Towards Location-based Games

D. Nicklas, Ch. Pfisterer, B. Mitschang

Abstract—In this paper we investigate the basic properties of location-based games. This new type of game is made possible by the recent advances of mobile computing hardware and infrastructure. Players act not by pressing buttons or moving pawns on a board, but by moving around themselves in the real world.
We present a simple classification of location-based games, and show how these games can be designed and implemented. With some adaptations, game concepts from existing board and computer games can be mapped to make location-based games more interesting and fun to play. Our methods are demonstrated with three actual game examples. Further, common infrastructure requirements are deduced and we show how the open platform developed by the NEXUS working group fulfills them.

I. Introduction

The rapid development of mobile computing hardware in recent years has led to devices which are not only capable of wireless data communication, but also provide the computing power for custom applications. Through positioning systems like GPS or GSM-based techniques, such devices can learn about their geographical location. This combination has opened up the road to location-based services. Typical examples for location-based services are route planning, tourist guides, yellow pages-like services or calling help in an emergency. One application area that has not been explored in depth so far are mobile, location-based games.

This paper is organized as follows. Section II discusses what location-based games are and what sets them apart from traditional computer and board games. In section III we discuss the game characteristics of mobile games. Section IV shows how existing board games can be converted to location-based games. Three example games are depicted in section V. Finally, section VI deduces implementation requirements and shows how they can be fulfilled by the NEXUS platform.

II. Location-based Games

In traditional computer games, players sit in front of a computer screen and interact with the game through a range of input devices like keyboards, joysticks and mice. The game takes place on one stationary computer or on several stationary computers connected through a LAN or the Internet.

In a location-based game on the other hand, the players themselves move around in the real world. They interact with the game by changing their position and visiting certain places that are of interest to the game. They still interact with their computers using various standard input devices, but that is secondary to the game. Players can meet in the real world, and interact with each other in the game context.

Since the players are mobile, they must be equipped with mobile computing devices. Recent cell phones and personal digital assistants serve this need well. Notebooks can be used for games with higher performance or visualization requirements. Some projects even use wearable Augmented Reality equipment [4], although the cost associated with that is prohibitive for casual users.

The devices used in the game are connected through a wireless communication network. At present, only GSM with its various data communication sub-protocols is up to that task. In the future that place may be taken by UMTS, Bluetooth ad-hoc networks or extended IEEE 802.11 networks.

Positioning can also be achieved through GSM. Recent GSM-based methods can discover the location of a mobile phone with an accuracy in the range of 100 meters (using Enhanced Observed Time Difference, [5]). When better precision is required GPS or Differential GPS can be used outdoors. Other methods using infrared beams or visual cameras [4] [7] can be deployed indoors and close to buildings.

Location-based games can be divided into three categories with different degrees of dependence on location information as follows:

- *Mobile games*. In this kind of game, game events only occur when two players happen to meet. The game doesn't need to track the full position of the players. Proximity sensing and local communication are sufficient.
- *Location aware games*. In this kind of game, the geographical positions of the players matter. Game events can happen when a user visits a certain location.

mailto:[danickla|mitschang]@informatik.uni-stuttgart.de,
Institute of Parallel and Distributed High-Performance Systems (IPVR), Breitwiesenstr. 20-22, 70565 Stuttgart, Germany.

- *Spatially aware games*. This kind of game integrates the real world surroundings of the players into the game. Buildings, roads and the landscape are available for use in the game. Game events occur when the player is in a certain spatial context (e.g. entering a church – wherever it is).

Clearly, this is an inclusive property, that is, any spatially aware game is also location aware, and, in turn, any location aware game is also mobile.

III. Game Characteristics

As we will see below, the style of gameplay depends on the characteristics of a player's mobility.

A. *Communication*

In a board game all players are assembled in one room and can communicate directly. The same applies to computer games played at a typical LAN party. In these cases, communication cannot be targeted at specific players (e.g. only members of your team) – everyone gets to hear everything. To allow targeting, most computer games provide for communication via text messages. This is also useful for Internet play. The drawback is that typing text distracts attention from the game.

Location-based games need to adopt a similar communication style. Since players are not close enough to each other to communicate directly, the game must provide communication channels. Voice communication is preferable, but text messages may be the only choice when bandwidth is low. A location-based game can offer a range of targeting choices for messages. Apart from targeting specific players, e.g. one specific player, all teammates or all players, a location-based game can target players by location, e.g. all players in a public place or all teammates within 20 meters.

B. *Rule Violation*

Security and rule violation become real issues when there is no direct eyesight between players. Contemporary online computer games come with a wide range of cheating counter-measures. Most violations can be blocked by the software before they happen.

Location-based games don't have this privilege. The software that drives the game has no control over the physical world, so rule violations cannot be prevented. However, violations can be detected by the software and reported to the other players. This acts as both a deterrent and as an actual control instrument. The effect can be boosted with a tableau system that marks regular cheaters.

C. *Massive Multiplayer Gaming*

Massive multiplayer gaming comes quite naturally to location-based games. Since the game takes place in the real world, space is no constraint. There are no technical barriers, either. The mobile devices carried by the players constitute a distributed system. Thus there is a strong incentive to use distributed systems algorithms to implement the game. One must pick an implementation that scales well to large numbers of players.

IV. Converting Board Games

One focus of our research was transforming existing board games into location-based computer games. Depending on the game in question, some interesting problems show up. An overview of the resulting mappings is given in table 1.

For our purposes, we will distinguish game objects into pieces and items. A game piece has a fixed association with one player and usually marks a position on the game board. It is moved across the board according to the game's rules. A game item has no fixed association and can by acquired by any player. Items represent status or act as tokens for actions. Figure 1 shows an abstract, but typical arrangement.

Board Game Element	**Location-based Game Element**
Game board with discrete locations	Augmented World Model, virtual parcours through the real world
Game pieces moved by players	Players themselves move
Cards and other items	Virtual cards, virtual game objects
Turn-based gameplay	Real-time gameplay

Table 1: Basic mappings

A. *Mapping the Board*

When converting a board game to a location-based game, the game board must be mapped into the real world. Unfortunately, a typical game board consists of a set of discrete locations, which is hard to map into the continuous, physical world. We propose three ways to deal with this:

- Stick to discrete locations. Each game location is associated with a geographical zone in the real world. Players are only allowed to move between these locations according to the game's rules. This must be monitored by the game software.
- No discrete locations, just a parcours, which is a predefined path that corresponds with the game flow. Players can move freely along the parcours, but are not

allowed to take shortcuts. By changing the parcours in a dynamic and player-specific way, most rules from race-type board games can be implemented. For example, an attack by an opponent can force the affected player to go back part of the way.

- No restrictions. Players move around the real world freely. This may require major changes to the game concept. The game can still require players to visit certain locations, for instance to pick up virtual game objects.

B. Pushing the Pieces

Pushing actual pieces around the physical world doesn't make much sense in a location-based game. The natural solution is to let the players themselves take the place of the game pieces. That implies that the player herself is now subject to the game's rules.

A direct mapping between pieces in the board game and players in the location-based game is only possible when each player controls exactly one piece in the board game. Another group of board games gives each player a set of pieces to control. These can be mapped in the same way if enough players are available. For instance, if one player controls four pieces in the board game, there would be four players acting as a team in the location-based game. Unfortunately there are board games that allow any player to move any piece or that don't have moving pieces at all. These can't be converted to location-based games without reworking the game concept.

C. Game Items

Game items in board games come in all shapes and sizes – cards, wooden pieces, even small statues. While it is possible to carry and display such physical objects in a location-based game, it is inconvenient. Virtual game objects are not only more convenient, but also more flexible. Virtual objects can be made visible even to remote players and it is possible to transfer them to another player that is not in the vicinity. Furthermore, virtual objects can appear and disappear dynamically under the game's control.

D. Concurrency

The vast majority of board games are turn-based, meaning that only one player can act at any given time. This is not appropriate for real-world and real-time location-based games. Players don't expect a location-based game to restrict their freedom to move. So the natural choice is to adapt the game's concept to allow for concurrency.

Lifting turn-based restrictions can make a game unfair, because a player can now act continuously and thus cause more "damage" to other players. That calls for concepts that limit the agility of players to a fair level without reintroducing turns. The main idea of such concepts is making actions dependent on a resource with limited availability. Such resources can be inherent to the game, like in real-time strategy games. If inherent resources are not available one can introduce "action tokens" – each player is granted a limited amount of action tokens per time unit and each action in the game uses up one action token. Alternatively, each action can be assigned a time it takes to complete the action. If only one action is allowed to take place at one time this results in a sufficient constraint.

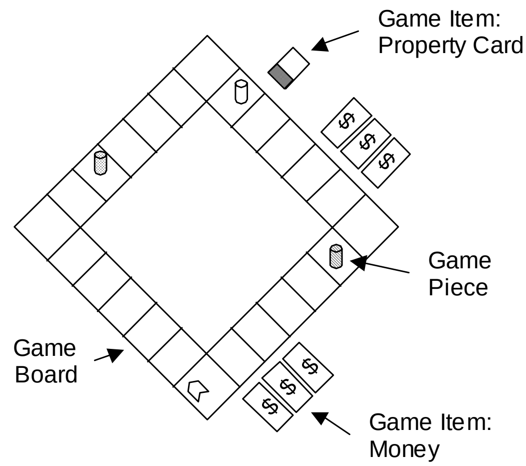

Fig. 1: A typical board game

E. Game Logic

The game logic in a board game is defined by the rules and fulfilled by the players. As stated in III.B, the compliance to the rules is secured by the players. In a distributed scenario, a trusted game server or special protocols must control this.

In a board game, changes on the board can be immediately recognized by the other players. For that, update protocols must be implemented, that either perform discrete updates (every 10 minutes or turn-based) or continued updating (every change to the game world is immediately visible for the other players).

V. Game Examples

To demonstrate our findings, we will present several examples of location-based games. They will be used later on to deduce requirements for the implementation architecture.

A. The Big Tournament

This first example is a simple game which can be classified as a "mobile game". It is a massive-multiplayer game organized as an open tournament. Players register with the tournament organization before the game starts. During the game, players are informed whenever they meet other players that take part in the game. When this happens, players have the choice to "flee" or to take a stand and play a small two-player action game like Tic-Tac-Toe or Snake against their opponent, using their mobile devices and infrared communication. Only the winner of that match continues in the tournament. The open tournament phase ends when there are only a small number of players left. They are invited to meet for the finals in a public place.

B. The Virtual Scavenger Hunt

In the second example, one player, the Fox, moves around in the world and leaves virtual marks. The other players try to catch him, following the hints he has left. This can include small puzzles to solve. A variation of this is a real world Scotland Yard game [6], where the players get the information which kind of transportation system (train, bus, underground train) the Fox was using. At certain times, the position of the Fox is also revealed. Together with a map of the public transportation system the players can try to catch the Fox. This game is "location aware", since it tracks and publishes the actual locations of the players.

C. Crossroads: A Real World Role Playing Game

The last example presents itself as a mobile, real-world counterpart to MUSHes (Multi-User Shared Hallucination). Instead of typing text commands to a central server to indicate their movements and actions, the players themselves take physical action. The result is a massive multiplayer online role-playing game (MMORPG) that takes place in the real world. The game uses the existing landscape and townscape and attaches new meanings to them. For instance a university library can become a place to buy magic spells and scrolls of wisdom in the game.

When players meet, they can interact in the game context by talking, exchanging goods and knowledge, or by fighting each other. Fights are carried out like in traditional role play games, using a complex point system that is essentially driven by chance, but takes character attributes, weaponry and experience into account. Players are free to choose any "career" they like – they can settle down in a peaceful community, make a living of their fighting skills, become a roaming trader, or try to gain influence and forge kingdoms.

In addition to player-player interactions, players can invoke special "adventures" involving computer-controlled, simulated monsters and puzzles.

Since the Crossroads game is sensitive to a user's environment, it falls into the "spatially aware" category.

VI. Architecture Requirements

Realizing location-based games with more than one player requires the coordination of distributed devices, and with the prospect of massive-multiplayer gaming a scalable, distributed architecture becomes necessary. In this section, we will deduce some requirements for an open, reusable architecture suitable for location-based games and show how they can be fulfilled by the NEXUS architecture.

A. Requirements for Location-based Games

Positioning and Location Management -- To build most location-based computer games, you will need some kind of positioning system to determine the position of the user. In the Big Tournament example, only the position of one user relative to another is relevant (to detect a possible meeting), which can also be achieved with proximity sensors], but in most cases, there is a need for geographical positioning. Ideally, this should provide accurate position measurements both indoors and outdoors. Unfortunately, this positioning system does not exist yet, and so at least two different sensors (for outdoors and indoors) must be used to provide a high coverage of possible game places.

Also, the positions of the users must be managed by a location service, to detect legal and illegal moves or to determine the recipients of a geographical message. This location service must be capable of a high number of updates and must scale well for a high number of users.

If virtual, mobile objects are used in the game, such a location service has to manage them as well.

Communication -- There is a need for communication between the player and the game control as well as between the players themselves. This can be human-generated messages, sounds, pictures or just game data. Similar to the positioning system mentioned above, different technologies have to be supported to provide the best coverage, lowest cost and highest bandwidth that is available in a certain context. For example, for short range communication between two players meeting each other (like in the Big Tournament example) the game can use infrared communication, which involves and provides no connection costs at a moderate bandwidth. For transmitting high amounts of game data (a video stream of some in-game event like a crowning in the Crossroads example) the players could enter a building that is equipped with a high bandwidth wireless LAN. And for sending small amounts of game data like a special command to all players regardless where they actually are, the Short Message Service can be used. There is also a need for supporting geographical messaging, which is sending a message to a certain area.

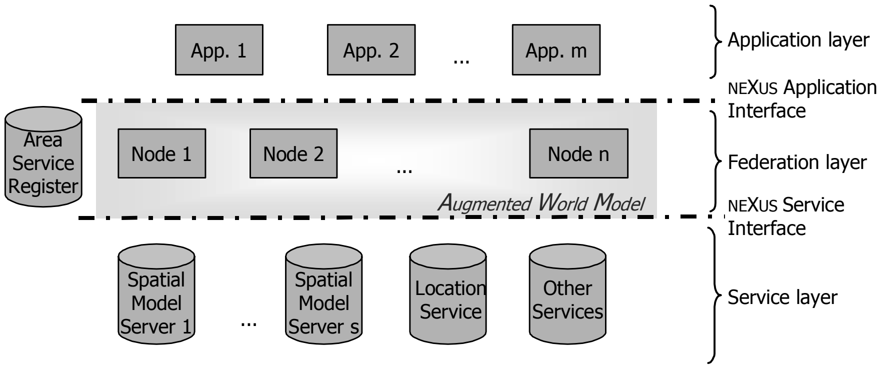

Fig. 2: The Architecture of the NEXUS platform

World model with real and virtual objects -- For complex location-based games like the Crossroads example, the game needs a model of the real world, with buildings, streets, important objects and even transportation facilities. To give those objects a new meaning in the game context, there are either general rules in the games that interpret real world objects in a new way, or the world can be augmented with virtual objects like the Magic Shop in the library or a simulated merchant that moves through the city.

Distributed location-based data storage -- For a stand-alone, simple location-based game there is no need for enhanced data storage. It can be run with local data storage attached to a single server. But if you want to build spatially aware games allowing a lot of users to participate, you will need to distribute the data storage. It would be convenient to have different servers that are responsible for certain game areas, and for mobile users. However, there must be hand-overs between the servers.

Open platform – Using an open platform that supports the requirements mentioned in this section in the context of location-based games leads to the following advantages:The tasks are quite the same for all applications and if for example the costs for maintaining the world model are shared, it is much cheaper to develop a new game. One can see the dynamism that strives from an open platform in the success story of the World Wide Web or Unix. Because everybody can participate and develop a part for the platform, there are lots of tools, support and services available.

In the next section, we present the **NEXUS** platform, which was initially developed for spatially aware applications in general, but which can be used for location-based games as well.

B. *The NEXUS platform*

The NEXUS architecture is an open platform for mobile, spatially aware applications [8]. It is built up in three tiers: NEXUS applications, nodes and services. If we compare this to the World Wide Web, NEXUS services play the role of web servers, while NEXUS applications are similar to web applications like browsers or applets. The middle tier, consisting of the NEXUS nodes, does not exist in the WWW architecture: it is an open federation of the underlying data providers, based on the location as the main integration scheme. The NEXUS applications only have to contact their closest NEXUS node to get an integrated view on all location-based data currently available. This view is called the *Augmented World Model*, which is a common data model for both the applications and the services (see below).

NEXUS applications are mobile and spatially aware, i.e. they run on a mobile client and communicate wirelessly with the infrastructure. Their functionality depends on the location of the user in relation to the world around, like location-based information systems or mobile navigation tools. We envision that for common mobile devices basic NEXUS client components are available, similar to WWW browsers or TCP sockets. They provide a simple query interface, basic communication functions to the platform and a standardized sensor component for different positioning systems. Additionally, there might be a simple NEXUS browser for the Augmented World Model and an Application Finder to detect new location-based services and applications. New NEXUS applications can use these basic services to provide more sophisticated spatially-aware functions and better user interfaces, just like web services do use HTTP and browser technology in the World Wide Web.

NEXUS Nodes mediate between NEXUS applications and NEXUS services. Every node has the same function. They are responsible for distributing queries to the different services and for composing the results, thus providing an integrated and consistent view to the applications, the *Augmented World Model* (AWM). This view is an object-oriented information model of the real world, which is augmented by additional information (called Virtual Objects).
To integrate the different data of the underlying services, NEXUS nodes use the Area Service Register, which relates geographical areas to the associated Spatial Model Servers. This enables an efficient selection of the minimal subset of servers required to process a certain query.

In the service layer, *Spatial Model Servers* (SpaSes) are used to store the static objects of the *Augmented World Model*, like buildings, streets or Virtual Information Towers (VIT), which are used to assign external data (e.g. WWW data) to locations [1]. A VIT is a Virtual Object that contains information which is relevant to a certain geographical area. If a user enters this area, the VIT becomes visible, just like a real information tower is visible only in certain places.

The objects are organized in so called *Augmented Areas*, which are coherent sets of NEXUS objects (i.e. the buildings of one town or the rooms of one building). SpaSes are not suitable for storing mobile objects because of the high number of updates. Instead, a *Location Service* [2] is used to manage them. Furthermore, the NEXUS architecture contains *other services* like an Event Service for managing spatial events, which are not discussed in this paper.

C. Building Location-based Games with NEXUS

Building a location-based game within an existing platform like the NEXUS architecture has some advantages over building an isolated application.

Use of existing infrastructure -- The standardized communication components make the application independent from the physical protocol and enable seamless transitions from one network to another. The sensor component can use different sensors to locate the user. This enables the development of games that work in many different contexts (like indoor or outdoor) without a high effort on device issues.

The data necessary for a particular game can be stored on a Spatial Model Server, and will then be integrated into the Augmented World Model by the federation layer. Thus the game application can get the integrated view of both the game data and information that is gathered from other NEXUS services.

The NEXUS Location Service can manage a high number of users, which is necessary for massive multiplayer games like the Crossroads example.

Use of existing data -- In an open platform for spatially-aware applications data is already available which can be used in the context of the game. For example, a city could maintain a model of the downtown area for the Tourist Guide Application they provide to visitors. A game developer can now rely on that model and add the game context to it, implementing the Crossroads game. Also the games can become more vivid and interesting if they use existing, real-world data like recent WWW pages.

Be part of the community -- Like mentioned above, developing for an open platform has some technical advantages. Additionally, there will be other developers of location-based games that can give support and helpful tools for the development process. And the platform can be used to announce and deploy the game itself, too.

What to do? -- So if you want to build a location-based game within NEXUS, there is a lot of support and components available. In [3], we have described a step-wise approach for building NEXUS applications. In short, that is: develop just the application logic, visualization and data design (not the positioning and communication infrastructure), deploy your new application using the platform itself and have fun!

VII. CURRENT AND FUTURE WORK

Currently, the NEXUS project is a research group spread among four departments at the University of Stuttgart. We are building a small-scale prototype of the NEXUS platform described in Section VI.B. With that, we will evaluate the usability of the platform through the development of several mobile, spatially aware applications. One will be a location-based information system, a city guide that is based on the VIT application [1], but we also plan to develop some location-based games next year.

VIII. CONCLUSION

Location-based games are a promising new application of mobile computing and mobile communication technology and can help in driving the adoption of next-generation mobile communication networks.

Converting existing board and computer games to location-based games creates a rich pool of game concepts. A new aspect of location-based games is that you can play them anywhere and anytime. You don't have to enter a special game situation, but you can play in your real live whenever you encounter other players, game items or places relevant to the game – or they can encounter you [9].

By using an open platform to build such games, we also hope to spur the development of location-based games and spatially aware applications in general. The open architecture makes the development easier and more attractive in the first place, but also eases the formation of a community with an open exchange of ideas, algorithms and code. We believe that games appeal to a larger community of developers than traditional applications. Attracting these developers ultimately helps advance the open platform itself and thus all applications that use it.

IX. REFERENCES

[1] A. Leonhardi, U. Kubach, K. Rothermel, A. Fritz: Virtual Information Towers - A Metaphor for Intuitive, Location-Aware Information Access in a Mobile Environment, Proceedings of the Third International Symposium on Wearable Computers (ISWC'99), San Fransisco, CA, USA, IEEE Press, 1999.

[2] A. Leonhardi, U. Kubach: An Architecture for a Distributed Universal Location Service. Proceedings of the European Wireless '99 Conference, Munich, Germany, pp. 351-355, ITG Fachbericht, VDE Verlag, 1999.

[3] D. Nicklas, B. Mitschang: The NEXUS Augmented World Model: An Extensible Approach for Mobile, Spatially Aware Applications. Proc. of the 7th International Conference on Object-Oriented Information Systems, 8/2001.

[4] B. Thomas, B. Close, J. Donoghue, J. Squires, P. De Bondi, M. Morris, W. Piekarski: ARQuake: An outdoor/indoor augmented reality first person application, Fourth International Symposium on Wearable Computers (ISWC'00), Atlanta, Georgia, 2000.

[5] Cambridge Positioning Systems (2001), „CPS Home Page", URL: http://www.cursor-systems.com

[6] Scotland Yard Game. Projekt-Team III: Scotland Yard. Ravensburger 1983.

[7] S. Björk, J. Falk, R. Hansson, P. Ljungstrand: Pirates! - Using the Physical World as a Game Board. Paper at Interact 2001, IFIP TC.13 Conference on Human-Computer Interaction, July 9-13, Tokyo, Japan.

[8] F. Hohl, U. Kubach, A. Leonhardi, K. Rothermel, M. Schwehm: "Next Century Challenges: Nexus - An Open Global Infrastructure for Spatial-Aware Applications", Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99), Seattle, Washington, USA, August 15-20, 1999, T. Imielinski, M. Steenstrup, (Eds.), ACM Press, 1999, pp. 249-255.

[9] A&B Producoes, Lda., PolyGram Filmed Entertainment and Propaganda Films: The Game. Motion Picture, directed by David Fincher, 1997.