

A Platform for Supporting the Development of Mixed Reality Environments for Educational Games

Luis Arenas, Telmo Zarraonandia^(✉), Paloma Díaz, and Ignacio Aedo

Computer Science Department, Universidad Carlos III, Madrid, Spain
{larenas, tzarraon, pdp}@inf.uc3m.es, aedo@ia.uc3m.es

Abstract. In this work we present MR-GREP, a platform that supports educators in the design and implementation of mixed reality educational games. More specifically the system allows the instructional designers to create game experiences that can be played simultaneously in an augmented reality environment as well as in a virtual space that replicates the physical space. This seeks to support the implementation of collaborative learning experiences that combine the flexibility provided by the virtual worlds with the realism and physical component that training in real scenarios allows. The development of this type of environments usually demand advanced technical knowledge that educators normally lack. The platform aims to overcome this problem by providing a set of authoring tools and applications, which does not require programming skills from the user.

Keywords: Digital Educational Games · Mixed reality · Augmented reality · Virtual reality

1 Introduction

In recent years the development of Digital Educational Games (DEG) has been an active research line. DEGs aim at helping to acquire knowledge and developing skills while maintaining user engagement and motivation. This type of educational resources has proven to be specially useful for the development of skills such as problem-solving and critical thinking [1]. Despite their benefits, the introduction of DEGs in the learning system is not as widespread as one would expect. The development of a digital game requires a variety of skills and specialized knowledge that make necessary the assistance from technical experts. This represents a prohibitive cost for many educators willing to embrace new technology. Although some platforms such as < e-Adventure > [2] and StoryTec [3] provide authoring tools for designing and implementing DEGs without requiring programming skills, the type of DEGs they can produce is restricted to the adventure genre. Action games that train the physical or mental agility of the learner are beyond the scope of these tools, for example.

Current DEGs provide a mock scenario in a virtual environment in which the player performs certain tasks and where she is allowed to explore different courses of action. While this can suit well the training of certain skills such as decision taking, it does not support so well exercising other abilities in which the physical component is key.

This is the case of those games in which the stamina or speed of the player is a decisive factor in the success of the experience. This scenario could be improved if the game is played in an Augmented Reality (AR) environment, which would allow the enhancement of a physical space with virtual information, supporting the implementation of a situated learning approach. In AR games such as ARQuake [4] or Human Pacman [5] the action of the game takes place in a real setting, and the players are equipped with AR devices that allow them to visualize the game entities as superimposed on their own vision of the environment. In an educational context, this type of games would present unique advantages as they combine the benefits of trainings in a physical space, while retaining the inexpensive management of virtual objects and user safety provided by virtual reality simulations. Unfortunately, the design and implementation of this type of experiences are specially challenging. On top of the difficulties associated to the production of traditional DEGs, they add the complexity of designing a consistent AR environment in which games can be played. The designer could use high level authoring tools like DART [6] and CATOMIR [7], which bring the definition of AR experiences closer to non-experts. However these platforms may still require some programming and are not particularly focused on game development or education.

The objective of this work is to provide educational designers the tools and platforms that facilitate the design and development of DEGs that can be played in a Mixed Reality environment, minimizing dependence from technical assistance in the process. In this case we use the term Mixed Reality (MR) to make reference to an environment which merges physical and virtual worlds, where the elements from both realities coexist and interact with each other. MR extends the interaction possibilities that AR offers by allowing to act not only from the augmented real world but also from a replicated virtual setting.

The type of games whose production we aim to support would allow interacting with the educational experience both from an augmented physical space as well as from a replica of it in a virtual reality environment. The final aim is to allow different players to participate in a gameplay that is simultaneously played in the two types of environments, interacting in real time with each other and with the other game entities included in the game. This type of setting will open exciting possibilities for training and education, as they will offer a more complete training scenario where trainees can interact with the game from the type of environment that better supports the acquisition of the specific skill to develop. Furthermore, it will also deliver unique possibilities for collective trainings and educational experiences where the actions of the players in the virtual world will modify the experience of the players in the real environment and vice versa.

As a first step towards this goal we present MR-GREP (Mixed Reality – Game Rules scEnario Platform). The system provides a set of tools and applications for supporting the design, development and execution of mixed reality DEGs. The current implementation of the platform allows the creation of mixed games where players in the augmented real world act as on-site observers of the game action that takes place in the virtual space. Presently the use of this type of systems in real-world learning environments is restricted due the weight and cumbersomeness of current models of fully immersive AR headmounted displays. However, this work serves as exploratory research to enable the study of the potential usefulness, difficulties and limitations of

mixed reality educational games. The insights obtained through the use of this platform will help to develop better games to be used when the technology matures and reaches the general public.

The rest of the document is structured as follows: Sect. 2 presents related work in educational game development, Sect. 3 details MR-GREP platform and its components, Sect. 4 presents a use case created for illustrating the proposal, and Sect. 5 describes the conclusions and future lines of work.

2 Related Work

Even though there already exist several AR and MR games both for entertainment and education, these applications tend to be produced as standalone projects tailored for specific domain [8]. To the best of our knowledge, there is little support for the development of these kinds of games by the hands of end users (EUD).

Efforts have been made in order to alleviate the complexities of AR and MR application programming through the implementation of libraries and frameworks like ARToolkit [9] and OSGART [11]. However, these tools are not suited for end users since programming skills are required. Higher level authoring tools such as DART and CATOMIR allow the definition of interactive AR and MR experiences. These authoring tools bring the development closer to end users by allowing the use of visual programming techniques instead of coding. However, they do not specifically support the creation of games but isolated experiences, and they are not specially oriented to educational environments.

With regards to DEGs, game development has typically been delegated to experts in C or C++ programming and low level graphical libraries like DirectX and OpenGL. This scenario has been improved with the advent of game engines such as Unity or Unreal Engine [10]. These platforms provide specific support for game aspects like sound, animation, physics, behaviours and game deployment. Although they contribute to lower the level of difficulty of programming digital games, they are only appropriate for programmers. StoryTec and < eAdventure > are two of the few existing authoring tools that do focus on end user game development. StoryTec allows the creation and execution of interactive stories through an authoring platform and a runtime engine. Its authoring platform is composed of five visual components that let users define story logic, stage, action set, assets and properties. On the other hand, < eAdventure > specifically concentrates on providing instructors with the required tools for creating story-driven games as well as to integrate these games and the associated results into e-learning environments. Unfortunately, these tools are restricted to the development of adventure games.

3 MR-GREP

The MR-GREP system integrates different tools and applications for supporting both the design and development of mixed DEGs as well as their execution. Figure 1 depicts the relations between these tools and presents them organized by type of environment

and activity supported. This way, the tools depicted in the top half of the diagram (GREP Editor, GREP Player) provide support for the definition and execution of the game action that takes place in the virtual world, whereas the ones included in the lower half (Augmented Scene Editor, Augmented Scene Player) allow the translation of these actions into an augmented real environment. In the same way, the modules on the left hand side of the picture (GREP Editor, AS Editor, AS Player), assist the designers in the process of creation of the EGs, while those on the right (GREP Player, AS Player) allow the players the retrieval and execution of the EG produced. EG definitions are stored in the platform’s game server (middle of the diagram), which also ensures that the state of the virtual game elements and their corresponding augmented representations are synchronized.

The process of defining a mixed reality game involves the following steps:

1. Virtual scene and game definition: As a first step, the user designs a virtual reality game replicating a real world space. This process includes the definition of the game scene as well as the game rules.
2. Augmented scene definition: The replicated area in the real world space is augmented with digital graphical representations.
3. Matching between virtual and augmented scene elements: Finally, the elements of the virtual and augmented scenes are matched so that changes in the positions and states of the entities of the virtual games are translated into changes of the graphical representations added to the augmented scene.

The following sections describe the different parts of the system that support the definition of the virtual world and the game rules, the augmentation of the counterpart scenes in the real world, and the matching between the virtual and augmented scenarios.

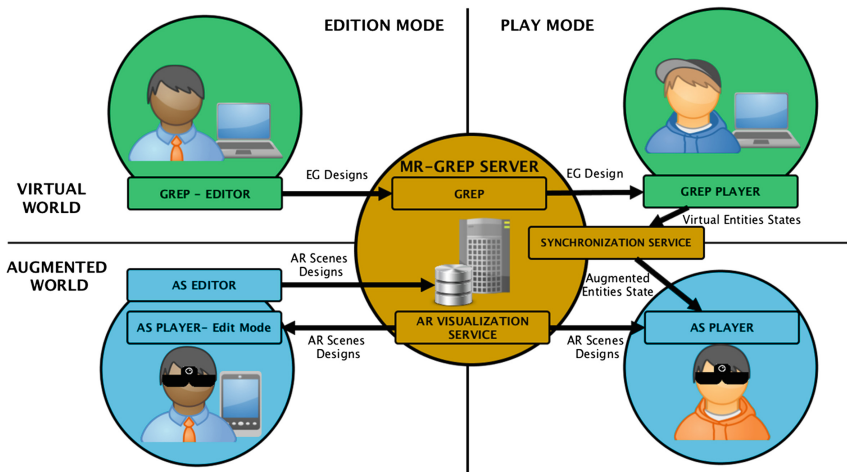


Fig. 1. System diagram

3.1 Virtual World and EG Definition: GREP

The definition of the EG is carried out using GREP (Game Rules scEnario Platform): a system able to interpret descriptions of EGs expressed in XML files and to generate 3D games based on them. The platform has been implemented using the Unity 3D engine. The descriptions of the games the platform interprets should follow the schema of the GREM model (Game Rules scEnario Model) [11], which provides a set of components and design entities for defining EGs. GREP provides different types of implementations for these game components, activating for each game the ones that suit better their descriptions in the XML files. For example, and with regards to the game interface, GREP provides different types of inventory windows, score sections, status bars and a mini-map view. For supporting different types of game mechanics, the platform provides listeners based on entities' current positions, thresholds of attribute values, collisions and the triggering of actions, among others. In the same way, it includes device components implementations for supporting the definition of games compatible with a keyboard, a mouse and a Microsoft Kinect device. In addition, the platform also counts with different repositories that the designers can populate with resources for their games.

The designer defines an EG and generates its corresponding XML description file using a graphical authoring tool named GREP Editor (Fig. 2). The process of defining an EG using the editor involves four steps:

1. *Definition of static scenes*: The process starts by setting up the main elements and parameters of the game scenes. This includes the specification of the size of the scenes, the lighting, and the definition of the floors and walls and the other non-interactive background elements included in them.
2. *Definition of game entities*: The designer continues defining all the interactive elements of the game or *game entities*. These definitions should include the entity name, the lists of attributes, states and actions that the entity can carry out, and the graphical models and animations of the platform repository that will be used to represent those actions and states.
3. *Definition of the game scenes*: The designer completes the definition of the game scenes by placing in the static scenes instances of the game entities previously specified. To aid the designer in this process the editor provides a *scene view*, which allows to navigate through the scene, to add and remove entities to it, and to modify their position, size and orientation.
4. *Definition of the game rules*: The designer specifies the rules of the game. These are described as a collection of game events that capture conditions of success and failure, as well as the triggering of the entities actions, the modification of their attributes, or their appearance or removal from the scene, for example.

The XML description files are stored in the GREP repository and made available to the learners through the GREP Player (Fig. 2). Using a web browser the learners launch the GREP Player and select an EG to play. The platform starts processing the corresponding XML file of the game, and adjusts the components of the platform to the requirements specified in it.



Fig. 2. GREP Editor (left) and GREP Player (right)

3.2 Augmentation of the Real World

To transform an EG played in a virtual environment into a mixed-reality experience it is necessary to specify the augmented representations that will be used to represent the game entities and to place them in a real world environment that replicates the virtual world scene. To support the designers in this process and to allow players in the real world to visualize the game action, the platform provides two different tools: the Augmented Scene Editor (AS Editor) and the Augmented Scenes Player (AS Player).

3.2.1 Augmented Scenes Editor (AS Editor)

The AS Editor is an authoring tool for supporting the design of the *augmented scenes* in which the EG will be played. An *augmented scene* is a real-world environment augmented with 3D virtual elements, which are placed at fixed positions with fixed orientations and scales. To aid in the process of definition of the *augmented scenes* the AS editor provides three different modules: the scene manager, the model manager, and the scene editor:

The *model manager* module provides the designers a repository of graphical models to be used in their augmented scenes.

The *scene manager* allows the designers to describe the physical space to be augmented. This description should include the real dimensions of the place to augment, and a scaled top view plan of it.

Finally, the *scene editor* allows to define the augmented view of the scene. The editor uses the top view plan previously introduced to obtain references of the positions at which the physical space should be augmented. The designer defines virtual elements to display in those positions, and assigns them identifiers, names and the graphical models from the repository that better represent them. Figure 3 depicts a screenshot of the scene editor, where the framed icons represent the virtual elements placed by the user in the scene. Once a virtual element has been specified, the user can complete its definition by modifying their default distance from the floor, orientation and size.

In order to tackle the problem of providing visual feedback about the elevation on a two dimensional space, a color gradient is implemented. The background color of framed icons are set with an elevation-color mapping from ground level to the highest elevation in the scene. This value is set through a slider. Even though users may find it hard to infer the absolute elevation only from the color, the color gradient eases the

identification of elements at the same elevation and the identification of which element is set higher than another. Nevertheless, the exact value in centimeters may be seen by selecting a virtual object and observing the value displayed on the elevation slider.

The orientation of the virtual elements defines the direction they are facing. The user controls the orientation by using a slider, which adjusts the rotation with respect to the vertical axis. This allows the user to regulate the yaw of the virtual element, making the element pivot on itself, effectively turning it left or right. The orientation is displayed with an arrow head pointing outwards.

Finally, the size of the virtual elements is set by defining a scaling factor with respect to the original size specified by the model. Once again, a slider is used to specify the scale, which will be represented in the scene by the size of the icon that represents the virtual element.

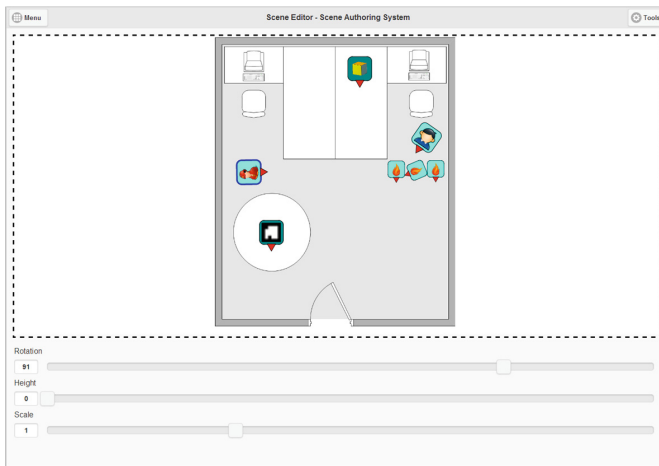


Fig. 3. Screenshot of the scene authoring tool

To ease the edition of multiple virtual elements at once the *scene editor* features multiple selection.

The scene authoring tool is implemented as a HTML5 website for multi-platform compatibility. It is mainly designed for touch interaction on medium-sized screens such as those in tablets, although interaction with a mouse and keyboard on bigger displays is also supported. Specifically, JQuery Mobile was used to design the web interface. The server is implemented in PHP5 with asynchronous AJAX calls and a MySQL database.

3.2.2 Augmented Scenes Player (AS Player)

The AS Player is the subsystem of the platform in charge of augmenting the user view of the physical environment by rendering graphical representations of virtual objects on top of the user view. The system requires the user to be equipped with an optical see-through AR headmounted display, which should permit the augmentation of the whole

visual field. In addition he/she should place in the space to augment fiducial markers to be used as references. With regards of these markers it is necessary to note that they are not directly linked to virtual elements, but they encode reference locations, whose coordinates and orientation are stored in a database. To track the user position and orientation in the area it is necessary that at least one fiducial marker is captured by the camera at any given time. Therefore, the user should distribute the markers in the environment to augment in a way that allows him/her to move around freely, being able to observe the augmented elements from any position. To assist the user in this process, the player can be used in combination with the AS Editor. Thanks to tablet portability, the user of the AS editor may wander around the environment, testing different configurations of the markers positions, and visualizing through the AS Player the different views that the system generates based on them. Figure 4 shows a user placing markers and performing on-site edition of a scene with a tablet, assisted by the player.

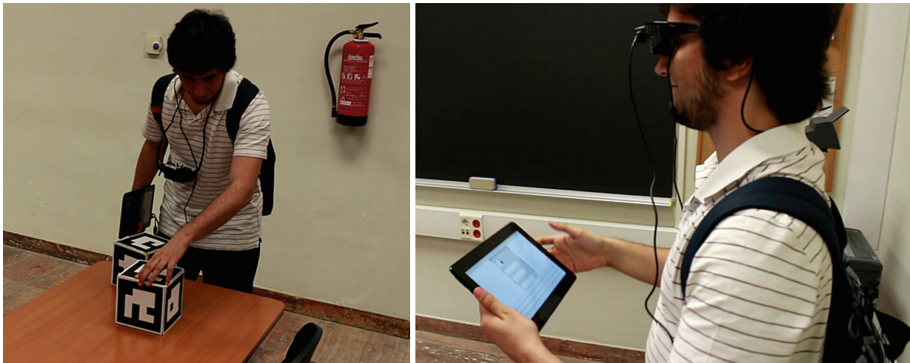


Fig. 4. AS Player-assisted on-site scene edition

The user can improve the accuracy of the location tracking using cubic markers whose six faces provide references of the same position from the six orthogonal perspectives. Figure 4 depicts these cubic markers used as references.

3.2.3 Virtual and Augmented Scenes Matching

Once we have defined an augmented scene and a virtual scene it is possible to define matchings between them in order to have *mixed scenes*. Thus, a mixed scene establishes a correspondence between elements of a virtual scene and elements of an augmented scene (Fig. 5). This way the actions and changes that occur in one scene are replicated in the corresponding elements of the other scene.

The matchings between the virtual reality entities (GREP game entities) and their corresponding elements in augmented scenes are managed through relations in a database. The database schema permits maintaining different matchings between virtual scenes and augmented scenes. This means that one virtual scene can be linked to different augmented reality scenes, each including different representations to suit the requirements of different learner profiles. The connections between *scenes* are static, meaning that they define the initial position of all the elements that are present both in the virtual world (in GREP) and in the real world (through the augmented reality gear).

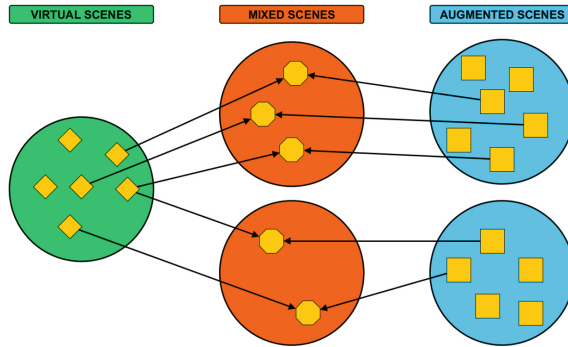


Fig. 5. Mixed scene creation by virtual and augmented scenes matching

Once a matching is done it is possible to define plays in mixed mode, where players could either execute their actions and interactions with the game in the virtual or the real scenes. Consequently, two kinds of users would exist, those who act in the virtual world and those who act in the real world.

A fully mixed game could allow the following:

- A player that plays in the virtual world is able to modify the augmented scene with her actions.
- A player that plays in the augmented scene is able to modify the elements in the virtual world with her actions.

The platform currently supports the first type of interactions. The actions performed by a virtual player affect the virtual objects which, in turn, affect the augmented representations in the augmented scene.

The synchronization of changes in the virtual world affecting the augmented world has been implemented through an event system in GREP. Changes detected on the state of GREP elements are applied to the corresponding augmented scene elements observable by augmented scene players. This implementation consists of a javascript module extending GREP that executes calls to a PHP-based state update service.

4 Use Case: Firefighter Simulation

To evaluate the system a use case has been developed consisting in the implementation of a firefighter and rescue training game in mixed reality modality. The game simulates emergency scenarios that firefighters may stumble upon. In these contexts physical factors are fundamental and time is critical. Firefighters need to wear special protections and gear, whose weight critically reduce their movement freedom and put their stamina to the limit. Additionally, heat and smoke pose obstacles to the realization of their tasks. The implementation of a virtual reality game for firefighter training may be useful for consolidating the studied courses of action to be taken during emergencies, but they lack the real-world characteristics that are so important in these emergency scenarios. On the contrary, a mixed reality game brings these situations closer to the

real emergency situation by having the user play in a real physical environment rather than through a computer screen. In addition, the cost of executing these simulations is rather low since victims and fire may be simulated through augmented reality, and firefighter and victim safety is not risked.

Having defined the rationale behind this use case context selection, the design process of the game with the implemented system is now detailed. As a first step, the real world where the simulation would take place is virtualized and designed in GREP. For this use case, we designed a virtual scene that replicates one of the classrooms of our university, including the disposition of the tables, size of the blackboard, windows, etc. Once the virtual environment is defined, the game entities were defined. The objective of the game is to save a kid from the flames by controlling a firefighter. Consequently, three game entities were defined and added to this game: a firefighter, a kid and flames. Instances of these entities were placed in the classroom scenes replicating a situation that required the firefighter to extinguish the flames to reach the kid. After this, the rules that govern the entities and game execution are created. In our case, one of the rules stated that when a firefighter approaches the flames, his/her health level decreases by some amount per second. Finally, when all game rules are defined a playable GREP virtual game is ready.

Once the virtual game had been defined we proceeded to define the augmented scene following the process detailed in the Sect. 3.2. This way, a simple top view plan of the real classroom was uploaded to the tool to create an empty augmented scene, together with the graphical models to represent the entities of the game. Next, the on-site edition of the scene was performed using an augmented reality headset, specifically Vuzixs Star 1200 model was used for this project. In this case two cubic markers were enough to obtain references of the whole space to augment. With regards to the placement of the augmented entities, the kid was positioned on top of a table, trapped behind some flames, and with the firefighter located on the opposite side of the room.

Finally, the matching between the virtual scene created in GREP and the augmented scene defined with the scene authoring tool was performed, linking the entities that represent the kid, the firefighter and the flames in the two types of environments.

Gameplay.

The current implementation of the mixed game considers two roles: player and observer. The player is in charge of controlling the firefighter and plays the game in the virtual world using a laptop. Meanwhile an observer in the classroom can observe the equivalent game action in the real classroom using AR gear. Figure 6 depicts the firefighter and fire as perceived by the viewer (right) while the virtual reality player plays (left). While the virtual reality player moves along the corridor, so does the augmented reality representation in the viewer perspective. Accordingly, when the firefighter extinguishes the flames, the fire disappears from viewer sight. Finally, the test ends when the firefighter approaches the freed victim and leads him out of the building.

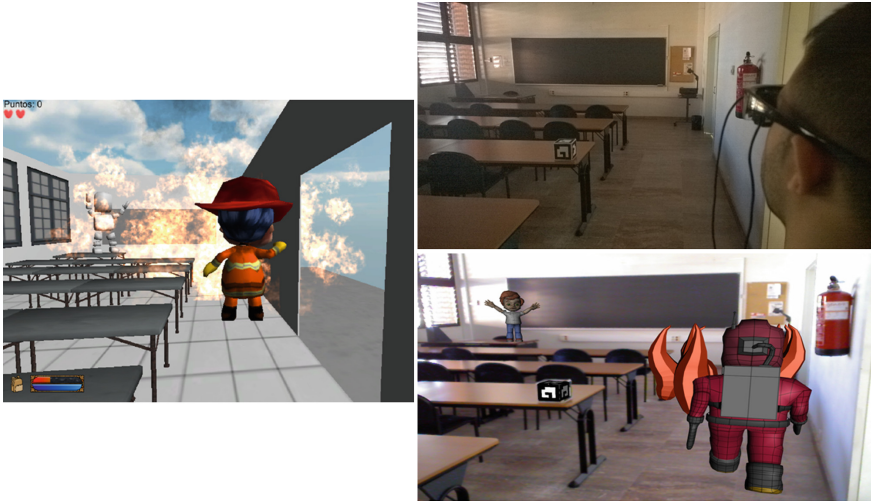


Fig. 6. Virtual view (left), observer direct view (upper-right) and augmented view (lower-right)

5 Conclusions and Future Work

Mixed reality technology offers the possibility to take advantage of the physical characteristics of a real environment that virtual reality systems fail to capture or simulate. These advantages can be exploited in the form of mixed reality DEGs, which allow to combine the flexibility provided by virtual DEGs and the benefits of delivering instructions in a real context. Unfortunately, the technical knowledge required to create this type of experiences often poses an unsurmountable barrier for educators. This work has presented MR-GREP, a platform that supports the design and development of mixed reality DEGs with minimal programming involved. The platform is able to reproduce the action of the game in a real context by means of AR technology.

Current work is carried out with the aim of extending the possibilities of interaction supported by the mixed reality environments created by the platform. On the one hand, a wearable gesture recognition system for the players in the real world is currently under development. The system will allow to translate the gestures and movements of the players into changes of their representations in the virtual world. On the other hand, work is being carried out to integrate the platform with an interaction toolkit for developing cross-reality experiences. This will allow to interconnect the physical and virtual actions in the game with changes and readings in real objects placed in the scenario of the game.

References

1. Johnson, L., Adams, S., Cummins, M.: The NMC horizon report: 2013 higher education edition. New Media Consortium (2013)

2. Torrente, J., Del Blanco, Á., Marchiori, E.J., Moreno-Ger, P., Fernández-Manjón, B.: < e-Adventure > : introducing educational games in the learning process. In: 2010 IEEE Education Engineering (EDUCON), pp. 1121–1126. IEEE (2010)
3. Göbel, S., Salvatore, L., Konrad, R.A., Mehm, F.: StoryTec: a digital storytelling platform for the authoring and experiencing of interactive and non-linear stories. In: Spierling, U., Szilas, N. (eds.) ICIDS 2008. LNCS, vol. 5334, pp. 325–328. Springer, Heidelberg (2008)
4. Thomas, B., Close, B., Donoghue, J., Squires, J., De Bondi, P., Morris, M., Piekarski, W.: ARQuake: an outdoor/indoor augmented reality first person application. In: The Fourth International Symposium on. Wearable Computers, pp. 139–146. IEEE (2000)
5. Cheok, A.D., Goh, K.H., Liu, W., Farbiz, F., Fong, S.W., Teo, S.L., Li, Y., Yang, X.: Human Pacman: a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing. *Pers. Ubiquitous Comput.* **8**, 71–81 (2004)
6. MacIntyre, B., Gandy, M., Dow, S., Bolter, J.D.: DART: a toolkit for rapid design exploration of augmented reality experiences. In: Proceedings of the 17th Annual ACM symposium on User Interface Software and Technology, pp. 197–206. ACM (2004)
7. Zauner, J., Haller, M.: Authoring of mixed reality applications including multi-marker calibration for mobile devices. In: 10th Eurographics Symposium on Virtual Environments, EGVE, pp. 87–90 (2004)
8. Brederode, B., Markopoulos, P., Gielen, M., Vermeeren, A., de Ridder, H.: Powerball: the design of a novel mixed-reality game for children with mixed abilities. In: Proceedings of the 2005 Conference on Interaction Design and Children, pp. 32–39. ACM (2005)
9. ARToolKit Home Page. <http://www.hitl.washington.edu/artoolkit/>
10. Game Engine Technology by Unreal. <https://www.unrealengine.com/>
11. Zarraonandia, T., Diaz, P., Aedo, I., Ruiz, M.R.: Designing educational games through a conceptual model based on rules and scenarios. *Multimedia Tools and Appl.* 1–25 (2014)