

Mobile Mixed-Reality Games

Studienarbeit

Dominik Schreiber



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fachbereich Elektrotechnik
und Informationstechnik
Fachbereich Informatik (Zweitmitglied)

Fachgebiet Multimedia Kommunikation
Prof. Dr.-Ing. Ralf Steinmetz

Ehrenwörtliche Erklärung

Hiermit versichere ich, die vorliegende Studienarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in dieser oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 16. Juli 2013

Dominik Schreiber

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Zielsetzung	1
1.3	Methodik	1
1.4	Aufbau der Arbeit	2
2	Technologien	3
2.1	Hardware	3
2.1.1	Ortung	3
2.1.2	Tags	5
2.1.3	Smartphones	8
2.1.4	Head-Mounted Displays	9
2.2	Software: Authoring Tools	11
3	Design-Elemente	14
3.1	Interaktion	14
3.2	Zeit und Ort	17
3.3	User-Generated Content	18
4	Fazit	20
4.1	Ergebnis	20
4.2	Ausblick	20
	Literaturverzeichnis	21

Abbildungsverzeichnis

2.1	Schematische Darstellung der Triangulation bei GPS (mit zweidimensionalen Kugeln)	4
2.2	QR-Code mit benötigten und freien Elementen	6
2.3	Oculus Rift – in endgültigem Aussehen, als Entwickler-Version in Aktion	10
2.4	Google Glass – Hardware, Navigationsinterface	11
2.5	TOTEM.Designer – Shape für Tidy City, außerdem ein passendes Marble mit Pebble	13
2.6	TOTEM.Scout – Pebble auf einer Kartenansicht, neu erstelltes Pebbles	13
3.1	Geocaching – ein Cache (Größe “small”), ein Trackable (Travel Bug)	15
3.2	Ingress – Karte, Portal, Inventar	17
3.3	Zombies, Run! – virtuelles Dorf des Spielers, Karte mit Position und Zombies, gesammelte Gegenstände nach abgeschlossener Mission	18
3.4	TidyCity – Karte mit Rätseln, in-situ Generierung von Content, Authoring Web-Anwendung	19

Zusammenfassung

Hintergrund

Moderne Technologien, wie beispielsweise Smartphones, machen Spiele möglich, die weit mehr als Eingabe verwenden, als Mausklicks und Tastenanschläge. Mit GPS und verschiedenen Tags ist es möglich, den Aufenthaltsort des Spielers zentral in ein Spiel einzubinden und ihm Informationen und Aktionen ortsgebunden zur Verfügung zu stellen. Solche Spiele bezeichnet man als Mobile Mixed-Reality Games.

Methodik

Ziel dieser Arbeit war, einen *Überblick über Technologien und Design-Elemente* zu geben, die in Mobile Mixed-Reality Games häufig verwendet werden und deren Entwicklung vereinfachen. Dazu wurde Literaturrecherche betrieben, die besonderen Wert auf die Arbeit der Forschungsgruppe TOTEM des Fraunhofer FIT [[WBO⁺11](#)] legte.

Ergebnis

Das heute meistbenutzte Setup für Mobile Mixed-Reality Games ist eine *Smartphone-Anwendung*, die über die darin enthaltenen Sensoren den Spieler ortet und ortsbezogen Informationen und Aktionen freischaltet. Spiele wie *Ingress*, *Zombies*, *Run!* oder *Tidy City* überlagern dazu die reale Welt mit Informationen einer parallelen Realität. Das reicht von wenigen versteckten Rätseln bei *Tidy City* bis zu großflächig verteilten außerirdischen Portalen bei *Ingress*.

Ebenfalls auffällig ist, dass fast alle Mobile Mixed-Reality Games in irgendeiner Form *User-Generated Content* erlauben. Vom Vorschlagen neuer Portale bei *Ingress* bis hin zu Datenbanken von neu erstellten Caches bei Geocaching oder Rätseln bei *Tidy City* – die Spielentwickler stellen das Spielprinzip, die Spielmechanik, die Community erstellt den Spielinhalt. Hier sind besondere Authoring-Tools wie das *TOTEM-Framework* nötig, die Authoring sowohl am Desktop-Rechner als auch in-situ erlauben.

Fazit

Mobile Mixed-Reality Games sind noch am Anfang ihrer Entwicklung. Neue Geräte und Interaktionsmöglichkeiten ermöglichen immer anspruchsvollere Anwendungen, die Realität und Fiktion immer mehr verschmelzen lassen.

1 Einleitung

1.1 Motivation

Seit dem Wegfall der Selective Availability des GPS im Jahr 2000 [ZG01] gibt es immer mehr Spiele, die den Ort des Spielers in das Spiel integrieren und teilweise zum zentralen Spielement erheben. So lassen sich neue Spielerfahrungen entwickeln, die weit über eine rein virtuelle Welt vor dem heimischen PC hinaus gehen.

Solche “Location-based Games” reichen derzeit vom sehr einfach gehaltenen *Geocaching* [O’H08] – einem Urvater aller Location-based Games – bis hin zu komplexen “Augmented-Reality Games” wie *Ingress*¹ oder *Zombies, Run!*².

Auch im Bereich der Serious Games gibt es Ansätze, den Faktor Ort in das Spielgeschehen einzubinden. So wird beispielsweise intensiv daran geforscht, Autoren Werkzeuge an die Hand zu geben, um Inhalte für ortsbasierte Spiele zu entwickeln. Ein besonderes Augenmerk fällt dabei auf sogenannte “Mobile Mixed-Reality Games”, also Spiele, in denen die reale Welt mit einer virtuellen Wirklichkeit verknüpft wird [WBO⁺¹¹].

1.2 Zielsetzung

Im Rahmen dieser Arbeit soll der aktuelle Stand der Forschung zu “Mobile Mixed-Reality Games”, insbesondere im Kontext von Serious Games, erarbeitet und dargelegt werden. Neben existierenden Theorien soll besonders auf bestehende praktische Anwendungen, wie realisierte Mobile Mixed-Reality Games oder zur Verfügung stehende Hardware, eingegangen werden.

Dies umfasst zwei Kernaspekte:

- *Existierende Technologien.* Einerseits: Welche Hardware eignet sich zum Einsatz bei Mobile Mixed-Reality Games? Wie lassen sich Realität und virtuelle Wirklichkeit verbinden? Andererseits: Gibt es Software, die die Entwicklung eines solchen Spiels erleichtert?
- *Existierende Design-Elemente.* “Woraus” werden Mobile Mixed-Reality Games gemacht? Welche wiederkehrenden Muster gibt es? Existieren versteckte Faktoren, auf die bei der Realisierung eines solchen Spiels besonders geachtet werden muss? Wie wurden diese Anforderungen bisher umgesetzt?

Diese und ähnliche Fragen sollen im Rahmen dieser Arbeit beantwortet werden, außerdem soll ein Ausblick auf mögliche Weiterentwicklungen und neue Forschungsfelder gegeben werden.

1.3 Methodik

Zu Beginn waren einige Quellen bezüglich der Virtual-Reality Brille *Oculus Rift* und der Forschungsgruppe zu Mobile Mixed-Reality Games *TOTEM Games* vorgegeben, und der Forscher *Leif Oppermann*

¹ <http://www.ingress.com>

² <https://www.zombiesrungame.com/>

([WBO⁺¹¹]) besonders hervorgehoben. Von diesen Quellen ausgehend wurden über Google Scholar³, die ACM⁴ und Springerlink⁵ weitere Arbeiten mit Suchanfragen nach "mobile mixed-reality games", "leif oppermann" oder "tidy city" gefunden. Andere Informationen, insbesondere über aktuelle Technologien, wurden in populären Technologiemagazinen wie Engadget⁶, Gizmodo⁷ und Kickstarter⁸ gefunden und zum jeweiligen Ursprungslink verfolgt.

Es war verhältnismäßig schwierig, wissenschaftliche Literatur zum hier betrachteten Thema zu finden, da Mobile Mixed-Reality Games topaktuell sind, und gerade verwendbare Technologien häufig nicht in wissenschaftlichen Arbeiten, sondern meist in Websites und Magazinartikeln besprochen werden. Das führt dazu, dass in dieser Arbeit häufig solche Websites als weiterführende Literatur angegeben sind.

1.4 Aufbau der Arbeit

In diesem 1. Kapitel wurde die Arbeit *motiviert*, ihr *Ziel* umrissen und die angewandte *Methodik* erläutert.

Das Kapitel 2 erläutert *Technologien*, die bei der Entwicklung von Mobile Mixed-Reality Games zur Verfügung stehen und die Entwicklung erleichtern. So werden Methoden zur *Ortung* der Spieler erläutert, außerdem Möglichkeiten, Informationen mit Hilfe von *Tags* ortsbasiert zu hinterlegen. Es gibt einen Überblick über die Technik in aktuellen *Smartphones* und die Entwicklung bei *Head-Up Displays*. Das Kapitel schließt mit der Beschreibung spezieller *Authoring Tools* für Mobile Mixed-Reality Games.

In Kapitel 3 werden anhand existierender Spiele *Design-Elemente* erläutert, die für Mobile Mixed-Reality Games charakteristisch sind. Dies sind zunächst besondere Arten der *Interaktion* mit dem Spieler. Weiterhin die Bedeutung von *Zeit* und *Ort* sowie *User-Generated Content* als Basis für den Spielinhalt.

Zuletzt wird in Kapitel 4 ein Fazit der Arbeit gezogen. Dazu werden zunächst kurz die gewonnenen *Erkenntnisse* *rekapituliert*. Dann wird ein *Ausblick* auf die Entwicklung von Mobile Mixed-Reality Games und mögliche Ansätze für weitere Forschung gegeben.

³ <https://scholar.google.com>

⁴ Association for Computing Machinery, <http://acm.org>

⁵ <http://link.springer.com>

⁶ <http://engadget.com>

⁷ <http://gizmodo.com>

⁸ <http://kickstarter.com>

2 Technologien

In diesem Kapitel werden Technologien vorgestellt, die die Realisierung von Mobile Mixed-Reality Games unterstützen. Dabei wird zunächst in Abschnitt 2.1 auf Hardware eingegangen, die in solchen Spielen eingesetzt werden kann. Danach werden in Abschnitt 2.2 Software-Projekte vorgestellt, die – beispielsweise als Authoring-Tool – die Realisierung der Spiele vereinfachen.

2.1 Hardware

Immer neue Hardware eröffnet immer neue Möglichkeiten für Mobile Mixed-Reality Games. Ob das Abschalten der Selective Availability von GPS oder die heutige Allgegenwart von Smartphones – alles hat Entwickler zu neuen Mobile Mixed-Reality Games inspiriert und die Entwicklung weiterer Spiele vereinfacht. Ältere Mobile Mixed-Reality Games, beispielsweise *Live-Action Role-Playing Games* (LARPs), bei denen sich Spieler treffen, um für ein Wochenende eine Fantasy-Welt Realität werden zu lassen, kommen gänzlich ohne digitale Technik aus. Doch heutige Spiele wie *Zombies, Run!* (3.2) verwenden alle ihnen zur Verfügung stehenden Sensoren, um eine möglichst nahtlos mit der Realität verschmolzene Spielwelt zu erschaffen.

Im Folgenden werden zunächst, in 2.1.1, Techniken zur *Outdoor- und Indoor-Ortung* vorgestellt, dann Tags als weiteres Interaktionselement betrachtet (2.1.2), weiterhin, in 2.1.3, die Bedeutung und Möglichkeiten von Smartphones erläutert, und zuletzt Head-Mounted Displays als ganz neue Art der Interaktion mit Mobile Mixed-Reality Games dargelegt (2.1.4).

2.1.1 Ortung

Um den Aufenthaltsort des Spielers in ein Spiel zu integrieren, ist es Grundvoraussetzung, diesen zu kennen. Es werden verschiedene Technologien eingesetzt, die auf verschiedene Problematiken reagieren und so eine Ortung des Nutzers in unterschiedlichem Terrain möglich machen.

Keine dieser Technologien ist für jeden Anwendungsbereich geeignet, und je nach der Ausrichtung des Spiels wird sich der Autor für die eine oder andere Technologie – oder auch einen Mix aus mehreren Technologien – entscheiden.

GPS

Das erste Satellitenortungssystem, *Transit*, wurde bereits 1958 vom amerikanischen Militär in Betrieb genommen. 1985 wurde es durch den heutigen de-facto Standard *GPS* ersetzt [Log92] – alternative, ähnlich funktionierende Technologien sind das russische *GLONASS* und das europäische *GALILEO*. Doch erst mit dem Abschalten der *Selective Availability* im Jahr 2000 hielt GPS Einzug in die Welt der Unterhaltung. Mit einem Mal war eine zuverlässige Ortung auf wenige Meter genau möglich [WW05]. Die zunehmende Verbreitung von GPS-Empfängern in Alltagsgeräten wie beispielsweise Smartphones führt mittlerweile dazu, dass der Ort des Nutzers ein zentrales Element verschiedenster Anwendungen wird.

GPS basiert in der Hauptsache darauf, aus der Konstellation mehrerer GPS-Satelliten durch Triangulation genaue Rückschlüsse über die eigene Position ziehen zu können.

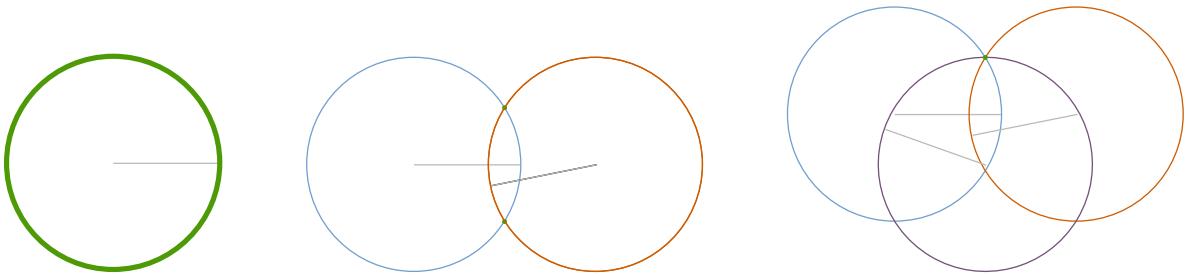


Abbildung 2.1: Schematische Darstellung der Triangulation bei GPS (mit zweidimensionalen Kugeln)

Jeder GPS-Satellit sendet in festen Zeitintervallen seine *Ephemeriden* – die Position im Orbit – und den *Sendezeitpunkt*. Empfängt man nun eine solche Nachricht, kann man die aktuelle Position des Satelliten – die sich ja während dem Sendevorgang verändert (mit etwa 3.9 km/s), und damit die Entfernung zu ihm, anhand eines *Almanachs* von Bahndaten der Satelliten, berechnen. Um die eigene Position zu bestimmen ist es jedoch nötig, die Nachrichten von wenigstens vier GPS-Satelliten nahezu gleichzeitig zu empfangen. Aktuell gibt es 24 GPS-Satelliten, sodass immer wenigstens acht Satelliten “sichtbar”, ihr Signal also empfangbar, sind.

Kennt man nun die Entfernung x des Empfängers zu einem Satellit, bedeutet das, dass sich der Empfänger irgendwo auf einer Kugel um den Satellit mit dem Radius x befindet. Bei mehreren (n) Satelliten erhält man entsprechend n Kugeln mit den Radien x_1 bis x_n . Um die Position des Empfängers zu bestimmen schneidet man nun alle n Kugeln miteinander. Die genaue Position ist der Schnittpunkt dieser Kugeln. Abbildung 2.1 zeigt dieses Vorgehen (zur besseren Übersicht mit zweidimensionalen Kugeln).

Die Welt ist dreidimensional, es sollten also drei Kugeln reichen, um eine Position eindeutig zu bestimmen. Dazu wäre es jedoch nötig, eine Uhr zur Verfügung zu haben, die *exakt mit der Uhr der Satelliten synchronisiert* ist. Das ist nicht zu bewerkstelligen, man erhält also eine weitere, vierte Variable, die *Zeitgenauigkeit*. Eine vierte Kugel liefert weitere Daten, sodass man statt drei Gleichungen mit vier Unbekannten nun vier Gleichungen mit vier Unbekannten hat. Die kann man dann beispielsweise mit dem Gauß-Verfahren [Fis05] lösen.

Bis zum Jahr 2000 wurde das vom US-amerikanischen Militär betriebene GPS für den zivilen Einsatz künstlich ungenau gemacht – man nannte das *Selective Availability*. So sollte verhindert werden, dass feindliche Mächte die Technologie im Kampf einsetzen können. Durch diese Ungenauigkeit konnte ein GPS-Signal auf etwa 100 Meter genau geortet werden, heute sind es zwischen 10 und 15 Metern.

Um die Ortung noch genauer zu machen, wird sogenanntes *Assisted GPS* (A-GPS) [DR01] verwendet. Das beschreibt verschiedene Technologien, mit Hilfe von weiteren Daten und Kommunikationswegen eine schnellere und genauere Ortung möglich zu machen. Häufige Methoden sind:

Mobile Station Assisted GPS hier sendet der GPS-Empfänger (meist ein Smartphone) die von den verschiedenen Satelliten empfangenen Daten über das Mobilfunknetz an einen Server. Der berechnet daraus die aktuelle Position und sendet sie wieder an den GPS-Empfänger zurück. Damit erreicht man sowohl eine deutliche Geschwindigkeitssteigerung, weil benötigte Daten wie ein aktueller Almanach auf dem Server ständig vorgehalten werden können, als auch eine große Genauigkeitssteigerung, da der Server weitere Hilfsdaten, beispielsweise aus dem GSM-Netz, mit in die Berechnung einfließen lassen kann.

Mobile Station Based GPS im Unterschied zum MSA GPS werden Berechnungen hier nicht auf einem Server, sondern auf dem GPS-Empfänger selbst ausgeführt. Hilfsdaten werden dafür wiederum über das GSM-Netz vom Mobilfunkanbieter an den Empfänger gesendet. Die Steigerungen sind ähnlich, aber da die Berechnungen auf den Empfänger ausgeführt werden, ist dieser Ansatz etwas langsamer.

Mit A-GPS kann die *Time To First Fix* (TTFF), also die Zeit vom Beginn der Positionsbestimmung bis zur korrekt berechneten Position, von 30-60 Sekunden ohne A-GPS auf weniger als 10 Sekunden reduziert werden [Bes08].

Mit der mittlerweile auch für zivile Anwendungen verfügbaren hohen Genauigkeit und der durch A-GPS erreichbaren kurzen Antwortzeiten (TTFFs) scheint GPS geradezu perfekt geeignet für Mobile Mixed-Reality Games. Auch die steigende Verbreitung von GPS-Empfängern in Smartphones, Navigationssystemen und sogar MP3-Playern erleichtert den Einsatz bei der Entwicklung von Mobile Mixed-Reality Games stark.

Ein Problem besteht jedoch weiterhin: während GPS im Gelände wunderbar funktioniert, steigen die TTFFs in geschlossenen Räumen stark an, die Genauigkeit nimmt ab oder die Ortung schlägt komplett fehl.

WLAN

Den Schwächen von GPS bei der Indoor-Ortung begegnet meist durch Orten von *WLAN-Signalen*. So erreicht man auch in Gebäuden eine schnelle und genaue Ortung – aktuelle Empfänger verstecken die Verwendung von (A-)GPS und WLAN-Ortung, sodass immer der für die aktuelle Situation passendste Dienst verwendet wird.

Mittlerweile sind, gerade in Ballungsräumen, WLAN-Hotspots stark verbreitet. Allein das Informatik-Gebäude der TU Darmstadt¹ verfügt über 54 WLAN-Antennen [dTd13]. Es passiert also äußerst häufig, dass ein Empfänger gleichzeitig Signale von mehr als einem WLAN-Hotspot empfängt.

Kennt man zudem die genauen Standorte der WLAN-Hotspots, dann kann aus den verschiedenen Signalen seine eigene Position triangulieren. Dies funktioniert ähnlich dem schon für GPS beschriebenen System: für jeden WLAN-Hotspot in Reichweite befindet sich der Empfänger auf einer Kugel mit einem von der Empfangsstärke abhängenden Radius – je besser der Empfang, desto kleiner der Radius. Für diese Kugeln kann man wieder den Schnittpunkt bestimmen, beispielsweise mit dem Gauß-Verfahren. Die eigene Position ist dann genau dieser Schnittpunkt.

Ortung über den Standort von WLAN-Hotspots funktioniert im Unterschied zu GPS in Gebäuden und Ballungsräumen mit vielen verfügbaren Hotspots erstaunlich genau. So kann man in Gebäuden eine Ortungsgenauigkeit von unter einem Meter Abweichung erreichen.

2.1.2 Tags

Neben der genauen Ortung des Spielers über (A-)GPS oder WLAN kann man auch durch die Verwendung von Tags ortsbezogene Daten speichern und weitergeben. Tags werden an bestimmten Orten platziert und können von bestimmten *Lesegeräten* (sogenannten “Readern”) gelesen werden. Dies kann als Alternative dazu verwendet werden, dem Spieler auf Basis seiner Position bestimmte Informationen anzuzeigen – er liest an der gewünschten Position einen Tag ein und erhält so ebenfalls die gewünschte Information. Das ist besonders nützlich, wenn der Spieler kein Ortungsgerät mit sich führt.

Die Verwendung von Tags kommt eigentlich aus der Logistik und Transportplanung. Hier werden sie unter anderem dazu verwendet, detaillierte Informationen über den Inhalt von Paketen und Paletten zu speichern. Doch auch im Bereich von Mobile Mixed-Reality Games werden Tags bereits eingesetzt – zum Beispiel im Serious Game “World Park”², in dem QR Codes im New York Central Park verteilt wurden, um diesen in ein interaktives Trivial Pursuit-Spiel zu verwandeln.

¹ S2|02 Robert-Piloty-Gebäude, Hochschulstraße 10, 64289 Darmstadt

² <http://www.theworldpark.com/campaign>

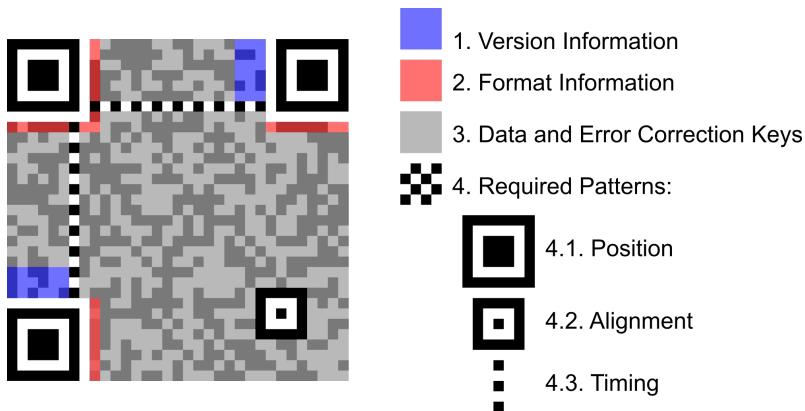


Abbildung 2.2: QR-Code mit benötigten und freien Elementen⁴

Im Folgenden werden zwei vielversprechende Tag-Systeme beschrieben. Zunächst die insbesondere von Smartphones gut lesbaren *QR-Codes*, dann die komplexeren *RFID-Tags*, die in verschiedenen Formen wesentlich mehr Funktionen bieten, aber meist ein eigenes Lesegerät benötigen.

QR-Code

QR-Codes (“Quick Response Codes”) wurden 1994 von Denso Wave, einem Zulieferer des Autoherstellers Toyota, entwickelt, um Baugruppen und Komponenten in der Logistik zu markieren [Ley07]. Die Spezifikation des Verfahrens ist frei verfügbar. Die einfache Lesbarkeit führte dazu, dass heute nahezu für jedes Smartphone Anwendungen existieren, die QR-Codes lesen und die darin enthaltenen Informationen verfügbar machen können.

Ein QR-Code ist eine quadratische *zweidimensionale Punktmatrix* mit 21×21 bis 177×177 Elementen. Drei der vier Ecken des Quadrats sind dabei eindeutig markiert, damit Lesegeräte aus jedem Winkel die Informationen in der richtigen Reihenfolge einlesen können. Dank einer Reed-Solomon-Codierung [WB99] können je nach Fehlerkorrektur-Level zu 7 bis 30 Prozent zerstörte Nachrichten rekonstruiert werden. Maximal – 177×177 Elemente, niedrigstes Fehlerkorrekturlevel – können in einem QR-Code 2953 Byte Information gespeichert werden. Abbildung 2.2 zeigt einen QR-Code mit vom Standard benötigten Elementen und frei vom Anwender nutzbaren Feldern.

Information kann in QR-Codes auf verschiedene Art kodiert werden [GPJ07], [Eby13]:

Numeric Mode (0001) Lediglich aus den *Ziffern 0-9* bestehende Information wird in 10-bit *Chunks* kodiert, die je drei Ziffern enthalten (z.B. “999” → 11 1110 0111³).

Alphanumeric Mode (0010) Kommen außerdem noch *Großbuchstaben und Satzzeichen* hinzu, ergeben sich stattdessen 11-bit Chunks mit je zwei Zeichen (z.B. “HI” → 011 0000 1111).

Byte Mode (0100) Ist die Information komplett in ISO-8859-1 kodierbar, werden einzelne Zeichen als 8-bit Chunks mit ISO-8895-1 kodiert (z.B. “#!” → 0010 0011 0010 0001).

Kanji Mode (1000) Weiterhin wird komplett aus *Kanji* bestehende Information – QR-Codes kommen ursprünglich aus Japan – in einzelnen 13-bit Chunks mit JIS X 0208 [MPC93] kodiert.

Um die komplette Information in einem QR-Code zu bestimmen werden nun folgende Informationen aneinander gereiht:

1. die Kennzahl der Kodierung
2. die Anzahl Zeichen der Information (vor der Kodierung, “hallo” hat z.B. Zeichenlänge 5)

³ Leerzeichen wurden nur für bessere Lesbarkeit eingefügt

⁴ Bild von [Wikimedia Commons](#), veröffentlicht unter [CC-BY-SA](#) von Richard Wheeler, abgerufen am 29.06.2013

-
3. die kodierte Information
 4. 0000 als Ende der Information

Der so entstehende Bitstring wird in 8-Bit Chunks aufgeteilt und gegebenenfalls mit 0 aufgefüllt. Bis zur Kapazität der verwendeten QR-Code Version werden dann abwechselnd die Codes 1110 1100 und 0001 0001 angehängt. Dann werden für den kompletten Bitstring mit einer Reed-Solomon-Kodierung Fehlerkorrektur-Informationen erstellt.

Den vollständigen QR-Code erhält man nun, indem man zunächst die drei Positionierungs-Ecken, das Ausrichtungs-Pattern, sowie Timing-, und Versionsinformationen für den fertigen QR-Code anlegt. Der eben erstellte Bitstring und die Fehlerkorrektor-Informationen wird dann, beginnend in der unteren rechten Ecke in vertikalen Schlangenlinien mit je zwei Einheiten Breite eingefügt. Zuletzt werden nach unterschiedlichen Kriterien acht verschiedene "Masks", die Umkehrung der Elementfarbe der Datenelemente von dunkel nach hell und von hell nach dunkel, ausprobiert, und das für Lesegeräte beste Muster ausgewählt und in den Format-Informationen eingetragen.

Für diesen Vorgang zum Erstellen von QR-Codes gibt es Libraries für alle gängigen Programmiersprachen⁵, außerdem kostenlose Web-Anwendungen, die diese Konvertierung übernehmen.

QR-Codes lassen sich insbesondere mit nahezu allen Smartphones, aber auch mit speziellen Lesegeräten – die sich hauptsächlich durch eine höhere Lesegeschwindigkeit abheben – auslesen. Daher eignen sie sich perfekt, um kleinere Informationen ortsbezogen abzulegen und verfügbar zu machen.

RFID

RFID ("Radio Frequency Identification") wird ebenfalls in der Logistik- und Transportindustrie weitläufig eingesetzt, um Gegenstände zu identifizieren und zu lokalisieren. Aufgrund unterschiedlicher Bauformen sind jedoch weit größere Anwendungen möglich als mit einem Bar- oder QR-Code. So können *aktive RFID-Tags* über Sensoren Daten sammeln, verarbeiten und kommunizieren, und so beispielsweise direkt auf einer Palette speichern, was mit ihr geschehen ist und geschehen soll. Für Mobile Mixed-Reality Games wurden RFID-Tags bisher nicht eingesetzt, da bisher immer ein eigenständiges Lesegerät ("Reader") erforderlich war und die Vorteile nicht überwogen haben. Mit der zunehmenden Aussattung von Smartphones mit NFC-Chips ("Near Field Communication") können diese aber auch bestimmte RFID-Tags auslesen, sodass diese Technik auch für Serious Games interessant wird [RBC⁺⁰⁶].

Grundsätzlich bestehen RFID-Tags aus einer Antenne zur Kommunikation mit dem Lesegerät und einem Mikrochip [She05] – außerdem einem Trägermaterial oder Gehäuse, auf das Antenne und Mikrochip angebracht sind. Man unterscheidet zwischen *passiven*, *semi-passiven* und *aktiven* RFID-Tags danach, ob zusätzlich noch eine Energieversorgung an den Tag angebracht ist, und wie diese genutzt wird:

passiver RFID-Tag hier ist keine eigene Energieversorgung angebracht. Der Tag wird ausschließlich beim Lese- oder Schreibvorgang mit Induktionsstrom durch das Lesegerät mit Energie versorgt. Solche Tags sind günstig – aktuell wenige Cent pro Tag, mit zunehmend druckbaren Tags noch weniger – und können in großer Stückzahl produziert werden.

semi-passiver RFID-Tag solche Tags haben eine eigene Energieversorgung. Die wird allerdings ausschließlich zum Betreiben von Sensoren verwendet, nicht für die Datenübertragung – insbesondere können solche Tags daher keine Datenübertragung initiieren. Für die Übertragung wird wie bei passiven Tags Induktionsstrom verwendet.

⁵ zum Beispiel:
Javascript: <https://github.com/soldair/node-qrcode>,
Java: <https://code.google.com/p/zxing/>,
.NET <http://platform.twit88.com/projects/mt-barcode>

aktiver RFID-Tag ein solcher Tag besitzt eine eigene Stromversorgung und setzt sie sowohl für den Betrieb von Sensoren als auch für die Kommunikation mit anderen Tags oder Readern ein.

Für Mobile Mixed-Reality Games scheinen *passive RFID-Tags* am besten geeignet. Sie können – je nach Bauart – bis zu 32 Kilobyte Information speichern, also mehr als das zehnfache eines QR-Code [GM04]. Diese Kapazität reicht nicht nur für Textinformationen sondern bereits für kleinere, entsprechend komprimierte Bilder. Man könnte also selbst solche Informationen ortsabhängig verfügbar machen, ohne eine zusätzliche Internetverbindung oder lokal gecachte Daten vorauszusetzen.

Viel interessanter als die Kapazität eines Tags ist aber die Möglichkeit, die auf ihm gespeicherten Daten *zu verändern*. Das ist keine Eigenschaft aller RFID-Tags, gerade bei sehr günstigen passiven Tags wird darauf verzichtet, aber RFID-Tags können ohne weiteres mit einem beschreibbaren EEPROM- oder FRAM-Speicher ausgestattet werden. Damit können Mobile Mixed-Reality Games entwickelt werden, bei denen Spieler die Information an bestimmten Orten direkt verändern können. So könnte zum Beispiel eine Liste der letzten Besucher hinterlegt werden, unterschiedliche „Fraktionen“ könnten Orte für sich beanspruchen indem sie dies in den dort angebrachten RFID-Tags speichern, oder Spieler könnten Nachrichten an bestimmten Orten für den nächsten Leser hinterlegen.

Verschiedene RFID-Tags verwenden unterschiedliche Frequenzen zur Kommunikation mit einem Reader. Je nach Anwendung wählt man RFID-Tags im passenden Frequenzband aus. Standardisierte Frequenzbänder sind:

Low Frequency mit 9-135 kHz und einer Lesereichweite von weniger als 0.5 Metern. RFID-Tags in diesem Frequenzband werden beispielsweise verwendet, um Tiere zu identifizieren.

High Frequency mit 6.78, 13.56, 27.125 und 40.680 MHz. Die Lesereichweite beträgt hier etwa 0.5 Meter. Dieses Frequenzband wird daher häufig für digitale Schließsysteme verwendet.

Ultra High Frequency mit 433.920, 869, und 915 MHz sowie 2.45 GHz. Hier beträgt die Lesereichweite zwischen 3 und 6 Metern. Daher wird dieses Frequenzband in der Logistik zum Tagging von Paletten und Werkstücken verwendet.

Super High Frequency mit 5.8 und 24.125 GHz. Mit diesem Frequenzband werden Lesereichweiten bis zu 10 Metern erreicht, weshalb es unter anderem zur Identifikation von Fahrzeugen verwendet wird.

Besonders interessant für Mobile Mixed-Reality Games sind hier *High Frequency RFID-Tags* auf der Frequenz 13.56 MHz. Auf derselben Frequenz arbeiten NFC-Chips, die bereits in einigen aktuellen Smartphones integriert sind. Mit entsprechenden Anwendungen⁶ kann man aber nicht nur NFC-, sondern auch RFID-Tags zugreifen. Man spart sich den zusätzlichen Reader. Und damit sind die Vorteile der RFID-Technik, nämlich *große Datenmengen, Lese- und Schreibzugriff* und eine *nicht benötigte GPS- und Internetverbindung*, für Entwickler und Spieler von Mobile Mixed-Reality Games verfügbar.

2.1.3 Smartphones

Seit der Vorstellung des *iPhone*, durch die Firma Apple, im Jahr 2007, haben *Smartphones* einen rasanten Siegeszug in den Alltag der heutigen Gesellschaften angetreten. In Deutschland nutzten 2012 38 Prozent der Männer und 26 Prozent der Frauen ein Smartphone [CSG12]. Bei der Entwicklung von Mobile Mixed-Reality Games kann man also auf von Smartphones bereit gestellte Technik zurückgreifen und erreicht direkt einen großen potentiellen Nutzerkreis.

Smartphones unterscheiden sich von herkömmlichen Handys vor allem in der zur Verfügung stehenden *Rechenleistung* – hier können sie mit wenigen Jahre alten Desktop-Rechnern konkurrieren – und den

⁶ z.B. NFC TagInfo für Android, <https://play.google.com/store/apps/details?id=at.mroland.android.apps.nfctaginfo>, abgerufen am 29.06.2013

eingebauten *Sensoren*, wie verschiedenen Lage- und Beschleunigungssensoren, einem GPS-Empfänger und wenigstens einer Kamera.

Für Mobile Mixed-Reality Games sind weit verbreitete Geräte mit einer so großen Ausstattung hinsichtlich Sensorik und Rechenleistung äußerst wertvoll. Zu Beginn der zivilen Nutzung von GPS im Spiel *Geocaching* (3.1) mussten Spieler noch dedizierte GPS-Empfänger besitzen und mit sich führen, jegliche Information wurde direkt in den Caches abgelegt – zum Beispiel in Form von Zugriffslisten oder Stempeln. Mit der heutigen Smartphone-Technik muss der Spieler lediglich ein Smartphone mit sich führen, was er in den allermeisten Fällen ohnehin tut, und sich dort das Spiel installieren. Dann sind komplexe Anwendungen wie beispielsweise *Ingress* (3.1) möglich, die eine vollständige erweiterte Realität auf dem Smartphone simulieren.

Häufig vorhandene Sensoren sind⁷:

Ortung über A-GPS, digitalen Kompass, GSM-Netz und WLAN (2.1.1)

Kamera in Spitzengeräten heute mit Auflösungen von über acht Megapixeln, häufig eine weitere, niedrig auflösendere Kamera auf der Vorderseite für Videogespräche

Gyroskop misst die *Rotationsgeschwindigkeit* des Smartphones in den verschiedenen Achsen. Dieser Vorgang basiert im Kern auf der *Coriolis-Kraft* und dem *Sagnac-Effekt*. Eine gute Einführung in die dahinter stehende Technik liefert [SD07]

Accelerometer misst die Beschleunigung des Smartphones, indem die auf eine Testmasse wirkende *Trägheitskraft* mit Hilfe von Federn bestimmt wird. In modernen Smartphones sind diese Sensoren als *mikro-elektro-mechanische Systeme* (“MEMS”) integriert, die bei einer äußerst geringen Größe und Energieaufnahme genaue Ergebnisse liefern [DW10].

Näherungsschalter (Proximity Sensor) erkennt die Annäherung des Smartphones an andere Objekte [MOL+08]. So schaltet sich beim iPhone der kapazitive Touchscreen ab, wenn das Gerät zum Telefonieren an das Ohr gehalten wird.

Diese Sensoren finden in der Spieleentwicklung breite Anwendung. Für Mobile Mixed-Reality Games scheinen aber *Ortung* und *Kamera* die entscheidenden Aspekte: sie ermöglichen einerseits eine *genaue Positionsbestimmung* und damit das Verbinden von Spielementen mit realen Umgebungen, andererseits ein einfaches *Aufnehmen und Teilen von Informationen*. Gyroskop, Accelerometer und Näherungsschalter eignen sich gut, um Inhalt auf dem Smartphone-Bildschirm zu steuern. Diese Eigenschaften sind freilich auch bei Mobile Mixed-Reality Games gefragt, die besonderen Spielkonzepte bauen aber eher auf Ortung und Kamera auf.

2.1.4 Head-Mounted Displays

Neben Smartphones und GPS-Empfängern, die aktuell als Hardware-Basis für Mobile Mixed-Reality Games verwendet werden, zeichnet sich ein Trend zu *Head-Mounted Displays* ab, die eine ganz neue Verschmelzung von Realität und Fiktion ermöglichen.

Neben reinen *Virtual-Reality* Brillen wie der Cinemizer⁸ oder dem Oculus Rift⁹, bei denen Realität komplett durch Virtuelle Realität ersetzt wird – was jedoch auch völlig neue Interaktionsmöglichkeiten erlaubt, werden zunehmend auch *Augmented-Reality* Brillen wie Glass¹⁰ und Cast AR¹¹ marktreif gemacht, die gerade einen Übergang von Realität in Virtuelle Realität ermöglichen.

⁷ am Beispiel des Apple iPhone 5 (<http://www.apple.com/iphone/specs.html>, abgerufen am 30.06.2013)

⁸ <http://cinemizer.zeiss.com>

⁹ <http://oculusvr.com>

¹⁰ <http://google.com/glass>

¹¹ <http://technicalillusions.com>



Abbildung 2.3: Oculus Rift – in endgültigem Aussehen, als Entwickler-Version in Aktion

Oculus Rift

Die über Crowdfunding finanzierte Virtual-Reality Brille *Oculus Rift* ist seit Dezember 2012 als Entwicklerversion verfügbar [VR12]. Sie unterscheidet sich von Videobrillen von Carl Zeiss, Sony oder Vuzix hauptsächlich im Grad der von ihr erzeugten Immersion: während andere Videobrillen dem Nutzer einen großen Bildschirm direkt vor dessen Augen suggerieren, weckt das Oculus Rift das Gefühl, durch eine Skibrille direkt in die Virtuelle Realität “hineinsehen” zu können (siehe Abbildung 2.3).

Die hohe Immersion des Oculus Rift wird vor allem durch eine niedrige Verzögerung, zwischen Bewegung und Änderung der virtuellen Realität, erreicht. Dafür wird eine Bewegungsüberwachung, bestehend aus einem Gyroskop, einem Accelerometer und einem Magnetometer, das Änderungen zum Erdmagnetfeld erkennt, eingesetzt. Sie wurde speziell entwickelt, um mit einer Wiederholrate von 1000 Hz eine Verzögerung von lediglich 2ms zu ermöglichen [VR13b] – erst Verzögerungen ab 7 ms sind spürbar [Abr12].

Oculus Rift unterstützt derzeit die Game-Engines *Unity 3D* [VR13a], *Unreal Engine 4* [VR13c] und *Source* [Val13]. Über HDMI und DVI kann das Gerät an beliebige Desktop-Rechner, aber auch an Spielekonsolen, Handhelds und Smartphones angeschlossen werden. So hat der Spiele-Entwickler größtmögliche Freiheit bei der Wahl der besten Technologie.

Für Mobile Mixed-Reality Games eignet sich Oculus Rift jedoch nur bedingt. Für ein Spiel die ganze Zeit eine solche Brille zu tragen scheint wenig praktikabel – man müsste ständig Angst haben, sich zu stoßen, sich zu verlaufen oder ins Leere zu treten.

Eine mögliche Anwendung – und dafür eignen es sich dann wieder sehr gut – lässt sich in Verbindung mit anderen Geräten finden. So könnte ein Spieler von Ort zu Ort ohne Brille, dafür mit GPS-Gerät gelangen, um dann am Zielort die Brille aufzusetzen und einen für dort vorgesehenen Inhalt zu genießen. Dadurch könnte er beispielsweise mehr über die Geschichte eines Ortes oder wichtige Ereignisse, die dort stattgefunden haben, erfahren.

Glass

Google entwickelt derzeit die Augmented-Reality Brille *Glass*. Seit Februar 2013 steht sie als Entwicklerversion zum Kauf bereit. Dabei ist auf einem Brillengestell über dem rechten Auge ein transparentes Display sowie Kamera, Tonübertragung (über KnochenSchall) und Gestensteuerung angebracht [Inc13]. Abbildung 2.4 zeigt ein Exemplar.

Informationen werden bei Glass nicht in die Umgebung integriert – zum Beispiel indem bei einer Navigationsanwendung direkt die Fahrbahn hervorgehoben wird, die der Nutzer befahren soll, sondern in der oberen rechten Ecke des Blickfelds des Nutzers angezeigt (siehe wiederum 2.4). Somit kann Glass



Abbildung 2.4: Google Glass – Hardware, Navigationsinterface

als Übergang von einem “ständig in den Blick gehaltenen Smartphone” zu einer Augmented-Reality Anwendung gesehen werden. Derzeit bedeutet es eher, Informationen ständig im Blick haben zu können, ständig Fotos schießen und veröffentlichen zu können, ständig Leute anzurufen, dank Bluetooth sogar ohne das Handy aus der Tasche holen zu müssen.

Dennoch kann Glass das “next big thing” für Mobile Mixed-Reality Games sein. Auch Geocaching (3.1) kommt ohne Augmented-Reality aus und hat mittlerweile über 4 Millionen aktive Spieler [Geo13]. Und Glass bietet die Möglichkeit, Daten immer direkt im Blick zu haben, sowie mit Kamera, Gyrosensor und Accelerometer diese Daten mit dem aktuellen Blickfeld zu verknüpfen. Über die Verbindung zu einem Smartphone können Daten beliebig nachgeladen werden – dies erleichtert Updates und User-Generated Content erheblich.

Glass-Anwendungen, sogenannte *Glassware*, sind web-basierte Anwendungen, die in Java¹², PHP¹³ und Python¹⁴ entwickelt werden können. Google bietet dazu die *Google Mirror API* an, mit der Berechnungen nicht auf dem Glass selbst ausgeführt werden müssen, sondern in “die Cloud” – also ein Google Rechenzentrum – verschoben werden.

2.2 Software: Authoring Tools

Mindestens so wichtig wie die für Mobile Mixed-Reality Games zur Verfügung stehende Hardware (2.1) ist die *Software*, mit der die Spiele entwickelt werden können. Solche Spiele haben besondere Anforderungen an ihr Authoring Tool, da mit *Zeit* und *Ort* neue Aspekte wichtig werden, die in herkömmlichen Authoring Tools nicht adäquat beachtet werden.

Anforderungen an Authoring Tools

[WBOJ12] beschreibt zwei *Use Cases* für solche Authoring Tools. Einerseits könnten Nutzer ein *konkretes Spiel* vor Augen haben und dafür Inhalt generieren wollen. Andererseits könnten Nutzer *interessante Inhalte* festhalten wollen, ohne zu wissen, in welchen Spiel sie später eingesetzt werden. Daraus werden folgende Anforderungen abgeleitet:

Nutzer sollen Strukturen für Spielobjekte anlegen können, die verschiedene Medienformate wie Text, Bild, Video oder Positionen beinhalten. Beispielsweise verwendet das Mobile Mixed-Reality Games Tidy City (3.3) Spielobjekte jeweils bestehend aus Titel, Schwierigkeit, Kategorie, Startposition, textuellem Hinweis, Hinweisbild, Zielposition, Belohnungstext und Belohnungsbild.

¹² <http://java.com>

¹³ <http://php.net>

¹⁴ <http://python.org>

Nutzer sollen Spielobjekte aus dieser Struktur erstellen können. In Tidy City wäre ein solches Spielobjekt ein einzelnes Rätsel, beispielsweise mit dem Titel “Love is a bridge, and Niklas and Verena are no exception.”

Nutzer sollen Notizen und Medien in-situ, also direkt an dem Ort, an dem das Spielobjekt sich später befinden soll, erstellen können. Bei Tidy City könnte sich der Nutzer beispielsweise direkt an der Kölner Hohenzollernbrücke eine Notiz machen, dass am Anfang der Brücke ein Vorhängeschloss mit der Beschriftung “Niklas & Verena” hängt.

Nutzer sollen Spielobjekte am Desktop-Rechner erstellen und ändern können. Dies bietet sich an für textlastige Informationen – die lassen sich mit einer Tastatur wesentlich leichter schreiben als mit einer Touchscreen-Tastatur. Wiederum in Tidy City könnte der Nutzer seine Notizen nun zum Spielobjekt mit dem Titel “Love is a bridge, and Niklas and Verena are no exception.” machen.

Nutzer sollen Spielobjekte in-situ erstellen und ändern können. Bei Positionen und Medien bietet sich das an – die Position kann automatisch vom GPS-Empfänger abgefragt werden, Ton, Bilder und Videos von der Smartphone-Kamera können direkt integriert werden. In Tidy City könnte der Nutzer ein Spielobjekt mit der Position der Hohenzollernbrücke und einem Bild von dem Vorhängeschloss mit “Niklas & Verena” anlegen.

Nutzer sollen Spielobjekte für andere Spiele exportieren können. Dies adressiert den zweiten Use Case – der Nutzer hat ein Spielobjekt mit verschiedenen Informationen angelegt und möchte es nun für ein anderes Spiel, als ursprünglich geplant, nutzen.

Nutzer sollen Daten zwischen unabhängigen Tools synchronisieren können. Hier stand wieder der zweite Use Case Pate – wird das Authoring Tool nur zum Sammeln von Daten verwendet kann es gut sein, dass das eigentliche Spiel mit einem anderen Tool entwickelt wird.

Um dieser Anforderungen Herr zu werden, werden in dem in [WBOJ12] beschriebenen *TOTEM Framework* (“Theories and Tools for Distributed Authoring of Mobile Mixed-Reality Games”) drei Datenstrukturen eingeführt: *Pebbles* (“Geröll”) als unsortierte Datenansammlungen, *Shapes* (“Umrisse”) als Strukturen für Spielobjekte, und *Marbles* (“Murmeln”) als in Form bestimmter Shapes gebrachte Pebbles – einzelne Spielobjekte. Durch die Unterscheidung von Shapes und Marbles ist es möglich, dass Spieldesigner lediglich Shapes erstellen und jeder Nutzer eigene Marbles für das Spiel hinzufügt.

Die Authoring-Tools des TOTEM-Frameworks

Zum einen verwendet das TOTEM-Framework den *TOTEM.Designer*, eine Web-Anwendung für Desktop-Rechner. Die erlaubt anspruchsvolles Erzeugen und Editieren von Shapes, Marbles und Pebbles. Der Nutzer kann hier Shapes anlegen, die zwangsläufig nur einen Titel und eine Farbe, zur einfachen visuellen Unterscheidung, haben müssen. Existieren im Workspace des Nutzers bereits Shapes, kann er aus ihnen Marbles anlegen. Dafür erhält er basierend auf dem gewählten Shape ein Formular mit allen notwendigen Daten. Weiterhin kann der Nutzer Pebbles anlegen, die eine Beschreibung, eine Position und beliebige Medien enthalten. Marbles können als JSON¹⁵ exportiert werden, wodurch auch der zugehörige Shape exportiert wird. Diese Exporte können leicht in andere Programme importiert werden. Abbildung 2.5 zeigt die Oberfläche von *TOTEM.Designer*.

Zum anderen stellt es *TOTEM.Scout*, eine mobile App für das Smartphone-Betriebssystem Android, bereit. Der Funktionsumfang ist angepasst auf mobile Nutzung. So kann der Nutzer keine neuen Shapes anlegen, sondern nur bereits existierende Shapes einer Datenbank laden. Marbles können aus Shapes angelegt, und existierende Marbles bearbeitet werden. Dabei können Positionen und Medien direkt aufgezeichnet und in das Marble integriert werden. Die Erstellung von Pebbles ist besonders einfach – lediglich eine Beschreibung wird benötigt, die Position wird automatisch angezeigt, beliebig viele Medien können leicht hinzugefügt werden. Über XML-RPC werden Daten zwischen *TOTEM.Scout* und *TOTEM.Designer* synchronisiert. Abbildung 2.6 zeigt die Oberfläche von *TOTEM.Scout*.

¹⁵ <http://www.json.org/>

The screenshot shows the TOTEM.Designer interface. At the top, there are tabs for GAMES, PEBBLES, SHAPES, MARBLES, GUI, BEHAVIOR, and EXPORT. The SHAPES tab is active.

- EDIT SHAPE Panel:**
 - General:** Name: TidyCity, Color: #414944
 - Properties:** Buttons for New Text and New Integer. Fields for Name, Category, Difficulty, and Textual Hint.
 - Locations:** Buttons for New GPS Circle and New NFC. Fields for Initial Position and Destination Position.
 - Media:** Buttons for New Image, New Audio, and New Video. Fields for Photo Hint and Reward Photo.
- EDIT MARBLE Panel:**
 - General:** Name: Kennedy bridge, Category: Building
 - Properties:** Textual Hint: Named after a former president...
 - Difficulty:** Integer: 2
 - Locations:**
 - Media:** Photo Hint: An image of a bridge at night.
- PEBBLES Panel:**
 - Kennedy bridge:** Text: The Kennedy bridge is located in the city center of Bonn. It was recently renewed.
 - Media:** Images, audio, and video files associated with the Pebble.
 - Main station:** Text

Abbildung 2.5: TOTEM.Designer – Shape für Tidy City, außerdem ein passendes Marble mit Pebble

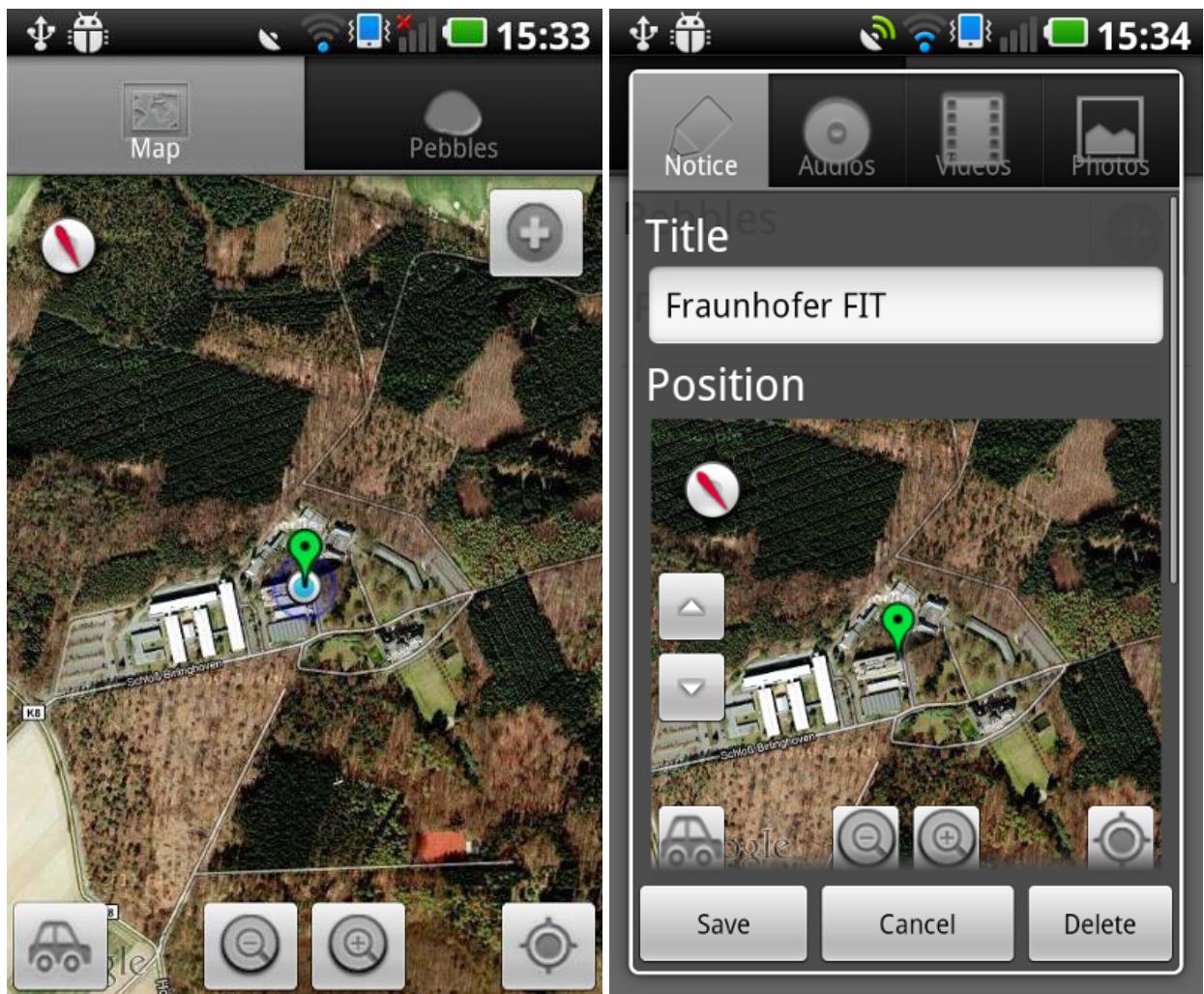


Abbildung 2.6: TOTEM.Scout – Pebble auf einer Kartenansicht, neu erstelltes Pebbles

3 Design-Elemente

In diesem Kapitel werden markante *Design-Elemente* verschiedener Mobile Mixed-Reality Games betrachtet und jeweils anhand eines oder mehrerer Anwendungen erläutert. Die dabei zum Einsatz kommenden Anwendungen sind sowohl im Bereich der Serious Games, als auch im Bereich der Unterhaltungsspiele zu finden. Bei der Beschreibung wird immer wieder auf Themen des vorherigen Kapitels verwiesen, die nun in verschiedenen Anwendungen eingesetzt werden, und zuvor mehr “für sich” betrachtet wurden.

Als wichtigstes Design-Element wird zunächst auf die *Interaktion* mit dem Spieler eingegangen (3.1), was sowohl das Anzeigen von Information als auch das Aufnehmen von Nutzereingaben beinhaltet. Beispiele für gelungene Interaktion sind *Geocaching* und *Ingress*.

Darauf folgend wird in 3.2 das Design-Element *Zeit und Ort* vorgestellt. Hier wird das Spiel *Zombies, Run!* als Beispiel verwendet.

Zuletzt wird das Design-Element *User-Generated Content* beleuchtet (3.3), das einerseits zur stärkeren Bindung der Spieler an das Spiel verwendet wird, andererseits aber auch *Multiplayer-Spiele* erleichtert. Das Serious Game *TidyCity* ist hier ein gutes Beispiel.

3.1 Interaktion

Die Interaktion des Spielers mit dem Spiel ist ein entscheidendes Kriterium für anhaltenden Spielspaß, erreichten Grad an Immersion und Akzeptanz des Spiels. Im Bereich der Unterhaltungsspiele hat die *Nintendo Wii* mit der dort eingesetzten Bewegungssteuerung eine völlig neue Ära von Spielekonsolen eingeleitet – statt an “Hardcore-Gamer” richten diese sich nun mehr an Familien und Freundeskreise, die gemeinsame Unterhaltung suchen. Und auch die Allgegenwart von *Smartphones* (2.1.3) und deren neue Interaktionsmöglichkeiten haben den Markt der Unterhaltungsspiele in eine gänzlich neue Richtung verändert – statt großen “AAA-Titeln” mit multi-millionen-Dollar Budget werden nun vermehrt “casual games” entwickelt, die nicht auf gute Story und moderne Grafik setzen, sondern auf schnellen Spielspaß.

Mobile Mixed-Reality Games profitieren besonders von neuen Interaktionsmöglichkeiten, viele solcher Spiele wären ohne sie nicht möglich. So entstand das Spiel *Geocaching* (3.1) als “Antwort” auf das Abschalten der Selective Availability von GPS (2.1.1). Spiele wie *Zombies, Run!* (3.2) oder *Ingress* (3.1) machen so intensiven Gebrauch von den Fähigkeiten moderner Smartphones (2.1.3), dass sie ohne diese undenkbar wären.

Man kann den Begriff “Mobile Mixed-Reality Games” etwas weiter auslegen, nämlich dahingehend, dass die reale Welt mit einer *fiktiven* Welt verknüpft wird. Dann existieren mit verschiedenen *Live Action Role-Playing Games* (LARPs) [THBK06] sogar einige Mobile Mixed-Reality Games, die komplett auf digitale Unterstützung, moderne Sensoren und neue Interaktionsformen verzichten. Sie basieren hingegen auf der Fantasie, dem Einfühlungsvermögen und dem schauspielerischen Talent der Spieler, und der dadurch generierten Atmosphäre einer “Parallelwelt”. So treffen sich LARP-Spielgruppen, um für einen Tag, ein Wochenende oder sogar länger eine Fantasy-Welt aufzuleben zu lassen. In dieser Arbeit werden solche Spiele jedoch nicht weiter vertieft, da sie für moderne, digital- und sensorgestützte Mobile Mixed-Reality Games wenig hilfreiche Anregungen liefern.



Abbildung 3.1: Geocaching – ein Cache (Größe “small”), ein Trackable (Travel Bug)

Im Folgenden werden die Spiel- und Interaktionskonzepte von zwei Mobile Mixed-Reality Games erläutert. Zunächst das minimalistische *Geocaching*, dann das aufwändige *Ingress*.

Minimalistisch: Geocaching

Das Spiel Geocaching [Geo13] wurde 2000 entwickelt, um den Wegfall der Selective Availability von GPS (2.1.1) zu feiern. Mittlerweile hat das Spiel über vier Millionen aktive Spieler und mehr als zwei Millionen *Caches* in 184 Ländern [Sch13]. Dabei kommt das Spiel mit äußerst wenig Regeln, Hilfsmitteln und Technik aus – wohl auch ein Grund für die große Popularität.

Geocaching ist im Kern eine *GPS-unterstützte Schnitzeljagd*. Spieler verstecken Kisten, sogenannte *Caches*, die jeweils ein “Logbuch”, einen Stift und eventuell tauschbare Gegenstände beinhalten. Die Positionen der Verstecke werden veröffentlicht. Andere Spieler begeben sich dann mit einem GPS-Empfänger auf die Suche nach den versteckten Caches. Hat man einen Cache gefunden, trägt man sich in das darin enthaltene Logbuch ein und tauscht eventuell einen der tauschbaren Gegenstände aus. Abbildung 3.1 zeigt einen typischen Cache.

Dieses einfache Konzept wurde durch andere Spielideen immer wieder erweitert. Dabei rückt immer mehr eine Gemeinschaft von Spielern in den Mittelpunkt. Bekannte Erweiterungen sind unter anderem:

Gemeinschafts-Websites wie [geocaching.com](#). Hier werden nicht nur Cache-Positionen veröffentlicht, sondern auch gesammelte Caches markiert. So hat jeder Spieler eine Statistik über versteckte und gesammelte, es gibt Achievement-Systeme und einen aktiven Wettbewerb um “first to finds” (FTFs), also die Tatsache, als erster Spieler einen Cache entdeckt zu haben.

Trackables Das sind eindeutig identifizierbare Gegenstände, die in Caches abgelegt werden. Findet man ein Cache mit einem Trackable, kann man diesen mitnehmen – und etwas anderes dafür da lassen – und im nächsten gefundenen Cache wieder ablegen. Das trägt man auf speziellen Websites ein. Da die Trackables eindeutig identifiziert werden können (z.B. über eine ID) kann man so den Weg eines Trackable durch verschiedene Caches nachverfolgen. Abbildung 3.1 zeigt einen Travel-Bug, eine bestimmte Art Trackables, die an die “Hundemarken” des amerikanischen Militärs angelehnt ist.

Besondere Cache-Arten Neben dem oben beschriebenen “Traditional Cache” – ein Versteck, eine Position, ein Cache – entwickelten sich weitere Arten von Caches. Besonders verbreitet sind dabei *Multi-Caches*, die aus einer Reihe von Caches (sogenannte *Stages*) bestehen, und in jeder Stage Informationen über die Position des endgültigen Caches und der nächsten Stage enthalten. Weiterhin verbreitet sind *Mystery-Caches*. Dabei wird nicht die Position des Caches veröffentlicht, sondern ein Rätsel, dessen Lösung entweder die Position des Caches oder die erste Stage eines Multi-Caches ist. Solche Rätsel können letztlich beliebig gestaltet werden, häufig sind sie jedoch mathematischer oder kryptographischer Natur.

Geodashing Dies geht mehr in Richtung Mannschaftssport als in Richtung Freizeitunterhaltung. Gespielt wird in zwei Mannschaften mit je bis zu fünf Spielern über einen Zeitraum von einem Monat. Beide Mannschaften versuchen, möglichst viele GPS-Positionen (Dashpoints) zu besuchen, belegt wird dies durch eine detaillierte Beschreibung des Ortes oder ein Foto. Die Mannschaft, die nach Ablauf des Spiels die meisten Dashpoints besucht hat, gewinnt das Spiel¹.

Aufwändig: Ingress

Einen ganz anderen Weg geht das Mobile Mixed-Reality Games *Ingress* [Hod12]. Das Spiel wird von *Niantic Labs*, einer Tochterfirma von Google, entwickelt. Seit November 2012 befindet es sich in einer geschlossenen Betaphase. Bisher existiert das Spiel als App für Android-Smartphones, sowie als inoffizieller Port für iOS-Geräte. Abbildung 3.2 zeigt Ansichten der Nutzeroberfläche.

Während Geocaching mit möglichst minimalistischer Ausrüstung – einem GPS-Empfänger – auskommt, nutzt Ingress alle Möglichkeiten moderner Smartphones, um ein Gefühl einer parallelen Realität zu erzeugen.

Das Setting des Spiels ist futuristisch: Die Erde wurde von “Exotischer Materie” (XM) durchzogen, die die Ausserirdischen *Shaper* hinterlassen haben. Diese Materie tritt an verschiedenen Stellen der Erde (sogenannte *Portale*) auf. Zwei Fraktionen, *Enlightened* (“Die Erleuchteten”, grün) und *Resistance* (“Der Widerstand”, blau), kämpfen um die Vorherrschaft an den Portalen. Die Enlightened wollen XM zu ihrem Vorteil nutzen, die Resistance jegliches XM zerstören.

Zu Beginn des Spiels entscheidet sich der Spieler für eine der beiden Fraktionen, Enlightened oder Resistance. Danach bekommt er Zugriff auf die Karte der virtuellen Welt, in der Portale und die sie aktuell dominierende Fraktion eingetragen sind. Portale sind meist Sehenswürdigkeiten oder interessanten Orten der realen Welt zugeordnet. Die Karte überlagert dabei die Karte der realen Welt und erweitert sie eben um solche Portale. Befindet sich ein Spieler physisch in der Nähe eines Portals, kann er dieses “hacken”. So gewinnt er das Portal für seine Fraktion, erhält virtuelle Erfahrung und Gegenstände.

Direkte Interaktion mit dem Spiel geschieht über die Smartphone-Anwendung. Hier wird die Karte angezeigt, hier kann der Spieler Portale zum hacken auswählen, hier chattet er mit Spielern seiner Fraktion. Dabei prüft die Anwendung permanent die aktuelle Position des Spielers – über GPS, WLAN oder digitalen Kompass (2.1.1).

Neben dem offensichtlichen Effekt, dass Ingress-Spieler sich für das Spiel physisch bewegen müssen – man kann Ingress daher durchaus ein Serious Game nennen – entsteht für den Entwickler Google ein weiterer, lohnenswerter Effekt: er erhält eine aktuelle, genaue Datenbank von Sehenswürdigkeiten. Neue Portale werden, mit Standort und Fotos, von der Community vorgeschlagen und nach einer Prüfung durch Niantic Labs freigeschaltet. Dieser Vorschlagsvorgang ist denkbar einfach gestaltet: es genügt eine Mail an super-ops@google.com mit einem Foto der Sehenswürdigkeit, in dessen Metainformationen die genaue Position abgelegt ist. Eine solche Datenbank wäre algorithmisch nur äußerst schwer zu erstellen,

¹ <http://geodashing.gpsgames.org/>, abgerufen am 01.07.2013

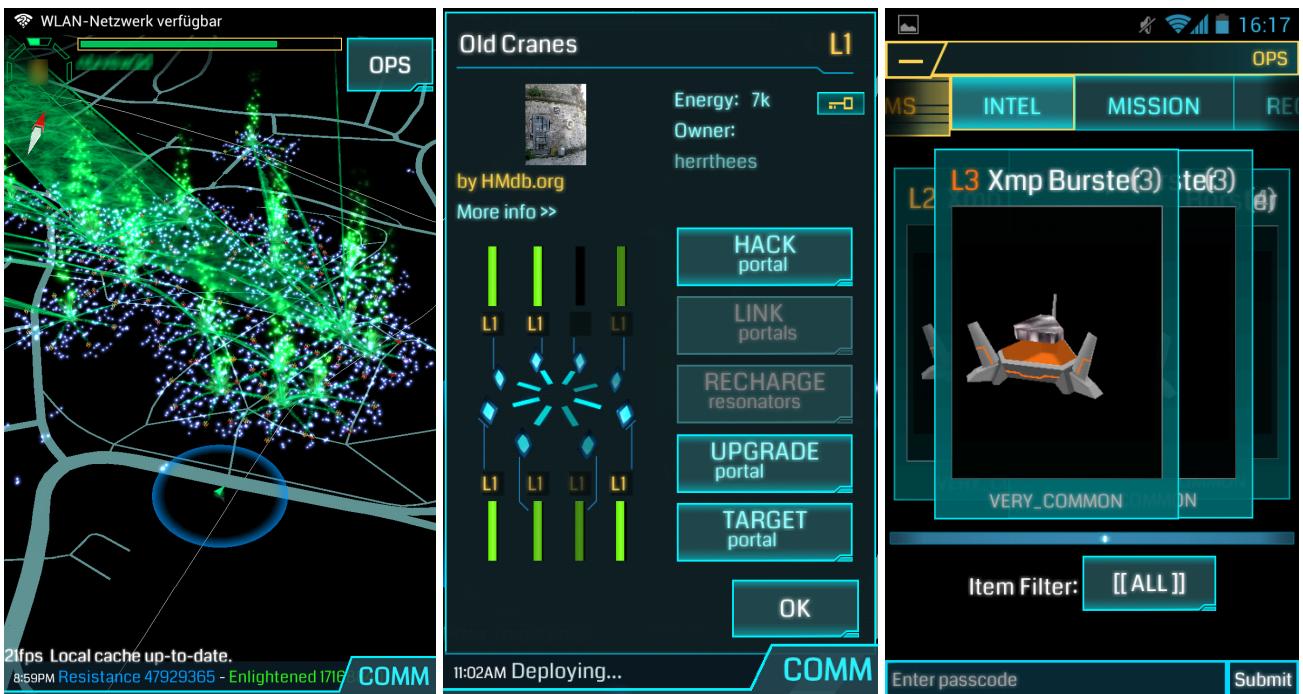


Abbildung 3.2: Ingress – Karte, Portal, Inventar

der Algorithmus müsste beispielsweise die Bilder aus Google Street View² auswerten und selbstständig Sehenswürdigkeiten als solche erkennen. Infofern ist Ingress ein gelungenes Beispiel für *user-generated content*.

3.2 Zeit und Ort

Ein von klassischen Spielen meist nicht verwendetes Design-Element ist die Abhängigkeit von *Zeit und Ort* des Spielers. Ortsabhängigkeit ist eine Grundvoraussetzung für Mobile Mixed-Reality Games – ansonsten wären sie zwar Mixed-Reality Games, aber nicht Mobile. Zeitabhängigkeit ist kein zwingendes Element, kann aber den Spielfluss deutlich interessanter gestalten [Mon05].

So könnten Informationen nur zu bestimmten Zeitpunkten verfügbar sein, oder Aktionen nur in gewissen Zeiträumen, beispielsweise nachts, durchführbar sein. Das kann den Grad der Immersion der parallelen Realität stark erhöhen.

Zombies, Run!

Als beispielhaftes Spielkonzept wird hier das Mobile Mixed-Reality Games *Zombies, Run!*³ vorgestellt. Das ist ein Serious Game, durch das Spieler zum Laufen angeregt werden. Im Februar 2012 war es die meistverkaufte Fitness-Anwendung im iOS App Store. Abbildung 3.3 zeigt verschiedene Ansichten der Anwendung.

Der Spieler wird dabei in die Welt einer Zombie-Apokalypse [Kha11] versetzt. Als “Runner 5” muss er das “Abel Township” mit Nachschub versorgen. Außerhalb der Mauern Abels lauern Horden von Zombies, die die Runner verfolgen und den Nachschub abschneiden.

² <http://www.google.com/streetview>

³ <https://www.zombiesrungame.com/>

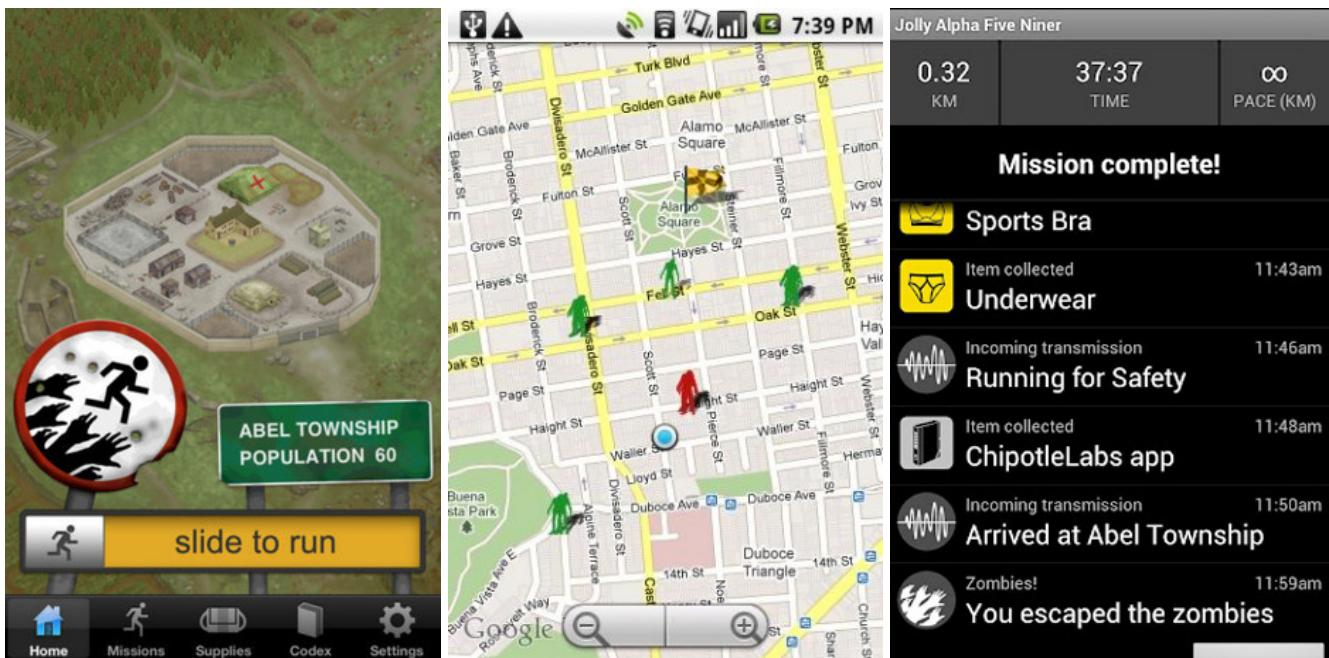


Abbildung 3.3: Zombies, Run! – virtuelles Dorf des Spielers, Karte mit Position und Zombies, gesammelte Gegenstände nach abgeschlossener Mission

Durch Missionen, Läufen in der realen Welt, die über GPS oder das Accelerometer des Smartphone aufgezeichnet werden, sammelt der Spieler Nachschub. Damit wird nach und nach die Stadt ausgebaut. Die Storyline endet mit der Rettung des Abel Township durch Entsatz-Truppen der nahe gelegenen Militärbasis “New Canton”.

Zombies, Run! nutzt das Smartphone so gut wie möglich aus, auf dem es ausgeführt wird. Während Läufen zeigt es auf einer Karte an, wo sich gerade Zombies befinden, denen der Spieler besser ausweichen sollte. Weiterhin nutzt es die Tonausgabe des Smartphone, um neben allgemeiner Musik zum Laufen auch Geräusche von sich nähernden Zombies abzuspielen. Das alles dient dazu, den Spieler weiter zum Laufen zu motivieren und das Laufen interessanter zu gestalten.

3.3 User-Generated Content

Ein drittes, sehr bedeutsames Element von Mobile Mixed-Reality Games ist *User-Generated Content*. Das bedeutet, dass der Entwickler zwar die Spielmechanik und eventuell einige Missionen vorgibt, aber der eigentliche Spielinhalt – der Großteil der Missionen, Informationen und Daten – von Spielern der Community zusammengetragen und angelegt wird.

Für den Entwickler hat User-Generated Content einen großen Vorteil. Hat er einmal die Spielmechanik definiert, wird das Spiel ein Selbstläufer. Er ist nur für Updates der Spielmechanik verantwortlich, die Spielinhalte wachsen “von allein”. Es bedeutet natürlich, in besonderem Maß auf die Community und deren Wünsche eingehen muss. Da diese für die Content-Generierung verantwortlich ist, ist sie deutlich mächtiger als bei “klassischen” Titeln. Doch auch für den Spieler hat User-Generated Content einen großen Vorteil. Da Nutzer jederzeit neue Spielinhalte hinzufügen können, verliert das Spiel nicht an Reiz, wenn die offizielle Storyline durchgespielt ist. Außerdem kann er sich selbst als Autor betätigen, und beispielsweise speziell für seinen Ort, Content hinzufügen.

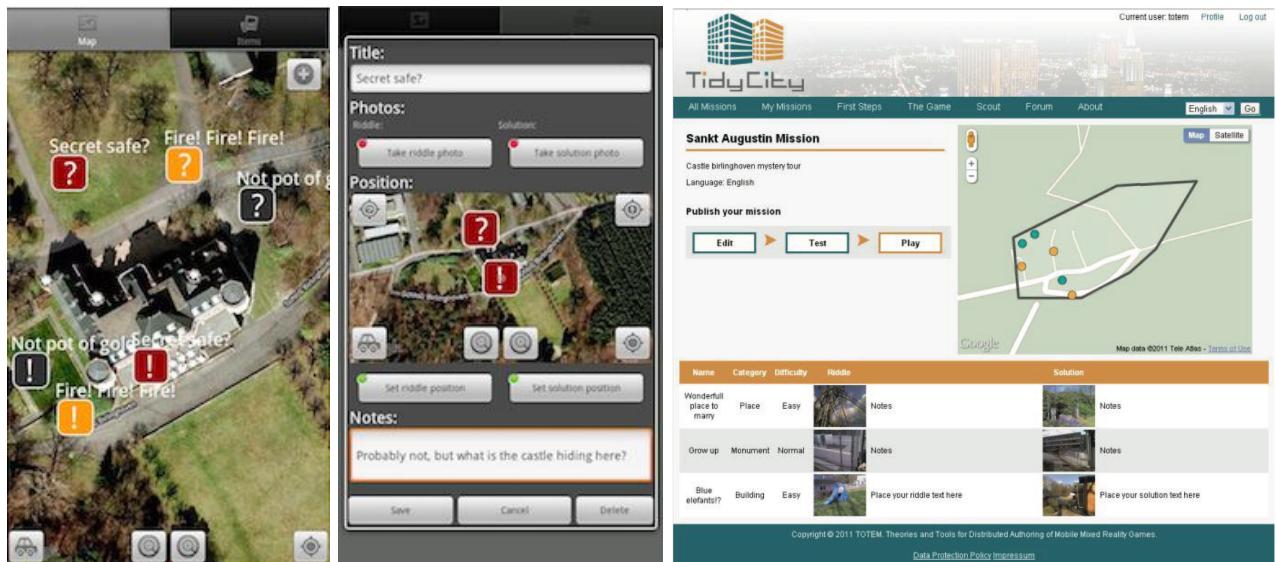


Abbildung 3.4: TidyCity – Karte mit Rätseln, in-situ Generierung von Content, Authoring Web-Anwendung

Tidy City

Das Serious Game *Tidy City* (“aufgeräumte Stadt”) [WBO12], das eng mit dem TOTEM-Projekt (2.2) zusammen arbeitet, setzt genau hier an: Die Spielentwickler geben ein *Shape* für das Spiel vor, Nutzer können selbst *Marbles* für diesen Shape erstellen und so Spielinhalte schaffen. Abbildung 3.4 zeigt Ansichten der Android-Anwendung und die auf TOTEM.Designer basierende Web-Anwendung zur Generierung von Content.

Als Forschungsprojekt ist Tidy City von der Storyline her minimalistisch gehalten und möchte möglichst viel Entscheidungsfreiheit der Community überlassen. Die Spielidee ist, dass in der parallelen Welt Rätsel verteilt sind. Diese Rätsel muss der Spieler “aufsammeln” und “an den Richtigen Ort bringen” – damit “räumt” er “seine Stadt auf”. Der Richtige Ort ist jeweils die Lösung des Rätsels. Ist der Spieler am Zielort, kann er das Rätsel abschließen und eine (virtuelle) Belohnung kassieren.

Tidy City lieferte folgende Erkenntnisse [WBO⁺11]:

- Durch eine *minimalistische Spielmechanik* können Spieler aller Altersgruppen das Spielkonzept schnell verstehen.
- Dank diesem schnellen Verständnis fällt es Spielern leicht, *neue Rätsel zu erstellen*.
- Das *in-situ Authoring* (siehe Abbildung 3.4) motiviert dazu, die *eigene Umgebung kennen zu lernen*. Dadurch macht auch das Erstellen neuer Rätsel Spaß.
- Die Spielidee ist gut geeignet, in *kleinen Gruppen* gespielt zu werden. So fördert es die soziale Interaktion zwischen den Spielern.

Durch das Authoring-Tool TOTEM.Scout (2.2) interagiert Tidy City in einem von anderen Mobile Mixed-Reality Games nicht bekannten Maß. Zwar kann man z.B. auch bei Ingress (3.1) neue Portale vorschlagen, doch Tidy City basiert existentiell darauf, dass Spieler selbst kreativ werden.

4 Fazit

In dieser Arbeit wurden Technologien (2) und Design-Elemente (3) untersucht, die in Mobile Mixed-Reality Games besonders stark vorkommen und sie besonders beeinflussen.

Hier wird nun rekapituliert, was diese Untersuchung ergeben hat (4.1). Darauf basierend wird ein Ausblick gegeben, wohin sich Mobile Mixed-Reality Games entwickeln können und welche Technologien und Design-Elemente dabei eine besondere Rolle spielen werden (4.2).

4.1 Ergebnis

Es wurden verschiedene Möglichkeiten zur Ortung von Spielern und zur Informationsweitergabe beschrieben. Hier ist besonders *GPS* hervorzuheben. Der Aufenthaltsort des Spielers ist ein zentraler Bestandteil von Mobile Mixed-Reality Games. *GPS* bietet einen einfachen, komfortablen und genauen Weg, diesen zu bestimmen.

Unter den vorgestellten Geräten zeigten sich *Smartphones* als derzeit vielversprechendste Größe. Die Möglichkeiten von Head-Up Displays sind zwar weitreichend, die hier erläuterten Kandidaten befinden sich aber noch in der Entwicklung. Es gibt schlicht noch keine Mobile Mixed-Reality Games, die auf Head-Up Displays basieren. Smartphones hingegen haben sich zum hauptsächlichen Interaktionsmedium entwickelt.

Das Spiel *Tidy City* und das damit verbundene *TOTEM-Framework* weisen den Weg für Mobile Mixed-Reality Games, die auf user-generated content basieren. Sie machen es auch unerfahrenen Nutzern sehr einfach, Spielinhalte zu generieren. Aufwändige Spiele wie *Ingress* zeigen, was derzeit im Bereich der Mobile Mixed-Reality Games möglich ist. Wie zukunftsträchtig solche Spiele sind, muss sich allerdings zeigen.

4.2 Ausblick

Die hier vorgestellten Technologien bilden nur die Spitzenspitze einer neuen Generation von Spielen. Mit der zunehmenden Verbreitung von Smartphones, die mit immer mehr Technologie ausgestattet werden, und der Entwicklung von alltagstauglichen Head-Up Displays werden Spiele möglich, die noch vor fünf Jahren unvorstellbar waren.

Gerade der Aspekt der *Augmented Reality* wird stark zunehmen. Zunächst könnte man, wie schon im Spiel *Ingress* zu sehen, eine Karte der Realität mit der einer Parallelwelt überlagern. Ein nächster Schritt wäre dann, eine Kamera, in Smartphone oder Head-Up Display, zu nutzen, um Bilder der Realität mit gerenderten 3D-Modellen der Parallelwelt zu vermischen.

Auch im Bereich des *user-generated content* kann noch weiter erforscht werden. Interessant wäre hier, einen konkreten Zusammenhang zwischen dem Langzeiterfolg von Spielen und deren Unterstützung für user-generated content nachzuweisen. Weiterhin können Wege erforscht werden, die Generierung von Content noch einfacher zu gestalten.

Insgesamt ist der Bereich von Mobile Mixed-Reality Games sehr zukunftsträchtig. Wir können hier in den nächsten Jahren noch einiges erwarten.

Literaturverzeichnis

- [Abr12] Michael Abrash. Latency – the sine qua non of AR and VR. <http://blogs.valvesoftware.com/abrash/latency-the-sine-qua-non-of-ar-and-vr/>, Dezember 2012. Abgerufen am 30.06.2013.
- [Bes08] Siegfried W. Best. Assisted GPS: erhöht Genauigkeit, verkürzt TTFF. *elektronik industrie*, 3:83–85, 2008.
- [CSG12] Bianca Corcoran-Schliemann and Stefan Gerhardt. Fast jeder zweite Jugendliche nutzt Smartphone. http://www.gfk.com/de/documents/pressemitteilungen/2012/20120405_pm_smartphones_dfin.pdf, April 2012. Abgerufen am 29.06.2013.
- [DR01] G.M. Djuknic and R.E. Richton. Geolocation and assisted gps. *Computer*, 34(2):123–125, 2001.
- [dTID13] Hochschulrechenzentrum der TU Darmstadt. WLAN-Hotspots. http://www.hrz.tu-darmstadt.de/netz/vpn_wlan/wlan_2/wlanhotspots, 2013. Abgerufen am 26.06.2013.
- [DW10] S Dixon-Warren. Motion Sensing In The iPhone 4: MEMS Accelerometer. <http://www.memsjournal.com/2010/12/motion-sensing-in-the-iphone-4-mems-accelerometer.html>, December 2010. Abgerufen am 30.06.2013.
- [Eby13] Carolyn Eby. QR Code Tutorial. <http://www.thonky.com/qr-code-tutorial/>, April 2013. Abgerufen am 29.06.2013.
- [Fis05] G. Fischer. *Lineare Algebra*. Vieweg-Studium : Grundkurs Mathematik. Vieweg, 2005.
- [Geo13] Geocaching.com. Media FAQ. http://www.geocaching.com/articles/Brochures/footer/FAQ_Media.pdf, 2013. Abgerufen am 03.06.2013.
- [GM04] Birgit Gampl and Rolf AE Müller. Rfid: Technologie der zukunft? *Integration und Datensicherheit-Anforderungen, Konflikte und Perspektiven. Referate der 25. GIL Jahrestagung*, 25:33–36, September 2004.
- [GPJ07] J.Z. Gao, L. Prakash, and R. Jagatesan. Understanding 2d-barcode technology and applications in m-commerce - design and implementation of a 2d barcode processing solution. In *Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International*, volume 2, pages 49–56, 2007.
- [Hod12] Hal Hodson. Google's ingress game is a gold mine for augmented reality. *New Scientist*, 216(2893):19, 2012.
- [Inc13] Google Inc. Google Glass Tech Spec. <https://support.google.com/glass/answer/3064128>, Mai 2013. Abgerufen am 03.06.2013.
- [Kha11] AS Khan. Preparedness 101: zombie apocalypse. *Public Health Matters. Centers for Disease Control and Prevention*, 2011.
- [Ley07] David Ley. Ubiquitous Computing. *Emerging Technologies for Learning*, 2(6):64–79, 2007.
- [Log92] Tom Logsdon. *The Navstar global positioning system*. Van Nostrand Reinhold New York, 1992.

- [MOL⁺08] Emiliano Miluzzo, James MH Oakley, Hong Lu, Nicholas D Lane, Ronald A Peterson, and Andrew T Campbell. Evaluating the iphone as a mobile platform for people-centric sensing applications. *Proc. UrbanSense'08*, 2008.
- [Mon05] Markus Montola. Exploring the edge of the magic circle: Defining pervasive games. In *Proceedings of DAC*, page 103, 2005.
- [MPC93] Jun Murai, Erik M Poel, and Mark Crispin. Japanese character encoding for internet messages. 1993.
- [O'H08] Kenton O'Hara. Understanding geocaching practices and motivations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1177–1186, New York, NY, USA, 2008. ACM.
- [RBC⁺06] Omer Rashid, Will Bamford, Paul Coulton, Reuben Edwards, and Jurgen Scheible. PAC-LAN: mixed-reality gaming with RFID-enabled mobile phones. *Comput. Entertain.*, 4(4), October 2006.
- [Sch13] Eric Schudiske. Celebrating Two Million Geocaches – List by Country. <http://blog.geocaching.com/2013/02/celebrating-two-million-geocaches-list-by-country/>, Februar 2013. Abgerufen am 01.07.2013.
- [SD07] Robert Bosch GmbH (Stuttgart) and K. H. Dietsche. *Kraftfahrtechnisches Taschenbuch*. Kraftfahrtechnisches Taschenbuch. Vieweg + Teubner Verlag, 2007.
- [She05] Steven Shepard. *RFID: radio frequency identification*. McGraw-Hill New York, 2005.
- [THBK06] Anders Tychsen, Michael Hitchens, Thea Brolund, and Manolya Kavakli. Live Action Role-Playing Games Control, Communication, Storytelling, and MMORPG Similarities. *Games and Culture*, 1(3):252–275, 2006.
- [Val13] Valve. Source SDK 2013 Release. <http://store.steampowered.com/news/10962/>, Juni 2013. Abgerufen am 30.06.2013.
- [VR12] Oculus VR. Oculus Rift: Step Into the Game. <http://www.kickstarter.com/projects/1523379957/oculus-rift-step-into-the-game>, 2012. Abgerufen am 03.06.2013.
- [VR13a] Oculus VR. 4-Month Unity Pro Trial for Oculus Devs. <http://www.oculusvr.com/blog/4-month-unity-pro-trial-for-oculus-devs/>, März 2013. Abgerufen am 30.06.2013.
- [VR13b] Oculus VR. Building a Sensor for Low Latency VR. <http://www.oculusvr.com/blog/building-a-sensor-for-low-latency-vr/>, Januar 2013. Abgerufen am 30.06.2013.
- [VR13c] Oculus VR. Oculus-ready Unreal Development Kit Coming in April. <http://www.oculusvr.com/blog/oculus-ready-unreal-development-kit-coming-in-april/>, März 2013. Abgerufen am 30.06.2013.
- [WB99] S.B. Wicker and V.K. Bhargava. *Reed-Solomon Codes and Their Applications*. Wiley, 1999.
- [WBO⁺11] Richard Wetzel, Lisa Blum, Leif Oppermann, Michael Sträubig, and Feng Feng. Tidy City: A location-based game for city exploration based on user-created content. In Maximilian Eibl, editor, *Mensch & Computer*, pages 487–496. Oldenbourg Verlag, 2011.
- [WBO12] Richard Wetzel, Lisa Blum, and Leif Oppermann. Tidy city: a location-based game supported by in-situ and web-based authoring tools to enable user-created content. In *Proceedings of the International Conference on the Foundations of Digital Games*, FDG '12, pages 238–241, New York, NY, USA, 2012. ACM.

-
- [WBOJ12] Richard Wetzel, Lisa Blum, Leif Oppermann, and Audrius Jurgelionis. Shapes, Marbles and Pebbles: Template-based Content creation for location-Based Games. In *Proceedings of IADIS International Conference Game and Entertainment Technologies*, Juli 2012.
- [WW05] T.H. Witte and A. M. Wilson. Accuracy of WAAS-enabled GPS for the determination of position and speed over ground. *Journal of Biomechanics*, 38(8):1717 – 1722, 2005.
- [ZG01] J.F. Zumberge and G. Gendt. The demise of selective availability and implications for the international GPS service. *Physics and Chemistry of the Earth, Part A: Solid Earth and Geodesy*, 26(6 — 8):637 – 644, 2001.