FUN IN NUMBERS (FINN): A PLATFORM FOR DEVELOPING AND PLAYING MULTI-PLAYER PERVASIVE GAMES

Orestis Akribopoulos
Computer Engineering & Informatics Department
University of Patras, Greece
akribopo@ceid.upatras.gr

Marios Logaras
Computer Engineering & Informatics Department
University of Patras, Greece
logaras@ceid.upatras.gr

Nikos Vasilakis
Computer Engineering & Informatics Department
University of Patras, Greece
basilakn@ceid.upatras.gr

Panagiotis Kokkinos
Research Academic Computer Technology Institute
and University of Patras, Greece
kokkinop@ceid.upatras.gr

Georgios Mylonas
Research Academic Computer Technology Institute
and University of Patras, Greece
mylonasg@cti.gr

Ioannis Chatzigiannakis
Research Academic Computer Technology Institute
and University of Patras, Greece
ichatz@cti.gr

Abstract Fun in Numbers (FinN) is a platform for developing and playing mobile, locative and collaborative distributed games using wireless sensors. Using FinN, a very large and diverse set of games can be enhanced, by maximizing the on-game experience and collecting statistics for off-line, web-based view. At the same time the essence of such games remains the same: *fun in large numbers, in every place and at any time*. FinN is implemented using a combination of JAVA Standard and Mobile editions, while on the hardware part we use wireless sensor devices, called Sun SPOTs. In the future, mobile phones that have some kind of sensors embedded, or other custom devices can be used for the same purpose. We report a number of examples of games created with FinN and briefly present the architecture of our platform.

1. Introduction

Games have been a major part of the computer industry for the last decades, and are generally recognized as a means of pushing the technological boundaries, both in software and in hardware. Recent advancements in mobile phones technology have produced new products that integrate various kinds of sensors into the handsets. However, few of the applications proposed are related to mobile, interactive, multi-player games where users carry devices with sensing capabilities. Although there have been some previous related attempts, these works are rather limited in number and scope. It is our belief that there is great potential hidden in the combination of sensors and mobile devices for producing entertainment applications.

FinN's architecture is based on a hierarchy of layers for *scalability* and *heterogeneity*. A number of services that allow localization of wireless devices in indoor environments (e.g., neighbor discovery), performing sensing tasks while on the move, coordination of actions (e.g., leader election) and delay-tolerant communication are currently implemented. Moreover, statistics are gathered, processed and stored, for off-line view through a web-based user interface. The use of our platform is rather intuitive and does not distract the developer from his main task, the design, development and testing of exciting game applications.

2. FinN Games

The key characteristic of the games developed with and played through FinN, is that players engage in interactions with each other and their surrounding environment by moving and gesturing, as a means to perform game related actions. The player, as a physical entity, is the center of the game and there is lesser need for providing continuous visual feedback than most of the video games played today. Furthermore, the input the players give to the game is kept to minimum (e.g., by means of performing a specific gesture) or indirect (e.g., based on the location of the player). Similarly, the feedback of the game to the player is again minimum (e.g., win or lose) and some times sporadic (e.g., indicating that the player reached a specific location or is close to an opponent). FinN games can be played in the neighborhood, in the city, in the beach, in the camp, in every place and at every time. After the game finishes, the players can upload the data collected by their devices for processing and producing the statistics.

Example games implemented through our platform are *Moving Monk*, *Hot Potato* and *Anonymous*:

Moving Monk: Each player in the game is a "monk" moving continuously amongst a predefined set of "temples". The goal for each player is to visit temples as fast as possible, perform specific "prayers" in each location. A temple is defined by the coverage range of a base station and the prayers performed are specific gestures. To help the monk find the temples, clues can be given regarding the exact location of a temple.

Hot Potato: Each device has the potential of generating a Hot Potato, which "explodes" after a specific amount of time, eliminating the last player who was carrying the potato. Each player can randomly pass the potato to one of the neighboring players. The goal of the game is for each player to pass the potato to the rest of the players, thus eliminating the danger of the potato exploding on his device. The winner is the player standing alive.

Anonymous: A powerful player is the "master", while the rest of the players are the "apprentices", unaware of the master's identity. On the one hand, the apprentices are in search of the master's identity and aiming in combining their powers to eliminate her. On the other hand, the master must hide her role, locate the apprentices and eliminate each one of them by performing specific gestures. Last player(s) standing is the winner.

We believe that these games reveal, to a certain extent, the variety and the joyfulness of the games produced by FinN. The *Moving Monk* game features location-aware services, while *Anonymous* and the rest of the games demonstrate player interaction, offer context-aware services and support delay tolerant networking.

3. The FinN Platform

3.1 Software

The architecture of our system is based on a hierarchy of layers where each layer is comprised of one or more peers. Each peer may be a traditional networked processor or a wireless sensor device.

Each player carries a wireless sensor device that executes a software component called the *player peer*. Player peers communicate with a collection of stations that form a wireless backbone infrastructure and are used to

monitor the evolution of the game and interact with nearby players. When another peer is discovered, the user is prompted for further action. The user can use the sensors and buttons to explicitly trigger actions for interaction. Each action will cause the exchange of data and may involve neighboring peers. Player peers provide services that allow them to interact even when they are disconnected from the network infrastructure for extended periods of time.

The wireless backbone is established by *station peers*, each controlling a specific physical area. During the initialization of the system one station peer becomes the *game engine*, responsible for the coordination of the infrastructure. Stations communicate with the users' devices either through local ad-hoc networks or via personal area non-IP networks, providing location-aware and context-aware services to the player peers. The game engine communicates with the *world peer*, a central peer that is accessible via the Web and offers high level services. It maintains the player related statistics and keeps track of game history. The world peer is updated asynchronously to avoid computational and communication overhead.

Our system allows the implementation of games that require near real-time response to specific aspects of the game (e.g., role playing and sports games); it offers location-oriented services for games that need players to visit specific places (e.g., puzzle, strategy and treasure-hunt games); it enables the development of services that can augment the physical reality (e.g., via Google Earth); it also allows games to be played by devices that do not offer Internet connectivity. We implemented our platform using the Java programming language and the available technologies, providing in this way homogeneity and ease regarding the maintenance and extension of the code.

3.2 Hardware

As our hardware platform we use Sun's SPOT platform, which is a platform targeted mainly to experimentation and prototyping. The SPOT device is built upon the IEEE 802.15.4 standard. It is a small, battery-operated, device running the Squawk Java Virtual Machine (VM), which acts as both operating system and software application platform, thus allowing programming of the devices in the Java Micro Edition (J2ME) platform. A Sun Spot includes a very simple interface (two buttons and a number of LEDs) and a variety of sensors (e.g., accelerometer, thermistor). We plan to port FinN to mobile phones, in order to take advantage of the huge existing user base. We believe that such a port will be quite straightforward, since J2ME is also available in all current-generation mobile phones. We also consider developing a specific and easily reconfigurable FinN game device, which can be built and delivered in numbers to the FinN players, for games in any environment and under any condition.

Acknowledgments

This work has been partially supported by the European Union under contract numbers IST-2005-15964 (AEOLUS) and ICT-2008-215270 (FRONTS).