Trove: a Physical Game Running on an Ad-Hoc Wireless Sensor Network

Sarah Mount(1)*, *Elena Gaura*(1)**, *Robert M. Newman*(1)***, *Alastair R. Beresford*(2), *Sam R. Dolan*(3), *Michael Allen*(1)****

(1) Cogent Computing, Coventry University,
Priory Street, Coventry, CV1 5FB, UK.
* s.mount@coventry.ac.uk
** e.gaura@coventry.ac.uk
*** r.m.newman@coventry.ac.uk
**** allenm@coventry.ac.uk

(2) Computer Laboratory, JJ Thomson Ave.,
Cambridge, CB3 0FD, UK, arb33@cam.ac.uk

(3) Cavendish Laboratories, JJ Thomson Ave,
Cambridge, UK, srd24@cam.ac.uk

Abstract

This paper describes *Trove* a physical game implemented on a wireless sensor network (WSN). Architecturally, the WSN is a decentralized system, exhibiting local node processing and information extraction, collaborative inter-node behaviour and local decision making capabilities. From the perspective of the players, Trove is a multi-player, real time, physical game. The user-centered narrative, configuration and game play of Trove are presented as well as its design and implementation.

Trove will be used at Coventry University as a pedagogical aid in under- and postgraduate modules which incorporate concepts from pervasive computing and sensor networks; and also for the dissemination of research work to members of the public. Although educational through its use, the work presented here concerns, from a technical viewpoint, the very specifics of physical WSN design, implementation and deployment and forms a good basis for further proof of concept experimentation within the domain.

1. Introduction and motivation

Although relatively new, the field of wireless sensor networks (WSNs) has become a very prolific one, both in terms of number of projects and publications and the specialisms involved in researching various topics within the area. The past five years have seen a large variety of 'dream applications' for WSN being proposed, from Berkeley's Smart Dust [12] to Coventry's Mars exploration [11]. The field has become highly multi- and inter-disciplinary, with scientists from both Engineering and Computer Science, and various application specialists being involved in the research. One of the consequences of this explosion in activity is the proliferation of WSN themes at a large number of conferences informing traditional science domains, such as Control Engineering, Robotics, Automation, Microelectronics, all branches of the Computer Sciences, Biomedical Engineering, etc. One of the outcomes of any academic research is to reflect back and adjust the taught curricula at both undergraduate and postgraduate levels, hence, it is no surprise that WSN have already become part of a number of courses, in addition to the obvious Pervasive Computing ones.

The work presented here stemmed from the authors' experience of having already attempted to introduce and deliver this topic to students on undergraduate and postgraduate traditional Computer Science and Computer Systems courses. It was found that students find it surprisingly difficult to conceptualise fully decentralised systems and that traditional educational tools have proved, for us, inadequate at explicating the basic concepts behind WSNs.

In spite of the enormous research effort in the WSN area, there are very few practical WSN deployments as yet (one of the most successful one that the authors are aware of is the SECOAS project [10]). Existing sensornets are relatively limited in terms of number of nodes and complexity of design/implementation, compared with those proposed in theoretical studies. This leaves educators with few real life examples to aid the delivery of the dry abstract concepts which support the design and implementation of WSN. The teaching task is so much more difficult if the core of the particular degree within which WSNs are delivered is not fully within the Pervasive Computing area and the time limits for delivery are tight. Therefore, we see an urgent and real need for a powerful tool to aid delivery and ensure thorough understanding of concepts.

Initial efforts within the Cogent Computing research group resulted in the development of a WSN application scenario, which has been scripted and animated by professional artists [4]. The film identifies key points within a WSN deployment to highlight various engineering and design challenges faced by the development team. These key points then inform a discourse on the specific challenges and anchor the delivered theory back to its overall purpose.

A second round of efforts, however, looked at gaming as an interactive aid. It was decided to use gaming not only for learning and delivery of abstract concepts, but also as a test-bed for students' practical work and exploration of ideas in the domain. Furthermore, research ideas and novel concepts developed currently by the group members will be tested using the hardware set-up for the game (an example here is the group's work on redundancy, fault detection and management for WSNs).

The paper is structured as follows: Section 2 describes the user-centred aspects of the game, game configuration and game play. Section 3 describes the system design and implementation and Section 4 concludes.

2. Game play

Trove is a competitive, multi-player game. The game has been tested for 10 players but the design is scalable. The object of the game is for the players to negotiate a closed environment (usually a room, although the game could be played outside) and reach hidden treasure (the authors were inspired by the narrative of Indiana Jones [14]). Each player wears a hat embedded with a MICA2 mote and sensor board (see full hardware description in Section 3.1) whose sounder will 'beep' if its sense data exceeds a threshold which causes the player to lose a life. The first player to reach the treasure, with lives still intact, is the winner.

Facing the players, usually on a projector screen, is a graphical interface which indicates the number of lives each player has remaining, which sensors are currently close to or over their thresholds and reports on whether any motes are unable to communicate (see Figure 1).

Figure 1: The graphical interface of Trove during a ten-player game. The bars next to each players name indicate the amount of lives left. On the right of each health bar are two indicators showing the current status of each players light sensor and accelerometer.

Overall, the game environment consists of:

- the players and their motes;
- a PC connected to a single mote known as the *base station;*
- a projector displaying the games graphical interface as described above; and
- a non-player known as the *supervisor*, who manages the game configuration in software (via the GUI) and organises the players.

During the game each mote acts autonomously (individually or collaboratively with other motes) to calculate whether its players should lose a life. The base station is connected to the PC via a serial link and is only used to passively receive sense data from the motes and update the GUI in a 'best effort' fashion.

Two types of game may be played: a basic game or a collaborative game. These are described below.

2.1. Basic game play

In a basic game, either the light sensor, accelerometer or both may be in play and each can be triggered by exceeding a threshold.

The light sensor is triggered by overhead lights. Strip lighting is particularly useful, as players can learn to walk between rows of lights in order to make progress without losing lives. The two axis accelerometer is set off by sudden movements or by players tilting their heads. Steady movement through the room is therefore the safest tactic.

2.2. Collaborative game play

Basic game play may be used to explicate the use of MEMS sensors, demonstrating the nodes ability to collect sense data and transmit and receive signals from the base station. During a basic game, nodes do not communicate with one another (although they do send data to the base station). Trove provides a second type of game to demonstrate collaborative behaviour in WSNs.

In a collaborative game, the sensors trigger threshold (upon which a life is lost) is set by each players physically nearest neighbor. So, if player A has a light sensor reading of n_A and their nearest neighbour B has a reading of n_B, and $n_B > n_A$, then the player B loses a life and A does not.

This can lead to players taking some interesting tactical decisions. If players agree to work in teams, one or more can be 'sacrificed' leaving the last group member to cross the finishing line. On the other hand, players may choose to sabotage their nearest neighbours chances by standing very still in a dark area, leaving their neighbour to try to move closer to other players before they run out of lives.

2.3. Game configuration

It is intended that one non-player, known as the 'supervisor', will set up and coordinate the game. This person is responsible for entering the names of each player, choosing the type of game to play, configuring Trove before the game starts and organising the players.

For any game implemented on a WSN, where thresholds on sense data are used to reduce the number of player lives, the following step must be taken to configure the game:

- mote discovery;
- calibration; and
- upload.

During 'mote discovery' the game interface attempts to contact as many motes as there are players, via the base station, and grab the latest available sensor readings.

If all motes have returned data to the application, then the motes can be calibrated. The supervisor should place the motes somewhere appropriate and instruct the interface to perform calibration. The interface then applies an offset to the sense data it is receiving from all the motes and displays the newly calibrated values, rather than the raw probe data. It is important to keep the motes level and still during calibration and to place them somewhere where relatively low light levels reach the photoreceptors (unless the supervisor wishes to make the game fiendishly difficult!).

Calibration may be performed many times. Once the supervisor is content with the results, s/he should instruct the interface to upload the calibrated offsets to the motes, which will set the thresholds for triggering loss of lives.

When the players are ready the game can be started. After this, the supervisor need only intervene if a mote becomes unreachable (in which case it can be 'Reset' in the game progress dialog – see Figure 1).

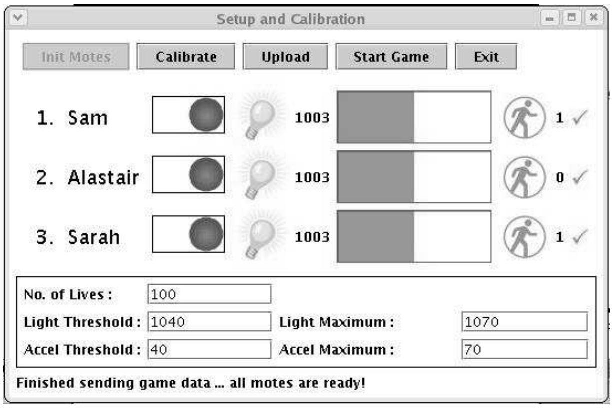

Figure2: *Setup and calibration dialog.* At this point, the sensors have been calibrated and the ticks on the right of each accelerometer value show that the new offsets have been successfully uploaded to all the motes.

2.4. Pedagogical issues

For pedagogical purposes, we were keen that each type of sensor could be used alone, so that players may be able to familiarise themselves with each available sort of sense data separately.

Equally, the two different types of game enable us to introduce decentralised sensing and network collaboration separately. This ensures that the supervisor can start each game when s/he considers the players to be ready to engage with the relevant concepts. Depending on the background of the players, several games may be played in one session, or these may be spread out over several weeks, perhaps over a series of lectures.

3. Implementation

3.1. The hardware

We have implemented Trove on Mica2 motes from UC Berkeley [6]. These are small, battery-powered sensors which utilise an ATMega128L processor (see figure 3). They contain 4kb of Ram, 128kb of program memory and 512kb of serial flash memory in which to log data. As standard, the motes are equipped with 3 LEDs (red, yellow and green) and a ChipCon CC1000 radio for wireless communication, operating in the 900MHz frequency band. In addition to these standard capabilities, we have equipped the motes with MTS310 sensor boards [7], providing a microphone, sounder, dual axis magnetometer, dual axis accelerometer, as well as temperature and light sensors. The motes software is loaded via a purpose-built programming board, connected to the serial port of a PC. Each mote is powered by two AA batteries which, depending on their capacity, can enable them to run from 1½ to 17 months, based on a 'duty cycle' of 1% working time and 99% sleeping. The Mica2 motes run an operating system called TinyOS (Tiny Micro threaded Operating System) [9]. Developed at University of California, Berkeley, TinyOS is an operating system designed to run specifically on an embedded sensor. TinyOS consists of a scheduler and a set of linked components. Components within the system interact by way of receiving commands and handling events, which propagate from the highest-level component all the way down to the hardware level, where component handlers are connected directly to the hardware interrupts.

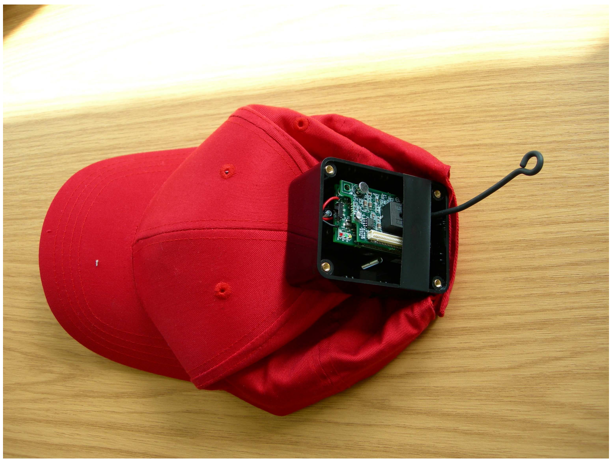

Figure 3: A MICA2 mote [6] with MTS310 sensor board [7] attached to a baseball hat. These are worn by players of Trove, during the game.

3.2. Hardware interface and sensing

Our implementation of Trove is programmed in an event-driven style. A timer is used to trigger regular sampling of sensor data (e.g. readings from the photoresistor). The results are retrieved asynchronously, and compared with the thresholds set when the game was initialised. If the current readings from the sensors exceeds the threshold value, a life is deducted, the red LED is illuminated and the sounder 'beeps'.

TinyOS allows the programmer to define custom radio packets. The current implementation of the basic Trove game defines two payload types:

1. control packets, which are sent from the base station to the motes, and are used to perform mote discovery and set game thresholds for each of the mote sensors (as specified in Section 2.3); and
2. in-game packets which are transmitted by the motes to the base station to provide a best-effort update of the shared display.

These latter packets contain both the raw sense data readings as well as the motes current count of the number of lives the player has left.

We are currently in the process of introducing a third packet type to enable collaborative gaming. This third packet type is used to provide *ranging* by measuring the difference in arrival times of packets transmitted via the sensor board's audio sounder and a packet sent via the radio interface. The transmitting mote sends the audio and radio packets *simultaneously*; any mote receiving both these messages can use their time-difference-on-arrival to determine the distance between the transmitting and receiving motes. An estimate of distance can be derived because there exists a large disparity in the speed of sound and the speed of light in air [1,13]. The transmitted audio tone is pulse modulated with the identity of the mote, together with a unique sequence number. The mote identity enables other motes within hearing range to use the microphone sensor to determine the identity of the transmitting mote. The sequence number enables all motes who successfully receive an audio packet to correlate this message with the correct radio transmission. The radio transmission associated with the audio tone includes details of the current sensor levels, number of lives the mote has, and current collaborator. This ranging method enables each receiving mote to maintain two data structures:

1. a list of neighbours ordered by distance; and
2. for each neighbour a list of their current sensor readings.

The third radio packet type contains similar information to the data sent by the in-game packets used for the basic game. Therefore, in the collaborative game, the base station can listen to the ranging messages to update the details shown on the shared display.

Simultaneously creating both a fun, and yet educational, game on a WSN is a challenging problem. With this implementation we have tried to maintain simplicity and clarity over excessive "feature creep", and therefore ensure the underlying code remains readable and useful as an educational tool; it is important however that this requirement doesn't make the game dull or hard to play. The human-computer interface of the motes is too restrictive to provide adequate feedback to participants.

In our implementation, we believe that the shared display has been particularly successful: without it the game would be much more difficult to play, and its use as a teaching aid would be greatly reduced.

3.3. User interface

As the human-computer interface of the motes is minimal, the shared visual display, projected onto a wall or viewed on a large screen, complements it. Data packets from the motes are recorded by the computer, which is used to drive the display. The computer uses the in-game data to update the display in a best effort fashion, providing the contestants and general audience with an easy way to determine current game information.

Figure 1 provides an example snapshot of the shared display, which includes current sensor readings and number of lives for each player, along with the amount of game time remaining; the game supervisor can use the interactive elements of the display to start and stop the game, as well as use it to reset the status of individual players.

A separate GUI allows the game supervisor to configure the motes, as outlined in Section 2.3. Figure 2 shows the setup and calibration dialog, which the game supervisor can use to configure the motes prior to starting the game. A description of the two screens follows:

3.3.1. The setup and calibration screen

Each step in the setup process corresponds to a separate button on the form. Step one is to initialise the motes, and to begin listening for incoming data (‘Init Motes’). Sensor readings are displayed as they arrive. Step two is to calibrate the motes against each other (‘Calibrate’). A linear offset is applied to sensor readings so that all motes give the same values under the same starting conditions. At this stage, the user selects the maximum threshold values for each sensor type (see text). The last step is to transfer the game settings to the motes (‘Upload’), and to begin the game (‘Start Game’).

3.3.2. The in-game screen

The purpose of the screen is to track the current game state, and to provide simple visual feedback to the players. For each player the display shows the current sensor readings (light sensor and accelerometer panels), and remaining lives (horizontal bar). Colours are used to alert a player when their sensor readings are close to (yellow) or exceed (red) the threshold values. Buttons on the form allow the supervisor to start and stop the game, or to reset motes individually.

4. Related work

Various games have been implemented which make use of sensing devices and wireless communications. For example, the game narrative of *Pirates!* [3] takes place in a virtual ocean. Players command a virtual ship and sail the ocean trading and fighting to gain wealth. Real life RF technology was used to determine distances between players and game resources and to trigger game events.

The object of *Unmasking Mister X* [2] is to determine which player is ‘Mister X’, based only on sensor readings from Mister X’s wearable sensors.

In *Can You See Me Now?* [8] players either run around (real) streets, tracked by GPS, or move an avatar around an on-line 3D representation of the same streets. Runner players attempted to 'catch' on-line players by chasing their (virtual) location. When a running player coincides with an on-line player the latter is out of the game.

Treasure [5] is a game which exploits the lack of connectivity in WSNs. Players move outside wireless coverage (tracked by GPS) to collect virtual 'coins', then move in to an area of high network strength to 'upload' the coins to a game server. Players within network range of one another can steal each others coins. Other game events are triggered by the physical location of players.

However, these games are typically run on equipment with relatively large resources (often PDAs with GPS capability). Trove is novel in that it is implemented on sparsely resourced hardware and it is intended for pedagogical use.

5. Conclusions and future directions

We have presented Trove, an educational game running on an ad-hoc wireless sensor network. Players succeed in the game by crossing the physical environment without loosing all lives. Lives are lost when sense data from their attached motes exceed a configurable threshold.

We intend to incorporate more sense data into the game (bearings can be obtained from the dual axis magnetometer on the MTS310, for example). Ideally, sensors could be combined using propositional calculus style rules. For example, a game may be run where players loose a life if the photoreceptor reads in excess of 1000ca OR the accelerometer reads less than $500m/s^2$ OR the player is facing South.

As outlined in the Introduction, as well as using the game to explore educational and teaching issues, we are considering using the hardware for testing more sophisticated algorithms which directly inform the current research agenda in the domain. The design of the WSN and its physical implementation have been produced so that new functional modules can be slotted elegantly. The implementation is amenable to testing of several localization algorithms we have developed theoretically as well as assessing the robustness of our node fault detection and management strategies.

6. Acknowledgments

The authors would like to thank Coventry University for generously funding the implementation of Trove.

7. References

[1] Allen, M. (2005) *Sound location in wireless sensor networks.* BSc(Hons) Computer Science final year thesis, Coventry University, UK.

[2] Antifakos, S. and Schiele, B. (2002) *Bridging the gap between virtual and physical games using wearable sensors*. In proceedings of the 6th international symposium on wearable computers, Seattle, Washington, USA, October 2002, pp 139-140.
[3] Bjork, S., Falk, J., *et al* (2001) *Pirates! - Using the physical world as a game board.* In *proceedings of Interact 2001*, Tokyo, Japan, 2001, p9-13.
[4] Burns, J., Gaura, E.I., Mount, S.N.I., Newman, R.M. and Woodcock, A. (2005) *Delivering difficult concepts using visual representations.* In *Contemporary Ergonomics 2005*. Editted by P.D. Bust and P.T. McCabe. p 120-124. Published by Taylor & Francis, London, UK.
[5] Chalmers, M., Bell, M., Brown, B., Hall, M., Sherwood, S. and Tennent, P. (2005) *Gaming on the edge: using seams in ubicomp games.* In *proceedings of ACM Advances in Computer Entertainment (ACE).*
[6] Crossbow Inc., *MICA2 Wireless Measurement System Datasheet*
http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf
[7] Crossbow Inc., *MTS Sensor, Data Acquisition Boards Datasheet*
http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MTS_Datasheet.pdf
[8] Flintham, M., Anastasi, R., *et al*. (2003) *Where on-line meets on-the-streets: experiences with mobile mixed reality games*. In *proceedings of ACM Computer-Human Interaction*, p569-576.
[9] Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D. and Pister, K. (2000) *System architecture directions for networked sensors.* In *proceedings of the Ninth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Cambridge, MA, USA.
[10] Ian Marshall (2004) SECOAS: Quarterly Progress Report: Pervasive Computing in the Environment http://www.ee.ucl.ac.uk/pipermail/secoas/att-0050secoas04.doc
[11] Newman, R.M., Gaura, E.I., Tobor, J. and Mount, S.N.I. (2005) *A realistic dream - a top-down feasibility study for MEMS planetary exploration.* In *Proceedings of the NSTI Nanotechnology Conference and Trade Show (Nanotech'05)*, Anaheim, California, USA. p363-366.
[12] Pister, K.S.J., Kahn, J.M., and Boser, B.E. (1999) *Smart dust: wireless networks of millimeter-scale sensor nodes*. Highlight article in *Electronics Research Laboratory Summary.*
[13] Sallai, J., Balogh, G., Maroti, M. and Ledeczi, A. (2003) *Acoustic ranging in resource constrained sensor networks.* Technical report, ISIS, Vanderbilt University.
[14] Spielberg, S. (1981) *Raiders of the lost ark*. Paramount Pictures, USA.