KTH Computer Science and Communication

Development Guidelines for Mobile Multiplayer Games

Bachelor of Science Thesis (Degree project, in Computer Science, first level, DD143X, 6 hp)

Supervisor: Vahid Mosavat

2013-02-27

Authors:

Mian, Badar

David bagares gata 6

111 38 Stockholm

+46 70 470 00 70

badar@kth.se

Thakrar, Nikhil

Professorsslingan 9

114 17 Stockholm

+46 70 026 42 66

thakrar@kth.se

Stockholm, Sweden
2013-04-12

Abstract

In the past decade the advances in mobile technology have produced new powerful generations of mobile devices that come with ever-increasing computing powers and high-speed wireless connection technologies. It is more attractive than ever to develop mobile multiplayer games and many developers are straining to bring more and more multiplayer games into the market. However, there are not many concrete guidelines or helping-tools for developers programming and designing them.

The purpose of this thesis is to investigate the relevant infrastructures of mobile multiplayer games and with the obtained results produce some development guidelines for the developers. Before the commencement of the investigation, the authors reviewed existing literatures using a structured review process. A search for relevant articles was conducted and the references used in the selected articles were searched in order to find additional relevant references for the investigation.

There are many different factors and technical aspects to consider when developing mobile multiplayer games. For example different game genres have different prerequisites and it also governs the degree of complexity of the game design. It is important to also consider the different mobile networks and the technical capabilities they encompass so that the developed game does not consume more bandwidth than is supported by the different mobile networks. The biggest hurdle to overcome when developing mobile multiplayer games is the issues regarding network latency. There exist different approaches for handling this and they are brought up in the thesis.

In the end of the thesis, guidelines are presented for facilitating the development of mobile multiplayer games.

Sammanfattning (in Swedish)

Under det senaste decenniet har stora tekniska framsteg skett inom mobil teknologi som har producerat nya kraftfulla generationer av mobiltelefoner. Det är mer attraktivt än någonsin att utveckla mobila multiplayer spel och många utvecklare kämpar med att få fler och fler multiplayer spel på marknaden. Tyvärr så finns det inte många konkreta riktlinjer eller hjälpverktyg för utvecklare som programmerar och designar dem.

Syftet med den här uppsatsen är att undersöka de relevanta infrastrukturerna för mobila multiplayer spel och med de erhållna resultaten producera utvecklingsriktlinjer för utvecklare. Innan påbörjandet av undersökningen utförde författarna en sökning efter relevanta artiklar och de referenser som användes i de valda artiklarna söktes för att finna ytterligare relevanta referenser för undersökningen.

Det finns många olika faktorer och tekniska aspekter att beakta när man utvecklar mobila multiplayer spel. Till exempel har olika spelgenrer olika förutsättningar och den valda genren har en inverkan på graden av komplexitet vid speldesignen. Det är viktigt att också beakta de olika mobilnät och de tekniska möjligheter som de besitter så att det utvecklade spelet inte förbrukar mer bandbredd än vad som stöds av de olika mobilnäten. Det största hindret att övervinna när man utvecklar mobila multiplayer spel är problem rörande nät latens. Det finns olika metoder för att hantera detta som tas upp i uppsatsen.

I slutet av uppsatsen presenteras utvecklingsriktlinjer som är till för att förenkla utvecklingen av mobila multiplayer spel.

Table of contents

1 Introduction

This thesis is written as part of the course DD143X Degree Project in Computer Science. It is given by the School of Computer Science and Communication at the Royal Institute of Technology in Stockholm, Sweden.

The thesis focuses on the relevant infrastructures and the technology used in mobile multiplayer games. An investigation is carried out by examining journals and with the obtained results some guidelines are presented with the target group of mobile multiplayer game developers in mind.

1.1 Background

In the past decade the advances in mobile technology have produced new generations of powerful mobile devices that come with ever-increasing computing powers and ground breaking graphics. Mobile games, which constitute the majority of mobile applications (Manweiler, Agarwal, Zhang, Choudhury, & Bahl, 2011, p. 71), have significantly increased in popularity the past few years and it is attracting millions of players/subscribers around the world. In fact, mobile games are dominantly present in the list of top applications downloaded in both Android and iOS markets. One fundamental difference between the mobile game market and the more traditional game markets (consoles and PCs) is the hitherto rarity of multiplayer games on mobile devices. The reason for this is among others the issues related to latency, packet loss, and low bandwidth (Wang, Jarret, & Sorteberg, 2009, p. 1). As the technology is advancing, however, this is rapidly changing and more and more multiplayer games are starting to appear.

1.2 Problem Statement

This thesis aims to answer the following question: What are the relevant infrastructures and designs of mobile multiplayer games and what are the factors that pave the way for success in terms of gaming experience and game quality.

1.3 Purpose

The purpose of this thesis is to investigate the relevant infrastructures of mobile multiplayer games and with the obtained results, produce some guidelines for developers designing and programming them. The purpose of the guidelines is to facilitate and to shed light on what is important and worth considering when developing mobile multiplayer games, as for example the considerable impact network latency has on the gaming experience, and how the prerequisites differs between different genres of multiplayer games. It is important to point out, however, that the guidelines are based on the findings of a theoretical investigation of journals and other existing literatures. It is not empirically tested and it will not cover all the different aspects of game development as there are too many.

1.4 Definitions

MMORPG:

An acronym for Massively Multiplayer Online Role-playing Game. This game type falls under the category of real-time multiplayer games (for further information, see section 2.2)

FPS:

An acronym for first-person shooter, also a real-time multiplayer game (for further information see section 2.2)

3G:

The third generation of wireless network technology (for further information, see section 2.3)

4G:

The fourth generation of wireless network technology (for further information, see section 2.3)

WLAN:

An acronym for wireless local-area network. It is used for wirelessly connecting two or more devices and it typically also provides a connection to the Internet.

Latency:

The delay in time between the sending of a request and the receiving of the response.

Bandwidth:

The amount of data that can be transmitted from point A to point B in a given time period

bps (kbps, mbps):

Bits per second (1000 bits per second, one million bits per second)

Client/Server model:

An approach to computer networking. The server stores resources such as files and databases and clients initiate contact with the server to make use of its resources.

Peer-to-Peer model:

A model for computer network programming where each unit can act as both a client and a server for the other units in the network.

2 Background

A theoretical investigation of the relevant infrastructures of mobile multiplayer games was carried out before the writing of the thesis. The authors reviewed existing literature of mobile multiplayer games and of the relevant infrastructures. In the reviewing of literature, the approach of a structured review process was taken. Firstly, a search for abstracts of articles was conducted with the keywords: "mobile multiplayer games", "mobile game infrastructures", and "mobile games latency" on the online database KTH Primo. Secondly, a filtration was made to discard the articles that held no or little relevance to the purpose of this thesis. Thirdly, the references used by the selected articles were searched in order to find additional references. Lastly, the body copies of the selected articles were reviewed.

The results of the investigation are presented in the following subsections with a preceding clarification of what is meant by the designation 'infrastructure'.

2.1 Definition of Infrastructure

The meaning of infrastructure in the context of this thesis is the technology behind the multiplayer games and the factors that affect and govern the gaming experience. A list of infrastructures, which was included in the investigation, is presented below:

- Mobile resources (battery, display, CPU etc)
- Power management, load balancing, consistency management, latency preservation approaches
- Scalability
- Mobile networks (2G, 3G, 4G, WLAN, Bluetooth), network protocols (UDP or TCP)
- Bandwidth and latency
- Distributed topologies (Client-Server, Peer-to-Peer)
- Game designing

2.2 Different Types of Multiplayer Games

In the literature there are many different names and genres given to mobile multiplayer games. In this section the different genres are explained and elaborated.

Michael Powers discusses in his article that there has been an on-going evolution of successful mobile multiplayer games in the game market the past recent years. The first networked interactivity that appeared, added some additional community features to already existing single player games. These features included high-score lists, tournaments, and chats that brought with them the primitive aspects of contemporary multiplayer games. Players could for instance compare their high scores amongst each other and make it into a competition of who can score the highest points. This type of multiplayer games is appropriately addressed as community-based competitive games (Powers, 2006).

Next in line followed turn-based games. Turn-based, as the name suggests, means that the players have to wait for their turn in order to play. A typical example is board games such as chess, where one waits for the other player to take his turn. These games are typically also

called slow-paced games and they do not require as much interactivity as say action games. This category of games is quite popular and it is easy to implement from a developer's perspective as the mobile network communication designing is relatively easy. Turn-based games consume little bandwidth, owing a debt of gratitude to the fact that the player's pause to think through his next move, is generally greater than the measured network latency (Powers, 2006).

Today, the development of real-time multiplayer games is becoming increasingly popular and companies around the world are straining to bring more games of this type into the market. Real-time games, also known as event-based games, are far more interactive than its counterpart: turn-based games, as each player is free to play independently and the game is played in real time. Everything on the scene and all the interactive game play are rendered in real time by the game engine. The communication design is considerably more complex and because more information is transmitted, a greater bandwidth is required. It is typically also required that the network latency is low due to the need of fast responses (Powers, 2006).

2.2.1 Online and Offline Multiplayer Games

This thesis puts a great emphasis on online multiplayer games. Online-games require the users to connect to a central server in order to play against other users by using some mobile network, as for example 3G. In offline-games, players can use either Bluetooth or WLAN to create a local area network with other players, although with the restriction that the players are within proximity of 10 metres when using Bluetooth (Soh & Tan, 2008, p. 36).

Below follows descriptions of concrete genres falling under the categories already discussed above.

2.2.2 MMORPG

MMORPG is an acronym for Massively Multiplayer Online Role-Playing Game. This is a genre where a big population of players come together to play interactively in real-time within a virtual game world. Each player typically adopts a character that they have control over. One typical characteristic of this genre is that the game world continues to exist and evolve even when the players are offline. The most renowned and prominent game of the MMORPG genre is World of Warcraft.

MMORPG is quickly establishing itself as a popular form of entertainment. However, building the infrastructure required for these games is a non-trivial undertake.

2.2.3 FPS

FPS is an acronym for first-person shooter. All of the games that fall under this genre have a first-person perspective, meaning that all the events take place looking through the eyes of the protagonist. Typically, FPS games centre around combats with guns and other artillery. A very suitable and prominent representative of this genre is the game Doom.

Both of the genres MMORPG and FPS have been proved to be the game types that require a lot of real-time player interaction and therefore, the game design gets far more complex and a

greater demand of cutting-edge technology exists. (Bhojan, Akhihebbal, Chan, & Balan, 2012, p. 22)

2.3 Mobile Networks

Wireless technology is advancing very fast. In this section an overview is given over the different mobile networks and their associated advantages and drawbacks.

2.3.1 Bluetooth

For offline multiplayer games, Bluetooth is generally used. The advantages are the low cost, the availability, and its simplicity. The connections can be made seamlessly without having to install any extra drives. It was originally designed for replacing cables for connecting electronic devices with each other such as mobile phones or computers. Nowadays, however, it has become a wireless standard for the connecting of electronic devices to create so called 'Personal Area Networks' (PANs). The downside of Bluetooth is its short-range, the players would have to be within a proximity of 10 metres for the connection to establish successfully.

There exist two configurations for PANs: piconets and scatternets. Piconets can be thought of as client/server models. There can only be a total of 7 clients connected to a server at any time. The clients have no way to communicate and transmit data to each other, this task is given to the server. However, the possibility exists of changing roles between the devices whenever a client wishes to take over the server's role.

A scatternet is a set of two or more piconets. Any device in such a network can be a client in more than one piconet but it can be a server in only one. A device that is a client in more than one piconet, works as a gateway, forwarding data packages between the subnets. By letting a server of one piconet be a client of another, a construction of a peer-to-peer network is made possible. (Dideles, 2003, p. 11f)

2.3.2 WLAN

Another option when playing offline multiplayer games is to use WLAN. For some devices that do not support Bluetooth, WLAN is the sole option. Bluetooth is comparatively easy to use whereas WLAN can be a bit tricky when connecting to other devices. However, WLAN has a much wider coverage area than Bluetooth.

2.3.3 First generation, 1G

The first generation (1G) of mobile networks appeared in the1980s. The 1G network used analogue radio transmission techniques for traffic, which were almost exclusively voice calls. (Samukic, 1998)

2.3.4 Second generation, 2G

The 2G was first introduced in the beginning of the 1990s and it has much greater capacity in comparison with its predecessor, displaying better sound quality. The biggest contrast between 2G and 1G is that 2G uses digital radio transmission whereas 1G uses analogue radio signals. Furthermore, 2G supports data rates in the range of 9.6kbps and 14.4kbps. GSM is the

most prominent and widely used 2G system and it is an acronym for Global System for Mobile.

In the early 21st century a successor of the 2G network appeared in the market, the so called GPRS network also referred to as "2.5G". This network includes some additional upgraded advancements and features of the 2G networks. The data rate generally lies between 30 and 100kbps.

Not long after the arrival of GPRS, the system EDGE appeared and it is just as GPRS an advancement of the outdated GSM-system. EDGE is an acronym of Enhanced Data Rates for GSM Evolution. This system provides a greater data rate than GPRS of around 240kbps. Even EDGE is sometimes referred to as "2.5G" as it is in the crossing between the 2G and the 3G networks.

2.3.5 Third Generation, 3G

3G appeared around the same time as the "2.5G" systems as a trial version. It supports data rates of 2Mbps, which is considerably more than its predecessors (Yang, 2012, p. 344). UMTS (Universal Mobile Telecommunications System) is a standardisation of the third generation networks (3G) and is interchangeable with the term 3G in some contexts.

An enhanced 3G system is the so called HSDPA, High-Speed Downlink Packet Access also commonly known as "3.5G". It gives much higher data rates than UMTS, providing rates up to 14.4Mbps.

2.3.6 Fourth Generation, 4G

With data rates of 100mbps, the mobile network 4G takes the champion's position among the other cellular networks. On account of 4G's wide bandwidth, technologies such as fast mobile broadband are made possible. 4G technologies are usually marketed as LTE (Long-Term Evolution) in Europe.

All of the above mobile networks are employed for online multiplayer games.

2.4 Latency and Limited Bandwidth

High latency and limited bandwidth constitute the two main sources of network bottlenecks when using the above described mobile networks for online games. The following sections bring to light the issues and challenges associated with both network latency and limited bandwidth.

Latency is the measure of the data packets' time delay as they flow through the network. It is more specifically defined as the time between the sending of a request and the receiving of the response. Real-time games typically require low latencies, which means that a delay in response time could cause a degradation of the users' gaming experience. Games that can stand higher latencies do not get as affected by a noticeable delay and these are generally the games that do not require immediate responses, e.g. turn-based games (Yang, 2012, p. 348).

The biggest challenge mobile multiplayer game developers face today is the issues related to high network latency. High latencies affect the players' views and consequently lead to a high degree of inconsistency and unfair advantages to some players, thus removing the sought after element of fairness in games. The data packets sent between the server and the clients should not delay for too long in order for this problem to be evaded (Wang, Jarret, & Sorteberg, 2009, p. 1). Michael Powers (2006) writes in his article that there is nothing more that affects the player's gaming experience than the impact of latency.

Another hurdle to surmount is the issue regarding network bandwidth. Bandwidth can be defined as the amount of data that can be transmitted from point A to point B in a given time period. Usually mobile operators either charge a flat fee for unlimited network usage or charge their subscribers on the data volume used. It is in both the players' and the operators' interest that the data transmissions are kept to a minimum. This is something important game developers have to take into consideration when developing multiplayer games.

The different types of multiplayer games in the market have historically largely been developed with the inherent problem of high latency in mind. The two most common types have hitherto been turn-based games and other slow-paced games. According to Wang et al.(2009), both of these can stand round-trip delays of a few seconds without the gameplay being spoilt. In accordance with what was previously stated in the thesis, real-time multiplayer games are becoming increasingly popular and with them accompany the demands of lower response times.

2.4.1 The correlation between latency and different multiplayer game genres

To show the correlation between the performance and the gaming experience in multiplayer games and the different genres of the games with respect to network latency, this thesis presents a summary of a series of investigations of how latency affects the performance and the game play of first-person shooter-, strategy-, sport-, and car racing games.

Sheldon et al. (2003) share their findings of the impact of network latency on strategy games, stating that there is no perceptible effect on the game performance when the latency is increased in the interval of 0 to 3500 ms. However, when exploring i.e. orienting and locating other players on a game map, which is very typical for this specific type of multiplayer game, the correlation between latency and exploration time is more evident and sensitive. Latencies of around 800 ms could cause a degradation of the game experience and make the view erratic whilst exploring.

Pantel et al. (2002) conducted a similar investigation but of car racing games instead. The latency was increased with portions of 50 ms all the way up to 500 ms. for beginners and average players around 50 ms was enough to worsen the lap time. For the more advanced players, around 150 ms would cause the same effect as latterly described. The authors concluded that network latency of 50 ms is hardly noticeable, 100 ms is acceptable if no strict demands apply that the game should be realistic, 200 ms is strongly noticeable, and 400 ms or higher is not acceptable.

Henderson et al. share their findings of how network latency relates to first-person shooting games and it turns out that the impact of network latency is more perceivable on this particular genre (Henderson & Saleem, 2003). On online forums the authors have found statements of players stating that they generally cannot live with latencies of around 100 ms. The investigation's results concludes that players can live with 200 ms and that 300 ms would be the limit where most players would leave the game servers.

Nichols and Claypool (2004) have found that when investigating network latency in sport games, 500 ms or below were not enough to considerably affect the user performance. With latency higher than 750 ms, the players would report an erratic and unsmooth game view.

From the summary above, it can easily be concluded that mobile network games such as strategy games and sport games are not as sensitive to network latency as are first-person shooter games and car games.

Another criterion for sustaining the smoothness of the gameplay is to let all game events be updated regularly among the players. In what frequency the updates should be made is largely dependent on what game genre is being played. If it is a shooting or a racing game, a higher update rate is necessary in comparison to say strategy games (Wang, Jarret, & Sorteberg, 2009, p. 2).

2.5 Distributed Topology

There are two major computer network architectures for laying the groundwork for mobile multiplayer games: the client/server model and the peer-to-peer model. In the client/server model, a centralised game server is used and all the clients (i.e. players) log on to this server to start playing games with other clients. All the events made by the players are sent to the game server in so called action messages. The server processes the incoming action messages sequentially and then notifies the clients of the events' effects in so called state update messages. The communication only occurs between the game server and the clients, never between the clients directly.

The trend in the mobile game market thus far, has been to use client/server models for multiplayer games on mobile devices. This, however, is starting to change as more and more research is being done on this subject. The conservative client/server model is, in the light of some recent findings, not always the most suitable architecture for mobile multiplayer games. Rather a peer-to-peer approach with or without a centralised game server (to help match players with each other) for the game control could serve as a better option.

The potential danger of utilizing a client-server model is that depending on how popular the game is and also how small the bandwidth is, the server could act as a bottleneck as there could be hundreds of users playing the game.

How much of the game that is run on the server and how much that is run on the clients locally, could bear significant importance on how the gaming experience and the consistency of the game is perceived by the users. Ideally, the load on the mobile devices should be minimised (Wang, Jarret, & Sorteberg, 2009, p. 1).

One consequence of using the peer-to-peer model is that information will be transmitted between the mobile devices rather than having the server to forward messages. The downside of using this technique is that since the consumption of bandwidth stands in proportion to the amount of connections between mobile devices, there would be a great need of a wide bandwidth. According to Michael Powers (2006) each additional player causes the consumption of bandwidth to increase exponentially.

2.6 Approaches to Minimise the Impact of Latency and Limited Bandwidth

According to Wang et al. (2009, p. 3) the effects of latency in network games can be categorized in the following way: network efficiency and utilization, visual consistency, game world consistency, and fairness. Over time, several measures have been developed against the above stated vulnerabilities caused by network latency and in the following sections descriptions of these measures are provided.

2.6.1 Network Efficiency and Utilization

To attack the issues of network efficiency and utilization, there are some appropriate approaches game developers can use. Fritsch et al. (Fritsch, Ritter, & Schiller, 2006) talk about how the technique of Content Addressable Network (CAN) can be employed to better broadcast over the Internet. The biggest advantage of employing CAN, is that fewer packages are required in contrast to simple broadcasting.

For peer-to-peer multiplayer games, Zhu et al. (2007) present an approach where game state updates can be exchanged between neighbour nodes in an efficient manner. This shift coding approach is reported to have reduced network traffic and better utilize the network bandwidth. Another approach to make better use of bandwidth and to lower latency in network multiplayer games is to buffer game state messages. By bundling the user events and transmitting them in intervals, it lessens the burden on the network as opposed to sending each occurring user event in turn.

2.6.2 Visual Consistency

To overcome the challenge of visual inconsistency in online games, there is a widely used approach called the Dead Reckoning Technique [Wang et al. 2009 s.3]. This technique extrapolates an object's exact position by the knowledge of its last known position and velocity. A real-time mobile multiplayer game usually has many game objects such as vehicles, different creatures, and artillery under the control of the players. For all the players to locally get an updated game view of the different objects, it would require an enormous amount of information to be transmitted between the game server and the clients. According to Michael (2006), this would consume a lot of bandwidth and also increase the risk of being subject to network latency. By using the dead reckoning technique, however, the clients only need to supply changes to the velocities of the game objects under the client's control and the behaviour of the objects can be transmitted in a significantly smaller amount of data (Powers, 2006). The sacrifice to pay for using dead reckoning technique is that every client has to run

the extrapolation algorithm for each object in the game locally, which could cause problems on mobile devices with limited CPU and memory power. Additionally, there is little to reap from using dead reckoning technique if the game objects behave too randomly and unpredictably.

Another technique to minimise visual inconsistency is called the Movement Prediction. This approach specifically removes the phenomenon of warping, which occurs due to loss of network packages or low bandwidth. Missing network packages can cause moving objects to erratically bounce from one position to another and thus impairing the visual consistency. It goes without saying that this phenomenon is highly disliked by most game players.

2.6.3 Game World Consistency

Wang et al. (2009, p. 3) further state that there exist approaches for tackling issues regarding game world consistency as well. To clarify, consistency means that a shared and an equivalent game state exist across all clients. The authors mention an approach called the Bucket Synchronisation Mechanism. This method collects all game event messages and stores them in a bucket (hence the name) at the receiver. In accordance with a specified interval, all the messages in the bucket are processed and a local view of the global state is created. The bucket synchronisation mechanism is best suited for peer-to-peer multiplayer games.

A more widely used approach (Wang et al. 2009, page 3) involves using a time warp algorithm whenever some inconsistencies appear in the local game state. With the help of the algorithm the game can roll back to a previous and a more consistent state.

2.6.4 Fairness

Maintaining fairness is an essential part of making a game just and fun for the players. There should be no unfair advantages to some players over the others due to latency. Lin et al. (2002) have discovered two issues regarding fairness in games created with the client/server model. To tackle these issues there is a method named Sync-MS used to promote fairness in games by balancing fairness and response time. One issue is regarding the accessing of update messages amongst the clients and the unfairness it can inflict. Since different clients all over the world experience varying latencies in receiving the update messages, some players can take actions with respect to the latest update before some other players even have received the update messages. Ideally, all clients should receive the update messages from the game server simultaneously and this is achieved by the Sync-out mechanism of Sync-MS. It puts the update messages in a queue and awaits the messages to have arrived at every client. Only after every client has received it, it is delivered to the game application. In effect, all clients can react to the same game state justly. The second issue concerns the order of which the server processes the action messages from the clients. An action message sent from one client could due to bad internet connectivity be pushed behind events occurring afterwards. This causes the server to process the messages in a out of sync order manner irrespective of their real-time occurrence. To cope with this, Sync-MC uses its Sync-in mechanism to insert waiting periods on each action message so that a fair processing of the messages can ensue.

2.7 Measuring of Network Performance

When measuring the network performance there are two variables that ought to be taken into consideration: response time and transfer speed. Response time has to do with the latency in the communication between the server and the clients and it measures the so called 'round-trip time' (RTT), the time which a network package travels from the server to the client and then back again. Transfer speed, in turn, is the measure of how much data that can be sent back and forth between for example the server and the clients. The transfer time limits the number of people that can play the game concurrently, i.e. the scalability of the game (Wang, Jarret, & Sorteberg, 2009, p. 4).

Wang et al. performed a test of the network performance for different mobile networks using the network protocols UDP and TCP. The yielded results with respect to transfer speed show that when using TCP, the size of the network packages affects the transfer speed, although leaving WLAN as an exception. When the UDP protocol was used, however, there was no substantial impact on the transfer speed caused by different package sizes. With UDP the different networks (GPRS, UMTS, EDGE, and WLAN, see section 2.3) all have transfer speeds of 500 ms or less, whereas some mobile networks used with TCP had transfer speed of one second or more.

As deduced, UDP gave much better results for all networks than TCP with respect to the round-trip time. WLAN had the lowest RTT, staying below 200 ms at all times and the third generation networks were the first-runner up. When using TCP the latency difference between WLAN and UMTS was considerably high in comparison with when using UDP.

The both network protocols have varying performances on the different mobile networks. As previously discussed, Wang et al. concluded from their tests of transfer time that whilst using UDP the package sizes do not affect the transfer time (up to a certain size) and that the response time for the different mobile networks were much lower with the UDP protocol. Consequently, UDP is the wisest choice for games requiring low response time and transfer time. However, the disadvantage of this protocol is that it does not manage packet loss which could impair the game consistency.

From a series of test conducted by Wang et al. (2009), it is evident that WLAN is the mobile network that has the shortest response time as well as the quickest transfer speed. However, since WLAN is not accessible everywhere it loses its candidacy for being the most promising mobile network for mobile multiplayer games. To take its place, 3G networks come next in line and according to Wang et al. it is the most suited mobile networks for mobile multiplayer games. 3G can deliver data rates up to 2mbps. However, it is important to make the reader aware that the journal was published in the year of 2009. Since then many advances have been made and as Yang (2012) states, more recent fourth-generation (4G) mobile networks stand a competitor and can deliver data rates significantly higher than its predecessor (3G), and it provides greater bandwidth. Yang further mentions that according to a speed test conducted on a 4G smartphone, 4G delivered data rates up to 100 Mbit/s downstream and 50 Mbit/s upstream. With this result in hand, it is interesting to consider the type of games and other mobile applications that will emerge in the near future. Unfortunately, however, there

exists surprisingly little literature of mobile multiplayer games and other applications related to the high-speed wireless 4G networks.

2.8 Battery

Another determinant in the development of multiplayer games on mobile devices is the battery as these games typically consume large amounts of energy. Although mobile devices are displaying a tremendous progress, the battery technologies are not able to keep up the pace and are lagging behind (Bhojan, Akhihebbal, Chan, & Balan, 2012, p. 21). The problem lies in the fact that this could affect the potential amount of hours the game playing could carry on for. To circumvent this problem many reported measures have been taken. One example of this is the middleware framework 'ARIVU'. Year 2011 Bhojan et al. published a paper in which they presented 'a scalable power aware' middleware ARIVU that has the ability of conserving the energy consumption of the following resources: wireless interface, display and processor, and the system state. All of this is performed by the middleware without neither degrading the gaming experience nor the quality perceptibly (Bhojan, Akhihebbal, Chan, & Balan, 2012, p. 21). The middleware is reported to have saved up to 60% of the total energy consumption by the 802.11 g wireless interfaces for First-Person shooting game (FPS, see section 2.2.3) and up to 35% when it comes to Massive Multiplayer Online Role-playing Games (MMORPG, see section 2.2.2) by the same wireless interfaces.

A study by Bhojan et al. (2012) of how ARIVU can be used to save the energy consumption of the resources stated was carried out on the two major games: Quake III and Ryzom. Quake III is a first-person shooting game (FPS) and Ryzom an MMORPG game (Bhojan, Akhihebbal, Chan, & Balan, 2012, p. 22). Both of them are open sourced and they can be considered as good representatives for their respective genres as they are very popular. The study's results are consistent with what was expected, a whole 60% of wireless interface power could be saved without having any impact on the gaming experience or the quality. What was moreover touched upon, however superficially, is the extent to which gaming quality should take precedence over power saving, and the authors stated that balancing the trade-off can be tricky at times.

2.9 Designing for Mobile Multiplayer Games

Mobile devices have some severe constraints that have to be taken into consideration by the game designers. These are among others: small screens with the possibility of changing orientation (landscape and portrait), limited input controls, the need to deal with other tasks, the nonphysical delivery mechanism (wireless network for example) and the differences in performance and capabilities in different mobile devices (Callow, Beardow, & Brittain, 2007, p. 40).

One game design consideration, as mentioned above, is whether or not landscape and portrait views should be supported. For example as brought up by Callow et al. (2007) a labyrinth game could have first person view in portrait mode and a map when the screen in rotated to a landscape view.

Callow et al. (2007) argues in his paper, that the biggest barrier mobile multiplayer games have to overcome, is the issue regarding a sufficing player base. For a person to be able to play, he first has to find an opponent (or a team player). The implementation of multiplayer games thus far, is done in such a way that the users usually have to wait for long periods whilst a multiplayer session is being started. The users then have to wait in a lobby for other players. Callow et al. (2007) suggest that a good game design would be to have the servers initiate bots, either AI-based or predefined server-side players, so that the players would not have to wait for too long. Ideally, the users should not be able to tell the difference between the bot and a real life player. This paves the way for another problem, the problem of personalising the game. When using the traditional consoles or computers it is possible to multitask in the sense that players can both play the game and at the same time communicate with each other through the means of chat rooms or by voice. On mobile devices this is not as feasible considering the limited screen and input options. A proposed idea is to use predefined taunts (e.g. battle cries).

Michael Powers argues that when connecting to a multiplayer game session with a mobile device it should be as effortless as possible, with no unnecessary registration, waiting, or other 'formalities' preventing the players from doing what they want to, to play the game. In designing a multiplayer game, the developers should take into consideration that players can leave the game deliberately or accidentally and that this, in turn, should not result in disruptions for the other players in the game.

Mobile multiplayer games require a screen area that can hold the landscape of the game, and also the more players that join the game, the more pixels are needed to render the game state and game view to show a lot of players interacting with each other. The graphic user interface must be developed with this in mind.

With the appearance of hundreds of different mobile devices with different shapes, sizes, and input methods it is of paramount importance that the game developers design their games to work effectively across the different hardware designs.

2.10 Required Resources

The immense popularity of mobile applications such as games has led to innovations in mobile hardware, e.g., the recent introduction of mobile quad-core CPUs (Le, Fragouli, & Markopoulou, 2012, p. 13). However, there was astonishingly little literature covering the topic of how CPU and graphics play their respective roles in the context of the development of mobile multiplayer games and to what degree they can have an effect. Anh Le et al. (2012) imply that although CPUs are becoming increasingly powerful it is still important to consider that a large group of people still have mobile devices with less powerful CPUs and less cutting-edge technology than is available today. It would be unwise to discard this fact and only aim to develop games for the more advanced technologies, the games should work for the older mobile devices as well. The same applies for graphics. The best approach is to use a technological standard which is used in many mobile devices as a baseline for what should be supported.

3 Results

In this section some guidelines are presented based on the findings of the investigation from section 2. Background. The guidelines target developers designing and programming mobile multiplayer games with the purpose of facilitating and bringing up points worth considering when designing them.

Guidelines

3.1 Mobile Multiplayer Game Genres

First it is important to be clear about what type of multiplayer game it is that you want to develop. Is it a turn-based game like a board game or a more interactive and dynamic game as a real-time multiplayer game? Knowing what type of game you are developing is important for designing the game and it also governs the degree of complexity of the design.

Another thing worth considering is whether the game should be an online or an offline multiplayer game. Offline games have the advantage that they can be played in circumstances where there is generally no availability of mobile networks, e.g. when traveling or when located in rural areas. In this case Bluetooth or WLAN should be supported so that PANs (Personal Area Networks) can be created. It is highly recommended that you familiarise yourself with the two configurations of PANs: piconets and scatternets.

If developing online multiplayer games, the next step in the developing processes is to look into the different mobile networks and their respective capabilities. The intended game should be developed with the restrictions and the technical scope of the mobile networks in mind because upon doing so, many complications regarding network latency can be evaded.

3.2 Mobile Networks

When developing online multiplayer games, an internet connection for the mobile device must be used. It is important to remember here what game genre you have chosen because they bring with them their own set of points to consider in the developing and the designing of the game. Say that you are developing a real-time multiplayer game, then there is a greater need for a wide bandwidth as more information is typically transmitted and there should be very low network latency so that the gaming experience of the players does not suffer (e.g. WLAN with high speed internet connection or 4G). Turn-based games usually have the advantage of hiding the effects of network latency in the time the players pause to think through their move (they will work with 2G or 3G networks as well).

The best mobile network capable of high data rates and providing greater bandwidth in the market today is the fourth generation mobile network (4G).4G is to be used for real-time multiplayer games, eg. MMORPG games or FPS multiplayer games to guarantee the best gaming experience, evading the issues regarding high latencies and low bandwidth. 4G supports data rates up to 100mbps and it is becoming more and more available all over the world. In Sweden for example there are a handful of cities that support it. However, as there are still mobile

devices that do not support 4G, the game should work almost as effortlessly for 3G networks as well. This is an important remark.

The biggest hurdle to overcome when developing networked multiplayer games is network latency as it is what impairs the game-play experience the most for users. Network latency generally affects network efficiency and utilization, visual consistency, and fairness in the games. There are some techniques to lessen the impairment of network latency and limited bandwidth that are worth looking into. It is recommended that you use the Dead-Reckoning technique, which is especially useful when developing real-time multiplayer games. It extrapolates a game object's exact position by looking at its last known position and velocity. There are typically many different game objects under the control of the players e.g. Artillery, vehicles, other creatures et cetera. In order for an update of the whole game view to occur for all the players, a big amount of data would be required to be transmitted between the game server and its clients. By using this technique, however, the clients need only supply changes to the velocities of the game objects and then the extrapolation algorithms will sort out the rest. This is to significantly cut down the amount of data to be transmitted.

It is also recommended that you use the so called Movement Prediction technique which reduces the likelihood of warping to occur. Warping is a phenomenon that occurs due to loss of network packages and/or low bandwidth. It causes objects to erratically jump around on the screen and ruins the visual consistency of the game.

Network latency can even compromise fairness in the game. Some players can benefit from other players suffering from the effects of latency. To avoid this problem you should consider using the so called Sync-MS method to promote fairness in your game. It works with two different mechanisms: the Sync-in and the Sync-out mechanism. Sync-out is used to ascertain that the update messages from the game server arrive concurrently at the clients and Sync-in is used to

Using the network protocol UDP is quicker for all mobile networks.

3.3 Designing the Graphical Interface of the Games

Mobile devices have many constraints and some of them demand special attention. The foremost constraints are the small screens and the limited input methods. As a developer you have to be creative with how you make the most out of whatever little you are provided with. If you perhaps have a strategy game in mind, you can use the portrait and the landscape views to change between the game view and the landscape map just by tilting the phone. Also, do not neglect the issue of scalability (i.e. amount of players) of the game, especially when designing MMORPG games. For each additional player extra pixels are needed for rendering the game state and the game view (so the player can see every player that is in its surrounding view and not feel that players in same area are shown as a clump moving around). One smart way of handling this issue is to allow the player the option to zoom in and out.

Another important remark is that the game you are developing should be supported by and run effectively over a wide range of different mobile devices. Today there are many different varieties of mobile phones in the market, with different shapes, sizes, and input methods.

It is important how you handle the issue regarding a sufficing player base. The players should not have to wait in a lobby for other players for a long period of time. In some cases you should consider having a bot to fill the gap of other players. Also, the players should not have to wait for too long for a multiplayer game session to be initiated. When the players connect to the game, it should go as effortlessly as possible, without having any tedious registrations or long waiting periods to overcome first. As a game developer for mobile devices you must have in mind that mobile games are largely played on the go, when traveling back and forth to school/work, when waiting et cetera. It should be as seamless to start the game as to quit the game and the game should not be disrupted because of dropouts, i.e. players accidentally leaving the game because of bad internet connections for example.

3.4 Distributed Topologies

You can choose either between using the client/server model or the peer-to-peer model as the architectural groundwork for your game. There are advantages and drawbacks for both methods. Important to consider, though, is how much load there should be on the clients and server respectively.

If you decide to use the Client/Server topology, it is important that all the players from any part of the world are not connected to a server situated in a single location as this could cause network latency and lags correlated to the physical distance from the server. A smarter idea is to have several servers placed strategically around the globe and that players in neighbouring locations always play on the same server.

3.5 Mobile Resources

There is much advancement in the field of mobile technology and mobile hardware and more is still to come. It is not feasible for everyone to keep up the pace and buy the newest mobile phone every time it appears in the market. Therefore, it is important that the game is developed so that it supports not only the newest technology but the technology that is used by most players. It is worth designing the game so that it uses the CPU and memory sparingly.

Since powerful batteries are evolving quite slowly in comparison to other mobile resources, you should consider integrating some power-saving features into your game. ARIVU, a power aware middleware, has several features for conserving the energy consumption of for example the display and monitor and for wireless interfaces. By having supporting some of these features in your game, the players would be able to play the game for a longer period of time without having to worry about battery life. A game or an application that drain the battery too much are generally not appreciated and quickly discarded by most people.

4 Conclusion

What became quite evident after having conducted the investigation of the relevant infrastructures for mobile multiplayer games was that although mobile technology is entering a new era of cutting-edge technology there are still some flaws and drawbacks in the technology. Furthermore, developers still have to pay consideration to the fact that not everyone purchases the most recent technology as quickly as it appears in the market. The games still have to be developed and designed with respect to the slightly older technology so that the game is supported across a wider population.

We quickly understood the importance of having a good mobile network especially when developing an FPS game or MMORPG, which fall under the genre real-time multiplayer games. Generally, for this type of games big amounts of data are transmitted and therefore, a wide bandwidth is essential for the game state and the game view to be consistent. For other genres of multiplayer games the impact of latency is not as perceptible and it does not cause a substantial downgrading of the gaming experience. Developers should test their games on different devices with different network connections to widen their player base. It can happen that players in different locations of a town can have different signals strengths. By performing these tests, the risk of the game getting bad reviews because of inconsistencies in the game play is decreased.

It is also important to consider the fact that batteries for mobile devices are not advancing in the same pace as mobile devices which means one has to consider the consumption of resources. Players will not like to play a game on their way to somewhere if the game drains their battery too quickly thus preventing them from using the phone for its original purpose- to communicate with other people.

We believe these guidelines can help many developers create games where the designs have been thoroughly considered with the points covered in the guidelines. This way the games stand greater chance of succeeding and not falling for “bad reviews” because there were some aspects that were left unconsidered.

As stated in the beginning of the thesis, the guidelines are not empirically tested. As future work, developers can test these guidelines thoroughly in order to add some empirical ground to the fact that these guidelines actually do facilitate the developing. An empirical study can be made with interviews and questionnaires to developers to see if they believe these guidelines are useful and the advice and the recommendations given are valid. If sufficient resources are given, researchers can also analyse exiting famous games and see if these are developed with all of the factors mentioned in the guidelines in mind.

5 Bibliography/References

Bhojan, A., Akhihebbal, A. L., Chan, M. C., & Balan, R. K. (2012). ARIVU: Making Networked Mobile Games Green. *The Journal of SPECIAL ISSUES on Mobility of Systems, Users, Data and Computing, 17*(1), 21-28.

Callow, M., Beardow, P., & Brittain, D. (2007, Nov 1). Big Games, Small Screens. *Queue, 5*(7), pp. 40-50.

Dideles, M. (2003, 06). Bluetooth: a technical overview. *Crossroads, 9*(4), pp. 11-18.

Fritsch, T., Ritter, H., & Schiller, J. (2006). CAN mobile gaming be improved? *Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games. article no. 44.* Singapore: ACM.

Henderson, T., & Saleem, B. (2003). Networked games: a QoS-sensitive application for QoS-insensitive users? *RIPQoS '03 Proceedings of the ACM SIGCOMM workshop on Revisiting IP QoS: What have we learned, why do we care?* (pp. 141-147). Karlsruhe: ACM.

Le, A., Fragouli, C., & Markopoulou, A. (2012). MicroPlay: A Networking Framework for Local Multiplayer Games. *Proceedings of the first ACM international workshop on Mobile gaming* (pp. 13-18). Helsinki, Finland: ACM.

Lin, Y., Guo, K., & Sanjoy, P. (2002). Sync-MS: synchronized messaging service for real-time multi-player distributed games . *10th IEEE International Conference on Network Protocols* (pp. 155 - 164). Paris, France: IEEE.

Manweiler, J., Agarwal, S. , Zhang, M., Choudhury, R. R., & Bahl, P. (2011). Switchboard: A Matchmaking System for Multiplayer Mobile Games. *MobiSys'11* (pp. 71-84). Maryland: ACM.

Nicholas, J., & Claypool, M. (2004). The effects of latency on online madden NFL football. *Proceedings of the 14th international workshop on Network and operating systems support for digital audio and video* (pp. 146-151). Cork: ACM.

Pantel, L., & Wolf, L. (2002). On the impact of delay on real-time multiplayer games. *Proceedings of the 12th International workshop on Network and operating systems support for digital audio and video* (pp. 23-29). Miami, Florida: ACM.

Powers, M. (2006, November). *Mobile Multiplayer Gaming, Part 1: Real-Time Constraints*. Retrieved 04 12, 2013, from Oracle: http://www.oracle.com/technetwork/articles/javame/gamepart1-155645.html

Samukic, A. (1998). UMTS Universal Mobile Telecommunications System: development of standards for the third generation. *Global Telecommunications Conference, 1998. GLOBECOM 1998. The Bridge to Global Integration. IEEE. 4*, pp. 1976-1983. IEEE.

Sheldon, N., Girard, E., Borg, S., Claypool, M., & Agu, E. (2003). The effect of latency on user performance in Warcraft III. *Proceedings of the 2nd Workshop on Network and System* (pp. 3-14). Redwood City, California: ACM.

Soh, J. O., & Tan, B. C. (2008, 03). Mobile Gaming. *Communications of the ACM, 51*(3), pp. 35-39.

Wang, A. I., Jarret, M., & Sorteberg, E. (2009). Experiences from Implementing a Mobile Multiplayer Real-Time Game for Wireless Networks with High Latency. *International Journal of Computer Games Technology, 2009*, 1-14.

Yang, S. C. (2012). Mobile applications and 4G wireless networks: a framework for analysis. *Campus-Wide Information Systems, 29*(5), 344-357.

Zhu, Y., Liu, Y., Ngan, H., & Chen, Q. (2007). Shift Coding: Efficient State Update in Mobile Peer-to-Peer Multiplayer Games . *International Conference on Parallel Processing Workshops* (pp. 58-64). Xi'an: IEEE.