sowie eine darunterliegende Persistenzschicht. Wir nehmen an, dass der Client über asynchrone HTTP-Anfragen mit dem Server kommuniziert. Außerdem setzen wir einen Web-Browser voraus, der über eine JavaScript-basierte Kontextschnittstelle verfügt.

Die Architektur unseres Frameworks ist in Abbildung 3 dargestellt. Kern des Frameworks ist eine generische Objektverwaltung, die sämtliche Entitäten, die in einem Spiel vorkommen, bereitstellt und persistiert. Dadurch müssen die konkreten Spielobjekte, z.B. Gold, Holz, Munition, Wegepunkte, usw., nur noch mit ihren Attributen modelliert werden. Neben erforderlichen Attributen, wie beispielsweise *Name*, *Preis* oder *Besitzer*, kann der

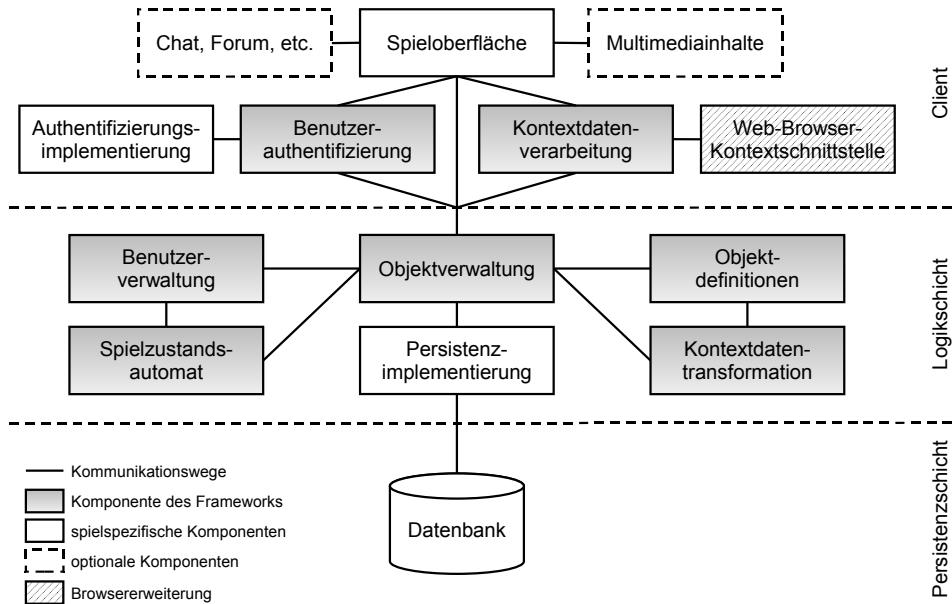


Abbildung 3: Architektur unseres Frameworks für mobile ortsbasierte Browserspiele

Spielentwickler jeden Objekttyp mit beliebig vielen weiteren Attribut-Wert-Kombinationen versehen. Für alle erwerblichen Gegenstände existiert ein Interface, das einen bedingten Austausch zwischen Spielern (z.B. durch Kauf oder Tausch) oder zwischen einem Spieler und der Spielwelt (z.B. Aufnahme oder Ablage) ermöglicht.

Auf der Objektverwaltung aufbauend stellt das Framework einen abstrakten Zustandsautomaten zur Verfügung. Der Zustandsautomat erlaubt es, die verschiedenen Zustände des Spiels (Anmeldung → Spielstart → ... → Spielende) mit seinen einzelnen Spielstufen zu modellieren, für jeden Zustand ausführbare Aktionen zu definieren und den Spielzustand zu speichern. Das Framework enthält ferner ein allgemeines Benutzerkonzept samt einer Schnittstelle für Authentifizierungsmechanismen. Für den Umgang mit Kontextdaten (i.d.R. Positionsdaten) stehen Funktionen wie Koordinatentransformationen und geometrische Prädikate zur Verfügung. Die Objektverwaltung übernimmt die Kommunikation zwischen Client und Server, indem sie eine Serverschnittstelle bereitstellt, über die die Spielobjekte ausgetauscht werden. Die Objektverwaltung reicht die Objekte an die entsprechenden Komponenten weiter oder speichert sie mit Hilfe der Persistenzschicht.

Das Framework ist modular aufgebaut, so dass es leicht an die jeweiligen Gegebenheiten eines Spiel angepasst werden kann und insbesondere die Einbindung von Drittsoftware unterstützt. Der Authentifizierungsmechanismus für die Benutzerverwaltung, sowie das Speichersystem der Persistenzschicht sind über entsprechende Schnittstellen abstrahiert und müssen durch eine spezifische Implementierung realisiert werden. Dadurch kann ein Spiel einfach auf ein externes Benutzerauthentifizierungssystem zurückgreifen oder das Speichersystem wechseln.

Die Benutzerschnittstelle ist bei jedem Spiel ein zentrales Unterscheidungs- und Alleinstellungsmerkmal. Deshalb lässt das Framework die Oberfläche komplett in der Hand des Spieleentwicklers, der die Komponenten des Frameworks in die Oberfläche einbindet. Gleichsam werden externe Komponenten, wie (Flash-) Multimediainhalte oder Kommunikationsmittel, bei Bedarf direkt in die Oberfläche integriert.

Um das Konzept des Frameworks zu evaluieren, implementierten wir es prototypisch unter Verwendung des Google Web Toolkits [Goo06], Java-Servlets, sowie von MySQL und Hibernate als Speichersystem. Wir setzten das Framework bei der Entwicklung von T⁴ erfolgreich ein und konnten dadurch den Entwicklungsaufwand erheblich senken.

5 Zusammenfassung

In diesem Papier stellten wir das Konzept mobiler ortsbasierter Browserspiele vor und illustrierten es anhand zweier Beispiele. Wir untersuchten die speziellen Eigenschaften mobiler ortsbasierter Browserspiele. Da sie über den Web-Browser eines mobilen Geräts gespielt werden, besitzen sie eine besonders niedrige Einstiegshürde für den Benutzer. Sie können leicht auf Web-Resourcen wie Bilder, Karten oder soziale Netzwerke zugreifen. Sie können Drittsoftware wie Chats und Authentifizierungssysteme nutzen, oder komplett in bestehende Webseiten integriert werden. Verglichen mit Spielen, die auf dem mobilen Geräten installiert sind, müssen mobile Browserspiele noch mehr mit Unterbrechungen des Spielflusses umgehen. Sie müssen sich auf einfachere graphische Darstellungsmöglichkeiten beschränken und ohne dauerhafte Datenhaltung auf dem mobilen Gerät auskommen. Schließlich präsentierten wir eine komponentenbasiertes Framework, das die Entwicklung mobiler ortsbasierter Browserspiele erleichtert.

Wir gehen davon aus, dass sich in naher Zukunft eine Kontextschnittstelle für mobile Web-Anwendungen am Markt durchsetzt und mobile Geräte mit entsprechenden Web-Browsern weite Verbreitung finden. Dadurch erwarten wir eine große potenzielle Spielerschaft für mobile ortsbasierte Browserspiele. Wir sehen darüber hinaus Bedarf für weitere Forschung in Bezug auf die Frage, wie neben der Position noch weitere Aspekte des Spielerkontexts zum Spielerlebnis mobiler Browserspiele beitragen können.

Danksagung

Wir bedanken uns bei unseren Kollegen Nazario Cipriani und Carlos Lübbe, sowie unseren Studenten Alexander Martin, Oleg Marin, Dominik Morar und Thomas Würfel für die Unterstützung unserer Arbeit. Außerdem danken wir Aarne Taube und Nokia Research Center für die Leihgabe von mobilen Geräten zur Entwicklung der vorgestellten Spiele.

Literatur

- [BFD⁺04] Steve Benford, Martin Flintham, Adam Drozd, Rob Anastasi, Duncan Rowland, Nick Tandavanitj, Matt Adams, Ju Row-Farr, Amanda Oldroyd und Jon Sutton. Uncle Roy All Around You: Implicating the City in a Location-Based Performance. In *Proc Advanced Computer Entertainment at ACE 2004*. ACM Press, 2004.
- [BKB⁺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: CHI '07 extended abstracts on Human factors in computing systems*, Seiten 1929–1934. ACM, 2007.
- [BNSM08] Andreas Brodt, Daniela Nicklas, Sailesh Sathish und Bernhard Mitschang. Context-Aware Mashups for Mobile Devices. In *Web Information Systems Engineering - WISE 2008 9th International Conference on Web Information Systems Engineering, Auckland, New Zealand, September 1-3, 2008, Proceedings*, Lecture Notes in Computer Science. Springer-Verlag, 2008.
- [Bro08] Andreas Brodt. Telar DCCI. <http://telardcci.garage.maemo.org>, 2008.
- [DMC⁺08] Karen Detken, Carlos Martinez, Darren Carlson, Varvara Guljajeva, Mari-Klara Oja und Andreas Schrader. ECHOES - A Crazy Multiplayer Pervasive Game. In Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach und Christian Scheideler, Hrsg., *GI Jahrestagung (1)*, Jgg. 133 of *LNI*, Seiten 489–494. GI, 2008.
- [FBA⁺03] Martin Flintham, Steve Benford, Rob Anastasi, Terry Hemmings, Andy Crabtree, Chris Greenhalgh, Nick Tandavanitj, Matt Adams und Ju Row-Farr. Where on-line meets on the streets: experiences with mobile mixed reality games. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, Seiten 569–576. ACM, 2003.
- [Goo06] Google Inc. Google Web Toolkit. <http://code.google.com/webtoolkit/>, 2006.
- [Goo08] Google Inc. Gears - Improving Your Web Browser. <http://gears.google.com/>, 2008.
- [Ope07] OpenID Foundation. OpenID. <http://openid.net/>, 2007.
- [O'R05] Tim O'Reilly. What is web 2.0 - Design Patterns and Business Models for the Next Generation of Software. <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>, 2005.
- [Sot02] Olli Sotamaa. All The World's A Botfighter Stage: Notes on Location-based Multi-User Gaming. In *Proceedings of the Computer Games and Digital Cultures Conference, June 6-8, 2002, Tampere, Finland*, Seiten 35–44, 2002.
- [Spo95] Alexander Spohr. SOL. <http://www.freepoert.de/Sol/>, 1995.
- [Tra08] Travian Games GmbH. Travianer. <http://www.travianer.de/>, 2008.
- [WG08] Stephan Wolff und Barbara Grüter. Context, emergent game play and the mobile gamer as producer. In Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach und Christian Scheideler, Hrsg., *GI Jahrestagung (1)*, Jgg. 133 of *LNI*, Seiten 495–500. GI, 2008.
- [WHR⁺07] Keith Waters, Rafah A. Hosn, Dave Raggett, Sailesh Sathish, Matt Womer, Max Froumentin, Rhys Lewis und Keith Rosenblatt. Delivery Context: Client Interfaces (DCCI) 1.0. W3C Candidate Recommendation, <http://www.w3.org/TR/DPF/>, 2007.