# Using Prototypes in Early Pervasive Game Development

ELINA M.I. OLLILA, RIKU SUOMELA, AND JUSSI HOLOPAINEN
Nokia, Tampere, Finland

In this article we discuss various prototyping methods in early pervasive game development. The focus is on pervasive games that are played with mobile phones. Choosing the right prototyping method is crucial in achieving results that can be used for validating or developing further design ideas. In this article we give guidelines that help the selection process and give ideas on methods that can be used in different situations.

We have play-tested pervasive game prototypes using agile software prototype development methods, forum prototypes, and guided paper prototyping methods. We give examples of five pervasive games where these kinds of prototyping methods are used. In concluding, we compare the results and discuss their benefits and disadvantages in the game development process, that is, when the methods should be used and what should be considered when using them.

Categories and Subject Descriptors: K.8.0 [**Computing Milieux**]: Personal Computing – *General, games*

General Terms: Design, Experimentation, Human Factors.

Additional Key Words and Phrases: Mobile, game design, paper prototyping, physical prototyping, play testing, rapid prototyping, agile software development, preproduction

**ACM Reference Format:**
Ollila, E.M.I., Soumela, R., and Holopainen, J. (2008) Using Prototypes in Early Pervasive Game Development. *ACM Comput. Entertain.* 6, 2, Article 17 (July 2008), 17 pages. DOI = 10.1145/1371216.1371220
http://doi.acm.org/10.1145/1371216.1371220

## 1. INTRODUCTION

Prototyping is a commonly used design method in game development (e.g., Fullerton et al. [2004]). Prototyping pervasive games is often more difficult than prototyping traditional digital games that are played with a computer or console. In this article we focus on prototyping methods that can be used for testing and improving game concepts in the preproduction phase of the game development process (e.g., Koivisto and Palm [2005]).

Various definitions for pervasive games have been presented in the literature. For instance, in her dissertation McGonigal [2006] defines pervasive games as disruptive, highly visible, and often artistic events. She argues that ubiquitous games are often used as a synonym for pervasive games. Walther [2005] claims that ubiquitous games are a subgenre of pervasive games (which is closer to our view as well).

According to Montola [2005] pervasiveness in games can be defined as those that break the temporal, spatial, or social boundaries[1] of games. Spatial pervasiveness means that the game can be played in different places and the location can affect the game-play. It also means that games can be found in places where they are not expected.[2] Temporal pervasiveness means that the game

Authors' address: Nokia, Tampere, Finland, POB 100, FIN 33721, Tampere, Finland; contact author's email: Elina.ollila@nokia.com

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date of appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Permission may be requested from the Publications Dept., ACM, Inc., 2 Penn Plaza, New York, NY 11201-0701, USA, fax: +1 (212) 869-0481, permissions@acm.org
©2008 ACM 1544-3574/2008/07-ART17 $5.00 DOI = 10.1145/1371216.1371220
http://doi.acm.org/10.1145/1371216.1371220

[1] The boundaries here refer to the Magic Circle, a term coined by Johan Huizinga [1955].
[2] For instance, an advertisement could lead the player into a game.

can be played during everyday activities. Games can contact players when they are not actively playing. The social pervasiveness means that players can change their roles flexibly from being a nonplayer or an audience to being an active player, and vice versa. It can also mean that nonplayers are used as game resources; not all of these boundaries need to be broken for the game to be pervasive.

For the purpose of this article we chose to use Montola's definition. The prototyping methods that are discussed in this article can be used for any game, and are particularly recommended for pervasive (as defined earlier) games, or if the game is very innovative.

Considering the three aspects of pervasiveness (spatial, temporal, and social), it is easy to understand that prototyping pervasive games can be challenging. Pervasive games are often completely new kinds of games. When designing a first person shooter game with slightly different elements than implemented in other earlier games, the designers already have a rather good idea of what the game-play would be like. In the case of pervasive games, the designers have often never played games similar to those they are designing, and it is difficult to understand the game-play before trying it out in practice.

The more inexperienced the game designers are, the more difficult it is to imagine what kind of game-play will emerge when the players, and potentially the environment, interacts with it. However, even the most experienced game designers, such as Will Wright[3] [2004], say that they benefit from experimenting with physical prototypes when designing (nonpervasive) games.

In the iterative design process, play-testing games with prototypes is typically done for three reasons: to test game design ideas or concepts, to generate new design ideas, and to probe the attitudes, opinions, and behavioral patterns of potential players. When prototypes are tested, typically all these kinds of observations are made. In addition, prototypes are used to demonstrate ideas.

To enhance game design we have developed pervasive game prototypes during the early phases of game development projects. This article discusses using agile software development, ready-made software components such as forums, and physical or paper prototypes in pervasive game development; we also give guidelines for choosing the right methods.

The term "physical prototype" in this article covers all prototypes that are constructed with paper, miniatures, or actors – components that do not focus on using software. This method is sometimes also called *paper prototyping*, even if the prototypes contain objects other than paper. Augustin et al. [2007] call the method for developing very rapid prototypes that are thrown away in game development *game sketching*. They argue that prototyping is used to both prove that the team can build the game and to test whether the game ideas work. It is important to make a clear distinction between them; this is why Augustin et al. [2007] prefer to call the very early prototypes sketches rather than prototypes.

The article is organized as follows: First we discuss related work on prototyping games; we then describe agile software prototyping, software-component prototyping, and physical prototyping methods we used in this study, and give examples of prototypes that we developed using these methods. Before concluding, we give guidelines for selecting the right prototyping methods in pervasive game projects.

## 2. RELATED WORK

We studied related work to learn what kinds of prototyping methods were used in the early development of game and play applications. Paper prototyping is a commonly used game design method among game researchers and designers (see e.g., Fullerton et al. [2004] and Signman [2005]). Often, physical prototyping and testing are not conducted with the actual users, but with project members and other colleagues; it is also commonly used in usability testing of nongame applications. At times such testing is conducted with potential end users of the application. Variants of paper or physical prototyping of applications and services have been presented earlier. Ehn and Kygn [1992] demonstrated how combining prototyping and games can be used to improve communication of the concepts to end users. Iacucci et al. [2000] developed the method further by using a similar approach to that of Ehn and Kygn for testing mobile applications and services.

[3] Designer of the Sims.

Iacucci et al. [2000] found that playing the prototype as a game enhanced the user's understanding of contextual changes and the contexts of other users.

Testing pervasive games with physical prototypes is not straightforward, since the games can last for a long period of time and involve real-life activities. Höysniemi [2006] describes in her dissertation using the Wizard-of-Oz[4] method to play-test physical games. She found that Wizard-of-Oz prototyping was useful because it was flexible enough for demonstrating behavior patterns that the designers did not expect beforehand. For instance, when the players used swimming motions that the designers did not expect, the Wizard could change the prototype to support that kind of interaction on the fly. On the other hand, she noticed that the Wizard's conceptual and motor skills set limitations on what could be done with the prototype.

Ballagas and Walz [2007] used several prototyping methods in their location-aware city-exploration game, *REXplorer*. They used a board game prototype to successfully demonstrate the game, and noted that it was particularly useful for getting a feel for travel times, expressing spatiality, judging proximity of sights, and ensuring that the game was fun to play. Like Iacucci et al. [2000], they used event cards for simulating a more realistic environment. To test game interaction, Ballagas and Walz also prototyped the game at the locations where the game was supposed to take place. They used a GPS signal detector for designing the "hotspots" (i.e., the areas where something new was to happen in the game), and tested how the game worked in the hotspots with the Wizard-of-Oz method, whereby the wizard shadowed the players and input their locations.

Focus group discussions[5] and interviews were used to discover the attitudes and opinions of actual users, often when there was no implementation of the concept available. However, there has been criticism of this method, since it may be difficult for users to evaluate something that they have not seen or tried out in practice. (our previous study also supports this statement [Koivisto and Wenninger 2005]). Such focus group discussions are better at finding out the opinions, attitudes, and behavior of the players. The players should not be asked directly if they liked certain concepts [Ermi and Mäyrä 2005]. To make concepts more concrete in focus groups, scenarios (e.g., comic strips, [Lankoski et al. 2007]) or acting [Strömberg et al. 2004] should be used instead. Ethnographical studies were found useful in designing mobile leisure applications. Esbjörnsson, et al. [2004] used an ethnographical study when designing Hocman, a social networking application for motorcyclists on the road. They called their method *associative design*; its key idea is that ethnographers and designers work in very close collaboration, instead of just delivering reports on paper.

Pervasive games can greatly benefit from rapid software prototyping. Agile methods [Cockburn 2002; Beck 1999; Agile Alliance: http://www.agilealliance.com], which in recent years have become widespread in the software development industry, have helped in the design of pervasive games [Koskinen and Suomela 2006]. To minimize the risks in the development process, agile software focuses on making software in short iterations. Every iteration contains all parts of the software development process, including design, coding, and testing. Agile software development emphasizes working software, which allows the team to constantly measure the quality of the software. Such an approach has a great advantage in correcting design. If a feature is evaluated as inadequate, it is easy to detect and fix it early on in the process.

Agile methodologies can be applied to games as they are, but there are well-known problems in doing so. A typical game project consists of large teams, but agile methodologies apply best to small teams. However, actual game software development does not differ that much from traditional software development, and agile development has been successfully applied in game development as well. We found agile methods useful for game prototyping because they make it possible to change the functionality of the prototype quickly when needed. A sequential software development model like the waterfall model [Royce 1970] does not allow the flexibility necessary for the early development of prototypes.

[4] In Wizard-of-Oz prototyping, a prototype is controlled by a human. The human involvement is not usually visible to the test user.

[5] Here, we are *not* referring to product-interactive focus groups [Lee et al. 2004], where the attendees try out prototypes or existing products.

Ready-made software components were also used for game prototyping. Manninen [2000] published two studies in which a mobile game console and a card game were prototyped in a virtual environment. He found that using game engines like Unreal for game prototyping was a promising approach, particularly when the interaction with the physical environment did not need to be very complex. According to Manninen [2000], developing the prototypes was fast and made work in distributed locations possible.

## 3. PROTOTYPING STUDIES

In this section we present research done on rapid game development: one study was conducted with a ready-made software component, and in the other study two games were tested with physical prototypes. Later, we will compare the results and give guidelines for selecting the right kind of prototyping method.

### 3.1 Rapid Game Development

We conducted some experiments on rapid development of context- and location-aware games. Such games should react to the users' immediate environment via location or some other input. They are pervasive in nature, and since they deal with the real world, it is not feasible to first create the games and then test the final product. The real world is something that cannot be controlled in the digital domain, and the design process should take this into account by observing how the game prototypes actually react to the environment.

In our experiments we created location-aware games in 24 hours [Suomela et al. 2004] and context-aware games in one week [Koskinen et al. 2006]. Our focus was to design and implement a game in the given timeframe, which each group managed to do. The aim was not to create finished products, but to create working prototypes that the other participants could play at the end of the sessions.

The sessions focused on the very first phases of the game design process. The participants came up with the idea and immediately proceeded to make the first running prototype. We focused on working software, not on game design -- as a consequence the games were fairly simple, and only the selected game features were highlighted. However, these sessions were very successful in communicating the game design. At the end, every participant was able to test the game and immediately give feedback to the developers.

The prototyping sessions were hectic. Learning, design, and implementation followed each other very rapidly, and the time left was used for testing. This kind of a setup is useful for testing one or a few features in a game, but since the testing phase is reduced to a few hours, it is not useful for testing a game that takes a long time to play. Still, this approach gives valuable information on the feasibility of the concepts and on some of their features.

In yet another session [Koivisto and Suomela 2007], two experts developed three game prototypes in a single day, based on ideas given by visitors to an event. Out of the three games, one seemed to be good, one had problems in the design, and the last was nice but not interesting. In the context of this article, the game (called *Hot Potato*) with the design problems was the most enlightening. It involved a variable number of players and was persistent, so that it could, for instance, be played during a work day, among other activities.

The idea in *Hot Potato* is that one player at a time has a hot potato that he or she must hold on to for a certain period of time. A player cannot hold the potato forever because it becomes too hot and must be handed on to another player nearby before it burns one's hands. The initial analysis of the game design pointed out a few strong points. The game is very social, since the players are directly interacting with each other via a game object, and it would be easy to add multimedia content to the potato, making it aware of its history and past owners.

Problems began to appear after the prototype of the game was put on mobile phones. First, players who did not have the potato had nothing to do; second, if there were no other players in the proximate range of the sensors (Bluetooth), the potato could not be passed on; third, if the potato was on a device owned by a player who was leaving, the game stopped (the potato was out of the game).

There are many ways to counter these problems, like adding several potatoes to the game or sending potatoes over a distance, and so on; but they would all radically change the game concept.

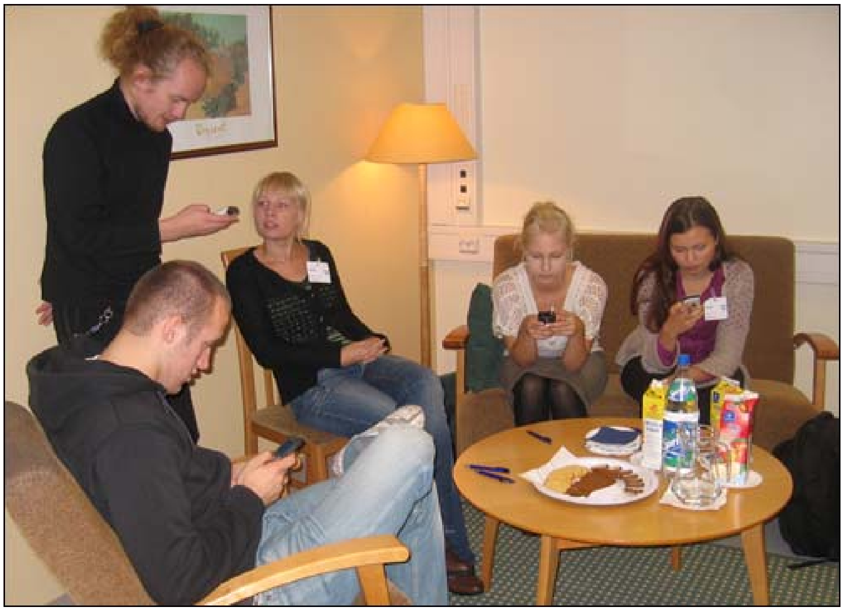


Fig. 1. Play-testing *PhotoQuiz*; an observer's  point of view.

These features could have been spotted at design time, but they were much easier to spot with a functioning prototype that was built in two hours by two developers.

Feedback is very important in pervasive context-aware games, since the games can act very differently in the real world than they were initially designed to do. If the game requires a certain real-world condition in order for content to trigger, the condition might never occur in the actual game.

Another software prototype, called *PhotoQuizz*, was play-tested in 2006 in two focus groups at the Technology Research Centre of Finland (VTT). A typical view of the test setting appears in Fig. 1. The players sat around one table, with two facilitators in the same room; other observers were behind a transparent mirror. Seven players participated the sessions and played two games, one was *PhotoQuizz.*

The game was developed at the Helsinki Institute of Information Technology by the MoMUPE project. In this game, the players took pictures of objects and other players tried to guess the words that the pictures represent. We organized two test sessions with researchers at VTT that lasted for two hours each.

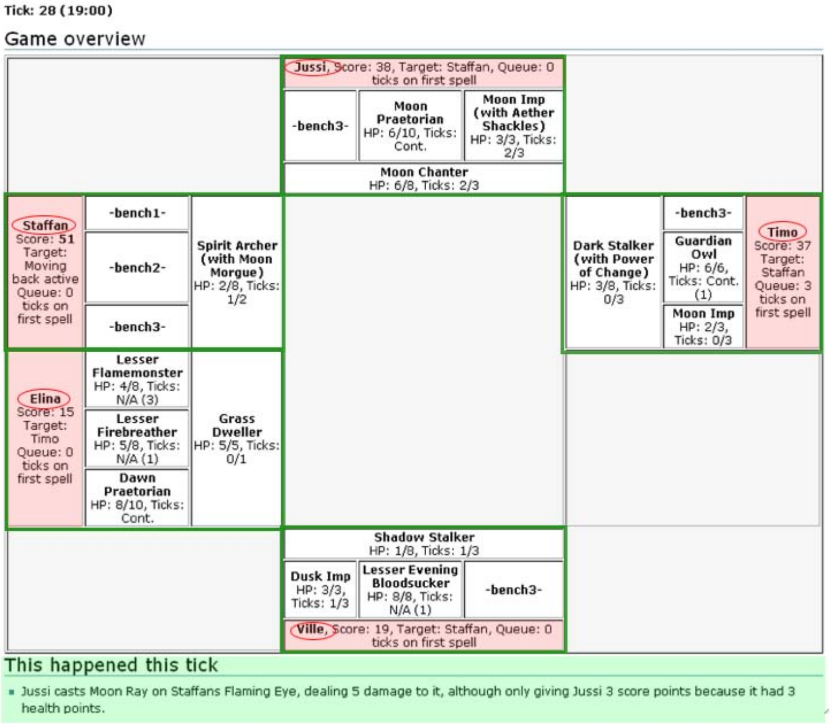
We received feedback from the test sessions and made observations on the attitudes and opinions of the players and on game-play and usability issues. We received some data on ergonomic and technical issues as well. The game-play did not need much explanation and was easy for the players to understand since they got to try the game out.

### 3.2 Prototyping with Ready-Made Software

In the IPerG project [IPERG] we prototyped a pervasive game called *Mythical*: *The Mobile Awakening*,[6] which is played with mobile phones. The game was tested with both a physical prototype and a prototype consisting of a ready-made software component: a web forum.

The target of the prototyping sessions was to design and test one of the main modes of play in the game. The main design requirements for the entire game were to support blending activity and interrupt-ability (i.e., so that the game could be played meaningfully even when the player's main focus was on other tasks, e.g., attending a lecture). These requirements were fulfilled by making the

[6] http:// www.mythicalmobile.com



**Tick: 28 (19:00)**

Game overview

Jussi, Score: 38, Target: Staffan, Queue: 0 ticks on first spell

**-bench3-**

**Moon Praetorian** HP: 6/10, Ticks: Cont.

**Moon Imp (with Aether Shackles)** HP: 3/3, Ticks: 2/3

**Moon Chanter** HP: 6/8, Ticks: 2/3

Staffan Score: **51** Target: Moving back active Queue: 0 ticks on first spell

**-bench1-**

**-bench2-**

**-bench3-**

**Spirit Archer (with Moon Morgue)** HP: 2/8, Ticks: 1/2

**Dark Stalker (with Power of Change)** HP: 3/8, Ticks: 0/3

**-bench3-**

**Guardian Owl** HP: 6/6, Ticks: Cont. (1)

**Moon Imp** HP: 2/3, Ticks: 0/3

Timo Score: 37 Target: Staffan Queue: 3 ticks on first spell

Elina Score: 15 Target: Timo Queue: 0 ticks on first spell

**Lesser Flamemonster** HP: 4/8, Ticks: N/A (3)

**Lesser Firebreather** HP: 5/8, Ticks: N/A (1)

**Dawn Praetorian** HP: 8/10, Ticks: Cont.

**Grass Dweller** HP: 5/5, Ticks: 0/1

**Shadow Stalker** HP: 1/8, Ticks: 1/3

**Dusk Imp** HP: 3/3, Ticks: 1/3

**Lesser Evening Bloodsucker** HP: 8/8, Ticks: N/A (1)

**-bench3-**

**Ville**, Score: 19, Target: Staffan, Queue: 0 ticks on first spell

This happened this tick

- Jussi casts Moon Ray on Staffans Flaming Eye, dealing 5 damage to it, although only giving Jussi 3 score points because it had 3 health points.

Fig. 2. Screenshot of the final forum prototype (color added to make the table more readable). The table shows the game situation in one tick; it shows all the players (the names are circled), the game situation (each player's score, current target, spell queue, and minions' status), and actions (here a player has attacked another player).

mode of play a slow-update one, where the player did not have to pay attention to the game all the time. The problem from the prototyping perspective was that a slow-update game could take days or even months to finish. The development team decided to use physical prototyping to test the core game mechanics and web-based forum prototyping to quickly test whether the core mechanics worked in a slow-update version before nailing down the game-play features for the next version of the game. The team designed and tested several versions of physical and web-based forum prototypes before deciding on the core game mechanics for the final forum prototype (Fig. 2). In each version, the core game mechanics were first tested and modified with a physical prototype before testing with the web-based forum prototype. The development team conducted the tests, as it was considered too early to bring in external testers since the game-play features were still mainly undecided.

The game-play features changed considerably between each version as new game-play problems and opportunities were identified. The main design conflict was between making the game complex enough to be interesting while keeping actions and representations simple enough to make it playable on a mobile phone. For the last prototypes, the team's graphic designer made mock-ups of the mobile phone user interface to test whether it was possible to display the required game state information on a small screen.

During the whole process it was necessary to use physical prototyping to test whether the core game mechanics worked at all and then use the forum prototype to test if the game-play was interesting enough even in the slow-update mode. In hindsight, more

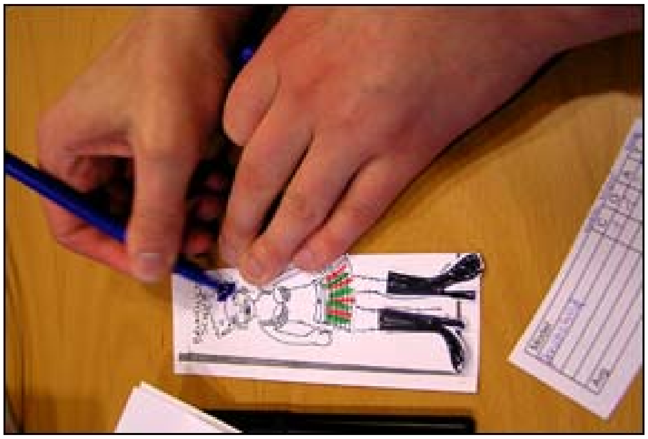


Fig. 3. A player customizing his model in game 1.

attention should have been paid to representational complexity in some of the intermediate versions. Even though some features were interesting, they had to be rejected because they would have been too difficult to display on a mobile phone screen. It would have been good to make quick UI mock-ups before every forum test to see if the design was at all feasible on a mobile phone.

### 3.3 Physical Prototyping and Guided Physical Prototyping

In the first physical prototyping study we play-tested (with six colleagues) a physical prototype of a multiplayer online mobile game. The game was pervasive in the temporal sense, and could be played over a long period of time; the in-game communication continued over the course of the day during which the player performed other real-life activities. The game was targeted towards female players, as the theme was managing fashion models.

Play-testing was conducted in 2005 in Finland, and took two hours. The physical prototype was constructed of paper, pens, and an Excel sheet for calculating the results of the players' actions. Fig. 3 shows how a player customized one of the game characters he controlled.

The second study involved three researchers and eight potential players. The game was a pervasive massively multiplayer online game (MMOG) called *GED* (for *Garden of Earthly Delights*) [Koivisto and Eladhari 2006]. The game was designed to be played on a mobile phone and a stationary computer. Each play-testing session (conducted in Sweden in 2005) lasted 1.5 hours on average; most of the test subjects were university students.

The prototype was constructed of paper, and the researchers had three roles: one was a "computer" who changed the screens that the player would see, one was a storyteller who set the player up in different kinds of situations, and one was an observer. A picture of a typical session can be seen in Figure 4.

Both physical prototyping studies were very fruitful in finding ideas for improvement and problems in the game design. The *Fashion* game prototype did not demonstrate the user interface in detail and there were no comments related to usability. The second prototype concentrated on evaluating the game user interface as well.

Observations from the play-testing session of the *Fashion* game concentrated mostly on game-play issues (which was the purpose of the testing session, as testing was con-

Fig. 4. Play-testing game 2: A player, storyteller, and "computer."

ducted very early in the design process while the team was still working on the game design). The test results were used to validate the playability of the game concept and to develop it further. This was done, and the game concept was "sold" to a third party to for further development. Unfortunately, it was never made into a real product.

The pervasive MMOG prototype provided a lot of feedback on the game-play and user interface issues, but there were also plenty of observations on the attitudes and opinions of the real players. For example, the game was designed to be collaborative, but there was feedback that at times the players wanted to play the game alone as well.

Play-testing the *Fashion* game brought out 51 problems, opinions, and ideas; play-testing *GED* turned up 81 findings (however, considerably more time was spent on organizing the testing sessions). There are two reasons for this. First, in usability studies, the number of *new* findings gets rather small after testing the application with five test users [Nielsen 1994]. In play-testing mobile games, we noticed that often it is enough to test the game with six players to find most of the issues related to playability and usability. However, in this case we wanted to involve more players to get a better understanding of their opinions and attitudes as well. Second, most of the findings in the case of the *Fashion* game were ideas for its improvement. Experts are often used to play-test the physical prototype in order to create new ideas as well as to evaluate existing ones.

### 3.4 Comparing Test Results

The physical prototypes that were used to test the *Fashion* game and the very early versions of the *Mythical* game were very effective in testing and generating ideas. Several fixes were done based on the feedback. Because we managed to recruit a colleague who had previous experience as a fashion model to test the *Fashion* game, we received expert feedback on potential users and on the theme of the game. But we could not gather reliable data on the culture of the players or their attitudes and opinions. When play-testing *GED* (pervasive MMOG) with potential players, however, we received a lot of data on their opinions and attitudes.

The physical prototype of *GED* was not very good at revealing potential problems related to environment and movement in the real world. The results could have been improved by using a real- world map, miniatures, and event cards, as in related studies mentioned earlier. However, even then the findings would have been somewhat limited by the researcher's imagination (e.g., the kinds of events that could happen). Also, in the case of *GED*, the virtual game world was overlaid on the real one and the interaction between them was very difficult to test with a physical prototype and real players.

The forum prototype was very effective in play-testing slow-update prototypes with a distributed team. It did not require any specific software for the test players to participate. When some of the players forgot (frequently) to update their actions, the facilitator backed them up and simulated them.

We found that it is a lot easier to see the technical problems and those related to the context in which the game is to be used if it is tested with a software prototype. For instance, in the case of *Hot Potato*, the problem was lack of other players in the environment, which could have been difficult to see via the physical prototyping method. Physical prototyping is often organized for a fixed number of participants for a fixed period of time. Since pervasive games can include one or more of three specific features--temporal, spatial, and social--these variables should also be changed in the test setting. Some simulation can be done when using the guided physical prototyping method; but the scenarios are designed by the test organizers, who cannot always predict the problems or situations that may arise.

### 3.5 Comparing Resource Intensity

The purpose of creating game prototypes is to provide a tool so that game mechanics can be tried out in practice before there is a real implementation of the game. In the early phases of a project, the aim is to get something working fast so that the design ideas can be tested either by the team or by potential players. This leads to an important question: How fast is fast, and how many resources are needed to create something that is usually thrown away after it is tried out?

When using rapid software prototyping platforms or tools to create game prototypes such as MUPE [Suomela et al. 2004], simple software prototypes can be developed within a couple of hours, as shown in our example of the *Hot Potato* game earlier. However, before they can be used efficiently for prototypes, software frameworks must always be learned first.

Provided that developers work with tools that are familiar to them, a game prototype can be created quickly; but "quickly" can refer to weeks, days, or hours, depending on the complexity of the task. MUPE is a platform that is tailored for pervasive games, and it is possible for an experienced developer to create a working prototype in hours.

Developing simple physical prototypes of the core game-play can be rather fast; developing the materials for testing the *Fashion* game took approximately one day. The work included drawing pictures of the characters' clothes, creating character and team sheets for the players, creating a spread sheet to calculate results of rounds of the game, recruiting colleagues to test the game, and copying material. The test session itself lasted two hours; five test players and the organizer participated. The approximate amount of working time to create and test the prototype took 20 hours.

Developing and testing the prototype of the pervasive MMOG required more effort (play-testing the game consumed 120 hours, and the overhead, e.g., sending emails before testing, was not counted by Koivisto and Wenninger [2005]). One reason for taking such a long time was due to involving "real users" to play-test the game, and testing it one player at a time; another is that this prototype was more complete than the first, since it demonstrated how the user interface in the game would work. The pervasive MMOG concept was also slightly more complex than that of the *Fashion* game.

In the case of the *Mythical* game, testing each of the early physical prototype versions took three persons between one and two hours. The forum prototypes took from one to three days to play through, with one person as facilitator and two to five players. The forum prototypes required that the players spend a minute or two every couple of hours checking the game situation and playing the game, while it took the facilitator ten to twenty minutes every two hours, excluding night time, to synchronize the game state.

## 4. CHOOSING THE RIGHT PROTOTYPING METHOD

Prototypes should be created as early as possible in the game project. Inventiveness, new technologies, use of physical and social context, all increase the need for prototyping, since there are more unknowns in the design. The more inexperienced the game designers the more important the prototyping. However, even the most experienced game designers use and benefit from early prototyping.

We presented several prototyping techniques in this article, some used in our own studies and some by other researchers. Based on our own studies and related research, we recommend the following guidelines for choosing prototyping methods for pervasive games. To some extent the guidelines can be applied to nonpervasive games as well.

### 4.1 The Purpose of the Prototype

*The desired results*. The early prototypes are typically used in the iterative game process for validating ideas, creating new ones, or probing attitudes, opinions, and behaviors. As stated earlier, every test session with a prototype usually generates all of this data. However, the choice of prototyping method and test subjects will affect the results.

Obviously, the prototype, whatever it is, will provide more reliable data on the attitudes, opinions, and behaviors of the end users if it is tested with the potential players of the game. Hence particular attention needs to be paid in the recruitment process to make sure that the test subjects match the target group of the game. Some data can be gathered when using colleagues as well, and some of it may be reliable if the target group is similar to the test subjects. But it should always be remembered that game designers are typically very experienced gamers and represent only a very small specific group of players.

Table I. Choosing the Prototyping Method by the Desired Result (any prototyping method can be used)

| *Purpose of the prototype* | *Test subjects* | *Required prototype completeness* |
|---|---|---|
| Probe attitudes, opinions, and culture of users | Real players | Medium; complete enough for players to understand; focus groups and ethnography are useful too. |
| Generate ideas | Experts, players in focus groups, and ethnography | Low; sketchy prototypes encourage idea generation; good if change on-the-fly is possible. |
| Test ideas | Experts and real players | Medium; good if change on-the-fly is possible. |

The less polished prototypes are useful for generating ideas. When the prototype does not look like a finished product, it is easier not to comment on its looks [Snyder 2003] but to concentrate on its design instead. If the prototype can be changed on-the-fly, like the one by Höysniemi [2006], the test session can even be adapted to the new and unexpected behavior of the players. Our experience in testing prototypes with colleagues is that they are enthusiastic in presenting new ideas which could be used to design better games.

*The prototype's target audience*. In addition to testing games with potential players, prototypes are often used to demonstrate the game to stakeholders, such as clients, in the project. These kinds of prototypes should be instantly playable and the players should be able to complete something within a few minutes. If this is not possible, it may be a good idea to demonstrate the game-play with a video. Augustin et al. [2005] highly recommend that prototypes used to demonstrate the game to third parties and those used to generate ideas or to test them (game sketching) not be confused.

## 4.2 The Game Type

The game type has a huge effect on choosing the right prototyping method. Earlier, we said that pervasive games break the boundaries of "traditional" games in three ways: temporally, spatially, and socially. The prototypes that need to be built depend on these aspects, as shown in Table II.

Social games are played with multiple players, and sometimes the other players can be simulated (as in our GED example); however, real players will provide more realistic data. Some pervasive games that break the social boundaries of game-play involve a fixed number of players, but more often these games involve a varying number of players.

Persistent games can be played over a long period of time, typically with varying numbers of players, and possibly in various locations. The forum prototype that we presented earlier worked well in such circumstances. Due to the extended playing time, physical prototypes can be difficult to use for persistent games; but the basic game mechanics are easy to test in a "blitz mode" (i.e., quick mode; the physical prototype for the forum prototype was tested in "blitz mode" before it was built).

It should be possible to play location-based games at various locations, which can be simulated as well (as in the GED example). However, it may be difficult for the test players to completely understand what the game-play should be if the game does not require actual movement. If the game breaks the spatial boundaries of game-play with context-aware game-play, then sensors are needed, which can be simulated as well; but using rapid software prototypes will provide more reliable data.

Table II. Requirements for Prototyping.

| *Game feature* | *Prototyping requires* |
|---|---|
| Social/multiplayer | Multiple players |
| Social/varying number of players | 0-N players, changing over time. |
| Temporal/persistent | Playing the game over a longer period of time; involves a varying number of players. |
| Spatial/location-based | Varying locations. |
| Spatial/context-aware | Varying use conditions based on selected sensors. |

In the following, we list the game features that affect the choice of prototyping method.

*Context-awareness.* If the game-play relies much on context-sensor input (e.g., location or temperature), it is often more feasible to create a software prototype. Such changes are difficult to simulate with a physical prototype, and the test organizers cannot always think of all the relevant cases that could occur. If real sensors cannot be used, the Wizard of Oz method can be useful.

*Discreteness.* If the game is very discrete, that is, if events happen in an easily predictable manner and do not depend on mathematical functions, it is easy to test the game with physical prototypes. However, if the game events depend on continuous functions, (e.g., with respect to location), it is better to test the game with software, since the outcome of these functions would be difficult to predict. An example of a discrete game could be a location-based game where the events happen in hot spots; an example of a continuous game could be a location-based game where the power of a spell depends on the player's location; $f(x, y, z)$.

*The level of technical novelty.* Using new technologies is always a risk and adds uncertainty to a project. The way technology will work in certain kinds of situations cannot always be anticipated. In the development of the GED game, one of the major obstacles to testing its software prototype was the network operator's unwillingness to provide the location technology that the game was going to use. In their study, Ballagas and Walz [2007] assessed the problem of technical uncertainty by using a specific GPS scanner to find the feasible places for hotspots in their game.

*Social or behavioral novelty.* If the game puts the players in new kinds of social situations or requires them to change their behavior, some feedback from the players can be gathered in focus group discussions or ethnographical studies. Scenarios, acting, comics or videos can be used to make focus group discussions more concrete (e.g., Ermi and Mäyrä [2005]; Strömberg et al. [2004]). Physical prototypes can be played, and may help us to better understanding how game-play situations could develop. For instance, when play-testing the GED game, many of the players said that they would not like to do "borderline" gaming all the time.

It can sometimes be difficult to imagine or consider new kinds of situations, hence it is more effective to actually put the players in the actual situation. Ballagas and Waltz [2007]: used Wizard of Oz prototyping for the REXplorer game, In play-testing they found that some of the older players felt awkward gesturing with the magic wand in public spaces and decided to provide an alternative way of casting the spells from a menu.

Abowd et al. [2005] tested a pervasive application that was not a game but a short-term memory aid used for recording conversations. They tested its social acceptability by acting out potential situations with real users. After the conversations, the users were asked how they would have felt had the conversation been recorded to help the other person remember it. Abowd et al. called this method *paratyping*, as it is a prototyping method that does not involve a functional prototype. They found that the method helped the test subjects relate to the questions. Such paratypes could also be useful in probing the attitudes and opinions of players of pervasive games.

*Complex interactions between various gaming platforms.* When testing the GED game with a guided physical prototype, we noticed that prototyping complex interactions between physical and virtual game worlds could be difficult. In GED, the virtual game world was mapped on the physical one, and during the test session the players could move in both worlds [Koivisto and Eladhari 2005]. When play-testing GED, it was particularly difficult to understand how players using different platforms to play the game could interact. Complex interactions between the virtual and physical worlds can be difficult to demonstrate with physical prototypes.

*Persistence***.** As stated earlier, it can be difficult to realistically test persistent games that run over long periods of time with physical prototypes only. However, some results can be obtained by playing the games in a faster mode; this can be particularly useful when testing the core game mechanics. Physical prototyping of the game in blitz mode was done successfully for testing the first versions of the *Mythic* game. Later, the forum prototype worked very well in predicting what game-play would be like when mixed with the players' everyday activities. Persistence can be also simulated, as in testing the GED game.

*Player-to-game interaction.* Player-to-game interaction is generally easy to test with physical prototypes since it often involves testing the core mechanics of the game and because parts of it can be tested separately with simple prototypes. If the focus of the game is manipulating the game objects physically, as in dexterity-based games or moving in the game world, it would be difficult to imagine what gameplay would be like with only a physical prototype. Fullerton [2006] play-tested the *Clouds* game with a software prototype very early in the project. Physically manipulating the clouds on a computer

Table III. Selecting Prototyping Methods Based on Game Type

| *Game type* | *Prototyping method* |
|---|---|
| Context-aware (sensor input needed) | Often easier to implement as a software prototype; Wizard of Oz prototyping is a good alternative- |
| Discrete (events occur in predictable manner) | Physical prototypes as well as software prototypes. |
| Continuous (e.g., events are functions of location and other sensor input) | Software prototype is useful. |
| Technically innovative | Software and/or hardware should be used early to test technical aspects. |
| Social novelty | Real users should be involved in realistic situations. Both software and physical prototypes can be used, e.g., Wizard-of-Oz prototyping or paratyping. Can also be a supported with interviews, focus group discussions, and ethnographic studies. |
| Complex interaction between various gaming platforms | Can be difficult to demonstrate with physical prototypes. |
| Persistent, long-term | Software prototypes or prototypes with software components are good. Testing with physical prototypes is difficult but can be useful in testing core mechanics. |
| Player-to-game interaction: dexterity-based games | If manipulating game objects physically is central in the game, as in dexterity-based games like Tetris, software prototype is needed. |

screen is a central part of game-play, and would have been difficult to experience with a physical prototype. Testing arcade games like *Tetris* with physical prototypes could be challenging, but rather easy to prototype with software.

In Table III we summarize the selection criteria for prototyping methods on the basis of the game type.

### 4.3 The Project Type

*Skills of the project group.* The skills of the project group will affect the kinds of prototypes that can be created. The developers' skills will affect the quality of the prototypes dramatically, particularly for rapid software prototyping. Skill is also required for building and testing physical prototypes. When the team is lacking some competences that would be necessary to build a suitable prototype, it can be necessary to hire outside people to implement them. For instance, in the study by Strömberg et al. [2004], the research group decided to use actors in their prototype and hired them from a local theater.

*Structure of the project group***.** When a project team works in one location, it is easy to create prototypes quickly. In the case of a distributed team, it is more difficult. The forum prototype of the *Mythical* game demonstrates how very early prototypes can be implemented online. Virtual worlds can offer new possibilities as well, as demonstrated by Manninen [2000]. In our earlier study [Koivisto and Wenninger 2005], we experimented with conducting focus group discussions in virtual worlds, with promising results. For the guided physical prototyping of the GED game, a lot of the time needed for testing was used by the three facilitators in traveling to Stockholm (travel time and other overhead were not included in the 120 hours needed for the actual testing).

### 4.4 The Phase of the Project

It is very useful to try out ideas quickly in the early phases of the project. The prototypes can be very simple, either physical or software, and it is often more fruitful to test them with colleagues, since the prototypes may be difficult for outsiders to understand. Recruiting people who match the game's target group takes time, and when the game concept is not yet clear, it is a good idea to quickly run a few experiments before recruiting outsiders for the test sessions. However, the real users should be involved in the process as soon as possible.

Table IV. Project-Related Guidelines for Selecting a Prototyping Method

| *The project* | *Impact on prototyping* |
|---|---|
| Skills of the project group | Skills limit what can be done but if it seems crucial to create a certain prototype hiring resources is recommended. |
| Structure of the project group | Distributed projects can benefit from using software prototypes over the Internet. Virtual worlds can be used to both demonstrate products or conduct user research (e.g., focus groups). |
| Phase | All kinds of prototypes can be used once the idea of the game concept has been created. Prototypes should be created as early as possible, preferably along with the concept . |

Ethnographical studies and focus group discussions with potential players can be also very useful early in the project, even before any design is done. Such methods can be used to see what direction the game design should take and to understand the culture, opinions, and attitudes of the game's target group.

Software prototypes can be created and are useful in any phase of the project, even right after the development of the game concept itself, assuming that the project team has the skill or external resources needed for creating a simple prototype.

The guidelines related to the project are summarized in Table IV.

## 5. DISCUSSION

In the future, we expect to see more pervasive games and pervasive game research. Additionally, we predict that mainstream games will have more pervasive features. People who play them do not necessarily consider them pervasive games. For instance, once GPS (General Position System) becomes more common in mobile phones, it is also probable that more mobile games and applications that use location features will be created. As games like *Alternate Reality* gain more publicity, it may be that attitudes towards gaming will also change, and more people will actively seek games in places where they are not supposed to be found. A large part of the world's population has been playing digital games since they learned to walk, which also changes attitudes towards playing digital games in general.

All of this means that pervasive games – or rather games with pervasive features - will be a significant research topic. We would like to see more research on experimenting with prototyping methods for pervasive games and features.

## 6. CONCLUSIONS

We presented examples of physical prototyping, rapid software development, and use of ready-made software to build pervasive game prototypes in the very early phases of a game project. The focus was on testing simple game ideas or complete game concepts, rather than on generating ides in the early phases of the game development project when the final game design is still very much undecided.

Based on these studies and earlier research, we presented guidelines on selecting the right prototyping method. We found that choosing the right method depends on the following considerations: the purpose of the prototype, the game type, the project type, and the phase of the project. Guidelines for selecting prototypes were given on the basis of these considerations.

Both physical and software prototyping can be very useful. In our experiments we saw the value of early prototyping (which was also cited in earlier research). We also noticed that particularly in the case of pervasive games, where interacting with the player's environment is often prevalent, software prototyping was valuable in the conceptual or early design phases of the project. When testing the *Hot Potato* game (software prototype) and GED (guided physical prototype), we noticed that real-life events and interactions were often easier to understand with software prototypes. Even when guided physical prototyping is used and the facilitators present scenarios to the users, the test does not often reveal other problems that the test situation is not designed to (e.g., if the test situation does not take into account that there might not be always other player's around, the problems that can arise in that kind of situation would not be revealed by the test setup).

Both physical and software prototyping can be very time-consuming when the whole game is prototyped and is complex. The physical prototype-testing of the GED required approximately 120 working hours, and overhead (traveling time, coffee beaks) was not counted. It may sometimes be appropriate to just quickly create a software or physical prototype of part of a game (e.g., concentrating on the core game-play). Physical and software prototyping methods can often be combined.

We also discussed using “real” players and professional test players. Players who belong to the target group of the game usually provide more relevant data and are useful for understanding the players’ attitudes, opinions, and behavior. Using professional test players (e.g., colleagues) for testing enables faster iteration, and is also beneficial when new ideas are needed. If the prototype is very incomplete, it can be difficult for outsiders to understand, so it is useful to have both kinds of test players in the same project.

There are software prototyping platforms available, such as MUPE, that enable a faster development phase. Such tools can enable prototype development that is as fast or even faster than physical prototyping. Testing with software can also be more efficient if the prototype is complex, since otherwise a lot of guidance would be needed for testing a physical prototype (e.g., GED). However, it must be noted that learning to use the tools takes time; fast development time is typically achieved after learning to use the platform – which may take a few hours or even weeks, depending on the developer’s skill and the development tool.

In addition to software development platforms, complete software products such as forums or virtual environments can be used to test games and ideas. Our example of the *Mythical*: *The Mobile Awakening* forum prototype showed that this kind of approach enables quick testing of early game features in distributed teams.

## ACKNOWLEDGMENTS

We give special thanks to Tarja Systa and Mark Ollila for valuable comments on this article. We also thank Hanna Strömberg, Veikko Ikonen and Kari Heikkinen who worked with us on developing the evaluation and design process in the MoMUPE project. The Projects MoMUPE and IPerG were invaluable in making it possible to create most of the prototypes in this article. We also thank the project members who worked with us in creating the prototypes: Lassi Seppala, Kai Kuikkaniemi, Marko Turpeinen (PhotoQuizz), Ville Nenonen, Hannu Korhonen, Timo Nummenmaa, Robin Milding, Johan Peiz, Johan Olsson, Alastair Hapshire, Alain Becam (Mythical), Ari Koivisto (all the other MUPE prototypes in this article), Mirjam Eladhari, and Christian Wenninger (GED).

## REFERENCES

ABOWD, G.D., HAYES, G.R., IACHELLO, G., KIENTZ, J.A., PATEL, S.N., AND STENVENS, M.M. 2005. Prototypes and paratypes: Designing mobile and ubiquitous computing applications. *IEEE Pervasive Computing 4*, 4, 67-73.

AUGUSTIN, M., CHAUNG, G., DELGADO, A., ORTEGA, A., SEAVER, J., AND BUCHANNAN, J.W. 2007. Game sketching. In *Proceedings of DIMEA 2007*: *Second International Conference on Digital Interactive Media In Entertainment,* 38-43.

BALLAGAS, R. AND WALZ, S.P. 2007. REXplorer: Using player-centered iterative design techniques for pervasive games. In *Pervasive Games: Applications*, C. Magerkurth and C. Röcker (eds.), Shaker, Aachen.

BECK, K. 1999. *Extreme Programming Explained*. Addison-Wesley, Reading, MA.

COCKBURN, A. 2002. *Agile Software Development*. Addison-Wesley, Reading, MA

EHN, P. AND KYGN, M. 1992. Cardboard computers; mocking-it-up or hands-on the future. In *Design at Work: Cooperative Design of Computer Systems*, Lawrence Erlbaum, Mahwah, NJ.

ERMI, L. AND MÄYRÄ, F. 2005. Player-centered game design: Experiences in using scenario study to inform mobile game design. *Int. J. Computer Game Research 5*, 1 (Oct).

ESBJÖRNSSON, M., JUHLIN, O., AND ÖSTERGREN, M. 2004. Traffic encoutners and hocman: Associating motorcycle ethnography with design. *Personal Ubiquitous Computing 8*, 92-99.

FULLERTON, T., SWAIN, C., AND HOFFMAN, S. 2004. *Game Design Workshop: Designing, Prototyping and Playtesting Games*. CMP Books, San Francisco, CA.

FULLETRON, T. 2006. That Cloud game: Dreaming (and doing) innovative game design. In *Proceedings of the Sandbox Symposium 2006* (Boston, MA).

HÖYSNIEMI, J. 2006. *Design and Evaluation of Physically Interactive Games. Dissertations in Interactive Technology, 5*; University of Tampere.

HUIZINGA, J. 1955. *Homo Ludens: A Study of the Play Element in Culture*. Beacon Press, Boston, MA.

IACUCCI, G., KUUTTI, K., AND RANTA, M. 2000. On the move with a magic thing: Role playing in concept design of mobile services and concepts. In *Proceedings of* the *DIS 2000 Conference* (New York).

IPERG. Integrated project on pervasive gaming. http.//www.pervasive-gaming.org.

KOIVISTO, E.M.I. AND ELADHARI, M. 2006. User evaluation of a pervasive game concept. In *Proceedings of the DIME-arts 2006 Conference* (Bangkok).

KOIVISTO, E.M.I. AND PALM, T. 2005. Iterative design of mobile games. In *Proceedings of Game Design and Technologies Worksho*p (Liverpool).

KOIVISTO, E.M.I. AND SUOMELA, R. *2007*. Using prototypes in early pervasive game development. In *Proceedings of the SIGGRAPH Sandbox Symposium* (San Diego, CA).

KOIVISTO, E.M.I. AND WENNINGER, C. 2005. Enhancing player experience in MMORPGs with mobile features. In *Proceedings of the Digital Games Research Conference* (Vancouver, B.C.).

KOSKINEN, K. AND SUOMELA, R. 2006. Rapid prototyping of context-aware games. In *Proceedings of the 2nd International Conference on Intelligent Environments* (IE06, Athens, July 5-6).

LANKOSKI, P., HELIÖ, S., NUMMELA, J., LAHTI, J., MÄYRÄ, F., AND ERMI, L. 2007. A case study in pervasive game design: The Songs of the North. In *Proceedings of the NordiCHI Conference* (Tampere, Finland).

LEE, Y.S., SMITH-JACKSON, T.L., NUSSBAUM, M.A., BHATKHANDE, Y., AND TOMIOKA, K. 2004. Use of product-interactive focus groups for requirements capture and usability assessment. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (New Orleans, LA, Sept.).

MANNINEN, T. 2000. Contextual virtual interaction as part of ubiquitous game design and development. *Personal and Ubiquitous Computing 6*, 390-409.

MCGONIGAL, J. 2006. This might be a game: Ubiquitous play and performance at the turn of the twenty-first century. PhD dissertation.

MONTOLA, M. 2005. Exploring the edge of the magic circle: Defining pervasive games. In *Proceedings of DAC 2005 Conference* (IT University of Copenhagen, Denmark).

NIELSEN, J. 1994. *Usability Engineering*. Academic Press, London.

ROYCE, W.W. 1970. Managing the development of large software systems: Concepts and techniques. In *Technical Papers of the Western Electronic Show and Convention* (WesCon, Los Angeles, CA, Aug. 25–28).

SIGNMAN, T. 2005. The siren song of the paper cutter: Tips and tricks from the trenches of paper prototyping. *Gamasutra* (Sept.). http://www.gamasutra.com/features/20050913/sigman_01.shtml.

STRÖMBERG, H., PIRTTILÄ, V., AND IKONEN, V. 2004. Interactive scenarios – building ubiquitous computing concepts in the spirit of participatory design. *Personal Ubiquitous Computing 8*, .200-207.

SUOMELA, R., KOSKINEN, K., AND HEIKKINEN, K. 2005. Rapid prototyping of location-based games with the multi-user publishing environment application platform. In *Proceedings of the IEE International Workshop on Intelligent Environments* (University of Essex, Colchester, June 28 – 29),143-151.

SUOMELA, R., RÄSÄNEN, E., KOIVISTO, A., AND MATTILA, J. 2004. Open-source game development with the multi-user publishing environment (MUPE) application platform. In *Proceedings of the Third International Conference on Entertainment Computing* (ICEC 2004, Eindhoven, The Netherlands), Lecture Notes in Computer Science, Springer, New York.

SNYDER, C. 2003. *Paper Prototyping – The Fast and Easy Way to Design and Refine User Interfaces.* Morgan Kaufman, San Francisco, CA.

THE AGILE ALLIANCE. http://www.agilealliance.com/.

WALTHER, B. K. 2005. Atomic actions – molecular experience. Theory of pervasive gaming. *ACM Computers in Entertainment 3*, 2.

WRIGHT, W. 2004. Triangulation: A schizophrenic approach to game design. Speech given at the *Game Developer's Conference 2004,* (San Jose, CA).

Received October 2007; revised November 2007; accepted April 2008