Marker-Based Embodied Interaction for Handheld Augmented Reality Games

Michael Rohs

Deutsche Telekom Laboratories, TU Berlin
Ernst-Reuter-Platz 7, 10587 Berlin, Germany
michael.rohs@telekom.de
www.deutsche-telekom-laboratories.de/∼rohs

Abstract

This article deals with embodied user interfaces for handheld augmented reality games, which consist of both physical and virtual components. We have developed a number of spatial interaction techniques that optically capture the device's movement and orientation relative to a visual marker. Such physical interactions in 3D space enable manipulative control of mobile games. In addition to acting as a physical controller that recognizes multiple game-dependent gestures, the mobile device augments the camera view with graphical overlays. We describe three game prototypes that use ubiquitous product packaging and other passive media as backgrounds for handheld augmentation. The prototypes can be realized on widely available off-the-shelf hardware and require only minimal setup and infrastructure support.

Keywords: pervasive gaming, augmented reality, embodied user interfaces, interaction techniques, camera phones, handheld devices, mobile devices, visual codes, visual markers

First presented at the 3^{rd} International Workshop on Pervasive Gaming Applications (PerGames 2006) (revised and extended)

1 Introduction

With advances in hardware capabilities, camera-equipped handheld devices gain more and more interest as a platform for augmented reality (AR) [MLB04b, WS03]. Handheld AR systems use displays and integrated cameras to provide video see-through augmentations. The handheld device acts as a window that shows a computer-augmented view of the real world. Head-mounted displays are still cumbersome to wear and it is difficult to imagine that they will be used on a day-to-day basis outside of laboratories or special workplace settings. In contrast, handheld devices are well suited for certain applications of AR. Camera phones in particular are small, unobtrusive, and a constant everyday companion for many people.

A central problem of AR is tracking the position and orientation of physical objects in order to accurately align the computer-generated overlay graphics with objects in the real-world view. We solve this problem with the aid of visual codes [Roh05] and use marker-based interaction techniques with well-defined interaction primitives [RZ05].

We propose to use ubiquitous passive media, such as product packages, tickets, and flyers as background media for handheld AR games. Such media provide visual surfaces and stable reference frames on which AR games can operate. Three prototypical games for handheld devices are presented that incorporate both physical and virtual components: a simple "shooting" game for product packages, a strategy game for table-top boards, and a memory game for physical cards. The games represent different game types and demonstrate slightly different interaction styles. For the shooting game, this is a one-to-one spatial mapping of physical device orientation to virtual character orientation. The strategy game implements a more abstract mapping of device postures to a discrete state space. Finally, the memory game uses the device in an unobtrusive way upside down. The device is only picked up and comes into play in specific situations; otherwise it acts as a passive observer of the users' actions.

After a short review of related work, we discuss the underlying technology and concept. The prototype games are presented in sections 4, 5, and 6. We conclude with a brief summary and an outlook into future research.

2 Related Work

Augmented reality (AR) [Azu97, ABB$^+$01, Fei02] in general is concerned with supplementing the user's sensory perceptions with computer-generated information. Typically, AR research focuses on the sense of sight, with head-mounted "see-through" devices as the typical setup. Other senses, such as hearing, touch, and smell might also be employed by an AR system. The characteristic features of AR systems are [ABB$^+$01]: the combination of real and virtual objects in a real environment; interactivity in real time; and registration (alignment) of real and virtual objects with each other. The computer-generated graphical augmentation is integrated in the user's view of the real world. This can potentially make tasks easier to perform, since hopefully relevant information is provided at the right position at the right time.

Wagner et al. [WPLS05] is an example of work in *handheld AR*. The "Invisible Train"[1] is a multiplayer game in which a virtual train is overlaid onto physical toy rail-tracks. The course of the train can be controlled by altering the track switches and the train speed can be adjusted. The game is controlled using touch screen input. By contrast, our work focuses on embodied user interfaces and has a stronger focus on physical actions as input. In order to enable marker-based tracking, Wagner and Schmalstieg [WS03] have ported the ARToolKit [KBWF99] to the PocketPC platform. Moehring et al. [MLB04a, MLB04b] present a video-see through augmented reality system for camera phones. Their system supports optical tracking of passive paper markers and the registration of 2D and 3D graphics.

Several other handheld AR games have been developed and are described in the literature, e.g. [PR05]. "Mozzies" is a game for Siemens SX1 smartphones, which uses 2D motion detection. The player's task is to shoot virtual moths superimposed over the live video stream by aiming at them with a crosshair. The game won an award as the best mobile game in 2003. In "AR-Soccer"[2] the player can kick a virtual ball with her real foot into the virtual goal [PRS04]. The foot is tracked with the integrated camera of a PDA. "ARTennis"[3] is a two player face-to-face game for camera phones [HBO05]. The virtual scene is superimposed over AR toolkit markers [KBWF99] laid out on a table. The user moves the phone like a tennis racket in order to play the virtual ball. "SymBall" [HW05] is an augmented table tennis game. The phones are tracked with a colour-based feature detection approach. [BBS05] compares joystick and marker-based input for a virtual maze game. The joystick provided fastest completion times, but also lowest levels of user engagement. Tilting provided the greatest degree of user involvement, although it was perceived as challenging.

3 Marker-Based Embodied Interaction for Mobile Games

The goal of this work is to augment physical games with information technology in a way that imposes minimal hardware and infrastructure requirements. We use marker recognition on camera phones to link physical and virtual game components. The markers provide a number of orientation parameters (see Figure 1), which are used in the presented interaction techniques. Moreover, the markers provide a reference coordinate system for producing graphical overlays over the physical game components.

The visual code system [Roh05] has been designed for mobile phones with limited computing capabilities. It recognizes visual codes at arbitrary orientations (tilting up to 60°) and under a wide range of lighting conditions, including daylight and artificial light. A crucial factor with respect to robustness under different lighting conditions is the automatic exposure control and white balance, which are well-developed features in current camera phones. Visual codes store 76 bits

[1] www.studierstube.org/invisible_train(last visited April 12th, 2007)

[2] www.kickreal.de(last visited April 12th, 2007)

[3] Video: www.hitlabnz.org/fileman_store/ar_tennis_lq.wmv(last visited April 12th, 2007)

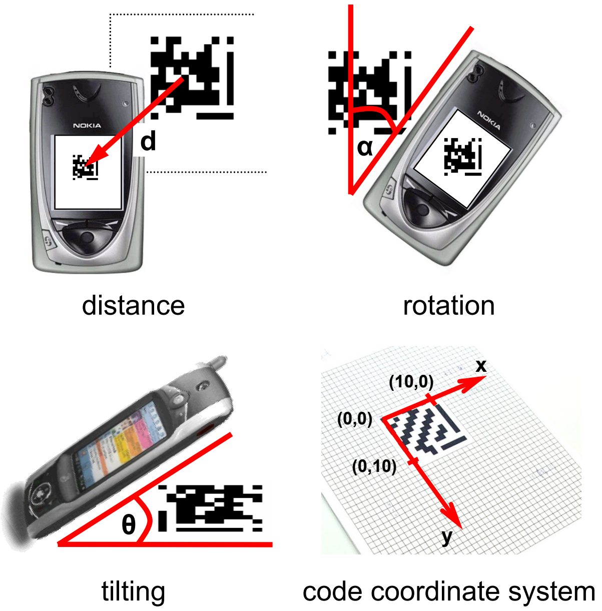


Figure 1: The orientation parameters $(x, y, d, \alpha, \theta_x, \theta_y)$ of the visual code system.

of data in the standard version and are suited for paper print, electronic screens, as well as projection. The algorithm detects multiple codes in a single pass through the camera image at a rate of 10 to 15 updates per second, depending on the frame rate and processing capabilities of the device. In addition to the encoded value, the following orientation parameters are computed for each marker in each frame:

- *Code coordinate system* (x, y). For each frame, the recognizer determines a perspective mapping of points in the image plane to corresponding points in the code plane, and vice versa. As shown in Figure 1, each code has a local coordinate system with its origin at the upper left edge of the code. Areas that are defined with respect to the code coordinate system are invariant to perspective distortion. This allows to generate precisely aligned graphical overlays in the marker plane.

- *Rotation* (α). The rotation of the code in the image is measured in degrees counterclockwise in the range of $0°$ to $359°$.

- *Horizontal and vertical tilting* (θ_x, θ_y). Denotes the amount of tilting between the image plane and the code plane. Horizontal tilting is defined as the angle between the x axis of the code coordinate system and the image plane. Analogously, vertical tilting denotes the angle between the y axis and the image plane.

- *Distance* (d). The reading distance between a marker and the camera is defined such that a value of 100 is assigned to the maximum recognition distance. This distance is reached with current phone cameras if the code occupies about 3% of the image area. Defining distance in this way, instead of using metric values, keeps the distance parameter independent of the particular camera characteristics. It is still adequate for the interaction purposes we envision.

- *Relative movement* $(\Delta x, \Delta y, \Delta \alpha)$. Finally, the recognition algorithm is combined with a visual movement detection algorithm that solely relies on image data provided by the camera. The movement detection provides $(\Delta x, \Delta y, \Delta \alpha)$ triples for relative linear and relative rotational movement (see Figure 2). The movement detection does not rely on the presence of a marker in the image.

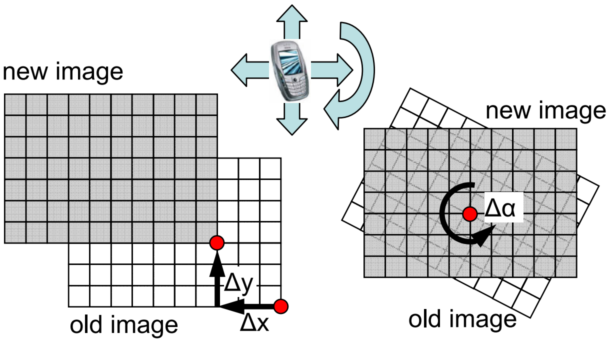


Figure 2: Relative movement is optically detected in $(\Delta x, \Delta y, \Delta \alpha)$ dimensions.

All of the parameters can be computed without knowing the optical characteristics of the particular camera, like the focal distance. If these parameters are known, the tilting angles can be computed in degree and the distance in cm. The devices on which the current implementation has been tested have a camera resolution in view finder mode of 160×120 and 176×144 pixels, respectively, and operate at a frame rate of 15 fps. The real time marker recognition and movement detection has been implemented in C++ on Symbian devices. Devices of similar performance exist for Windows Smartphone.

Manipulative or *embodied user interfaces* [FGH$^+$00] treat physical manipulations on the body of a handheld device as an integral part of its user interface. Embodied user interfaces try to extend the input language of handheld devices and artifacts by incorporating a variety of sensors into them. Example sensors are accelerometers, tilt sensors, gyroscopes, magnetometers, and capacitive sensors. Users can interact with such a device by tilting, moving, rotating, or squeezing it. The physical manipulation and the virtual representation are integrated and tightly coupled within the same device. Whereas in traditional GUIs virtual objects are *directly manipulated* [Shn81], embodied user interfaces allow for the direct manipulation of physical artifacts that embody the user interface. Continuously tracked embodied user interfaces adhere to the principles of direct manipulation, since the actions are rapid, incremental, and reversible, and feedback is visible immediately. Embodied user interfaces mediate – i.e. sense and interpret – the actions of the user in an unobtrusive way. The means to achieve this is to take advantage of everyday spatial skills and make the interaction more similar to physical manipulations of ordinary non-computerized physical objects. This notion of embodied interaction is narrower than Dourish's concept of *embodiment* [Dou01], which denotes, in a more general way, "the presence and occurrentness of a phenomenon in the world", including physical objects, conversations, and actions.

Camera phones are very well suited for manipulative interaction, since they are small and lightweight and can be operated with a single hand. In our *marker-based interaction* paradigm we use the camera as a single sensor to implement physical hyperlinking and manipulative interaction. The markers establish a 3D space whose boundaries are defined by the maximum recognition distance. Using digital zoom this range can be extended significantly. Device position, orientation, and camera focus point in the marker plane are tracked within this space and interpreted in terms of an input vocabulary of physical gestures. The semantics of the interaction is a function of the marker identity and of one or more dynamic gestures or static postures. The camera phone embodies a "symbolic magnifying glass" that enables similar manipulations like an optical magnifying glass. Examples are the choice of focus point and distance. This will not show a magnified version of the physical object, but control some aspect of the game, as well as the graphical output. The phone becomes an intuitive mediator between physical and virtual components of a game.

The "magnifying glass approach" [Rek95] is similar in that it augments the view onto the real world with a handheld display that recognizes color markers and produces textual output. The lens metaphor is also used in the "Total Recall" [HSG03] system, which uses a tracked handheld device for in-place viewing of captured whiteboard annotations. "Mixed interaction spaces" [HELO05] uses the position and size of a circle in the camera image to estimate the position of a handheld device within a physical 3D space.

The following examples show how marker-based interaction can be employed to realize various kinds of computer-augmented physical games.

4 Handheld Augmented Reality on Product Packages: Penalty Kick

Linking handheld augmented reality games to product packages is interesting, because there are billions of both product packages and camera phones. Product packages are truly ubiquitously available in our everyday life. As a part of marketing campaigns, simple games and quizzes on packaging have been around for a long time. Especially larger packages, like cereal boxes, often contain games on the back panel of the box. Even though these games have been very simple and appeal only to a limited proportion of the population – like children and teenagers – such games did not disappear.

Improving the current state means increasing the attractiveness of the product and improving the communication channel between the customer and the manufacturer. Therefore, we propose to realize marketing-related games on product packages as handheld AR games: Packages provide the background for AR games in the form of a playing field and a visual code. The playing field containing the visual code is captured by the camera. The games are controlled by applying embodied interaction gestures to the handheld device. It is likely that such games would appeal to a larger proportion of consumers and increase participation rates in the marketing campaigns they are part of. Moreover, the approach opens up new design possibilities for marketing campaigns on product packages in the virtual realm.

Packaging games are targeted at a broad audience, basically any person who is a potential customer of the product. Interaction techniques thus have to be quickly learnable [DFAB04] and must allow for serendipitous use, i.e. they have to be designed for walk-up-and-use without a significant setup procedure. Moreover, product package interactions are often executed in public space, for example in a store, in which the player cannot fully concentrate on the interaction, but is distracted by other events in the environment.

The idea is not limited to product packages, but can be generalized to include other kinds of background media, such as magazines, flyers, tickets, comic books, CD covers, and situated displays. This highlights the broad applicability of the proposed concept to a large range of media.

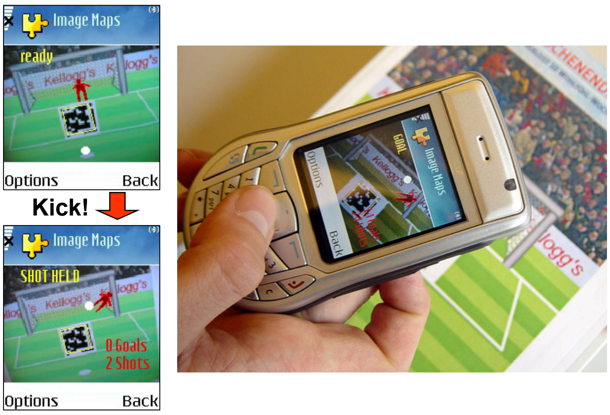


Figure 3: "Penalty Kick" handheld augmented reality game on a cereal package.

The "Penalty Kick" prototype is a simple game that consists of a printed playing surface on a cereal box and virtual overlays generated by a camera phone. The playing surface shows a visual code in the center of the scene, a soccer field, a goal, the penalty spot, and spectators in the background. The display of the camera phone shows the virtual goal keeper, initially standing on the goal line, a virtual ball, initially positioned on the penalty spot, and textual overlays (see Figure 3). The code coordinate system of the central visual code is used to register the generated graphical overlays (the goal keeper and the ball) with elements in the camera image (the goal line and the penalty spot). For large game boards, multiple visual codes would be required to ensure that a visual code is visible in the camera image at all times, or other image features would have to be used. Of course, the code might be covered in the camera view by a virtual overlay graphic.

The visual code value is used as a key to identify the game implementation. In the prototype implementation, the game was previously installed on the phone via a data cable. In a real deployment, the phone might contain a general purpose visual code recognition engine, for example as part of its Web browser, that then downloads the game via the mobile phone network. This would require a Web-based resolver service that maps marker values to URLs. This is a general problem of markers with low data capacity. If the marker is capable of storing a full URL, the resolver is not necessary. The standard version of visual codes has a capacity of 76 bits, which is enough to store an 8 bit header, an IP address, a port number, and a 20 bit identifier. There is an extended version of visual codes, which subdivides each data element into four quadrants. It has a data capacity of 284 bits, which is enough to store a short URL. The game could either be a full phone application or a compact game description that is identified by a specific MIME type and passed on to a game engine. If no visual code recognizer is available on the phone, the process might as well be bootstrapped by printing a short phone number and an identifier next to the code. The game implementation would then be downloaded by sending an SMS to the given number. Alternatively, the game description could be downloaded via Bluetooth from a nearby access point. This mode of operation might be provided as an added value service in a store or at an information kiosk, which shows the game's background on a large display. In order to be safe, the downloaded game should be signed by a trustworthy third party, like the product manufacturer or the service provider.

In contrast to other handheld AR games, such as the "Invisible Train," the user is not confined to control the game by extensive use of the keypad or touch screen. Instead, the game was designed for embodied interaction. It applies visual code interaction primitives, such as rotation and tilting, that the player performs on the device. The necessary actions to interact with the game are simple and intuitive spatial mappings. Aiming at the goal requires rotation and tilting. The amount of device rotation relative to the goal controls the horizontal shot direction. The amount of tilting of the device controls the vertical shot direction (high or flat shot). The user kicks the ball by pressing the joystick key. Upon pressing the key, the ball quickly moves towards the goal (or misses it if the user did not aim right) and the goal keeper jumps to one of four possible positions to either catch or miss the ball with a fixed probability.[4] The textual overlay shows the number of shots and achieved goals.

[4] Video: www.vs.inf.ethz.ch/res/show.html?what=visualcodes-packaging(last visited April 12th, 2007)

The game instructions require just a few words, they can even be explored by users in a few trials without any instructions. Extremely short learning time and intuitive spatial mappings are crucial to reach a large target audience. Few people would be willing to read long instructions before playing a game like this. In a real deployment, the game would be associated with a lottery, for example, to win another box of cereal. The proposed concept allows for the design of exciting single player games on product packages, in which the computer assumes the role of the second player. Most conventional non-computerized games on product packages are only playable with at least two players.

In a qualitative evaluation, we let users play the game and afterwards collected demographic data and users' opinions with a questionnaire. We introduced the game by demonstrating it very briefly. Users could play the game as long as they liked. A total of 16 subjects tried the game (11 male, 5 female). We tested two age groups, the first was 11 students at TU Berlin (age range 20-27, mean age 23.8), the second was 5 school children (age range 13-16, mean age 14.6). None of the participants owned a PDA, all 16 owned a mobile phone, 11 owned a camera phone. Text messaging was used quite extensively, with 13 subjects writing text messages daily, 2 subjects weekly, and 1 subject monthly (Figure 4). Usage frequency of the camera was significantly lower: Of the 11 camera phone owners, 3 used the camera daily, 3 weekly, 4 monthly, and 1 never. Mobile phone games were played by 4 subjects daily, by 3 subjects weekly, by 3 monthly, and by 6 never.

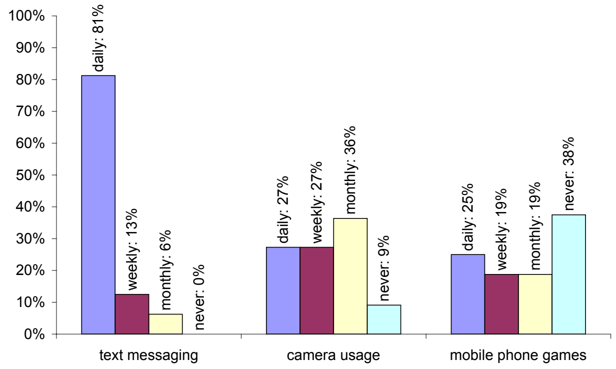


Figure 4: Usage frequency of text messaging, camera, and mobile phone games.

Our impression was that all subjects pretty quickly managed to use gestural control to play the game. Using rotation and tilting to control the direction of the shot seemed to be intuitive for most subjects. The verbal comments were very positive, except one subject, who found it useless to link games to packages and would rather like to play with the mobile device only. Some users said they found the game "a cool idea." Two participants suggested to add sound effects, like a cheering crowd when a goal was scored. Currently the prototype game does not have any sound or vibration effects. One subject remarked that the graphics could be improved. One subject wanted a game with more action and suspense. Yet another was concerned that Kellogg's cornflakes might get more expensive.

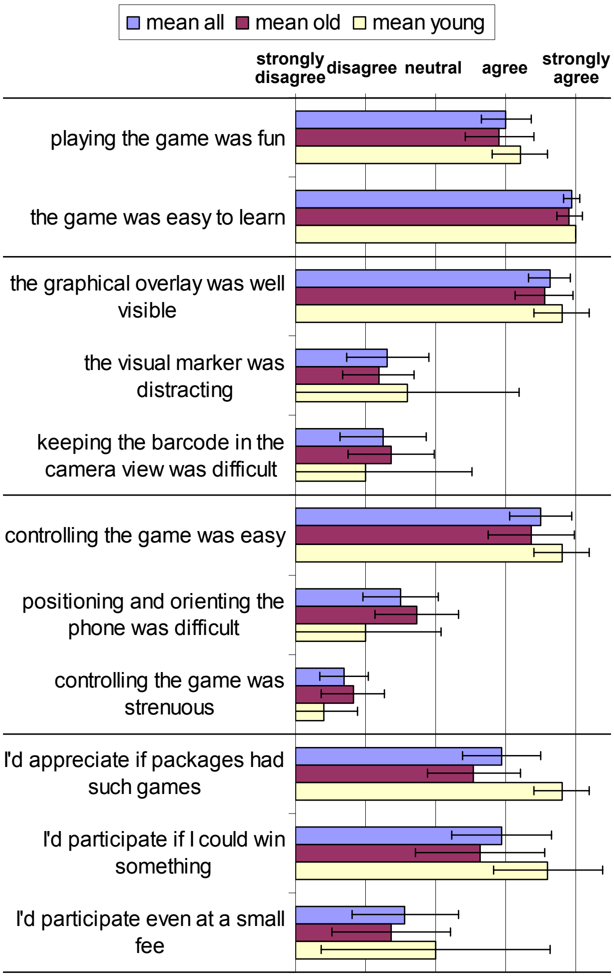


Figure 5: Results of the questionnaire. Horizontal error bars denote 95% confidence intervals.

We used a five-point Likert scale for measuring the participants' responses. The results are given in Figure 5. On average, subjects found the game very easy to learn and fun to play. The graphical overlay over the camera image was well visible. The visual marker was neither distracting nor difficult to keep in the camera view. Some users did not even remember that there was a “barcode” next to the penalty spot and we had to show them the package again. However, we noticed that sometimes users left the recognition distance and marker detection was not reliable in that situation. However, this effect can be avoided by adapting the digital zoom to an appropriate level. Most users found controlling the game simple and intuitive and did not feel tired after playing the game. On average, the younger age group rated the game slightly better than the older age group. However, the only statistically significant difference between the groups was found for the statement “I would appreciate if product packages really contained such games” (ANOVA $F_{1,14}$=5.6, p=0.03). The younger age group agreed more strongly to this statement than the older one. When asked about such a game being part of a lottery, most users would play when participation was free. Expressed willingness to participate dropped considerably, when participation would cost a small fee, such as the cost of a text message. Overall, we were quite surprised by the positive feedback we got for this extremely simple game.

5 Handheld Augmented Board Game: Impera Visco

“Impera Visco” is a turn-based strategy game for 2 to 4 players. 36 tiles are arranged in a 6 × 6 array on the table (see Figure 6). There are 6 different tile types that represent resources and operations on these resources. Each tile is made up of four quadrants, as shown in Figure 6. Quadrant 1 contains a unique visual code, the others show a type-specific image. The game also requires a dice and four playing pieces. The computing hardware of the prototype implementation includes a Symbian camera phone and a PC with Bluetooth. The game realizes the mechanics [LB03] *complex commodities*, *modular board*, *ubiquitous information*, and *informative board*. Game mechanics are specific patterns that summarize and classify game rules. Games can be regarded as composed of multiple game mechanics.

The mission is to achieve wealth on the island of “Visco” by mining, trading, fighting, and cultivating land. First, the game board is scanned row by row in a snake-like fashion to capture the arrangement of the game tiles. The initial position of each player’s piece is determined by throwing the dice. Each turn is structured as follows: First, the player dices and moves his or her piece accordingly in horizontal and vertical direction, such that the sum of steps equals the diced value. If other players are located on the target field, these players fight against each other. Players can train to have higher chances of winning fights. An invisible monster is located somewhere on the field. In each round it moves randomly. If it is located on the field of a player, it might hurt him. A clairvoyant can tell if the monster is on the target field. Since a user always has to perform an operation on the target field, the system knows the new position from the unique code value. In the prototype application, a single mobile phone is passed from one player to the next after each turn. This could easily be extended to multiple devices.

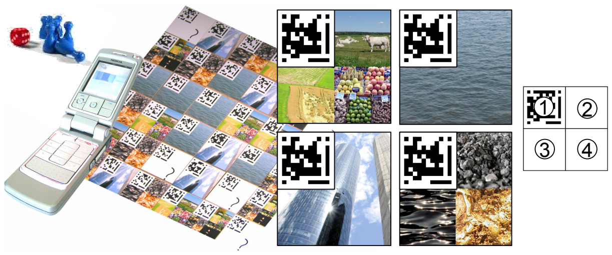


Figure 6: Physical components of the augmented board game “Impera Visco.”

Some of the tile types are shown in Figure 6. Aiming the phone at quadrant 1 (the visual code itself) has a fixed semantics: It shows the score of each player, detailed statistics for the current player, the phone’s current information about the position of all players, as well as help information about the possible operations on the tile. There are the following tile types:

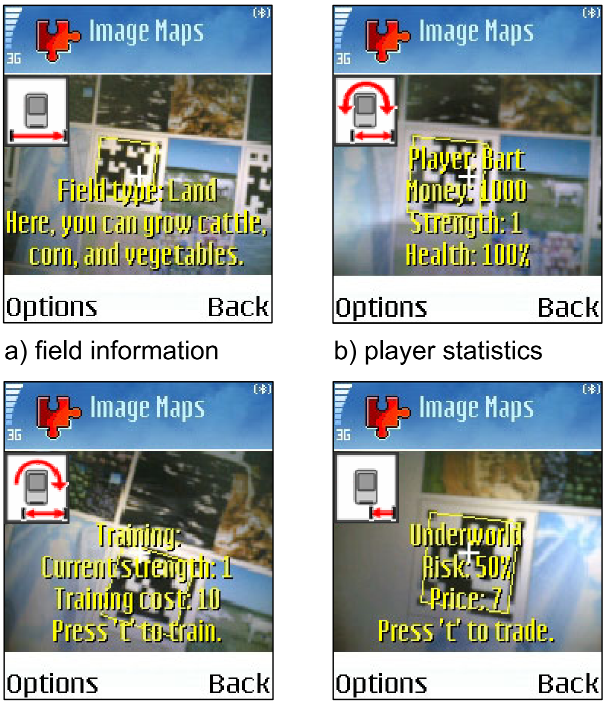
- *Land tiles* are for growing cattle, corn, and vegetables on quadrants 2, 3, and 4, respectively. Animals grow slowest, vegetables fastest. Cattle is more valuable than corn and vegetables. Opponents can loot land tiles. However, players can place invisible guards on fields.

- *Water tiles* cannot be crossed if two or more of them are adjacent. A player may not be positioned on a water card. Water cards have a positive effect on adjacent land fields; they accelerate growth.

- *Mining tiles* produce gold, ore, and oil. Mining gold is slowest, ore fastest. Mining tiles can also be looted.

- *City tiles* allow for trading. Prices change over time. Cities have three layers: in the lowest layer things are cheap, but the chances of fraud are high; the higher the layer, the lower the risk. Since cities are well watched, the monster avoids the city.

- *Bank tiles* allow to convert collected goods to money, yet prices change frequently, thus it is worthwhile to check prices often.



Field type: Land
Here, you can grow cattle, corn, and vegetables.

a) field information

Player: Bart
Money: 1000
Strength: 1
Health: 100%

b) player statistics

Training:
Current strength: 1
Training cost: 10
Press 't' to train.

c) player training

Underworld
Risk: 50%
Price: 7
Press 't' to trade.

d) choosing where to buy (risk vs. price)

Figure 7: Virtual components of the augmented board game "Impera Visco."

The game uses the interaction cues that are described in detail in [RZ05]. These composable iconic cues tell the user, which embodied interaction postures are possible on a specific tile. A game operation is selected by focusing a quadrant, rotating the phone, or changing the distance. Operations are triggered by pressing the joystick button. Figure 7 shows a number of game states, game commands, and their corresponding interaction cues. Since the graphics are overlaid onto the camera view, particularly textual output can interfere with the camera image in the background. For improved readability, we use a technique, called *anti-interference font* [HV96] that mitigates this issue.

The visualization of the game state and the interpretation of embodied interaction postures is implemented in the visual code image maps framework [RZ05] on Symbian phones. For ease of prototyping, the actual game logic runs on a PC and is implemented in Java and the Jess[5] rule-engine. Jess allows to concisely describe the game components and current state as facts and rules in a knowledge base. Moving pieces on the board and triggering actions results in new facts. These facts potentially activate rules, which in turn produce new facts. The Jess source code consists of about 1000 lines of code.

The initial goal in developing the game was to evaluate the use of fine-grained embodied interaction within the framework described in [RZ05]. Less emphasis was put on producing an actually playable and enjoyable game [DCT04]. This remains an area of future research. During test games we discovered, that the different states must not be too complex. Otherwise interaction tends to take too long and the game flow is interrupted.

6 Handheld Augmented Card Game: Smart Memory

"Smart memory" is a computer-augmented version of the classic "memory" game (see Figure 8), in which pairs of identical pictures have to be found among a number of cards that are laid out upside down on a table. One card side shows a picture, the other side a visual code. In each turn, a player reverses two cards. If their pictures match, the player gets the pair and continues, otherwise it is the next player's turn.

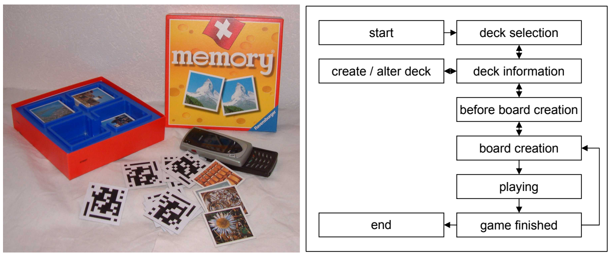


Figure 8: Physical components and flow diagram of the smart memory game.

The design goal was to augment the game and enable new functionality, yet to retain the game's simplicity. The additional time spent interacting with the computer, as well as the setup time, was tried to be kept at a minimum. The game realizes the mechanics [LB03] *computerized clues* and *keeping track.*

[5] www.jessrules.com(last visited April 12th, 2007)

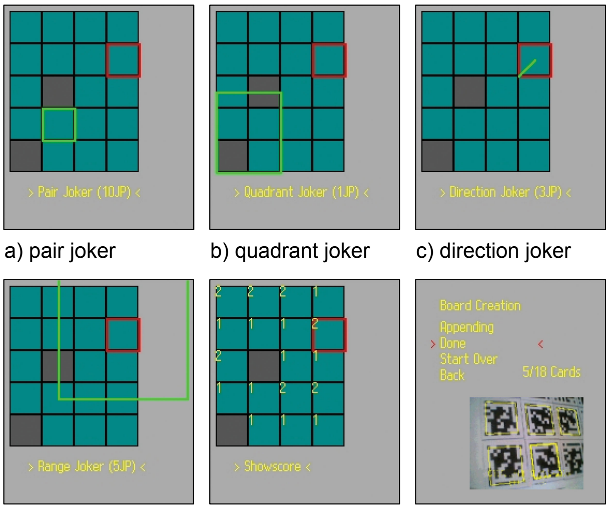
The main enhancement is the introduction of *virtual jokers*. A joker provides additional information about the game, like the general direction in which the matching card of a given card can be found. At the start of the game, each player gets an account with joker points. The use of each joker costs some amount of joker points, depending on the usefulness of the joker.

A minor enhancement is that a different number of points is assigned to different pairs. In the original game, only the number of collected pairs counts. The phone keeps statistics on which cards are consistently collected late in a game. This is a hint that such pairs are difficult to find, for example, because their pictures are very similar to other pairs. Difficult pairs are rated higher than others.

A flow diagram of the game is depicted in Figure 8. The players might select a subset of the cards for a game (this is called "deck selection" in Figure 8). The visual code values for a pair are different, otherwise players could try to match visual similarities in code patterns. The phone learns which code values belong to each pair by placing the pairs next to each other in a row, reversing, and then scanning them in that order ("create deck" in Figure 8). This step has to be performed only once, when the set of used cards changes. After the pairs are learned, the cards are arranged on the table ("board creation" in Figure 8). The board is scanned to build up an internal representation of its layout.

During play, most of the time the phone is placed upside down on the table, such that the camera faces upwards. Each card that the user reverses is shortly held over the phone, until audio feedback is given to the user. The phone recognizes the card and updates the board state. Thus, during normal play, handling the phone is not required. The technology unobtrusively stays in the background. Since the phone counts points, it is no longer necessary to pile up collected cards. They can be placed back on the board, which enables the *replay joker* (see below).

Only if a player wants to use a virtual joker, he or she picks up the phone and chooses the joker. The cost of the joker is then deduced from the user's account and its result is displayed on the screen. The following jokers are implemented (see Figure 9):



a) pair joker b) quadrant joker c) direction joker

d) range joker e) statistics joker f) board creation

Figure 9: Virtual jokers of the smart memory game.

- The *quadrant joker* shows in which quadrant the pair corresponding to a given card is located.

- The *pair joker* shows the exact location of the corresponding card. It is more expensive than the other jokers.

- The *direction joker* points in one of eight directions to indicate the rough direction of the corresponding card.

- The *range joker* shows the range in which the matching card is located.

- The *statistics joker* shows the number of times that the cards on the board have been flipped during a game.

- The *destruction joker* destroys five joker points of the next player. It costs ten joker points.

- The *replay joker* allows to restart the whole game with the same board. It costs all of the initial joker points. This is useful if a player has memorized the positions of pairs during a game.

During test games it was discovered that some players were very reluctant to use their joker points, since they considered this as cheating, having the original memory game in mind. Apparently, players have very clear expectations about well-learned games. It might therefore be better to create new games than to try to enhance existing ones. Another complaint was that the phone still distracts too much from the game. During the time one player used a joker, others tended to forget about the locations of cards. For a game in which concentration is very important, it is crucial that technology stays in the background and requires as little attention and interaction time as possible.

7 Conclusions and Future Work

In this article, we demonstrated the feasibility of handheld AR games, which are controlled using marker-based embodied interaction. Three prototypical games of different type were presented, each integrating specific physical components: a simple "shooting" game for product packages, a strategy game for tabletop boards, and a memory game for physical cards. Each game type features a slightly different interaction style. For the penalty kick, this is a one-to-one spatial mapping of physical orientation to virtual orientation in the game. The strategy game implements a more abstract mapping of device postures to a discrete state space. Finally, in the memory game the device is used in an unobtrusive way upside down and only comes into play in specific situations; otherwise it just acts as a passive observer of the users' actions.

The proposed games impose minimal requirements in infrastructure and setup. Marker-based interaction is generic in the sense that in principle it can also be implemented with other types of markers, such as *Data Matrix* [Int00b] and *QR Code* [Int00a], as long as these markers provide the required orientation parameters. In particular, the code coordinate system described above is essential for detecting the focused point and for the precise alignment of graphical overlays. In addition, the markers and orientation parameters have to be recognized in real time in the stream of camera frames without a perceptible delay.

The general concept and the demonstrated game prototypes are promising. However, much work remains to be done. On the technical side, we are trying to include additional sensors beyond cameras for further interaction possibilities. We are also working on a description language for handheld AR games that takes sensor input into account and allows to specify graphical output as well as to formulate game rules. On the gaming side, it is still an open question, which types of physical games benefit most from a virtual component and what factors are important to make a game actually enjoyable and fun to play [DCT04].

Acknowledgments

Thanks to Jean-Daniel Merkli for designing and implementing the penalty kick game as part of his Diploma thesis, to Nicolas Born for designing and implementing "Impera Visco" in a semester project, and to Erich Laube for designing and implementing "Smart Memory" in a semester project.

References

[ABB+01] Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre, *Recent advances in augmented reality*, IEEE Computer Graphics and Applications **21** (2001), no. 6, 34–47, ISSN 0272-1716.

[Azu97] Ronald Azuma, *A survey of augmented reality*, Presence: Teleoperators and Virtual Environments **6** (1997), no. 4, 355–385.

[BBS05] Sam Bucolo, Mark Billinghurst, and David Sickinger, *User experiences with mobile phone camera game interfaces*, MUM '05: Proceedings of the 4th international conference on Mobile and ubiquitous multimedia (New York, NY, USA), ACM Press, 2005, ISBN 0-473-10658-2, pp. 87–94.

[DCT04] Heather Desurvire, Martin Caplan, and Jozsef A. Toth, *Using heuristics to evaluate the playability of games*, In Extended abstracts of the 2004 Conference on Human Factors in Computing Systems, CHI 2004 (Vienna, Austria), ACM Press, 2004, ISBN 1-58113-703-6, pp. 1509–1512.

[DFAB04] Alan Dix, Janet E. Finlay, Gregory D. Abowd and Russell Beale, *Human-Computer Interaction*, 3rd Edition, Prentice Hall, 2004, ISBN 0-13-046109-1.

[Dou01] Paul Dourish, *Seeking a foundation for context-aware computing*, Human-Computer Interaction **16** (2001), no. 2–4, 229–241, ISSN 0737-0024.

[Fei02] Steven K. Feiner, *Augmented reality: A new way of seeing*, Scientific American **286** (2002), no. 4, 48–55.

[FGH+00] Kenneth P. Fishkin, Anuj Gujar, Beverly L. Harrison, Thomas P. Moran, and Roy Want, *Embodied user interfaces for really direct manipulation*, Commun.

ACM **43** (2000), no. 9, 74–80, ISSN 0001-0782.

[HBO05] Anders Henrysson, Mark Billinghurst, and Mark Ollila, *Face to face collaborative AR on mobile phones*, ISMAR '05: Proceedings of the Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (Vienna, Austria), IEEE Computer Society, 2005, ISBN 0-7695-2459-1, pp. 80–89.

[HELO05] Thomas Riisgaard Hansen, Eva Eriksson, and Andreas Lykke-Olesen, *Mixed interaction space: Designing for camera based interaction with mobile devices*, In Proceedings of the 2005 Conference on Human Factors in Computing Systems, CHI 2005, extended abstracts, Session: Late breaking results (Portland, OR, USA), ACM Press, 2005, ISBN 1-59593-002-7, pp. 1933–1936.

[HSG03] Lars Erik Holmquist, Johan Sanneblad, and Lalya Gaye, *Total recall: in-place viewing of captured whiteboard annotations*, In Extended abstracts of the 2003 Conference on Human Factors in Computing Systems, CHI 2003 (Ft. Lauderdale, Florida, USA), ACM Press, 2003, ISBN 1-58113-637-4, pp. 980–981.

[HV96] Beverly L. Harrison and Kim J. Vicente, *An experimental evaluation of transparent menu usage*, CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press, 1996, ISBN 0-89791-777-4, pp. 391–398.

[HW05] Mika Hakkarainen and Charles Woodward, *SymBall – camera driven table tennis for mobile phones*, ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE), June 2005.

[Int00a] International Organization for Standardization, *Information technology – Automatic identification and data capture techniques – Bar code symbology – QR Code. ISO/IEC 18004*, 2000.

[Int00b] International Organization for Standardization, *Information technology – International symbology specification – Data Matrix. ISO/IEC 16022*, 2000.

[KBWF99] Hirokazu Kato, Mark Billinghurst, Suzanne Weghorst, and Tom Furness, *A mixed reality 3D conferencing application*, Technical Report R-99-1 Seattle: Human Interface Technology Laboratory, University of Washington, 1999.

[LB03] Sus Lundgren and Staffan Bjrk, *Game mechanics: Describing computer-augmented games in terms of interaction*, Proceedings of the 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE 2003), 2003.

[MLB04a] Mathias Moehring, Christian Lessig, and Oliver Bimber, *Optical tracking and video see-through AR on consumer cell phones*, Workshop on Virtual and Augmented Reality of the GI-Fachgruppe AR/VR, 2004, pp. 193–204.

[MLB04b] Mathias Moehring, Christian Lessig, and Oliver Bimber, *Video see-through AR on consumer cell phones*, IEEE/ACM International Symposium on Augmented and Mixed Reality (ISMAR'04), 2004, pp. 252–253.

[PR05] Volker Paelke and Christian Reimann, *Vision-based interaction – a first glance at playing MR games in the real-world around us*, Proceedings of the 2nd International Workshop on Pervasive Gaming Applications (PerGames) at PERVASIVE 2005 (Munich, Germany), May 2005.

[PRS04] Volker Paelke, Christian Reimann, and Dirk Stichling, *Foot-based mobile interaction with games*, ACE '04: Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology (New York, NY, USA), ACM Press, 2004, ISBN 1-58113-882-2, pp. 321–324.

[Rek95] Jun Rekimoto, *The magnifying glass approach to augmented reality systems*, International Conference on Artificial Reality and Tele-Existence '95 / Conference on Virtual Reality Software and Technology (ICAT/VRST '95), 1995, pp. 123–132.

[Roh05] Michael Rohs, *Real-world interaction with camera phones*, Second International Symposium on Ubiquitous Computing Systems (UCS 2004), Revised Selected Papers (Tokyo, Japan) (Hitomi Murakami, Hideyuki Nakashima, Hideyuki Tokuda, and Michiaki Yasumura, eds.), LNCS 3598, Springer, July 2005, ISBN 3-540-27893-1, pp. 74–89.

[RZ05] Michael Rohs and Philipp Zweifel, *A conceptual framework for camera phone-based interaction techniques*, Pervasive Computing: Third International Conference, PERVASIVE 2005 (Munich, Germany) (Hans W. Gellersen, Roy Want, and Albrecht Schmidt, eds.), LNCS 3468, Springer, May 2005, ISBN 3-540-26008-0, pp. 171–189.

[Shn81] Ben Shneiderman, *Direct manipulation: A step beyond programming languages*, Proceedings of the joint conference on Easier and more productive use of computer systems. (Part - II) (New York, NY, USA), ACM Press, 1981, ISBN 0-89791-064-8, p. 143.

[WPLS05] Daniel Wagner, Thomas Pintaric, Florian Ledermann, and Dieter Schmalstieg, *Towards massively multi-user augmented reality on handheld devices*, Pervasive Computing: Third International Conference, PERVASIVE 2005 (Munich, Germany) (Hans W. Gellersen, Roy Want, and Albrecht Schmidt, eds.), LNCS 3468, Springer, May 2005, ISBN 3-540-26008-0, pp. 208–219.

[WS03] Daniel Wagner and Dieter Schmalstieg, *First steps towards handheld augmented reality*, Seventh IEEE International Symposium on Wearable Computers (Washington, DC, USA), IEEE Computer Society, October 2003, ISSN 1530-0811, pp. 127–135.

Citation

Michael Rohs, *Marker-Based Embodied Interaction for Handheld Augmented Reality Games*, Journal of Virtual Reality and Broadcasting, 4(2007), no. 5, March 2007, urn:nbn:de:0009-6-7939, ISSN 1860-2037.