PROTOTYPING MOBILE GAME APPLICATIONS
PRACTICAL USE EXPERIENCES

Johan Sanneblad, Lars Erik Holmquist

Abstract: Mobile game devices today include novel features, such as ad-hoc wireless connections, advanced audio and visuals, and peer-to-peer networking. Using these devices to design new game concepts, however, can be difficult and take much time. We have developed OpenTrek – a free, Rapid Application Development game platform for mobile devices. OpenTrek is to be used as a prototype platform on standard Personal Digital Assistants (PDAs) to quickly visualize new mobile game concepts. In this paper, we describe the platform and present results from the platform in use in an educational setting. Use experience indicates that the platform successfully can be used to prototype games for handheld devices in a limited time frame.

Key words: Mobile computing, hand-held devices, games, prototyping

1. Introduction

Inexpensive handheld game devices are emerging, providing features such as peer-to-peer networking, advanced graphics and access to online services. Creating interactive games for these devices can be difficult, with developers having to consider not only different hardware configurations, but issues such as limited screen space, restricted CPU instruction set, low memory bandwidth and small cache sizes as well. Many of these issues come from the lack of game middleware software such as DirectX [4], which is only available on stationary computers.

We have developed OpenTrek, a Rapid Application Development and prototyping platform for mobile game devices. OpenTrek comprises most of the features available on game middleware platforms on stationary computers such as DirectX [4], SDL [8] and RAD Game Tools [7], and is specifically designed to be used in prototyping projects. Using OpenTrek it

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35660-0_65

R. Nakatsu et al. (eds.), *Entertainment Computing*

is possible to not only prototype game software for handheld devices, but to create final shipping software products as well (see e.g. the game PocketKobo [6]).

In this paper we present practical use experiences of OpenTrek in an educational setting. OpenTrek was used by a class of M.Sc. students to create the ad hoc networked game CaféTrek in a limited time frame. CaféTrek comprises twelve game modules communicating with each other using a common middleware.

2. Game Prototyping

Game prototyping is used in classroom settings to teach game development concepts (see e.g. [3]). Using high level languages on stationary computers, students learn the principles of game design and are able to implement simple proof of concept applications. Low level game middleware platforms such as those used in commercial games (e.g. DirectX [4]) are generally considered too large and too complex for use in the limited time frames imposed on the students [3].

The use of high level languages imposes several restrictions to what types of games the students are able to create. Languages such as Java are not capable of providing visually interactive graphics suitable for games, even on stationary computers. Most games today require the use of not only full screen access, but direct access to the display hardware as well.

Handheld game devices such as the Cybiko [2] or Personal Digital Assistants (PDAs) do not provide high level languages or game middleware software. Creating games for these devices requires the developer to not only use a low level language such as C/C++ or assembler, but also to access the device hardware directly. This requires knowledge not only in embedded device development, but specific knowledge on the device itself.

3. OPENTREK

To enable games prototyping on handheld devices we have created OpenTrek, a game middleware platform specifically designed for mobile game devices. The OpenTrek architecture was designed to be as similar to the stationary game middleware platform DirectX [4] as possible. We chose the DirectX platform as a reference platform due to its long term use and widespread availability.

OpenTrek comprises most of the graphics and network features found in DirectX, and also adds several features found in other platforms such as

RAD Game Tools [7] and the Simple DirectMedia Layer [8]. These features include graphics features such as real time alpha blends, rotation, zoom, fonts and collision masks, as well as network features such as automatic multicasts, message timeouts, congestion control and message fragmentation. The design of OpenTrek was motivated based on the following issues with games development for handheld devices:

1. Game development for handheld game devices requires the use of a low level language, such as C or C++.
2. Developing games for handheld devices requires the developer to directly access the device hardware. This requires specific knowledge not only on the device itself, but also knowledge on embedded development in general.
3. Without a game middleware, the developer has to create efficient graphics routines as well as complete network support (including session initialization and session management). This can provide a high threshold at the beginning of a project, preventing possibilities of prototyping.

OpenTrek targets these issues as well as provides additional features for games prototyping. The features of OpenTrek enabling prototyping are:

1. High level class hierarchy. OpenTrek uses a high level C++ class hierarchy to interface with the device hardware (where stationary middleware tools often use C (SDL), COM (DirectX) or other proprietary interface mechanisms).
2. OpenTrek Launcher. All games have to provide a main interface to start and manage game sessions. The OpenTrek Launcher provides this, and also enables the developer to customize its appearance using Lobby modules.
3. Automatic network session management. Multiplayer games need a way for devices in ad hoc peer to peer networks to physically find each other. The games also need a way to manage players joining and leaving the different game sessions. OpenTrek provides awareness support as well as a session initiation protocol optimized for handheld devices.

We will now describe the OpenTrek Launcher and the OpenTrek Session Management in more detail. Both the OpenTrek Launcher as well as the OpenTrek Session Management comprises features not found in stationary game middleware platforms, and were specifically added to enable games prototyping on handheld game devices.

3.1 OpenTrek Launcher

The OpenTrek Launcher is a cross platform client application used to start games built using the OpenTrek platform. The OpenTrek launcher was added to the OpenTrek platform to provide an easy way to start and manage game sessions, by providing support for custom Lobby modules (a lobby is an interface whose primary purpose is to enable players to arrange games and meet). By design, the OpenTrek Launcher operates in a way similar to MSN Messenger [5], where a list of "online" users is represented using a Lobby module. In OpenTrek, online users represent people who have their device switched on and are using the same network hardware as the local user. The OpenTrek Launcher ships with a fully functioning Lobby module, allowing the game developer to disregard the design of such an interface until later design phases. The OpenTrek Launcher and a typical Lobby Module are pictured in Figure 1.

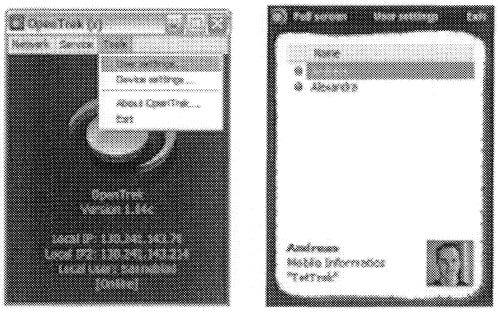

Figure 1. The OpenTrek Launcher on Windows XP (left), and the OpenTrek Launcher running a Lobby module (right).

3.2 OpenTrek Session Management

Handheld devices can be switched on and off instantaneously, which sets some design implications for how game sessions should be initialized. When a player wants to initiate a game session with some other players, the OpenTrek Launcher initializes game sessions by starting the first two devices in a session synchronously, where the rest are allowed to join asynchronously. This approach was a user design suggestion from early tests using the platform, where map data used in a game session had to be pre-calculated before allowing players to join. The synchronized approach enabled the game developers to split the map creation process on the first two devices in a session, and hold other devices from joining until the map creation process was complete.

3.3 Implementation

The OpenTrek platform was implemented on the handheld device IPAQ 3630. OpenTrek was created in C++ and requires game developers to use the same language. All C++ interfaces were designed specifically to prevent memory leaks and misuse, sacrificing some memory optimization and speed for safety.

4. Cafétrek

The game concepts for Cafétrek were taken from an old computer game created in 1984 called Elite, which can be described as a game of space trade. The aims in Elite were to slowly amass money, trade items, gain a better ship, pirating or escaping pirates, and improve one's rating to get the elusive 'elite' status.

In creating Cafétrek we transferred the game properties of Elite into a mobile game environment. Pirates and other spaceships encountered are real people. Planets, primarily used for trading and resource managing, are stationary computers stored inside public Cafés. To gather resources for trading, players must embark on a "café trek": purchasing resources cheap in one café and selling them expensive in another. Similar games [1] have recently been developed, but since they do not function without specific infrastructure they cannot be used in most public places.

In Cafétrek, each player uses a Wireless-LAN equipped PDA. The PDA contains the OpenTrek Launcher together with the custom Cafétrek Lobby Module, whose main interface much resembles that of the old Elite game. The interface contains a view from inside the player's space ship, and provides status information on the ship's current cargo hold and whether or not there are any planets or other players around. When physically approaching other planets or players (using the ad hoc wireless network) the player may choose to start one of the twelve Cafétrek game modules.

The Cafétrek game modules fall into two categories: Player-versus-player and Players-versus-planets. Player-versus-player services are trading modules and interactive games where players play against other players, each betting one or several items before the game begins. Player-versus-planet services can be either interactive games, or modules for resource collection (farming). Players can cooperatively play with each other performing recon missions on a planet, trade, or buy farms or mines to harvest planets resources

4.1 Project

To test how the OpenTrek platform would work for prototyping, the platform was given to 28 M.Sc. students divided in twelve groups. Each group varied in size from one to three people. The students had recently finished a three week course in C++, and were told to use this programming language to write their games. None of the students had previous experience in games development.

The students were introduced to the CaféTrek environment, and each group was assigned a game module based on the overall CaféTrek design concept. The time available to design, develop and test the application was five weeks, where one week was dedicated to design, three weeks for developing the game, and one final week to add the networking code. During development, the students had the opportunity to consult an experienced game developer for design advice.

4.2 Result

All student groups managed to demonstrate a working module after the five weeks had passed. The twelve modules ranged from trading modules and farms to turn based card games and advanced real time action games (as shown in Figure 2). Development of all games had begun as single-player applications, and all groups added the necessary interface to run their game from the OpenTrek Launcher in less than one day. All games were integrated with the CaféTrek middleware, allowing the player to "win" or trade items in one game, which could provide an extra bonus in another game.

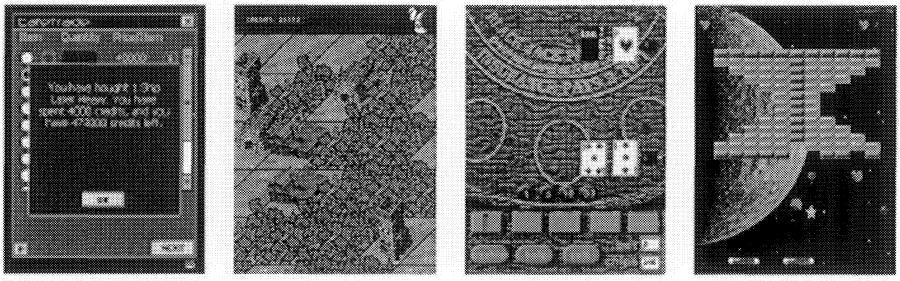

Figure 2. The game modules ranged from trading modules and farms to card games and advanced real time action games.

Network communication was implemented differently between the groups. Some of the techniques used were client/server, multicast and token-ring. Four of the twelve games used advanced graphics, including multiple back buffers and dynamically assembled, scrolling backgrounds. Two of the

games supported more than two players in each game session. We have listed some of the game parameters in Table 1.

Table 1. A summarization of game features.

Parameter	Number of games
Total number of games developed	12
Multithreaded games	9
Games supporting more than two players	2
Advanced graphics (smooth scrolling)	4

Some games comprised features exploring not only the novel properties of handheld devices, but some of the unique properties of wireless ad hoc networks as well. Two of these games were TrekFighter and TetTrek (as pictured in Figure 3).

TrekFighter is a game specifically built for ad hoc wireless networks. Trekfighter is based on the classic multiplayer game snake, often found in mobile phones. Snake is a game where each player controls the left/right turn of a small snake on the display. As time passes, the length of the snake increases, occupying valuable display space. The winner of the game is the player who is the last to run into a wall or the tail of the other persons snake. TrekFighter is played on a map that is larger than the display area. As the ship moves, the background scrolls smoothly. Other people can join the game at any moment, and are placed in a corner of the map.

TetTrek is based on the classic game Tetris. The game is played where the player controls falling bricks. If the bricks can form a solid horizontal line, that line is removed from the screen. The goal is to keep playing for as long time as possible. TetTrek is played using a rotated display, providing a wider game area than is normally used on handheld devices. The possibility of physically rotating the display is something normally not considered when targeting stationary computers.

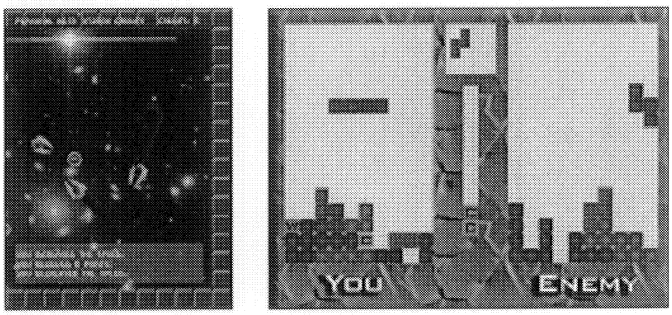

Figure 3. The advanced real time action games TrekFighter and TetTrek.

5. Conclusion

We have introduced the OpenTrek platform for Rapid Application Development and prototyping games for mobile devices such as PDAs. During a period of five weeks, 28 M.Sc. students with no prior experience of game programming delivered twelve different game modules, exploring the unique properties of wireless ad-hoc networks.

Future work involves making the platform available on additional devices. OpenTrek has already been ported to stationary computers, and will be transferred to other mobile game devices as well. Creating new games using OpenTrek enables the games automatically to run on all platforms supported by OpenTrek, due to its layered architecture. We will also use Cafétrek and some of the modules for an evaluation study, where we will focus on the use of ad-hoc network applications in everyday environments.

6. Acknowledgements

This research was performed at the Future Applications Lab, which is funded by the Swedish Research Institute for Information Technology (SITI AB). Special thanks go to all our students who have worked very hard in creating CaféTrek.

7. Download

The OpenTrek platform can be downloaded freely from the web using the URL http://www.opentrek.com. The graphics component of OpenTrek can be downloaded from the web using the URL http://www.gapidraw.com

References

[1] Bjork, S., J. Hansson, P. Ljungstrand (2001) "Pirates! - Using the Physical World as a Game Board," *INTERACT 2001*, IFIP TC.13 Conference on Human-Computer Interaction.

[2] Cybiko, http://www.cybiko.com

[3] Design and implementation of computer games: a capstone course for undergraduate computer science education, In *Proceedings of the thirty-first SIGCSE technical symposium on Computer science education*, 2000, Austin, Texas, United States.

[4] DirectX, http://www.microsoft.com/directx

[5] PocketKobo, http://pocketkobo.sourceforge.net

[6] RAD Game Tools, http://www.radgametools.com

[7] The Simple DirectMedia Layer, http://www.libsdl.org