ABSTRACT
Emerging technologies for positioning and context recognition together with advances in mobile computing and wireless communication technology enable the creation of new styles of games that are not only suited to mobile use but also exploit the dynamic context of the user. While spatial aspects and the task of “navigation” have been present in computer games from the beginning this has mostly been in the form of simulation on a fixed display. This paper presents an exploratory study on the use of real world geographic environments for and within games.

Categories and Subject Descriptors
I.3.m [Computer Graphics]: miscallenous

General Terms
Design, Human Factors.

Keywords
Mobile gaming, Adaptive.

1. INTRODUCTION
Work on mobile gaming has initially focused on techniques to deal with the inherent constraints of mobile devices, namely small displays and limited interaction modalities. More recently mobile gaming has started to explore the physical movement of players for gaming, largely driven by technology. An example for this is “geocaching”, in which GPS receivers are used to find “geocaches”: small waterproof containers containing a logbook who’s positions are published on websites [4]. Commercial experiments with mobile games that exploit positioning technology include SingTel’s “Gunslingers”, [8] and “Can you see me now” by Blast Theory [2]. While these games limit the use of the player’s real-world context to positioning, the potential of the available technology is much larger: Large real-world geographic environments could be used as “game areas” (as large as the whole world in the case of geocaching) where the real-world is incorporated into the game content. In addition to this, user actions besides physical locomotion can be captured and used as interactions within games. An early example for the incorporation of real-world objects into the game content is the ARQuake system [7], in which the popular first-person shooter can be played in outdoor environments. Using an AR-system consisting of a head-mounted displays and a backpack with a laptop and GPS positioning system the game content is integrated into the real surroundings of the user.

These developments show that positioning and context sensing enable the exploitation of spatial aspects and geographic environments in existing game concepts. In order to explore the resulting design space we have conducted an initial experiment with a location aware mobile game that aims to make meaningful use of real world features. The following sections first discuss the influence of context in mobile games and the resulting need for adaptation at run-time. Then the MobEE (Mobile Entertainment Engine) game engine is introduced that provides the technological basis for run-time adaptation. The following section introduces “Forgotten Valley”, an exemplary MR game that was implemented with MobEE. Finally, experiences gathered during the development are discussed and implications for future developments are derived.

2. CONTEXT FOR MOBILE MIXED REALITY GAMES
The mobile context of use differs significantly from the traditional use of computer games not only in the available devices but also in the characteristics of the environment and most importantly in the user’s activities. Since the user is on the move the context can change continuously. If context is to be exploited the game design has to ensure that the game remains not only playable but also enjoyable within a changing and only partly controllable environment. We have decided to use a mixed-reality approach to resolve this design challenge. Mixed-reality, as defined by Milgram [6], covers the complete spectrum between Virtual Environments (VE) and Augmented Reality (AR), where interactive computer graphics are integrated into real-world environments. In our system the game presentation can be adapted between a conventional (completely virtual) and augmented reality style (using parts of the environment as game content).

This way the user can complete a game under all circumstances. As figure 1 shows we consider three different aspects of context, namely the technological constraints of the user’s device, the user’s physical context and the user’s activities to which the user interface, content and presentation are adapted at run-time.

Adaptation allows addressing the limited and varying display and interaction capabilities of mobile devices [5]. In our system the presentation style is chosen from a number of formats including
The activities of the user also require adaptation of the game content. Typically, mobile applications are used for short interactions on mobile devices is constraint by the available input modalities. In addition to adapting the user interface to those modalities supported by the user’s device our system also exploits the gathered context information, namely the user’s real world locomotion and actions as input to the game itself, turning the world around the user into his “game board”.

A game-engine for mobile entertainment computing must address various requirements to be useful for application developers. In addition to support for the creation of the narrative application content the game-engine should also support the run-time presentation of this content on a wide variety of mobile hardware platforms, enabling adaptation to different devices.

3.1 Device-Independent Story Structures

To enable the use of an entertainment application across a variety of hardware platforms the story structure that captures the narrative content must be kept in a device independent format. By separating the story structure from its representation, users can be presented with the same storyline and information regardless of the device that they are using. Only the presentation format/media needs to be adapted to the capabilities of the interaction device.

The device-independent representation of the story structure in MobEE is implemented by finite automata controlled by a variable pool. Each hot-spot accessible by the user changes variables and thereby the states of the automata. Each hot-spot and automaton-state-change is linked to narrative information such as text, graphics, sound or animation which is displayed to the user upon activation.

3.2 Adaptable Presentations: Text, Graphics and AR

Because there are some many different types of displays on mobile devices the game engine has to consider all of these differences and take them into account by offering various types of presentations.

A simple output of text can be realized on almost every mobile device regardless of display and processing speed, but for many entertainment applications a text only presentation is not sufficient anymore [3]. However, because some mobile devices do not provide other means of output text representation is supported by MobEE.

Nevertheless, 2D Graphics offer a more attractive representation and is the more suitable presentation modality in many entertainment computing scenarios. In its current implementation MobEE supports sprites, scrolling, animations and video (limited) for the 2D representation of entertainment applications.

3D animation as the state-of-the-art representation for entertainment computing is becoming more interesting for mobile computing as acceleration hardware finds its way into mobile devices. Nevertheless, the complexity of 3D scenes that can be rendered in real-time on most mobile devices remains quite limited compared to current game consoles or PCs.

Another possible approach is to use the real world surroundings of the user as a high fidelity background, an approach that we are investigating in the mixed-reality version of “Forgotten Valley”.

In this version specific real-world locations are linked to game content. If the user is at such a location he can use the camera of his mobile device to capture an image of his surroundings that is then augmented with the graphical game content. While GPS data is sufficiently accurate to determine if the user is approaching a “game-location” and inform him accordingly, it does not provide the required accuracy for augmenting images of the user’s surroundings spatially correct with game information. Therefore, the current prototype uses ARToolKit [1], a computer-vision fiducial based tracking system for AR-applications for the actual augmentation. Due to performance reasons we have implemented a “snap-shot” AR approach: The user takes a single picture with the PDA’s camera, which is then analyzed and taken as a static background for rendering. Since only the augmentation graphics have to be rendered the impact of the hardware constraints are reduced since the user has high-fidelity context information from his real-surroundings. This way interactive framerate (>10 fps) with appealing graphics can be realized on most Pocket PC PDAs. We have found that the static image is usually sufficient to establish the link between the game content and the environment, although real-time 3D tracking and augmentation remain a desirable goal.
3.3 Context Refresh
As mentioned in the introduction, developers of mobile entertainment applications have to take into account different levels of attention of the user. Because there may be longer periods between the uses of a mobile entertainment application it may prove necessary to provide users with a context refresh of his recent activities in the application, especially if the game content had been adapted to context that is no longer available.

Such a context refresh is realized in MobEE by keeping track of all user actions. Since a complete history of all his actions every time he starts the program would be boring or even annoying the context refresh has to select significant actions and display these. Unimportant detail and older actions that are no longer relevant are left out of the context refresh, focusing only on important and more recent details.

3.4 Mobile Interaction Techniques
In the mobile context of use it is also necessary to adapt the interaction modalities to those supported by the device. We explore the use of non-standard input modalities in order to exploit context information (e.g. location) and physical actions of the user as means to control the game. Position sensing through a GPS receiver can be used to link the game-content to the user’s physical context. Thus the interaction task of “navigation” can be replaced by physical motion of the user in a real-world environment, making explicit interaction unnecessary. Section 4.1 describes how position sensing is exploited in the current prototype.

We have implemented camera-based interaction techniques for the interaction tasks select and quantify, based on the popular taxonomy of Foley et al. [3]. The select task refers to symbolic selection from a set of options. Our first approach is based on the tangible computing paradigm and can be used if the set of options can be represented by associated physical objects. Then selection can be effected simply by placing the camera so that the object is in the camera’s field of view. Section 4.1 describes how the MR version of the adventure game “Forgotten Valley” uses this technique (based on ARToolKit) to enable users to solve a riddle through physical actions in the real world.

3.5 Game Authoring & Establishing Spatial References
If a game should be playable in the virtual as well as in the real world (with augmentations) one would typically have to create two different games and switch between them. But this would also double the effort needed for the game development. As shown before, the MobEE engine allows keeping the story structure in a single point, “just” switching the used visualisation- and interaction-techniques. Having this in mind the creation of a mixed reality game with the MobEE–Engine starts with creating a normal game with a virtual game world. When it comes to the transition from the virtual to the mixed reality game, the already created content has to be aligned to the real world locations. When the real playground is inspected by the game designer, he uses the before created traditional game to establish the spatial references. He moves his character to the corresponding point in the virtual world and simply hits the “Align” button. After this is done for all game locations a new map is automatically created by distorting the existing game map.

4. EXAMPLE: “FORGOTTEN VALLEY – MR EDITION”
To illustrate how the MobEE game-engine described before can be used in practice we have created a small adventure game “Forgotten Valley”.

By choosing to play a new game the user finds his Avatar placed in the middle of a unknown map, not knowing where he is or how he got here. In mixed-reality mode the user can start physically anywhere on the university campus that is our real-world “game board” for “Forgotten Valley”.

Figure 2: Riddles to solve (“Gate” left and “Oracle” right)
The user has to solve several little puzzles (see figure 2) and talk to the people populating the valley to eventually find his way out.

When the user chooses to continue a game that he started at an earlier time, he is presented with an automatically generated re-narration of his previous adventures in the game world (see figure 3). The context refresh shows the most important events in the storyline (as specified by the game designer). The context refresh or scenes therein can be skipped by the user by pressing the “fast-forward” button.

Figure 3: Context Refresh, showing an important part of the story
The game uses background music, spoken parts and written text to tell a story that is designed to be interesting and captivating. Clicking on the menu-bar the user can choose between different combinations of output modalities (e.g. text, graphics, audio, or
mixed-reality). The same adventure can thus be played as a pure text-adventure, as a 2D graphics game or a mixed-reality experience using the same game-engine. To ensure an enjoyable game experience in text-only mode more detailed descriptions of the locations could be added to substitute for the graphics and a linked map in order for the avatar to move around. The following sub-section describes the mixed-reality mode in more detail.

4.1 The Mixed-Reality Mode

Gameplay in mixed-reality mode is similar to that in normal mode as described before: While navigating the user is presented with a scrolling raster map of the university campus on which icons representing the “game locations” are added if the user has explored the corresponding part of the game. At a “game-location” the user can interact with the real-environment using the camera on the PDA. Our current version of the mixed-reality setup is implemented on a HP iPaq Pocket PC PDA with a plug-in camera (FlyCam). To track the users position in the real world while he is walking around, we use a GPS-sensor (Holux GR-230), which has a wireless Bluetooth connection to the PDA. Every second the GPS-sensor sends the user’s current coordinates using the NMEA-protocol. Since the low update-rate of the GPS can be problematic for the interactive “feel” of the game, the navigation automaton updates the position information every 333 milliseconds, extrapolating two in-between positions from the last measurements.

At the “game-locations” or hotspots the user interacts with the game more intensively than just navigating. Here he meets NPCs (Non-Player-Characters), solves riddles, fight dragons and so on. In this way the “game-locations” are used to continue the narration of the game. As mentioned above, the GPS data is not sufficient for these interactions. Instead the user can take snapshots of the environment into which AR-Toolkit markers have been placed. These pictures are then augmented spatially correct with the corresponding game content. Depending on the game content taking snapshots of specific markers is also used as an interaction technique to trigger actions within the game.

Figure 4 shows the same riddles as in Figure 4 within the physical environment on the campus.

![Figure 4: MR-locations “Oracle” (with Markers)](image)

When the user approaches the group of stones (Figure 7, left) the scrolling map on the PDA signals a possible “game-location”. When the user takes a picture of one of the markers the “Oracle-Riddle” starts. When he succeeds, additional information is displayed, that tells him about “a dangerous dragon of huge ancient wisdom” and the story continues.

5. CONCLUSIONS AND OUTLOOK

We have presented the prototype of a mobile mixed-reality game that adapts the game presentation and content to the user’s context. First informal tests of the game have produced some interesting results: While navigating on the university campus test-users did not use the PDA continuously, but only sporadically checked it for new information. Instead they were trying to spot the characteristic markers on the “game-locations”, an artefact of our current implementation. Since such a marker based approach is only viable for short tests, the replacement of marker-based position identification is an important prerequisite for larger scale evaluations. Possible approaches include the use of differential GPS and orientation and acceleration sensors.

Despite many technical shortcomings of the prototype, early test-users were satisfied with the overall game experience. Especially the idea of being able to complete a game in conventional mode at a later time received positive comments. With regards to the question raised in the introduction the answer thus seems to be: yes, it is possible to create compelling gaming experiences in which a physical environment is employed as a meaningful part of the game. However, this raises many more questions in the areas of technology, game design and content authoring that will have to be addressed if mobile MR gaming is to become a practical reality.

From a game design perspective mechanisms that support intermittent use over an extended period of time must be further developed. The same is true for the use of user body motion as an interface technique.

6. REFERENCES