W41K: Digitally Augmenting Traditional Game Environments

Steve Hinske
Institute for Pervasive Computing
ETH Zurich
8092 Zurich, Switzerland
+41 44 632 0768
steve.hinske@inf.ethz.ch

Marc Langheinrich
Faculty of Informatics
University of Lugano (USI)
6904 Lugano, Switzerland
+41 58 666 4304
langheinrich@acm.org

ABSTRACT
Augmented game environments use unobtrusively embedded technology to augment traditional games with virtual information and novel interaction capabilities. This article establishes and discusses a set of guidelines for designing and implementing such environments, based on our experiences in creating digital augmentations of existing play environments. We suggest a two-step process comprised of *game flow virtualization* and *physical artifact augmentation* to create augmented game environments based on existing table top games. We will then demonstrate how these guidelines can be put to practice by presenting the augmented version of a miniature war game.

Keywords
Augmented Game Environment, Pervasive Computing, Design Guidelines, Digital Augmentation

1 INTRODUCTION

Computer and video games have the potential to engage players for hours in fantastic and enthralling worlds. Especially the tremendous progress in computer hardware within the last decade has enabled game designers to create highly immersive scenarios and worlds: players can visit imaginary and surreal worlds or far away and long gone places, take on all kinds of roles, and test their problem-solving skills on many levels. These games furthermore "provide for complex simulations, evolving environments, impartial judging, the suspension of disbelief, and the ability to save the state of the game" [16].

Despite these apparent advantages of computer and video games, traditional games such as tabletop games continue to appeal due to the socializing aspects that usually come with gathering all players in a single location (e.g., around a table). Additionally, such games allow for the interaction with the physical world, e.g., providing the tactile feedback of game figures, or requiring physical activities by the players – a benefit that their virtual counterparts only recently started to explore (e.g., Nintendo Wii).

The paradigm of pervasive computing enables a third alternative, the combination of the virtual and the physical world, and the advantages thereof, i.e., the endowment of traditional play environments with virtual information and novel interaction capabilities. These elements can enrich the players' play experience, e.g., by providing them with useful in situ information about the game and its objects.

The digital augmentation of already existing game environments, however, can be challenging: integrating pervasive computing technologies should not change the *look-and-feel* of the original game; the technology should be almost invisible as to not distract the players. This objective entails a number of aspects to be taken into consideration if the resulting augmented game is to really enhance the play experience.

In this paper we establish and discuss general guidelines for designing and implementing augmented game environments by means of pervasive computing technologies. Based on these theoretical design criteria, we then report on the digital augmentation of a popular miniature war game called Warhammer 40.000™ (W40K).

W40K is an excellent example as it is a rather complex game with very distinct requirements: not only does a typical game session comprise numerous units with individual skills and characteristics that are subject to constant changes – requiring the players to keep track of a large number of unit sheets –, but it is essential to localize the game objects on the battlefield within the range of millimeters, which has to be done manually using rulers and protractors. In addition to this, the game consists of myriad rules covering countless exceptions.

This paper is structured as follows: first, we present and discuss the design guidelines for digitally augmenting traditional games. Second, we briefly explain the game W40K and outline why such a game could strongly benefit from integrating pervasive computing technologies. We then introduce W41K, the digital augmented version of W40K and how we applied the previously presented guidelines. The final section of the paper contains a summary of the findings and our conclusions.

2 DESIGNING AUGMENTED GAMES

Digitally augmenting traditional game environments can be challenging and difficult. If the players are truly to benefit from the integration of virtual elements into real, physical objects, a number of design criteria are to be met. With this paper, we aim at deriving general design guidelines that might support other game designers when digitally augmenting game environments.

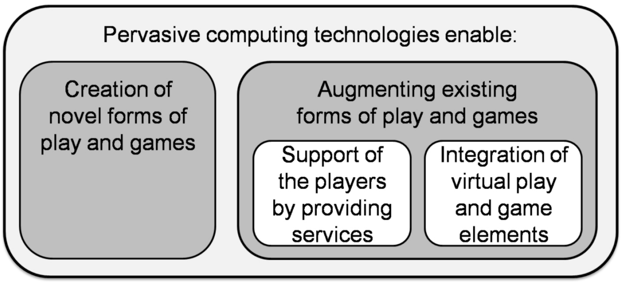

Figure 1. Applying pervasive computing technologies to the domain of play and games. In this paper, we focus on supporting the players by providing services in digitally augmented traditional game environments (center).

It is worth noting that there is no commonly agreed definition of *augmented games* (often also called "pervasive (computing) games", e.g., [6,17,20]). For the purpose of this paper, we focus on already existing game environments that are digitally augmented using pervasive computing technologies to support the players (see Fig. 1 and cf. [12]). However, many of the discussed aspects can also be applied to other forms of pervasive games, e.g., augmented reality games as defined and presented in, e.g., [2].

Floerkemeier and Mattern [5] give an example of augmenting an existing form of a game in the context of playing cards: by equipping a regular card deck with RFID technology, the players can be supported in a variety of card games through automatically counting points or by helping novice players to remember the rules.

In [6], Hinske et al. discuss how pervasive computing technologies can support the different aspects of a game. These guidelines – though somewhat abstract – give first insights into what can be done to enhance the play experience by means of such technologies (see Tab. 1).

While these aspects certainly play an important role when designing pervasive computing games, they do not discuss how to actually design and implement augmented game environments. We now focus on the actual realization of such a digitally augmented game. There are two main aspects involved here:

1. The *game flow virtualization* of the game, i.e., modelling its rules and game objects, and
2. The *physical augmentation* of the game, i.e., the integration of pervasive computing technologies into the real-world game elements.

2.1 Game Flow Virtualization

Games, in contrast to toys, have the advantage of being organized and structured: the goal of the game is clear to all players; the rules guarantee a fair competition; the current state as well as the outcome of the game can be measured and compared; the players must make decisions based on the available, countable resources. In other words, the rules are the central component of a game and *virtualizing* them can yield benefits for the players, mostly because it is thus possible to

- automatically check for rule consistencies and violations, and to
- provide players with relevant information and rules fast and in situ (i.e., they do not have to find the appropriate rule or table themselves but are presented with it right when needed).

Table 1. How pervasive computing technologies can support different aspects of a game (synthesized from [6,13]).

Aspect	*Support through Pervasive Computing*
Concentration	Support players in switching between in-game tasks and important secondary factors.
Challenge	Stimulate and support players in their own creation of game scenarios and pacing.
Player Skills	Enable players' skills to be developed in a flexible pace set by the players.
Control	Enable players to quickly get a picture of the current status in the game world.
Immersion	Enable players to shift focus between virtual and physical parts without loss of immersion.
Social Interaction	Support and enable possibilities for game oriented, meaningful and purposeful social interaction within the gaming system.
Clear Goals	Support the players in forming and communicating their own intermediate goals.
Rules	Make game state accessible to players at all times and report rule violations immediately and in an adequate way.
Competition	Provide means to the players for a smooth engagement in a fair competition.
(Quantifiable) Outcome	Keep score of the game and allow players to always inquire the current score.
Decisions	Allow players to make decisions anytime and give immediate feedback by suitable means.
Emotional Attachment	Provide the players with a compelling experience that seamlessly combines different media and multimodal interactions.

This virtualization necessitates the understanding and modelling of the game and all its game objects, which can be difficult and laborious, in particular in complex games that would benefit significantly from such an augmentation. The actual act of game flow virtualization, however, depends very much on the specific game that is augmented, making it difficult to arrive at a set of general guidelines. Below we list four recommendations that have proven helpful when building augmented games (also cf. Tab. 2).

Clearly, an iterative design and implementation process helps with the gradual synchronization of virtual rules with their "printed" counterparts. We also noticed that "pure" object-oriented approaches did not lend themselves well to implementing messy rule books – we usually were more successful with a combination of object-oriented modelling / programming and scripting languages (this approach is also common with computer and video games). Another very promising approach, depending on the game, is to model the game using sequence diagrams, which can then be easily translated into state machines (also cf. Fig. 10).

There is a further aspect to be considered at this point: some games permit the (physical) addition or removal of new game objects, either before or during a game session. In a game like W40K, e.g., players can constantly purchase new units for their army. Consequently, the world model of the game must be flexible and extensible.

Since rule virtualization requires that the system is able to track the physical movements of all game objects (see section 2.2), this information can be additionally recorded in order to allow players to review and analyse a played game. It also makes it possible to stop and resume a game anytime: "many board games take longer than the typical two or three hour period of a single session. Thus, persistency becomes an issue, which includes recording game events and possibly the creation of a corresponding game history" [15]. Explicit session management must thus be addressed early in the design process.

2.2 Physical Augmentation

In terms of physical objects that can be digitally augmented, a game environment typically consists of two components:

- Game infrastructure (e.g., a game board), and
- Game objects (e.g., figures).

Digitally augmenting these components raises different problems, depending on the components: figures are often rather small and many look alike. Such characteristics can have a significant influence on the choice of technologies.

Since our goal is to combine the best of two worlds, there are two main objectives: first, the technology should be integrated unobtrusively, so that the natural game experience is neither disturbed nor rendered unusable if the technology fails (i.e., the game can still be used in the traditional way). Second, the system should not negatively influence the rich social interactions of the original game.

Table 2. Important aspects in game flow virtualization.

1. *Mixed programming environment:* while a particular modeling / programming approach alone might turn out to be insufficient or problematic, a combination of several approaches might yield better results and simplify the process.
2. *Sequence diagrams:* these can significantly simplify the modeling process of the game flow.
3. *Flexible world model:* provides support for in-game addition of new game objects.
4. *Session management:* allows players to stop and resume anytime. If desired, include event history so that players can review and maybe replay an already played game.

Table 3. Design guidelines for physical augmentation (adapted and extended from [8], see also there for further details).

1. The technological enhancement should have an added value.
2. The supported actions and tasks need to be clearly specified.
3. The focus should remain on the game and the interaction itself, not on the technology.
4. Technology integration should be done in a way that is unobtrusive, if not completely invisible.
5. The game should still be playable (in the "traditional" way) even if technology is switched off or not working.
6. Design and implementation should be tightly coupled.
7. The technology should be reliable, durable, and safe.
8. Players should receive simple and efficient access to information. Feedback should be immediate and continuous.
9. The added technology should support the high dynamics of the game environments.
10. Development should follow an iterative process, including rapid prototyping and testing.
11. The operation of the integrated technology should be as maintenance-free as possible.
12. Secondary user interfaces should be minimized.

Hinske et al. [8] established a set of design guidelines for augmented *toy* environments. While toy environments are typically much less structured than game environments (see [6] for a discussion), some of their recommendations can still be applied here, especially those that concern the integration of technology into traditional play objects. Tab. 3 offers our variation (and slight expansion) of these guidelines in the context of augmented *game* environments.

Apart from slightly adapting Hinske et al.'s toy environment guidelines 1-10 for game environments, we added two additional criteria. Firstly, the actual operation of the system should be as maintenance-free as possible (guideline 11): the players should not be burdened with maintaining the technology but focus on the game itself. This includes tasks such as recharging batteries.

Secondly, the number of additional user interfaces should be minimized (guideline 12). Ideally, game objects would remain the major (tangible) user interfaces. If, however, the augmentation requires additional interfaces to provide the information and services, these interfaces should be as unobtrusive as possible to not disrupt the original gameplay. If possible, they should be designed in such a way that they encourage and support collaboration, thus contributing to the players' social experience.

While these guidelines describe *how* to augment game objects, it is equally important to know *what* and *why* to augment. Following generally established processes of interaction design (e.g., [3,19]), we suggest the following six-step *cycle of digital augmentation* (also see Fig. 2):

1. Game Analysis (typically part of or closely related to game flow virtualization). Relevant questions are:
    a. How can the players be supported?
    b. What are the most cumbersome tasks? What tasks can players be possibly relieved of?
    c. What are the goals of the augmentation?

2. Determine the parts that are to be digitally augmented. This part also includes assessing the effectiveness, efficiency, and (economical) feasibility of augmentation.
    a. Where should additional interfaces be placed?
    b. Which objects are suitable for augmentation?
3. Determine how to augment the parts with regard to the aforementioned guidelines (see Tab. 3).
4. Choose the appropriate technology that contributes to the goals while meeting the guidelines.
5. Implement the augmentation / build the prototype.
6. Test if the result satisfies the set goals and, if required, iteratively improve the prototype. Frequent evaluation.

We now discuss how these guidelines can be practically applied. To this end, we briefly describe W40K and then present W41K, our digitally augmented version of W40K.

3 WARHAMMER 40K™

W40K is a miniature war game, in which two or more players engage each other in battle, commanding an army of many game objects representing combat units, usually with the goal of eliminating the adversarial forces. A W40K play session encompasses numerous units on a typically customized battlefield (see Fig. 3). Each unit has its own data sheet containing important information such as health points, armor, and weapons, some of which are subject to constant changes throughout the session. Furthermore, for move and attack phases, the exact position of the figures on the field is required.

Games such as this are usually very complex and intricate. They comprise countless different units with individual characteristics and special skills, as well as many rules and corresponding exceptions[1]. Players often spend much time reading, both before and during the game, and need to continually keep their units' data sheets up-to-date.

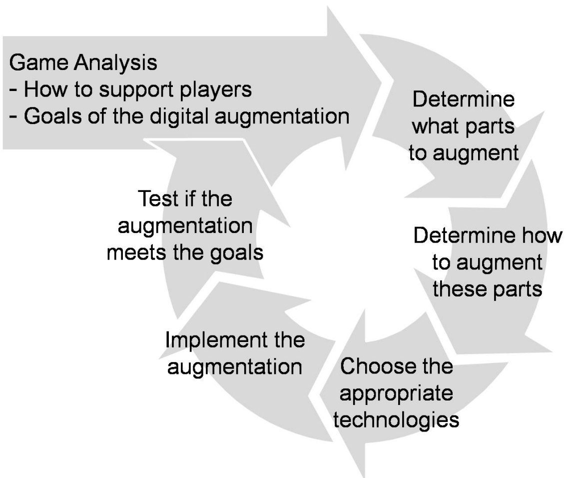

Figure 2. Overview of the design process for digitally augmenting traditional game environments.

[1] In this paper, we focus on the fourth edition of W40K (cf. http://en.wikipedia.org/wiki/Warhammer_40,000).

Figure 3. A typical Warhammer 40K™ scenario. The battlefield consists of the players' units (soldiers and vehicles) as well as landscape elements and buildings.

The game is round-based and each round roughly consists of three phases: moving, ranged combat (i.e., shooting), and close combat. Combat requires an element of chance, which is realized by using six-sided dice. Having decided which unit and how to attack, players must roll one or more dice and the result determines the success of the attack.

Warfare in such games very much depends on the exact location and orientation of game pieces, in order to properly assess the visibility of enemies, or the range and effect of weapons. For that, the players must manually measure the distances using rulers and apply templates for attacks that cover larger areas (e.g., grenades).

Based on the brief introduction of W40K, we will now discuss how the players could be potentially supported during the game. Principally speaking, the idea is to relieve the players of annoying and/or cumbersome tasks and thus allow them to focus more on the game (i.e., the strategic decisions) and the socializing aspect (i.e., chat with the other players). Players should be supported by providing them with context-relevant information and by helping them during the execution of their moves. This support should not compromise the traditional *look-and-feel* of the battlefield, the units, or the dice.

To this end, we first analyzed the game, i.e., we thoroughly scrutinized the rules and played it several times. We then observed and informally interviewed several seasoned players to verify our initial analysis. This also helped us to elicit further aspects that are not part of the formal rules but nonetheless contribute to the game experience.

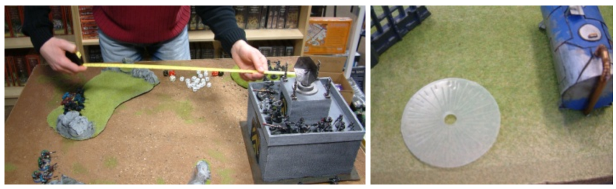

Figure 4. Manual measuring: a player measures the distance the between two units using a ruler (left) and a template for determining the blast radius of a grenade (right).

Based on our findings, we digitally augmented the following four parts of the game:

- The game field and game objects: to automatically determine the position and orientation of units.
- The rule system: to automatically check the validity of the players' decisions and to give them the information required for making a decision.
- The dice: to automatically recognize rolled results and to forward them to the background system.
- The data sheets: to keep up-to-date information on all units of the current game session.

According to the interviewed players, these augmentations could be especially beneficial for tournaments, since they provide for an impartial entity that could be consulted in cases of disagreement or uncertainty.

4 WARHAMMER™ 41K

We now describe how we realized these steps. Though the physical augmentation usually comes second, we present it first: on the one hand, this ensures a better understanding of how the system works and, on the other hand, we only describe the physical infrastructure briefly as we have discussed it in detail in other publications before [7,9-11].

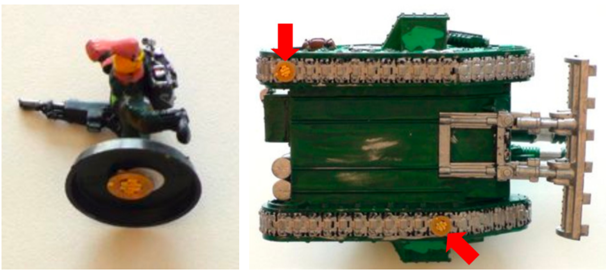

Figure 5. W40K objects unobtrusively equipped RFID tags.

4.1 Physical Augmentation in W41K

To provide the players with the aforementioned services, it is essential to automatically determine where each unit is located at any given moment. To this end, we need an infrastructure that enables the identification and tracking of game objects. This information will then be processed and adequately made available through user interfaces.

We chose RFID technology to digitally augment the game objects since this technology has the following advantages (for more details on appropriate technologies, see [7,9]):

- Tags can be hidden and thus work unobtrusively,
- Augmented objects are almost maintenance-free (except for exchanging damaged RFID tags),
- There is no need to calibrate the equipment,
- Each object is unambiguously identifiable,
- No line-of-sight is required, and
- Costs are low by comparison.

In earlier publications we reported on our development of an RFID-based infrastructure that can be used to identify and localize game objects [7,9]. We managed to improve resolution accuracy to within a few millimeters using a moveable RFID antenna. The sensing system then forwards the position of all detected tags to the gaming application. The game objects are all equipped with one or more RFID transponders of the type Ario 370-S SDM by TagSys[2], which feature a diameter of only 0.9 cm and can thus be easily integrated into the game objects (see Fig. 5).

Using the same RFID technology, we also designed and implemented several augmented dice [10,11]. The six-sided dice have the same size like commercially available off-the-shelf dice but are capable of automatically recognizing the players' rolls and forward the results to the system (see Fig. 6). This allows the players to play in the traditional manner with the same look-and-feel of regular dice.

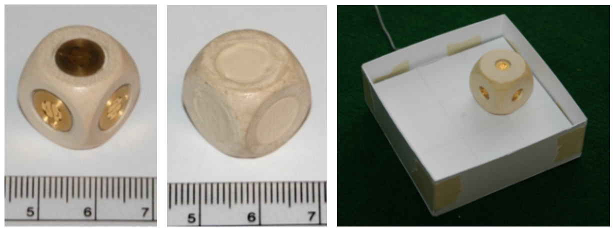

Figure 6. The augmented dice prototypes, featuring a perfect recognition rate with a form factor of ordinary, off-the-shelf dice (edge length 1.6cm) [10,11]. On the right side is the dicing ground with one of the earlier, bigger prototypes.

These augmented real-world interfaces have the advantage that the figures and the game field can be left unaltered, which is not possible when, for example, using big touch screens [16] or a projection-based battlefield [14] (especially for game fields with 3D objects). Also, these alternatives do not offer the high accuracy required for locating objects in W40K.

4.2 Game Flow Virtualization in W41K

As mentioned before, the localization data is forwarded to our gaming application, which then processes the data (i.e., translating the recognized tag IDs into battlefield coordinates of the corresponding game objects). For game units with only one RFID transponder this is straightforward since the transponder position is the position of the unit. With multi-tagged objects such as tanks, however, this task is not that simple: we have to find a position and orientation of the game object that fits the detected tag positions as precisely as possible.

For each multi-tagged game object, the system knows the relative position of the tags on the object after an initial registration step. During actual gameplay, the system calculates the center of gravity for the set of recognized tag positions and aligns it with the center of gravity of the registered model (see Fig. 7). The system then simply measures the angles between the center of gravity and all tag positions, and averages over all angles. The resulting angle indicates the orientation of the game object. This algorithm worked extremely well in all our test scenarios.

2 http://www.tagsysrfid.com

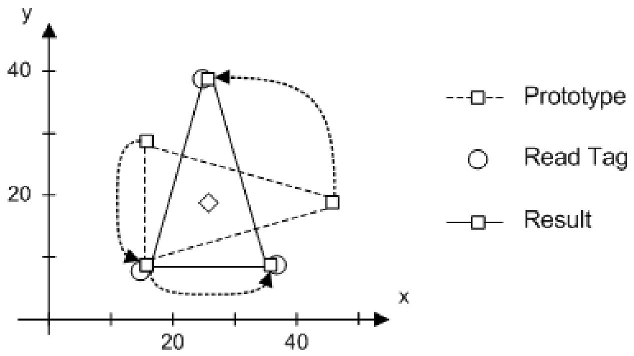

Figure 7. Determining the position and orientation of multi-tagged objects: the center of gravity for the unit and the detected RFID tags are placed on top of each other. The average of the angles between the center and the individual tags yields the required rotation of the object.

Once we have the positions and orientations of all game objects on the battlefield – in addition to the army units we also have the virtual representations of all static objects such as buildings – we can then verify if moves and attacks are valid. Thereunto, we have to inspect three things:

1. The distances:
   a) Is this unit allowed to move to this point?
   b) Is the target in range of the unit's weapons?
2. The viewing angle (only vehicles):
   a) Is the target within the viewing angle of the weapon? (Most weapons mounted on vehicles often have a viewing angle of 180°.)
3. Is the target in the line-of-sight of the aggressor?

Since all objects can be modeled as a set of lines (i.e., vehicles, buildings, terrain sections) or a circle (i.e., foot soldiers), checking these three constraints can be reduced to calculating the intersection of two lines. The small figures are placed on round bases (see Fig. 5 left); in this case we take the line that is orthogonal to the line-of-sight and goes through the center of the round base: since we know the diameter of the base, we can determine if the base of the target figure is within the line-of-sight of the attacker.

This information is used to support the players during movement, shooting, and close combat phases of the game. To properly assess if all rules are satisfied, we virtualize these phases and the game objects with their individual characteristics (e.g., strength, range, weapons, etc.). The phases can be represented in sequence diagrams, which give a good overview of how the game is structured. Fig. 9, for example, displays the diagrams covering the shooting phase. Note that these two diagrams do not cover all possible exceptions and rules of the shooting phase – including them would have rendered the diagrams rather confusing and hard to understand.

Based on the sequence diagrams, the gaming application is aware of the current game state and can now forward relevant information to the players and verify their moves without their having to manually measure distances, angles, etc. and cross-reference them with the individual and current capabilities of the involved units (e.g., health points, fallback condition, PSI support by special units, etc.): the players can simply make their moves and will be informed if and why a move is not valid according to the rules. To this end, a screen will display the current condition of the battlefield at all times, including buildings and terrain sections (see Fig. 10 top).

By simply moving the mouse cursor over a unit, a player can receive all relevant and up-to-date information[3] (see Fig. 10 left). Similarly, the screen will notify the players whose turn it is and what moves are possible: if a move is not allowed, the otherwise black "lines of movement" are highlighted in red (see Fig. 10 right).

4.3 Discussion

W41K meets most of the design guidelines presented in Tab. 3 rather well. The guidelines 4, 5, 7, and 11 are inherently met by the chosen technology (RFID technology) – also demonstrating how important the right choice of the technology is. The benefits of choosing and integrating this technology in the first place (criterion 1) should have become obvious in the previous section.

The players can also decide to disregard rule violations, as this can be part of a current scenario or simply the players' desire to bend the rules [4]. The idea of W41K is not to simply replace a rule book with a computer screen but allowing the players to focus on strategic decisions and social aspects by relieving them of cumbersome tasks and providing them with relevant and up-to-date information about all game objects – if *desired* –, thus realizing guidelines 3, 5, and 8.

To realize the physical augmentation, we iteratively developed and improved the prototypes to meet the required criteria: each component was subject to at least three iterations as discussed in the corresponding references. The design and implementation processes were realized in accordance with the augmentation cycle (see Fig. 2) and design guidelines 2, 6, and 10. The established game model reflects the high dynamics of this game (guideline 9, also cf. aspect no. 3 of Tab. 2).

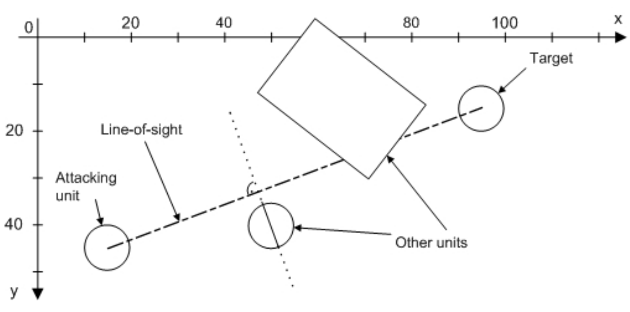

Figure 8. Determining the line-of-sight between objects.

[3] In fact, the general idea is take the concept of virtualization even one step further by representing each game object (i.e., the army units) by its own website [14].

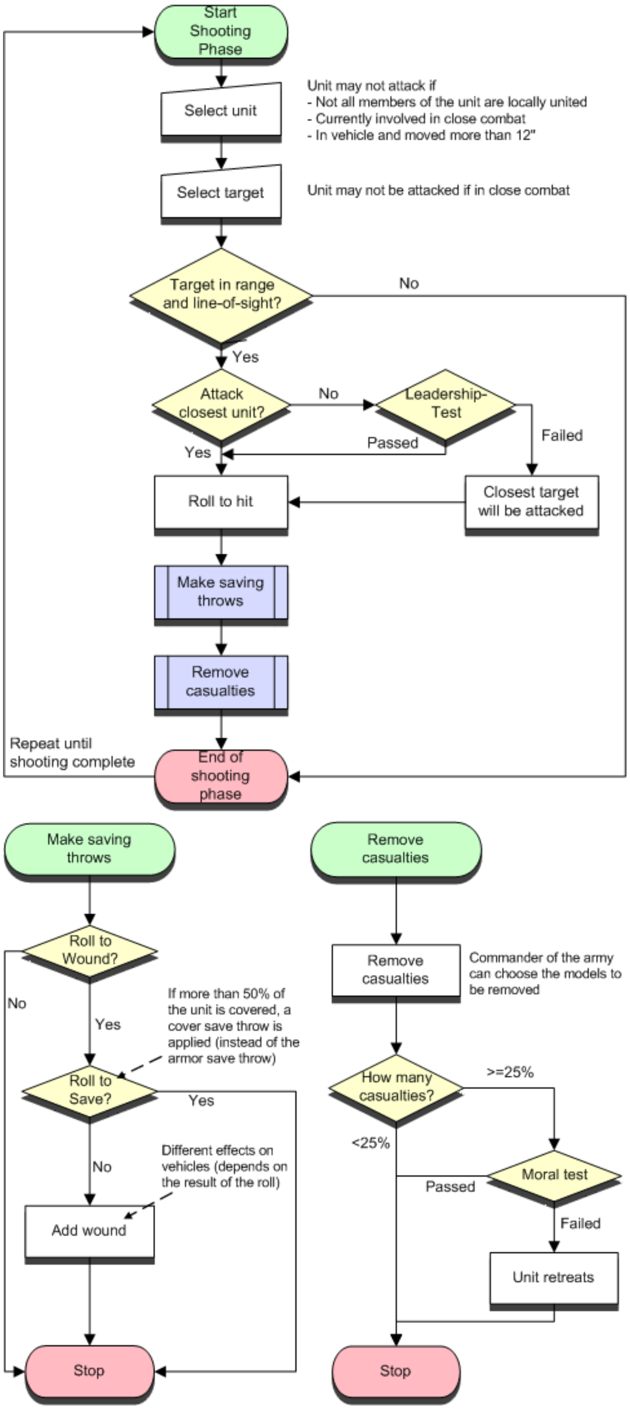

Figure 9. The sequence diagram for the shooting phase of W40K. The diagram at the top describes the complete shooting phase while the two diagrams at the bottom are extensions to the main diagram.

The only insufficiently met criterion is the one on secondary user interface (no. 12): our system uses a screen to display the positions of the units and the context-relevant information, which is not optimal in terms of natural, tangible interaction with the battlefield and the game objects. Also, to date, we have not realized a session management and game history yet, which is part of future work (also see section 5).

5 CONCLUSIONS

In this paper we established and discussed a set of guidelines for digitally augmenting traditional game environments. Based on our experiences in augmenting such environments, we suggested a two-step process comprised of *game flow virtualization* and *physical artifact augmentation*. In order to demonstrate how these guidelines can be practically applied, we designed and implemented an infrastructure that is capable of relieving the players of these cumbersome tasks and providing them with context-relevant information in-game, hence allowing them to focus on strategic decisions and the social aspects of the game.

The application of the approach and the design guidelines presented in this paper is, however, not limited to augmenting existing game environments. They can also be used to design completely new augmented games. If a game can be designed from scratch without any constraints, it features a higher degree of freedom with regard to the coupling of the physical and digital. Additionally, developers can adjust the physical shapes of the play objects accordingly, allowing them to make bidirectional adjustments – augmenting traditional game environments do only allow for unidirectional coupling (i.e., make the digital (technology) fit the physical objects).

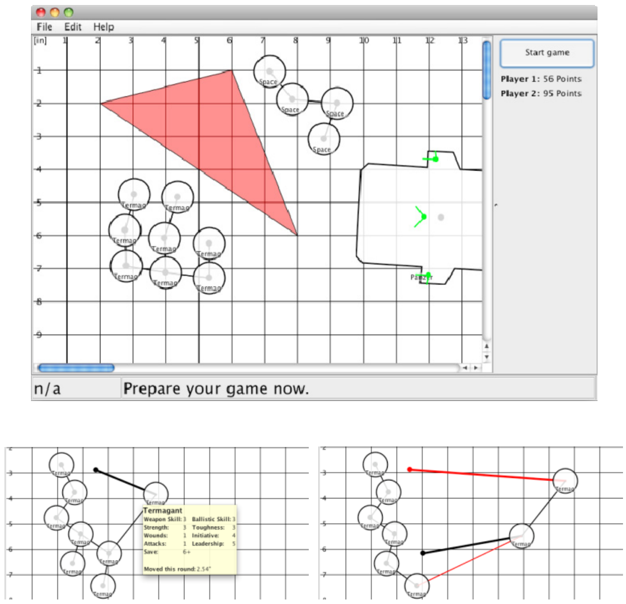

Figure10. The user GUI displaying recognized game objects (top). The user can easily receive additional information about the units when moving the mouse over a unit (left) and will be automatically notified if planned moves are invalid (right).

Our prototype serves as a fair example of how an augmented game environment might look, i.e., data is collected unobtrusively and neither the battlefield nor the game objects are perceptibly modified. While initial trials showed that the prototype works very well, we found several aspects that could be improved in subsequent design iterations:

- Reducing the time required for scanning the battlefield (approx. one minute at the moment).
- Increasing the dice ground: right now, we can read 5-6 dice at once, but the game sometimes needs over 20 dice rolled simultaneously.
- Encompassing all rules: though we have implemented all major rules, there are still several exceptions to be covered.
- Session management and game history (see section 4.3).
- Conducting an extensive user study: the prototype has not been tested under real circumstances.
- Possibly replacing the screen for displaying the information with a more naturally integrated interface (cf. section 4.3) or maybe use a projector to display the information directly on the game field. While a projector potentially violates the fourth design criterion (see Tab. 3), it might on the other hand provide for additional effects (i.e., visualizing explosions and exchange of gunfire) – this trade-off is certainly worth exploring.

Improving these aspects should further contribute to creating a truly compelling augmented game environment.

REFERENCES

1. Andersen, T.L., Kristensen, S., Nielsen, B.W., Grobaek, K.: Designing an Augmented Reality Board Game with Children: The Battleboard 3d Experience. In: Proc. of Interaction Design and Children, ACM Press (2004)
2. Broll, W., Ohlenburg, J., Lindt, I., Herbst, I., and Braun, A.: Meeting technology challenges of pervasive augmented reality games. In Proc. of 5th ACM SIGCOMM Workshop on Network and System Support For Games. ACM, New York, NY, USA (2006)
3. Cooper, A., Reimann, R., Cronin, D. About Face 3.0: The Essentials of Interaction Design. 3rd Revised Edition. Wiley & Sons. (2007)
4. Dunnigan, J.: Wargames Handbook: How to Play and Design Commercial and Professional Wargames, 3rd Edition. Writers Club Press, January (2000)
5. Floerkemeier, C., Mattern, F.: Smart Playing Cards – Enhancing the Gaming Experience with RFID. In Proceedings of the Third Intl. Workshop on Pervasive Gaming Applications - PerGames 2006, May 2006
6. Hinske, S., Lampe. M., Magerkurth, C., Röcker. C.: Classifying Pervasive Games: On Pervasive Computing and Mixed Reality. In: Magerkurth, C., Röcker, C. (eds.): Concepts and technologies for Pervasive Games. Shaker, Aachen, Germany (2007)
7. Hinske, S., Langheinrich, M.: An RFID-based Infrastructure for Automatically Determining the Position and Orientation of Game Objects in Tabletop Games. In: Magerkurth, C., Röcker, C. (eds.): Concepts and Technologies for Pervasive Games. Shaker, Aachen, Germany (2007)
8. Hinske, S., Langheinrich, M., Lampe, M.: Towards Guidelines for Designing Augmented Toy Environments. In: v.d. Schijff, J., Marsden, G., Kotze, P. (eds.): Proc. of the 6th ACM Conf. on Designing Interactive Systems (DIS 2008), ACM Press, New York, NY, USA (2008)
9. Hinske, S., Langheinrich, M.: Using a Movable RFID Antenna to Automatically Determine the Position and Orientation of Objects on a Tabletop. In: Proc. of EuroSSC 2008. Springer, Berlin Heidelberg NY (2008).
10. Hinske, S., Langheinrich, M.: RFIDice – Augmenting Tabletop Dice with RFID. In: *Journal of Virtual Reality and Broadcasting*, Vol. 5, No. 4 (2008)
11. Hinske, S., Langheinrich, M., Alter, Y.: Building RFID-Based Augmented Dice with Perfect Recognition Rates. In: Proc. of Fun and Games 2008. Springer, Berlin Heidelberg New York, October (2008)
12. Ishii, H., Wisneski, C., Orbanes, J., Chun, B., Paradiso, J.: PingPongPlus: Design of an Athletic-Tangible Interface for Computer-Supported Cooperative Play. In Proc. of CHI '99, pp. 394-401, ACM Press (1999)
13. Jegers, K. Pervasive GameFlow: Understanding Player enjoyment in Pervasive Gaming. *ACM Computers in Entertainment*, Vol. 5, No. 1(2007)
14. Kindberg, T., Barton, J., Morgan, J., Becker, G., Caswell, D., Debaty, P., Gopal, G., Frid, M., Krishnan, V., Morris, H., Schettino, J., Serra, B., Spasojevic, M.: People, Places, Things: Web Presence for the Real World. *Mobile Networks and Applications*, Vol. 7, No. 5 (2002)
15. Magerkurth, C., Stenzel, R., Prante, T.: STARS – A Ubiquitous Computing Platform for Computer Augmented Tabletop Games. In: Ljungstrand, P., Brotherton, J. (eds.): Video Track and Adjunct Proc. of the 5th Intl. Conference on Ubiquitous Computing (2003)
16. Mandryk, R. L. and Maranan, D. S.: False Prophets: Exploring Hybrid Board/Video Games. In CHI '02 Extended Abstracts on Human Factors in Computing Systems (CHI '02). ACM, New York, NY (2002)
17. Nieuwdorp, E.. The Pervasive Discourse: An Analysis. *Computer Entertainment*, Vol. 5, No. 2 (2007)
18. Nilsen, T., Looser, J.: Tankwar Tabletop War Gaming in Augmented Reality. In: Proc. of 2nd International Workshop on Pervasive Gaming Applications (PerGames) at Pervasive (2005)
19. Shneiderman, B., Plaisant, C.: Designing the User Interface: Strategies for Effective Human-Computer Interaction, 4th edn. Addison-Wesley, Reading, MA (2005)
20. Stenros, J., Montola, M., Mäyrä, F.: Pervasive games in ludic society. In: Proc. of Future Play, New York, NY, USA, ACM (2007)