

SmartBlaster – A Realtime Multiplayer Game for Mobile Ad-Hoc Networks

Heiko Roßnagel¹, Felix Zikesch², Thomas Kalbe²

¹Johann Wolfgang Goethe Universität Frankfurt
heiko.rossnagel@m-lehrstuhl.de

²Technische Universität Darmstadt
felix@cheruskia.de
kalbe@rbg.informatik.tu-darmstadt.de

Abstract: Multiplayer games have become very popular in the PC market. Almost none of the current games are shipped without some support for multiplayer gaming. At the same time mobile devices are becoming more powerful and popularity of games on these platforms increases. However, there are almost no games that support multiplayer gaming despite the multiple options of these devices to connect with each other and build mobile ad hoc networks. Reasons for this lack of multiplayer support are the high diversity of mobile devices as well as the different protocols and their properties that these devices support. With “SmartBlaster” we developed a multiplayer game for several different platforms that is using several different channels (Bluetooth, IrDa, 802.11 and other networks supporting TCP/IP) to communicate between them.

1. Introduction

Multiplayer games have become very successful in the desktop market. Almost every game that is distributed today has some kind of support for multiplayer gaming. The increase in popularity of distributed gaming has even lead to the creation of organized gaming communities. Due to the high availability of internet access people play against each other on game servers throughout the net or at organized LAN Parties.

At the same time mobile devices have made a huge increase in performance allowing the creation of more complex and hence more enjoyable games that run on these devices. With the addition of color displays mobile gaming is becoming more and more popular. And given the high market penetration [GS04] of mobile devices there seems to be a great market potential for mobile gaming.

Most devices shipped today do support various personal area networks like IrDa or Bluetooth to enable data exchange with and provide connectivity for other mobile devices like laptops and PDAs. However there are almost no games on these platforms that support realtime multiplayer gaming despite the multiple options of these devices to connect with each other and build mobile ad hoc networks.

With “SmartBlaster” we developed a multiplayer game for several different platforms that is using several different channels (Bluetooth, IrDa, 802.11 and other networks supporting TCP/IP) to communicate between them. With this contribution we will examine the suitability of wireless personal area networks as well as current mobile devices for realtime multiplayer gaming, which is one of the most demanding and most often used applications of IT. We also show how one can develop software for different platforms while providing multi-channel support.

In Chapter 2 we will state the initial goals of our project and give an overview of the application and the supported networks. In Chapter 3 we will take a look at the actual implementation and the problems that we experienced while implementing for limited mobile devices and multiple channels. Chapter 5 will give a short summary of our findings.

2. Project Goals

The goal of the project was to develop a realtime multiplayer game on several mobile devices that can compete with state of the art games on mobile game consoles (like Gameboy, etc.) and is just plain and simple fun to play with.

We wanted to show what modern mobile devices are capable of and how big the potential of mobile gaming is. We decided to re-implement DynaBlaster, a game that was very popular on the Amiga platform and has been ported to several different platforms like UNIX (X-Blast) and Windows (Atomic Bomberman) and has achieved a high level of success on each of these platforms. This game doesn’t need any sort of scrolling but still enjoys a very high popularity especially because of its multiplayer support. A screenshot of our implementation can be seen in figure 1.

In order to have a successful game we figured that it is essential to support multiple different platforms as well as the most common network protocols. On the other hand we needed devices that do have color displays and enough computing power to support the demands of our game. Therefore, we decided to implement this game on the Microsoft Windows CE based devices as well as on the Windows PC platforms. Reasons for our decision were the similarities between WindowsCE and the pc-based Windows platforms, which enabled us to easily develop the game for different platforms at the same time.

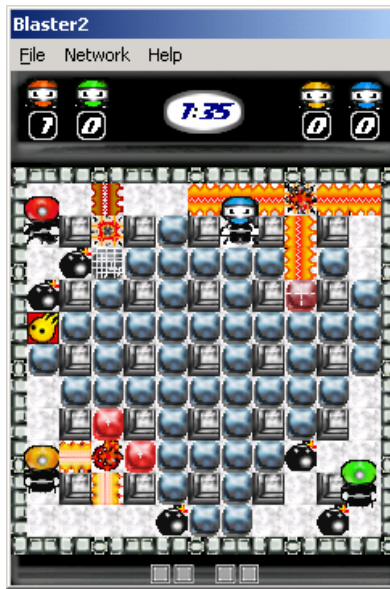


Figure 1: Screenshot of SmartBlaster

Our game runs on devices with PocketPC 2000 and 2002 (like the T-Mobile MDA or the HP IPaq) and on the SmartPhone (like the Orange SPV) as well as on any PC-based Windows that supports DirectX. Our vision was to have a realtime multiplayer game that runs on multiple mobile devices connected over several protocols. A screenshot of the planned communication infrastructure can be seen in Figure 2.

3. Implementation

3.1 Game Design

The game was completely written in eMbedded C++ using Microsofts eMbedded Visual Tools. We started with the development of the game area [La99] and the computer controlled players for the desktop version, since running and debugging the code on the mobile device is very time intensive. The code of the game area and the computer player [Fa01] is identical to the one used on PocketPC and SmartPhone. For the 2D animation of the desktop version DirectX was used.

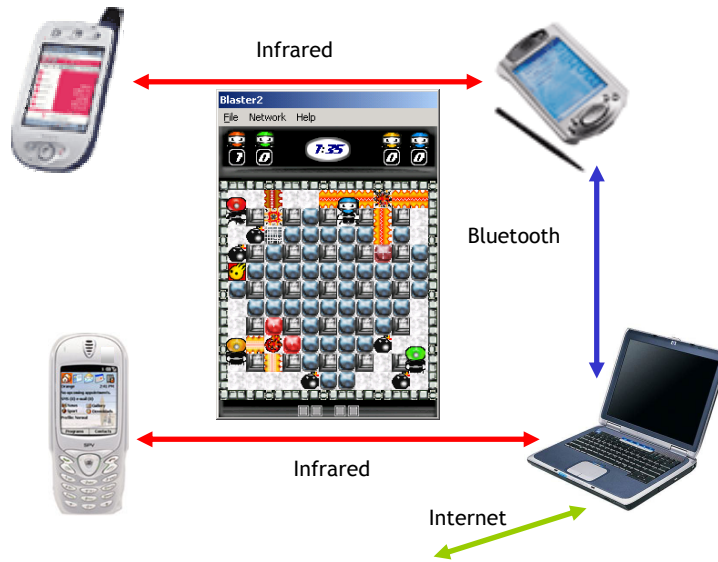


Figure 2: Communication Infrastructure

3.2 Graphics on Mobile Devices

In a second phase we implemented the visualization on PocketPC and SmartPhone. Since they do not support DirectX we used the GameAPI (GAPI) instead. Because the GAPI is not supported by the emulator of our developing environment we had to test and debug all related code directly on the mobile devices. Since the blitter of the GAPI [Ke01] was too slow for our realtime constraints we developed our own blitter to visualize the 2D animation on PocketPC and SmartPhone. Once the Blitter was written for the PocketPC, we just had to change the configuration to use it on the SmartPhone. The blitter is our implementation of the DirectX functionality for PocketPC and SmartPhone. Using our blitter the StrongArm processors of our test devices had no problems to display the 2D animations in realtime [KZ03].

3.3 Communications

Finally we implemented the communication infrastructure. Once again we started with the desktop PC. We used DirectPlay for TCP/IP based communication. DirectPlay is also supported by PocketPC but the mobile devices we had did not support 802.11. In addition there is no way to establish a TCP/IP connection through Bluetooth or infrared between two mobile devices or a mobile device and a desktop.¹

¹ WindowsCE does support TCP/IP connections using Bluetooth. However, we could not find a mobile device that actually used the MS Bluetooth stack.

Therefore, we implemented the Bluetooth connectivity through a serial Bluetooth connection. This trick works pretty fine to run the game on the desktop pc, but the implementation of the serial Bluetooth protocol on a PocketPC was too slow to play the game. Furthermore, using a Bluetooth serial connection to play the game requires a lot of configuration of the devices before the game can be started [KZ03].

We implemented the infrared communication using windows sockets. The sockets were similar to program on desktop and Pocket PCs, but we used different modes of the sockets on the different platforms. Because our devices only supported Serial InfraRed (SIR) we experienced serious speed problems [Ir03]. We finally managed to get a playable version using infrared by skipping abstraction layers. However, sometimes the game fun is decreased by to large delay times of the network packets, depending on the quality of the infrared connection [KZ03].

4. Conclusion

We successfully implemented a realtime multiplayer game on three different platforms. For the core of the game the same code is used on all platforms. We developed a blitter for PocketPC and SmartPhone that enabled us to display the 2D animation fast enough. The game can be played in a single user mode with up to 3 computer controlled players or as a multiplayer game with communications via infrared, Bluetooth or 802.11. However, we could only establish fast enough connectivity between mobile devices via infrared. The complete source code, binaries and documentation are available online [Sm04].

References

- [Fa01] Joseph Farrell, "Game Programming Genesis PartIX: Adding Characters" gamedev.net/reference/articles/article1338.asp
- [GS04] GSM Association: GSM Statistics www.gsmworld.com/news/statistics/index.shtml
- [Ir03] IrDA Miniport Driver Packet Coding Schemes, 2003
http://www.osr.com/ddk/network/210irda_6up3.htm
- [KZ03] Thomas Kalbe, Felix Zikesch: "Developing an Multiplayer Action Game on Mobile Devices with Network Support", www.smartblaster.net/downloads/smartblaster.zip
- [Ke01] John Kennedy; "Time for GAPI", 2001,
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnroad/html/road11142001.asp>
- [La99] Andre LaMothe; "Tricks of the Windows Game Programming Gurus", SAMS, 1999
- [Sm04] The SmartBlaster Homepage, www.smartblaster.net