Augmented Reality Games on the iPhone

What are some of the possibilities and problems associated with the creation of augmented reality games for the iPhone?

Amanda Rösler
Bachelor Thesis • Blekinge Institute of Technology • Spring 2009

Contact Information

Author

Amanda Rösler
Alice Tegnérs Väg 18
374 35 Karlshamn
amanda.rosler@gmail.com

University Advisor

Hans Tap
School of Computing (COM)
Blekinge Institute of Technology
hans.tap@bth.se

Abstract

Augmented reality opens up a lot of possibilities for new types of games, where the real and the virtual world are mixed. Despite this, augmented reality games are still not very common, probably due to the fact that head-mounted displays (which are often used for augmented reality) are expensive. However, in recent years mobile phones have become more and more powerful, and since many of them have built-in cameras and rather large screens, they are a potential platform for augmented reality games.

This thesis explores some of the problems and possibilities associated with the creation of augmented reality games for the Apple iPhone. In order to do that, a multi-player augmented reality game for the iPhone was implemented, and then a number of performance tests and a user study were conducted.

The most important conclusion that was reached is that performance is a definite problem when creating augmented reality games for the iPhone.

Table of Contents

1. Introduction

1.1 Background

Augmented reality is the integration of computer generated data with the real world, which for example can be done by rendering computer graphics on top of real-time footage. Augmented reality can be used for many different things, such as displaying driving directions to a person sitting in a car with a head-up display, helping a doctor by inserting extra information into his or her field of view (such as an X-ray of the patient), or virtually restoring old, historic buildings to what they once were.

Augmented reality is often associated with head-mounted displays, which are also often used for virtual reality. A head-mounted display is a display that is mounted on the user's head, so that the display is positioned in front of the user's eyes. Modern versions of these look similar to goggles or even regular glasses. However, head-mounted displays are still not commonly available, and therefore more and more people are exploring other platforms for augmented reality, such as mobile phones.

One interesting application of augmented reality is *gaming*. Augmented reality opens up a lot of possibilities for fun, immersive or realistic games where the player can interact with the game through his or her environment. Games for mobile devices can also take gaming to a whole new level, by making the player move around instead of just sit in front of a computer screen. Some examples of mobile augmented reality games (using head-mounted displays or mobile phones) can be found in Appendix A.

One mobile phone that seem very suitable for augmented reality games is the Apple iPhone, since it has a number of features that could be used in such a game. Some of them are:

- A relatively large screen (3.5 inches)
- A 2.0 megapixel camera
- Multi-touch
- GPS
- An accelerometer

1.2 Research Question

Mobile phones have several disadvantages compared to head-mounted displays and portable computers: they are not as powerful, they have smaller screens, and they require that the user holds the phone in front of his or her eyes while using the application, which might be tedious or might make the gaming experience less immersive. Despite those disadvantages, mobile phones are the next best thing to head-mounted displays, and therefore this thesis aims to explore what possibilities and problems come with the creation of augmented reality games for mobile phones, specifically the iPhone. The chosen research question is:

What are some of the possibilities and problems associated with the creation of augmented reality games for the iPhone?

1.2.1 Subquestions

- How could the iPhone's different sensors be used to enhance an augmented reality game?
- What restrictions does the iPhone's hardware and software put on an augmented reality game?
- How suitable is the iPhone as a platform for augmented reality gaming?

1.3 Methodology

In order to answer the research question, the following steps were taken:

1. Implementation an augmented reality game for the iPhone.
2. Conduction of five sets of performance tests where the frame rate in the game was measured. The tests were made with different features enabled and disabled each time, in order to find possible bottlenecks.
3. Conduction of a user study where four users have tested the game and answered a number of questions about user interaction and performance.

1.4 Delimitation

This thesis will not address other uses of augmented reality than its use in games.

1.5 Target Audience

Game programmers interested in augmented reality.

2. Theoretical Background

2.1 Introduction to Augmented Reality

Augmented reality is a way of mixing the real and the virtual world. Virtual elements are added to the real world, in order to enhance it or to add more information to it. In order to do this, one can connect a video camera to a computer and render virtual images on top of the video feed, so that the user can view the real world, with the added computer graphics, on the computer screen.

One common definition of augmented reality states that an augmented reality system has to fulfill the following requirements (Henrysson 2007):

- It combines the real and the virtual
- It is interactive in real time
- It is registered in three dimensions

The first requirement merely states that an augmented reality system has to combine the real and the virtual world, which can be done by rendering computer graphics on top of a video stream. The second requirement states that the system has to be interactive in real time, which practically means that it has to be supplied with a live video stream and update at least five times per second (Henryson). There also has to be a way to interact with the system in real time. The third requirement specifies that the virtual data has to be registered in three dimensions, which means that it has to be positioned inside the real, three dimensional world (even if the data itself is represented in two dimensions, such as text or 2D images).

2.2 Approaches to Augmented Reality

2.2.1 Head-Worn

One way to create an augmented reality system is to make the user wear a head-mounted display, which can be either semi-transparent or not transparent at all. *Semi-transparent* means that the wearer can see through the display, but also see computer graphics that are rendered on the display. A head-mounted display that is not transparent requires that the wearer also has a video camera and that the video stream from the camera is displayed on the screen, so that the wearer can see the real world on the display. The camera is usually positioned on top of the user's head. Apart from a head-mounted display, the user also needs to carry a computer, usually in a backpack.

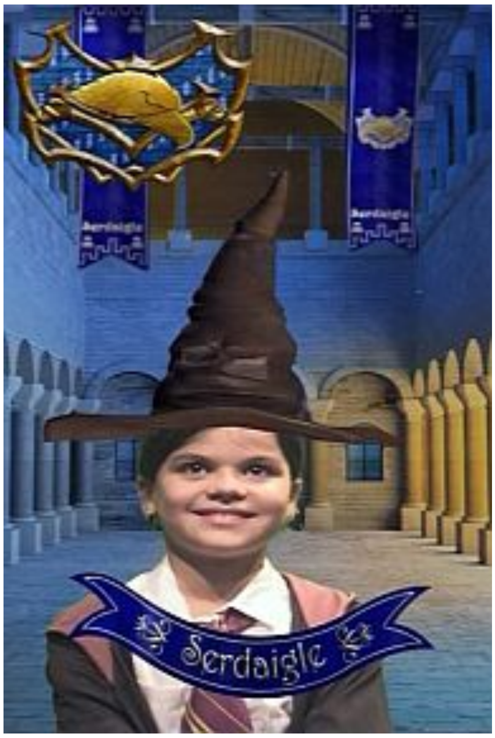
The concept of making the user of an augmented reality system see the real world through a display with added virtual elements on top of it, is often called the *magic lens* metaphor (Cawood & Fiala 2007).

An example of the magic lens metaphor. The user, who is wearing a head-mounted display, sees a virtual woman in the real-world graveyard.
(Image courtesy of David Stuart and Smalldog Imageworks)

2.2.2 Stationary

An augmented reality system can easily be set up with a stationary computer, a monitor and a webcam. This type of set up can use either the magic lens metaphor, or the magic mirror metaphor. The *magic mirror* metaphor is a technique that instead of making the user see through the display, makes the user see him or herself in the display, as if it were a mirror (Cawood & Fiala 2007). This can for example be achieved by putting the webcam on top of the monitor and pointing it at the user.

The webcam is pointing at a paper with a symbol on it, and the computer screen displays a pot plant on top of the symbol. This is an example of the magic mirror metaphor.
(Image courtesy of olliebray.com)



Another example of the magic mirror metaphor. A girl sees herself inside Hogwarts, wearing a Harry Potter school uniform.
(Image courtesy of Total Immersion)

2.2.3 Handheld

Nowadays. a lot of mobile phones have built-in cameras and rather large screens, which is exactly what is needed for an augmented reality system. Of course, mobile phones don't have as much processing power as regular computers, so there are limits to what one can do with them.

Mobile phones usually have a camera on the back, pointing away from the user (if he or she is looking at the screen). That makes mobile phones suitable for the magic lens technique. However, there are some mobile phones and PDAs that also have a camera on the front, which would make it possible to use the magic mirror technique.

An example of the magic lens metaphor on a mobile phone.

(Image courtesy of Graz University of Technology)

2.3 Tracking Methods

As Daniel Wagner states in his thesis 'Handheld Augmented Reality', any augmented reality system needs to track the position and orientation of the user, so as to know where the user is in regard to the real world (Wagner 2007). This is usually done by vision-based tracking methods, though lots of other methods have been tried, such as mechanical and magnetic tracking. Vision-based tracking has proved to be the method with the most potential, since it is both accurate and doesn't cost a lot of money (Haller, Billinghurst & Thomas 2007). The two main types of visual-based tracking are *natural feature tracking* and *marker tracking*.

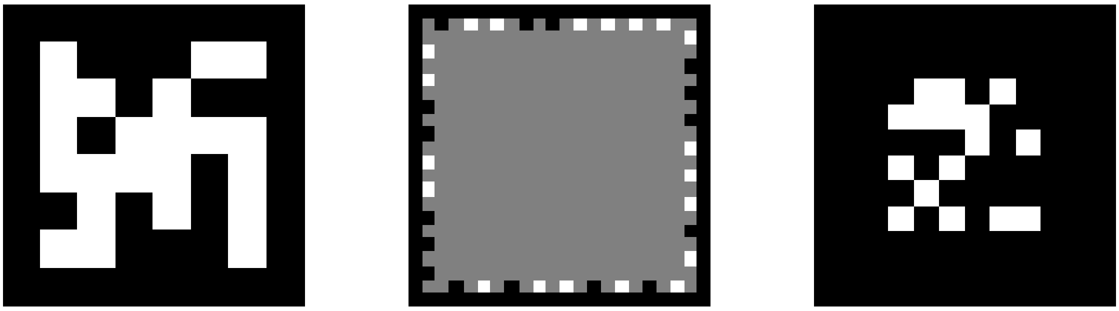
2.3.1 Natural Feature Tracking

Natural feature tracking is a method for recognizing natural features in the environment, such as edges and textures. This is done by comparing objects in the video stream to known shapes, colors or textures.

The main problem with natural feature tracking is that it requires a lot of processing power, and is therefore not suitable for use on mobile phones (Wagner 2007).

2.3.2 Marker Tracking

Another way of tracking the user's pose is by using so called *fiducial markers*. Fiducial markers are easily-recognized objects that are placed throughout the scene in the real world, and can be used as a point of reference in the video stream. It is common to use two-dimensional barcodes (so-called data matrix codes) as fiducial markers, since they are designed in such a way that they are easy to find and identify in a video stream. Each marker is completely unique, and can be recognized even if it has been rotated. The pattern on the marker usually represents an ID number, though if using a high density pattern, one can store even more data directly in the marker (Henrysson 2007).



Three fiducial markers

Using markers makes it much faster and easier to determine the user's pose, and it is therefore the best solution for augmented reality on mobile phones (Wagner 2007). The downside is, of course, that the scene has to be prepared with markers in advance, and that the user is restricted to the marker-prepared area.

3. The iPhone's Sensors and How They Could Be Used in an Augmented Reality Game

3.1 Overview

The iPhone has five sensors: a multi-touch screen, an accelerometer, a GPS sensor, a light sensor and a proximity sensor. These could be used in various ways to make interaction with an augmented reality game more intuitive, to determine the user's pose or to add interesting features to the game that it would not have been possible to add if the game had been implemented for a stationary computer.

3.2 Multi-Touch

The iPhone's multi-touch screen is the user's main way of interacting with the phone, since the iPhone has very few physical buttons. For example, the user can tap on icons on the screen to select them, make images zoom in or out by pinching them with two fingers, drag things on the screen or scroll up and down. The multi-touch screen makes it possible to design specific finger gestures that can be used to interact with an application. Some possibilities include:

- Moving virtual objects in the scene by tapping on them and dragging them.
- Enlarging/shrinking virtual objects by tapping on them and then pinching in and out.
- Drawing symbols on the screen, with one or several fingers, to trigger specific actions.

3.3 Accelerometer

The iPhone has an accelerometer that can detect movement on the x-, y- and z-axis. It is used mainly for detecting when the user turns the iPhone upside down (so that the image on the screen can be rotated accordingly). However, a lot of games also use it to control movement of objects in the game. For example, in one game the player guides a ball through a maze by tilting the phone, and in another the player controls a racing car by tilting the phone left or right. Several drawing applications allow the user to shake the phone to clear the screen, or to undo the last action. In an augmented reality game, the accelerometer could be for example be used to:

- Connect different game modes to the directions the phone is pointing in. For example, when the user points the phone towards the ground, the menu could open, and when he/she holds it upright again, the game could resume. This could be appropriate to use in a action-based game, where one needs to switch between game modes quickly.

- Determine the direction of the gravitation, in order to align 3D objects with the ground.

- Trigger specific actions when the user moves the iPhone in a certain, pre-determined way (for example moving the phone quickly up and down three times, shaking the phone, or "slashing" with the phone as if it were a sword). One advantage of using this method is that the user would be able to interact with the game without having to look at the screen.

3.4 GPS

Like many other mobile phones, the iPhone has a built-in GPS sensor. This could be used to:

- Calculate the distance between two players.

- Track the player's position, so that the game can notify the player or trigger specific actions when he/she enters a certain area. It could also be used to notify the player if he/she accidentally leaves the game area.

- Display the players' positions on a map.

3.5 Light Sensor

The iPhone's light sensor is mainly used for adjusting the brightness of the screen in relation to the brightness of the user's environment. However, in an augmented reality application, it might also be possible to use the light sensor in order to make the lighting of 3D objects in the scene more realistic.

3.6 Proximity Sensor

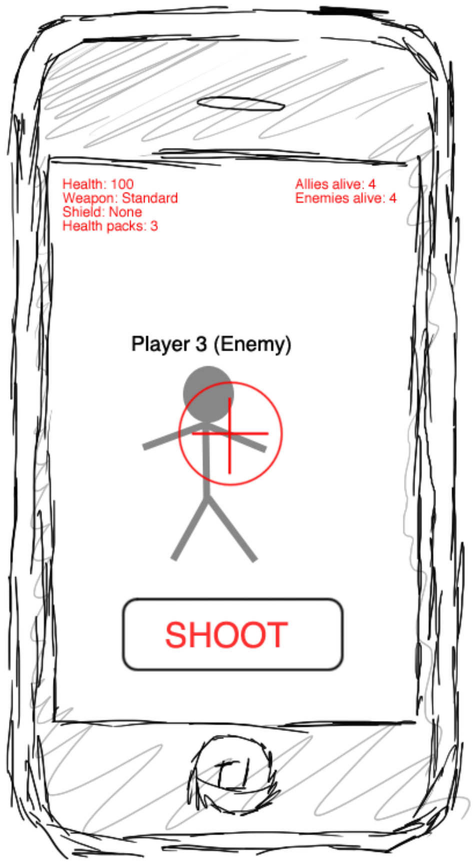
The proximity sensor is used to turn off the iPhone's screen when the user answers a call and holds the phone to his/her ear, in order to save battery life and to make sure that the user doesn't accidentally touch any buttons on the screen. This sensor is not often used for anything other than that, in fact, there is only one application on the App Store (Apple's online store for iPhone applications) that does. The application, which is called *One Tap Trick*, is a card trick game that uses the proximity sensor to "magically" change the card on the screen when the user holds his hand close to the sensor.

It would probably not be a good idea to use the proximity sensor in an augmented reality game since it is very easy to cover up the sensor by accident, especially if one holds the phone horizontally.

4. Implementation of an Augmented Reality Game for the iPhone

4.1 Description

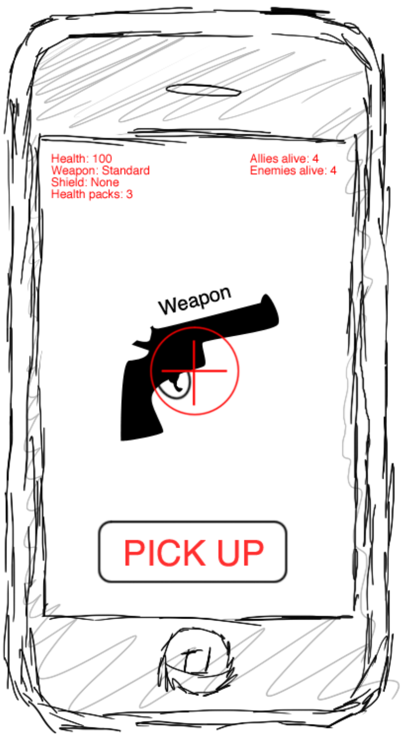
In order to answer the research question, a simple augmented reality game for the iPhone was implemented. The game supports 2 or more players, ideally 4-8. The players are divided into two teams that compete against each other.



Each player has a large fiducial marker on his/her front and back. The teams' objective is to "shoot" at the other team, by pointing the iPhone's camera at a player's marker and pressing a button on the screen. Since the game continuously scans all markers in its field of view, it can display information about the other players on the screen in real time. When a player has been hit, his/her health points are reduced. The game is over when one of the teams has lost all of its players.

There are also fiducial markers representing items such as health packs, weapons and shields, placed at strategic locations in the game area. When a player points his/her iPhone at one of these markers, it scans the marker and displays a 3D rendering of the item. The player can then pick up the item by pressing a button on the screen.

To use an item that the player has picked up, the player has to make a certain movement with the iPhone, which is detected by the iPhone's accelerometer. The point of this is to make it easier and faster for players to activate a shield or use a health pack when in danger. For example, if the player is running from the enemy, he or she can activate a shield with a quick hand movement, without having to look down at the screen.



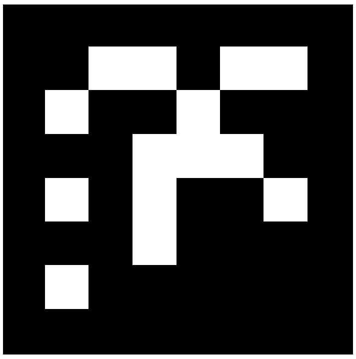
Health packs are used by quickly moving the iPhone horizontally from left to right, and back again, three times. Shields are activated by quickly moving the iPhone up and down three times. The game automatically chooses the best weapon that the player has, so that one doesn't have to spend time on choosing weapons.

All information about the players, such as names, positions, health points and items they have picked up, is stored on a remote server. This information is sent to the clients once every second, so that the game can display accurate information about other players and items when the player scans their markers.

4.2 Implementation

The game was written in Objective-C. OpenGL ES, which is a subset of OpenGL made for mobile phones and other embedded devices, was used for 3D rendering.

For marker tracking and pose estimation, a software development kit called Studierstube Tracker (StbTracker) was used. StbTracker was developed specifically for mobile phones, and has therefore relatively high performance and low memory requirements. It supports several different types of markers; simple binary markers (pictured to the right), each containing an ID number, were used in this project.



5. Testing

5.1 Performance

5.1.1 Overview

A series of performance tests were conducted in order to find possible bottlenecks in the augmented reality game. The tests were conducted on a 1st generation iPhone, which has a 620 MHz processor (underclocked to 412 MHz) and 128 MB RAM. In each test, the number of frames per second were measured while pointing the camera at a marker. In the last test, however, the camera was turned off completely after having captured one image of the marker, then the application ran the tracking code on that single image over and over again.

The test has been performed three times for each different set of settings.

5.1.2 Test A - All Features Enabled

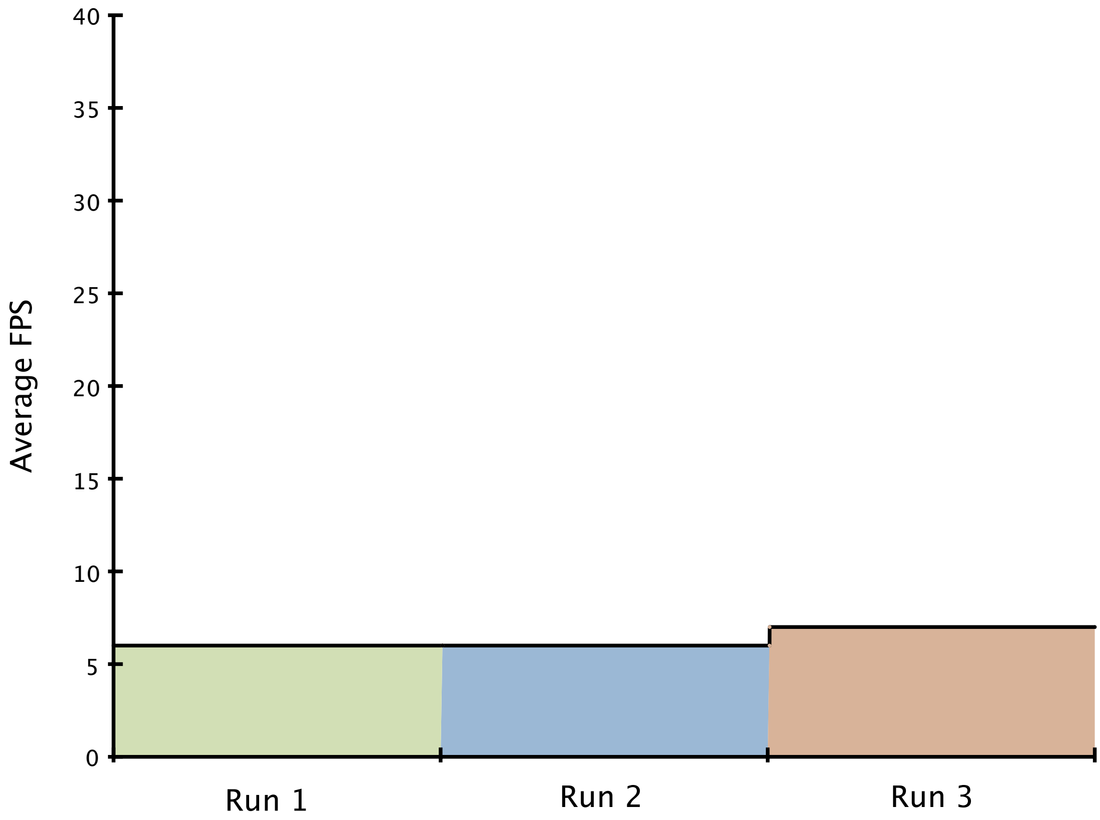
Test duration: approximately 1 minute

Marker tracking: enabled

Video capture: enabled

3D-rendering: enabled

Network: enabled



5.1.3 Test B - Network Disabled

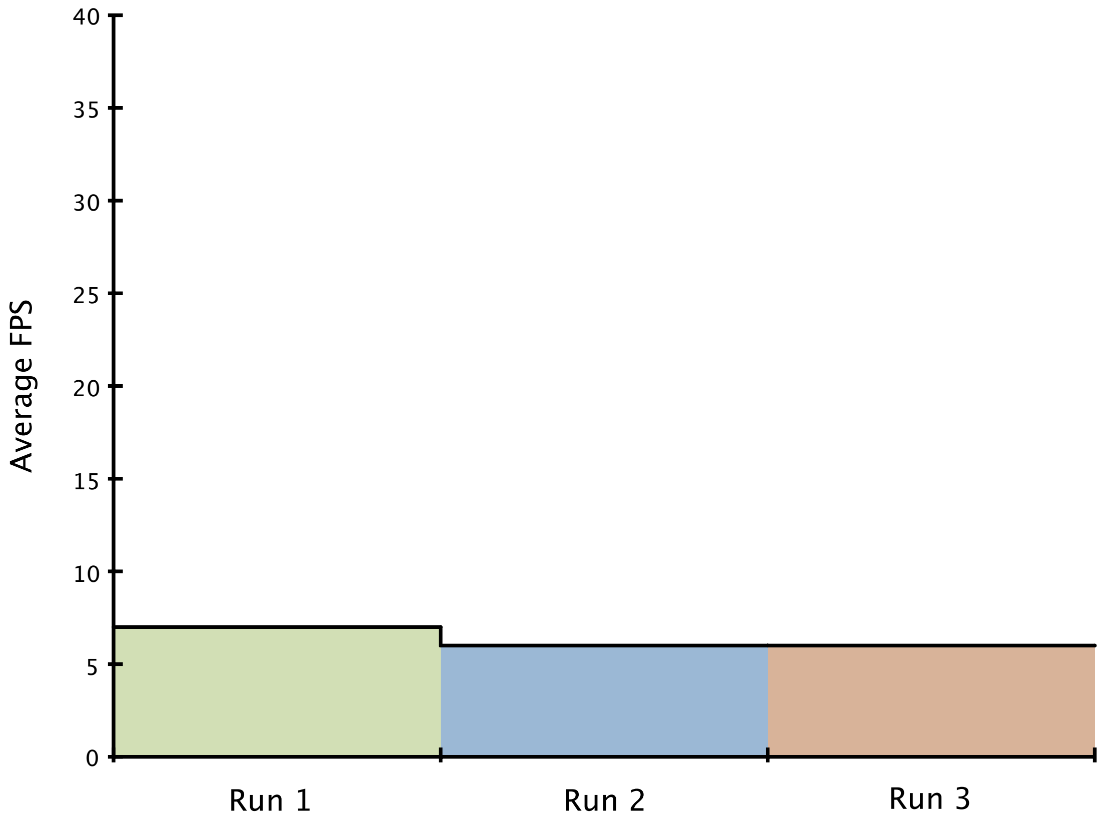
Test duration: approximately 1 minute

Marker tracking: enabled

Video capture: enabled

3D-rendering: enabled

Network: **disabled**



5.1.4 Test C - Network And 3D-Rendering Disabled

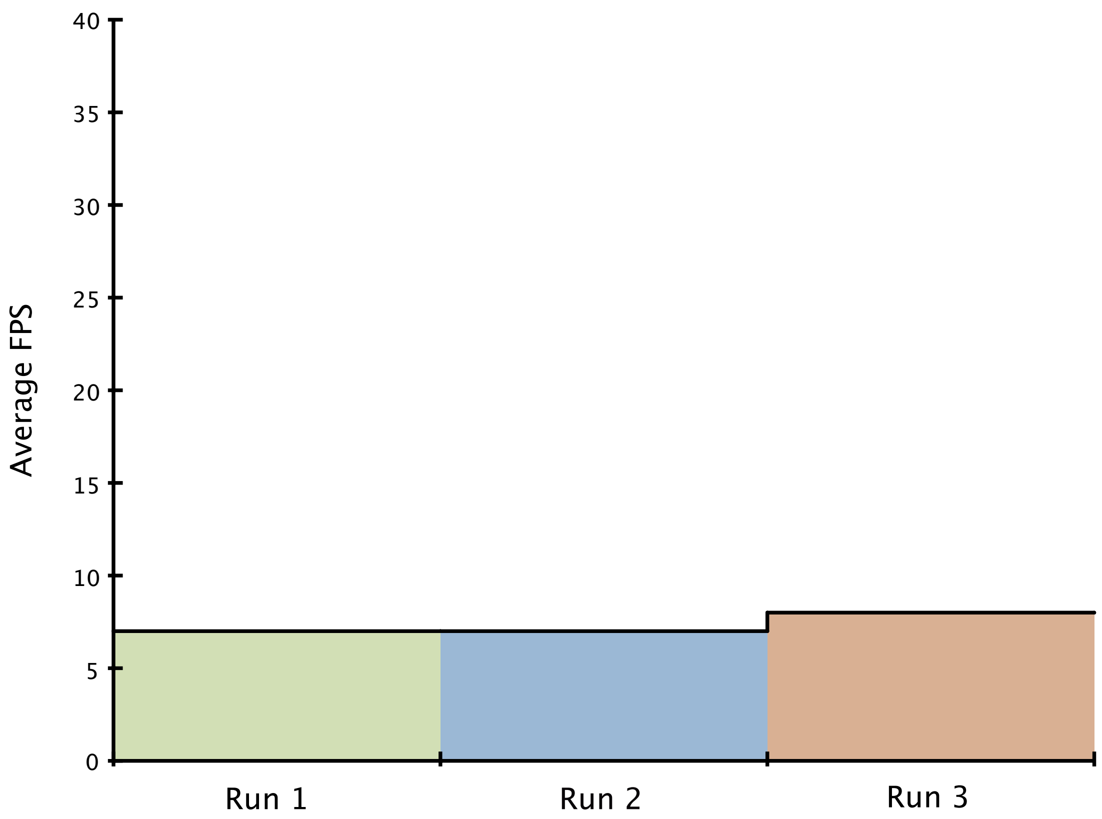
Test duration: approximately 1 minute

Marker tracking: enabled

Video capture: enabled

3D-rendering: **disabled**

Network: **disabled**



5.1.5 Test D - Marker Tracking Disabled

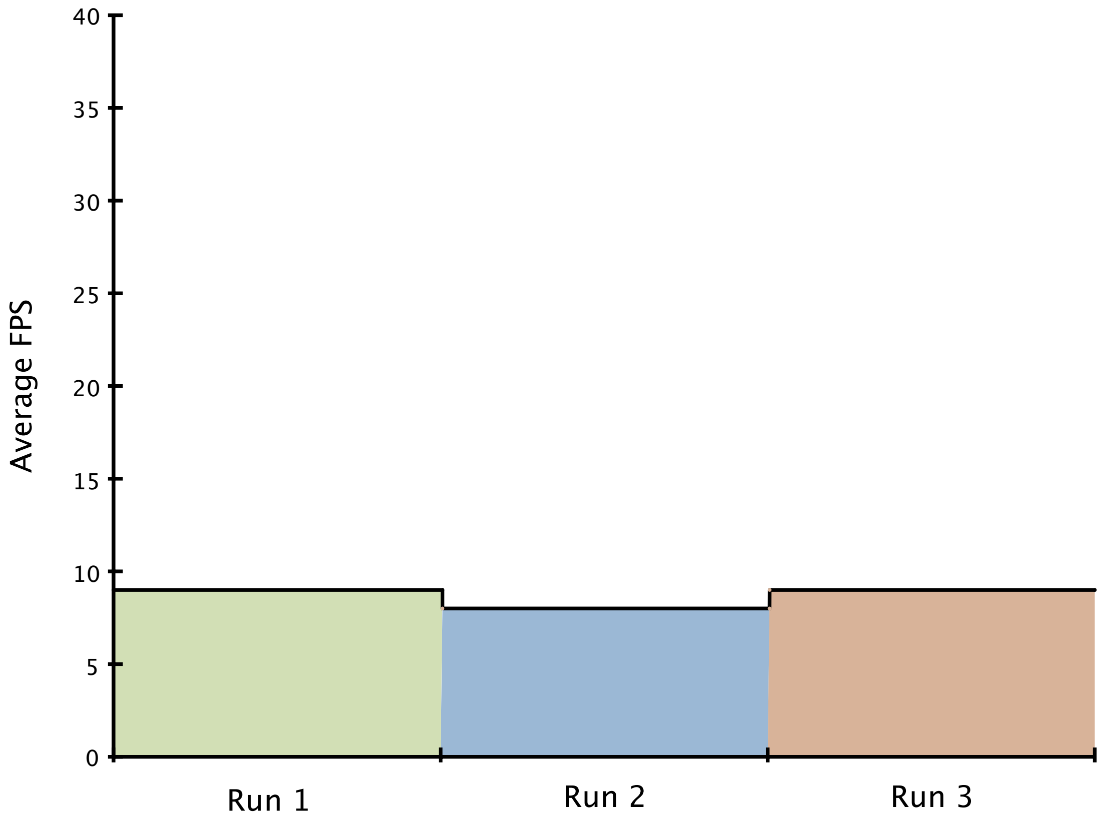
Test duration: approximately 1 minute

Marker tracking: **disabled**

Video capture: enabled

3D-rendering: enabled

Network: enabled



5.1.6 Test E - Video Capture Disabled

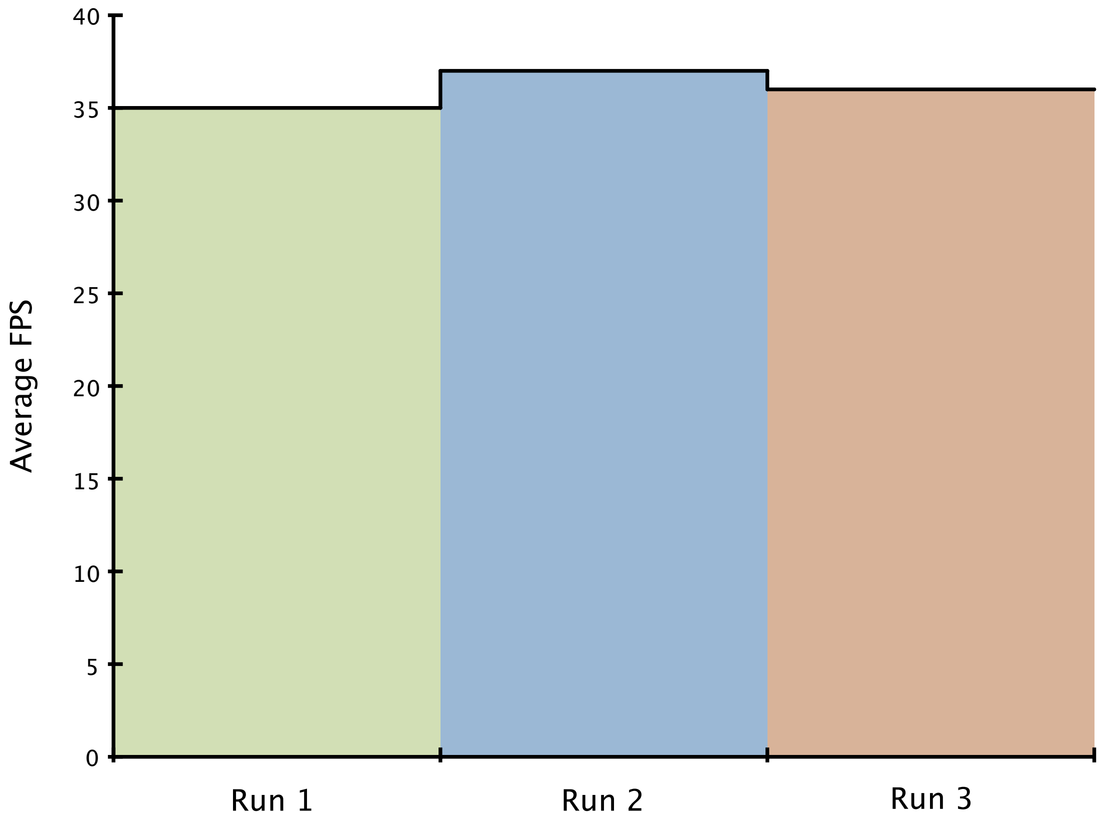
Test duration: approximately 1 minute

Marker tracking: enabled

Video capture: **disabled**

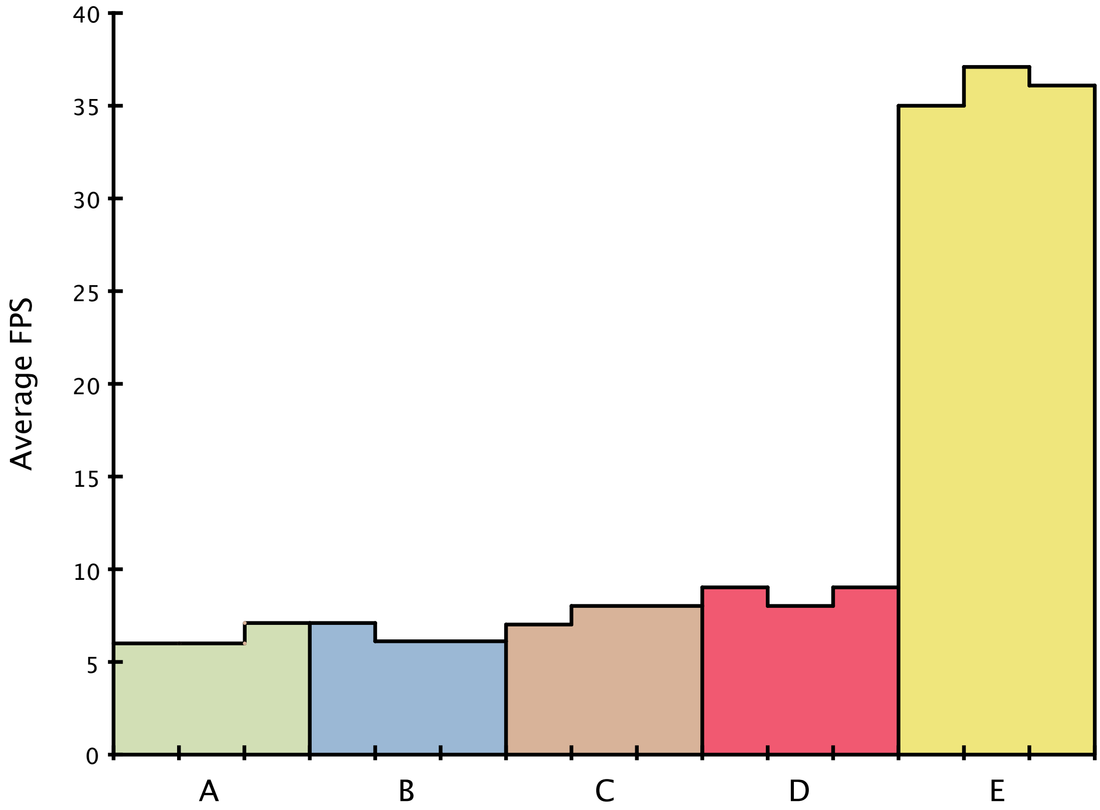
3D-rendering: enabled

Network: enabled



5.1.7 Summary of Test Results

Test A shows that the frame rate when all features are enabled is about 6 fps (frames per second). Tests B, C and D show that disabling 3D-rendering, updates via the network or marker tracking makes very small improvements (if any) to the frame rate. Test E shows that disabling video capturing increases the frame rate to about 36 fps.



5.2 User Study

5.2.1 Overview

A small user study, where four people tried out the game and then answer a number of questions about performance and user interaction, was conducted. The point of the study was to find out how well an augmented reality game works on the iPhone, in terms of performance, screen size and user interaction.

5.2.2 Questions

1. Did you find it difficult to shoot at other players or pick up items because the game couldn't recognize the markers?

    A. Yes, always

    B. Yes, sometimes

    C. No

2. What did you think about the responsiveness of the camera?

    A. Good

    B. Acceptable

    C. Not acceptable

3. Do you think that the iPhone's screen is too small for a game like this?

    A. Yes

    B. No

4. Did you like that items are used by shaking the phone, or would you have preferred to have a menu where you could choose to use an item?

    A. I liked it

    B. I would have preferred to have a menu

    C. I didn't like it, but I wouldn't have liked to have a menu either

5. Did you ever use an item by accident during the test run?

    A. Yes, several times

    B. Yes, once

    C. No

5.2.3 Answers

Question	User #1	User #2	User #3	User #4
1	C	C	B	B
2	B	B	B	B
3	B	B	B	B
4	A	A	B	A
5	C	C	C	C

5.2.4 Summary of Test Results

Half of the testers thought that it was sometimes difficult to shoot at other players because the game couldn't recognize the markers, and the other half didn't think it was difficult at all.

All testers thought that the responsiveness of the camera was acceptable, but not good.

All testers thought that the iPhone's screen is big enough for an augmented reality game.

Three of the testers liked shaking the iPhone to activate items, the other tester would have preferred to activate items through a menu.

None of the testers used any items by accident.

6. Discussion

6.1 Performance

The performance tests show that the game runs at about 6 frames per second, and that video capturing is the worst bottleneck. In fact, when video capturing was disabled, the game ran at about 36 fps, which is six times faster. When disabling marker tracking, the game only ran slightly faster than when all features were enabled, at about 9 fps.

Daniel Wagner, one of the creators of StbTracker, has tested the performance of StbTracker on several mobile phones, and presents the results in his thesis 'Handheld Augmented Reality'. According to his results, the Palm Treo700W and the Motorola Q, which both have 312 MHz processors, could track a single marker at 40,2 respectively 33,2 fps (Wagner 2007). Wagner also concludes, in another test, that the tracking performance is proportional to the speed of the processor. The reason that the Palm Treo ran at a higher frame rate than the Motorola Q, even though their processors are equally fast, is likely due to the fact that the Palm Treo has faster video capturing than the Motorola Q (Wagner 2007).

The results presented by Wagner raises the question: why does the game implemented for this thesis run at a much lower frame rate than Wagner's application? As shown in Test C (page 21), when everything but the marker tracking and video capturing was disabled and a single marker was tracked, the fps was only about 7, as opposed to Wagner's 40 and 33 fps. The iPhone's processor is faster than the Palm Treo's and Motorola Q's processors, so logically, it should be able to run faster than them.

One possible explanation for this could be the size of the iPhone's screen. The iPhone's screen has 480x320 pixels, and is thereby larger than the Palm Treo's 240x240 pixels screen and the Motorola Q's 320x240 pixels screen. The larger screen size could make video capturing and marker tracking slower, since there is more data to process. However, when the video image was cropped to 192x256 pixels, the game ran at about 10 fps, so this can't be the only explanation.

Wagner writes that on all the mobile phones used in the test, video capturing is done in a separate thread (Wagner 2007). This is *not* the case on the iPhone. There is currently no official support for video capturing in the iPhone SDK, and therefore the camera image had to be manually fetched from the camera every frame, which was done in the main thread. If this had been done in a separate thread, and if Apple had provided a faster way of retrieving the video stream from the camera, it might have been possible to achieve a higher frame rate.

Despite the low frame rate, the game testers thought that the responsiveness of the camera was "acceptable". This shows that, with better video capturing support or a faster CPU, the iPhone could definitely be suitable for an augmented reality game, at least performance wise.

6.2 User Interface and Interaction

Most of the game testers agreed that using the accelerometer in order to interact with the game worked well, though one of them argued that shaking the iPhone made it impossible to aim at an enemy at the same time as you used an item. This is true, but adding more user interface elements would obscure the video stream and make it harder to scan the enemies' markers, so in that sense it might be better to lose one's target for a few seconds.

All testers agreed that the size of the iPhone's screen is large enough for a game like this, which is a good sign since it is not probable that any future version of the iPhone will have a larger screen, seeing that it is already quite large compared to other mobile phones.

Unfortunately, there was not enough time to thoroughly investigate the accelerometer, let alone the other sensors. Also due to lack of time, the user study was very small and consisted of only four users, so any results from the study can only be seen as a possible indication of what kind of results a larger study might yield.

7. Conclusions

The tests in Chapter 5.1 show that the biggest issue with augmented reality games on the iPhone is performance. The frame rate in the game implemented in Chapter 4 was acceptable, but not more. The reason for this is likely due to the lack of support for video capturing on the iPhone, but that theory has not been tested. However, performance might not be as big of an issue in a slower paced game; a fast-paced game naturally requires higher responsiveness.

The user study conducted in Chapter 5.2 shows that at least one of the iPhone's sensors, the accelerometer, can be used effectively as a way to interact with the game. One can also conclude that the size of the iPhone's screen is acceptable, at least for the type of game that was implemented for this thesis.

8. Future Work

In this thesis, visual-based tracking with fiducial markers was used in order to determine the user's pose. However, since the iPhone has both an accelerometer and a GPS sensor, and the next hardware update is likely to include a built-in compass, one could explore using these for pose estimation instead.

References

Augmented Environments Lab, 'Art of Defense: a Mobile AR Game with Sketch-Based Interaction and Dynamic Multi-Marker Building Techniques', in *Augmented Environments Lab*, accessed 2 May 2009, from <http://www.augmentedenvironments.org/lab/research/handheld-ar/artofdefense/>

Cawood, S & Fiala, M 2007, *Augmented Reality: A Practical Guide*, Pragmatic Bookshelf, Raleigh, N.C.

Haller, M, Billinghurst, M & Thomas, B 2007, *Emerging Technologies of Augmented Reality*, libris.kb.se, accessed 24 April 2009, from <http://libris.kb.se/bib/10550996>

Henrysson, A, Billinghurst, A & Ollila, A 2006, 'AR Tennis', in *HIT Lab NZ*, accessed 2 May 2009, from <http://www.hitlabnz.org/wiki/AR_Tennis>

Henrysson, A 2007, 'Bringing Augmented Reality to Mobile Phones', Dissertation, Department of Science and Technology, Linköping University.

Mulloni, A 2007, 'A collaborative and location-aware application based on augmented reality for mobile devices', Master Thesis, Facoltà di Scienze Matematiche Fisiche e Naturali, Università degli Studi di Udine

Sandhana, L 2005, 'Pacman comes to life virtually', in *BBC News*, accessed 2 May 2009, from <http://news.bbc.co.uk/1/hi/technology/4607449.stm>

Ten, S 2008, 'AR Tower Defense' , in *Mobile Augmented reality*, accessed May 2 2009, from <http://www.cellagames.com/artd.html>

Thomas, B, Close, B, Donoghue, J, Squires, J, De Bondi, P, Morris, M & Piekarski, W 2000, 'ARQuake: An Outdoor/Indoor Augmented Reality First Person Application', *4th Int'l Symposium on Wearable Computers*, Atlanta, USA, October, 2000, pp. 139-146.

Wagner, D, Pintaric, T & Schmalstieg, D 2005, 'The Invisible Train', in *Studierstube Augmented Reality Project*, accessed 2 May 2009, from <http://studierstube.icg.tu-graz.ac.at/invisible_train>

Wagner, D 2007, 'Handheld Augmented Reality', Dissertation, Institute for Computer Graphics and Vision, Graz University of Technology.

Appendix A - Examples of Augmented Reality Games for Mobile Devices

ARQuake

ARQuake is a project made by the Wearable Computing Lab of the University of South Australia. Basically, they have turned the first-person shooter game *Quake I* into an augmented reality game, where the player, who wears a head-mounted display and a portable computer, runs around and shoots at virtual monsters. The player sees the real world through a semi-transparent head-mounted display, and the *Quake* elements, such as enemies and weapons, overlapped onto it. The game also has multiplayer support, just like in the real *Quake* game. It can be played outdoors on the university's campus, which has been mapped and made into a *Quake* level. The player's position and direction is determined with GPS and an orientation sensor, so that the game knows exactly where in the level the player is, and can align the *Quake* world with the real world (Thomas, Close, Donoghue, Squires, De Bondi, Morris & Piekarski 2000).

The player

(Image courtesy of the University of South Australia)

What the player sees

(Image courtesy of the University of South Australia)

Human Pacman

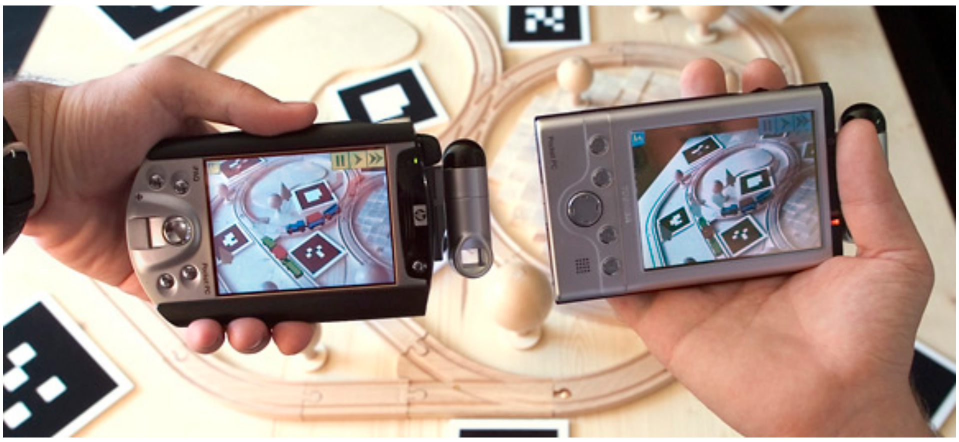
Human Pacman is an augmented reality game, where one player takes the role of Pacman who has to "eat" yellow cookies that are rendered on top of the real world, and the other player is a ghost that has to chase Pacman down. The players wear head-mounted displays and computers on their backs. Pacman eats the cookies by walking through them, and the ghost catches Pacman by tapping him on his back (Sandhana 2005).

This is what Pacman sees. The yellow dots are the virtual cookies that he's trying to eat.
(Image courtesy of BBC News)

Pacman (to the right) and the ghost (to the left).
(Image courtesy of BBC News)

The Invisible Train

The Invisible Train is a multi-player augmented reality game for PDAs (handheld computers). The game board consists of a miniature railroad track and fiducial markers, in the real world. When seen through a PDA, there are trains on the railroad tracks. The players can guide the trains and change their speed through their PDAs, in order to prevent the trains from colliding with each other. The game was implemented with the help of the Studierstube framework (Wagner, D, Pintaric, T & Schmalstieg, D 2005).



The Invisible Train
(Image courtesy of Vienna University of Technology)

Cows Vs. Aliens

Cows Vs. Aliens is a multi-player game for mobile phones, where the players are divided into two teams that compete against each other. A number of fiducial markers are put up on the walls of a room, and when the players look at the markers through their phones' cameras, they see either cows grazing on a meadow, or a stable where the cows are safe. Each team has its own stable, and when the aliens start invading and shoot down cows, the players have to move the cows to their team's stable through pathways that connect the different locations. The team that manages to save the most cows wins (Mulloni 2007).

Cows Vs. Aliens

(Image courtesy of Graz University of Technology)

The game uses the Studierstube ES framework to recognize the fiducial markers and render 3D objects on top of the markers (Mulloni 2007).

AR Tennis

AR Tennis is a two-player game for mobile phones. The players sit at the opposite ends of a table with fiducial markers on it. Their mobile phones render a tennis court and a tennis ball on top of the markers. Each player then has to hit the ball to make it fly back and forth between them, by moving his or her phone so that it "collides" with the virtual tennis ball (Henrysson, Billinghurst & Ollila 2006).

The game is implemented using a port of ARToolKit for Symbian (Henrysson *et al* 2006).

AR Tennis
(Image courtesy of Eliot Phillips)

AR Tower Defense

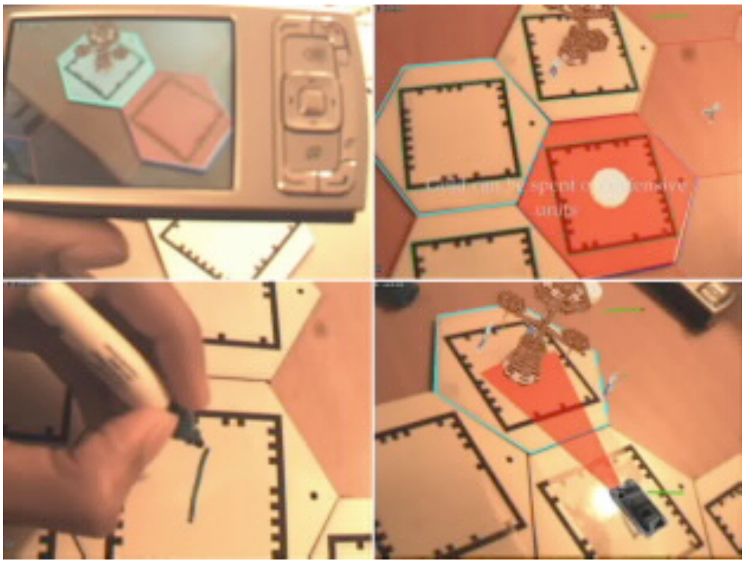
AR Tower Defense is an augmented reality, real-time strategy game for one player. It is available for Symbian phones.

AR Tower Defense
(Image courtesy of cellagames.com)

The game uses a number of fiducial markers that are put on a flat surface, to determine the camera's position. To interact with the game, the player points the camera at the game's virtual towers and other buildings (Ten 2008).

Art of Defense

Art of Defense is another tower defense game for mobile phones (Nokia N95), though very different from *AR Tower Defense*. The game board consists of a number of erasable tiles with laser-engraved fiducial markers on them. The tiles can be combined to form a larger game board, and the player can draw things on them to build and upgrade towers. For example, the player can draw a triangle on a tile to build a tower there, and a circle around it to upgrade it (Augmented Environments Lab).



Art of Defense
(Image courtesy of Augmented Environments Lab)