Experiences from the Development of the Pervasive Game CityZombie

Meng Zhu, Alf Inge Wang and Øyvind Rolland
Norwegian University of Science and Technology (NTNU)
Sem Sælandsvei 7-9, NO-7491, Trondheim, Norway

ABSTRACT

Pervasive games employ pervasive technology like location-based techniques, utilize player context in real world, and thus produce high immersion. In this article we will introduce a pervasive game prototype CityZombie, which is a multiplayer strategic game we developed and tested in Trondheim, Norway.

The experience from developing and evaluating CityZombie is presented in this article, which covers both technical and gameplay aspects. CellID as a simple and popular technique shows high feasibility and appropriateness to the game, and social interaction plays a dominate role in attractiveness and enjoyment brought by CityZombie. Our research also reveals that strategic game could be a promising genre of pervasive game, due to its relatively low requirement to platform capabilities.

KEYWORDS

CityZombie, Pervasive Game, Location-based Game

1. INTRODUCTION

Handheld devices have shown sufficient computing capability, high mobility, wireless connectivity, and great popularity. This makes cell phones feasible and attractive as gaming devices. Thus mobile games have become common in the last decade. The development and evolution of location-based technology brings new possibilities to mobile gaming, which leads to appearance of Pervasive Games, and motivates research in this new game genre.

In this paper we will introduce a pervasive game, CityZombie, which is result of a student project at the Norwegian University of Science and Technology (NTNU). The game is a multiplayer networked game running on cell phones that utilizes the CellID API. CellID is supported by all cellular phone systems, where each of cell sites has a unique id that enables the system to locate a cellular user so that it can route calls to the correct cell (Rashid, Mullins et al. 2006). By mapping cells to geographical areas in development time, our game could position players at runtime. Pervasive game is an overloaded term. Although there seems to be a general hunch or understanding about the nature of the concept, a generally accepted definition has not yet been defined (Nieuwdorp 2007). We labelled CityZombie pervasive game, because it utilizes positioning techniques, uses real world as play ground and supports social gameplay.

The motivation of our research is to explore development of location based game application, especially focus on utilizing of CellID technique. Experience gained from developing CityZombie is summarized in this article, which is concrete and practice oriented contribution to this field.

The next section will give an introduction to CityZombie, both from game design and technical points of view. Further, Section 3 presents our experience, and Section 4 contains related work. The article is concluded with Section 5.

2. CITYZOMBIE

The pervasive game, CityZombie, is a prototype implemented by a Master student (the third author) in five months. It is a multiplayer networked pervasive game with typical client-server architecture. The server software runs on a PC with a Java SE platform installed, while client software requires mobile phones supporting Java ME, with MIDP 2.0 and JSR 179 (Location API for Java ME).

2.1 Game Design Overview

The background of CityZombie is set in urban areas, where players of the game are split into factions of humans and zombies. Both factions have the same goal for the game: taking control of the virtual world by entering *virtual zones*, clearing out enemies (players of rival faction) within that zone in order to take control of the zone. A captured zone could generate points for its owner team continuously when time elapses till it's taken over by the rival team. Points for zombies and humans team are named *fear* and *hope* respectively.

Virtual zones are the reflections of physical play areas in the game world, and a city with 2-200 play areas could be mapped to a game world in CityZombie with the same number of *virtual zones*. The *virtual zones* are represented in the game via a colour-filled map and the colour filling a zone indicates who controls that zone, or nobody (empty). The first player entering an empty zone by walking into the corresponding play area in real world will take control of that zone. The zone will then stay in hand of his faction until a rival player attacks the zone and successfully destroys the owner's *defence*.

Defence of a zone can be 1) Players of the zone-owner faction, or 2) a mine placed by the players. In the first case (i.e. there are players of the faction in control presenting when the attacker enters this zone), a one-one fight will start. Game server will gather information on both attacker and defender to simulate a battle between them. The clients then represent the battle round by round, in a turn-based manner. In the second case, a player walking on a mine will not capture the zone but will destroy the mine in cost of receiving *penalty of death* of the game, which will be elaborated in the next sub section.

The game is played as an open world where players come and go as they wish, never having to wait for a game session to start. The first player entering the game starts a game session, which lasts until one faction has achieved a preset limit of *fear* or *hope*. After that, a new session can be started -keeping the game pace high.

2.2 Gameplay Highlights

Firstly, putting the game in a game world combining reality and virtual scene produce enjoyment of pervasiveness supported by location techniques. This is the most important feature of the gameplay. Figure 1 shows how Trondheim (a city of Norway) is mapped to the game world, where important buildings like railway station, Nidaros Cathedral, etc. in the real city have virtual projections.

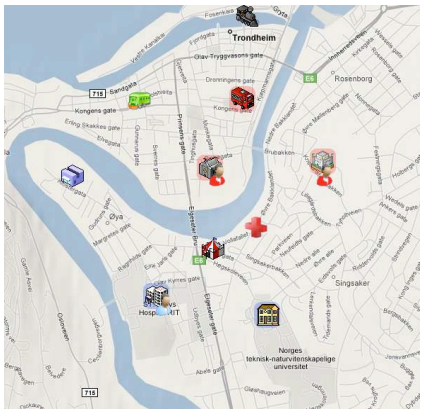


Figure 1. Mapping Trondheim to a game world

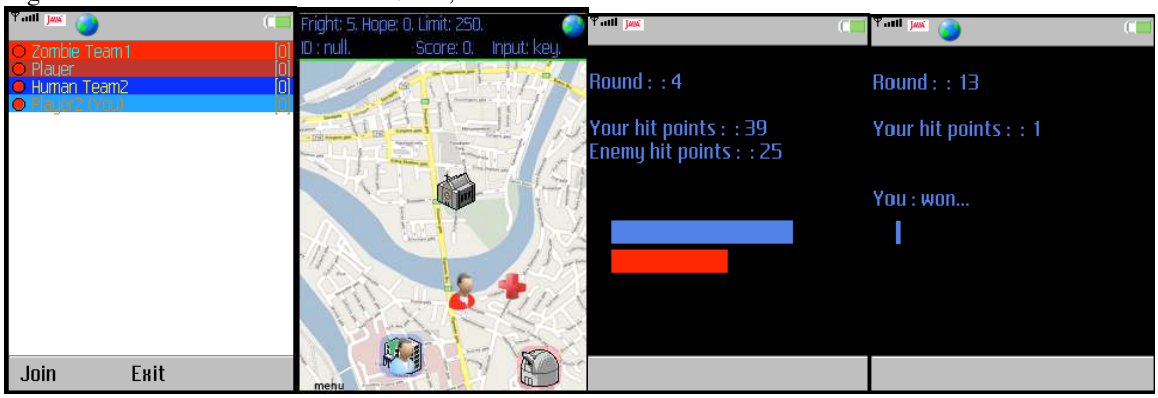
Secondly, cooperation and competition between randomly joined players produce enjoyment of sociality, which is another highlight of gameplay. Social is considered as an element of GameFlow(Sweetser and Wyeth 2005) model, which helps to understanding player enjoyment. CityZombie supports competition and cooperation both in virtual and real world. The goal of competition between teams can be 1) producing more *fear* or *hope* within a time limit, or 2) holding control of majority of the city for a certain amount of time. Supporting of PVP mode also enables competition between individual players. The cooperation among team members is various, e.g. two players form a defence team to secure a virtual zone, or more players form an attacker team to take a zone with a very strong enemy who has relatively rich hit-points.

Thirdly, the penalty of death in this game is novel. When a player is either defeated by his opponent or killed by a mine, he will receive the penalty of death: the player will "lose conscious" thus unable to attack in the game, until entering a neutral revival zone. This implies a compulsory physical movement in real city.

2.3 User Interface

Two user interfaces are presented by the game: 1) *Client UI* for players; and 2) *Server UI* for game administrator.

Client UI is rendered by the cell phone running the game, which consists of three major views: *Lobby View, Game View* and *Battle View*. Each view is used for a specific scenario of the game. *Lobby View* is presented when player enters the game or a game session ends. Player could then create a new game, or join an existing one. Figure 2.a shows a typical Lobby View on a Human player's ("player2") screen, where the game is created by another player ("player") who has chosen Zombie team. *Game View* is presented when player has entered the game, and is walking or running in the city, where a map of the city is shown with some virtual buildings indicating virtual zones in game. Each player has an icon representing his or her character, whose colour differs on faction. Figure 2.b shows a Game View with two characters and three virtual buildings. Some general information like *fear* value and *hope* value is also displayed on top of the window. *Battle View* is presented when player starts a PVP battle by entering a zone defended by rival player. This view shows current Hit Points (HP) of the player using this cell phone and the enemy player. The presentation of HP updates round by round, with both numbers and graphical HP bars. Figure 2.c and Figure 2.d are screenshots of Battle View, where both in-battle screen and battle-result-screen are shown.



a b c d

Figure 2. Screenshot of Client Display

Server UI is presented on the PC running server software, with which game session information is accessible to game administrator. The information includes IDs of the players, Score of teams, Time limit, Limit to total players, Support item limit, Individual score limit, and Team mode enabled. Server UI is also a configuration panel, but as a prototype, we only implemented function of changing support item limit, with a slide bar on bottom of the application window.

3. EXPERIENCES AND EVALUATION

3.1 Location Positioning and Use of Location Data

CellID is used to locate players in the game. An advantage of using CellID is avoiding extra battery consumption. Further the technical challenges are not demanding, since a simple line of code will retrieve your actual position. One shortcoming of CellID is the additional work of pre-mapping. CellID only provides information of what base station (cell site) the player is connected to, while what the game cares is the player's physical location. Manual mapping is required before and in development. The process of manual mapping could be made easier if connecting a GPS to the phone. It is possible to use available services like Cellspotting.com(Ferner) to do such mapping. They are insufficiently accurate and lack of boundary information. However, they do provide support for relating latitude and longitude to base station coverage.

To position the player, the game must provide different location mapping for different types of mobile network used. GSM base stations use a lower frequency signal with a higher intensity giving larger cells

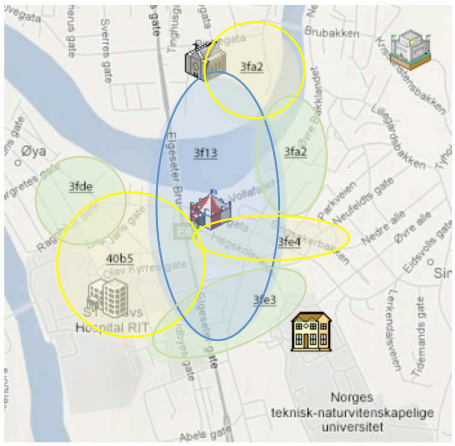


Figure 3. GSM base station coverage

compared to UMTS. To make it easy to map cells to physical positions and positions in the game, we implemented a mapping service, which supports mapping both GSM and UMTS zones. Figure 3 shows the various GSM base stations found in Trondheim related to the areas in the game. The game zones could not be smaller geographically than we had designed. From a practical point-of-view, the player has to move for some distance to switch her game zone to another. The distance from the university in the lower part of the map to the church on the centre upper part is about 1km (Figure 3). So it was a physical as well as a tactical challenge to play the game.

Making use of the geographical mapping of base stations is another challenge. Few areas can be mapped to real-world locations to help players to understand the game world. Those who do fit well with the street map are often overlapping with other cell areas. The difference between operator networks makes it even harder to make a game that fits a broad range of players. However, our testing in Trondheim proves that it is possible to find well-suited zones and structures.

3.2 Technical Measurements

Measurements are carried out to see the response time of the game, and how much data the game transfers. The tests were done with two SonyEricsson cell phones (K850i and W910i) along with a PC server.

We found the response time by measuring the time it takes for a data packet from the server, to the client and back. The test was performed both for GMS and UMTS networks and both when using the TCP and the UDP protocol. We performed several measurements and computed the average to get reliable values. It was proved that the response times are within acceptable limits for a location-aware game where the game zones are not too small (close). The response time for UDP is 250ms in UMTS and 300ms in GSM. With TCP, it's 700ms in UMTS and 750ms in GSM. We also measured how much data is required to be transferred between the clients and the server in a normal game session. The formation is useful for computing the price of play such a game with payment per megabyte. Table 1 shows the bytes transferred and the price connected. Even a 10 minutes game with four players costs less than 10 cents.

Table 1. Data transfer and game cost.

Type	Data transfer	Cost
Create game session	2528 bytes	$0.005
1 min idling in lobby	3220 bytes	$0.007
5 min game 2-players	18811 bytes	$0.040
10 min game 2 players	30330 bytes	$0.065
5 min game 4 players	17834 bytes	$0.038
10 min game 4 players	35148 bytes	$0.075

3.3 Testing the Game

A Survey before testing shows that players are generally excited to play the game. Further, Players that regarded themselves as *physically active above average were generally more excited* by the game premise than those who regarded themselves under average. The game was first tested on students using an emulator, and moving 500 meters is simulated by pressing a single key, which decreased the playability to some extent. This emulation testing was useful to test the main game mechanics before trying it out in the field.

The field test was conducted by equipping players with cell phones with the game installed along with an introduction on how to play the game. The first observation was that the players played much more carefully in real game compared to the emulator play session. Further, some players reported that they thought that the game was broken as the update of positions very slow and the screen was seldom updated (due to the size of the zones). Another issue was that some players lost track of their avatars. This was remedied by altering avatars to differ between fellow team members and their own player. Another feedback from some players was the limited support for communication between fellow players. Players are forced to shout to each other or simply deduce movement based on the display. In any game where you have to rely on fellow players for

success, limited communication support is an annoying factor. The players also occasionally experienced network problems that sometimes gave a very unresponsive experience. Players themselves seem relatively supportive of location-based gaming, mostly because it represents something new, but also because of the social aspect. Multiplayer games seem more appealing despite the network cost. Play-testing during the early stages of development were not so valuable when turned towards average players, at this stage experts or people with background in mobile development are much more helpful in providing concepts and ideas that are both challenging and practical. Furthermore, as developers, we were made very aware of the need for field-testing at an earlier stage. Half of the bugs and gameplay issues arose first when tried with actual players outside. This is very time consuming, more so because of the sluggish testing process with different key locations instead of any relative movement. Having a mobile development studio would definitely help out at this stage, keeping developers more in contact with actual play sessions.

4. RELATED WORK

Botfighters (Sotamaa 2002) is a location-based multiplayer combat game for cell phone. The game concept is a bit similar to CityZombie, where players send out a SMS to find other players (opponents) in vicinity, and then shoot the robot of the opponent through sending another SMS. The major difference in game design is that there is no "faction" concept in *Botfighters*, so players joining the game are in a melee combat. Competition constitutes the only social element of the game. Moreover, taking control of virtual zone brings more strategic gameplay to CityZombie than the pure kill-you-all design in *Botfighters*. CellID are used in both of the games, but the communication protocol in *Botfighters* is SMS, while TCP/UDP is used in CityZombie, which is more expensive but faster. *Undercover and Undercover2: Merc Wars* are Location-based MMORPGS, the technology used are similar (CellID, J2ME), but support greater social interaction: Player could visit real cities in 44 countries, in order to complete missions, look for friends, enemies and landmarks in real streets.(Rashid, Mullins et al. 2006) The major difference between Undercover serials and CityZombie is in game concept: As MMORPGs, undercover serials feature a slow progress and open-ended game session, while CityZombie as a strategic game follows a faster rhythm. A game session could be finished within one day, where the competition is more intense and tight, the physical effort is also higher. More CellID games include: *The Journey II*, *Songs of the North*, *I Like Frank*. They are reviewed in (Rashid, Mullins et al. 2006). Due to game genres, their gameplay differs more from CityZombie.

5. CONCLUSION

Pervasive games are promising for future of the game industry, and have become an attractive research field. We have explored pervasive game technology as well as game concept design with developing and evaluating a location-based game: CityZombie in Trondheim, then summarized our experience in this article. The prototyping and assessing shows that 1) CellID is easy for programming, but very time consuming to do pre-mapping; 2) Making use of location data is challenging; 3) Data transfer cost could be low, with a non-real-time gameplay. 4) Social aspect becomes a dominate factor to attract players in CityZombie, which reveals potential direction of successful pervasive game.

REFERENCES

Ferner, C. J. Cellspotting. Retrieved 26 Jun., 2010, from www.cellspotting.com.

Nieuwdorp, E., 2007. The pervasive discourse: an analysis. *In Comput. Entertain.*, Vol. 5, No. 2,pp 13.

Rashid, O., I. Mullins, et al, 2006. Extending cyberspace: location based games using cellular phones. *In Comput. Entertain.*, Vol. 4, No. 1,pp 4.

Sotamaa, O., 2002. All The World's A Botfi ghter Stage: Notes on Location-based Multi-User Gaming. *Computer Games and Digital Cultures Conference,2002*.

Sweetser, P. and P. Wyeth, 2005. GameFlow: a model for evaluating player enjoyment in games. *In ACM Computer Entertainment*, Vol. 3, No.3.