REPORT

Background information.

Location based services has recently emerged as a new tool for information sharing and opened up another dimension in the field of mobile computing. With the advancement of GPS-enabled smart phone devices and operating systems such as iPhone OS and Android, geographical data are now being utilized to enrich existing services and deliver customized and location-aware data to end-users. For example, search engines are incorporating location information to provide better ranking and filtering of search results. Social networking applications, such as Twitter, are providing new functionalities that enable the distribution of local topics on a global scale in real time ("local trends") [2]. The infrastructure upon which these services are built, namely the Internet and its geo-coding & reverse geo-coding API, has lead to the notion of "Geo Mobile Web" [1], in which information is identified, tagged, and shared within a location context. The application of location-based services also has other great potentials, for example, in building emergency networks in response to location-relevant events such as earthquake.

1. What were the research goals?

In this project, we approached the study of location-based services via prototyping a multiplayer game system that enables end-users to play in real-time within a location-aware context. The objective is to identify both technical & non-technical requirements of building a location-based system, the potential challenges in delivering location-based data to mobile end-users, and the feasibility of a pervasive game product on top of location-based services.

2. Goals achieved and the techniques applied.

We built a prototype system ("iConquer"), which is a location-based multiplayer real-time strategy (LBMRTS) game. In iConquer, player can move in a city map and claim territory blocks by dropping agent bots called "Swarm". Swarms can attack/occupy a territory ("Swarm Site") and defend it on behalf of the player. Each player can carry a certain amount of swarms in his "Suitcase" and release them to conquer new sites. The suitcase is refilled automatically every day at a rate adjusted by the number of territories player claimed. Players can also capture "rogue" swarms that respawned at arbitrary geo locations and move randomly across the map.

The game system adopts a Client/Server design architecture, where the client-side program is installed on mobile device and acts as the end-user interface for accessing the game. GPS coordinates are continuously sent back to the server, which is then used to deliver in-game context data to the player. The iConquer server maintains a persistent and consistent view of the game world to all players and is capable of handling concurrent client-side request at the same time.

Through the project, we identified a number of elements that are essential towards building a sound location-based system. Firstly, data network and GPS signals on the end-user side are unreliable and error-prone: network jitters and connection losses may occur frequently, and GPS signals may have slow warn-up time and delays. This suggests that a location-based system shall be designed to handle errors in a robust and responsive manner: For example, communication protocols shall incorporate error message body and provide appropriate feedbacks whenever possible; the system shall also minimize the client-side waiting timeout period in order to reduce user frustration. Furthermore, location based services shall be able to adjust itself dynamically according to GPS accuracy. For example, by turning off certain service modules or disabling certain in-game actions that require accurate client-side GPS fix.

Scalability is also identified as a key element for location-based system design. Any location-enabled service inherently operates on a global scale: client request may come from any valid geo-location within the coordinate range (latitude: [-90, +90], longitude: [-180, +180]). This presents challenges for server-side design, in particular the requirement for an efficient data storage and search scheme. Furthermore, end-user mobile devices often have limited computing power, bandwidth, and disk storage. Therefore the delivery of location-context data must be optimized without degrading end-user experiences. In iConquer, we experimented the notion of "Fog of War", where a location window is associated with each player such that only context fall within the window will be delivered to the player. These contexts include, for example, nearby swarms, territories, point of interests, etc. In the game, we also summarize information in a hierarchical manner. For example, details of in-game objects are not delivered to the player until their geo-distance is close enough. This effectively reduces the overall data transfer rate. Other techniques have been tried as well, for example, local caching of game data on client-side device, timestamp tagging for incremental data synchronization between client and server, etc.

Information security presents another challenge in the design. Raw data stream may expose sensitive information such as player's current location and password over the network, therefore leading to potential security concerns such as data packet "eavesdropping" and location data mining/tracking. In iConquer, we imposed several design-level constraints to address the issues. For example, iConquer server will only record player's most recent location information, and does not broadcast private information to other players. We also adopted transportation layer security protocol (SSL) that encrypts data stream sent between the client and iConquer server. We also investigated Public Key Infrastructure (PKI) in protocol design utilizing the OpenSSL toolkit [3], which effectively reduces the risk of common attacks such as "Man in the Middle" (MITM).

In a location-based service, Point of Interest (POI) serves as the basic building blocks in the creation and expansion of location-aware context. Geocoding and reverse-geocoding service, on the other hand, only translates between geo-coordinates and real-world address (street address, zip code) and does not suffice the task of context awareness. Therefore, building an accurate and content-rich POI database becomes a prerequisite in a location-based system. In this project, we case studied several industrial applications [4][5] and also experimented alternative ways of constructing such a database. We found that a user-driven (user-innovation) model is widely adopted, where individuals can propose, edit and contribute to POI content in a peer-reviewed manner. Established systems are also providing public APIs [6][7][8] via, for example, RESTful web service, to encourage content sharing and collaboration within location-based development community.

In this project, we also evaluated the game prototype from the perspective of pervasive game design. We found that location-based service enhanced the gameplay by providing an augmented reality context that introduces new elements in both concepts and mechanism. Spatial and time constraints, for example, could be embedded to achieve gameplay balance. In iConquer, players can manage their territories only if they are physically present on site; "rogue" swarms will move towards arbitrary directions at fixed time intervals therefore increases the difficulty of being captured. Aside from location-based context, we also experimented device-dependent functionality for the gameplay. iPhone accelerometer, for example, could be used to detect various user gestures and trigger corresponding in-game actions. We also tested iPhone GameKit to build an ad-hoc peer-to-peer network over blue tooth, for example, for trading swarms between players. A combination of these techniques could be blended together to deliver gameplay experience that is pervasive and unique for mobile platform.

Limitations, however, still exist on current mobile device and OS that prevents the development of complex distributed game system. iPhone, for example, does not allow background processes therefore players cannot receive real-time in-game notifications while running other iPhone applications. The limited computing power, battery life and data transfer cost also prevents the implementation of autonomous agents on client-side program. In iConquer, for example, we simulate and maintain the states of in-game NPCs (e.g. swarms) on the server and only deliver the content to

clients upon request. Screen size may also prevent the delivery of rich user interface, however the issue may be alleviated by future model updates (e.g. iPad).

3. Summary of the outcome of the research

To summarize, we designed and implemented a location-based game prototype and demonstrated that location-based services could be integrated to augment pervasive gameplay that surpass the traditional keyboard/monitor experience. We identified and evaluated a number of requirements and challenges that we believe are crucial towards building a sound location-based system. We also addressed limitations on existing mobile platforms and the constraints they imposed on system design.

4. What are the future research plans?

Seamful design provides many insights into developing location-based mobile games that are both robust and streamlined. It is worth investigating how various seamful strategies could be utilized to tackle the current challenges. Established location-based systems often span across multiple platforms, including iPhone, Android, WinCE, BlackBerry, and Palm OS etc. It is therefore important to design service interfaces that deliver location-aware context in a ubiquitous way. Furthermore, public location-based APIs are being recognized as an emerging collaborative development tool and business model as well. Therefore it is also worth further exploration.

5. References:

[1] Geo Mobile Web Powered by Google Maps http://googlegeodevelopers.blogspot.com
[2] Twitter Local Trends. http://blog.twitter.com/2010/01/now-trending-local-trends.html
[3] OpenSSL – Cryptography and SSL/TLS Toolkit. http://www.openssl.org/
[4] FourSquare http://foursquare.com/
[5] Gowalla http://gowalla.com/
[6] Yahoo Local Search Web Services http://developer.yahoo.com/search/local/V3/localSearch.html
[7] FourSquare API http://groups.google.com/group/foursquare-api
[8] Google Geo API http://code.google.com/more/#google-geo