Knowledge Media Institute

CitiTag Multiplayer Infrastructure

Kevin Quick and Yanna Vogiazou

KMI-TR-138

http://kmi.open.ac.uk/publications/papers/kmi-tr-138.pdf

March, 2004

Introduction

The current technical report describes the architecture and components of the CitiTag wireless location based game, developed by the Knowledge Media Institute's Centre for New Media in close collaboration with HP Labs' Mobile Bristol team. CitiTag is a case study for Yanna Vogiazou's PhD work in social and ubiquitous computing and has been conceived and developed by Yanna Vogiazou (KMi) and Bas Raijmakers from the Interaction Design Department at the Royal College of Art.

Yanna Vogiazou's PhD research explores how the design of interactive technologies can enhance social interaction and the sense of presence among large numbers of people through play. Previous online studies have shown that spontaneous social behaviours can 'emerge' among groups present in multi-user environments, even without explicit and verbal communication. CitiTag takes this a step further by blurring virtual presence with real world, physical presence. The aim is to explore technology mediated social experiences and emergent behaviours in public spaces through playful interaction, based on the awareness of other players' presence. The project is motivated by the hypothesis that very simple game rules based on presence states (e.g. I am Green and 'tagged') can result in an enjoyable social experience, stimulated by real world interaction among players. Another hypothesis is that certain interactions develop once a critical mass of users has been achieved, making the experience different every time as it is stimulated by group dynamics. A further part of the project, undertaken by Bas Raijmakers is focused on how such experiences can become part of the fabric of daily life and the use of video in the design process. Through this project we will identify design implications for future technology mediated social experiences and find out how experiments like the CitiTag user trials can inform the experience design process.

This project would not have been possible without the invaluable contribution of the following people. We would like to thank Jo Reid (HP Labs) for project management and support, Erik Geelhoed (HP Labs) for guidance in research methodology, Ben Clayton (HP Labs) for developing the Mobile Bristol application and technical support, Peter Scott (KMi) for project support and ingenious suggestions, John Linney (KMi) who incorporated the Flash client logic, Lewis McCann (KMi) who provided infrastructure support and Marc Eisenstadt (KMi) for general support, great ideas and guidance as Yanna Vogiazou's PhD supervisor. We also thank Phil Stenton and Richard Hull from HP Labs whose interest in the key ideas initiated this project and for their contributions in concept development.

1 The Game Design

CitiTag is a multiplayer team game. It is played using GPS (Global Positioning System) enabled, handheld, PocketPCs connected to a wireless network. As a player, you belong to either of two teams (Reds or Greens) and you roam the city, trying to find players from the opposite team to 'tag'. When your PocketPC detects a player of the opposite team in the neighbourhood you get the opportunity to 'tag' that player.You can also get 'tagged' if one of them gets close to you. If this happens, you need to try and find someone from your team in vicinity to set you free, 'untag' you. Each game event (e.g, someone is close and you can tag/untag them) appears as an alert on the iPaq with a sound (Fig. 1). A team wins by tagging all of the players on the opposing team.

Figure 1. Three different game states. Figures on top of the screen show the number of free and tagged players for each team.

Figure 2. CitiTag in action: a user trial at the Open University's campus

2 CitiTag Architecture

This report gives a technical overview of the programme logic and the architecture for the CitiTag game.

Brief description of the game components

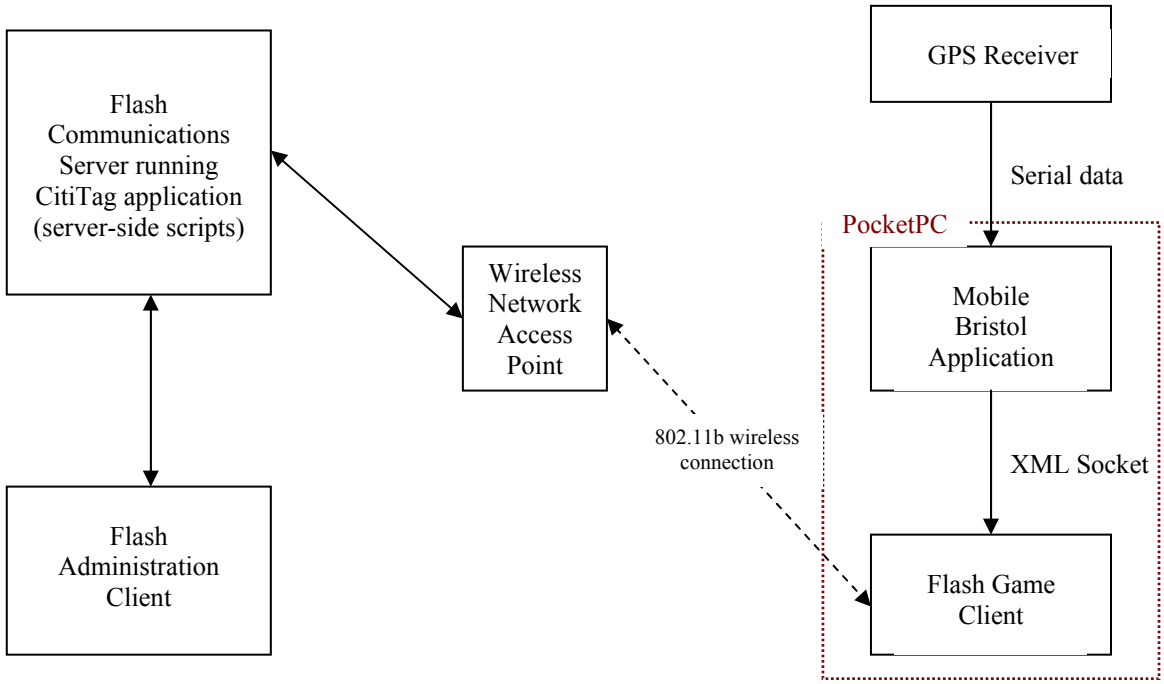

Figure 1. Showing in general terms the structure used for CitiTag

2.1 GPS receiver

The GPS is used to pinpoint the location of players in the game. The GPS units have a serial output, providing the GPS data in a format conforming to an international standard. The GPS units used in the game trials were Garmin GPS-35, and these were connected to the RS232 port of the Hewlett-Packard, iPaq PocketPC.

2.2 PocketPC

The handheld computer used in the game was an Hewlett-Packard, iPaq PocketPC, running Microsoft's Pocket PC 2002. The iPaqs were also WiFi (802.11b) enabled, allowing them to connect to wireless networks. Additionally, the following software was also required:

- The Mobile Bristol Application files (see below)
- Macromedia Flash 6 for PocketPC (www.macromedia.com)
- The Flash Game Client (SWF file), settings XML file, and associated HTML web page (see below)
- FlashAssist (www.ant.com) - a utility that allows Flash files to be viewed locally on the PocketPC, and for them to occupy the entire screen space (i.e. no menu bars visible)

2.3 Mobile Bristol Application

The Mobile Bristol Application has been developed by Mobile Bristol (www.mobilebristol.co.uk). In the context of the CitiTag game, the Mobile Bristol Application is used to read the serial data from the GPS receiver and to pass it via an XML socket, to the Flash Game Client. The XML data transmission is event driven i.e. data is sent when either there is a new position from the GPS, or, when the GPS gains a position fix, or, when the GPS losses position fix.

2.4 Flash Game Client

The program on the PocketPC used to actually play the game is written in Macromedia Flash. This program forms an XML socket with the Mobile Bristol Application, and also connects, wirelessly, to the CitiTag application on the Flash Communication Server (www.macromedia.com). The Flash Game Client interface allows users to enter a game after having first completed their name and having selected a team (red or green). Once in the game, coordinates passed to the Flash Game Client from the Mobile Bristol Application are relayed to the CitiTag application (server-side scripts) on the Flash Communications Server. In turn, the CitiTag application relays back changes in state e.g. being tagged, or, being able to tag someone etc. and the Flash Game Client presents the user with the appropriate screen. A more detailed description of the programming is given below.

2.5 Flash Communications Server

The Flash Communication Server is a multi-user server, produced by Macromedia (www.macromedia.com). When used in conjunction with Flash movies the Flash Communication Server, allows sharing of data amongst clients, streaming audio and video and many other features. In the case of the CitiTag game it is used as the central hub, maintaining the game state, and receiving and sending data to all connected players etc.

2.6 Flash Administration Client

The Flash Administration client is used to administer and monitor the game. Its functions include

- Starting games
- Pausing games
- Resetting for a new game
- Adjustment of critical game variables
- Monitoring the number of players on each side and how many are tagged
- Keeping a log of players movements and game events

The Administration Client is a Macromedia Flash movie which is viewed on a computer via a web browser. The computer has to be connected to the network used by the game clients, either via a physical cable or via a wireless link. In the case of the game trials at the Open University, March 2004, the administration client was viewed on the same machine that was running the Flash Communication Server.

The Administration Client connects to the CitiTag application on the Flash Communication Server.

3 Game processes

3.1 Flash Game & Administration Client start-up and connection negotiation

Both the Flash Game Client and the Administration Client need to make a connection to the CitiTag application on the Flash Communication Server. To do this they need to know the address of the server and a port which can be used to make a connection.

For flexibility XML files are used for both clients (co-located with the Flash files). The XML files contain the address of the server and a port number to use for the connection. This allows for relatively easy reconfiguration for different networks. When launched, the Flash clients load the data from the XML file.

3.2 The very first connection

The very first client to connect to the server-side CitiTag application, causes a new instance of the game to be created on the Flash Communications Server. This launch triggers an 'Application Start' function to be called in the CitiTag's server-side scripts (this sequence of events occurs whether it is the Administration Client or a PocketPC Game Client connecting first). The 'Application Start' function initialises variables used in the game.

3.3 Events which occur on connection

Each client which connects to the CitiTag application triggers an 'On Connect' function. This function has a range of parameters passed to it, from the Flash Game Client. These include:

- the name of the player connecting. This is used as a unique identifier i.e. only one person of a particular name is allowed in the game. The name is provided by the player on the logon screen.
- the team the player has selected (red or green)
- the latest GPS coordinates
- Client type i.e. whether the client connecting is a Flash Administration Client or a Flash Game client

The operation of the 'On Connect' script depends on the client type, and is explained in detail below.

3.3.1 Game client connection

The script scans through all the players currently connected to see if there is an open connection to a player of the same name (as mentioned above, the name is used as a unique identifier). If an open connection to a client of the same name is found, the connection is closed to that player. This sequence of events is to handle reconnections after an unclean disconnect event, and also to ensure that the name remains a unique identifier.

The scripts then loop through all the registered players in the game, comparing the name of the connecting player with those already stored. If a matching name is found, then this identifies a reconnecting player, and previously stored data is retrieved for that player e.g. whether they are 'tagged', how many players they have 'tagged' and how many players they have 'untagged' etc. If no matching, existing player is found, then the a new player is added to those already in the game.

Next, the script looks to see if the current game has in fact ended, if it has the server-side script calls a 'game ended' function on the connecting player's game client, passing a number of parameters including:

- Which team won
- How many members on each team
- How many members on each team are 'tagged'
- How many players had been 'tagged', or, 'untagged' by the player connecting

The connecting player's game client, in turn shows the 'end game' screen with appropriate statistics displayed.

If the game has not ended, the scripts, calculate the current number of people in each team, and how many of those have been tagged etc. If an administration client is connected, the server-side scripts call a function on the administration client to pass the updated statistics. The server-side scripts also return this data to the connecting client, together with the players status (i.e. 'tagged' or 'untagged'). The connecting player's Flash Game Client, then presents the correct screen and statistics.

Finally the 'On connect' script calls a function on the connecting client to pass the last text message the administration client has broadcast to the players.

3.3.2 Administration client connection

The server-side scripts call functions on the administration client, passing back data such as:

- How many members on each team
- How many members on each team have been 'tagged'
- Current setting of 'tag distance' (defines the radius of a circle around a GPS coordinate, within which a client can be 'tagged' or 'untagged')
- Current setting of 'log distance' (defines how much a player's GPS position must change before the administration client makes a record of the players position in its log)
- Game status i.e. either in progress or paused

3.4 Maintenance of connection status, handling disconnections and game drop-outs

Once the Flash Client has connected to the server-side CitiTag application, the game has to robustly handle disconnection events and subsequent reconnections. The reconnection techniques employed have already been discussed in the previous section, and this also includes one part of the disconnection strategy (see also below).

3.4.1 A disconnection detected by the Flash Communication Server

A player's disconnection which is detected by the Flash Communication Server, automatically calls an 'On Disconnect' function in the CitiTag server-side scripts. This function sets the player's status as 'unavailable', and as such they cannot be tagged, or, untagged.

3.4.2 A disconnection not detected by the Flash Communication Server

To cater for this eventuality, every two seconds, each connected Flash Game Client, calls a 'New Position' function on the CitiTag server-side scripts, passing the latest

GPS coordinates (see below for the further explanation of this function). The server-side scripts log the time this function was called (a separate value is kept for each player). The scripts then examine the times when each of the other players (whose status is 'available') last updated their position, and if over seven seconds have elapsed since the last update, the scripts trigger an 'On Disconnect' event for that player, and consequently set the disconnected player's status to 'unavailable'. Furthermore, if a player has been disconnected, and they have not updated their position for more than five minutes, the scripts automatically change that players game status to 'tagged'. Following such an 'auto-tagging' the server-side scripts call functions on connected clients to pass new game statistics (i.e. number of people on each team and how many of them are tagged), and also call a function on the administration client (if one is connected) to append the tagging event to the game log. Auto-tagging is necessary to prevent the game being adversely affected by players dropping permanently out of the game e.g. due to equipment failure, or even as a ruse to prevent their team losing (whilst disconnected a player cannot be tagged by another player).

The Flash Game Clients, react to disconnections, by returning to the logon screen, and thereby allowing for a player to try to reconnect.

3.5 Position calculations & game event alerts

The Flash Game Clients, store a player's current GPS coordinates in variables. These are updated when the Mobile Bristol application passes new values via the XML socket. If the GPS loses its position fix, then the Mobile Bristol application passes this event to the Flash Game Client, which in turn sets the GPS coordinates (longitude and latitude) to –1. On loss of GPS the Flash Game Client also calls a CitiTag server-side function, which sets the player's game status to 'unavailable'.

As explained in the previous section, every two seconds the Flash Game Client calls a 'New Position' function on the CitiTag server-side scripts. This server side function, also sets the player's status to 'unavailable' if the GPS coordinates are –1. If an administration client is connected to the game, the server-side scripts calculate the distance between the current position and the previously logged values. If the distance exceeds a critical value (which can be set from the administration client), then the coordinates, and date/time are sent to the administration client for appending to the game log.

If the player is 'unavailable', because of loss of GPS fix, the scripts check whether they are 'tagged' or not, and subsequently pass the result to the Flash Game Client, so that the appropriate screen can be shown.

If the player is 'available', i.e. has valid GPS coordinates, the scripts calculate the distance between the player, and all the other players. The outcome of this comparison depends on whether the player is tagged, and follow a strict order of priority. The possible outcomes are as follows, and are listed in order of priority (highest first).

Player is not tagged

1) Another player of the opposite team is within range to tag. Server-side script calls function on players Flash Game Client, to go to the screen for tagging a

player. The function call also passes the name of the person who can be tagged.

2) A tagged player of the same team is within range to untag. Server-side script calls function on players Flash Game Client, to go to the screen for untagging a fellow team member. The function call also passes the name of the person who can be untagged.
3) No player of the opposite team is within range to tag, and no tagged player of the same team is within untag range. Server-side script calls function on players Flash Game Client, to go to the basic 'you are on red', or, 'you are on green' screen.

Player is tagged

1) Another player of the same team, and who has not been tagged, is in range to untag the player. Server-side script calls function on players Flash Game Client, to go to the screen for indicating that they are in a position to be untagged. The function call also passes the name of the person who can untag them.
2) No player of the same team is within range to untag the player. Server-side script calls function on players Flash Game Client, to go to the screen showing that they are tagged.

3.6 Tag Event

If a player's Flash Game Client is showing the 'tag' screen, they can attempt to tag the opponent by pressing the tag button. The button calls a CitiTag server-side function for tag events. Through this function call, the Flash Game Client passes the details of the person being tagged. The script first checks that the player who is attempting the tag is not tagged themselves, and that the target player is 'available' for tagging, and if this is the case, then the status of the target player is changed to 'tagged'. The player initiating the tag has their tally of successful tags incremented.

Following a successful tag the events following occur:

- Details are sent to the administration client (if connected), for appending to the game log i.e. which player tagged whom, and the date/time at which it occurred.
- Updated game statistics are sent to all players (i.e. total number of people on each team and how many have been tagged). The updated statistics are also sent to the administration client if it is connected.

If the tagging has resulted in a victory the following events also occur:

- The server scripts call functions on all the Flash Game Clients to send them to the appropriate 'end of game" screen.
- Details of the victory are sent to the administration client (if it is connected) for appending to the game log.

3.7 Untag event

If a player's Flash Game Client is showing the 'untag' screen, they can attempt to untag a player by pressing the 'rescue' button. The button calls a CitiTag server-side function for untag events. Through this function call, the Flash Game Client passes the details of the person being untagged. The script first checks that the player who is

attempting the untag is not tagged, and that the target player is 'available' for untagging, and if this is the case, then the status of the target player is changed to 'untagged'. The player initiating the untag has their tally of successful untags incremented.

Following a successful untag the following occur:

- Details are sent to the administration client (if connected), for appending to the game log i.e. which player untagged whom, and the date/time at which it occurred.
- Updated game statistics are sent to all players (i.e. total number of people on each team and how many have been tagged). The updated statistics are also sent to the administration client if it is connected.

3.8 Administration Client

The administration client has the following features (some of which have already been described).

- Maintaining a log of a game.
- Showing statistics of how many people are playing the game and how many are tagged.
- Pausing a game. A button on the administration client which calls a server-side function which toggles a variable between true and false. If the variable is false, tag events and untag events are blocked (i.e. game paused).
- Changing the 'tag distance'. The updated 'tag distance' is passed as a parameter in a function call to a server-side function. This function updates the appropriate server-side variable.
- Changing the 'log distance'. The updated 'log distance' is passed as a parameter in a function call to a server-side function. This function updates the appropriate server-side variable.
- Sending a text message to all Flash Game Clients. The message is sent as a parameter in a call to a server-side function. This server-side function in turn passes the message as a parameter in a function call to all the Flash Game Clients. Finally, the Flash Game Clients, display the message on the iPaq screen.
- Resetting a game. This calls a server-side function which resets all game variables, and disconnecting all Flash Game Clients ready for a new game.

References

1. Mobile Bristol Application Framework. www.mobilebristol.com

2. Vogiazou et al (2004) *'You got tagged!': The city as a playground.* Accepted for the Second International Appliance Design Conference, HP Labs, Bristol, May 11-13 (Also available as KMi-TR-139, Knowledge Media Institute, The Open University, Milton Keynes, UK, 2004, http://kmi.open.ac.uk/publications/techreports-text.cfm)