Design Patterns for Augmented Reality Learning Games

DOI:
10.1007/978-3-319-71940-5_15

Document status and date:
Published: 01/12/2017

Document Version:
Peer reviewed version

Design Patterns for Augmented Reality Learning Games

Felix Emmerich[1], Roland Klemke[1,2], Thomas Hummes[1]

[1] Mediadesign Hochschule, Faculty of Gamedesign, Düsseldorf, Germany
[2] Welten Institute, Open University of the Netherlands, Heerlen, Netherlands
fnemmerich@gmail.com, Roland.Klemke@ou.nl, Thomas.Hummes@gmail.com

Abstract. Augmented Reality (AR) is expected to receive a major uptake with the recent availability of high quality wearable AR devices such as Microsoft's Hololens. However, the design of interaction with AR applications and games is still a field of experimentation and upcoming innovations in sensor technology provide new ways. With this paper, we aim to provide a step towards the structured use of design patterns for sensor-based AR games, which can also inform general application development in the field of AR.

Keywords. Augmented Reality, Game-design Patterns, Interaction Patterns, Learning Games

1 Introduction

"The convergence of mobile computing and wearable computing with augmented reality is naturally of great interest to interaction designers" [1], while "the convergence of wearable computing, wireless networking and mobile AR interfaces" is bringing "a new breed of computing called 'augmented ubiquitous computing'" [2].

Augmented Reality (AR) can be defined as "the fusion of any digital information with physical world settings, i.e. being able to augment one's immediate surroundings with electronic data or information, in a variety of formats including visual/graphic media, text, audio, video and haptic overlays" [3]. Features, "most of which are present in most AR systems" [4] are listed as: "Sense properties about the real world; process in real time; output (overlay) information to the user; provide contextual information; recognize and track real-world objects; be mobile or wearable" [4].

Examples of AR systems utilizing senses other than sight include [5], whose application for cultural sciences students' field trips focused on audio augmentation, arguing that "just like a user should – while driving a car – use sight as much as possible to drive, we believe that with location based learning, a learner's eyes must be primarily used to examine the environment". Haptic feedback corresponding to virtual objects may be transferred from Virtual Reality to AR [6].

Location-based AR outputs information based on the user's position [3, 7, 8]. Points of Interest (POI) are defined and associated with virtual assets – "when a user […] explores a space the POIs are revealed and the content can be accessed" [8]. *Vision-based* AR functions by using computer vision techniques to identify and track patterns known as *fiducials* (visual markers [9]) in the environment [3, 8]. Both of these approaches have their advantages and disadvantages: Fiducials can only be used with systems trained to recognize them and if conditions like inadequate lighting do

Preprint version. Cite this paper as: Emmerich F., Klemke R., Hummes T. (2017) Design Patterns for Augmented Reality Learning Games. In: Dias J., Santos P., Veltkamp R. (eds) Games and Learning Alliance. GALA 2017. Lecture Notes in Computer Science, vol 10653. Springer, Cham

not interfere. Location-based systems can suffer from inaccuracy or loss of tracking [10].

A way to combine the advantages of both approaches may lie in hybrid systems as described by [11] or image understanding [12]. The Microsoft HoloLens utilizes a depth camera and tracks head movements through various sensors. A technique called "spatial mapping" [13] is used to construct a three-dimensional model of the surroundings and display virtual content at the appropriate coordinates.

Design patterns for general interaction [14] and games exist [15] and high-level patterns for Mixed Reality games have been proposed [16]. We aim to close the gap between low-level interactions in sensor-based wearable AR systems and high-level game-design patterns for learning games by providing a framework of design patterns for AR games. While such a framework can generally be applied to all kinds of AR games, our main target is to guide the construction of AR learning games. However, pedagogical in relation to the design patterns are not in the focus of this research. Instead, here our focus is more on interactivity and visualization. Definitions, approaches, potentials and limitations of AR are presented, followed by our framework and its first prototypical implementation.

2 Augmented Reality for Learning games

AR has been applied to many domains, including "hands-free instruction and training, language translation, obstacle avoidance, advertising, gaming, museum tours, and much more" [4], maintenance and repair [17], or Big Data visualization [18], where AR "might solve many issues from narrow visual angle, navigation, scaling, etc.".

Games are an application particularly well-suited for the medium of AR, as "augmented reality is an active, not a passive technology" [7], which emphasizes the "dialogue between the media and the context in which it is used" [3]. Although commercial AR games can be said to go back as far as 2003's EyeToy [10], efforts were for a long time focused on research, until the advance of smartphone technology, which made devices with AR capabilities widely available [16].

Pokémon GO [19], an AR game based on both, the well-known Pokémon franchise and *Ingress* [20], is a rare example of a mobile AR game with a large player base. In the field of learning games [21] reports that Mixed Reality games offer the opportunity to "sense and feel being 'someone' else", while first person experiences make it challenging to develop empathy. *Locatory* is an AR adaptation of the game *Memory*, requiring players to find virtual cards spread around the environment and then match them to real landmarks, to foster orientation skills [22].

Game design patterns have been used to map cognitive and affective learning outcomes in AR games for learning [23]. Similarly, a recent literature review identifies three design principles for learning-oriented AR – "enable and then challenge," "drive by gamified story," and "see the unseen" [24].

Knowledge about how to best approach the design of AR games is still lacking [16], a sentiment [25] shares: "Little is known on how to systematically apply game-design patterns to augmented reality". Similarly to these, [24] attempts to extrapolate design guidelines from the AR game *Dino Dig*, which despite having educational content was primarily intended to entertain.

3 Design Patterns for AR-based Games

Design patterns describe precisely how to use design techniques in order to achieve certain positive effects, at the same time providing insight and creating a shared vocabulary in the form of a pattern language [16, 26]. More precisely, design patterns "express a relationship between particular design contexts, forces […], and desired ('positive' or good) features" [26].

Björk and Holopainen [27] collected game design patterns, concerned with idea generation. General characteristics of patterns can be outlined as: "Operational and precise"; "positive"; "flexible"; "debatable (the Pattern is clear enough to criticize)"; "testable"; "end-user oriented." [26]. A well-defined game design pattern language would allow for efficient communication, documentation and analysis "e.g. for purposes of comparative criticism, re-engineering, or maintenance" [28].

The patterns collected by Björk & Holopainen [27] do not utilize a problem-solution approach, with Björk, Lundgren, & Holopainen arguing that "not all aspects of design can or should be seen as solving problems, especially in a creative activity such as game design" [15]. The Game Ontology Project describes, analyzes and studies games with pattern-like entries existing in a hierarchy the top level of which includes interface, rules, entity manipulation, and goals [29]. More on the pedagogical side [30] define a framework for the construction of learning games.

The literature revealed only a few pattern approaches for the domain of AR, mainly as *interaction patterns*. The examples below are presented informally but fit characteristics from [26]. They provide data, which the framework presented in this paper was able to expand on and have thus been included.

The *Point Of Interest* (POI) interaction pattern is often implemented in mobile AR browsers. When arriving at pre-defined points, users receive information about the environment through a choice of channels [5]. Browsers may also direct the user towards nearby points of interest [3]. The *Head-Up Display* (HUD) presents information from a fixed point of view, i.e. the information is not assigned some coordinate in 3D space [1]. The *Tricorder* interaction pattern refers to scenarios in which information is scanned from the environment, adding "pieces of information to an existing real-world experience" [1]. *Holochess* experiences consist of presenting entirely virtual objects to the AR environment. *X-Ray Vision*-based experiences allow "seeing beneath the surface of objects, people, or places" [1].

Design patterns for mobile games have been mapped to cognitive and motivational effects in educational AR games [31]. A short, preliminary list of patterns "which take advantage of AR potential" comprising names and short descriptions consists of: *Localization*, *video recording and view sharing*, *synchronous communication*, *contextualization*, and *object recognition* [25]. There are still challenges for AR to overcome, which inform the framework for interactions in AR in the remainder of this paper, applying a pattern approach, incorporating elements from the various sources discussed above, while adhering to the general characteristics laid out by [26].

4 A Pattern-based Framework for AR Learning Games

Our pattern-based framework, mirroring above-mentioned approaches to game design patterns, is a classification of possible interactions, akin to the game mechanic terminology.

Method. The framework and the software development are based on design patterns, which we defined according to the various approaches presented above [14, 16, 25–28]. Technologically, we build on the comparison of available AR systems and sensors performed by [32]. In this first iteration of the framework, the pattern elements that were used in at least three of the six papers are present. They are:

Name: A succinct name for the pattern.

Forces/Problem: The issues the pattern is intended to combat.

Feature/Solution: A description of one way to solve the problem.

Effects/Consequences: The positive and negative consequences of applying the pattern, including design choices required for implementing the pattern.

Requirements: We introduced requirements, which must or may be met to implement a pattern. This allows game designers interested in implementing patterns to ascertain whether a given pattern fits their criteria.

Scope. Challenges to AR can roughly be sorted into those pertaining to technology, user interface and social acceptance [33]. Due to the scope of this paper and the framework's focus on the interactive medium of games, of these only user interfaces – visualization and interaction – will be covered. Additionally, some patterns focus on the development side of AR applications. The content of the patterns listed below is derived from the literature mentioned, a brainstorming session with participants of the WEKIT project [34], and the characteristics of existing AR games acquired through play testing. We grouped the patterns into six groups: directional, environmental, input, non-visual feedback, media-related, and multi-user, displayed in table 1.

Table 1. Basic Patterns

	Pattern	Forces/Problem	Feature/Solution	Effects/Consequences	Requirements
Directional	Directed Gaze	Direct a user's attention to something when they have full control over their view.	Use icon to direct user attention. Icon points to the object of interest, if it is not currently visible.	Structure AR experience, guide user. Combinable with *Gaze Cursor*. May obstruct other elements and cause screen clutter. Multiple focus points require *Inf. Filtering*.	System needs awareness of the focus point position relative to user location and head rotation.
	Directed Move-ment	Applications may require the user to move to certain locations.	Display icon at the target location. Icon points to the target location, if it is not currently visible.	Structures AR experience and guides users. May obstruct other elements and cause screen clutter. Multiple focus points require *Information Filtering*.	System needs awareness of the focus point position relative to user location and head rotation.
Environmental	Environ-ment-Adapta-tion	Design a game that can be run anywhere, while taking the environment into account	Make the game automatically adapt to the characteristics of the environment.	The game will run in many environments while making use of AR. Games cannot be planned as stringently and different users may have different experiences.	System needs awareness of surrounding. Depth sensors, cameras may be sufficient.
	Environ-ment-Inde-pen-dence	How can you develop AR games for unknown environments?	Make the game not interact with the environment.	Games will function more reliably and predictably. User enjoyment may be reduced due to missing AR experience.	Games should function without extra sensors.
	Environ ment Require ments	Game interacts with different environments without changing the game itself.	Make the game's environmental requirements explicit.	Ensures that AR game works as intended. Possibly requires user to make changes to their environ-ment to play the game.	Requirements must be explained and checked.

	Point of Interest (POI)	Provide information to users based on location. Bind information to locations not accessible..	Bind information to locations, making it available upon getting within a certain range and allowing to direct users to such points.	Multiple POIs should not overlap or the system must handle overlap. Avoid screen clutter. Precision is low if only location data is used – may be combined or replaced with a *Gaze POI*.	Requires location technology.
Input-related	Gaze Cursor	How do you select the object to apply actions to?	Base actions on gaze, indicated with visual cursor. Select objects, complete actions (e.g. based on gaze duration).	Guide interactions. Inaccurate or lagging cursors may mislead. Cursor might obscure object of interest. Tunnel vision may be provoked.	Awareness of the user's gaze required .
	Gaze Point of Interest (Gaze POI)	How can events in AR be triggered?	Perform actions when the player looks at specific objects (real or virtual). Events may indicate when something leaves or enters field of view.	Focus on important information, context-sensitive information. Combine with *Gaze Cursor* to show selected objects. In larger areas, *POI* may be used. Unintended events may be confusing.	Gaze direction and environmental model required.
	Gesture-based Inter-action	Interact intuitively with AR environment.	Allow the manipulation of objects through gestures.	Without special input devices, AR systems may be more immersive and intuitive. Non-intuitive or accidental inputs can be frustrating.	Hand movement tracking and gesture recognition required.
	Voice Com-mands	How do you allow user input while keeping the user's hands free?	Enables the user to perform actions by speaking appropriate phrases.	Can enhance user experience. Simple phrases to avoid frustration. May face acceptance issues in shared spaces.	Requires one or more microphones.
Non-Visual	Haptic Feed-back	Receive feedback when "touching" augmented objects.	Give feedback via the haptic sense. Users may be more responsive to non-visual feedback.	Feedback device can limit freedom of movement. Inappropriate feedback may break immersion.	Additional feedback channels via feedback devices
Media-Related	Auto-Play	Trigger events in AR without requiring or despite varying kinds of user input.	Automatically starting, pausing, and/or stopping events (e.g. based on location/gaze).	Help to avoid tunnel vision and to increase pleasure. Automatic warnings can prevent injury or incorrect use. Can confuse or cause screen clutter. Needs way to stop unwanted content.	*Auto-Play* may be based on position, gaze, orientation, audio input, or fiducials.
	Infor-mation Filtering	Screen clutter by too much information. Too little reduces usefulness.	Filter information e.g. by distance, angle, or relevance-based information.	Makes systems easier to use. Ensure that the filter reflects the intention to avoid too little or too much information.	Required sensors depend on the exact filter approach.
	Ob-scured Infor-mation Visuali-zation	Keep track of content that is hidden behind other objects, real or virtual, at a given time.	Visualize an object even, or specifically, if it is obscured (various approaches).	Can afford the user a better understanding of their (augmented) environment. However, inappropriate approaches may cause depth perception issues.	Tracking of real and AR objects, tracking head and environments.

Multi-User	Shared Pointer	How can multiple people in an AR environment communicate efficiently?	Use (gaze) cursors, also visible to other players or allow users to leave markers at set points.	Allow users to communicate easily. Requires directed user input (e.g. *Gaze Cursor*). See *Directed Gaze* and *Directed Movement*.	Positions and orientations of users must be tracked and synch"d in real-time.

5 Combined patterns for AR Learning Games

The idea of the framework described above is twofold:

(1) We aim to work towards a best practice collection of patterns, which enable players of AR learning games to intuitively understand interaction mechanisms. This aspect has been presented in the review of existing AR games and the patterns found within, which found its place in the effects/consequences column of each pattern.

(2) We want to provide a design toolbox, enabling game designers and developers to construct complex games utilising and combining the basic patterns described. This section is about the second aspect. In line with [27], we see our pattern collection as part of a more general game design language. In this sense, not only should the patterns described here be combinable with each other, but also should a game designer be enabled to combine these patterns with patterns from other collections, such as those reviewed above. Consequently, we see the examples of proposed pattern combinations presented below as a demonstration of the possible use of our framework (table 2).

Table 2. Combined Patterns usable for learning games

Pattern	Forces/Problem	Feature/Solution	Effects/ Consequences
Exten-ded Room	The gameplay may require additional virtual rooms, spaces, and structures.	*Obscured Information Visualization*, *Environment Requirements*, and *Environment Adaptation* enable virtual objects to overlay physical structures (e.g. showing non-existing rooms). "Magic Doors" allow to virtually enter a different room.	Useful e.g. for historic learning (what did a scene look like previously) or for change planning (what might it look in future). Also useful for virtual travels.
Explo-ration and Search	Virtual Objects should be hideable behind/under physical objects to force the player to search them within the physical environment.	Based on *Obscured Information Visualization*, *Information Filtering*, *Environment Requirements*, *Environment Adaptation* from this collection and *Exploration*, *Clues*. Optional hints can be used for guidance.	Explorative learning tasks to foster orientation, exploration strategy
Asym-metric Multi-player	Two players require different information, e.g. according to individual roles, locations, or progress.	Based on *Information Filtering*, *Shared Pointer* from this collection and *Asymmetric Information* information can be individualized.	Asymmetric information can be bound to teacher/ learner roles or to reflect individual learning needs
Aug-mented Ghost Track	A player can see and follow the track of another player in the physical environment	*Shared Pointer*, *Information Filtering*, *Directed Gaze*, *Directed Movement* can be combined to create ghost tracks.	A learner can follow guided steps or work on continuous improvement of own ghost track data.
X-Ray Vision	Visualizing internal processes or mechanisms not visible	*Obscured Information Visualization*, *Environment Adaptation* and *Gaze POI* are the basis for showing hidden	Explain hidden features of complex setups, can be extended to allow for

to the eye.	parts	interactivity and animation

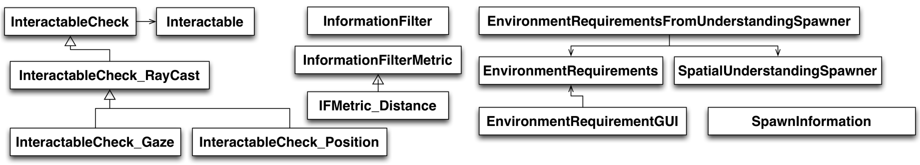

Figure 1: Pattern framework classes

We implemented a prototype based on concepts of the open source ARLearn system for mobile serious games [35], adapted for HMDs as a proof of concept. The patterns have been developed for the HoloLens, using the Unity game engine and the HoloToolkit [36]. Some of the patterns (Directed Gaze, Directed Movement, Gaze Cursor, Gesture-based Interaction, and Voice Commands) are already available in HoloLens, others were newly implemented in our prototype (figure 1, table 3).

Interactable, *InteractableCheck_Gaze*, and *InteractableCheck_Position* together form the basis for *Point of Interest* and *Gaze Point of Interest*. The scripts inheriting from *InteractableCheck* provide different ways of detecting *Interactable* objects. The *EnvironmentRequirements* class implements the pattern of the same name. Requirements consist of an environmental feature, an amount, and whether the amount represents an upper or lower limit. This data is compared to that gathered by the HoloLens to assess if the requirements defined on application level are met. *EnvironmentRequirementGUI* represents one way of visualizing the available information.

The spatial understanding features of the device are further utilized in *SpatialUnderstandingSpawner*, an implementation of Environment-Adaption which uses predefined sets of rules and constraints from *SpawnInformation* to find suitable spots for instantiating objects, e.g. on a wall or on a floor, far from the player. *EnvironmentRequirementsFromUnderstandingSpawner* provides a bridge between the two previous mechanisms by generating simple requirements from the parameters of a *SpatialUnderstandingSpawner*.

In first internal usability tests we could show the general functionality and operability of the patterns. In a next step, we aim to integrate the patterns into the general expertise-training framework of the WEKIT architecture [34].

Table 3. Implemented Patterns

Pattern	Implementation	Comments
Point of Interest, Gaze POI	*Interactable*, *InteractableCheck_Gaze*, and *InteractableCheck_Position*	The scripts inheriting from *InteractableCheck* provide different ways of detecting *Interactable* objects.
Environment Requirements	*EnvironmentRequirements* *EnvironmentRequirementGUI* represents one way of visualizing the available information	Consist of an environmental feature and an upper or lower limit. This is assessed against data from HoloLens.
Environemnt Adaptation	Spatial understanding features are utilized in *SpatialUnderstandingSpawner*	Uses predefined sets of rules and constraints from *SpawnInformation* to find spots for instantiating objects, e.g. on a wall/floor, far away.
Information Filter	*InformationFilter* executes tasks based on data from *InformationFilterMetric*-derived classes, such as *IFMetric_Distance* (distance between object and player).	The Information Filtering pattern performs actions according to different levels of proximity to the user, but can be extended to cover other metrics.

6 Conclusions

We presented an overview of AR definitions, approaches, and applications. We highlighted approaches towards specifying design patterns and created a framework of design patterns for AR games, provided ideas towards the construction of games based on these patterns, and exemplarily adapted a sample of them for Microsoft's Hololens using the Unity game engine.

The work reported here can be expanded in several ways. The framework only covers a fraction of AR interactions, as the scope was limited to user interaction and usability with the Microsoft HoloLens, and it is likely not complete. The results can be seen as a proof of concept. Although focused on learning games, the framework can be used in other AR contexts such as commercial applications, learning applications, or simulations, which aim to make use of AR. Consequently, we see the main contribution of this paper to be a step towards broadening the understanding of AR interaction and application development based on design patterns. As next steps, we aim to apply the framework to the WEKIT AR training solution and to evaluate in pilot application cases related to aircraft maintenace, medical equipment operations, and space craft subsystem integration [34]. Based on these evaluation, we aim to further develop the framework to cover a broad range of AR use cases and interaction scenarios.

Acknowledgments. Parts of this work were supported by the European Commission under the Horizon 2020 Programme under grant agreement No 687669 (WEKIT).

References

1. Lamantia, J.: Inside Out: Interaction Design for Augmented Reality.
2. Papagiannakis, G., Singh, G., Magnenat-Thalmann, N.: A survey of mobile and wireless technologies for augmented reality systems. (2008).
3. FitzGerald, E., Ferguson, R., Adams, A., Gaved, M., Mor, Y., Thomas, R.: Augmented reality and mobile learning: the state of the art. Int. J. Mob. Blended Learn. 5, 43–58 (2013).
4. Calo, R., Denning, T., Friedman, B., Kohno, T., Magassa, L., McReynolds, E., Newell, B.C., Roesner, F., Woo, J.: Augmented Reality: A Technology and Policy Primer. (2015).
5. Ternier, S., De Vries, F., Börner, D., Specht, M.: Mobile augmented reality with audio, supporting fieldwork of Cultural Sciences students in Florence. In: Eleftherakis, G., Hinchey, M., and Holcombe, M. (eds.) Software Engineering and Formal Methods - Proceedings of 10th International Conference, SEFM 2012. pp. 367–379. Springer (2012).
6. Benko, H., Holz, C., Sinclair, M., Ofek, E.: NormalTouch and TextureTouch : High-fidelity 3D Haptic Shape Rendering on Handheld Virtual Reality Controllers. In: Proceedings of the 29th Annual Symposium on User Interface Software and Technology. pp. 717–728. ACM (2016).
7. Johnson, L., Smith, R., Willis, H., Levine, A., Haywood, K.: The 2011 Horizon Report. The New Media Consortium, Austin, Texas (2011).
8. Munnerley, D., Bacon, M., Wilson, A., Steele, J., Hedberg, J., Fitzgerald, R.: Confronting an augmented reality. Res. Lerning Technol. 20, 39–48 (2012).
9. You, S., Neumann, U.: Fusion of Vision and Gyro Tracking for Robust Augmented Reality Registration. (2001).
10. Wetzel, R., McCall, R., Braun, A.-K., Broll, W.: Guidelines for Designing Augmented Reality Games. In: Proceedings of the 2008 Conference on Future Play: Research, Play, Share. pp. 173–180 (2008).

11. Schall, G., Wagner, D., Reitmayr, G., Taichmann, E., Wieser, M., Schmalstieg, D., Hofmann-Wellenhof, B.: Global Pose Estimation using Multi-Sensor Fusion for Outdoor Augmented Reality. In: Proceedings of the 2009 8th IEEE International Symposium on Mixed and Augmented Reality. pp. 153–162 (2009).
12. Furmanski, C., Azuma, R.T., Daily, M.: Augmented-reality visualizations guided by cognition: Perceptual heuristics for combining visible and obscured information. In: Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR'02). IEEE (2002).
13. Spatial mapping, https://developer.microsoft.com/en-us/windows/holographic/spatial_mapping.
14. Borchers, J.O.: A Pattern Approach to Interaction Design. AI Soc. 15, 359–376 (2001).
15. Björk, S., Lundgren, S., Holopainen, J.: Game Design Patterns. In: Level Up: Digital Games Research Conference 2003 (2003).
16. Wetzel, R.: A Case for Design Patterns supporting the Development of Mobile Mixed Reality Games. Found. Digit. Games. (2013).
17. Henderson, S.J., Feiner, S.: Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turret. In: IEEE International Symposium on Mixed and Augmented Reality 2009. pp. 135–144. IEEE (2009).
18. Olshannikova, E., Ometov, A., Koucheryavy, Y., Olsson, T.: Visualizing Big Data with augmented and virtual reality: challenges and research agenda. J. Big Data. 2, (2015).
19. Niantic: Pokémon GO. Game [Android]. (6 July 2016). Niantic, San Francisco, CA. Played August 2016., (2016).
20. Niantic: Ingress. Game [Android]. (14 December 2013). Niantic, San Francisco, CA. Played July 2016., (2013).
21. Kors, M.J.L., Ferri, G., van der Spek, E.D., Ketel, C., Schouten, B.A.M.: A Breathtaking Journey. Proc. 2016 Annu. Symp. Comput. Interact. Play - CHI Play '16. 91–104 (2016).
22. Specht, M., Ternier, S., Greller, W.: Dimensions of Mobile Augmented Reality for Learning: A First Inventory. J. Res. Cent. Educ. Technol. 7, 117–127 (2011).
23. Schmitz, B., Klemke, R., Specht, M.: Mobile gaming patterns and their impact on learning outcomes : A literature review. In: Ravencroft, A., Lindstaedt, S., Kloos, C.D., and Hérnandez-Leo, D. (eds.) 21st Century Learning for 21st Century Skills - Proceedings of the 7th European Conference of Technology Enhanced Learning, EC-TEL 2012, Lecture Notes in Computer Science. pp. 1–6. Springer, Saarbrücken, Germany (2012).
24. Dunleavy, M.: Design Principles for Augmented Reality Learning. TechTrends. 58, 28–34 (2014).
25. Antonaci, A., Klemke, R., Specht, M.: Towards Design Patterns for Augmented Reality Serious Games. In: The Mobile Learning Voyage - From Small Ripples to Massive Open Waters. pp. 273–282 (2015).
26. McGee, K.: Patterns and Computer Game Design Innovation. In: Proceedings of the 4th Australasian conference on Interactive entertainment. RMIT University (2007).
27. Björk, S., Holopainen, J.: Patterns in Game Design. (2005).
28. Kreimeier, B.: The Case For Game Design Patterns, (2002).
29. Zagal, J.P., Mateas, M., Fernández-Vara, C., Hochhalter, B., Lichti, N.: Towards an Ontological Language for Game Analysis. In: Proceedings of DiGRA 2005 Conference. pp. 3–14 (2005).
30. Carvalho, M.B., Bellotti, F., Berta, R., De Gloria, A., Sedano, C.I., Hauge, J.B., Hu, J., Rauterberg, M.: An activity theory-based model for serious games analysis and conceptual design. Comput. Educ. 87, 166–181 (2015).
31. Schmitz, B., Klemke, R., Specht, M.: An Analysis of the Educational Potential of Augmented Reality Games for Learning. In: Specht, M. and Multisilta, J. (eds.) Proceedings of the 11th International Conference on Mobile and Contextual Learning 2012. pp. 140–147. , Helsinki, Finland (2011).
32. Sharma, P., Wild, F., Klemke, R., Helin, K., Azam, T.: D3.1 Requirement analysis and sensor specifications – First version, (2016).
33. Azuma, R.T., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B.: Recent

advances in augmented reality. IEEE Comput. Graph. Appl. 21, 34–47 (2001).
34. Klemke, R., Limbu, B., Rasool, J.: WEKIT Framework & Training Methodology – First version. (2016).
35. Ternier, S., Klemke, R., Kalz, M., Specht, M.: ARLearn: augmented reality meets augmented virtuality. J. Univers. Comput. Sci. - Technol. Learn. across Phys. virtual spaces [Special issue]. 18, 2143–2164 (2012).
36. Microsoft: HoloToolkit, https://github.com/Microsoft/HoloToolkit.