Boxed Pervasive Games: An Experience with User-Created Pervasive Games

Richard Wetzel[1], Annika Waern[2], Staffan Jonsson[2], Irma Lindt[1], Peter Ljungstrand[2], and Karl-Petter Åkesson[3]

[1] Fraunhofer Institute FIT
[2] Interactive Institute
[3] Swedish Inst. of Computer Science
richard.wetzel@fit.fraunhofer.de, annika@tii.se, irma.lindt@web.de, staffanj@tii.se, peterlju@tii.se, kalle@sics.se

Abstract. Pervasive games are rapidly maturing - from early research experiments with locative games we now start to see a range of commercial projects using locative and pervasive technology to create technology-supported pervasive games. In this paper we report on our experiences in transferring the successful involvement of players in computer games to 'modding' for pervasive games. We present the design process, the enabling tools and two sample games provided in boxes to end users. Finally we discuss how our findings inform the design of 'modding' tools for a pervasive game community of the future.

Keywords: User-centered design, Games and infotainment, Programming tools: Integrated environments, Pervasive computing, Pervasive games, End user programming.

1 Introduction

Pervasive games take the player out into the real world [13]. They are rapidly maturing – from early research experiments with locative games [2,3,8], we now start to see a range of commercial projects using locative and pervasive technology [1] to create technology-supported pervasive games. Designing and staging pervasive games requires however both high technical skills as well as in-depth design knowledge about what make pervasive games engaging and fun. Very few pervasive games can yet be created or staged by non-professionals.

Whereas computer games can be run just about anywhere (just insert the CD into a sufficiently powerful PC), pervasive games are typically created or configured for the particular location of the game. Many games require physical objects such as special markers or RFID tags to be placed in the gaming area; others can be staged by marking the interesting game locations on a map – places that the organizer has visited and decided it was suitable for the game. Pervasive games also tend to be technology-supported rather than fully implemented in technology. They may require game mastering, the creation of non-tech props or special-built technology, and trained actors participating in the game.

© Springer-Verlag Berlin Heidelberg 2009

In this paper we report on our experiences in creating tools for 'modding' 'modding' pervasive games. In the context of computer games, 'modding' ranges from small modifications to existing games to the creation of entirely new games on top of the game engine of an existing one. To enable 'modding' we created an authoring and game-mastering tool and tested it in various ways with users that could be interested in – and able to – stage pervasive games and create their own pervasive games. The ultimate objective for the project is to create a 'modding' community for pervasive games.

2 Related Work

There is a long-standing tradition of end-user involvement in the computer science field. The vision of end-user programming [14], empowering end users to create their own applications without having to master abstract programming languages, has yet to come true. However, the increased availability of tools enabling user-generated content, popularized in the form of typical Web 2.0 applications such as blogs, wikis, and photo and video sharing sites provide a large step in this direction. The latest developments, in the form of easily manageable and configurable web services such as mashups, go beyond the pure management of non-computational media towards creating your own programs.

However, user-created or user-modified games are already legion in the domain of video and computer games. The first person shooter genre is famous for its active modding communities [11] and there also exist a large set of mods for the most popular games in other genres such as *World of Warcraft*[1]) and *Warcraft III*[2]. Modding communities tend to arise around highly attractive games which motivate the community members to dedicate time to create improved and innovative versions of the original game. The content that is produced by the players ranges from small modifications, such as the customized appearance of game artifacts and characters for *The Sims* game [21], via new game worlds and levels such as new maps for *Warcraft III*, to completely new games such as Counter-Strike[11], and Desert Combat [15]. Some games (see again *Warcraft III*) are released together with tools for scene development and scripting that together constitute a rich development environment. It is important to note that while most end users can benefit from the modding community, there are typically only a small number of players who are actively creating the mods.

End users are often able to invent ways of using the game assets that go beyond the intentions of the original designers; the modding communities are the 'lead users' [18] of the gaming industry. User-created modifications to a game can make the game richer or more interesting to play.

2.1 Pervasive Games

Pervasive games are games that take place in the real world rather than on-screen [13]. They leave the predefined playground, be it the screen or the tennis court, to be

[1] See http://www.warcraft-mods.com/ Although the World of Warcraft team discourages the development and use of mods, they are in frequent use in practice.

[2] Blizzard maintains an official Warcraft III mod site at http://www.battle.net/mod/

playable anywhere and at any time. There exist pervasive game designs that use no technology at all, but mobile and pervasive technology opens up new possibilities for pervasive game designs. Previous projects have investigated mobile Augmented Reality [9], GPS and RFID readers as well as custom-built hardware [17]. Advanced pervasive game designs often rely on a mixture of automatic and game-mastered game play, as this provides good support to deal with a changing environment and player improvisations. A good example of this is the success of the Alternate Reality Game genre [12], games that largely rely on semi-automatic gameplay in online sites and where the hidden 'puppet masters' constantly supply the players with new material.

The cost profile of a pervasive game is often that of quite low development costs combined with fairly high costs for staging and running the games. Despite the commercial success of Alternate Reality Games, this has hampered the development of commercial pervasive games. If players were involved in the authoring, staging and game mastering of pervasive games this would increase the number of games staged, dramatically reduce the costs of such games, and contribute to the attractiveness and adoption of this new type of game. Ultimately, the quality and innovation in the games as such would rise with an active modding community.

A first step towards this is to make the staging of the game and the localization of game content a game activity in itself. The game *Hitchers* [7] uses this approach by letting the players create game content and place it in the environment. Although this approach certainly is possible, it does not allow for the creation of full games that are tailored to a particular place or a particular player group; the game itself is not modified. *GeoCaching* is a similar activity that has grown out of the more and more commonly available GPS technology; a pervasive folk movement that comes very close to already being a pervasive game.

In this paper, we try to go one step further. We are looking for a model that allows players to stage games, contribute to the richness of their content, but also to create their own games. The most generic approach to date is probably the *Mediascapes* tool [10] which allows end users to create and place media content in a geographic landscape, either through interacting with a map online or on location through a mobile device. However, the *Mediascapes* system provides little or no support for creating the complex logics that typically are necessary to make a more complex game.

3 Boxed Pervasive Games

A future working model for 'modding' of pervasive games would rely on two things. Firstly, it would require access to technology and tools that can be used for modding. It also requires that there is a modding *community*; a community that knows about the game genre and is inspired to modify games and develop their own games. A central question for the project became what kind of tools and technology that potentially could inspire a modding community to form.

In order to select an overall approach, we considered two specific issues: that people typically don't own the technology needed to play an advanced pervasive game, and since they haven't played them, people don't know what pervasive games are.

Pervasive games do often (although not always) require unusual technology. As previously discussed, previous projects had used Augmented Reality, GPS and RFID readers, all of them generic but not commonly available technology requiring special hardware[3]. On numerous occasions, we had also developed custom-built hardware to fit the setting and theme of a special game. The solution to this problem was to create 'technology boxes', where the hardware and the development tools were packaged together and streamlined to work together. We also wanted to make it possible for a technology-savory user to build game-specific devices and integrate them with the same development tools.

The second issue cannot be completely addressed by just tools and technology. Understanding pervasive games typically requires that you have played them enough to understand something about their core game mechanics and what makes them fun. But this is not enough. A 'modding' community is driven by its own ideas and creativity, so the participants must feel that they both want to, and can, make truly great games. The latter is only likely to happen when pervasive games already are fairly common in society or there exists at least one blockbuster pervasive game.

In order to partially address the second issue, we settled for an approach where tools, technology and games were packaged together. The idea was to *package pervasive games in retail boxes together with the hardware needed to run the games*. The approach was inspired by the Lego boxes: we wanted each box to contain a game of its own much like Lego boxes contain instructions to build Lego model, but we also wanted it to contain the tools needed to build other games with the same hardware. Finally, we wanted the boxes to work together so that you could build more complex games if you combined several boxes. This approach lead to a number of design decisions concerning the hardware, software and games that we choose to put into our boxes. Towards the end of the article we will come back to these to discuss whether they were the right ones.

- The core system needed to rely on fairly standard hardware, and use only low complexity special-built hardware that could be packaged with the game. This was in order to keep the price of a box within reason. When special-built hardware was used, it needed to be semi-generic to support a range of games rather than just a single game.
- All software must be either pre-installed on the hardware or very easy to install. As much as possible should be available as on-line services.
- We assumed that the most common reason to buy a box would be to set up and play a game with your friends. This made it important to include games that were easy to grasp and easy to install.
- The content of a Box must inspire its buyer to start to create his or her own games. This issue mainly concerned the game; that should inspire novel game ideas.
- A Box must contain tools that allow people to build and stage their own games.

3.1 The Box Development Process

We have developed two example boxes. The *Magic Lens box* can enhance the physical world with invisible three-dimensional objects, which the player can see and interact

[3] Mobile Augmented Reality is a software solution, but it requires a particular type of powerful portable computer to run.

with through a portable computer. The *Location box* instead enhances the physical world with sound and hidden pictures, and builds on GPS and RFID technology.

The two boxes should be seen as examples of a generic approach. A core requirement was that each would contain technology to support a range of games, at the same time as it would offer an interaction metaphor that could be easily grasped. Finally, we wanted each box to be immediately useful through the inclusion of ready-made game, but at the same time be versatile enough to let its users develop their own games.

The boxes were developed in collaboration with three focus groups, ranging from high school students, university students in game and media studies, and a professional team producing a commercial production. The Gamecreator system discussed below was also used in this commercial production. The focus groups provided feedback on all aspects of the boxes, ranging from the selection of technology, the selection of games, and the functionality offered by Gamecreator.

The focus groups provided very good feedback during the development of the boxes. Towards the end of the development period, the focus groups as well as some other groups also got a chance to use the boxes for a limited period, to stage the games as well as develop their own games. This experiment was however less successful, as the participants had too little time to actually develop anything with the tools. Only one group managed to create a game of their own, and none of the groups staged the preinstalled games with friends. We will have reason to come back to the trials in the section on 'lessons learned'.

The boxes have so far been used in five different game projects. Firstly, the boxed games the *Alchemists* and *Crash* were developed using Gamecreator. Gamecreator has also been used to implement parts of a commercial production "*The Truth About Marika*" [6]. Although this was a truly enlightening experience, it was done in close collaboration with the Box development team and for commercial purposes. In this paper, we focus on the experience of using the boxes in external projects relying on volunteer efforts, as this usage situation more closely resembles that of a future modding community.

3.2 The Magic Lens Box

The Magic Lens box uses Augmented Reality technology [16] to create pervasive gaming experiences. The box contains a ultra-mobile PCs (UMPCs) with pre-installed software, printed markers for pose tracking and a user manual as seen in Figure 1. The software for the *Magic Lens box* is based on the MORGAN AR/VR Framework [16] and uses the ARToolkitPlus library [19] for marker based tracking. The system tracks specially designed black and white paper markers by calculating their exact position and orientation from a video stream. 3D models are overlaid on the video, creating the illusion of them really being where the marker is. The approach is particularly useful on a hand-held device with a camera mounted on the back, creating the illusion of a "magic lens" [5]. In contrast to other tracking methods this method is very easy to set up. All the user has to do is print out the markers or use the ones that come with the box and place them in the physical game world.

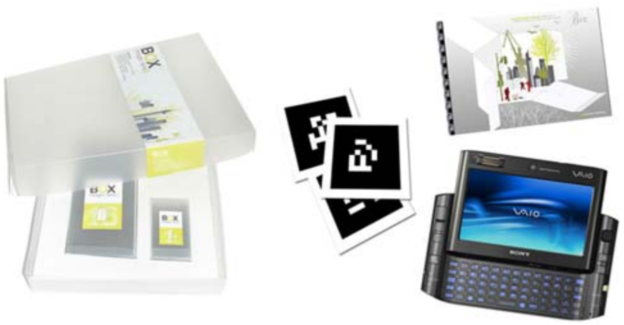

Fig. 1. The *Magic Lens box* (left) and its content (right)

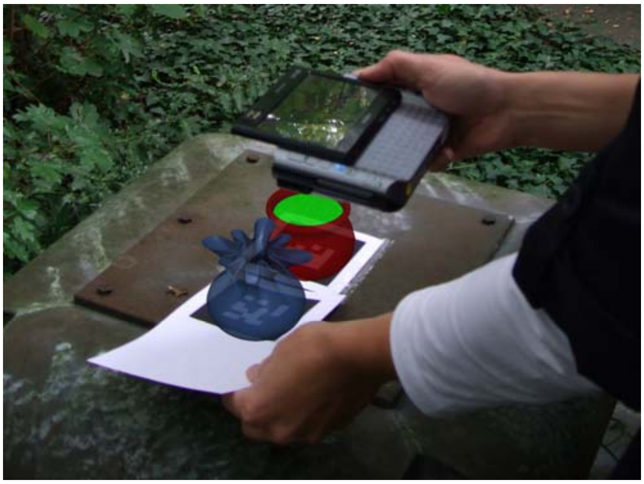

Fig. 2. Augmented view from the *Magic Lens box* game *The Alchemists*

The sample game that comes with the box is called *The Alchemists*, a simple treasure hunt game (see [20]). The game master hides a variety of markers in the playing area. These markers are associated with 3D models of potentially powerful alchemistic ingredients, which need to be found by the players. The players can then pick them up with their magical bags (also represented by a marker) and brew them together with other ingredients in their alchemistic cauldron (again, a marker). If the ingredients fit well together, the players are able to brew different types of potions and are awarded points depending on the quality of the mixture (see Figure 2).

Modifying the existing game and creating new games is done in the *Gamecreator* system discussed below. For implementing game mechanics, the *Magic Lens box* software supports two different interaction models: looking with the webcam at a single marker and looking with the webcam at two markers at the same time.

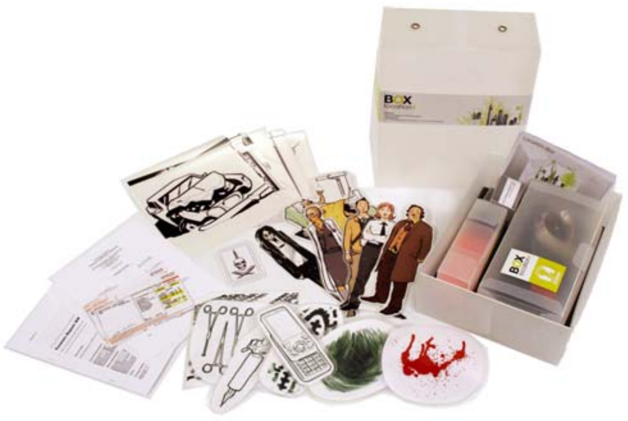

Fig. 3. The Location box and its content

3.3 The Location Box

The *Location Box* uses Internet-enabled mobile phones together with a GPS or a RFID reader connected via Bluetooth (Figure 3). The GPS units are ordinary off-the-shelf GPS receivers with Bluetooth, whereas the RFID readers are specifically built hardware that come in the shape of a wearable glove. The GPS allows to position game content to specific locations or areas while the RFID tags can either be placed at specific places or attached or hidden in artifacts. This provides flexible positioning of content with regards both to accuracy and scale.

The mobile phones primarily work as relay stations, gathering data from the GPS and RFID readers and sending it forth to a game server over Internet. They are also used as recipients of streamed content from the game server. The phone client is implemented in J2ME and enables images, texts, sound, and video files to be streamed to the phone and automatically displayed on the device. It is currently only available on one phone model, the Sony Ericsson K800.

The sample game that comes with the box is called *Crash* and is a simple detective clue hunt that can be set up and played as a party game. It uses only the RFID reader but can be extended by the organizer to also use the GPS. The players are invited into the game as junior police inspectors. Once they reach the place where the accident took place they start finding clues that lead them on into the story.

Game locations, characters and items are represented by printed images with RFID tags attached to them. The players use the RFID glove to interact with these images. This enables them to go to a place (and show where they are), examine evidence, and interact with game characters. The game responds by sending sound and image files to the player's phones. As the players progress through the game, the story unfolds further and gets more complex.

3.4 Gamecreator

A central decision in the project was to develop a joint tool for both boxes. This way, a person who had used one of the boxes would immediately recognize the development method for the next box. We also wanted the tool to support games that used hardware and software from both boxes in the same game.

The *Gamecreator* system was constructed to fill these requirements. It is directed towards the amateur developer of pervasive games, rather than towards the professional programmer or game designer. It still requires some level of programming or scripting proficiency. A target user could for example be a person who has some interest in pervasive games, and previous experience of Web 2.0 services (e.g. has set up and managed his or her own blog).

Gamecreator is developed in Ruby on Rails and runs as an online service. It supports both authoring of games, and the set up and game-mastering of existing games. Each box owner has an own account with Gamecreator, where he or she has access to a development area which contains a preinstalled game, and two empty game slots where the user can develop new games through an online interface. The pre-installed game is available with a full game script, so that the user can modify it at wish.

Authoring a new game

In order to make Gamecreator run with several different technologies and devices, we decided to make the core functionalities of Gamecreator device-independent. A central design decision for *Gamecreator* was to separate all information about what the technology could use as input (e.g. that an RFID tag had been read) or deliver as output (e.g. the 3D model that should be displayed) from its *meaning* in the game (e.g. that a player has entered into a game area, or found a game object).

The game object classes are not fully generic. This is done to support game developers to think about pervasive games, as the classes primarily represent aspects of the real-world. Gamecreator allows three types of objects: persons, places and items, primarily intended to represent real-world persons, places, and things. Each type of object can be identified by several different technologies. A location can e.g. be identified by a GPS area, or an RFID tag, or a Magic Lens marker, or even all of them. A person can be identified by a RFID tag and an item by an RFID tag of a Magic Lens marker. Game objects have game-specific properties that can be changed automatically (by scripted rules) or manually by a game master. Gamecreator contains an online system for scripting rules.

The Gamecreator objects will typically, but not necessarily, map to real-world locations, people and objects. The people category is often used to represent the different types of players (e.g. the team leader versus the team participants) or provide player roles depending on which technology they carry. It will sometimes be used to represent non-player characters (e.g. a paper doll, as in *Crash*).

The clear separation between technology-related input and output and in-game objects has several desirable effects. Gamecreator can fairly easily be integrated with new technology platforms. Gamecreator comes with a back-end API which can be used to query the current game state and trigger events from any external system. Also, the game can use hybrid methods to recognize in-game events. For example, the *Location box* allows indoor events to be triggered by reading an RFID tag and outdoor events by entering a GPS area. Finally, the in-game objects need not be recognizable by technology at all. An in-game location can be a virtual location (e.g. a website) or even a location that only exists as a game concept but never actually is visited by the players. Game objects that are not directly related to technology can still be manually managed by the game masters through changing their property values.

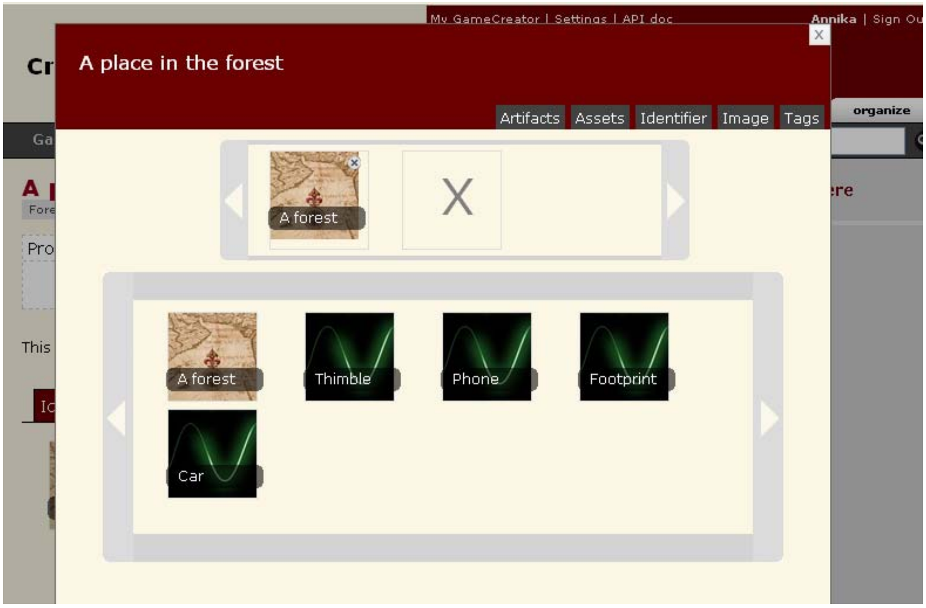

Fig. 4. Sample screenshot from Gamecreator. In the screenshot, a GPS location named 'A forest' has been added as the location of an in-game location named 'A place in the forest'.

Game Orchestration

More complex pervasive games often benefit from being game-mastered than fully automatic. As discussed above, Gamecreator also supports game mastering which was used extensively in the *Interference* game discussed below. There are two main functions in Gamecreator that support game orchestration. One is that Gamecreator maintains a log of all incoming events, and provides the game manager with an overview of the event history. Positioned events are plotted on a map using the Google map interface. Gamecreator also supports *manual* property changes; the game manager can select a subset of the game objects and change any of their properties. Finally, as the rules of a game are scripted in the Gamecreator system, even the rules can be changed while the game is running. This is seldom used in small games but can be very useful in more long-term games such as Alternate Reality Games.

4 Modding and Staging a Given Game: Interference

Our first example focusses on the restaging of a given game. The game *Interference* was developed within the project after *Crash* and the *Alchemists*, to show that the boxes can be combined to create more complex games. The game was restaged by a group of game design and media students at Mediadesign Hochschule Düsseldorf.

Fig. 5. Photo from the Interference stagings in Stockholm, January 2008

4.1 Gameplay

Interference [4] is a very different game compared to the two boxed games. Where *Crash* and *the Alchemists* are small and automatic games designed to be easy to set up and run, *Interference* was designed in collaboration with a professional artist as an artistic experience. It requires game mastering and a complicated staging consisting both of a technical setup and infrastructure, renting locales, and recruiting and training actors. The game had been staged four times in Kista, Stockholm before it was staged in Düsseldorf.

Interference is a game for six to eight players who play together as a single group throughout the whole game. The game is a pervasive mystery game, a kind of a 'treasure hunt' with a story line. The game starts out as in a fairly gamistic manner (the focus is on game mechanics rather than the narrative), as the players are given simple roles as electric engineers who are recruited by a company to fix their broken network. The background story unfolds through the players' meeting and communicating with the central characters, played by actors. Eventually, the players find the means to solve their original technical problem – but only at the expense of the death of one of the central characters. They are faced with a difficult decision and the game ends by letting the players collectively decide what to do.

Interference makes use of the Magic Lens device from the Magic Lens box as well as of GPS positioning from the location box. RFID tags and the RFID reader glove are not used in this game. The Magic Lens device is used to trace the network and find out spots where it is broken. The network is represented by 3D models visible only at certain locations (at the markers) and the players have to draw lines on a map to connect the nodes to each other. The GPS from the Location box is used to seek out magical memories in the landscape, located at emotionally interesting locations. The memories are sound and video files that are streamed to four different mobile phones. Finally, a special-built ‘magic’ doll was built for this game, which is able to recognize the tones played on a special bone flute. The magic doll was built from scratch, using audio circuits tuned to recognize the flute tones, and a Bluetooth circuit to connect the doll to the phone. Just as with the GPS and the RFID glove contained in the location box, the doll connects to a mobile phone that works as a relay station for communicating to a server. The flute is a real bone flute with no technology augmentation, made magic through the fact that the doll reacts to its tones. The players receive the flute only towards the end of the game, working its magic to mend the broken network.

Before the game can run in a new city, the organizers have to create a map of the fictional ‘broken network’ overlaying the new game area, and chose the 3D models to fit with the network intersections and lines. The network map should ideally fit the theme of the game, so that the memory traces picked up by GPS are placed in interesting and emotionally powerful locations. In particular, the last scene requires a dramatic location. Finally, there is the issue of language: if the game is replayed in another country the memories may need translation. In Düsseldorf, the sounds were re-recorded in German by the actors.

The staging team must also prepare several non-technological aspects of the game. As *Interference* is a technology-supported game, the players’ experience relies largely on non-technological activities such as plotting the network lines on a paper map, visiting interesting locations, meeting with actors, and playing the flute (see Figure 5). The staging team has to rent locales, draw and print maps[4] and visual markers, and training the actors. The last issue is perhaps the biggest challenge, as a successful staging depends on gradually change the attitude of the gamers from “happy questing” in the beginning, to deep involvement in a difficult choice towards the end. The three actors are central in this process; their involvement as game characters must engage the players sufficiently to make them dig deep into the story line. An additional complication is that the actors often will relay information between the game masters and the players. The task of, at the same time, acting out a role in an improvised theatre with the players and relaying messages from the game-masters is far from simple.

While the game is running, *Gamecreator* is used as an orchestration tool, and offers several ways to control the game progress. In particular, the game master can trace the players’ location and help them back on track in case they get lost, but they can also directly manipulate the game state to speed up or slow down the game progress. To enable the students to restage the game, two of the students participated in one of the Stockholm stagings, where they had a chance to play the game as well as participate in the briefing of actors and the game-mastering. We also ensured that there was at least one person from the development team onsite during stagings; although these people tried to be as little involved as possible this was necessary both to train the full student group and to help out with technical problems should any occur.

[4] This is a larger issue than it might seem. The physical map was one of the devices that actually broke when staging *Interference* in Stockholm - it survived much, but not a snowstorm.

4.2 Observations

The Düsseldorf student team was very active and enthusiastic about staging Interference. Two of them participated in the Stockholm stagings, to get an idea of the particular game as well as a general understanding of technology-supported pervasive games. Based on the experiences from the Stockholm stagings, they started out by doing some changes to the game. The first one was that they created a much more interconnected virtual network overlay for their selected game area. This change was done in order to make the first game task (scouting the network) more interesting to the players and minimize the risk of getting lost. They also modified the use of the Magic Lens markers, placing them on the flat on the ground instead for on poster areas and walls. This made it easier to trace the direction of lines for the players, again contributing to making the initial part of the game more fun.

The group also spent a lot of effort on scouting the area and selecting interesting places for the players to visit. They were particularly happy about the location for the final scene. In Stockholm, this scene had been placed in an office in a high tower overlooking Kista. In Düsseldorf, the student group managed to get access to the *rooftop* of the highest house in the vicinity and staged the final dramatic scene there.

Learning to use Gamecreator as an orchestration tool took some time. However, once the student team learned to use it, they appreciated its versatility:

> "I slowly start to understand Gamecreator. You can really do a lot with it – if you understand which is which." (Student comment)

Interference was staged on three occasions in Düsseldorf with a total of 22 players. The stagings in Düsseldorf received positive feedback from the majority of the players. These had no previous experience in pervasive games, and were very excited about the real world involvement in the game.

> "The best moment was when we met Matilda for the very first time. She was standing on a dark bridge in the dimly lit park and seemed to be watching us. I was not sure if she belonged to the game and it took us a while to build up the courage to talk to her."
> (Player comment, from postgame interviews)

There were some technical malfunctions during the stagings with GPS breaking down and loss of Internet connection for the cell phones. Ironically this was probably caused by some real life interference in downtown Düsseldorf. This however showed one of the strong points of Gamecreator. As it allows for both automatic and manual control of a game, the students were able to trigger rules manually when the some of the automatic functions did not work. During the restaging process another strong point of Gamecreator emerged: The separation between the actual game content and the game logic proved to be extremely helpful. Exchanging the 3d models, adapting the game to the new location and setting was possible without too much of an effort. Additionally, the ability to trace where the players were moving in real-time as part of the orchestration interface was very valuable to the game masters. This way they were always aware of the players position which made it much easier to react and assure a smooth flow of the game (e.g. when the players were spending too much time in one space discussion, the students would have one of the actors call the players to put more pressure on them).

Afterwards we were told by the organizing student group that both preparing and staging the game was a truly great experience for them. In the end, their teachers also complained that the students were spending too much time preparing *Interference* - skipping class way too often. While this on the one hand shows that the process of staging a pervasive game is still very demanding, the fact that the students were highly motivated throughout the process is very promising. Staging pervasive games can be a lot of fun, perhaps in particular when the game is demanding and the staging itself requires creative engagement.

5 Creating a New Game: *The Treasure*

The game *The Treasure* was created by three B-IT Bonn students over the course of one semester. This game was built using the Magic Lens box. This was a supervised student project, where the students were first introduced to the concept of pervasive games by trying out and experimenting with the sample game *The Alchemists*. The students were then asked to come up with a game design by themselves.

The students created a variety of ideas before their choice became *The Treasure* which they then developed over a couple of weeks and finally staged in March 2008. The students were given free hands with the development, but could always contact the Box development team in case they ran into technical problems.

5.1 Gameplay

The Treasure was inspired by classical adventure games where the main character has to travel to different locations, solve riddles and collect and combine objects with each other. The students settled for developing a game that was tied to a particular place, the castle Birlinghoven and its surrounding gardens where the FIT research group is located. The choice of location matched the team of the game, as the castle Birlinghoven is a twentieth century ‘remake’ castle that has somewhat of a ‘fairytale’ ambience.

According to the game storyline, a long dead king has hidden away his treasure to protect it from greedy, undeserving people. The king has set up several challenges to test the mind and heart of the brave who venture to retrieve this treasure. The challenges are associated to statues and images found in the castle and the garden: for example in one case the players have to find meat for an eagle and his starving offspring, while in another they have to solve a riddle given to them by an old monk.

The students combined the real world and the virtual *Magic* Lens augmentation in a highly engaging manner throughout the game. Instead of having the players interact only with virtual characters, they made use of the real statues and paintings in and outside the castle. These would come “alive” when being inspected with the UMPC and start to talk to the players. Else, the interaction model was the same as in *The Alchemists*: the players could interact with a marker through viewing it through the UMPC, and pick up and drop items through placing two markers close to each other and inspect them together through the UMPC.

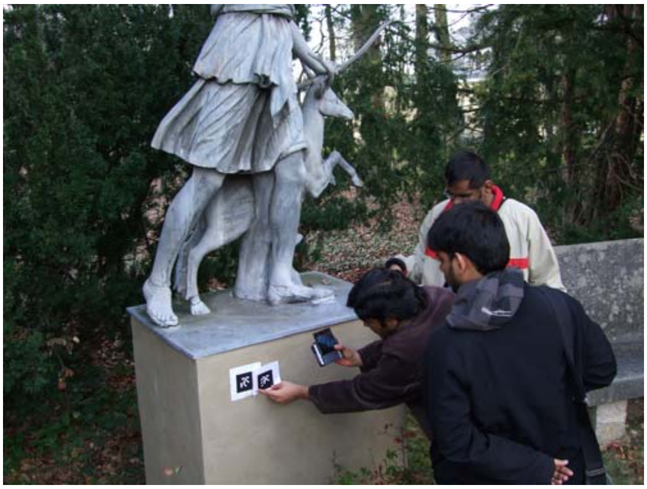

Fig. 6. Three players of *The Treasure* bringing a requested game artefact to the stone hunter

To solve the riddles, the players need to take a visual marker and bring it to the non-player character from who the player had received the task (see Figure 6). If the marker and its content did not solve the task, players would get some feedback and some further hints. The game ended when the player found a physical treasure map and could locate the treasure – which was a box of sweets.

5.2 Observations

Both boxes have an initially steep learning curve. Part of the problem is that the Gamecreator interface is far from self-explanatory (after all, it is a first prototype), but users also have a problem in understanding how to set up the boxed games. However, in this project the students rapidly overcame these issues and the process was smoother than in the *Interference* stagings, showing that Gamecreator is somewhat better at supporting authoring than orchestration. After only a short while the students had grasped the concept behind the *Magic Lens* box and *Gamecreator* and were already able to test small parts of their game. The game could then be developed and tested iteratively, contributing to the successful final design.

> "I personally liked this way to create the game, mostly because it was really feasible to create an Augmented Reality game at this short time period we had. If we would have to develop our own functionality it would not be possible at all, but with Gamecreator and the Magic Lens box we could just focus on the game design." (comment by another student developer)

The students ran into some problems with the built-in limitations of the box. The problems were a combination of the limited interaction model that the box implements and some restrictions in the scripting language implemented in Gamecreator.

"For the more of less complex games or games which have some new interaction concepts using Game Creator can be problematic because some functions can be just unavailable in it. This happened also during our game design and we were forced to use some "hacks" or change a game design in order to use Game Creator. I think that even after including more and more functionality into Game Creator such a situation ca accrue. One solution for such a problem is to allow user by some means (programming Game Creator core or something else) to extend Game Creator functionality according to their needs."

An interesting observation is that the game *The Treasure* manages to deliver a game experience that is quite different from *The Alchemists*, even though both games use the same way to model and specify a game, the same gaming hardware, and the same basic interaction techniques. The treasure game uses sound files to tell the background story and to guide the player through the game. In *The Alchemist* the background story is rather brief and told in a text document as part of the setup. Secondly, the treasure game suggests different walking paths. In *The Alchemist* players pick up ingredients and have to go back to their cauldron every time, but in *The Treasure* players need to bring game artifacts to different non-player characters, which makes the movement through the landscape more interesting. Finally, in *The Treasure* the game was closely adapted to the physical gaming area. This makes *The Treasure* much more of a true "mixed reality" experience. Apparently, the box and Gamecreator manage to create a good balance, providing guidance through its restrictions in interaction and scripting model but at the same time providing some degree of freedom that spurs the designers' creativity.

The play-tests with *The Treasure* show that the game was engaging also to players. The student team was happy with the outcome.

"During the playing session we were surprised how the players interacted with a new technique as if they had known it for a long time. What we were afraid of didn't appear, because the players were very enthusiastic and tried to hurry and run, even though we didn't mention something like having a deadline or time limit." (comment by one of the students developing the game)

6 Analysis

We now turn back to an analysis of our box idea. Could our boxes actually work as tools for a modding community? To answer this question, we reflect on the original design decisions to give a direction for future work in the area.

6.1 Designing Pervasive Game Boxes

Our first assumption was that the hardware should be cheap and off the shelf, or if not, at least semi-generic. This assumption was supported by the focus group feedback, which was a bit skeptic to the UltraMobile PC due to its price (to the extent that they were afraid to borrow and use it).

We also assumed that software should be pre-installed or easy to install, and that as much as possible should be available as on-line services, including the tools to create your own games. This proved to be very important and only partially fulfilled by the boxes. Gamecreator comes close to realizing this through its online scripting system, but its ease of use is compromised by its generality and a slightly obscure interface. The boxes have a steep initial learning curve: it is not easy to understand even how to set up and run the preinstalled games. To some extent this is due to quirks in the (prototype) interface and the setup procedures, but more it has to do with that the general concept of pervasive games is novel to the users. However, the experiences with *Interference* and *The Treasure* show that once a user has grasped the functionality and purpose of the system, they start to appreciate the versatile support offered by the system. The overall system seems to have created a good balance, providing guidance through its restrictions while at the same time providing enough freedom to spur the designers' creativity. Gamecreator is currently under re-development with an increased focus on configurability; something that also was desired by the Treasure developers. The new setup will allow games to be configured in a more generic way, and then scripted in a way that uses in-game concepts rather than generic concepts.

One of our assumptions turned out to be false: we assumed that the consumer would buy a box to play games with their friends, and that the games for this reason should be easy to grasp and easy to install. However, the boxed games were perceived as too simple to be interesting to stage by our focus groups, whereas the rather complex *Interference* game inspired intense engagement by the organizers and was successfully staged. We can expect the same to be true for a true 'modding' community for pervasive games: to inspire modding, the original game needs to be a rich, complex and inspiring game.

7 Conclusions

The most important lesson learned from this project is that the pervasive game boxes are not really targeted for a consumer market. Our focus on supplying small, easily staged, and easily understood games with the boxes was not successful, whereas providing a rich and complex game that required large investments to stage was. This is in line with the experiences from computer games: modding communities do not consist of ordinary consumers; they are formed out of a small percentage of the full player community for a game.

To create a modding community, the boxes need to be reshaped to support a truly attractive and rich game. One possible approach is to include a pervasive variant of a table top role playing game with a box. Such games are based on a very rich story space and rely extensively on game mastering. Game mastering such a game is a creative and skilful task which might attract 'modders'. The boxes could also be turned into a game themselves. The ideal game would be a continuously running massively multiplayer game service that requires special hardware. Players would buy a box to allow them to enter into the game, and they could be responsible for managing their own setup within the game. The Gamecreator system could then support players in creating modifications to their game clients, but also to build their own games.

With the ongoing advancement in technology, devices suitable for being used in pervasive games are already becoming more powerful and standard. The new iPhone3G by Apple for example already offers a decent sized screen and built-in GPS. This will make it easier in the future to package a box that makes full use of the hardware the potential players already have at their disposal, thus making the boxes more accessible and cheaper.

Acknowledgments

We thank our colleagues at the Interactive Institute, the Collaborative Virtual and Augmented Environments Department at Fraunhofer FIT and the Swedish Institute of Computer Science for their comments and contributions. We further wish to thank our project partners of the IPerG project for their ideas, cooperation, and support. IPerG (FP6-2003-IST-3-004457) was partially funded by the European Commission as part of the 6th Framework.IPerG. Finally we like to thank our focus groups for their insight, the students from the Mediadesign Hochschule Düsseldorf for doing a great job at restaging Interference and the student group from B-IT Bonn for developing The Treasure.

References

1. Ballagas, R.A., Kratz, S.G., Borchers, J., Yu, E., Walz, S.P., Fuhr, C., Tann, M., Hovestadt, L.: RExplorer: A mobile, pervasive spell-casting game for tourists. In: CHI 2007 extended abstracts on Human factors in computing systems, San Jose, CA, USA (2007)
2. Barkhuus, I., Chalmers, M., Tennent, P., Hall, M., Bell, M., Sherwood, S., Brown, B.: Picking Pockets on the Lawn: The Development of Tactics and Strategies in a Mobile Game. In: Beigl, M., Intille, S.S., Rekimoto, J., Tokuda, H. (eds.) UbiComp 2005. LNCS, vol. 3660, pp. 358–374. Springer, Heidelberg (2005)
3. Benford, S., Crabtree, A., Flintham, M., Drozd, A., Anastasi, A., Paxton, M., Tandavanitj, N., Adams, M., Row-Farr, J.: Can you see me now? ACM TOCHI 13(1), 100–133 (2006)
4. Bichard, J.P., Waern, A.: Pervasive Play, Immersion and Story: Designing Interference. In: Proceedings of DIMEA2008, Athens, Greece (2008)
5. Bier, E., et al.: Toolglass and Magic Lenses: The See-Through Interface. In: Proc. ACM Conf. Computer Graphics and Interactive Techniques (Proc. Siggraph), pp. 73–80. ACM Press, New York (1993)
6. Denward, M.: Broadcast Culture Meets Role-Playing Culture: Consequences for audience participation in a cross-media production. In: Proceedings of International Association for Media & Communication Research, IAMCR, Stockholm, Sweden (July 2008)
7. Drozd, A., et al.: Hitchers: Designing for Cellular Positioning. In: Dourish, P., Friday, A. (eds.) UbiComp 2006. LNCS, vol. 4206, pp. 279–296. Springer, Heidelberg (2006)
8. Falk, J., Ljungstrand, P., Björk, S., Hansson, R.: Pirates: proximity-triggered interaction in a multi-player game. In: Proc. ACM CHI, pp. 119–120 (2001)
9. Fischer, J.E., Lindt, I., Stenros, J.: Evaluation of Crossmedia Gaming Experinces in Epidemic Menace. In: Magerkurth, C., et al. (eds.) Proceedings of the 4th International Symposium on Pervasive Gaming Applications PerGames, Salzburg, Austria (2007)
10. Hull, R., Clayton, B., Melamed, T.: Rapid Authoring of Mediascapes. In: Davies, N., Mynatt, E.D., Siio, I. (eds.) UbiComp 2004. LNCS, vol. 3205, pp. 125–142. Springer, Heidelberg (2004)

11. Kücklich, J.: Precarious Playbour: Modders and the Digital Games Industry. Fibreculture Journal, ISSN: 1449 – 1443 Issue 5 Australia (2005)
12. McGonigal, J.: This Is Not a Game: Immersive Aesthetics and Collective Play. In: Proceedings of Digital Arts & Culture (2003)
13. Montola, M.: Exploring the Edge of the Magic Circle. Defining Pervasive Games. In: Proc. Of Digital Experience: Design, Aesthetics, Practice conference, Copenhagen (2005)
14. Nardi, B.: A small matter of programming. MIT Press, Cambridge (1993)
15. Nieborg, D.B.: Am I Mod or Not? - an Analysis of First Person Shooter Modification Culture. In: Creative Gamers Seminar - Exploring Participatory Culture in Gaming. Hypermedia Laboratory (University of Tampere) (2005)
16. Ohlenburg, J., Broll, W., Braun, A.: MORGAN: A Framework for Realizing Interactive Real-Time AR and VR applications. In: SEARIS, Workshop on Software Engineering and Architecture for Realtime Interactive Systems at IEEE VR (2008)
17. Stenros, J., Montola, M., Waern, A., Jonsson, S.: Play it for Real: Sustained Seamless Life/Game Merger in Momentum. In: Proceedings of the DIGRA conference (2007)
18. Von Hippel, E.: The Sources of Innovation. Oxford University Press, Oxford (1998)
19. Wagner, D., Schmalstieg, D.: ARToolKitPlus for Pose Tracking on Mobile Devices. In: Proceedings of 12th Computer Vision Winter Workshop CVWW 2007 (2007)
20. Wetzel, R., Lindt, I., Waern, A., Jonsson, S.: The Magic Lens Box: Simplifying the Development of Mixed Reality. In: Proceedings of DIMEA, Athens, Greece (2008)
21. Wright, W.: Triangulation: A schizophrenic approach to game design. In: Proceedings of the Game Developers Conference, San Jose, CA (2004)