

2009

MOBILE STATECHART-BASIERTE TOURISTIK-ANWENDUNGEN

Jens Krösche
Fakultät für Informatik

Martin Scheuchenegger
Fakultät für Informatik

Stephan Drab
Fakultät für Informatik

Stephan Selinger
Fakultät für Informatik

Follow this and additional works at: <http://aisel.aisnet.org/wi2009>

Recommended Citation

Krösche, Jens; Scheuchenegger, Martin; Drab, Stephan; and Selinger, Stephan, "MOBILE STATECHART-BASIERTE TOURISTIK-ANWENDUNGEN" (2009). *Wirtschaftsinformatik Proceedings 2009*. 162.
<http://aisel.aisnet.org/wi2009/162>

This material is brought to you by the Wirtschaftsinformatik at AIS Electronic Library (AISeL). It has been accepted for inclusion in Wirtschaftsinformatik Proceedings 2009 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

MOBILE STATECHART-BASIERTE TOURISTIK-ANWENDUNGEN

Jens Krösche, Martin Scheuchenegger, Stephan Drab,
Stephan Selinger¹

Kurzfassung

Das Paper zeigt wie im Rahmen des Projektes zur Entwicklung der Lomotain-Plattform eine generische Spiellogik basierend auf Statecharts für mobile Endgeräte entworfen wurde. Eines der Hauptziele des Projektes lag darin, die Funktionslogik eines mobilen Tourismusführers bzw. eines Spiels, das das Erlebnis eines Park oder Stadtbesuchs steigern soll, von der eigentlichen Anwendung auf dem mobilen Endgerät des Anwenders zu trennen. Die Funktionslogik selbst wird dabei mit Hilfe einer Authoring-Umgebung in Form von Statecharts definiert/verwaltet/angepasst und in einem eigenen Dateiformat an den mobilen Client übertragen. Dieser generiert dann hieraus zur Laufzeit die eigentliche Funktionslogik der Anwendung.

1. Lomotain

Ziel der Lomotain-Plattform (Location based Mobile Gaming and Entertainment) ist die Bereitstellung einer Plattform zur schnellen Umsetzung mobiler, multimedialer und spielbasierter Stadt-, Touren- und Adventure-Guides. Insbesondere ein Einsatz in etablierten Tourismusgebieten und Themenparks, die durch den Einsatz modernster Technologie ihre Wettbewerbsfähigkeit steigern können ist einer der zentralen Anwendungsschwerpunkte. Dabei hebt sich Lomotain vor allem durch seine integrierten Konzepte und Ziele von herkömmlichen Tourismusführern ab.

Ein vorrangiges Ziel war die Abkehr vom klassischen Tourismusführer, der die Aufmerksamkeit des Benutzers zu sehr beanspruchte und ihn dadurch von dem eigentlichen Ziel, die Natur/den Ausflug zu genießen, ablenkte. Da insbesondere Familien in der Regel den Wunsch haben ihre Ferien gemeinsam zu „erleben“ und nicht trocken über einen elektronischen oder analogen Tourismusführer präsentiert zu bekommen. Hier setzt Lomotain an: weg von der reinen Interaktion mit dem Endgerät hin zur direkten Interaktion mit der Umwelt und damit zum Erleben der Umwelt.

Zwar wird bei der Lomotain-Plattform, ebenso wie bei vergleichbaren Systemen [3, 4, 7] ein mobiles Endgerät benötigt, allerdings wird die Anwendung auf dem System nicht über Tastendrücke gesteuert, sondern durch die direkte Wechselwirkung mit den Gegenständen und Orten im Themenpark. Dies geschieht über die Integration von RFID-basierten Markern in dem

¹ FH OÖ; Fakultät für Informatik, Kommunikation und Medien; Studiengang Mobile Computing

Spielareal, die dem System eine Interpretation der Benutzerinteraktion ermöglichen. Wird ein Marker erkannt, reagiert das System mit einer entsprechenden Aktion (eine neue Aufgabe wird präsentiert, ein Video gezeigt oder ein Schatz gefunden) und der Spielablauf wird manipuliert. Basierend auf diesem Ansatz fungiert das Handy des Anwenders somit als Spiegel oder Tor in eine andere, verborgene Welt, die gerade für Kinder spannend gestaltet werden kann und die es zu entdecken gilt. Auch soll der Anwender nicht mehr trocken und rein informativ geleitet werden, vielmehr ist es ein Ziel von Lomotain den ganzen Ablauf spielerisch – quasi als Adventure – zu gestalten. Die entsprechende multimediale und multimodale Aufbereitung ist daher ein weiterer wichtiger Aspekt des Projektes.

Technisch basiert das System auf einer Kombination verschiedener Technologien wie z.B. GPS/Galileo, NFC/RFID, UMTS/GSM, Java2ME und entsprechenden mobilen Endgeräten wie Handy, Smartphone oder PDA, wobei im ersten Prototyp ein handelsübliches Nokia 6131 NFC in Verbindung mit einer Bluetooth GPS-Maus zum Einsatz kam. Dabei war es ein weiteres Ziel, den (von vielen alternativen Ansätzen verfolgten) Ansatz des Geräteverleihs zu durchbrechen und anstelle dessen die mitgeführte Hardware der Besucher als Systemplattform zu verwenden.

Der eigentliche Kern des Systems ist die generische Spiellogik, basierend auf Zustandsautomaten (Statecharts), über die das gesamte Spielgeschehen definiert wird. Diese Statechart-Logik kann dabei mittels einer entsprechenden Authoring-Umgebung erstellt und mit geringem Aufwand an die jeweilige Tourismusregion oder Anwendungssituation angepasst werden. Die Logik selbst wird im Anschluss daran in Dateiform, zusammen mit der Anwendung, auf das mobile Endgerät des Benutzers übertragen. Zur Laufzeit liest die Anwendung die Beschreibung ein und generiert daraus dynamisch die eigentliche Spiellogik. So lassen sich die Spielszenarien einfach durch den Austausch von Beschreibungsdateien ersetzen bzw. ergänzen.

Auf Grund der verwendeten Technologien, insbesondere durch den Einsatz von RFID-Technologie, kommen für die Tourismusregionen selbst nur geringe Um- bzw. Aufbaukosten zu. Es werden keine teuren Aufbauten (Figuren oder ähnliches) mehr benötigt, da die Informationen multimedial aufbereitet über das Handy vermittelt werden. Es ist lediglich notwendig an der korrekten Stelle einen entsprechend markierten RFID-Tag zu positionieren, der dem System den notwendigen Rückschluss über das Spielgeschehen ermöglicht. So lassen sich die vorhandenen Gegebenheiten extrem kostengünstig und vandalismussicher aufwerten. Auch sind dank der verwendeten Marker unterschiedliche Spielszenarien mit der gleichen Park-Hardware möglich, so dass eine ständige Anpassung bspw. des Parks nicht mehr zwangsläufig notwendig ist. Zusätzlich können alternative kontextsensitive Konfigurationen bspw. basierend auf der aktuellen Jahreszeit (Sommer/Winter) oder für verschiedene Altersgruppen (Kinder/Eltern) definiert werden und somit das Spielgeschehen an die Situation anpassen. Ebenso wird durch die Plattform eine Kooperation verschiedener Tourismus-Gebiete gefördert, um Synergieeffekte zu forcieren. Anwender die bereits einmal ein Spiel erlebt haben können über entsprechende Community-Plattformen, Highscorelisten, Statistiken und Gewinnspiele motiviert werden, auch andere angeschlossene Regionen zu erkunden

Nach dem kurzen Abriss der grundlegenden Aspekte der Lomotain-Plattform wird im Folgenden knapp auf den aktuellen state-of-the-art eingegangen, um danach zwei zentrale Konzepte im Detail zu besprechen. Nach der Vorstellung der Statechart-Logik soll vor allem das Administrationswerkzeug detailliert besprochen werden. Den Schluss bildet eine kurze Zusammenfassung, die die erreichten Ergebnisse präsentiert und einen Ausblick in die Zukunft von Lomotain gibt.

2. Forschungsstand

Konzeptionell fällt die Lomotain-Plattform sowohl in die Kategorie der mobilen Tourismusführer also auch in die der Location-Based Games (LBG). In beiden Kategorien wurden in den vergangenen Jahren viele Versuche hauptsächlich von Universitäten und Forschungseinrichtungen unternommen – mehr oder weniger erfolglos – solche Systeme gesellschaftsfähig zu machen und damit einer breiten Masse der Bevölkerung zur Verfügung zu stellen. Im Folgenden werden einige der bekannteren Projekte kurz vorgestellt.

Das LBG „Can You See Me Now“ [1] der Künstlertruppe Blast Theory basiert auf dem aus den Kindheitstagen gut bekannten „Fangen“-Spielprinzip. Dabei verknüpft das Spiel die Realität mit ihrer virtuellen Repräsentation. In einem vorab definierten Bereich eines Stadtgebietes werden die Fänger („runner“) auf die Jagd nach ihrem virtuellen Gegenpart geschickt. Ein Nachfolger von „Can You See Me Now“ ist z.B. „Uncle Roy All Around You“ [2].

Eines der am weitest verbreiteten – weil weltweit gespielten – LBGs ist das Geocaching [5]. Am ehesten lässt sich das Geocaching mit einer Schatzsuche, der Suche nach einem Cache (Schatz), vergleichen, bei der die Position des Schatzes zwar bekannt ist, der Weg dorthin aber erst gefunden/bewerkstelligt werden muss. Verschiedene existierende Kategorien von Geocaches erhöhen zusätzlich den Spielspass. So werden einige Verstecke durch Rätsel verschlüsselt, oder es müssen wie bei einer Schnitzeljagd, verschiedene Teilrätsel gelöst werden. Jedoch ist es das Hauptziel von Geocaching die Natur zu erleben und gleichzeitig Spaß zu haben.

Ein weiteres Beispiel für eine multimediale Schnitzeljagd ist das PaperChase [3] Projekt der Universität Oldenburg. Ziel war es neue Studierende auf der Grundlage einer Schnitzeljagd den Campus der Universität erkunden zu lassen. Dabei wurden diese über einen GPS gestützten PDA von einer wichtigen Location zur nächsten über verschiedene Rätsel geleitet.

Das Projekt GUIDE [4] stellte in der Stadt Lancaster eine der ersten mobilen und personalisierten Stadtführungen zur Verfügung. Basierend auf mobilen Endgeräten und einem WLAN-Netz, welches gleichzeitig zur Ortung und Informationsübermittlung verwendet wurde, konnten Touristen unter Berücksichtigung von vorhandenen Nutzerprofilen gezielt über die touristisch interessanten Plätze der Stadt informiert werden. Dabei spielte die Position des Touristen, welche anhand der verwendeten WLAN-Stationen relativ exakt bekannt war, eine entscheidende Rolle bei der Übermittlung, Präsentation und dem Inhalt der Informationen.

LoL@ [8] war ein mobiler, interaktiver und multimedialer Tourismusführer, der als Übertragungsmedium UMTS und als Endgerät ein Mobiltelefon oder Smartphone verwendete. Zentraler Baustein war eine Kartendarstellung, die zur Orientierung des Anwenders und der Präsentation von POIs und Routen diente. Der Anwender hatte die Möglichkeit, auf seiner Route zu touristisch interessanten Punkten multimediale Zusatzinformationen zu bekommen. Präsentiert wurden diese Daten akustisch, bildlich, als Video oder in einfacher Textform. Zu Navigationszwecken konnte der Anwender entlang vorausberechneter Routen oder von POI zu POI geführt werden.

Bereits dieser Auszug aus dem state-of-the-art zeigt, dass sich die Szene rasant weiterentwickelt. Auffällig ist vor allem, dass Spiele, die den Anwender zwingen selbst aktiv zu werden – mit der Realität zu interagieren – von den Anwendern sehr positiv aufgenommen werden. Daher findet sich dieser Ansatz auch innerhalb der Lomotain-Plattform wieder.

3. Generische Spiellogik

Statecharts sind eine Darstellungsform von endlichen Automaten in der Informatik und wurden in ihrer Urform von David Harel [6] eingeführt. Die Notation erlaubt die präzise Spezifikation von zustandsbasierten Systemen (s. Abbildung 1 – am Beispiel eines Kassettenrekorders), wie es auch die Spieldefinitionen im Rahmen der Lomotain-Plattform sind.

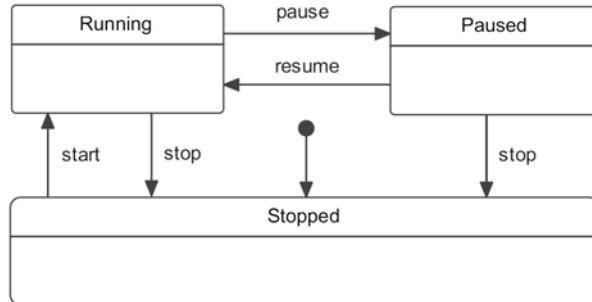


Abbildung 1: Ein vereinfachter Zustandsautomat eines Kassettenrekorders

Die im Rahmen der Lomotain-Plattform entwickelten Statecharts orientieren sich dabei in ihrer Funktionalität stark an den in der UML 2.0 spezifizierten Standards. Auf Grund des eingeschränkten Anwendungsspektrums wurden aber diverse Aspekte aus der UML 2.0 Spezifikation nicht übernommen oder entsprechend angepasst.

Ein Spiel, repräsentiert durch einen Zustandsautomaten (s. Abbildung 2), setzt sich dabei aus verschiedenen Unterzustandsautomaten (Aufgaben-Zustandsautomaten) zusammen. Sind auf der Ebene der Aufgaben-Zustandsautomaten vor allem die RFID-Ereignisse relevant, entscheidet auf der Ebene der Spiel-Zustandsautomaten vor allem die GPS-Position der Spieler darüber, welche Aufgabe gerade zu lösen ist. Hat ein Anwender ein Aufgabenareal erreicht, wechselt das System in einen der Unterzustandsautomaten.

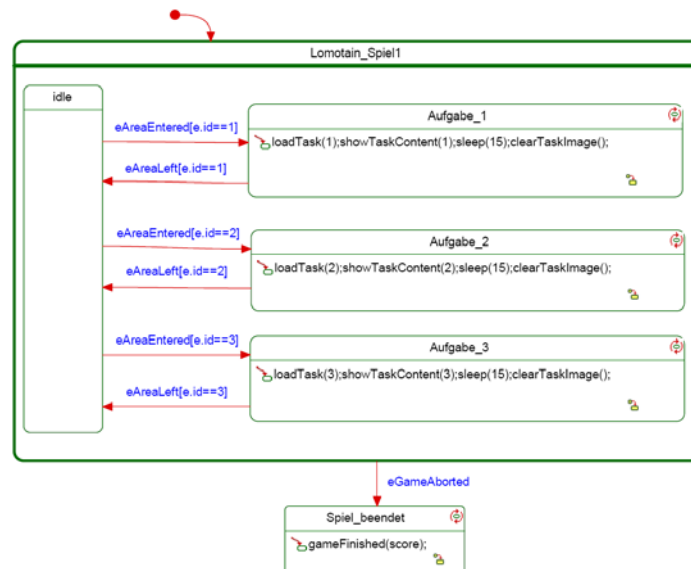


Abbildung 2: Ein Spiel-Zustandsautomat

Eine grundlegende Erkenntnis im Umgang mit Spielen ist, dass sich bestimmte Aufgaben in ihrem grundlegenden Aufbau stark ähneln (Sammeln oder Suchen von Gegenständen, Lösen von Rätseln,

...), so dass sich hieraus allgemeine Aufgabenschablonen ableiten lassen. Im Rahmen der Lomotain-Plattform wurden daher verschiedene Aufgabenschablonen extrahiert und realisiert, die für die Konstruktion eines Gesamtspiels beliebig zusammengesetzt werden können. Zu den zentralen zwei Schablonen (neben dem Labyrinth, dem Memory und weiteren) gehören dabei:

- **Multiple Choice Fragen**
Bei diesem Aufgabentypus bekommt der Spieler eine Frage gestellt, die er beantworten muss. Es gibt drei Antwortmöglichkeiten, die in Form von Bildern und/oder Texten präsentiert werden können. Die Antwortmöglichkeiten müssen aber zuerst gefunden werden, jede Antwort wird durch ein RFID Tag repräsentiert. Erst wenn der Spieler alle drei Antwortmöglichkeiten gefunden hat, kann er eine Antwort auswählen. Die Auswahl erfolgt durch zweifaches Einlesen des gleichen RFID Tags (analog zum Doppelklick auf dem Desktop).
- **Sammeln von Gegenständen**
Beim Sammeln von Gegenständen geht es darum, dass der Spieler Gegenstände in Form von RFID-Tags findet. Jeder gefundene Gegenstand wird am Bildschirm angezeigt. Müssen mehrere gleiche Gegenstände gefunden werden (z.B. fünf Goldmünzen), gibt es zusätzlich einen Zähler, der die Anzahl der gefundenen Gegenstände protokolliert. Ebenso besteht die Möglichkeit eine genaue Reihenfolge vorzuschreiben bzw. eine beliebige Reihenfolge zuzulassen.

Wie bereits angesprochen, werden die Statecharts mit Hilfe eines XML-Dokuments beschrieben und können auf diese Weise auf die Endgeräte übertragen werden. Damit dieses Dokument vom Statechart-Modul des Clients geparkt werden kann, muss es eine ganz bestimmte Struktur einhalten. Diese Struktur wurde im Rahmen der Lomotain-Plattform in Form einer speziellen XML-Schema-Spezifikation definiert. Beim Starten des Statechart-Moduls wird die XML-Beschreibung geparkt und die Statechart-Logik aufgebaut. Für weitergehende Informationen bzgl. der Statechart-Logik und den alternativen Konzepten sei an dieser Stelle auf eine alternative Lomotain-Publikation [7] verwiesen.

4. Authoring-Umgebung

Eine der wichtigsten Komponenten von Lomotain stellt die Authoring-Umgebung dar. Diese dient dazu, die Spiele zu konzipieren, zu entwickeln, anzupassen und zu generieren.

Die Planung und Entwicklung eines Spiels kann je nach Spielszenario und auf Grund der verwendeten Konzepte einen sehr komplexen Vorgang darstellen, der zum Teil nur von entsprechend ausgebildeten Personen durchgeführt werden muss. Da nicht jeder Benutzer die gleiche Ausbildung bzw. Intention hat, werden von der Authoring-Umgebung verschiedene Betriebsmodi (Basic-, Advanced- und Experten-Modus) unterstützt. Mit Hilfe dieser verschiedenen Modi kann das Werkzeug optimal an die Bedürfnisse und Anforderungen der späteren Benutzergruppen angepasst werden.

Die Spielerstellung mit Hilfe des Authoring-Tools gliedert sich in mehrere Arbeitsschritte (in Form von Registerkarten), die je nach dem in welchem Modus das System bedient wird, für den Anwender zur Verfügung stehen oder bereits durch einen Anbieter vorab erledigt wurden.

Der Basic-Modus verfügt über sehr eingeschränkte Manipulationsmöglichkeiten, ist dadurch aber entsprechend einfach zu bedienen. In ihm können nur komplette, vorab definierte Spiele (bestehen

aus mehreren bereits fertig angeordneten Aufgabenschablonen) importiert und nur deren multimedialen Inhalte verändert werden. Des Weiteren kann der Benutzer über die Landkartenansicht die Areale der einzelnen Stationen passend zu den geographischen Gegebenheiten festlegen bzw. ändern. Obwohl im Basic-Modus nur komplette Spiele importiert werden können, bietet das System im Bereich „Multimedialer Inhalt“ die Möglichkeit, die Spielinhalte thematisch anzupassen. Hier können alle Texte, Bilder, Videos, Audio-Dateien und Animationen der einzelnen Stationen bearbeitet und verändert werden. Der Basic-Modus bietet einem Anbieter die Möglichkeit aus einer Palette von vordefinierten Spielen (Piratenschatzinsel, Schneewittchen und die Sieben Zwerge etc.) auszuwählen, hierdurch ist er in der Lage schnell und unkompliziert ein Lomotain-Spiel in seinen Themenpark zu integrieren. Einzig die Installation der RFID-Tags, die Vermessung der Regionen, die Anpassung der Medieninhalte und die Integration in das Lomotain-Webportal sind notwendig.

Der Advanced-Modus erweitert die Authoring-Umgebung und den Basic-Modus um ein weiteres Funktionsmodul, den „Level Designer“. In diesem Modul hat der Benutzer selbst die Möglichkeit die Reihenfolge der einzelnen Aufgaben in einem Spielszenario zu definieren und ist nicht wie im Basic-Modus auf eine fixe Reihenfolge festgelegt. Dazu kann er im Level-Designer bereits vorhandene Schablonen importieren, die er dann beliebig aneinanderreihen und so den Spielablauf völlig frei konstruieren kann. Auch Teile der Inhalte der gelieferten Schablonen lassen sich an die regionsspezifischen Gegebenheiten anpassen, bspw. in dem nur drei Objekte statt fünf gesucht werden sollen. Der Advanced-Modus bietet einem Anbieter also die Möglichkeit sich ein eigenes Spielszenario aufbauend auf bereits definierten Aufgabenschablonen zu generieren, das optimal zu den örtlichen Gegebenheiten und seinen Vorstellungen passt.

Der Experten-Modus erweitert den Advanced-Modus dahingehend, dass auch die tiefste Ebene der Spiellogik, die Statechart-Ebene, bearbeitet werden kann. Dazu wird das Funktionsmodul „Statechart-Designer“ zur Verfügung gestellt, in dem die gesamte Spiellogik auf unterster Ebene entworfen und bearbeitet werden kann. Ebenso ist es möglich Schablonen zu generieren und für andere Benutzer zur Verfügung zu stellen. Da für diese Ebene bereits tiefere Kenntnisse der Spielstruktur und des Aufbaus eines logischen und konsistenten Zustandsbaumes nötig sind, ist diese Ansicht für Nichttechniker wenig geeignet und wird daher in den beiden anderen Modi nicht zur Verfügung gestellt. Zusätzlich ist im Experten-Modus noch eine weitere Ebene der Spiellogik zugreifbar. Die Spiellogik wird, wie bereits erwähnt, durch Statecharts beschrieben. Im Experten-Modus kann die generierte XML-Datei von Experten nochmals bei Bedarf angepasst und verändert werden. Als weitere Funktion ist im Experten-Modus ein Simulator geplant. Mit dessen Hilfe die entworfenen Spiele noch in der Authoring-Umgebung auf deren Realisierbarkeit überprüft werden können. Fehler in der Spiellogik können gegebenenfalls sofort, noch vor Auslieferung des Spiels, erkannt und ausgebessert werden.

Im Folgenden sollen einige der in den verschiedenen Betriebsmodi zur Verfügung stehenden Funktionsmodule und Dialoge im Detail erläutert werden.

4.1 Funktionsmodul Statechart-Designer

Das Funktionsmodul Statechart Designer ist das zentrale Modul für die Konzeptionierung von Aufgaben und Aufgabenschablonen. Wie Abbildung 3 zeigt, werden in ihm die für eine Aufgabe notwendigen Statechart-Diagramme gezeichnet, die wiederum die Funktionslogik der einzelnen Aufgaben/Spiele repräsentieren. Auch ist aus der Abbildung ersichtlich, warum der Statechart Designer nicht in allen Betriebsmodi zur Verfügung steht, denn die Komplexität der Diagramme

(insbesondere der Guards für die Zustandsübergänge) ist schon für relativ einfache Aufgaben sehr hoch, so dass sie für Laien schwer oder gar nicht zu erstellen sind.

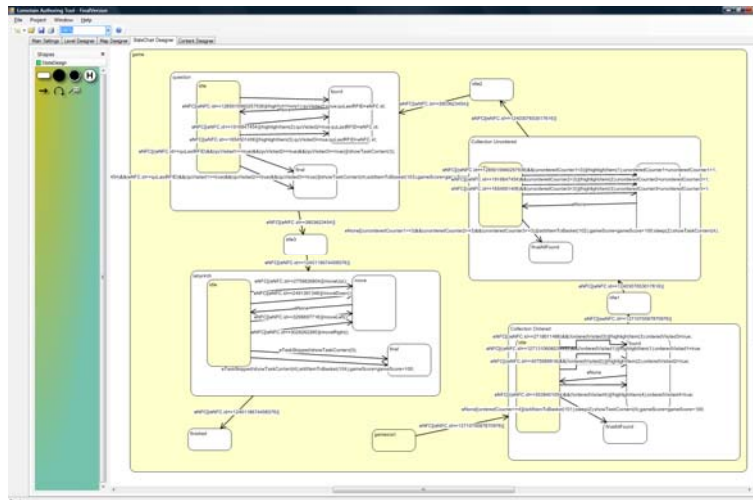


Abbildung 3: Screenshot des Funktionsmoduls „Statechart-Designer“

Über die Funktionsleiste am linken Bildrand in Abbildung 3 kann der Anwender ein State-Objekt auf die Zeichenfläche ziehen und hierdurch einen neuen Zustand definieren. Für diesen neuen Zustand muss der Anwender anschließend einige Eigenschaften, die sich über einen eigenen Dialog definieren lassen, setzen. Für jeden State sind dabei unter anderem seine On-Entry-, On-Exit- und In-State-Aktionen zu definieren, bspw. um bei Zustandseintritt eine Eingangsanimation für das jeweilige Spiel anzuzeigen. Zusätzlich ist es möglich Aspekte wie Spieltyp oder die Fragestellung, ob es sich bei dem Zustand um eine Aufgabe oder einen Initialzustand handelt, zu regeln. Ist ein Zustand definiert können mit Hilfe weiterer State-Objekte, die in den Grafik-Bereich des initialen States gezogen werden, eventuell notwendige Substates definiert werden und das gesamte Diagramm Schritt für Schritt aufgebaut werden.

Um ein funktionierendes Statechart-Diagramm zu erzeugen, werden neben den States und Substates noch die Transitionen, die Zustandsübergänge, die die States miteinander verbinden, benötigt. Analog zu den States selbst, werden auch die Zustandsübergänge mit Hilfe graphischer Verbindungslinien, die zwischen den States gezogen werden, erzeugt. Nach der graphischen Erstellung müssen die eigentlichen Eigenschaften der Transitionen mittels eines speziellen Dialogs festgelegt werden.

Für jeden Zustandsübergang können das auslösende Ereignis (Event), die Bedingung für den Zustandsübergang (Guard) und eine eventuell damit verknüpfte Aktion definiert werden. Ein auslösendes Ereignis könnte im Lomotain-Umfeld beispielsweise das Berühren eines RFID-Tags mit dem Handy oder das Betreten oder Verlassen eines Spielareals sein. Die Definition von Guards ist notwendig, um beispielsweise die Auswahlreihenfolge von RFID-Tag markierten Objekten zu gewährleisten. Als Aktion wäre beispielsweise das Abspielen einer Animation oder die Erhöhung eines internen Objekt-Zählers (n+1 Goldstücke gefunden) denkbar. Der Dialog unterstützt den Anwender bei der Eingabe, in dem er nur tatsächlich mögliche Eingaben akzeptiert bzw. Elemente anbietet.

4.2 Funktionsmodul Content-Designer

Wurde die Logik des Spiels in Form eines Statecharts entworfen, muss als nächstes der multimediale Inhalt des Spiels eingepflegt werden. Diese Funktionalität wird über das Funktionsmodul Content Designer abgedeckt. Im Content Designer-Modul stehen alle entworfenen Aufgaben als Auswahlliste auf der linken Seite des Bildschirms zur Verfügung (s. Abbildung 4).

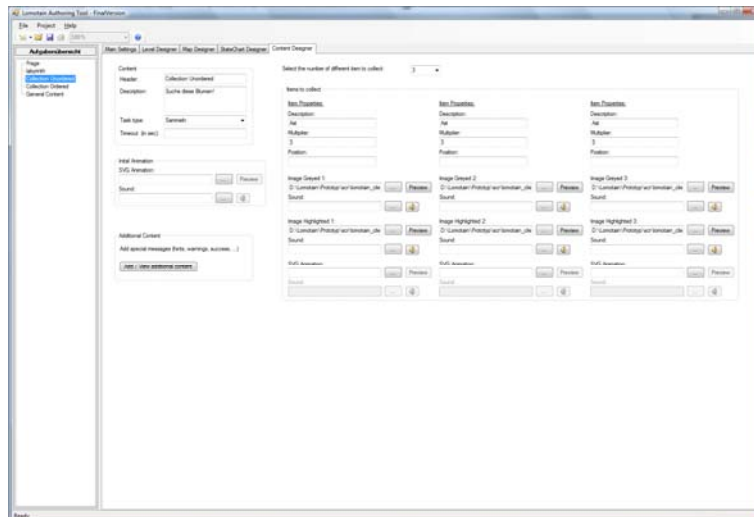


Abbildung 4: Screenshot des Funktionsmoduls „Content-Designer“

Wird eine der Aufgaben durch den Anwender ausgewählt, erscheint im Hauptbereich der Anwendung ein Dialog, über den die multimedialen Inhalte (SVG Animation, Bilder, Audiodateien, Text), passend zum jeweiligen Aufgabentyp, eingepflegt werden können. Die Ansicht des Content-Designers variiert dabei je nach Art der Aufgabe, da jede Aufgabe andere Informationen benötigt.

Da im Content Designer nur „statische“ Inhalte (Bilder, Animationen, Sounds sind bereits erstellt) eingepflegt werden können, wird für den Aufgabentyp „Labyrinth“ ein spezieller Editor benötigt, der Labyrinth Designer. Mit seiner Hilfe ist es möglich Labyrinth flexibel und variabel zu erstellen. Nicht nur die Größe (die Grundfläche) des Labyrinths, sondern auch die darin enthaltenen Wege und Hindernisse sowie den Start- und den oder die Endpunkte können so definiert werden. Ebenso ist es möglich die im Labyrinth verwendeten Texturen und Sprites festzulegen.

Auf Grund der Tatsache, dass gerade die Inhalte der Aufgaben auf den jeweiligen Erlebnispark zugeschnitten sein müssen, steht der Content Designer auch in allen Betriebsmodi zur Verfügung.

4.3 Funktionsmodul Map-Designer

Mit Hilfe des Funktionsmoduls Map-Designer (s. Abbildung 5) ist der Anwender in der Lage die Bereiche/Orte zu definieren, an denen die für ihn vom Lomotion-System zur Verfügung gestellten Spiele und Aufgaben tatsächlich gespielt werden können. Da die Position der späteren Spiele mittels GPS überprüft wird, werden auch die Spielareale im Map Designer in Form von GPS-Koordinaten abgelegt. Der Anwender bekommt eine Übersichtskarte präsentiert, auf der er schrittweise den jeweiligen Aufgaben Areale zuordnet. Analog zu den anderen Funktionsmodulen kann er eine Aufgabe auswählen und dann auf der Karte mit Hilfe der Maus einen polygonalen Bereich für die ausgewählte Aufgabe definieren. Zusätzlich ist es möglich die Positionen der verschiedenen RFID-Tags zu importieren bzw. sich anzeigen zu lassen.

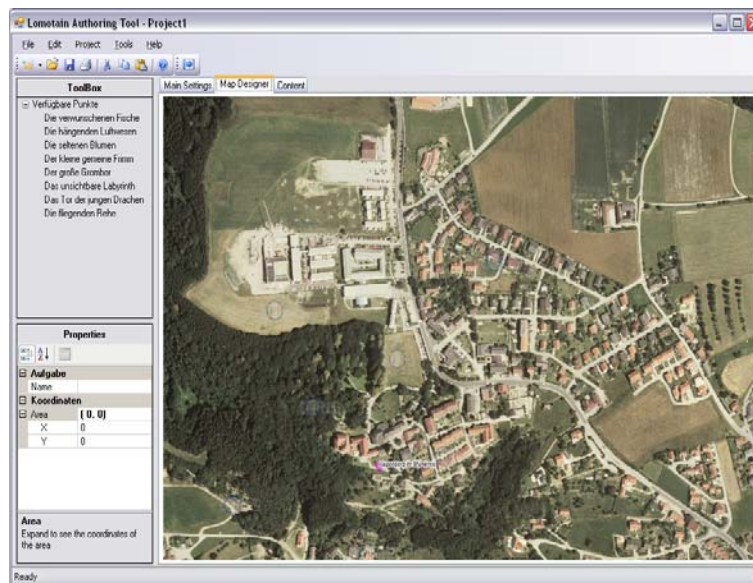


Abbildung 5: Screenshot des Funktionsmoduls „Map-Designer“

4.4 Funktionsmodul Level Designer

Als weiteres Funktionsmodul steht der Level Designer den Anwendern ab dem Advanced-Modus zur Verfügung. Über ihn ist es möglich die Aufgabenreihenfolge eines Spielabenteuers, ohne direkten Eingriff in die Statechart-Logik, zu manipulieren. Anhand von Aufgabenschablonen, die die Statechart-Logik einzelner Aufgaben kapseln, ist der Anwender in der Lage die Reihenfolge der Aufgaben beliebig zu manipulieren und miteinander in Beziehung zu setzen. Abbildung 6 zeigt wie im Level Designer die Spielschablonen als graphische Elemente miteinander verbunden und umsortiert werden können.

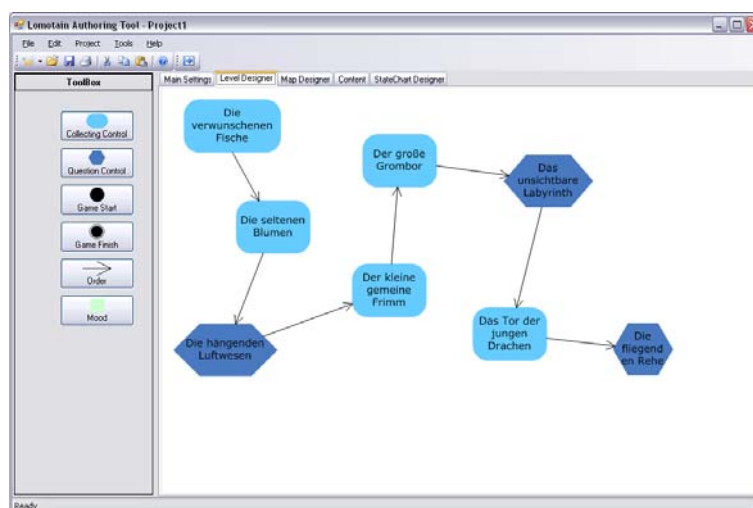


Abbildung 6: Screenshot des Funktionsmoduls „Level-Designer“

5. Zusammenfassung

Neben einem kurzen Überblick über die grundlegenden Konzepte der Lomotain-Plattform und einer Einführung in die zugrunde liegende Statechart-Logik wurde im Rahmen des Papers vor allem die

Generierung dieser Statechart-Logik mit Hilfe einer entsprechenden Authoring-Umgebung präsentiert.

Im Rahmen des Projektes hat sich vor allem gezeigt, dass der Ansatz, die Spiellogik von der eigentlichen Anwendung abzukapseln und mit Hilfe von Statecharts zu definieren, einer der entscheidenden Erfolgsfaktoren war. Erst die durch diesen Ansatz gewonnene Flexibilität ermöglicht einen breit gefächerten Einsatz in verschiedenen Szenarien (Touren- und Adventure-Guide, City-Guide, ...) und eine schnelle und kostengünstige Anpassbarkeit der verschiedenen Spiele/Führungen. Ebenso hat sich gezeigt, dass die Unterstützung der späteren Anwender bei der Erstellung, Verwaltung und Anpassung dieser Statecharts durch entsprechende Werkzeuge ein weiterer wichtiger Faktor war, wobei hier insbesondere die intuitive Bedienbarkeit eine große Rolle gespielt hat.

Trotz der Verwendung von Java2ME, die in dem hohen Unterstützungsgrade auf der Anwendergeräteseite begründet ist, hat sich in Tests gezeigt, dass sich die dynamische Generierung der Spiellogik auf Basis des XML-Exportformats der Statecharts in einem vertretbaren Rahmen hält und nur beim Spielstart von Bedeutung ist. Negativ hat sich allerdings die Verwendung von SVG als Vektorformat für die Animationen erwiesen, hier kommt es auf Grund der geringen Leistungsfähigkeit der mobilen Endgeräte zu unververtretbaren Verzögerungen. Als Alternative wird derzeit die Unterstützung verschiedener Videoformate analysiert.

Im Rahmen verschiedener Projekte wird an einer Weiterentwicklung der Lomotain-Plattform gearbeitet. Insbesondere die Umsetzung der Authoring-Umgebung auf der Eclipse Rich Client Platform und eine Erweiterung der Statechart-Logik hinsichtlich einer kontextsensitiven Anpassung der Automaten auf die aktuelle Situation stehen dabei im Vordergrund.

6. Literatur

- [1] BENFORD, S., CRABTREE, A., FLINTHAM, M., DROZD, A., ANASTASI, R., PAXTON, M., TANDAVANITJ, N., ADAMS, M. und ROW-FARR, J. Can you see me now? ACM Trans. Computer-Human Interaction, 13(1),100-133, 2006.
- [2] BENFORD, S., FLINTHAM, M., DROZD, A., ANASTASI, R., ROWLAND, D., TANDAVANITJ, N., ADAMS, M., ROW-FARR, J., OLDROYD, A., und SUTTON, J. Uncle roy all around you: Implicating the city in a location-based performance. In Proceedings Advanced Computer Entertainment, ACE, Singapore, 2004.
- [3] BOLL, S., KRÖSCHE, J. und WEGENER, C. Paper chase revisited – a Real World Game meets Hypermedia. In Proceedings International Conference on Hypertext (HT03), Nottingham, UK, 2003.
- [4] CHEVERST, K., DAVIES, N., MITCHELL, K., FRIDAY, A. und EFSTRATIOU, C. Developing a Context-Aware Electronic Tourist Guide: Some Issues and Experiences. In CHI'00, S. 17–24, Niederlande, 2000.
- [5] DUFFIN, B. Local Treasures: Geocaching Across America; Cartographica: The International Journal for Geographic Information and Geovisualization, 42(2), 2007.
- [6] HAREL, D. Statecharts: A Visual Formalism for Complex Systems, Science of Computer Programming 8(3) pp. 231-274, 1987.
- [7] KRÖSCHE, J., SCHEUCHENEGGER, M., DRAB, S. und SELINGER, S. Location-based Mobile Gaming and Entertainment im Tourismus – Lomotain. In Proceedings Georeferenzierung im Tourismus, Kitzbühel, S. 57-76, Österreich, 2008.
- [8] POSPISCHIL, G., KUNCZIER, H. und KUCHAR, A. LoL@: a UMTS location based service. In: International Symposium on 3rd Generation Infrastructure and Services, Athen, Griechenland, 2001.