

Research Article

Using a Camera Phone as a Mixed-Reality Laser Cannon

Fadi Chehimi, Paul Coulton, and Reuben Edwards

InfoLab21, Lancaster University, Lancaster LA1 4WA, UK

Correspondence should be addressed to Paul Coulton, p.coulton@lancaster.ac.uk

Received 29 September 2007; Accepted 31 January 2008

Recommended by Kok Wai Wong

Despite the ubiquity and rich features of current mobile phones, mobile games have failed to reach even the lowest estimates of expected revenues. This is unfortunate as mobile phones offer unique possibilities for creating games aimed at attracting demographics not currently catered for by the traditional console market. As a result, there has been a growing call for greater innovation within the mobile games industry and support for games outside the current console genres. In this paper, we present the design and implementation of a novel location-based game which allows us turn a camera phone into a mixed-reality laser cannon. The game uses specially designed coloured tags, which are worn by the players, and advanced colour tracking software running on a camera phone, to create a novel first person shoot-em-up (FPS) with innovative game interactions and play.

Copyright © 2008 Fadi Chehimi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Since the appearance of “snake” on the Nokia 16110 mobile phone in 1997, the potential for mobile games has excited the industry and market analysts alike producing forecasts of game revenues between 3.6 and 18.5 billion for 2006 [1]. However, the reality is that the mobile games have only generated approximately 2.6 billion in 2006, predominantly from games that were either cut down versions of old arcade games or very simple casual games [1]. Whilst these types of game no-doubt have their place in the market, there are increasing calls within this fledgling industry for developers to produce more innovative games and for operators to support and promote them on their portals.

This innovation will most likely be achieved by embracing both the challenges and opportunities [2] that mobile phones offer, particularly their mobility, connectivity, and rich feature sets which now include elements such as cameras and 3-D motion sensors. Mixed-reality games that enable users to see and interact with virtual computer generated content superimposed on views of the real (physical) world on mobile phones certainly would certainly fall into this category. This is because they generally take into account of the nature of the device often by incorporating location [3], proximity [4] and different modalities for user interaction [5] to create original game experiences.

Whilst location undoubtedly helps provide interactive game play, it is not always easy to achieve. It often requires additional hardware and/or software such as a global positioning system (GPS), or APIs to access Cell ID, or to utilise enhanced time difference of arrival (ETDOA) [3]. However, many of these features are uncommon in general handsets of these games to become successful, they would need to be able to be distributed. One way of ensuring this is to create games that run on as many handsets as possible and are not tied to a particular phone feature or fixed to a particular physical location. Therefore, part of the rationale of this research project is to produce a location-independent, mixed-reality game that utilises one of the most common features on mobile phone’s camera. The camera has become almost a default feature on any mobile phone as evidenced by the fact that Nokia is looked at as the largest manufacturer of digital cameras in the world.

Whilst this is not the first mobile game to utilise the camera, it is unique when compared to previous games in that it uses advanced image processing of camera frames dynamically in real time. The captured frames are analysed on a frame-by-frame basis and fed to the application framework to provide the game interaction and status definition. The game could be regarded as a mobile mixed-reality version of paintball or laser tag, and hence it has been dubbed “Mobilazer.”

To place the game in context with other available camera-based games, we will start with a brief review of the related games and research projects in this area.

2. OTHER PHONE CAMERA GAMES

Cameras are now a common feature of even the most basic mobile phone. Indeed, a reported 295.5 million were shipped in 2005 which represented nearly 40% of all phones shipped [6]. There is, thus, a real opportunity for their use within games but as very few innovative games have done so and in quite varied ways.

Some of the earliest such games, unsurprisingly, came out of Japan around 2003 such as “Photo Battler” from NEC which allowed players to turn photos into character cards that were then assigned various attributes enabling players to compete against each other. Around the same time “Shakariki Petto” appeared from Panasonic which was a virtual pet that a player fed by taking pictures of colours that represent food, for instance, the colour red represented apples. More recent games have also explored using the pictures themselves to create mixed-reality games such as the “Manhattan Story Mash-Up” [7] and “My Photos Are My Bullets” [8].

Other games have evolved to use the camera to detect movements of the phone and transfer them to movements within the game. Probably the best known are from game developer, Ojom, with its games “Attack of the Killer Virus” and “Mosquitos” for Symbian S60 mobile phones. In both games, the enemy characters that the player must “shoot” are superimposed on top of a live video stream from the mobile phone’s camera. The player moves around this mixed-reality space by moving his phone and firing using the centre key of the joy pad. Other games have evolved to use visual codes to either detect movement [9], to imply, locate, or interact with objects such as ConQwest by Area Code [3].

However, none has yet utilised the complex real-time marker tracking, player recognition and interactivity identification in a real shoot-em-up game style as of that of “Mobilazer.” There has been several marker tracking designs but almost all were based on black and white markers, whose recognition can be more efficient but has poorer information storage. In most cases, the tags were designed to: (1) get a piece of information instantly [10]; (2) feed-in input options to mobile phone applications [11]; (3) augment 3D models for interactive viewing [2, 5]; and (4) to locate landmarks for guiding visually impaired people [12].

3. GAME DESIGN

“Mobilazer” engages two or more teams of players in an unbounded physical location and employs camera phones to track special tags fitted on the “armour” worn by each individual player providing by this the means of game interaction. The specially designed software turns a player’s mobile phone into a form of a “laser” cannon whereby he can shoot opposing players and interact with designated game objects such as team bases and posser-up collection points.

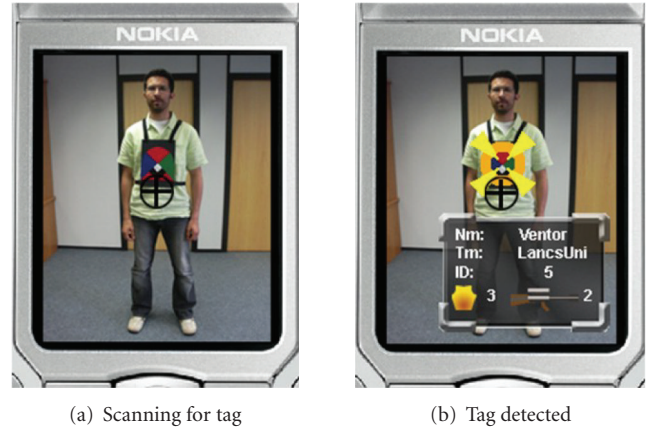


FIGURE 1: Tag detection.

The armour-like vests worn by each player have coloured tags mounted on their front and back faces (although this could be extended to include smaller tags on arms and legs). The double-facing tags enable players to shoot others from a variety of orientations simply by pointing the phone’s camera at the tag and aiming through the viewfinder (using the cross hair which is always in the middle of the screen) as shown in Figure 1(a). Once a tag is detected, “Mobilazer” displays a target sign centred at the coloured tag indicating the identification of the rival, and a box including his/her details (name, team, rank, weapons, etc.) as shown in Figure 1(b). The kills and points are controlled through interaction with a central game server through TCP/IP over the general packet radio service (GPRS).

3.1. Game modes

“Mobilazer” has been designed to accommodate four different playing modes which are selected by the preformed social grouping and controlled by a controller client unit (CCU) which is another mobile phone running as the game server.

3.1.1. Fortress

This game mode is a capture–the–flag style of game, engaging two or more teams. It starts by players representing each team agreeing on a location for their team base or Fortress which will be assigned a specific coloured tag. Each member of a particular team tries to protect his Fortress and attack the other teams’ Fortresses and members, who are then effectively removed from the game once shot. The game continues until members of one group are the only survivors, or when one base is left intact while all others are destroyed.

3.1.2. Last man standing

This is a battle mode where a player tries to shoot all other players in the game to become the last man standing. This mode is highly flexible as it has no defined team bases to attack or defend. It can operate over a wide range of game arenas, and therefore the time taken to complete it is dependent

on the size of this arena. The game can be preset to terminate after a designated period of time where the winner will be the person with highest number of kills.

3.1.3. Individual battle mode

This mode is a timed free for all battle where no participant is locked out from the game session if shot. The aim is to score as many points as possible in order to acquire advanced equipment. Every time a player shoots an opponent, they score points depending on the experience of the player being shot and the distance from which the shot was fired. Although players are not locked out of the game once shot, they have to wait a recovery period before their weapons are reactivated. This introduces a requirement to either hide or run.

3.1.4. Team battle mode

This battle mode is essentially a team version of the previous mode. Extra points are gained by each winning team member which he can use later to upgrade his/her weapons or armour.

3.2. Game balance

One of the essential elements for successful multiplayer computer games is differentiating players' experiences so that the game is balanced [13]. Balance means that all players feel they have an equal chance of competing, keeping in mind the differentiation between experienced and nonexperienced ones. Interestingly, this feature has not been given much consideration in mixed-reality games, although it likely ensures that players return to the game on multiple occasions. This is addressed in "Mobilazer" by introducing a variety of armours and weapons to allow both new players and more expert ones the opportunity to gain experience and be rewarded for repeated game play.

The system rates a player by the number of points they have collected. Certain points are gained depending on the status of the players they shoot and the distance between the two. These points qualify a player for a higher ranking and enable buying advanced gear each of which has a defined set of points to acquire and a number of hits to destroy as shown in Table 1.

4. GAME ARCHITECTURE

The Mobilazer architecture is composed of: the coloured tags, the mobile phone client application, and the mobile phone game server module as shown in Figure 2.

4.1. Tag design

The coloured tag is the vital entity in the whole system as it identifies players in the game and initiates their interaction between players and the game assets. The tag shown in Figure 2 has been designed in this way to facilitate easy de-

TABLE 1: Mobilazer scoring scheme.

Equipment type	Points scored	Points needed	Number of hits
None	10	—	1
Bronze armour	20	30	2
Silver armour	40	60	5
Gold armour	60	90	10
Sniper	90	90	1
Tracker gun	100	200	1
Dist. ≤ 5 m	2	—	—
5 m < dist. ≤ 10 m	4	—	—
10 m < dist. ≤ 15 m	8	—	—
15 m < dist. < 20 m	12	—	—

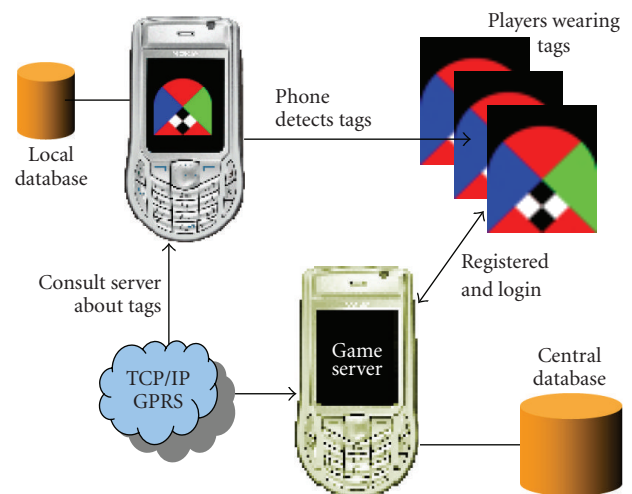


FIGURE 2: Mobilazer architecture.

tection by cameras through the following four features:

- (i) the triangular segments coloured red (top), green (right), and blue (left) which are used by the software to detect the centre point of the tag;
- (ii) the upper black boundary facilitates calculation of the physical distance between players;
- (iii) the black and white code area in the middle that contains the ID of the player or equipment;
- (iv) the lower red two boundary triangles below the code which are used to improve the readability of the ID.

4.1.1. Physical structure

The tag has a large dimension of 20 cm by 24 cm to facilitate detection from longer distances and is made of "felt" fabric that reduces the amount of light reflected of its surface, which causes problems for the tracking software when reading the tag. This is indeed a known issue for phones using the camera in 1D and 2D barcode reading applications.

4.1.2. Triangular segments

The three coloured triangular segments ensure that even when the tag is rotated the vertical distance between the centre point and top of the segments is preserved, which is used in distance calculation. This holds true as long as the whole tag falls within the boundary of the captured camera frame and as long as it is upright and not rotated more than ± 45 degrees from the upright.

4.1.3. Code blocks

The code area in the middle of the tag represents binary data and is capable of representing $2^4 = 16$ codes. The shape of a component block is not that important as tests have shown that it approximates to a square shape (pixels) when viewed from a distance. Rhombuses are used here just to optimise the space available and fit in as much blocks as possible while keeping them big enough to be identified.

4.1.4. Strategic points

There are three critical points in the design of the tag that ensure consistent recognition of the tag ID. The first point is the centre point where the tips of all coloured segments meet. It is necessary when calculating the distance between two players and identifying the upper corner of the code area. The second point is the right corner of the code area and the third is the left corner of the code. All three corners are used to read the ID through interpolation.

4.2. Client application

The client application runs on the phone and is responsible for

- (i) communicating with the game server to inform it of shots, collected items, and synchronisation of players' information;
- (ii) detecting tags, reading their embedded data, and displaying players' status and gear details in a box on screen.

When the Mobilazer client is launched, the player registers his/her nickname (used throughout the game), the team to which he/she belongs (if any), and the number associated with the tags on his/her armour. Each player submits these details to the server through connected TCP/IP sockets over GPRS and waits for acknowledgment to log in. When the client control unit (CCU) verifies that all players are registered it initiates the logging process.

Once in the game, the client detects a tag with a set of optimised image processing routines. It then reads the code on the tag and checks it against the records in its local database (which has been acquired from the server during the registration phase). Using a local database copy is more efficient since it decreases latency, reduces constant switching between the client and the server, and saves costs incurred from mobile network charges of GPRS use. The server will update this

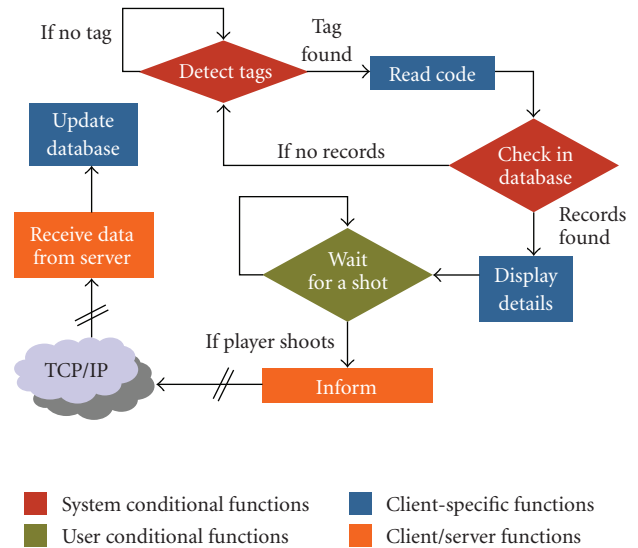


FIGURE 3: Client detection process.

local database regularly once a player's status has changed by pushing the new data to all phones.

When a tag ID is identified during the game, the application shows on phone screen a target sign centred on the tag and an information box containing the nickname of the target player, his/her team, his/her weapons, and his/her calculated distance from the shooter. If the attacker decides to shoot the opponent, the client sends a command to the server over TCP/IP through the connected socket requesting changing that opponent's status to indicate that he/she has been shot. The server then broadcasts this new information to all participants as well to update their local databases. This process is illustrated in Figure 3.

4.3. Game server

The server manages the communication and data exchange between players in a game session. Since the CCU operating the server will be present in the game field, it is used to define to the game server how many players will join a session before commencing the registration process. Then, the registration starts and the server displays on screen a counter of how many players have joined. When all players are registered, the CCU initiates the logging-in process. This two-phase initiation ensures that the server receives details of all players beforehand and then delivered all at once to each participant in the game.

The game server central database contains the nicknames of all players, their mobile phone IMEI numbers, their tag IDs, their groups (if available), their scores, and their acquired equipment. Players are identified uniquely by the IMEI numbers of their phones. Their weapons and scores will be associated with these numbers in the server database. Thus, each player must always use the same phone he used initially if he/she is to carry forward the gear possessed from previous battles to future ones.

5. GAME BALANCE FEATURES

To add depth to the game and, as we have highlighted previously, to aid the balance we have added four features that relate to players' experiences.

5.1. Distance measurement

The Mobilazer client is capable of measuring the distance between any arbitrary two players. The algorithm used to achieve this functionality is related to the tag. Once the centre point of a tag is found the application traverses the pixels above it vertically until it reaches a pixel on the arch between the red semitriangle and the black boundary. The number of pixels iterated r represents the radius length of the triangular. Experiment showed that a tag fills the screen area in Mobilazer when it is at 140 cm distance from the phone. In this case, the number of pixels between the centre point and the arch is 80 pixels. Based on these givens, the following formula gives us the new distance D (in centimetres and meters):

$$D_{cm} = Z \times \frac{80 \times 140}{r} = \frac{11200}{r} \implies D_m = Z \times \frac{112}{r}, \quad (1)$$

where Z is the zooming factor defined in a subsequent paragraph.

5.2. Zooming functionality

The camera API in Symbian OS provides the functionality for camera zooming either digitally or optically. In our test device the Nokia 6630 has a digital zoom of up to 4X. The default zoom value of the API (1X) displays tags too small on screen to be recognised. So the application resets this value to 3.6X which allows players view objects on screen almost in their actual physical sizes.

To allow flexibility and enhance player experience we have added ten zooming options, and all players with all weapons except snipers have this functionality enabled.

Although calculating the distance depends on pixels and tag size on screen, the distance will still adhere to the exact physical distance even when the image is digitally zoomed in or out. This is achieved by preserving the proportions in (1) with the zoom factor Z . As 3.6X is the defaulted zoom value, Z is calculated as follows where z is the new zoom adjusted to the scene by the player. Note that the fraction in the denominator has been simplified to eliminate floating point division which optimises the system for mobile phone processors that are at present based on fixed-point routines:

$$Z = \frac{z}{3.6} = \frac{25 \times z}{45} = 0.534 \times z. \quad (2)$$

5.3. Sniper

The Sniper feature allows a player to shoot tags that are much farther than the normal camera range. The effect is applied using Symbian-specific bitmap manipulation utilities that enlarge and clip images, rather than simply depending on the camera API zooming functions. The image will be enlarged 6 times its size on screen, neglecting any zooming effect, and then clipped to the size of the mobile phone screen



FIGURE 4: Sniper zoom.

to produce a sniper close-up feel. Once this is done a black mask is superimposed on top of the view with sniper-detailed crosshair, as shown in Figure 4, to give the player the impression of looking through a telescopic sight. At start a sniper has 7 shots. The player can reload it either by collecting more points or by encountering an equipment collection point.

5.4. Guided missile

This weapon is capable of locating its target even if it is not in line of sight. The user simply selects his/her enemy from the list of players downloaded from the server, and the guided missile finds its way to the unfortunate player. Since this is the most advanced piece of equipment, it requires the most number of points for a player to acquire and it has a limited number of shots. The process of loading this gun is similar to that of the sniper. Note that in case of "Fortress" mode the owner of a guided missile will not be able to target an opponent team's base remotely.

5.5. Armour shields

Table 1 highlighted the three different types of armours available in Mobilazer: bronze, silver, and gold. Beginner players will be provided the gold armours to provide the maximum protection as they learn how to play the game. As they become more experienced their armour level drops until they reach the point where they join a battle without any protection. However, it is still possible to acquire this protection during the game if a player collects enough points or triggers an equipment collection point.

Any armour can be combined with any other weapon. For example, a player may have a sniper gun and wears a silver armour. In this case the player will endure 5 hits instead of 1 before he is defeated.

5.6. Power-ups and gear collection points

Weapons, armours, and power-ups may be scattered around the battle field for players to collect and charge up. Special coloured tags can be registered in the server to represent such items. The players need to target their phone on the tag in the

same way as opponents and the corresponding item will be loaded to their account automatically once detected. Then, it will be updated in the server and forwarded through to other players. In case of picking up an armour by a player who already has one and has not been hit, the tag will have no effect, that is, the number of allowed hits will not be doubled. However, if the player has some hits, his/her armour will be renewed and hits will be eliminated. If a player picks up a weapon that he/she already has, his/her shots will be doubled.

6. CONCLUSION

This paper has highlighted the design and implementation of the game and as yet we have only completed controlled trials. In the next phase we intend to open up the game to a wide audience and present details of the user experience in a future publication.

With the current lack of innovation in the mobile games market, and in particular the failure to address much wider demographics means that new game genres should be considered for mobile gaming. Mixed-reality games create one such possible genre but it is often hampered by the fact that it demands utilising very high-end handsets with low critical mass. With this in mind we have shown that common camera phones have the potential for resolving this obstacle by creating highly sophisticated and immersive environments, and by being capable of performing complex image processing tasks in real time.

ACKNOWLEDGMENT

The authors wish to acknowledge the support of Nokia for the real-time hardware and software laboratory in Infolab21 at Lancaster University where much of this work was carried out.

REFERENCES

- [1] R. Tercek, "The first decade of mobile games," Keynote at GDC Mobile, San Francisco, USA, March 2007, www.roberttercek.com/.
- [2] P. Coulton, O. Rashid, R. Edwards, and R. Thompson, "Creating entertainment applications for cellular phones," *ACM Computers in Entertainment*, vol. 3, no. 3, p. 3, 2005.
- [3] O. Rashid, I. Mullins, P. Coulton, and R. Edwards, "Extending cyberspace: location based games using cellular phones," *ACM Computers in Entertainment*, vol. 4, no. 1, 2006.
- [4] H. Clemson, P. Coulton, and R. Edwards, "A serendipitous mobile game," in *Proceedings of the 4th Annual International Conference in Computer Game Design and Technology (GDTW '06)*, pp. 130–134, Liverpool, UK, November 2006.
- [5] P. Coulton, R. Edwards, W. Bamford, F. Chehimi, P. Gilbertson, and M. Rashid, "Mobile games: challenges and opportunities," in *Advances in Computers*, vol. 69, Elsevier Press, Amsterdam, The Netherlands, 2007.
- [6] Nokia, "The mobile device market," August 2005, <http://www.nokia.com/>.
- [7] V. Tuulos, J. Scheible, and H. Nyhom, "Combining web, mobile phones and public displays in large-scale: manhattan story mashup," in *Proceedings of the 5th International Conference on Pervasive Computing*, vol. 4480, pp. 37–54, Toronto, Ontario, Canada, May 2007.
- [8] R. Suomela and A. Koivisto, "My photos are my bullets—using camera as the primary means of player-to-player interaction in a mobile multiplayer game," in *Proceedings of the 5th International Conference on Entertainment Computing (ICEC '06)*, pp. 250–261, Cambridge, UK, September 2006.
- [9] S. Bucolo, M. Billinghurst, and D. Sicking, "User experiences with mobile phone camera game interfaces," in *Proceedings of the 4th International Conference on Mobile and Ubiquitous Multimedia (MUM '05)*, vol. 154, pp. 87–94, Christchurch, New Zealand, December 2005.
- [10] M. Rohs, "Visual code widgets for marker-based interaction," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW '05)*, pp. 506–513, Columbus, Ohio, USA, June 2005.
- [11] R. Ballagas, J. Borchers, M. Rohs, and J. G. Sheridan, "The smart phone: a ubiquitous input device," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 70–77, 2006.
- [12] J. Coughlan, R. Manduchi, M. Mutuzaki, and H. Shen, "Rapid and robust algorithms for detecting colour targets," in *Proceedings of the 10th Congress of the International Colour Association (AIC '05)*, Granada, Spain, May 2005.
- [13] C. Bateman and R. Boon, *21st Century Game Design*, Charles River Media, Rockland, Mass, USA, 2005.