

<http://www.cisjournal.org>

Building Multiplayer Games for Smartphones: Experience with Connect-4 Game

Dima Gupta, Ruba Abu Khalaf, Nesreen Abu Ajamieh, Abdallah Alashqur*

Faculty of Information Technology, Applied Science University, Amman, Jordan

* alashqur@asu.edu.jo

ABSTRACT

Social Networking has become very popular on smart phones. Game playing is an important part of social networking. However, most games that run on smart phones are single-player games. Multiplayer games on smart phones are an area that is still in its infancy. Not many games of this type are available on Smartphone platforms. A lot of research and experimentation is needed before the area of mobile multiplayer games can become mature and wide-spread. In this paper we report on our experience developing a multiplayer game that can be played by two players who are not geographically co-located using their smart phones. The system handles connectivity between the two smart phones. Whenever a player makes a move, the opponent player sees it at a near real time speed. The opponent can then make a counter move and so on. The system keeps track of the score and, when the game is over, it announces the winner to the two smart phones. It then prompts them to see whether they want to re-play the game or terminate.

Keywords: *Smartphone, GCM, gaming technology, social networking*

1. INTRODUCTION

The number of smart phones that are sold annually has recently exceeded the number of sold PCs. The demand for interactive applications that run on smart phones and for applications that aid in social networking is on the rise. One area that needs attention is multiplayer gaming on smart phones. Players can be geographically far from each other but desire to play games against each other using their smart phones. The expectation of Smartphone game players in terms of performance, quality, and richness of the game is high because many of them have experienced such activity on the PC. PC performance and richness is used by many players as a benchmark for judging their Smartphone's performance. When a player makes a move on his/her screen, that move needs to be reflected on the screen of the opponent's screen with minimum latency.

In recent years there has been a lot of work devoted to the area of designing and developing multiplayer games for smart phones. Reference [1] has studied the social aspects of multiplayer games and distinguished between known vs. unknown players. Reference [2] provided a full-system analysis and characterization of interactive Smartphone applications including games. In [3] a framework is provided that exploits the locality of players in order to improve the latency of games. A mobile multiplayer gaming system that takes advantage of opportunistic communications is described in [4]. Some challenges facing the development process when developing games for smart phones that are to be played outdoors are outlined in [5]. Other systems and approaches in the area of multiplayer gaming on computerized and mobile systems can be found in [6-11].

In this paper we describe our experience developing a gaming system for smart phones. The game has been implemented as a graduation project for the first three authors under the supervision of the fourth author.

The game is called *Connect-4*, or sometimes referred to as *four-in-a-line*.

The remainder of this paper is organized as follows. In section 2 we give a summarized technical background related to our gaming system. In Section 3 we give a brief description of the development of the system. Section 4 provides a description of the game and shows screen shots from live sessions to demonstrate the main aspects of the game (mainly the fact that it is a multiplayer system and that it is played on the Smartphone). Conclusions are provided in Section 5.

2. BACKGROUND

In this section we briefly describe the game, the development environment and the communications techniques used in our gaming system. In Subsection 2.1 we describe the game itself regardless of the implementation environment. In Subsection 2.2 we briefly describe the development environment and tools that we used to develop our gaming system. In Subsection 2.3 we describe a very important component of the system which is the communication mechanism that we used, especially Google Cloud Messaging.

2.1 Connect-4 Game

Connect-4 (sometimes called Four-in-a-Line) is a popular two-player game. The game has a grid consisting of number of columns and rows. A cell exists at the intersection of each column and row. The way the game works is that each of the players starts by choosing a color different from each other. Then the players take turns dropping colored circles vertically from the top into the columns. The pieces fall down to settle in the next available cell in the grid within that column. The goal of each player is to end up with four of his/her pieces (i.e. pieces of the same color) adjacent to each other horizontally, vertically, or diagonally. The players keep alternating moves until one of them reaches the goal state first and is declared as a winner. An example is shown in

<http://www.cisjournal.org>

Figure 1 where the player with the blue color is a winner since he/she reached the goal state first. As show in the figure, that player was able to place four blue circles vertically and next to each other.



Fig 1: Connect-4 winning state

2.2 Development Environment

Our objective is to build a game which can be played by two players remotely using their smart phones. The two players don't have to be physically co-located. When one of the players makes a move, the second player should be able to see that move on the screen of his/her Smartphone. Then the second player makes a counter move, which is seen on the Smartphone of the first player. The two players keep alternating moves until one of the two players reaches the goal state by having four pieces next to each other (horizontally, vertically, or diagonally) and is declared as a winner by the system.

The system has been developed using Java for the Android platform. The technologies/tools we used are described below.

- **JSON.** We used JSON (JavaScript Object Notation) because of its simplicity and compact nature. It is good for transmitting relatively small datasets which can be grouped into name-value pairs. It is also easy to parse. JSON format is familiar to programmers but it uses language-independent conventions. It is better suited to our task than HTML and XML.
- **ECLIPSE.** This is an integrated development environment (IDE) that is also a multi-language environment. We also used a plug-in for the Eclipse IDE, namely, Android Development Tools (ADT).
- **Apache WAMP.** WAMP stands for Windows, Apache, My SQL, and PHP. WAMP represents a bundle of free (open source) applications. WAMP is normally

combined with Windows server environment.

- **PHP** is a widely-used, open source, and general-purpose scripting language that is especially suited for web development. CURL in PHP is a library that lets you make HTTP requests in PHP.

2.3 Communications

In order to be able to communicate game moves and information between the two players we use Google Cloud Messaging (GCM). GCM is a free cloud service provided by Google that enables developers to send data from servers to their Android applications on Android smart phones. It also enables sending upstream messages from the user's smart phones back to the cloud. GCM service handles all aspects of queuing of messages and delivery to the target Android application running on the target Smartphone.

Figure 2 shows the architecture of GCM messaging as taken form [12]. The links labeled 1, 2, 3, 4, a, and b in Figure 2 are explained as follows [12].

- (1) Here Android Smartphone sends *sender_id* and *application_id* to GCM server in order to register.
- (2) After a successful registration is completed, GCM server issues *registration_id* to the Android Smartphone.
- (3) After receiving *registration_id*, the Smartphone sends it to the server.
- (4) The server stores *registration_id* in the database for later usage.
- (a) Whenever push notification is needed, the server sends a message to GCM server along with *smartphoneregistration_id* (as it was stored in the database).
- (b) GCM server then delivers that message to the respective Smartphone(s) using *Smartphone registration_id*.

When a player makes a move, we use GCM to communicate the change to the application running on the Smartphone of the opponent player. The application running on the opponent player's Smartphone then renders the change on the screen for the player to see.

3. SYSTEM DEVELOPMENT

Figure 3 shows an algorithm from our system that finds if there is a winner after a player makes a move. The algorithm has been written in Java script.

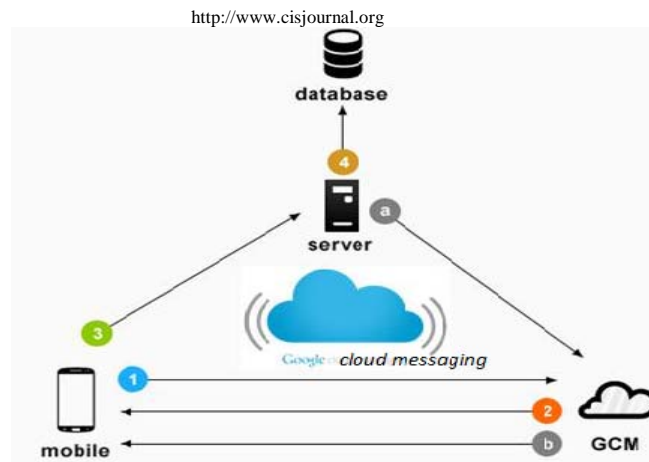


Fig 2: The role of GCM in communications

```

• public void Is Winning() {
•   in tu = history. Get Last().u;
•   in tv = history. Get Last().v;
•   // u,v are the coordinates of the cell
•   if (!Is In Range(u, v))
•     return;
•   int cell = get Cell (u, v);
•   if (cell == 0 || cell == 3)
•     return;
•   for (int direction = 0; direction < 8; direction += 2)
•   {
•     // In all directions...
•     Int count A = count Line (u, v, direction);
•     Int count B = count Line (u, v, direction + 1);
•     if (count A + count B > 4) {
•       victory Achieved = true;
•       for (inti = 1 - count B; i < count A; i++) {
•         intn U = u + i * directions[direction][0];
•         intn V = v + i * directions[direction][1];
•         winning Cells. Add (new Point(nU, nV)); }
•       return;}
  
```

Fig 3: Algorithm to determine if there is a winner

As part of our development, we created a database called GCM using My SQL to store the game related information. We also created a table called Players.

4. SYSTEM DESCRIPTION

A user can see the player list of players who are currently available. A player can request one of the players in the player list to play with him/her. The requested player can accept or reject. If the requested player accepts, the system keeps communicating between the two players using GCM as long as they are playing the game. When the game is over, the players can chose to re-play or terminate.

In this section we describe some screen shots from some gaming sessions that were played on the system. The screen shots are meant to give the reader an

idea about the look and feel of the system. Figure 4 show the icon representing the game on the bottom left of the screen. After logging in, the user can see the list of potential players as shown in Figure 5. If the player selects one of the available players on the list to be an opponent in a game, a notification goes to the selected player. The notification informs the selected player about who is the person who wants to play against him/her. This is shown at the bottom of the screen of Figure 6 (i.e., the text: "Omar wants to challenge you"). If the selected opponent accepts to challenge, the game starts. Figure 7 shows a screen shot of an on-going game between two players. When one of the two player reaches the goal state first (i.e., has four pieces adjacent to each other whether horizontally, vertically or diagonally), the system informs the two players. Figure 8 shows the message sent to the winner to let him know that the game is over and he is the winner.

<http://www.cisjournal.org>

Similarly, Figure 9 shows the message sent to the looser to let him know that the game is over and that he lost. Notice at the top of the screen of Figure 8 and Figure 9 the system shows the word “winner” followed by the color of the piece that belongs to the winner. Note that the screen of Figure 8 and Figure 9 show different piece arrangements because they were taken from two different game sessions and not the same exact game session. Figure 10 show the activity diagram between player 1, player 2, and the system.

5. CONCLUSIONS

In this paper we have described our experience developing a multiplayer gaming system for the Smartphone. We built our system for the Android platform but the ideas can be generalized to other platforms. We relied on Google Cloud Messaging to facilitate communications between smart phones. We chose Google Cloud Messaging because it is simple to use and provides the means necessary to aid in the success of our gaming system. In addition to Google Cloud Messaging, we relied on systems and tools such as Java, JSON, ECLIPSE, Apache WAMP, and PHP. The system was successfully

developed and demonstrated to audience. In this paper we also showed some screen shots taken from live sessions of the game. The screen shots demonstrate the steps in which the system is played and give the reader a flavor of the look and feel of the system.

We think that multiplayer game for the Smartphone platform is an area that is important to enrich and augment the social networking experience. Many multiplayer games exist for the PC platform. However, a lot of research and experimentation is still needed before multiplayer games for smart phones become a success and grow in terms of how widely it is used. Performance of the system as well as richness of the graphics and user interface is a requirement for the success of any multiplayer gaming system for smart phones.

ACKNOWLEDGMENT

The authors are grateful to the Applied Science Private University, Amman, Jordan, for the financial support granted to cover the publication fee of this research article.



Fig 4: Connect-4 app as an icon



Fig 5: Players list

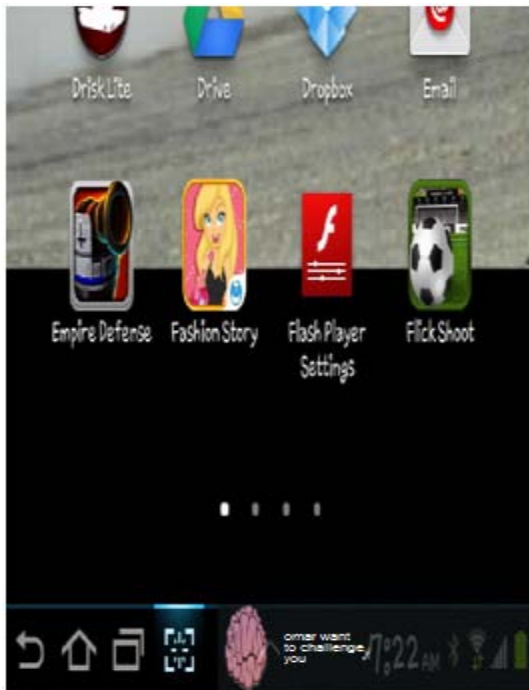


Fig 6: A notification to inform a player that someone is proposing to start a game with him

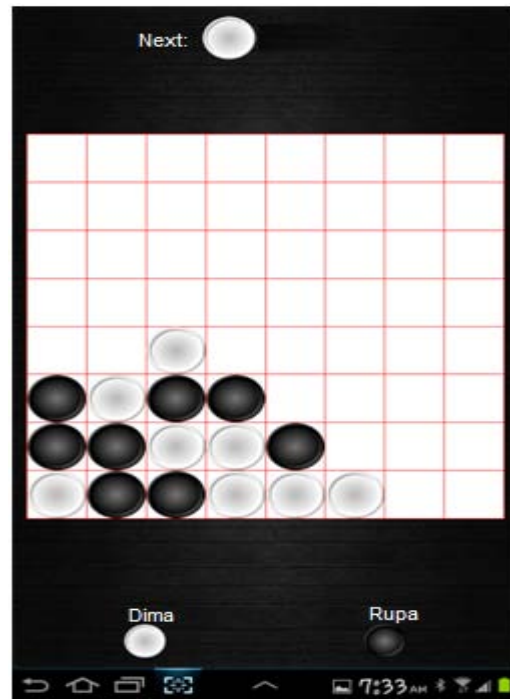


Fig 7: An on-going game between two players: Dima and Rupa

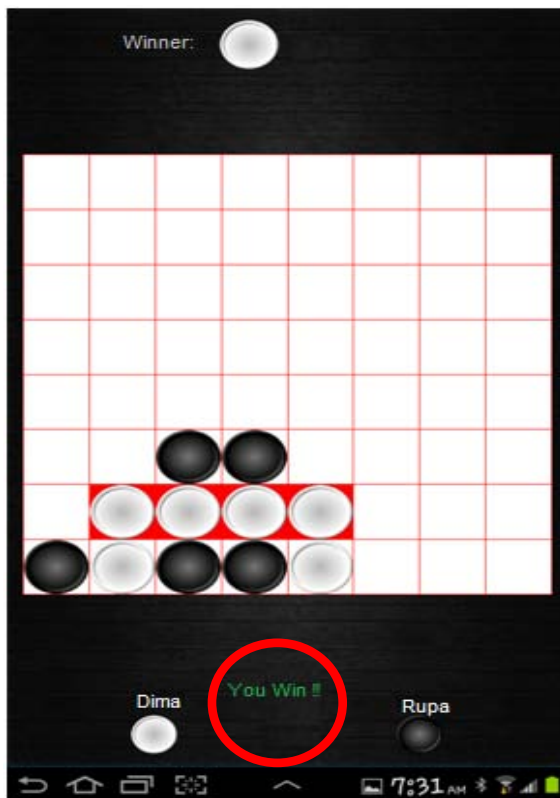


Fig 8: A message sent to the winner

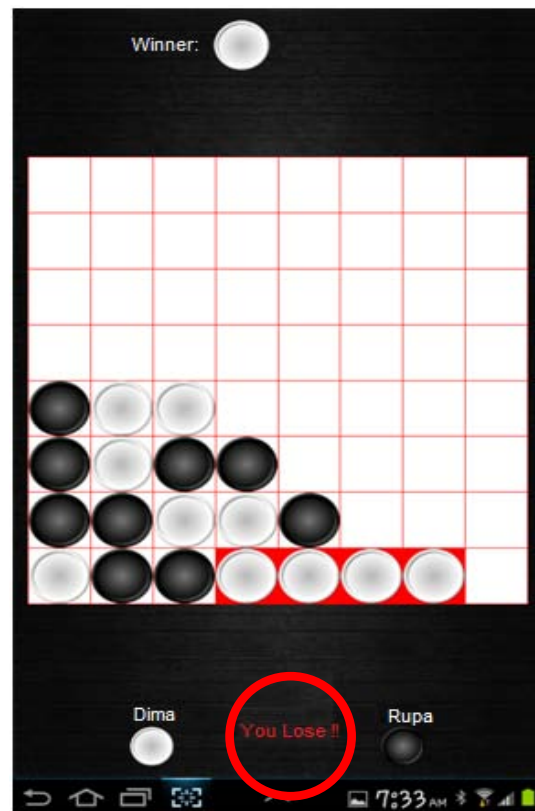


Fig 9: A message sent to the loser

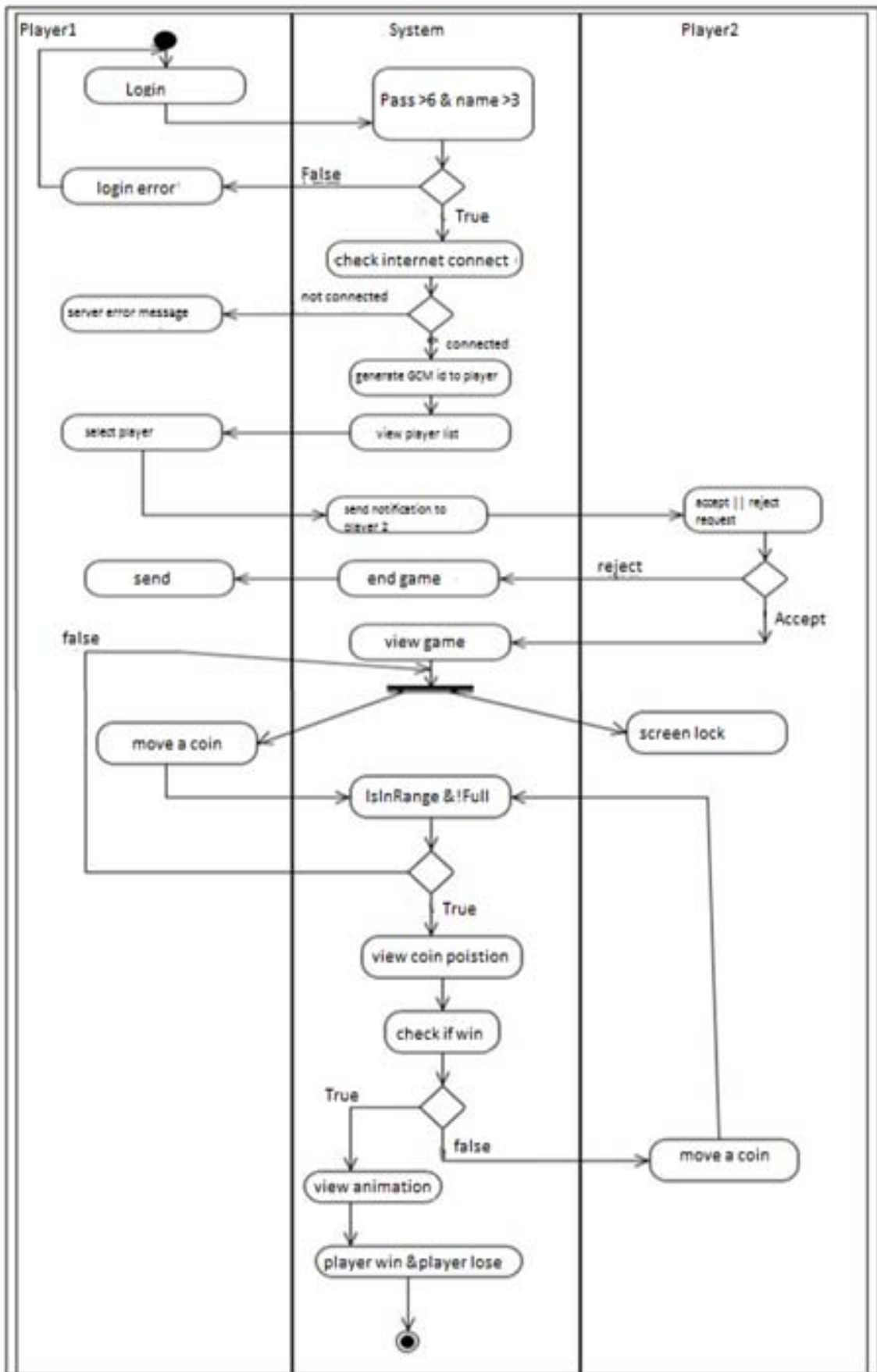


Fig 10: Activity diagram

REFERENCES

- [1] Kevin A. Li and Scott Counts. 2007. Exploring social interactions and attributes of casual multiplayer mobile gaming. In *Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology (Mobility '07)*. ACM, New York, NY, USA, 696-703.
- [2] A. Gutierrez, R.G. Dreslinski, T.F. Wenisch, T. Mudge, A. Saidi, C. Emmons, and N. Paver. Full-System Analysis and Characterization of Interactive Smartphone Applications. In *the proceedings of the 2011 IEEE International Symposium on Workload Characterization (IISWC)*, pages 81-90, Austin, TX, USA, 2011.
- [3] A. Le, L. Keller, C. Fragouli, and A. Markopoulou. Micro Play: a networking framework for local multiplayer games, in Proc. Proceedings of the first ACM international workshop on Mobile gaming, New York, NY, USA, 2012, pp. 13-18.
- [4] Roelof Kemp, Nicholas Palmer, Thilo Kielmann, and Henri Bal. 2010. Opportunistic communication for multiplayer mobile gaming: lessons learned from Photo Shoot. In *Proceedings of the Second International Workshop on Mobile Opportunistic Networking (MobiOpp '10)*. ACM, New York, NY, USA, 182-184.
- [5] Robert J Hall Software engineering challenges of multi-player outdoor Smartphone games Proceedings of First ACM Workshop on Games and Software Engineering, Intl Conf Software Engineering Workshop on Games and Software Engineering, Pages: 52-55 Waikiki, Honolulu , HI, USA — May 21 - 28, 2011
- [6] N. E. Baughman, M. Liberatore, and B. N. Levine. Cheat-proof payout for centralized and peer-to-peer gaming. *IEEE/ACM Transactions on Networking*, 15(1):1–13, Feb. 2007.
- [7] J. Manweiler, S. Agarwal, M. Zhang, R. Roy Choudhury, and P. Bahl. Switchboard: a matchmaking system for multiplayer mobile games. In Proc. of ACM MobiSys, pages 71–84., 2011.
- [8] A. Sardouk, S. M. Senouci, N. Achir, and K. Boussetta. Assessment of MANET broad cast schemes in the application context of multiplayer video games. In Proc. of ACM SIGCOMM Net Games Workshop, pages 55–60, New York, NY, USA, 2007.
- [9] M. Claypool and K. Claypool. Latency and player actions in online games. *Commun. ACM*, 49(11):40–45, Nov. 2006.
- [10] Y. Bernier. Latency compensating methods in client/server in-game protocol design and optimization. In Game Developers Conference, 2001.
- [11] E. J. Berglund and D. R. Cheriton. Amaze: A multiplayer computer game. *IEEE Software*, 2(3):30–39, May 1985.
- [12] Android Documentations <http://www.androidhive.info>, <http://developer.android.com/google/gcm/gcm.html>, and <http://thamilAndroid.wordpress.com/TAG/GCM>.

AUTHOR PROFILES

Dima Anil is a student graduating in 2013 from the Applied Science University (Faculty of Information Technology) in Amman, Jordan. Dima is specializing in Software Engineering and is interested in Java programming and software design.

Ruba Abu Khalaf is a student graduating in 2013 from the Applied Science University (Faculty of Information Technology) in Amman, Jordan. Ruba is specializing in Computer Science and is interested in programming and mobile applications.

Nesreen Abu Ajamieh is a student graduating in 2013 from the Applied Science University (Faculty of Information Technology) in Amman, Jordan. Nesreen is specializing in Computer Science.

Abdallah Alashqur is an associate professor in the Faculty of Information Technology at the Applied Science University in Amman Jordan. Dr. Alashqur has around 30 years of experience (industrial as well as academic) in database systems and related technologies. He obtained his Master's and Ph.D. degrees from the University of Florida, U.S.A.