

# Augmented Reality Agents in the Development Pipeline of Computer Entertainment

István Barakonyi

Dieter Schmalstieg

Graz University of Technology, Inffeldgasse 16a, A-8010, Graz, Austria  
{ bara | schmalstieg } @ icg.tu-graz.ac.at

**Abstract.** Augmented reality (AR) has recently stepped beyond the usual scope of applications like machine maintenance, military training and production, and has been extended to the realm of entertainment including computer gaming. This paper discusses the potential AR environments offer for embodied animated agents, and demonstrates several advanced immersive content and scenario authoring techniques in AR through example applications.

## 1 Introduction

The highly influential book of Bolter and Grusin titled Remediation [1] states that digital technologies such as the World Wide Web, computer games, virtual reality (VR) and augmented reality (AR) cannot separate themselves from earlier media forms such as photography, film or stage. The authors argue that a novel visual media form should first observe and respect, then attempt to surpass earlier media by refashioning, “remediating” them, just like photography remediated perspective painting, film remediated stage and photography, and television remediated film. Augmented reality has recently stepped beyond the usual scope of applications like machine maintenance, military training and production, and has been extended to entertainment including computer gaming. This step is expected to greatly contribute to a wider acceptance of this relatively novel research field, similar to the spread of 2D and 3D computer games supported by the rapid hardware and software developments in computer graphics that have also attracted significant public interest. In this paper we focus on entertainment applications in augmented reality environments.

Our paper has two major contributions. The first contribution is our advanced use of animated characters in augmented reality applications, which reaches back to more traditional media such as film or stage. Since AR environments are typically highly dynamic with a large number of events coming from the physical and virtual world, it is difficult to create detailed scripted behavior for characters to anticipate every conceivable situation, unlike in more controlled interactive 2D and 3D virtual environments. Consequently, it is desirable to employ autonomous software components that proactively and autonomously make decisions without continuously waiting for commands from the user. Therefore we propose *embodied autonomous agents* as a novel, promising direction for interaction development of AR applications.

The second contribution is the application of remediation theory to authoring tools in augmented reality environments. We argue that not only presented content and its perception need to be remediated, but media should also strongly influence authoring tools. While tools to author content for AR have been based on strong roots in classic computer graphics and VR, we claim that augmented reality should exploit medium-specific techniques to smoothen its production workflow: *AR has to penetrate its own authoring pipeline*. Nevertheless, we do not offer a complete solution for creating standalone applications as some larger frameworks do [2][3][4] but provide case studies to explore novel ways of authoring and enabling interaction.

## 2 Motivations and Related Research

Typical forms of computer-assisted entertainment include computer games and story-telling applications. As in all games and stories, a key experience element is fantasy. While playing or participating in a story, a virtual, imaginary world is created within our mind, inhabited by characters and other objects behaving according to the dictates of our imagination. In classic make-believe games this fantasy world and their characters connect to the real environment through physical game props to which various roles are assigned, thus making heretofore passive objects active players in the game story. AR applications are aiming at the same effect by superimposing a virtual world on top of the real environment. Since the virtual world is registered with the real one, they appear to coexist. As pointed out by Stapleton et al. in their mixed-fantasy framework [5], the combination of the real and virtual helps suspend disbelief and enriches the audience's fantasy experience. AR is able to visually change real world attributes, make passive objects appear animated or play sound effects in addition to sounds in the real environment further enhancing the atmosphere of the perceived mixed environment.

### 2.1 Authoring AR

Besides being a compelling environment for digital entertainment, augmented reality offers novel, intuitive tools for content authoring as well. MacIntyre et al. [6] make the point that the most relevant factor making AR an important and unique medium is the combination of three concurrent features: blurring the boundary between the real and the virtual world, constant user control of viewpoint and interactivity. Traditional tools for authoring VR and AR content, such as desktop-based 2D/3D content authoring tools, character animation programs, multimedia frameworks [3], or editors to compile and run scripts [2], offer the evident advantage of familiarity. However, an authoring environment possessing the aforementioned three qualities would be desirable to let content developers fully experience and understand the novel environment they develop for and tailor the content to it.

From the various extant tools for authoring content for AR applications we highlight the tangible AR-based system from Lee et al. [7], which is to our knowledge the only general purpose immersive authoring AR system existing to date. This system

implements an intuitive WYSIWYG editor, where users handle physical markers to manipulate virtual objects, their properties and their connections to other objects and properties to model interaction. Although the immersive environment and the tangible controls are similar to ours, the data flow model covers only basic functions of tangible AR applications without considering more complex aspects such as application events or multi-user aspects.

## 2.2 Autonomous Agents in AR

Animated characters have been particularly important and popular visual story and interface elements in a large variety of digital entertainment domains from animated 3D cartoons to computer games, where these characters have been used as avatars, allies, bystanders or competitors of the user. They are able to create a rich gaming experience, since their advanced multimodal communication capabilities including speech, facial and body gestures and posture engage users emotionally and socially as well. They are perceived as living, feeling, and “smart” digital creatures.

Characters in AR may possess a virtual or *physical* body, or both. They share users’ physical environment, in which they can freely move using all 6 degrees of freedom. Virtual characters in AR scenarios appear to have a solid, tangible body that can be observed from an arbitrary viewpoint, thus becoming integral parts of the physical environment. A typical AR environment is highly dynamic including several users equipped with various devices and constantly changing object poses. We think that a preferable way of implementing animated characters for AR environments is by using autonomous agents that proactively make decisions instead of demanding constant user guidance, while monitoring and observing changes in the physical and virtual environment. According to Bates [8] appropriately timed and clearly expressed emotion is an essential and often sufficient requirement for characters to behave in a believable way. Therefore we do not focus on equipping our agents with complex behavior engines or intelligence (without excluding them). Instead we have been researching the technical challenges, user requirements and possible applications that potential AR environments suggest given the possibility of populating them with animated agents.

Although extensively researched in VR, agents have only recently appeared in AR environments. An early AR application providing character support is the ALIVE system [9], where a virtual animated character composited into the user’s real environment responds to human body gestures on a large projection screen. This type of display separates the user’s physical space from the AR environment, which demands carefully coordinated user behavior. The Welbo project [10] features an immersive setup, where an animated virtual robot assists an interior designer wearing an HMD. The character lacks a tangible physical representation and can only interact with virtual objects. Another HMD-based system from MacIntyre et al. [6] creates an interactive theater experience by placing prerecorded video-based actors into an AR environment. The characters do not possess any autonomy, as their behavior is scripted, and interaction is limited to changing viewpoints and roles in the story. Cheok et al. [11] also experiment with mixed reality entertainment with live captured 3D charac-

ters, which enable telepresence of real people within a virtual or augmented reality setting but without any control of the environment. Cavazza et al. [12] place a live video avatar of a real person into a mixed reality setting, and interact with a digital storytelling system with body gestures and language commands. Balcisoy et al. [13] experiment with interaction techniques with virtual humans in mixed reality environments, which play the role of a collaborative game partner and an assistant for prototyping machines. Cassell et al.'s Sam agent [14] is a virtual playmate assisting children in a natural storytelling play with real objects. In this game access to real game props is shared between the child and the animated agent.

### **3 Events and Interaction in the Real and the Virtual**

Interaction with animated agents in an AR environment needs careful design. The set of possible application events is huge, since numerous events are generated by changes in the pose and attributes of multiple users, interaction devices and displays, pose-tracked application-specific props and the application itself. Monitoring all possible events and their combinations is a complex and time-consuming task that can affect the responsiveness of the agents if carried out naively. Therefore as one of the first application design tasks we need to limit the set of events exclusively to those meaningful in the context of the application and set up real and virtual sensors to measure and catch them. The events help the agent maintain its internal world model that attempts to match the perception of the user, and result in the agent making decisions and actions based on the current status of its internal world. To effectively design which application events may be useful for our agents, we wrote simple demo scenarios to test atomic behavior patterns, where various communication channels are utilized between the real and the virtual.

#### **3.1 Agent with Virtual Body Reacting to Events in the Real World**

Figure 1 illustrates a simple example how a virtual animated agent is able to respond to changes in attribute changes of the real world. A tangible, physical optical marker acts as a platform for a virtual monster artiste to stand on. The user holds and tilts around the marker in front of a webcam, while the artiste agent appears to be struggling to maintain its balance. If the surface of the marker becomes too steep, the monster falls down with a roar of despair. The application retrieves the current pose of the marker relative to the webcam using the ARToolKit optical marker recognition module [15]. The pose of the marker is directly mapped to the pose of the virtual platform of the artiste in the agent's world model. The webcam and the marker recognition library act as the agent's sensor for perceiving changes in the physical marker's attributes. The agent's control logic then checks whether the platform orientation is still within bounds and decides whether to play the "falling down" or the balancing animation. The balancing animation is a blended motion interpolating between the neutral center and four extreme points in the animation space with factors calculated from the platform's pitch and roll rotation angles.

### 3.2 Agent with Physical Body Reacting to Events in the Virtual World

Sensing events generated by virtual objects is usually not a complicated task since virtual sensors such as vision, hearing, touching can be implemented in software using various algorithms. However, using physical objects as bodies for autonomous agents and consequently as output communication modalities implies several constraints. Our entire physical environment cannot be affected by virtual control logics, only by specially prepared objects which require communication channels and actuators to be set up.

The screenshots shown in Figure 2 depict a sample scenario for a communication flow from the virtual into the real world. This scenario implements defensive behavior for a physical LEGO® Mindstorms robot that tries to avoid collision with a virtual character. The pose of the robot and the character is again tracked using ARToolKit. If the character enters the virtual “safety” area around the robot, a command is sent to the robot from the PC via an infrared link, instructing it to move away.

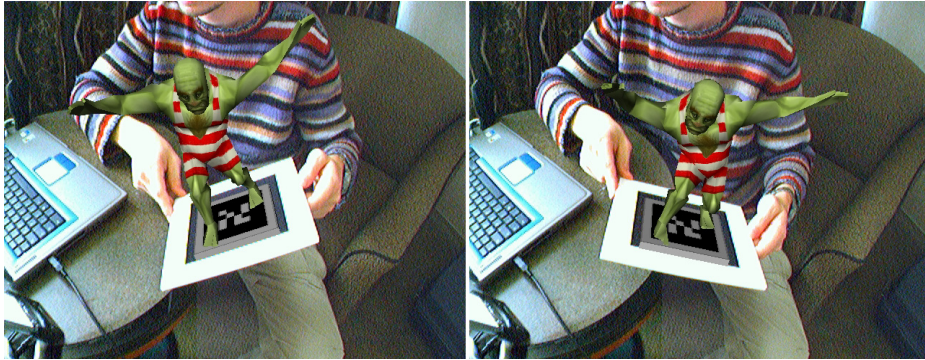


Fig. 1. Screenshots of a virtual animated character balancing on top of a tangible optical marker

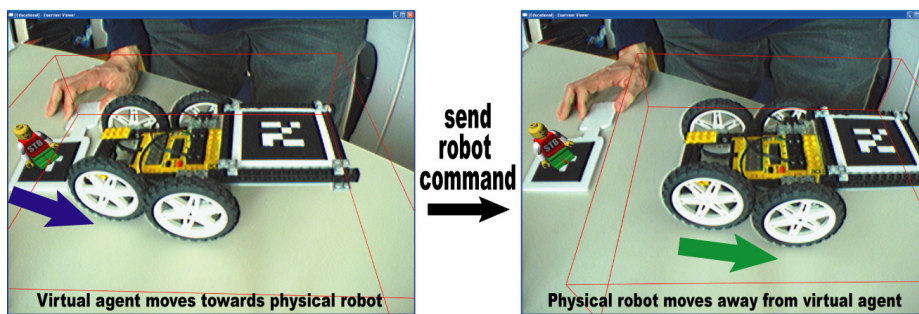


Fig. 2. Screenshots of a real LEGO® Mindstorms robot avoiding collision with a virtual character

## 4 MonkeyBridge

Games provide a challenging environment to test real and virtual world events with animated characters and demand various authoring tasks as well. Our MonkeyBridge application is a multiplayer AR game, where users place real and virtual objects onto a physical surface, thus influencing the behavior of virtual animated characters and responsive physical objects.

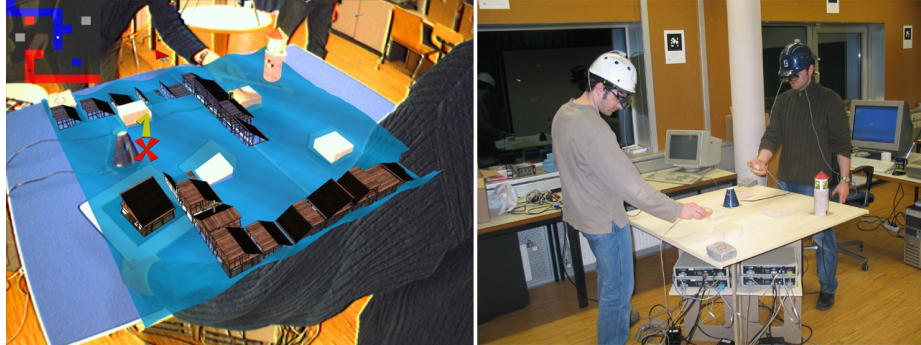
A “monkey bridge” is a fragile wooden construction over a river in South-East Asia. People frequently risk their lives as they try to keep their balance whilst crossing. In this application two players dynamically build a monkey bridge for their monster-like characters using virtual and physical pieces of landing stage, which vary in shape (Figure 3 shows two application screenshots). The goal is to reach a dedicated target in the middle of a virtual ocean. The game includes many spectacular virtual and physical visual elements such as animated 3D ocean waves, a flock of virtual seagull boids, a real, illuminated smoking volcano and lighthouse with rotating lights. Sound effects further enhance the game experience.

### 4.1 Autonomous Path and Motion Planning for Agents

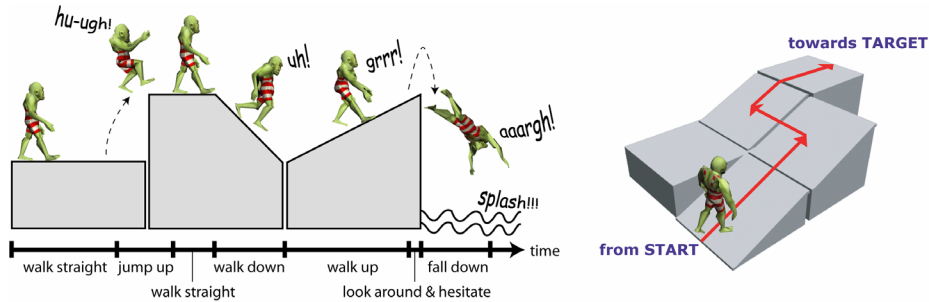
The MonkeyBridge characters are embodied autonomous agents. Their behavior does not require careful scripting, instead a dedicated control logic or virtual “brain” decides which animations and sound effects to play, which direction to turn or whether the target has been reached. The only factors that directly influence agent behavior are the spatial distribution, pose and shape of the virtual and physical building blocks placed on the game board (see Figure 4). The characters autonomously choose: the path they walk on; decide how to get from one platform to the other, e.g. jump up or down when there is a slight difference in height between platform edges; automatically choose the straightest path from several available tiles; and fall into the water if there is no suitable piece of landing stage to walk on. They cheer in triumph raising their hands in a victory salute when they win, and cry over a lost game.

### 4.2 Level Editing in AR

It is transparent to the agents whether they walk on physical or virtual tiles, which blurs the boundary between the real and the synthetic game environment and allows the arbitrary combination of real and virtual bridge elements on the game table. The different arrangements of bridge elements result in a large number of level combinations that prevent the game from becoming monotonous. Augmented reality enables the use of an intuitive tangible level editor, where users design the pose of the physical game elements in the game area with their hands prior game start, record the manually edited physical level and register it with its virtual counterpart to initialize the game. After the game starts, players dynamically build the virtual level elements during the game by manually placing them next to the physical tiles using their interaction devices.



**Fig. 3.** (left) Screenshot from the optical tracking-based setup. Note the real game elements that show through the virtual scene. (right) Game environment with the magnetic tracking setup.



**Fig. 4.** Autonomous agent behavior: a) choosing animation and sound based on platform type, b) path planning depending on the spatial distribution of available blocks

### 4.3 Character Animation Using AR Techniques

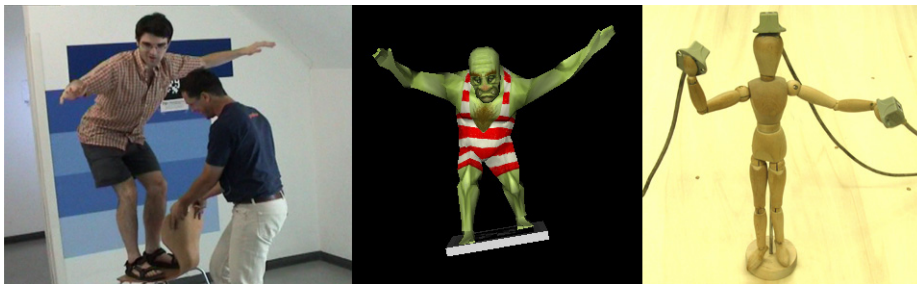
Modelers and animators often rely on real-life references to build and animate 3D characters for games or film production. Observing the real world by photographing or videotaping the subject, or asking someone to pose helps create more believable, precise and expressive character animation. The left image of Figure 5 shows a colleague posing while creating animation for the balancing demo described in Section 3.1. Professional artists use motion capture techniques or other expensive means of acquiring motion data (e.g. the Monkey kinematic tracker device [16]) to create an essential initial data set for the final, refined animation. With AR new possibilities open up within the field of character animation. The animated virtual model and the real-world reference can be merged, forming a single interactive modeling instrument. We are currently developing a tool using a wooden mannequin as an input device to animate skeleton-based virtual 3D models. The head and limbs of the mannequin are pose-tracked. The system maps real-time pose information to rotation information for the joints of the character skeleton using inverse kinematics and motion mapping techniques. AR not only enables close interaction with virtual models



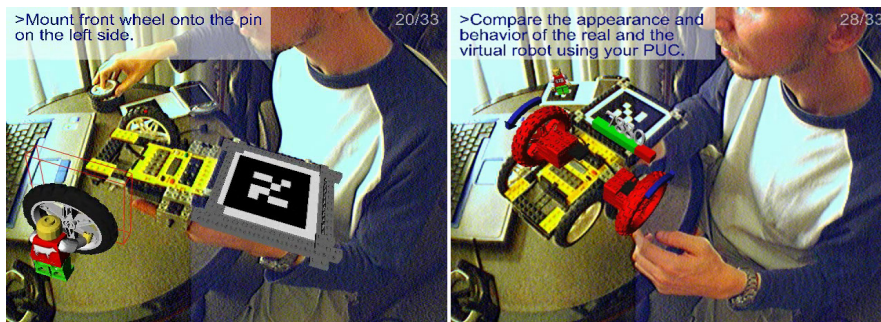
by using tangible objects but also the creation of complex motions like walking up stairs or lifting a ball, since animators can use actual physical models of stairs or balls in concert with characters in order to create realistic motions.

## 5 AR Lego

AR environments offer advanced exploitation of real-world objects. The AR Lego application [17] implements a playful machine assembly and maintenance scenario, in which two agents are employed to educate an untrained user to assemble, test and maintain machines composed of active (engines and sensors) and passive (cogwheels, gears, frames) parts. The two agents are a real LEGO Mindstorms® robot and a virtual animated repairman (see the screenshots of Figure 6). The PC-based application treats the robot as a first-class interaction entity by communicating commands via an infrared channel, in order to control the attributes of active parts (e.g. engine voltage and direction, type of the sensors) and query current robot state (e.g. sensor values, communication channel failures, or battery level).



**Fig. 5.** Human actor (left) and wooden mannequin with magnetically tracked limbs (right) as a model for the balancing monster animation (center)



**Fig. 6.** Screenshots from the AR Lego application. (left) An animated repairman mounts virtual wheels onto the physical Lego® robot (right) The system gives visual feedback about the behavior of the real robot while the wheels are turning.



The system provides step-by-step assembly instructions as to which block to mount next and how to verify whether the user is at the correct stage in the construction. While the verification of passive parts (i.e. inactive bricks) is only possible by visually comparing the appearance of the physical model to the virtual, testing whether the active parts (engines and sensors) have been mounted correctly is more straightforward. After mounting an engine, the application instructs the robot to turn the engine on by issuing an infrared command. If mounted in the right position and correct direction, the engine and all moving parts connected to it should behave as demonstrated by overlaid animated virtual models. Similarly, if we mounted the sensors properly, the right type and range of data should arrive from the robot. The system checks and visually reports inconsistency so that the user can go back one or more steps to double-check the construction.

### **5.1 Physical objects as GUI**

A novel and exciting aspect of AR agents is that physical objects like printers, digital pianos or interactive robots can be turned into intelligent, responsive entities that collaborate with virtual characters and users as equal, active partners. If we track and monitor relevant physical attributes and process this data, attribute changes can generate events that can be interpreted by other agents and application logic. Using network packets, infrared messages, MIDI code sequences or other means of low-level communication, physical objects can not only be queried for status information but can also be controlled by external commands that trigger actuators. Therefore physical objects can act as input and output devices in AR spaces.

In AR Lego the Lego® Mindstorms robot is a physical object that not only acts as an I/O device in the AR environment but also serves as an immersive modeling device. The user manually and dynamically assembles a real, tangible Lego® model from active and passive parts. During construction constant visual feedback is sent by a display combining real and virtual information, and by the actual physical model through actuators triggered by infrared commands. The ActiveCube project [18] implements similar ideas, where the user assembles a set of computerized tangible blocks equipped with I/O devices. However, in ActiveCube there is a clear border between the real and the virtual world since the virtual content is not overlaid on top the physical environment, which is a key requirement of true AR applications.

### **5.2 Personal Universal Controller**

An innovative way to configure AR agents is using the Personal Universal Controller (PUC) [19]. All agents provide an XML-based description of their relevant, configurable attributes and their supported commands together with the command syntax. All characters run a PUC service, which listens to incoming connections from PUC clients. A PUC client is able to render a Graphical User Interface (GUI) to control attributes that are described in the description, and are implemented on various devices and platforms including PCs, PDAs and smartphones. Figure 7 shows a control GUI rendered on a PocketPC. By checking the “skeleton” control checkbox (marked with

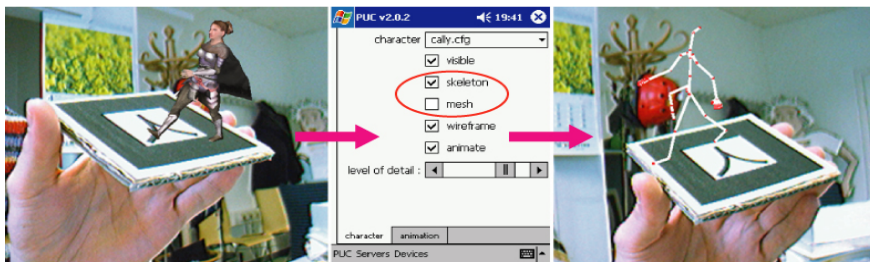
an ellipsoid) the user changes the rendering mode from mesh to skeleton mode allowing observation of the underlying bone structure. One PUC service can accept multiple clients, hence allowing collaborative, multi-user configuration.

PUC offers an intuitive way to configure animated characters in AR environments, where they are typically scattered in space. Instead of the usual static configuration steps prior to the start of an application, the user can now roam throughout the complete application space; walk up to the character to be configured; reach a convenient and comfortable distance; query the controllable parameters of the character using a PDA running a PUC client; and dynamically control the character's attributes.

## 6 Augmented Piano Tutor

Music has only recently emerged as an important medium for AR applications [20][21]. The Augmented Piano Tutor application serves as a piano teacher that educates users about basic chords and scales in an AR environment. It uses a desktop-based AR setup using a real keyboard that communicates with the control PC with MIDI in and out messages, a webcam, and a monitor to combine the real and the virtual scenes. On the screen the real keyboard can be seen augmented with virtual information instructing users to hit certain keys in a defined order, while giving feedback as to which keys have been pressed correctly and which ones were pressed by mistake. Sound is generated on the physical keyboard by MIDI commands in accordance with the visual feedback blended into the user's view, therefore the boundary between the user situated in the real environment with the instrument and the "teacher" giving instructions from the virtual world appears to be blurred.

The synthetic information is always synchronized with the audio. When the user is instructed to play a certain chord on the keyboard, the piano keys yielding the chord are visually highlighted indicating which should be pressed, while the very same chord is played on the keyboard to create a mental connection between the two. When the user tries to imitate the chord, the correctly pressed and missed keys are marked with different colors on top of the real keys, giving a hint how the error should be rectified. Figure 8 provides illustrations for our application.



**Fig. 7.** Animated character bound to a tangible optical marker and controlled by a PUC generated interface on a PocketPC (monitor + PDA screenshots)



**Fig. 8.** Screenshots from the Augmented Piano Tutor Application

This system can be used as an advanced music composition tool offering the complex functions and rich visual feedback of a sequencer program running on the PC, while preserving the simplicity and freedom of a keyboard. While the sequencer module analyzes the tunes currently being played, it could also suggest harmonizing background chords and appropriate solo melodies to be played immediately on the keyboard.

## **7 Conclusion and Future Work**

In this paper we introduced various augmented reality applications to illustrate the use of embodied autonomous agents as key visual interface elements and advanced immersive authoring techniques and tools exploiting AR environments. We did not aim to construct a standalone AR authoring framework but rather developed sample scenarios to provide inspiration for further experimentation with AR authoring for computer entertainment.

In AR applications users are typically not bound to a single physical location, device and display but tend to roam over large areas, working with several stationary and portable devices and displays. We are currently experimenting with persistent embodied autonomous agents that are able to seamlessly migrate between various devices and screens, and develop authoring tools accommodating the needs of mobile users working with animated agents in ubiquitous computing environments.

## **8 Acknowledgement**

This project has been sponsored by the Austrian Science Fund FWF (contract no. Y193) and the European Union (contract no. IST-2001-34204). The authors wish to say thanks to Ulrich Krispel and Christoph Schinko for integrating inverse kinematics into our character animation framework, to Joseph Newman for reviewing the manuscript, and to the members of the Studierstube team for their continuous support.

## References

1. Bolter, J. D., Grusin, R.: *Remediation: Understanding New Media*. MIT Press, Cambridge, Massachusetts, 2000, ISBN-0-262-52279-9
2. Ledermann, F., Schmalstieg, D.: APRIL: A High-level Framework for Creating Augmented Reality Presentations. In: Proc. of the IEEE Virtual Reality 2005 Conf., Bonn, Germany, 2005, pp. 187-194.
3. MacIntyre, B., Gandy, M.: Prototyping Applications with DART, The Designer's Augmented Reality Toolkit. In: Proc. of Software Technology for Augmented Reality Systems Workshop (STARS 2003), Tokyo, Japan, 2003.
4. Zauner, J., Haller, M., Brandl, A., Hartmann, W.: Authoring of a Mixed Reality Assembly Instructor for Hierarchical Structures. In: Proc. of the 2nd International Symposium on Mixed and Augmented Reality (ISMAR2003), Tokyo, Japan, 2003, pp. 237-246.
5. Stapleton, C., Hughes, C., Moshell, M., Micikevicius, P., Altman, M., Applying Mixed Reality to Entertainment. In: IEEE Computer 35(12), 2002, pp. 122-124.
6. MacIntyre, B., Bolter, J. D., Vaughan, J., Hannigan, B., Moreno, E., Haas, M., Gandy, M.: Three Angry Men: Dramatizing Point-of-View Using Augmented Reality. In: Proc. of SIGGRAPH 2002 Technical Sketches, San Antonio, TX, USA, 2002.
7. Lee, G. A., Nelles, C., Billingham, M., Kim, G.J., Immersive Authoring of Tangible Augmented Reality Applications, In: Proc. of IEEE and ACM Int'l Symposium on Mixed and Augmented Reality 2004 (ISMAR'04), Arlington, VA, USA, 2004, pp. 172-181.
8. Bates, J.: The Role of Emotion in Believable Agents. In: Communications of the ACM 37(7), Special Issue on Agents, 1994, pp. 122-125.
9. Maes, P., Darrell, T., Blumberg, B., Pentland, A.: The ALIVE System: Wireless, Full-body Interaction with Autonomous Agents. In: ACM Multimedia Systems 5(2), 1997, pp. 105-112.
10. Anabuki, M., Kakuta, H., Yamamoto, H., Tamura, H.: Welbo: An Embodied Conversational Agent Living in Mixed Reality Space. In: Proc. of Human Factors in Computing System (CHI 2000), Extended Abstracts, The Hague, The Netherlands, 2000, pp. 10-11.
11. Cheok, A. D., Weihua, W., Yang, X., Prince, S., Wan, F. S., Billingham, M., Kato, H.: Interactive Theatre Experience in Embodied and Wearable Mixed Reality Space. In: Proc. of International Symposium on Mixed and Augmented Reality (ISMAR'02), Darmstadt, Germany, 2002.
12. Cavazza, M., Charles, F., Mead, S. J., Martin, O., Marichal, X., Nandi, A.: Multimodal Acting in Mixed Reality Interactive Storytelling. In: IEEE Multimedia 11(3), 2004, pp. 30-39.
13. Balcisoy, S., Kallmann, M., Torre, R., Fua, P., Thalmann, D.: Interaction Techniques with Virtual Humans in Mixed Environments. In: Proc. of International Symposium on Mixed Reality (ISMAR 2001), Tokyo, Japan, 2001.
14. Cassell, J., Ananny, M., Basu, A., Bickmore, T., Chong, P., Mellis, D., Ryokai, K., Smith, J., Vilhjálmsson, H., Yan, H.: Shared Reality: Physical Collaboration with a Virtual Peer. In: Proc. of Human Factors in Computing System (CHI 2000), The Hague, The Netherlands, 2000, pp. 259-260.
15. ARToolKit Developer Homepage, <http://sourceforge.net/projects/artoolkit/>
16. Esposito, C., Paley, W. B., Ong, J. C.: Of mice and monkeys: a specialized input device for virtual body animation. In: Proc. of the Symposium on Interactive 3D Graphics, 1995, Monterey, CA, USA, pp. 109-114.
17. Barakonyi, I., Psik, T., Schmalstieg, D., Agents That Talk And Hit Back: Animated Agents in Augmented Reality. In: Proc. of the IEEE and ACM International Symposium on Mixed and Augmented Reality 2004 (ISMAR'04), Arlington, VA, USA, Nov. 2-5, 2004, pp. 141-150.
18. Watanabe, R., Itoh, Y., Asai, M., Kitamura, Y., Kishino F., Kikuchi, H.: The Soul of ActiveCube - Implementing a Flexible, Multimodal, Three-Dimensional Spatial Tangible Interface. In: Proc. of ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE 2004), Singapore, 2004, pp. 173-180
19. Nichols, J., Myers, B. A., Higgins, M., Hughes, J., Harris, T. K., Rosenfeld, R., Pignol, M.: Generating Remote Control Interfaces for Complex Appliances. In: CHI Letters: ACM Symposium on User Interface Software and Technology, Paris, France, 2002, pp. 161-170.
20. Poupyrev, I. R., Berry, R., Kurumisawa, J., Nakao, K., Billingham, M., Airola, C., Kato, H., Yonezawa, T., Baldwin, L.: Augmented Groove: Collaborative Jamming in Augmented Reality. In: Proc. of SIGGRAPH 2000 Conference Abstract and Applications, ACM Press, pp. 77, 2000.
21. Cakmakci, O., Berard, F.: An Augmented Reality Based Learning Assistant for Electric Bass Guitar. In: Proc. of the 10th International Conference on Human-Computer Interaction, Crete, Greece, 2003.