Implementing Multiplayer Pervasive Installations based on Mobile Sensing Devices: Field Experience and User Evaluation from a Public Showcase

Ioannis Chatzigiannakis, Georgios Mylonas, Panagiotis Kokkinos, Orestis Akribopoulos, Marios Logaras, Irene Mavrommati

Research Academic Computer Technology Institute (RACTI), Computer Engineering and Informatics Department, University of Patras, Greece, School of Applied Art, Hellenic Open University, Greece
{ichatz, mylonasg, kokkinop, akribopo, logaras, mavrommati}@cti.gr

Abstract

In this work we discuss Fun in Numbers, a software platform for implementing multiplayer games and interactive installations, that are based on the use of ad hoc mobile sensing devices. We utilize a detailed log of a three-day long public showcase as a basis to discuss the implementation issues related to a set of games and installations, which are examples of this unique category of applications, utilizing a blend of technologies. We discuss their fundamental concepts and features, also arguing that they have many aspects and potential uses. The architecture of the platform and implementation details are highlighted in this work, along with detailed descriptions of the protocols used. Our experiments shed light on a number of key issues, such as network scaling and real-time performance, and we provide experiments regarding cross-layer software issues. We additionally provide data showing that such games and installations can be efficiently supported by our platform, with as many as 50 concurrent players in the same physical space. These results are backed up by a user evaluation study from a large sample of 136 visitors, which shows that such applications can be seriously fun.

Keywords: mobile, middleware, gaming, sensors, pervasive, application, location-aware, context-aware, deployment, large-scale

1. Introduction - Motivation

We are currently witnessing an unprecedented level of activity in many fields related to the cross section of the digital and physical domain. We are gradually discovering what we can do (i.e., in a conceptual sense) and how to implement it (software and hardware ecosystems), getting closer to the pervasive computing future envisioned in the previous decades. In the meantime, our idea of how and where this vision is going to be applied has moved well beyond the classic “smart fridge” paradigm. In that respect, gaming and entertainment are two fields that we believe are ideal targets for applying this vision; they attract tremendous interest and have progressed along with all of the major advances in computing the past few decades.

In this work, we discuss our vision for creating unique multiplayer games and interactive installations, by enhancing them with concepts that have stemmed from the research community the last few years. We have witnessed the astounding activity and results produced by the Wireless Sensor Network (WSN) research community in the last decade. Such activity has focused on providing solutions on more general distributed computing problems or specialized application issues, producing a wealth of algorithmic solutions and protocols, delivering practical results. The breadth of the issues studied the past few years is astounding, mainly due to the paradigm shift that took place gradually, from small networked “islands” to the Internet of Things. Thanks to such software systems, we are seeing projects with sensor networks monitoring the seismic activity in a remote volcano [1] inside a jungle and robotic actors [2] sensing in real-time the reactions of the audience.

Furthermore, we have the so-called “Physical Computing” community, mostly related to new media artists and interaction designers, with the Arduino ecosystem as its pinnacle. Development environments simple and powerful enough for non-computer scientists, have allowed the creation of a multitude of interactive installation based on cheap embedded microcontrollers, sensing components and actuators. However, sophistication and efficiency are often not regarded as primary issues, and implementations in this category, more often than not, are considered as “prototypes”.

Moreover, at the same time frame, Pervasive Gaming has been a popular research subject (and a buzzword at the same time). In general, pervasive gaming refers to gaming applications conducted in the open space, combined with some degree of context-awareness in a scripted or unscripted fashion. The proliferation, initially, of PDAs and recently of smartphones (e.g., Ap-
ple’s iPhone), which feature networking and sensing capabilities, has provided a platform to implement such applications. Concepts used in pervasive gaming are also being adapted by more traditional mobile gaming platforms, such as proximity sensing and social networking.

We argue here that the pairing of efficient software systems and the above concepts can lead to astonishing results, especially in the unique context presented here. Up till now, we have not really seen the broad adaptation of the research community’s results by the creative community, even though both are using similar tools. Coming from a WSN research background, we have chosen inexpensive WSN hardware as a basis for implementing such concepts. In this work, we attempt to shed light into some of the main practical issues that arise when combining pervasive gaming and WSN, such as network scaling and number of users/players. We are overall interested in games and installations based on three features, movement, presence, and other sensing inputs, all provided by the use of sensor networking techniques. We focus here on the use of movement and presence to demonstrate our point. We build upon our previous work, utilizing the Fun in Numbers framework [3], a framework for creating mobile locative multiplayer games.

Such games are fine examples of distributed systems, the design of which must deal with a number of challenges, depending on the type implemented: distributed and reliable operation, scalability and efficient network communication, real-time operation constraints, energy efficiency, time synchronization, etc. Thus, it should be clear that this category of applications, apart from suitable hardware, needs efficient, flexible and reliable software systems to support their implementation. However, since this is a relatively new field, there are still some key issues, e.g., network scaling, real-time performance, maximum number of users, etc., that have not been thoroughly studied even on an experimental basis.

Fun in Numbers (FinN) is, in general, a platform for creating, deploying and administering multi-player games with pervasive and locative features that employ wireless sensor network nodes as the gaming devices. We discuss here its architecture and main features, along with a set of key implementation issues. We utilize a multi-tier approach, that can be customized for several different types of applications, based on tried models and technologies. We describe in detail the protocols upon which the implementation of the system services is based. Using FinN, we have implemented 4 installations (see Section 6) that fall into the discussed application category. We utilize Sun’s SPOT [4] nodes as our prototype implementation hardware platform,
which provide the basic functionalities of wireless sensor network nodes. A number of services are currently implemented, allowing location awareness of wireless devices in indoor environments, performing sensing tasks while on the move, coordinating basic distributed operations. We provide a series of experimental results, essentially the result of a three-day-long exhibition at a local theater, organized in order to validate the appeal of the installations and the overall suitability of the implementation. This is one of very few works presenting large-scale experimental results in realistic settings, and specifically in front of an unbiased audience.

Our results indicate a very positive response from the people that visited this exhibition and that the implementation platform is adequate in most of the aspects considered. We also opted for providing a detailed log of our field experiences from this exhibition, which emphasizes the difficulties we faced, along with a discussion on practical guidelines for deploying such large-scale systems - i.e., with up to 50 concurrent users.

Regarding the organization of this work, in Section 2 we discuss related work and differences from the approach presented here. In Section 3 our perspective on multiplayer games and interactive installations is presented. We provide sample scenarios, along with an overview of the architecture of our software platform. Important aspects, such as technologies used and services provided by FinN are discussed in Section 4. The evaluation of a set of cross-layer issues is discussed in Section 5. A description of the deployed installations follows in Section 6. In the remaining sections, we provide a day-to-day description of a public showcase of our platform, supported by experimental data (Section 7), while on Section 8 we provide results from a user evaluation study, that shows a very positive reaction towards our implementation. Lastly, we provide a discussion on lessons learned from such an exhibition, and conclude this work.

2. Related Work

There has been a wealth of activity in the pervasive gaming and interactive installations the last decade or so. Pervasive gaming can mean many different things; we prefer to define it generally as games played in the physical space, indoors or outdoors, using mobile handheld devices, context-awareness, and in certain cases some degree of infrastructure and scripting. Using “the world as a game board” is also a pretty good overall description; [5] provides a nice overview of recent trends in pervasive gaming, which can
range from geocaching games, to playing tag-and-chase [6] games in an urban scale. However, we believe pervasive gaming has not yet reached its full potential. We too use some of these familiar concepts in our work, but we emphasize real-time interaction and context-awareness, along with intense physical activity and large-scale deployments, features that have not been explored previously to such a degree.

More specifically, large-scale interactive installations and gaming and the practical aspects of such concepts have not been discussed extensively in the bibliography. There are a number of works, like [7], that discuss specifically about WSN deployment issues, but focus solely on stationary cases and standard WSN applications; they are also described only from the perspective of a research scientist as an end-user. More innovative applications based on WSN are discussed in [8, 9, 10]. Other works study pervasive games on a potentially urban scale, following a more centralized approach, where interactions between players are less frequent. Most of the works from the new media and interactive installations communities focus on small-scale deployments. Furthermore, there are few works discussing large deployments in front of the general public, e.g., [11]. However, inter-player interaction is on a different context, since they do not focus on rapidly changing network topologies, but usually discuss less dynamic scenarios with smaller number of players. There is a common pattern in such works; efficiency and scaling of the respective implementations are usually not backed up by real-world data. We discuss here the deployment and operation of 50 devices in parallel, at the same physical location and in real-life conditions (a public exhibition).

Pervasive gaming and interactive installations in a cultural setting has been a popular subject in recent years. [12] gives an overview of the types of applications developed for mobile devices in a museum setting. An example of a pervasive game example in urban scale with a touristic context is [13]. The work reported here can be used to extend such works, providing additional interactivity and context-awareness, with greater accuracy and efficiency as backed by the deployment data. An interesting concept of applying games is described in [14], where multiplayer games are utilized to stimulate human interest and engagement in a work environment; complementary to this approach, our work could be used to implement such games in larger, open physical spaces.

The last few years we are also seeing mobile gaming devices of all sorts gaining “pervasive” features; e.g., current smartphones (e.g., Apple’s iPhone) and next-generation portable gaming consoles (e.g., the Nintendo 3DS) utilize
concepts such as proximity detection to spontaneously ignite “tournaments”, or use social networking sites to add context to gaming experience. Adding to this dimension, we emphasize concepts from more “traditional” games, and take them to another level by allowing physical game interactions through computing devices. Conceptually similar to this work is Ubisoft’s “Battle Tag” [15], a multiplayer game made commercially available at the end of 2010; however, this is a very confined game, context-wise, and it’s current implementation allows a comparatively limited number of players (i.e., 8), whereas we currently support as much as 50 players in the same space.

Moreover, in this work, we utilize IEEE 802.15.4-compatible devices, a wireless networking solution that, in our opinion, suits better the application domain we are discussing here and offers more possibilities. To our knowledge, this is one of the first works to utilize such technologies in the pervasive gaming research area. Although smartphones and portable gaming consoles carry a number of integrated sensors and have built-in wireless capabilities (WiFi, Bluetooth), they are not as versatile and easy to scale, in order to develop ad hoc networking techniques. E.g., Bluetooth’s design does not allow quick and efficient detection of network neighborhoods in under 100 milliseconds time (a typical limit for human perception).

Regarding the specific interactive installations presented here, [16] describes a concept for drawing in 3D space, conceptually close to the “Chromatize Images” installation (Section 6), but we focus on multiplayer features and use of everyday objects as interface methods, rather than on accuracy and gesture recognition. Finally, our work comes in contrast to approaches like Nintendo’s Wii [17] or Microsoft’s Kinect [18]. The use of cameras and other optic technologies translate into constraints in the deployment field and the number of players supported. This work does not exclude fields with obstacles and other means of blocking line-of-sight and can be utilized outdoors. Additionally, it can support up to 50 players, whereas such technologies are typically limited to just 4.

This paper extends our own previous work [3, 19], by providing additional implementations of interactive installations, evaluation of software implementation issues, along with large-scale deployment data and user evaluation results in real-life conditions. The protocols used in FinN were briefly referenced in these works; here, they are described for the first time in much more detail, providing also insight to their actual implementation.
3. Fun in Numbers overview - Concepts and Architecture

We first discuss about the type of applications we are interested in: FinN targets scenarios where a large number of users participate using wireless handheld devices with sensing capabilities. These games and installations can take place in the same or different place and time. Overall, the operation of the games is either supported by a backbone infrastructure that provides a number of services (e.g., localization, context awareness) and through which a central entity coordinates and records the games’ progress, or all operation is done in an ad hoc fashion.

The users’ devices use wireless ad hoc networking protocols to establish network connections/neighborhoods, communicating with other user devices or infrastructure ones. Embedded sensors are utilized for providing input from the physical world, translated according to the rules of each game/installation. E.g., high temperatures could be translated as “prohibited” areas. Accelerometers are used for recognizing gestures made by users; gestures are mapped to player interactions, e.g., “pass on the token”. Proximity detection, currently made possible through the use of wireless signal characteristics, is also used as a means of input, as well as movement overall, i.e., not just gestures. Such input can be used to monitor the progress inside a game and also enforce the rules defined for it. By utilizing such input we essentially create games in a more “traditional” fashion, but with added interaction features.

We provide here a categorization of the applications we envision, in terms of interaction between users, existence of an infrastructure and time-wise operation:

*Spontaneous*: activities are set up instantly between players, e.g., imagine when a having a break from work at the office, playing a tag-and-run game with 15 players for a couple of minutes. No specific infrastructure is needed, with communication and sensing inputs be provided only by the mobile devices the players are carrying themselves. For example, game results could be uploaded and stored centrally after the game is over, in order to gather statistics or gain/gather game score points. No specific limits are imposed whatsoever regarding time and place.

*Storyline-based*: a degree of infrastructure is used in order to provide presence and other sensing input combined with a mixture of some sort of scenario and timely events. A central authority coordinates the game as mandated
by the given script. An example of such a game, could be an entertainment/educational installation inside a museum, where players are given an initial scenario and must discover elements to move forward in the game by visiting places, etc. The infrastructure intervenes both in the spatial and temporal domains to define the possible game outcomes, but all is performed within specifically defined limits.

**Community-based:** this is basically the rehash of traditional games, like “hide and seek”, augmented with the aid of computing devices with sensing abilities, which can help in game operation, rules arbitration, statistics bookkeeping, etc. They can be both infrastructure-less or infrastructure-based, but the distinct element here is the concept of the gaming community. This means, that there is a community playing games of various sorts, with the playing activity spanning across a long time period and with large gaming interaction among the members of the community.

**FinN**’s architecture is based on a hierarchy of layers for scalability and easy customization to different scenarios (heterogeneity). A number of services are currently implemented, allowing location awareness of wireless devices in indoor environments, performing sensing tasks while on the move, coordinating basic distributed operations and offering delay-tolerant communication. Players carry one or more handheld devices with wireless communication and sensing capabilities. A backbone infrastructure may also be available,
possibly forming a wireless mesh network. Such an infrastructure consists of Stations, which are able to support the player devices with a number of services (e.g., localization, context awareness) and may also operate actuators that add certain gaming elements. Each game instance is assigned to and coordinated by a specific Engine, which is the local authority for each physical game site. The World is the topmost element that manages multiple physical game sites and allows interaction with social networking sites, e.g., Twitter. Figure 2 illustrates these elements.

These heterogeneous elements (in terms of role, communication, computation and energy capabilities) form a loosely-coupled, highly modular and customizable hierarchy. This allows FinN to support both ad-hoc and infrastructure-based games (or interactive installations). In the first case, a very limited backbone infrastructure is available, thus almost all communication and computation must be performed by the handheld devices in a fully distributed manner. In the second case, we wish to take advantage of the backbone by following a “2-tier principle” [20], thus moving communication and computation to the fixed part of the network. Our approach allows services (e.g., discovery, proximity detection, gesture recognition) to be seamlessly reallocated from the player devices (where computation and communication is expensive) to the backbone infrastructure (where resources are practically unlimited) with minimum programming effort.

Guardian layer: This layer essentially involves the devices used by players during the games. The Guardian is the software component running in each player’s wireless sensing device and uses the device’s capabilities in terms of user interface, communication, etc. Protocols for the discovery and the
communication with the “backbone” infrastructure and other Guardians are provided (see next section). When another Guardian peer is discovered, the player may be prompted for further action, by using the sensors and the buttons of her device. For monitoring the evolution of the game, each game-related action is represented by an Event. Guardian peers also implement services that allow them to interact even when they are disconnected from the “backbone” infrastructure for extended periods of time. In particular, when an Event occurs, the Guardian stores it to the device memory and when communication with the infrastructure is possible, then all collected Events are forwarded (delay-tolerant communication service). Also, Guardians provide a subsystem, which processes the samples of the accelerometer and recognizes gestures that correspond to game-related actions.

**Game Station layer:** This layer implements the “backbone” infrastructure, which is important, though not mandatory for all of the games developed. It provides localization and context-awareness services and it is through this infrastructure that the data of the players are transferred to and from the higher layers of the architecture, for coordination and storing purposes. This wireless backbone is established by Station peers, with each Station controlling a specific physical area, responsible for the coordination of the infrastructure and of the game itself. The Stations communicate with the users’ devices either through local ad hoc networks or via personal area non-IP networks and act as gateways, essentially allowing communication between the players’ devices and the Game Engine. Multiple Stations can be attached to an Engine in order to extend area coverage or add points-of-interest. During the initialization of a game, Stations communicate with the Engine and retrieve data such as the set of players, which are registered for this game instance, player devices and places-of-interest. Stations are also responsible for the Guardians’ initialization and for forwarding all data generated during the course of a game to the Engine.

**Game Engine layer:** Each game instance is assigned to and coordinated by a specific Game Engine, i.e., it is the local “authority” for each game site. The Engine retrieves data from higher layers and stores them locally, for the duration of a specific game. In order to avoid computational and communication overhead, data between higher layers and the Engines can be synchronized periodically. Thus, the processing and storage of generated events during the game is done locally. The Engine is also a control mechanism that provides game-specific services and implements various game scenarios.
Communication between the Engine and the Stations is carried out through wired and/or wireless IP-based networks. Finally, the Engine features an embedded Web container for providing additional game specific information to players directly.

**World layer:** The *World* layer is the topmost layer of the hierarchy, enabling the management of multiple *FinN* games, physical game sites and users. In this layer lies the Hyper Engine, potentially interconnecting discrete games and installations, providing a unified system. E.g., in an exhibition with multiple installations visitors can interact with all installations using a single device. The system “remembers” the actions of a particular visitor from another installation or a previous visit to a particular installation. Thus, it supervises all on-going games and monitors the events generated. It also enables easy logging of various statistics during parallel game sessions (such as in Sec. 8). This layer also includes the *World Portal*, which is the central point of system management, providing interaction with all the different game instances operating in the real world. It is also the central storing point for all game-related data, such as player-related statistics and game history. Furthermore, it allows personalization capabilities and possible interaction with external social networking sites.

4. **Implementation details - Software Services**

We implemented *FinN* using Sun’s SPOT platform [4] for the players’ devices. It is a small, battery-operated device running the Squawk Java Virtual Machine, which acts as both an operating system and a software application platform, allowing programming of the devices in the Java Micro Edition (J2ME) platform. It uses an 180MHz ARM920T-based microcontroller with 512KB of RAM and 4MB Flash. An IEEE 802.15.4 compliant CC2420 transceiver is used for communication. They also provide a basic user interface (2 buttons and 8 LEDs) and a number of sensors (accelerometer, thermistor, light). We decided to use this particular platform due to the available computational resources; other WSN platforms usually offer 10KB of RAM and a processor at 8MHz. The 8 LEDs and 2 buttons improve the interfacing methods of the device.

A challenging aspect of this implementation was to enable transfer of data through the various heterogeneous layers. The complication arises from the fact that player devices are implemented using Java Micro Edition, while the
other layers using Standard Edition (see Fig. 3). Thus the standard Java Serialization API is not available across different editions. We decided to extend the Java Persistence Storage API to ensure that services and game events are communicated across layers in a seamless and efficient way. Communication between the Engine and World layers is implemented via Hibernate [21]; Remote Method Invocation (RMI) was used to allow communication between the Engine and Station layers. We implemented object serialization in combination with the Radiograms API within the Squawk VM to enable the seamless exchange of data objects between Stations and Guardians. As a result, this approach allows the programmer to use the same objects throughout all layers of the system without the need to keep separate versions for each Java edition. Clearly, this reduces the development and maintenance efforts. In Section 5 we evaluate the overhead to exchange objects across all layers of the heterogeneous hierarchy.

![Figure 3: Technologies and layers used in the system](image)

The key services provided by the platform, such as neighbor discovery, localization and information exchange are discussed in the following paragraphs. These services are fundamental in implementing our envisioned applications, while their interconnection is depicted in Fig. 4.

**Neighbor Discovery:** *Echo Protocol* offers local connectivity awareness, while it is designed to run on resource-limited devices. It is also robust, able to adjust quickly to frequent and significant topology changes and capable of distinguishing the different roles of the discovered neighboring nodes (i.e., player, backbone/mobile station). In addition, it allows customization of
the propagated messages. In particular, FinN player devices and backbone stations are characterized by a list of attribute-value pairs that describe the status of the game.

These descriptions are constantly broadcasted by the devices. In this respect, the Echo Protocol is also used as the building block of other protocols and services, such as localization and leader election. In particular, in the leader election service the ID of the device/player broadcasted, is used as key parameter for selecting the leader among a set of players. Echo Protocol is able to recognize whether the communication with the surrounding devices is bi-directional or not. In general, players interact in pairs or in groups by executing (simultaneously or not) actions. The coordination of player actions almost always requires symmetric communication. The majority of theoretical works consider wireless links to be symmetric, however in practice most of the times this is not true for a multitude of reasons. This fact significantly complicates game design; a unidirectional link cannot provide acknowledgments of receipt of messages. To overcome this problem, the Echo protocol attaches to each broadcasted packet the list of detected neighbors; i.e., the devices it has received a beacon from in the last $3 \times \text{beacon\_interval}$ period. This attribute-value pair may look like: $\text{neighbors} = [11, 81, 5, 43]$.

Echo Protocol relies on the devices’ IEEE 802.15.4 radio. In contrast to Bluetooth, data is exchanged without establishing a connection and concur-
rent transmission of messages to multiple devices is supported. Thus the interval between consecutive broadcasts can be set to less than 100ms time period while Bluetooth devices may take up to 10.24 sec [22], significantly improving discovery among devices. Clearly, by decreasing the value of the interval of broadcasts, a near real-time response to the topology changes can be achieved, at the expense of increased network load and energy consumption due to the beacon messages exchanged. Moreover, Bluetooth limitations such as concurrent connections to multiple nodes are handled much better by 802.15.4, which allows a completely ad hoc network topology. In practice, we have seen Sun SPOTs handling up to 20 concurrent connections, in the same 802.15.4 wireless channel, to neighboring nodes, while having sufficient system operation. The flexibility of selecting different 802.15.4 channels to use in Sun SPOTs, allows for a much greater total number of nodes as described in the experimental results section.

Furthermore, although IEEE 802.15.4 is not currently nearly as popular as Bluetooth or WiFi in mainstream mobile devices, e.g., smartphones, its use in combination with platforms such as Sun SPOTs possesses certain advantages. Apart from allowing access to the lower network levels, typical 802.15.4 radios offer the ability to modify the nodes’ communication range. Typical values range from a few centimeters (minimum range) to 30 meters (maximum). Setting a communication range of 2 meters allows the use of the radio transceiver to be used as a sort of proximity sensor, in cases of course where accurate distance estimation does not play a significant role. Such a typical case is in our ad hoc games; players tend to move rapidly while e.g., playing a tag game, while game interaction can take place when players are in small physical proximity of each other. Moreover, there is also the issue of energy consumption. Almost all of the current hardware sensor nodes use IEEE 802.15.4, for a good reason. It can have similar or lesser power requirements than Bluetooth, while having a greater communication range. Also, it is much better in handling power consumption than IEEE 802.11 radios, due to lower bandwidth, transmission range, etc.

Since IEEE 802.154 targets specific market segments until now, its adoption outside these environments and the research/academic sector is not wide. Therefore, there has to be a bridge/gateway device used in order to communicate with the 802.15.4 devices (since there are few integrated solutions available), which adds certain overhead and complexity. However, the provided Sun SPOT SDK alleviates many of these issues, lifting this burden from developers. Also, since it targets low power consumption, its band-
width is lower even than Bluetooth, and in cases where higher bandwidth is
needed (e.g., input from a camera), that would potentially pose a problem.

**Localization:** In certain games, user devices need to sense their relative
location to each other and to specific landmarks. A simple approach for
proximity detection is to use the attribute-value pairs of the *Echo protocol.*
By properly adjusting the *transmission power* and assuming that the players
hold the devices in a particular way (e.g., strapped on their knee), the dis-
tance to another device can be estimated by the player device. For example,
in a hidden treasure game devices can detect if the players have discovered
them. This practically translates close to proximity detection. We typically
set the communication range to be around 2 meters in cases where we utilize
such a kind of proximity detection (e.g., spontaneous ad hoc games [3]).

In other games/installations (e.g., “Magnetize Words”), more accurate
location information is required on the relative position of the players. The
incorporation of location awareness in ad hoc mobile sensor networks is a
well-studied subject. A radio-based approach was chosen again here, using
a fixed infrastructure consisting of at least three backbone stations (the so-
called anchor nodes) and the periodic broadcasts of the *Echo protocol.* This
approach allows the implementation of a wide range of centralized localiza-
tion algorithms (e.g., [23]). For each beacon received by these nodes, the
Received Signal Strength Indicator (RSSI) and the Link Quality Indicator
(LQI) are extracted to measure the power of the signal. These indicators are
continuously forwarded to the *Engine Layer* that computes the position of
the device based on a set of previously-collected values (training values). We
implemented the following algorithms: a) *Simple Localization*, the position is
that of the station with the bigger RSSI, b) *Average Localization*, using the
euclidean distance between received RSSI/LQI and previously trained val-
ues, c) *k-NN Localization*, based on k-nearest neighbors algorithm in order
to map the position of a node and d) *Hybrid Localization*, a combination of
the above algorithms, which was the one used in practice.

More specifically, our decision to use the hybrid approach was a trade-off
between accuracy and easiness of setup. Given the installation we wanted
to use the localization algorithm with (Section 6, “Magnetize Words”), there
were certain rather unique localization requirements. Essentially, what we
wanted to do was localization mostly on a single axis, i.e., in parallel to a
wall where certain images were projected, and not in a 2-axis plane (e.g., x
and y). Thus, we simplified the problem of localization to provide a quicker
response to the players’ movement along a projection wall. This led also to a simplified setup approach: we divide the area that is to be localized in discrete sub-areas. We then conduct an initial calibration phase, where we acquire readings from a single node placed in each one of the sub-areas. After this initial phase, and during operation of the installation, the hybrid algorithm uses two steps: a) it uses the Simple Localization algorithm to detect a general sub-area, b) it then uses one of the other two algorithms to compute a sub-area within the general sub-area computed in the previous step. This was due to the fact that the simple algorithm was better at detecting a general sub-area than the other two, limiting possible localization results and thus potential localization instabilities. In practice, we saw that the Average algorithm performed better in our deployment area.

**Information Exchange:** Given our choice of discovery and localization schemes, we now describe how game-related information is exchanged between player devices and across FinN layers. The basic element of communication is the *Event*, which maintains a list of attribute-value pairs. Notifications of events can be delivered in three different modes: *real-time, multi-hop* or *delay-tolerant*. *Real-time mode* is available when both sender and receiver devices are within transmission range. In the “Tug-of-war” interactive installation, this mode is used to transmit continuous readings from the accelerometer, so that the backbone Station can perform gesture recognition. In this mode, failed transmissions are not repeated. *Multi-hop mode* is available when sender and receiver are not within communication range and the event needs to be delivered without any time constraint. The default routing protocol provided by Sun SPOT is AODV - other routing protocols can be used depending on the expected network conditions and desired performance.

Furthermore, due to various reasons (e.g., arbitrary movement of players, game strategies) communication with the “backbone” infrastructure (the Stations) may not be always possible. During this period, the evolution of the game should not be affected, as players interact with each other and create events. In this case, the *delay-tolerant mode* allows operation on both connected and disconnected modes. When communication with the infrastructure is available, events are transmitted, otherwise events are stored and sent when communication is established. Thus, players can enter and leave the area covered by the infrastructure, enjoying the games and keeping their statistics or history consistent.

**User Interface:** we utilize the 8 LEDs on the devices, the 2 user buttons
and the accelerometer found on the SPOT boards. Player devices execute a subsystem, which processes the samples of a 3-axis accelerometer and recognizes gestures that correspond to game-related actions. We aimed at user-independent gesture recognition, with no training phase involved, and supported four basic gestures: i) clockwise, ii) counter-clockwise and iii) violent movement with direction to the right/left.

These specific gestures were chosen because of their relative simplicity to be recognized and the nature of the games implemented so far. Since we target multiple demographic groups and do not require large precision in our games, we chose to use a simple approach. We sample the 3-axial accelerometer of Sun SPOTs at 20Hz and continuously check the convolutions between different planes in a certain time frame; we map them to this small set of gestures when the samples received are above a certain acceleration threshold. If a more advanced gesture recognition system is required, this service can be transferred in upper architecture tiers in a similar way to the Localization module. However, as can be seen in the user evaluation results presented in this work, users were quite happy in practice with the gesture recognition results.

Regarding user interface methods that are not provided by the devices themselves, significant aspects are graphics, sound effects and music. We utilized Processing [24], which provides a prototyping environment with multimedia capabilities. In all installations described here, we developed graphical interfaces and also used sound (utilizing the Minim library [25]) to provide feedback and enhance the experience of the visitors. At the same time, the movement and number of players, the intensity of their gestures, and the concurrent gestures, generated a background soundtrack covering the whole location. Moreover, “Chromatize Images” and “Chromatize it!” utilized tangible objects: players “dipped” their device in “smart” paint buckets in order to “pick” a color, see Fig. 9, 13. The buckets were essentially containing a Station node with its communication range set to a minimum, acting in a similar fashion to an RFID tag.

5. Cross-layer Issues - Evaluation

In this section we take a closer look at issues introduced by having a tiered architecture with different hardware/software platforms. As mentioned in the previous sections and depicted in Fig. 3 and 4, we have the Guardian, Station and Engine layers to implement our distributed system. The proto-
cols discussed in the previous sections may execute in different layers and, in complex cases, objects may have to traverse all of \textbf{FinN}'s layers. Thus, cross-layer issues and their effect on the system’s overall performance are quite important. The experiments and setup discussed here provide insight into some of the issues we ran into during actual deployment inside more realistic settings.

During the evaluation presented in this section, we utilized laptops and desktop computers, along with a number of Alix Gateways \cite{26}, so as to build our backbone infrastructure. Their small size was ideal for using them as Game Stations or Engines. The lightweight Xubuntu Linux distribution was used on the Alix gateways, while communication between them was established over WiFi. Communication between the Engine for the experiments and the related database (MySQL) was established over 100 Mbps Ethernet. All experiments took place indoors, in a controlled environment, i.e., a number of our office rooms, while all nodes were placed in the same room during each of the experiments. In experiments with multiple nodes, all nodes were placed within a distance of 1-2 meters from each other, on the same height, using no special physical arrangement, i.e., not in a grid fashion, in order to simulate more typical topologies.

As a characteristic example, we study here the propagation time of events generated in the players’ devices, as they cross all of the system’s different layers. We initially assume that a device’s Guardian is already connected to the infrastructure. $time_G$ is the processing time of an event in the Guardian Layer, on its way to upper layers, while $time_S$ and $time_E$ is the time spend in the Station and Engine Layers respectively. In our experiments we transmitted 100 events each time. The overall time for 1 event is 46 ms on average. Individual duration on each layer is formed as follows: $time_G=38\text{ms}$, $time_S=2.5\text{ms}$ and $time_E=5.5\text{ms}$. This makes it clear that $time_G$ acquires the highest percentage of overall time, namely 83%. By further analyzing $time_G$ we observed that the biggest part of this time is due to the built-in communications functions of Sun SPOTs, which are also utilized by \textbf{FinN}. These functions set certain boundaries for the time efficiency of overall communication.

We also evaluated the case where players/Guardians are gathered around a Station, in order to upload the generated events. In this case, we measured the required time for all events to be inserted in the database in relation to the number of Guardians participating in this procedure. While keeping static the number of stored events on each Guardian (100 events), we increased the
number of Guardians sending the events. We were particularly interested in the event reception rate (ms/event) on the Station layer, as well as the event processing rate (ms/event) on the Station and Engine layers. The Event reception rate shows the time needed for an Event to be received by the Station, while event processing rate shows the required time for an Event to be processed on the Station and Engine layers respectively.

Our first set of experiments were conducted using a single Guardian. In this case, the event reception rate was 37.6 ms/event, while the event processing rate on the Station layer was 2.4 ms/event and on the Engine layer 5.3 ms/event. It is obvious that processing rates are significantly higher than the event reception rate. For this reason, the Station and the Engine remain idle for 35.2 ms and 32.3 ms per event respectively, meaning that a bottleneck effect is observed on the Guardian layer. This effect is caused due to the limitations of the devices. Also, we should note that our implementation allows events to be processed in an asynchronous way as already mentioned. This results in a pipeline effect regarding the uppermost layers of the system.

These effects, as well as the difference between event reception and processing rate, are illustrated on Figure 5. Due to this, the overall time for 100 events to be inserted in the database is total$_G + \text{time}_S + \text{time}_E$.

![Figure 5: The pipeline effect between the three layers, Guardian, Station and Engine. One Guardian is updating infrastructure with i events in number. The durations noted are measured for i = 100.](image)

Next, we increased the number of Guardians repeating the procedure described above. This created a realistic scenario, where more than one Guardians are updating the infrastructure simultaneously. We tried to decrease the event reception rate as little as possible, achieving a rate lower than the event processing rate on the Station. This would result in events to
be received faster than they could be processed forcing a significant number of events pushed in the queue. The results are summarized on Table 1.

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Guardian Total</th>
<th>Station Total</th>
<th>Engine Total Proc</th>
<th>Overall Total Proc</th>
<th>Overall Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3760</td>
<td>3722</td>
<td>242</td>
<td>3725</td>
<td>531</td>
</tr>
<tr>
<td>2</td>
<td>6031</td>
<td>5993</td>
<td>272</td>
<td>5996</td>
<td>840</td>
</tr>
<tr>
<td>3</td>
<td>7106</td>
<td>7068</td>
<td>425</td>
<td>7070</td>
<td>1216</td>
</tr>
<tr>
<td>4</td>
<td>8709</td>
<td>8669</td>
<td>511</td>
<td>8674</td>
<td>1839</td>
</tr>
<tr>
<td>5</td>
<td>9115</td>
<td>9074</td>
<td>620</td>
<td>9079</td>
<td>2295</td>
</tr>
<tr>
<td>6</td>
<td>21009</td>
<td>20969</td>
<td>768</td>
<td>20973</td>
<td>2648</td>
</tr>
</tbody>
</table>

Table 1: Distribution of time in milliseconds between different layers. Due to the bottleneck effect, Overall time is approximately same with Total Guardian time. Proc signifies the Processing time on each Layer.

6. Showcased interactive installations overview

We showcased a set of interactive installations at a three-day event, open to the public, that took place at the Lithografio theater, Patras, Greece. A set of 4 installations-games was demonstrated, each one of them revealing different characteristics and aspects of the applications discussed here. Each visitor was given a small lightweight device (a Sun SPOT, see Fig. 9). Visitors interacted with the installations either by physical movement (see Fig. 8) or by performing gestures (see Fig. 12).

Given the showcase venue, we decided on exhibiting a subset of the applications possible with FinN. More specifically, we decided not to include our spontaneous ad hoc games already implemented (Section 3), because of the space limitations we had in an indoors area, and also because of the “interference” and possible annoyances such activities would probably have in the more “static” applications described here. However, a set of experimental results on such applications and a more detailed description are available in [19] and [27].

The installations were placed in such a way inside the venue (see Fig. 6) as to allow guests to gradually experience a spectrum of potential interactions with the digital world. As soon as the visitor entered the “interactive area”, a word cloud from “Magnetize Words” would follow her. Requiring
limited interaction and based on motion visitor’s movement alone, “Magnetize Words” acts like an invitation and warm-up for the interaction about to follow. Moving to the right, on “Tug of War”, participants experience a different kind of interaction; they stand almost still, performing gestures using their arms alone. Moving further through the Venue, visitors combine motion and gestures to play “Chromatize It!” and “Chromatize Images”.

Magnetize Words Our presence and motion in the physical world is the most natural way to interact with the digital world. On the visitor’s entrance in the space of projections, as soon as the visitor is detected, a “poem” is formed and then decomposed into a cloud of words¹, following her motion across a 10 meter long wall. A localization algorithm is used to detect in real-time to recognize the movement of the users along this wall. The word

---

¹Much like the “Magnetic Poetry” kits
clouds following two or more visitors can mix as they physically approach each other, and words are “magnetized” and exchanged between the different clouds. The words are continuously rearranged according to the movement of visitors and duration of interactions with other players, with the word-order continuously alternating, affecting the perceived meaning.

**Tug Of War** Fast and violent hand gestures are used to express our feelings in many daily encounters. Visitors are invited to interact in a highly competitive game by continuously performing fast gestures. The game is played by 2 teams with over 10 persons on each team where each team tries to “pull the rope” and expand over the area of the other team. Certain gestures (that are randomly selected) must be repeated as many times as possible within a given time frame.

**Chromatize It!** Light is a magical medium of significant importance to our daily activities. Visitors move through the physical space that is illuminated by three light sources (red, blue and yellow). As soon as they enter the areas that are brightly illuminated, they “pick” the respective color; we are using proximity detection to detect players that enter these specific areas. These colors can be then used to color an initially white chromatic mass. The visitor must approach the mass and performing a throwing-like gesture to splash the color. The floating mass gradually changes its tone representing the mixture of the combined colors. More than one players can simultaneously participate to collectively color the mass to a particular target color. When the correct mixture has been achieved, players advance to the next level.

**Chromatize Images** Players dip their hand-held devices in paint buckets, “pick” colors, and throw them on the screen, in order to color familiar pop-art images. These images have been preprocessed and separated into color areas, thus players are adding progressively layers, and are not randomly “drawing”. Several players can mix their colors by throwing them simultaneously towards the projection surface and make color combinations. Sounds, different and related to each of the images, reward the players when completing the coloring of each image.

7. **An Exhibition log**

The decision for organizing such an event was taken on January 2010, five months before the exhibition. We had multiple goals: most importantly, it was time to demonstrate our work in front of a general audience. Until that time, our work was presented mostly to a computer-science related audience
Figure 7: Snapshot of the Lithografía Venue: Tug of War game in the center, Chromatize It! at the back/right, Chromatize Images at the back/left.

Figure 8: Visitors move in the physical space with “word clouds” following them around (Magnetize Words).

(i.e., colleagues, students, faculty, etc.). It would therefore be a great opportunity to get feedback from a broader public with varying degree of expertise on technological issues. Our target was to have a large-scale event, in terms of deployment and size of audience, in real-life conditions. In such an envi-
Figure 9: Toddlers and children interact with the smart buckets to pick one of the basic colors (Chromatize It!).

Figure 10: The red and green teams competing each other to “pull the rope” using gestures (Tug of War).

...environment, evaluation regarding satisfaction/fun, as well as the overall system performance on a technical and non-technical level, would be more realistic and accurate. The following five months we extended our work, reworked two already showcased installations (Tug of War, Chromatize It!) and implemented two additional ones (Chromatize Images, Magnetize Words). The
Day One – Monday.

After developing, testing and debugging for five months, we had only one and a half day to setup our showcase. Seven
PCs, 60 sensor nodes and lots of cables needed to be in place and work as expected. However, the setup was completed only one hour before the opening of the exhibition. As this was the first time for our team to setup an installation of this scale, there was a significant deviation on our time estimates. During that single hour, we had to perform an overall test of the installations. Visitors would take a device by entering the interaction area and move through the installation carrying this single device. Everything seemed to be ready.

During this one-hour test a major problem was revealed: it was not possible to move through the area and interact with all installations carrying a single device. The protocol deciding which game should be executed, based on the received broadcast messages of each game-related Station, was not working properly. This was due to the high transmission power of the devices and the nature of signal propagation in indoor environments. Due to reflections, etc., most stations received the majority of events, continuously entering a negotiations phase with the users’ devices and thus quickly saturating the medium. Additionally, the small beacon interval we had set meant that a lot of transmissions were taking place, further worsening this particular situation. We quickly switched to “plan B”, disabling this specific feature. That meant that each game would have a number of dedicated devices and
some extra labor for us reflashing the sensor nodes. However, at 19:00 everything seemed ready for the opening of the exhibition. As proved later, there was still heavy interference between the games because of the short distances between game Stations.

During the first day, visitors came and played from 19:20 until 22:30, with a total of 87157 recorder game-related Events. The Events received from each station, with respect to each installation, are presented in Table 2. An issue evident by simply observing the table, is that 5% of events were received from stations that were not related with the game described on the Event. For instance, when visitors were playing Chromatize Images their device was incorrectly connected to a Tug of War station $S_1$. As a result, properly generated gestures and generated Events had no visual feedback on the corresponding Game.

<table>
<thead>
<tr>
<th>BaseStation</th>
<th>TOW</th>
<th>CI</th>
<th>CImage</th>
<th>MW</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>7868</td>
<td>23</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>$S_2$</td>
<td>3018</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$S_3$</td>
<td>2669</td>
<td>24699</td>
<td>81</td>
<td>9</td>
</tr>
<tr>
<td>$S_4$</td>
<td>113</td>
<td>11</td>
<td>24578</td>
<td>29</td>
</tr>
<tr>
<td>$S_5$</td>
<td>405</td>
<td>0</td>
<td>0</td>
<td>7336</td>
</tr>
<tr>
<td>$S_6$</td>
<td>165</td>
<td>3</td>
<td>32</td>
<td>5899</td>
</tr>
<tr>
<td>$S_7$</td>
<td>830</td>
<td>16</td>
<td>11</td>
<td>9332</td>
</tr>
<tr>
<td>$S_8$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$S_9$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>22</td>
</tr>
</tbody>
</table>

Table 2: Events received from each station separated by game (from left to right: Tug of War, Chromatize It, Chromatize Image, Magnetize Words). 5% (4407 of 87157) events were received from stations which were not related to the game contained in the event message.

In Fig. 14 the analysis of the received events per game is illustrated. Events are divided into four types. i) Init/Start: events generated when a player’s device connects to a Station, ii) Disinit/Stop: events generated when a player’s device disconnects from a station, iii) Gestures: events generated when a player performs a gesture and iv) Errors: events received from unrelated stations.

The fact illustrated on Fig. 14 is that 85% of the events were Init/Start
and Disinit/Stop, 8% were error events and only 7% of the events generated from players’ gestures. Error events were caused by the short distances between the Stations; due to the layout of the venue we could not place them in greater distances.

The large number of Init/Start and Disinit/Stop events were caused for two reasons. The first one is the short distances between the Stations: visitors’ devices were constantly connecting and disconnecting on several Stations; not only to the Stations that were supposed to be registered for each game. E.g., someone was playing Chromatize Images but the device she carried was connecting to stations $S_1$, $S_3$, $S_4$. As a result, each disconnection and connection to another Station generated a Disinit Event and a Init Event respectively. The second reason is the topology discovery protocol (see Echo Protocol, [3, 28]). During the first day, we have enabled the operation of the Echo Protocol for real time response (i.e., 500ms) along with the distinction between uni-directional and bi-directional links. These choices, combined with the large number of participants (over 50 simultaneously), saturated the wireless medium causing continuous misreported connect/disconnect events.

In order to resolve these issues and be prepared for the second day of the exhibition, we proceeded to perform a set of changes. First of all, we set each game to its own radio channel. In this way, the generation of Error events was dealt with successfully, because the devices of one game could not communicate with the Stations of another game. Because of the fact that communication frequency is essential for the rightful operation of the localization protocol, we chose to use the default 802.15.4 channel on “Magnetize Words”, as it offers good granularity on RSSI values. Moreover, we enabled
a “light” version of the neighbor discovery protocol by disabling the bidirectional neighbor recognition and increasing the time response of the protocol (5000\textit{ms}). The mobility of the players on the interactive installations was limited and there was no need for immediate response to any changes regarding players’ position. These changes aimed to reduce the \textit{Init/Start} and \textit{Disinit/Stop} Events and decrease the network load.

**Day Two – Tuesday.** In the second day of the exhibition, visitors came from 19:26 until 22:23. As stated above, the devices’ wireless channel were set specifically for exactly one installation. So, instead of distributing them from a single stand upon entrance to the exhibition area, we had a set of switched on devices in front of each installation for visitors to use - i.e., the device could be not used on another one. All of the nodes in the deployment area were in continuous use throughout the exhibition, with a backup of a few nodes in “standby” in case of the nodes’ batteries depletion. Most of the nodes were in range of each other due to the indoors deployment area we chose. There were 73111 events generated and participants seemed to be enjoying themselves more than the previous day. The events received from each station separated by game are illustrated in Table 3. An overall improvement in comparison to the first day is obvious. There are no error events, due to operation on different radio channels on each game.

<table>
<thead>
<tr>
<th>BaseStation</th>
<th>TOW</th>
<th>CI</th>
<th>CImage</th>
<th>MW</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>29548</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S2</td>
<td>39939</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S3</td>
<td>0</td>
<td>503</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S4</td>
<td>0</td>
<td>0</td>
<td>2819</td>
<td>0</td>
</tr>
<tr>
<td>S5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>82</td>
</tr>
<tr>
<td>S6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>71</td>
</tr>
<tr>
<td>S7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>71</td>
</tr>
<tr>
<td>S8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>67</td>
</tr>
<tr>
<td>S9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11</td>
</tr>
</tbody>
</table>

Table 3: Events received from each station separated by game.

Fig. 15 depicts a breakdown of the received events per game. There are striking differences when compared to the first day. Apart from the fact that there are no error events, a different distribution of the events per type is
observed. First of all, the *Init/Start, Disinit/Stop* events were reduced at a great scale due to the changes in the *Echo Protocol*. Moreover, a great increase in the gesture events per game is observed, while the operation of the overall system was improved. There were no malfunctions, therefore no delays during the gameplay resulting in a significant increase of the performed gestures.

Nevertheless, there were complaints about the difficulty in performing gestures with the SPOT devices, same as in day one. The gesture recognition algorithm used is based on the 3-axis relative acceleration input from the device’s accelerometer. If the relative acceleration is smaller than a threshold, then the gesture is dismissed. In order to ease the gestures in general, we decide to decrease this threshold, making them “easier”. In addition to that change, we unified the results of gestures in “Chromatize Images” and “Chromatize It!” installation. Before that, visitors were supposed to perform only one of the six gestures in order to “throw” colors, making the gestures somehow difficult to visitor without previous experience.

**Day Three – Wednesday.** On the last day, about 250 visitors came from 19 : 40 until 22 : 50. A total of 75304 Events were recorded, reported in Table 4. There was a small increase on gesture events reaching 3% on each game, due to the changes in gesture recognition and report. The Figure with the overall Event statistics is similar to the figure of the 2nd day (Fig. 15) therefore is not presented. At last, everything was working the way it was meant. The smooth operation of the system was verified for a second consecutive day. Overall, around 450 visitors participated in this three-day exhibition. On average, 50 visitors were participating concurrently and specifically on the
Table 4: Events received from each station separated by game.

“Tug of War” installation over 20 players participated concurrently without any issues.

8. User Evaluation Results

After interacting with the installations, visitors were asked to fill in questionnaires regarding their overall experience and system performance, as well as specific details such as the response of the devices used, etc. In total, 136 of the participants (approx. 1/3 of all visitors) have filled in a questionnaire.

Visitors’ age ranged from 5 to 70 years old, the majority of which (i.e., 70%) was between 20 and 34 years old. We had slightly more male visitors than female ones (i.e., 57%–43%). Almost 1/3 of the participants were students, 12% were teachers and about 10% computer engineers. The profession of the rest of the visitors varied including artists, psychiatrists, economists, lawyers, electrical engineers, merchants. None of them had any sort of previous training on how to use the device. In contrast to our previous attempts, [3, 28], this time the audience was largely unbiased. It’s worth mentioning that about 80% of the visitors claimed that they had never seen such a set of interactive installations. For each question, guests chose from a scale from 1 – 5 indicating Strongly Disagree (1), Disagree (2), Neutral (3), Agree (4), Strongly Agree (5). The data extracted are clearly depicting the impact of the changes made after each day.

The first group of charts illustrates the opinion of the guests about each installation as well as the overall experience (Fig. 16, Q1 – Q6). During the
first day, the majority of the visitors (63%) agreed or strongly agreed that the interactive installation were overall fun (Fig. 16-Q1), even though there was a number of problems regarding the communication of the devices. After we dealt with the communication issues and improved the gesture recognition, the percentage of satisfied visitors was increased; during the second day 73% agreed or strongly agreed that the installations are overall fun while on Day 3 this percentage grew further reaching 89%.

Visitors responded well to the use of gestures as a means of interacting. Despite the malfunctions of the applications during first day, 70% preferred using gestures instead of pushing buttons. The percentage soared 84% and 97% on Day 2 and 3 respectively (Fig. 16-Q2).

A thin majority of 50% agreed that Tug Of War was overall fun on Day 1, with this mixed result most probably due to its partial operation during that day. When proper function of the installation was restored (Day 2 and 3), the vast majority of 88% strongly agreed that Tug of War was fun to play.

“Chromatize Images” and “Chromatize It!” had their enthusiasts from the first day with 55% and 40% agreeing that the “fun factor” of these two was high, or very high (Fig. 16-Q4 and Q5). They were also the two installations least affected by the communication problems during the first day. Both were relatively isolated from the rest of the installations, they relied less on radio communication - in comparison to Magnetize Words - and the number of participants was smaller that in “Tug Of War”. When the other two installations (“Tug of War” and “Magnetize Words”) started to operate without any problems, “Chromatize it!” and “Chromatize Images” lost this advantage.

The installation of “Magnetize Words” had the lowest fun factor according to the visitors (Fig. 16-Q6). During the first day, only 30% found the installation fun. The corrections on communication and improvements on localization are not reflected on the opinion of the guests. The rating of Magnetize Words remained in low levels (45% - 50%) most possible due to the limited level of interaction (i.e. the lack of gestures) and the abstract nature of the installation.

The second group of charts (Fig. 17, Q1 – Q5) illustrates the opinion of the audience regarding functional issues. When asked if the installations operate well (Fig. 17-Q1), on Day 1 40% agreed or strongly agreed indicating the excitement of the audience given the major problem that arose during that day. On the second day, after resolving all communication malfunctions, the majority (70%) was very satisfied from the operation of the installation.
Q1) Are these Interactive Installations overall fun?
Q2) Is using gestures more fun than pushing buttons?
Q3) Is the Tug of War game overall fun?
Q4) Is the Chromatize Images overall fun?
Q5) Is the Chromatize It! game overall fun?
Q6) Is the Magnetize Words game overall fun?

Figure 16: Eventually, is the “Fun in Numbers” platform overall fun?

During the third day, this percentage exceeded 88% after the modifications
regarding gestures were applied. Nevertheless, the effect of the changes we made after Day 2, simplifying gesture recognition, combined with those on the communication layer, are distinct on (Fig. 17-Q3); 78% of the visitors agree or strongly agree that on Day 3 the devices responded well to user input. That percentage was significantly lower during the first and second day (10% and 44% respectively).

Regarding the enabling devices, the majority of the visitors (78% on Day 2 and 83% on Day 3) agreed or strongly agreed that the device was easy to use (Fig. 17-Q3). Since SPOTs were not designed to be utilized in interactive installations, this leaves room for improvement. Despite that fact, young kids (even toddlers, Fig. 9) managed to play without significant problems. This is also related to the small size and weight of the device.

When asked about additional feedback means, 53% of the visitors answered that adding a screen to the carrying device would benefit the user interface (Fig. 17-Q4). This percentage was decreased during the following days (43% and 34%) when the operation was restored. An interesting point is also the fact that during the three-day more than 70% agreed or strongly agreed that using vibration would benefit the interaction (Fig. 17-Q5).

9. Lessons Learned - Conclusions

After the discussion about our experiences and the user evaluation results from this 3-day event, we now clearly understand that deploying multiple and multiplayer interactive installations in the same physical location can be very challenging; careful planning is needed in order to implement efficient systems. Utilizing results from the WSN community is a way to deal with such situations and application requirements. The fact that our implementation clearly surpasses other previous relevant results in terms of number of players, is the best argument supporting such an opinion.

There is a certain trade-off when trying to create a pervasive environment and deal with scalability at the same time. On the one hand, a completely ad hoc approach seems to scale and behave in acceptable levels up to around 20 players. On the other hand, if we wish to involve large groups one should adopt multilayer architectures, which introduce additional complexity in deployment and operation. Our approach was to keep this as simple as possible. It is obvious from our user evaluation results, that it was well worth it.

Regarding the user interface, providing a unified experience with multiple installations using multiple user inputs received a warm welcome, even
Q1) Did the Interactive Installations operate well?
Q2) Is the device used on the exhibition easy to use?

Q3) Do you think the device responds well to your movements? Would adding a screen benefit the user interface?

Q4) Would adding a screen benefit the user interface?
Q5) Do you think vibration in the device would be useful?

Figure 17: Questions about the proper operation of our platform.
enthusiastic, from a large sample of visitors. People especially welcomed the idea of having large player groups (i.e., up to 20) at the same time. Our experience was that especially in competitive games, the large number of player seemed to reinforce the whole concept, while certain players tended to revisit e.g., “Tug of War” for additional gameplay rounds.

Moreover, well-known human perception limits are more or less valid in the same sense here, although immediate visual feedback was not required in most cases. In practical terms, setting thresholds below 500 milliseconds, e.g., when detecting network neighborhood changes, was fast enough to create an immersive user experience. However, going over such limits, which would be necessary if we used other networking technologies like Bluetooth would have devastating results. Also, the large number of players needed for creating a compelling experience is served well by the hardware platforms we chose, apart from the efficiency of our software implementation. Thus, currently technologies like 802.15.4 transceivers seem the right way to go when implementing such concepts.

An interesting additional point extracted from visitors’ comments in the questionnaires, pointing to future work, was the following: about 70% of them saw educational extensions to the showcased installations. This fact, up to a certain degree, was backed up by the fact that even 2-year old children quickly adapted to the use of the provided devices and were able to interact with the installations. Thus, our future plans include, among others, informal education applications based on our current work.

Acknowledgments

This work has been partially supported by the European Union under contract numbers ICT-215270 (FRONTS). We would like to thank the Lithografion Theater, for providing us with the stage for experimentation and the sponsors of the three day event: “TO DONTI” publications and Parparoussis winery. We would also like to thank A. Aggelopoulos, T. Asproudis, E. Chita, G. Fotinou, A. Korovesis, V. Kappa, C.Koninis, A. Makrigianni, M. Petraki, E. Theodoridis, E. Thermos, A. Tsoumani, E. Tsota for volunteering to help with the exhibition.

References

[1] W.-Z. Song, R. Huang, M. Xu, A. Ma, B. Shirazi, R. LaHusen, Air-dropped sensor network for real-time high-fidelity volcano monitoring,


