Blue Danger: Live action gaming over Bluetooth

L. Wang, E. De Vial and L. Tokarchuk
Queen Mary, University of London,
School of Electronic Eng. and Computer Science,
Mile End Road, London E1 4 NS England

Y. Wang and A. Ma
MPI-QMUL Information Systems Research Centre,
Macao Polytechnic Institute,
Rua de Luis Gonzaga Gomez, Macao

***Abstract*—A community live action game over Bluetooth, namely Blue Danger, is proposed in this paper. The market penetration of Bluetooth provides an excellent media for such game in which a large number of players is required. Bluetooth has a relatively short transmission range that allows a dynamic mobile adhoc network environment to be created which is ideal for the action game proposed. In addition, reliability of Bluetooth as well as its detection and tracking have been examined during the development of the game. A prototype game framework has been built and tested.**

***Index Terms*— Bluetooth, Games and Mobile ad hoc networks.**

I. Introduction

As the penetration of mobile devices continues to grow, the capability of the handsets themselves also increases. Mobile phones have become more and more an integral part of society; far from just having the capability of receiving phone calls, mobiles devices are capable of a range of activities from checking your email, surfing the web, playing games, etc.

Bluetooth wireless technology provides short range communications which has been widely used. Although the range of Bluetooth is relatively short, the extremely high market penetration among mobile users provides an excellent opportunity in facilitating a community game environment with close to no extra costs to the users [1]. This type of environment provides a basic platform for the Blue Danger framework proposed in this paper.

In addition, the tentative mobile ad hoc network nature created by Bluetooth is ideal for generating a dynamic environment for such a live action game. As the Bluetooth connection status is dependent on factors such as signal strength, distance and physical barriers such as furniture and walls [2], using this framework allows forces the players to be intimately involved in both the physical world and the game world. For example, the success or failure of an action taken during the game is directly dependant on the location of the at least two players.

Blue Danger originated from the live-action role-playing game Assassin (also known as Gotcha, Paranoia, Killer, Tag, Elimination or Circle of Death), where the object of the game is to assassinate all the other players by a variety of soft methods. Elimination of targets is done by fake weapons such as water-pistols, plastic knife, pretend poison, etc. This game is typically very popular across university campuses and online guilds can be found all around the world. The game has appeared in a variety of different movies and TV shows including the Fresh Prince of Bel-Air, the Assassination game and many others. Some universities in the UK, such as Cambridge and Durham, even have published their own assassin guide [3][4].

Existing Bluetooth multiplayer games, such as Fatal Force: Earth Assault, Pat Cash Pro Tennis [5] and Pacman [6], operate in the basis that players download the game onto their mobile devices. Players can either choose the single player mode or multiplayer mode. The later uses Bluetooth to connect to other players, and depending on the game, players either play with or against each other. The Bluetooth ad hoc environment created in these types of games is relatively static in the sense that players are not expected to be moving around. There is therefore, few associated networking issues provided all the players are within the Bluetooth range.

In Blue Danger, the Bluetooth ad hoc environment is very different. Since the game environment is very dynamic, players are expected to be moving around to search for their target or hide from their assassin. There are also issues with point-to-point and point-to-multipoint connections. As a result, networking issues such as connection established time, processing time, data rate will all affect the running of this game.

This paper describes a novel framework for implementing a "search and destroy" type game that utilizes mobile devices and specifically, the Bluetooth capabilities of those devices. For the sake of simplicity and to aid in developing a coherent framework, the initial framework implemented uses only a basic subset of the original Assassin rules. It is intended that this framework will be able to serve as a viable substitute for the original game (without breaking any "antisocial behavior" laws), a platform to help teach Java by encouraging students to create their own weapon systems which can be easily incorporated into the original platform and a platform for further study of adhoc Bluetooth networks. Reliability and interoperability of Bluetooth have also been examined when developing the proposed game infrastructure. Furthermore, while location tracking is fairly sophisticated in WLAN [6][7], tracking capability in Bluetooth is still in infancy. Tracking capability in Bluetooth has been studied, and technical

978-1-4244-2309-5/09/$25.00 ©2009 IEEE

challenges involved are also discussed in this paper.

II. System Architecture

A. System Overview

Fig. 1 illustrates the overall system design of the live action game via Bluetooth in which, the different functions that are available in the game framework are shown. A database is used to store information related to players, individual games, weapons and defenses. This information is stored and retrieved via a server which allows players to register, join and leave a game. The server can be accessed via WiFi or LAN during registration and via GPRS for starting and ending a game.

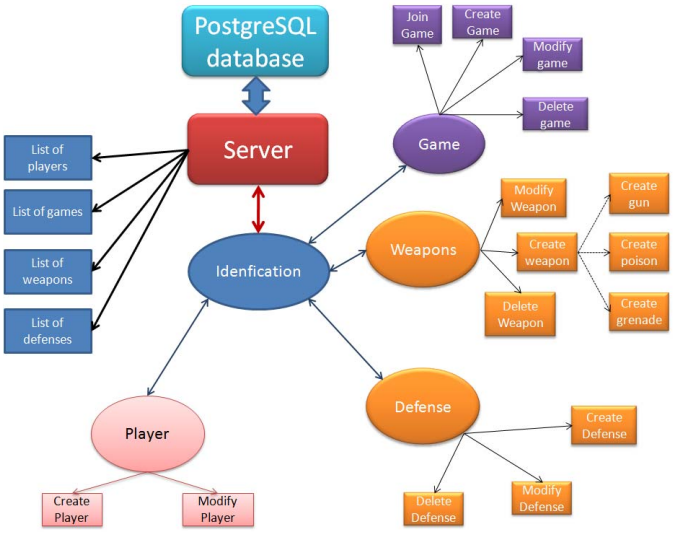

Fig. 1 System architecture

The game system is supported by GPRS and Bluetooth adhoc networks, see Fig. 2. To the best of our knowledge only one other game has combined this mix of technologies to create a mobile multiplayer game [8]. This framework however is novel in that the design of the game uses only Bluetooth for player interaction. In fact the success of the player is dependent on the Bluetooth interactions and thus, the player's physical interaction with the real world. Each player has to register their terminal's Bluetooth ID through the web and can join their game by connecting to the server via GPRS initially. During this phase, the server will send the game profile to each assassin such as start time, assassination target, allowable weapons and defenses. From registration point onwards, all players are allowed to purchase weapons and defenses. As the terminal has a list of the available weapons and costs, along with the player's available money, these interactions are simply logged to the phone until the player reports back at the end of the game (for them). This logging of game interactions locally on each individual terminal reduces data traffic and helps keep game costs to a minimum. Interactions among players are based on Bluetooth, which means that all attacks and other forms of communication are carried out using Bluetooth.

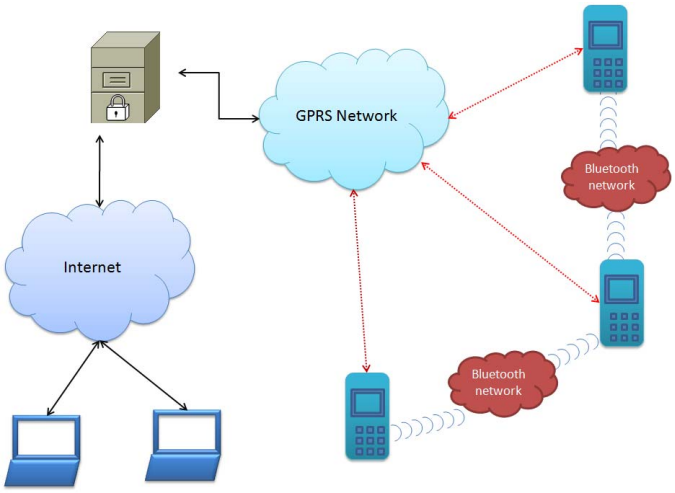

Fig. 2 Game environment infrastructure

B. Assassin Rules

Players are required to register using a Web interface to provide their Bluetooth address. Installation of the game on the mobile terminal is required. A library of default images for weapons and defenses will also be uploaded during installation.

Each player will be allocated a target in each game. As well as being as assassin, each player is also somebody else's target. The objective of the game is to remain alive and the sole survivor will be the winner of the game.

A game is set to have a minimum of 30 players which begins as soon as there is enough number of assassins have joined up, which is done by connecting the mobile terminal to the server via GPRS. Assassins will be notified by the server prior to the start of the game, see Fig. 3. This allows players to join a game at the start anywhere they like.

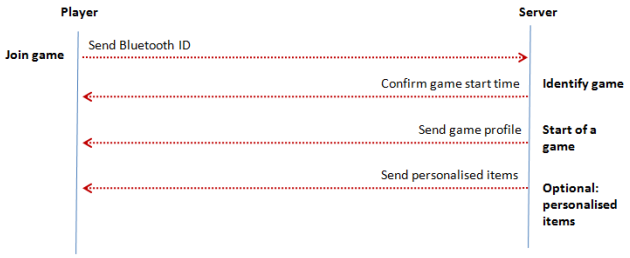

Fig. 3 Join up with the server via GPRS

After a successful attack, the target's energy will be reduced according to the strength of the weapon being used by the assassin. When the target has consumed all his or her energy, a successful assassination has taken place and the target is killed.

Once a successful assassination has taken place, the assassin will adopt the target of the victim. The assassin will be notified details of the new target. Fig. 4 and Fig. 5 illustrate the

allocation at the start of a game and the reallocation of a new target after an assassination has been carried out.

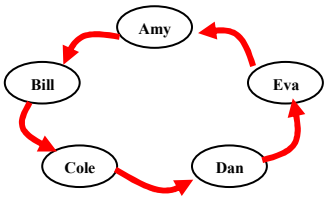

Fig. 4 Target allocation at the start of a game.

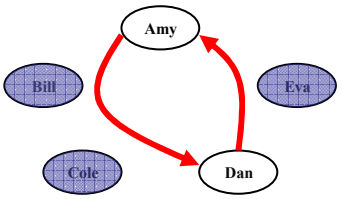

Fig. 5 Bill has been assassinated by Amy, she then picks up Bill's target who is Dan. Dan assassinated Eva, Amy is now Dan's target. Whoever survives in the end is the winner of the game.

A game is finished when there is only a sole survivor, and he or she will receive the overall prize. In the event of multiple survivors at the end of the game duration, the overall prize is equally shared among all the survivors. The overall prize is predefined by the server before the start of the game.

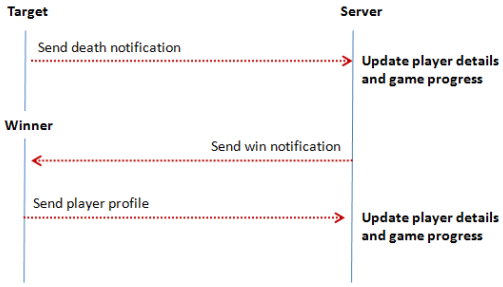

Fig. 6 Communication during the game with the server via GPRS

The server keeps track of the progress of each game by referring to death notification it receives after each assassination, see Fig. 6. When the game finishes, either when there is a sole survivor or game time out, it will publish statistics related to the game and rewards winners. The winner of the game will be entered to an overall league table. The league table is ranked according to the total number of wins.

One feature of the action game is to allow the assassin to savage the remaining credits from the victim. It will take a minute to savage a randomly selected item from the victim. This also means that the assassin is vulnerable to attack during this time. If the assassin escapes from attack half way during a savage, he or she loses their savage opportunity.

Another feature of the game is that assassins can trade their weapons and defenses among peers in the game. This function is carried out by Bluetooth communication between two players. Any unused weapons and defenses can be traded in for credit.

There is a player minimum daily vulnerable time. The idea is to avoid cheating such as hiding during the duration of the game. In the event if a player no longer wishes to take part in a game, he or she is required to make a drop out request to notify the server. His or her assassin will be reassigned his or her target.

C. *Use of Bluetooth*

While the game does make use of GPRS to initiate and terminate game play, Bluetooth has been chosen as the medium over which game play occurs. This choice was motivated by a variety of factors: Firstly, the desire to keep data costs to a minimum and even allow for future releases of the game to work on terminals without GPRS (utilizing SMS for initialization and termination). Secondly, Bluetooth provides a natural distance monitor. In order for any of the weapons to work, both assassin and target must be within Bluetooth range of each other for the entire duration of the weapon detonation. Also the short transmission range provided by Bluetooth means that any attack initiated by an assassin can be potentially missed, providing players with a reality touch.

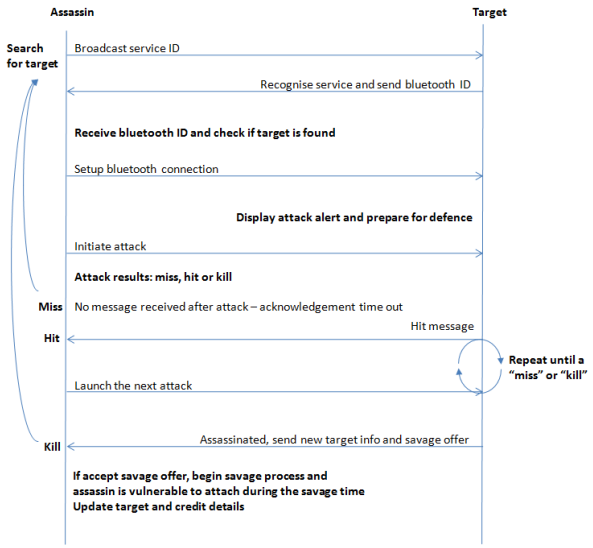

Fig. 7 Assassination via Bluetooth

Fig. 7 depicts a typical interaction diagram between the assassin and target. The assassin's terminal queries its potential target first by checking to see if it is playing Blue Danger by checking the service ID. After confirmation that the two terminals are playing the game, the assassin checks to see if the other terminal is its target. If it is the target, an attack available alert is sent to the assassin and a warning sent to the target. After deployment of a weapon, the assassin will either receive a miss, hit or kill notification, see Fig. 8. A miss is assumed if the target manages to physically get out of Bluetooth range before the Bluetooth attack message is received. A hit message indicates that while the attack was successful, it was not enough to terminate the target. Finally a kill message will indicate to the assassin's terminal that they have been successful and will also contain the next target (the old Target's Target) and the potential salvage booty. The salvage booty is available to the assassin if they opt to salvage, making them temporarily doubly vulnerable to attack.

Furthermore, sending of the kill message will also initiate the assassinated Target to report back to the server via GPRS to declare themselves out.

Fig. 8 Example game GUI

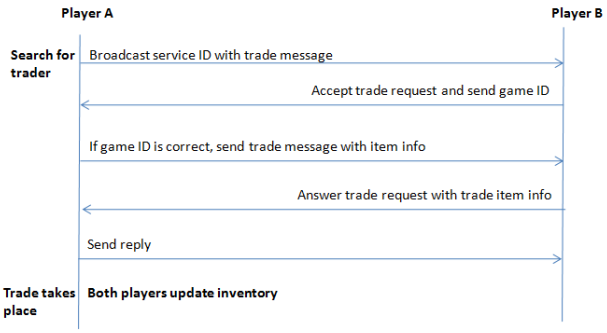

Fig. 9 Trading via Bluetooth

Trading between any two assassins in the same game is carried out by broadcasting a trade message together with the service ID using Bluetooth. A potential trader who in interested in trading will reply with his/her game ID. Once the game ID is confirmed, both traders will send the information of their trading item. If both traders agree on the proposed trading items, trade takes place, and players' inventory will then be updated accordingly, see Fig. 9.

D. Weapons and Defense Sub-system

The weapon and defense sub-system has been designed in order to allow new variations of weapons/defenses to be incorporated into the system at run-time. In order to easily provide for this behavior the sub-system was designed using the decorator design pattern [5]. The decorator design pattern allows us to extend the functionality of the sub-systems without requiring modification to the existing code base.

Fig. 10 gives a simplified schema for the weapon sub-system. Two abstracts classes have been created: Weapon and WeaponDecorator: The concrete weapons classes provide basic actual weapons, such as gun or grenade. These weapons fully function on their own, however players can create specialized weapons with added zing through the website by adding one or more WeaponDecorators to a base weapon. The WeaponDecorator classes have the same supertype as the Concrete weapon classes while having the ability to add behavior either before or after delegating the original object (such as a Gun or Grenade) to do the rest of the job.

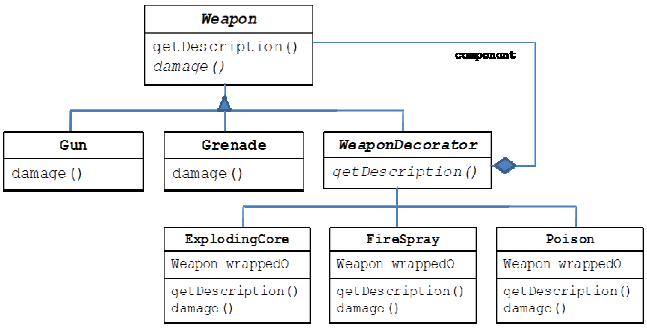

Fig. 10 Decorator Patter applied to the Weapon Sub-System

To create a weapon on the website, the player simply clicks on the weapon's section. They can then choose a base weapon, assign it a unique name and then choose whether or not to decorate it with additional capacities. For example, the player may choose to create a new type of Grenade, with lots and lots of FireSpray (x4). To create this weapon the server will track how many decorators are added to the base weapon class and then dynamically construct the call needed to create this weapon. Each decorator has different features and will impact the cost, duration, and damage done by the resulting weapon. All these calculations are defined in the concrete WeaponDecorator.

While the game is perfectly operational as is, players are encouraged to write their own specialized weapon decorators and/or weapons which can be added to the system dynamically at runtime (dependant on local game rules). Writing a Weapon sub-class or WeaponDecorator will allow new, highly specialized behavior to help individualize games. The final decision about what types of weapons will be allowed is down to the game organizer.

The same type of system has also been implemented for the Defense sub-system. It also provides web-creation and allows for the addition of new defenses and defense decorators.

III. Bluetooth Connection Issues

A. Too much delay

Bluetooth connections need to be set up between users, networking issues such as time required to establish a connection, one-to-many Bluetooth connection and reliability were investigated. Preliminary results have shown that the time required establishing a connection depends very much on the model of the mobile device.

For instance, when a Nokia N73 and Nokia 6300 were being used, it took over 30 seconds for the "assassin" to discover the "target" was in range. On the other hand, it only took a few second for a connection to establish between a Nokia N70 and

Nokia 3230. One of the reasons which causes the difference in connection time is the different versions of J2ME installed on the different devices.

B. *Connection request for Bluetooth*

When testing Blue Danger between Nokia N73 and Nokia 6300, a connection request was sent from the "assassin" to the "target" before a connection was established. This phenomenon also found on some other phones, but not all. With the presence of such request, effectively, the "target" can simply reject such request and prevent any attack to be taken place.

This again is caused by the different versions of Java Bluetooth implementations among mobile phones. One of the solutions to get round this problem is to disable the warning notification using additional code. Alternatively, other Bluetooth connection methods can be applied to get around this problem.

IV. Conclusion and Future Work

This paper has presented a novel design of a dynamic Bluetooth live action game, Blue Danger. This game will serve three purposes: Firstly, the dynamic nature of Bluetooth adhoc networks is studied during the game platform development. Related network reliability and position tracking are also being investigated. Results will be used to fine tune the overall game.

Secondly, software classes defined for this game are efficient and they will be exploited to help teaching topics such as inheritance, abstract classes and design patterns at both undergraduate and postgraduate levels. It is intended that Blue Danger will provide a fun platform through which the students will be able to see tangible and almost immediate results that they can upload to their own phones. Students will not only be testing the game but will be inspired to develop their own personalized items such as weapons and make them available as part of the framework.

Lastly, Blue Danger will provide a viable and legal alternative for the popular live action version. For the further study of adhoc Bluetooth networks, we have identified two primary targets for future study and implementation:

1) *Range detection:*

In wireless communications, *Radio Signal Strength Indicator* (RSSI) is a parameter can be extract from PHY later represents the received signal level from the transmitter device. Bluetooth also has such a parameter. In Blue Danger, RSSI can be used to calculate the approximate distance between two Bluetooth devices. Once an estimated distance is obtainable, a more sophisticated weapon system can be designed so that the damage caused by a weapon can be dependent on the distance an attack originated from. For instance, a bomb can cause more damage to a target when the attacker is closer to the target device.

The current Java ME Bluetooth implementation JSR 82 does not provide methods to extract the RSSI readings. Work is in progress to extend the JSR 82 with supporting RSSI readings [11], but until a new Java Bluetooth standard is made available, JNI and a native interface of the Bluetooth driver will be used.

2) *Position tracking*

The later development of the game will support mobile phone positioning using Bluetooth. Players will be able to check his own and other players approximate positions in the playing area (e.g. Campus wide). In order to locate players, Bluetooth dongles will be placed in different places across the playing area (e.g. in different classroom). These Bluetooth dongles are discoverable so that mobile phones are able to sense the surrounding dongles. As those dongles' locations are known, if a mobile phone can sense one or more surrounding dongles, it will know its proximate location.

More accurate position can be achieved using RSSI readings from nearby Bluetooth dongles, as the RSSI is directly related to the distance from the Bluetooth dongle and sensor (mobile phone). If more than three Bluetooth dongles can be sensed and RSSI readings can be extracted, a mobile phones location can be achieved using triangulations methods by the approximate distances to the surrounding Bluetooth dongles.

References

[1] M. Powers, "Mobile Multiplayer Gaming, Part 3: Multiplayer Games with Bluetooth", 2007 [online]. Available: http://developers.sun.com/mobility/midp/articles/gamepart3/

[2] M. Andersson, "Using Bluetooth in an Industrial Environment, Reliability and Robustness", White paper, 2006. Available: http://www.connectblue.com/fileadmin/Connectblue/Web2006/Documents/White_papers/Bluetooth_Reliability.pdf

[3] Cambridge University Assassins' Guide [online]. Available : http://www.srcf.ucam.org/assassins/

[4] Durham University Assassins Society [online]. Available: http://www.dur.ac.uk/assassins.society/

[5] http://www.gamezone.com/news/09_09_04_11_21AM.htm

[6] Mumtaz, A.; Faisal Tehseen Shah, "Games over Bluetooth," Computer Systems and Applications, 2006. IEEE International Conference on. , vol., no., pp. 1167-1170, March 8, 2006 Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1618503&isnumber=33913

[7] N. Koutsouris, N. Mitrou, Y. wang and A. Ma, "An agent-based system for supporting enhanced services over wireless infrastructures", CCCT'04 Texas, USA

[8] The Micro Monster Game. Available: http://www.micro-monster.com/

[9] Y. Wang, L. Cuthbert, F. J. Mullany, P. Stathopoulos, V. Tountopoulos, D. A. Sotiriou, N. Mitrou and M. Senis, "Exploring agent-based wireless business models and decision support applications in an airport environment", Journal of Telecommunications and Information Technology Mar 2004

[10] S.J. Metsker, W. C. Wake, "Design Patterns in Java", 2nd Edition Addison-Wesley, 2006, pg 287-304.

[11] C. d. Flora, M. Ficco, S. Russo, and V. Vecchio, "Indoor and outdoor location based services for portable wireless devices," in the 25th IEEE International Conference on Distributed Computing Systems Workshops, 2005.