ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2006, Vol. 35, No. 3

CONTEXT-AWARE INTERACTION AND NAVIGATION IN MOBILE GAMES

Silke Jahn, Aizhen Liu, Mihail Dimitrov, Michail Mazo, Friedrich Jürgen, Krauss Matthias

University Bremen

Abstract. In this paper an approach of using contextual information for structuring and displaying menus on small devices will be discussed, based on the implementation of a game for mobile games. CitizenMOB is a location-based, multiplayer, never-ending society-driven strategic mobile game that has been developed in order to understand today's possibilities and challenges in the design of complex games for mobile phones. Integrating a context-aware navigation and adaptive menu structure is an attempt not only to reflect the effect of new contexts of use on human-computer-interaction, it is also meant to overcome usability problems that occur when limitations of small screens are combined with complex rules and massive options in the next generation of rich mobile applications.

Keywords: context sensitive navigation, context aware menu structure, small screens, mobile gaming.

1. Introduction

Mobile systems have become ubiquitous and so did the amount of commercial applications for such devices. Mobile games have emerged and have technically progressed rapidly during the last years. But as those games become more and more complex, graphically sophisticated and detailed, new usability challenges have added to the old problems of designing for mobile devices.

In this paper, an approach of structuring and organizing a context-aware menu in order to meet some of those usability problems will be described by means of the game CitizenMOB. The game and the challenges to design that resulted mostly from its complexity will therefore be described in section 2. In section 3 the analysis of usability problems of mobile games will be discussed deeper and some guidelines proposed. Section 4 deals with the discussion of the concept of context-awareness and our approach will be explained. The practical conversion of the concept follows in section 5.

2. CitizenMOB – A mobile game

The key concept of the game CitizenMOB is to provide the player with sophisticated mixed reality experiences, using location tracking technology, story narration and virtual characters, which are the link between virtual and real world. Merging the real with the virtual world, the game is set to represent the player's location within the real world. Virtual aspects of the game are meant to be additions to the real life gaming experience, tools for socialising with other players, gaming and interaction, rather than providing the playing with a new concept of place and time. In order to play the game within the virtual and real world, the player has to control the behaviour of a Virtual Life Form (VLF), a digital customisable character. As the VLF is more than just a digital representation of the player's decisions – a creature with own will and moods – control is conducted not completely directly.

Regarding player-game interaction, main focus of the game can be seen in two aspects: maintenance of the creature and training / competition. In order to maintain the VLF, goods have to be gathered and traded and time has to be spent on the VLF to keep it in good mood and condition. The VLF has to be trained, too, since it might compete against other creatures, humanly and artificially controlled. While it can be trained by playing little games which require either strength or intelligence, competition with others takes place based on the players' actual location, interacting via short distance or online connection.

Interaction complexity / problems

When creating CitizenMOB game, challenges emerged in the interaction and navigation part of the application. General features of the game, such as that all control of the gaming process is done completely via the mobile client, as well as overall complexity of the application, raised the need of new approach to game navigation. The factor of limited space of a mobile screen stressed the problem. How to make all interactions and control of the game easily accessible to the user? How to enhance the usability of the

application? How to not harm the game functionality at the same time?

3. Interaction design challenges of mobile phones

Today's game development is still challenged and in many cases limited by what is actually possible to play. Unlike big screen computer and console games on which visions of game play are already reality, applications for mobile devices and especially mobile phones have yet to overcome technical limitations in order to produce challenging and engaging gaming experiences.

Besides considering bandwidth restrictions due to monetary reasons, limitations of phones compatibility and design limitations (J2ME drawing API), current forms of in-and-output systems prove to be the major problem in the development processes of mobile games in general and CitizenMOB in particular. Tedious input systems as "typing into a numpad" slow the user down and result in demotivating and discouraging effects on her if requested frequently. Small screens, on the other hand, considerably limit the overall usability of applications, the possibility to create engaging graphical environments and therefore challenging game play as they:

- limit the number of detailed characters and other graphical game parts to be displayed,
- require to clip or scale down information and graphics, which may render them unrecognisable for some users,
- provide little space for integrating unobtrusive help screens,
- reduces freedom of design decisions due to expanded need of contrast and clearly visible graphics.

All these issues concerned with the menu structure of games result in the following problem: small screens allow only a limited number of menu items to be displayed without scrolling. While scrollable menus would allow to insert an unlimited number of menu items it heavily reduces the speed of interaction and also isn't any help in providing a quick overview of possible moves and the functionality of the application. On the other hand, resigning on only few interaction options would mean to reduce the possibility to produce a game as planned with its large scope of rules and functionality.

Beside the design challenges for small screens on mobile devices, there is another problem to deal with when developing applications for mobile use. The working environments in which mobile phones are used differ greatly from that of traditional desktop PC work contexts. Kristoffersen and Ljungberg [5] point out four significant characteristics of mobile use after analyzing (analysing) the work of telecommunication service engineers and maritime consulting staff that can also apply to the vivid and highly active environment in which the game CitizenMOB takes place[1]:

- task hierarchy: external tasks the user is committed to are central while tasks that are processed within the device are supplementary,
- visual attention: the user focus is rather on the surroundings than what happens on screen,
- hand manipulation: while using the device, the user is also occupied with handling other objects in the physical world,
- mobility: the user may be highly active and uses the device in various states of mobility (e.g. seated, standing, moving, even running).

Those characteristics define a work environment in which the user cannot exclusively concentrate on operating the game and is likely to be occupied with other tasks (physical and cognitive). As a consequence, users of mobile devices have also little patience for learning to handle new applications as Chincholle et al. [3] argue. This results in the following requirements for designing usable interaction for mobile devices we have considered during development:

- providing fast access to information and functionality,
- allow high-speed interaction by providing control via numpad or touch-screen rather than keyboard control,
- design of brief human-computer dialogues if possible,
- communication of functionality during the use rather than providing lengthy manuals,
- intuitive user guidance,
- audible feedback.

In order to be able to develop a game with its complex rules and highly active gaming context a solution for those problems has to be found. Our approach, integrating a navigation structure that would use contextual information in order to structure, display and guide human-computer-interaction in an effective and efficient way, will be described in the following section.

4. Description of approach for contextual interaction and navigation on mobile phones

Context-awareness is "the ability of a device or program to sense, react or adapt to its environment of use" (Pascoe et al. [7]). The context or environment of use may vary due to location of the user, date or other devices, which connect to the system. Context-aware devices benefit from providing support for the user by

[1] Usually games are played in free time, generally by a player uncommitted to other tasks and free from disturbances. Thus their context of use differ from applications used in active work environments. However, playing CitizenMOB requires the user to participate actively in traffic and the search for or hiding from opponents in the real world.

adapting the system to varying tasks in changing situations. The context in which mobile devices are used differ from the work environment of common desktop computers, an important difference is the lack of time to spend on handling the device in a very dynamic environment. Anticipating user's needs by monitoring and analysing contextual information and adapting to them in order to provide easy and fast access to program parts and information is therefore the key advantage of context-aware computing.

CyberDesk, for example, is a component-based framework that supports automatic integration of software applications (Dey et al. [2]). CyberDesk automatically provides their services to the user. Rather than displaying the services of different programs to the user all the time, the CyberDesk's interface limits to those services that are relevant to the user's context. For example, when a user highlights a hyper-link in a text, corresponding services are provided, they ask the user whether she wants to search for the string or to open the URL page. In this software, user contexts, including the data the user is working with, the time of day, the user's physical location, emotional state, social environment and objects in the room, are used to enhance and speed-up the work-flow.

Yan and Selker [9] describe the office assistant for which contextual information also plays a crucial role. The Office-Assistant is an agent that interacts with visitors at the office door / the entrance to an office and manages the office owner's schedule. The identity of the visitor; the office owner's schedule status; the office owner's busy status and her willingness to accept meeting proposals are the contextual information used in the system. For example, when a visitor interacts with the office assistant, she will be asked about her name, and according to her name, the Office-Assistant will know if there is an appointment or suggest to set a new one. If she wants to make an appointment, the Office-Assistant will arrange one according to the owner's schedule.

The GUIDE Project (Cheverst et. al. [1]) describes how context-aware mobile devices allow new approaches and services to applications such as electronic tourist guides. In this hand-held context-aware tourist guide, the information presented to visitors by the system should are tailored to their context. Three broad classes of contextual information are used: personal information such as the user's interest, location awareness and date & time awareness such as the time of day and opening times of attractions. Using the tourist guide, the user can visit the city by accessing context-aware information and services that she is interested in.

Finally, Lankoski et al. [6] point out how contextual information can be used to create innovative pervasive games. They describe a location aware mixed reality game called "The Songs of North". Their approach is to connect the physical environment to the game using information or context gathered from the physical environment. This approach coincides with parts of the CitizenMOB game's concept, as CitizenMOB is a location based game as well, merging the real world with the virtual game.

Similarly to CitizenMOB, in these systems contextual information is gathered to enhance existing application services or to create novel services, to provide new content. But, in the development of CitizenMOB, we also thought about new possibilities to integrate those information in a useful way into the application. Our idea is to combine the concept of context-aware computing with an adaptive design of the overall menu structure of the game. According to the model of Gorlenko and Merrik [4], who divide context-awareness into the five categories of "location awareness", "environmental awareness", "mobility awareness", "health awareness" and "activity awareness" the game particularly utilises information that belong to the first and the last category. We argue to expand these categories in order to include a last category of "date & time awareness" as these generally influence the way how a system is used. This not just correlates with our experiences, the examples of context-aware applications mentioned above integrate date and time as specific factors, too.

Location awareness is the ability to determine the user's location and to provide her with related information (Gorlenko and Merrik [4]). In CitizenMOB the user's position is displayed on a map which not only represents the actual physical world but is additionally enriched by virtual elements such as game characters, buildings or items. Depending on the location of the user different kinds of activities can be undertaken. A typical example of using location aware unformation in CitizenMOB is the ability to explore an item on the map. When the user arrives at a real world location which is connected to an item, a special popup apears and shows information about the item together with a pick-up dialogue. Simultaneously, the navigation menu structure is changed. The left and right softkeys change their value according to the actions, than can be done on the item.

Activity awareness is the ability to track current physical or cognitive activities of the user such as reading, watching TV, or writing (Gorlenko and Merrik [4]). The game CitizenMOB provides different states of game-play which require different kinds of cognitive and physical activity of the user. For instance, game activities like training, fighting, chatting not only require different cognitive activities, e.g. fighting requires fast clicking reaction and chatting requires the user to read and write, they also set the user in different states of physical activity. When the system can assume, that the user is in high active mode, it adapts its dialogue with her. For example, is the user running because of a training game, the game will send "vibration messages" instead of simple pop-ups to gain the user's attention.

Date & Time awareness is the ability to integrate information about date and time usefully to adapt the

services to user needs. In CitizenMOB time and date play a crucial role, for example in the interaction with AI characters and it even influences behaviours of the player's VLF. Menu navigation and outlook are also influenced according to date and time factor. An example is the "night vision" function, which automatically changes the main color scheme of the menu and softkeys according to the day time.

Those different factors will define which navigational and game-play options will be displayed in the game menu. This structure will be discussed more concretely in the next section.

5. Context aware menu structure in the game CitizenMOB

The first idea was to develop a system with a main navigation concerning major tasks and additional sub menus that would appear on the screens on occasion, reflecting the current state and, according to that, the interaction in the game play. While the main menu, which was connected to the left soft key, would contain static navigation steps such as saving, quitting, map scanning, those sub menus would contain changing entries for various dialogues between player and game or in the interaction between human and human and appear within the screen.

In the example in Figure 1, the game option "use item from inventory" is visualised. After choosing the category "inventory" from the main menu the user is led to the sub menu from which she can choose again between different actions.

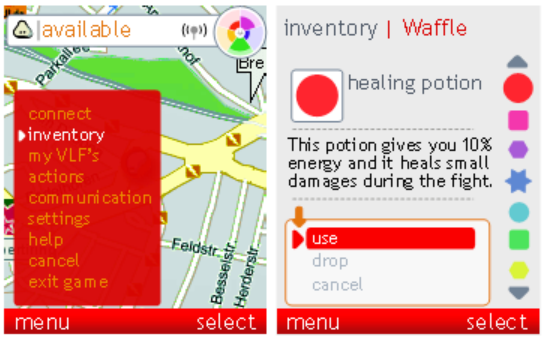
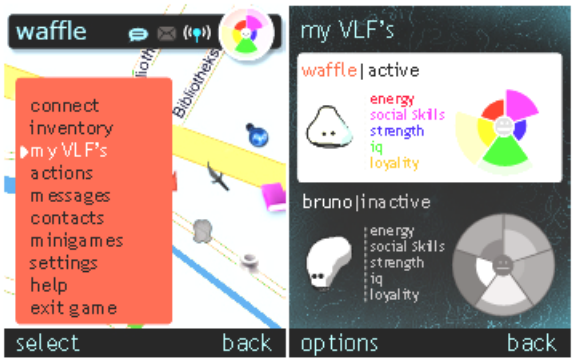


Figure 1. Main menu and sub menu, first draft
main menu is activated by the left soft-key and pops up, the sub menu appears within the screen

However, although this first idea of structuring the game navigation already provided a solution for the requirements on the navigation necessary for CitizenMOB as described in section 3, it keeps the main navigation clear but still offers plenty different options to interact with the system, it soon proved to be a bad design decision. Neither corresponds this way of structuring and handling the sub menus to a known convention nor did we find that inexperienced user managed to access the sub menu by intuition. The result of this design was the abandonment of easy handling in favour of having immediate access to the main functions during every step of conducting a task.

The new navigation structure relies on the main menu only when navigating through the system. By gathering and processing different context information including location, time and date, presence of other players, interaction options only appear if actual usable and conductible in the adaptive former main menu. In order to keep the size of this menu in proportion to the overall screen size of the device, options of interactions are bundled by functionality. Within certain steps of interaction processes (e.g. the interaction process "trading objects") the main menu will be replaced by local sub-menus. In order to support the user to differentiate between the main menu and those sub-menus easily, we decided to name the latter "options menu" and visualised them accordingly Figure 2 b.

In Figure 2, the user can handle one of her several VLFs. Instead of getting into a sub menu after choosing the menu item "my VLF's", the user gets access to the sub menu via the left soft-key and has to complete or abort this task before getting back to the actual main menu with its main game functionality. By sticking to very few structures of organising navigation and interaction, the game can be played without relying on the player's willingness to gain an understanding of different interaction concepts for playing but using the conventional navigation through the system via soft-keys.



a) b)

Figure 2. (a–b) main menu and sub menu, second draft
both main and sub menu pop up by clicking the left soft-key

One of the main occupations in the game CitizenMOB for the user is to travel the real world in search of challenges, treasures, items and opponents. According to contextual information which are gathered by using basic functions like ascertainment of time and date but also sophisticated technology like GPS or GPRS, options of interaction within the game may increase. For example, when encountering another VLF on the map, the "action" panel is added to the main menu dynamically, as it contains functions that have become a valid option of activity. The advantage of this approach is that game functionality will be more transparent as the process visualises the connection between the appearance of interactive elements on screen and an enhanced room for manoeuvre. This way, game concepts can be understood while playing

the game and walking through the virtual world without forcing the user to read through excessive game descriptions.

In Figure 3 after connecting to the game server via GPRS in order to scan the virtual world for other VLFs, the player's system gets notified that other players are nearby. Now, not only does the image representation of the strange VLF appear on screen, the program also adapts to the new situation in setting up the navigation for contacting the other player. The "action" option pops up in the main menu (Figure 3 b). By clicking on it a list of other VLFs nearby appears from which one can get selected (Figure 3 c). After choosing one of the VLFs from the list, different kinds of interaction can be started via the sub menu (Figure 3 d).

Figure 3. (a–d) context-aware menu accommodation

6. Conclusion

Chances of context aware computing not only lie in its ability to increase services of applications, enrich game experiences and also provide a new level of adaptation to user's needs in mobile applications. In the game CitizenMOB, contextual information was used to overcome some usability problems that came with the design of a complex mobile game. By making the menu structure flexible and customisable to ever changing game-play activities and locations we were able to reduce the menu size and provide faster access to commands and activities of complex applications.

On the other hand, the adaptive structure of the navigation lets an application react quickly and clearly to the user and his decisions. According to the player's choices the menu reacts, makes consequences apparent and instantly shows increased or decreased room for manoeuvre.

By using the context-aware menu structure we were also able to introduce a way to teach functionality and game concepts within the game-play, while keeping the complex rules and rich graphics that define the game CitizenMOB.

We therefore resume that using context-aware data to structure menus of complex mobile applications is an effective and efficient way to enhance the usability of such programs.

References

[1] **K. Cheverst, N. Davies, K. Mitchell, A. Friday.** Experiences of developing and deploying a context-aware tourist guide: the GUIDE project. *Proceedings of the 6th Annual international Conference on Mobile Computing and* Networking (Boston, Massachusetts, United States, August 06–11, 2000). *MobiCom*'00. *ACM Press, New York*, 2000, 20–31.

[2] **A.K. Dey, G.D. Abowd, A. Wood.** CyberDesk: a framework for providing self-integrating context-aware services. *Proceedings of the 3rd international Conference on intelligent User interfaces (San Francisco, California, United States, January* 06–09, 1998). IUI '98. *ACM Press, New York*, 1998, 47–54.

[3] **D. Chincholle, M. Goldstein, M. Nyberg, M. Eriksson.** Lost or Found? A Usability Evaluation of a Mobile Navigation and Location-Based Service. *F. Paternò (Ed.): Mobile Human-Computer interaction 2002, LNCS 2411. Springer-Verlag Berlin Heidelberg*, 2002, 211–224.

[4] **L. Gorlenko, R. Merrick.** No wires attached: Usability challenges in the connected mobile world. *IBM Systems Journal*, 2003, *Vol.*42, *No.*4, 639–651.

[5] **S. Kristoffersen, F. Ljungberg.** Making Place to Make IT Work: Empirical Explorations of HCI for Mobile CSCW. *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work, ACM Press,* 1999, 276 –285.

[6] **P. Lankoski, S. Heliö, J. Nummela, J. Lahti, F. Mäyrä, L. Ermi.** A case study in pervasive game design: the songs of north. *Proceedings of the Third Nordic Conference on Human-Computer interaction (Tampere, Finland, October* 23–27, 2004). *NordiCHI* '04, *Vol.* 82. *ACM Press*, *New York*, 2004, 413–416.

[7] **J. Pascoe, N. Ryan, D. Morse.** Issues in Developing Context-Aware Computing. *H.-W. Gellersen (Ed.), HUC'99, LNCS 1707, Springer-Verlag Berlin Heidelberg*, 1999, 208–221.

[8] **J. Pascoe, N. Ryan, D. Morse.** Using While Moving: HCI Issues in Fieldwork Environments. *ACM Transactions on Computer-Human Interaction*, 2000, *Vol.* 7, *No.* 3, 417–437.

[9] **H. Yan, T. Selker.** Context-aware office assistant. *Proceedings of the 5th international Conference on intelligent User interfaces* (*New Orleans, Louisiana, United States, January* 09–12, 2000). IUI '00. *ACM Press, New York*, 2000, 276–279.

Received June 2006.