

VISUAL ENVIRONMENT FOR DESIGNING INTERACTIVE LEARNING SCENARIOS WITH AUGMENTED REALITY

José Miguel Mota¹, Iván Ruiz-Rube¹, Juan Manuel Dodero¹ and Mauro Figueiredo²

¹University of Cadiz, Spain

²University of Algarve, Faro, Portugal

ABSTRACT

Augmented Reality (AR) technology allows the inclusion of virtual elements on a vision of actual physical environment for the creation of a mixed reality in real time. This kind of technology can be used in educational settings. However, the current AR authoring tools present several drawbacks, such as, the lack of a mechanism for tracking the students' activities, the capability of detecting collisions among virtual objects, allowing establish only one-to-one relationships between trackers and virtual objects, etc. In this paper, we present VEDILS, a visual tool based on the MIT App Inventor 2 environment for designing interactive learning scenarios that include Augmented Reality (AR) resources and can be deployed on Android devices. We have extended the App Inventor block language to include AR resources and obtain information about users' interaction with such resources. Furthermore, a use scenario consisting in the development of an example of learning scenario for Engineering students is included. With this prototype students can visualize 3D models of the mechanical parts with augmented reality in a mobile device, providing a better perception of the model 3D shape and improving the ability of making the 2D orthographic views and perspectives that they study in the first year of mechanical engineer.

KEYWORDS

Augmented reality, authoring tools, mobile learning, visual programming, App Inventor.

1. INTRODUCTION

AR technology is the inclusion of virtual elements on a vision of actual physical environment for the creation of a mixed reality in real time. This complements and enhances the perception our senses (sight, hearing and touch) have in the real world. AR provides various degrees of immersion and interaction that can help engage students in learning activities. As demonstrated Ángela Di Serio at (Di Serio et al., 2013), motivational factors of attention and satisfaction in an augmented-reality-based learning environment were better rated than those obtained in a slides-based learning environment. Azuma (Azuma, 1997) presented at 1997 a first survey which classifies possible fields of use of AR, such as medicine, manufacturing and repair, annotation and visualization, robot path planning, cultural expression, entertainment, military aircraft, etc.

The academic world is not immune to these initiatives and has started to introduce the AR technology in some of its disciplines (Krevelen and Poelman, 2010). Also, advances in mobile technology make it feasible to use AR technology for learning (Specht et al., 2011). However, the knowledge and the applicability of this technology in teaching are minimal. Some authors (Martin et al., 2011) state that the principal barrier is the need for having programming skills to develop AR experiences. Another of the reasons, according to Martin (Martin et al., 2011), is the lack of tools specifically designed for education, rather than for general purpose.

In this paper we present the foundations of an authoring tool, called Visual Environment for Designing Interactive Learning Scenarios (VEDILS), which enables users to develop ubiquitous, contextual and interactive learning scenarios based on AR. This tool tries to facilitate the inclusion of AR technologies in educational contexts. Instead of creating a new specific tool, which may limit the choices for the teachers experienced with programming, we have based our approach on MIT's App Inventor 2 (MIT App Inventor 2016), which is an easy-to-use online environment for developing Android apps. This democratizes mobile programming through its simple blocks language. App Inventor has been extended with several components for managing the device's camera, trackers, such as markers or visual objects, and virtual assets, such as 2D images or 3D models to be visualized. All of these components provide several visual programming blocks

that can be used along with the rest of built-in blocks of App Inventor to develop the logic of the learning scenarios.

The research methodology for this work is a combination of the design and creation strategy and the case study strategy, which is suitable for research projects where the computing application is the main focus (Oates, 2005). The design and creation must be combined with another strategy when the computing application is a vehicle for something else. In this case, the research issue is to examine how existing AR environments support the educational settings, and test the contribution of visual languages. Then, case study research is an adequate strategy to complete the design and creation approach. The rest of this paper is organized as follows: Section 2 presents the context of the present research, focusing on the role of the AR for educational contexts and the findings of a survey performed about AR technologies. Then, in Section 3, the VEDILS authoring tool is described. Section 4 presents a use scenario consisting in the development of a learning scenario for Engineering students from the University of Algarve. Finally, in the last section, conclusions and directions for future research are presented.

2. CONTEXT

2.1 Technology-Enhanced Learning and Augmented Reality

In recent years, technology-enhanced learning (TEL) research has increasingly focused on emergent technologies, such as AR, ubiquitous learning (u-learning), mobile learning (m-learning), serious games and learning analytics for improving the satisfaction and the experiences of the users in enriched multimodal learning environments (Johnson et al., 2014). These researches take advantage of technological innovations in hardware and software for mobile devices and their increasing popularity among people. The significant development of user modeling and personalization processes place the student at the center of the learning process. Research on AR has also demonstrated its extreme usefulness for increasing the student motivation in the learning process (Liu and Chu, 2010), (Di Serio et al., 2013), (Jara et al., 2011), (Bujak et al., 2013) and (Chang et al., 2014).

AR applications combine virtual objects (such images on 2-D or 3-D, sound, video, text, etc.) with a real environment in real time. Virtual computer generated and real objects appear together in a real time system in a way that user sees the real world and the virtual objects superimposed with the real objects. The user's perception of the real world is enhanced and the user interacts in a more natural way. The virtual objects can be used to display additional information about the real world that are not directly perceived (Ibáñez et al., 2014). Paul Milgram and Fumio Kishino (Milgram and Kishino, 1994) introduced the concept of a Virtuality Continuum classifying the different ways that virtual and real objects can be realized. In this taxonomy scheme AR is closer to the real world. Another commonly accepted definition was crafted by Ronald Azuma (Azuma, 1997) explaining AR as a technology that combines real and virtual worlds, real-time interactive and recorded in 3D.

In general, AR applications fall in two categories: Geo-based and computer vision based. Geo-based applications use the mobile device's GPS, accelerometer, gyroscope, and other technology to determine the location, heading, and direction of the mobile device. The user can see overlapping computer-generated images onto a real world in the direction he is looking. However, this technology has some problems. The major problem is imprecise location, especially indoors, which makes it difficult for the creation of photo overlays. Computer vision based applications use image recognition capabilities to recognize images and overlay information on top of this image. Inside this category we can find two different options: marker-based AR, marker-less AR. Markers are labels that contain a colored or black and white pattern that is recognized or registered by the AR application through the camera of the device in order to fire an event that can be, for instance, to show a 3D image in the screen of the device located in the same position where the marker is. Marker-less AR is based on the recognition of the object's shapes.

According to (Bacca et al., 2014), AR applications can be used on different learning scenarios, such as explaining and evaluating topics, simulating lab experiments, educational games, augmenting information, exploring, etc. Furthermore, these can be applied into different field of education, mainly in Science, Humanities&Arts and Engineering. This research reveals that most of the AR applications developed for educational settings use marker-based technology. A possible explanation for this result is that currently the

tracking process of marker-based techniques is better and more stable compared to the marker-less tracking techniques. However, the use of marker-based AR requires that the existing educational materials must be reformed to include the required markers, while the option of marker-less technology allows to reuse existing material, so this recognition system should be a required option when it comes to choose an AR tool.

2.2 Augmented Reality Authoring Tools

There are different tools for working with RA capabilities. The lowest level approaches provide complex toolkits, libraries and scripting frameworks for application developers. Examples of such tools are ARToolKit (ARToolKit 2016), ArUco (ArUco 2016), DroidAR (Droidar 2016), Wikitude SDK (Wikitude 2016) or Vuforia SDK (Vuforia 2016). These enable the development of powerful and tailored AR applications but they require a high knowledge of programming languages like Java, C#, C++ or JavaScript. On the other hand, there are a number of higher level graphical authoring tools available on Internet. These tools let non programmer users build AR scenes modifying parameters of the virtual objects, such as position, size, rotation or mark. Examples of this kind of tools are Layar (Layar 2016), Aurasma (Aurasma 2016), Augment (Augment 2016) and Aumentaty (Aumentaty 2016).

During the X Meeting of the Spanish teachers at different educational levels for the introduction of new technologies in teaching, Aulablog (Aulablog X Encuentro 2016), a survey on AR technologies was performed. Based on this survey, we conducted a study of the different tools used by teachers to create educational contents for their subjects. The distribution of the use of AR tools at different academic levels was: Aurasma (36.84 %), Layar (10.53%), Aumentaty (10.53%), QR Code (10.53%) and the rest of applications like Quiver, colAR, Chromville, Augment they obtained (5,26%). The main features of these tools can be found in Table 1.

In addition to the above tools, there are other high-level authoring tools for any type of AR applications targeting non- programmer users, such as DART (MacIntyre et al., 2004), ComposAR (Seichter et al., 2008), AMIRE (Haller et al., 2005) and MARS (Güven and Feiner, 2003) which are discussed briefly in (Wang and Langlotz, 2009). Some of these authoring tools supports both scripting and drag and drop interface. Higher level AR authoring tools address this need for interactivity. However, interactions are usually implemented in non-interpretive languages addressed through the XML parser, making it difficult to use by teachers. Also many of these tools do not consider the device on which they will be deployed, and thus do not use many of the features that mobile devices put at one's disposal. The main drawbacks found in all of these high-levels tools can be summarized into the following:

Table 1. Comparative between AR tools used by teachers

Software	Difficulty	Marker	Objects	Interaction	Description
Chromville, Quiver, colAr	Simple and Intuitive	Marker-less	Predefined 3D Objects	Interactions with objects	Very basic tool for primary education levels. It is not expandable; you can only use existing templates. No evaluation or recording allows interactions. No programming
Aurasma	Simple to use yet unclear the process of creating "auras"	Marker-less	2D, 3D and videos	Only in Design Mode	It allows you to create communities that share "auras". No evaluation or recording allows interactions. No programming
Layar	Easy to make an interactive document	Marker-less and Geolocation	2D, 3D and videos	Only in Design Mode	No evaluation or recording allows interactions. No programming
Aumentaty	Simple and intuitive interface	Marker-based	3D Objects and Text	Interactions with objects at design and real times	No evaluation or recording allows interactions. No programming. Problems when used on mobile devices
Augment	Simple and intuitive interface	Marker-less an QR	3D Objects	Interactions with objects at design and real times	No evaluation or recording allows interactions. No programming.

The main drawbacks found in all of these high-levels tools can be summarized into the following:

- The relationship between the markers and the virtual objects are always one-to-one.
- There are no responses to interactions (flip, rotate, etc.) with markers.
- Collisions among virtual objects can not be detected.
- Information about the user performing the activity (who, when, where, how, etc.) are not allowed to be collected.
- Assessing the acquired competences of the students by using the application is not possible.
- The lack of a mechanism for including use instructions in the application itself.

3. ENVIRONMENT FOR DESIGNING AR LEARNING SCENARIOS

In recent years software development tools have been created, which hide much of the complexity of the traditional programming languages in order to facilitate the development of new applications by people without programming skills. In this vein, we can note Scratch (Scratch 2016), a tool for developing videogames, and AppInventor, a tool for developing Android mobile applications. These two tools have in common that they use of a visual language based in blocks, which have to be assembled for designing the behaviour of the new applications. The popularity of this style of programming has been growing in recent years, and it has been used to create numerous blocks-based microworlds, including the popular Hour of Code (Hour of Code 2016) microworld.

In this point, we can state the following hypothesis: Would extended visual programming languages facilitate the development of learning scenarios based on AR? In order to support this hypothesis, we examined various successful programming environments to check which one could be extended with AR capabilities and, finally, we opted to extend the block language of AppInventor for dealing with AR concerns.

App Inventor was created by Google under an open source license and now maintained by the Massachusetts Institute of Technology (MIT). It is a blocks-based programming tool that empowers everyone, even novices, to start programming and build fully functional applications for Android devices. With this authoring tool, teachers and people in general can create simple applications deployable on mobile phones and tablets. Such applications can use the common features allowed by these devices, such as sensors (GPS location, orientation, accelerometer, etc.), multimedia elements (photo camera, microphone, video player, etc.), drawing and animation, sharing data with social networks and different web services, among others. AppInventor's architecture is composed of several modules: an GWT application for designing the user interface of the new apps, a Blocky editor for programming the behaviour logic of the app, a buildserver to turn the design and the logic above into an exportable file (apk), an interpreter that runs on the mobile device to debug the apps, and finally, a module with all the built-in components (visuals or non-visuals), which are needed by the other modules and available for the end users to develop their applications.

AppInventor components are the building blocks needed for defining the appearance and the behaviour of the apps and they consists of several properties, events and methods. In our case, the following kinds of components (see Figure 1) were developed:

- **ARCamera** This component represents the AR scene which shows the real image directly captured by the device's camera. The user is allowed to configure the screen orientation, the kind of camera (front or back one), an external database (optional) of physical targets to recognize and whether it requires stereoscopic rendering. This component fires an event when user taps the screen with his finger.
- **ARTrackers** This kind of component corresponds with the physical elements used for triggering some action in the final application. This physical element can be typically an AR marker, an external object (image, cuboid or cylinder), or also a region with a specific color or a piece of text. This component manages various events: when the physical element is recognized in the field of view of the camera, when its position changes or disappears.
- **ARAssets** This kind of component allows users to declare the virtual objects to be rendered in a given position on the screen. These elements can be texts or more commonly 3D models in different formats (3DS, OBJ, MD2, ASC) with textures (based on images or colours). There is an additional property 'StickTo' which is intended to bind the virtual object with a given tracker

and other properties to set the position, rotation and translation for that binding. Also, users will be able to rotate the virtual object by swiping with one finger on the 3D models. Furthermore, when a virtual object collides with another one on the screen, this component will fire a specific event.

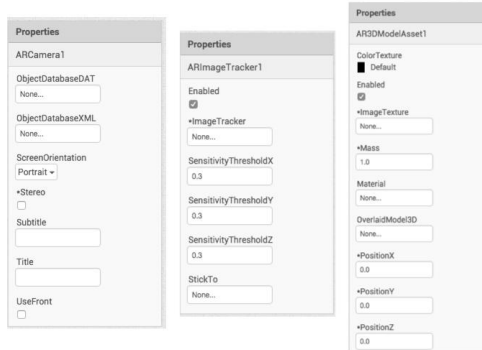


Figure 1. Properties of the App Inventor

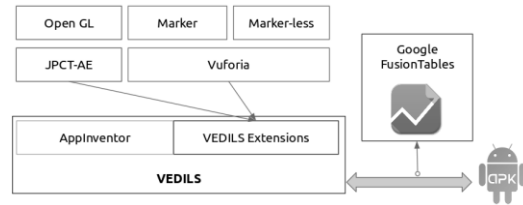


Figure 2. Structure of VEDILS

In order to develop these new components, some Java Android libraries were used. Vuforia SDK was the alternative chosen to recognize and track planar images and simple 3D objects in real-time. Vuforia includes the required computer vision algorithms, but doesn't include any facilities for displaying digital information on the screen in a easy way. To do this, we opted to use the jPCT-AE 3D engine, a set of libraries, above the OpenGL API, for rendering 3D models in Android.

The AppInventor components have been developed in a non-intrusive way, so it is possible to change the specific support libraries for recognizing and tracking physical objects and rendering virtual objects to other libraries without affecting the design of these components. Currently there is a single point of coupling, namely the external objects to be used as trackers in VEDILS have to be previously managed in a specific database of physical targets on the Vuforia website.

4. EVALUATION

With the aim of evaluating our proposal, according to one of the evaluation methods classified by Hevner (Alan et al., 2004), a descriptive use scenario of VEDILS is described here (see Figure 2). In subject Desing I of the curriculum of the degree on Mechanical Engineering at University of Algarve, students learn foundations of technical drawing including perspectives, layouts, spaces, measurement units, etc. With the aim of practising their abilities, the teacher gives their students several printable documents with some images of isometric views of different pieces. Then, the students can visualize 3D models of the pieces deployed over the isometric views, by using a free AR app for mobile devices, called Augment (Augment 2016). Previously, the teacher had to link the images with the corresponding 3D models for those pieces, by using the authoring environment provided along with this tool.



Figure 3. Designer view

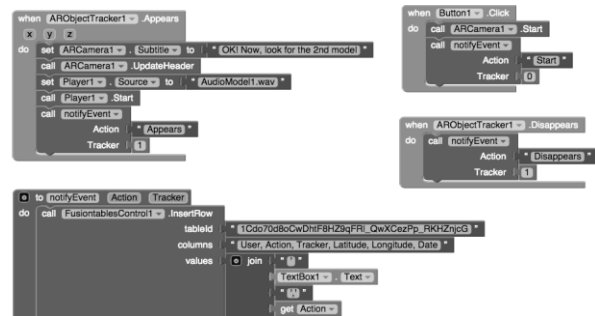


Figure 4. Blocks view

This tool aims at helping students to achieve a better understanding of 3D graphical projections. However, there might be several unanswered questions for teachers. For instance, how many students have practiced with the tool? How long did the students spend completing the activity? At which 3D model did the students pay more attention? These questions are out of the scope of this and similar AR authoring tools. Another drawback of this tool is the lack of a mechanism for guiding the student by giving specific instructions depending on the phase of the task assignment where he/she is. In summary, there is no way to include domain-specific logic into the application. To achieve this, it would require developing a specific application by using a programming language and some libraries. However, it would be a time-consuming task that requires considerable programming skills. We opted to design an Android application by using the VEDILS tool to demonstrate its ease of use.

The illustrative application designed for the subject enables students not only to visualize the 3D models, but also to track the students' activities and give voice explanations about the models and the steps required to complete the activity. A snapshot of the components used for the application in the VEDILS tools and a snippet of the main blocks required for the behaviour of the new app are shown in Figures 3 and 4.



Figure. 5. 3D model over tracker

User	Action	Tracker	Latitude	Longitude	Date
pedro	Disappears	3	36.50636	-6.26579	Jan 7, 2016, 1:56:32 PM
pedro	Finish	0	36.50636	-6.26579	Jan 7, 2016, 1:56:37 PM
user1	Start	0	36.50631	-6.26571	Jan 7, 2016, 2:42:47 PM
user1	Appears	1	36.50631	-6.26571	Jan 7, 2016, 2:43:07 PM
user1	Disappears	1	36.50631	-6.26571	Jan 7, 2016, 2:43:11 PM
user1	Finish	0	36.50631	-6.26571	Jan 7, 2016, 2:43:12 PM
amaral	Start	0	36.50637	-6.26554	Jan 7, 2016, 5:30:55 PM
amaral	Appears	1	36.50637	-6.26554	Jan 7, 2016, 5:31:11 PM
amaral	Disappears	1	36.50637	-6.26554	Jan 7, 2016, 5:31:46 PM
amaral	Appears	2	36.50637	-6.26554	Jan 7, 2016, 5:32:02 PM
amaral	Disappears	2	36.50637	-6.26554	Jan 7, 2016, 5:33:09 PM
amaral	Appears	3	36.50637	-6.26554	Jan 7, 2016, 5:34:09 PM
amaral	Disappears	3	36.50637	-6.26554	Jan 7, 2016, 5:34:17 PM
amaral	Finish	0	36.50637	-6.26554	Jan 7, 2016, 5:34:19 PM
lola	Start	0	36.5379	-6.20238	Jan 8, 2016, 11:51:29 AM
lola	Disappears	1	36.5379	-6.20238	Jan 8, 2016, 11:51:58 AM

Figure. 6. Data about student's activities

To develop the app with that functionality some AppInventor built-in components were required. For example, the FusionTablesControl component was used to issue data about the user, the identifier of the recognized trackers, the time spent for visualizing the corresponding 3D models and the location coordinates, to a shared spreadsheet on the cloud-based service for data management of Google. This component is invoked after the trackers appear and disappear of the field of view. The information about the coordinates of the user is provided by other AppInventor component, called LocationSensor. Moreover, the Player component is used to play audio files containing explanations of the teacher for the models. In Figure 5, the working application can be observed.

Collecting data about the interaction of the students with the trackers and the assets of AR offers us a lot of possibilities for learning analytics. In Figure 6 one can see a snapshot of the event log triggered by the user interactions during the visualization of the 3D models. With this information, the teacher can know, for instance, which of the isometric views of the pieces produces more difficulties to the students.

An example of AppInventor enriched with the VEDILS components is freely available online (Vedils web site 2016). The source project of the application described in this section, the apk ready to be installed in the mobile devices, the printable documents with the views of the pieces and some of the charts generated from the students' activities with FusionTables are available for downloading from VEDILS site (Vedils 2016).

5. CONCLUSION

AR technology allows the inclusion of virtual elements on a vision of actual physical environment for the creation of a mixed reality in real time. This kind of technology can be used in educational settings. In this work, different RA authoring tools have been analyzed. However, the current AR authoring tools present several drawbacks, such as, the lack of a mechanism for tracking the students' activities, the capability of

detecting collisions among virtual objects, allowing establish only one-to-one relationships between trackers and virtual objects, etc.

With the aim of tackling those problems, we have developed the VEDILS tools. Our proposal is a visual environment for designing interactive learning scenarios where teachers can author AR applications. Rather than design a new programming environment, we have extended the App Inventor platform, since it provides a simple drag-and-drop interface, generates mobile applications, and it is open-source. Vuforia SDK was the alternative chosen to recognize and track planar images and simple 3D objects in real-time. For rendering 3D models in Android we opted for using the jPCT-AE 3D engine, a set of libraries, above the OpenGL API.

This alternative of reusing existing software for including AR capabilities has been successfully applied in other works. For example, DART (MacIntyre et al., 2004) is an authoring tool for creating classical screen-based multimedia presentations and desktop Virtual Reality presentations, which extends Macromedia Director. Moreover, AR Spot (AR Spot 2016) is an augmented-reality authoring environment based on MIT's Scratch project, which allows children to create experiences that mix real and virtual elements.

Currently, we are working on the development of new improvements to the AR components, in order to support more types of trackers and virtual assets and avoid the coupling with the Vuforia database of physical targets. Furthermore, we are developing a new system to capture user interactions more intuitive than the Fusion Tables component already available in App Inventor. Finally, we are also beginning to explore new ways of interacting with virtual assets of AR by detecting gestures and voice commands.

ACKNOWLEDGEMENT

Authors thank the support provided by the University of Cadiz that funded this work with the program for Researching and Innovation in Education, with the projects sol-201400047718-tra "Desarrollo y despliegue de recursos de Realidad Aumentada en el aprendizaje", sol-201500054605-tra "La Realidad Aumentada como recurso didáctico para la mejora del aprendizaje de una lengua extranjera", sol-201500054613-tra "Herramienta para el despliegue de manuales virtuales secuenciales en dispositivos móviles". Many thanks also to University of Algarve by the material of the subjects used for testing.

REFERENCES

Journal

- Alan, R Hevner von, Salvatore T March, Jinsoo Park, and Sudha Ram (2004). "Design science in information systems research". In: MIS quarterly 28.1, pp. 75–105.
- Azuma, RT (1997). "A survey of augmented reality". In: Presence.
- Bacca, Jorge, Silvia Baldiris, Ramon Fabregat, and Sabine Graf (2014). "Augmented Reality Trends in Education: A Systematic Review of Research and Applications". In: 17, pp. 133–149.
- Bujak, Keith R., Iulian Radu, Richard Catrambone, Blair MacIntyre, Ruby Zheng, and Gary Golubski (2013). "A psychological perspective on augmented reality in the mathematics classroom". In: Computers & Education 68, pp. 536–544. ISSN: 03601315.
- Chang, Kuo En, Chia Tzu Chang, Huei Tse Hou, Yao Ting Sung, Huei Lin Chao, and Cheng Ming Lee (2014). "Development and behavioral pattern analysis of a mobile guide system with augmented reality for painting appreciation instruction in an art museum". In: Computers and Education 71.July 2015, pp. 185–197. ISSN: 03601315.
- Di Serio, Ángela, María Blanca Ibáñez, and Carlos Delgado Kloos (2013). "Impact of an augmented reality system on students' motivation for a visual art course". In: Computers & Education 68, pp. 586–596. ISSN: 03601315.
- Guyen, S. and S. Feiner (2003). "Authoring 3D hypermedia for wearable augmented and virtual reality". In: Seventh IEEE International Symposium on Wearable Computers, 2003. Proceedings. Pp. 118–126. ISSN: 1530-0811.
- Haller, Michael, Erwin Stauder, and Juergen Zauner (2005). "AMIRE-ES: Authoring Mixed Reality once, run it anywhere". In: The 11th International Conference on Human-Computer Interaction JANUARY 2005.
- Ibáñez, María Blanca, Ángela Di Serio, Diego Villarán, and Carlos Delgado Kloos (2014). "Experimenting with electro-magnetism using augmented reality: Impact on flow student experience and educational effectiveness". In: Computers & Education 71, pp. 1–13. ISSN: 03601315.

- Jara, Carlos A, Francisco A Candelas, Santiago T Puente, and Fernando Torres (2011). “*Hands-on experiences of undergraduate students in Automatics and Robotics using a virtual and remote laboratory*”. In: Computers & Education 57.4, pp. 2451–2461. ISSN: 0360-1315.
- Krevelen, D.W.F. Van W F Van and R. Poelman (2010). “*A Survey of Augmented Reality Technologies , Applications and Limitations*”. In: International Journal 9, pp. 1–20. ISSN: 10255834. Liu, Tsung-Yu and Yu-Ling Chu (2010). “*Using ubiquitous games in an English listening and speaking course: Impact on learning outcomes and motivation*”. In: Computers & Education 55.2, pp. 630–643. ISSN: 0360-1315.
- MacIntyre, B, Maribeth Gandy, Steven Dow, and JD Bolter (2004). “DART: a toolkit for rapid design exploration of augmented reality experiences”. In: Proceedings of the 17th . . .
- Martin, Sergio, Gabriel Diaz, Elio Sancristobal, Rosario Gil, Manuel Castro, and Juan Peire (2011). “*New technology trends in education: Seven years of forecasts and convergence*”. In: Computers and Education 57.3, pp. 1893–1906. ISSN: 03601315.
- Milgram, Paul and Fumio Kishino (1994). “*Taxonomy of mixed reality visual displays*”. In: IEICE Transactions on Information and Systems E77-D.12, pp. 1321–1329. ISSN: 09168532.
- Seichter, Hartmut, Julian Looser, and Mark Billinghurst (2008). “*ComposAR: An intuitive tool for authoring AR applications*”. In: 2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, pp. 177–178.
- Specht, Marcus, Stefaan Ternier, and Wolfgang Greller (2011). “*Dimensions of Mobile Augmented Reality for Learning: A First Inventory*”. In: Journal of the Research Center for Educational Technology 7.1, pp. 117–127. ISSN: 1948-075X.
- Wang, Yuan and Tobias Langlotz (2009). “*An authoring tool for mobile phone AR environments*”. In: Proceedings of New . . . Pp. 1–4.

Other Sources

- AR Spot (2016). URL: <http://ael.gatech.edu/lab/research/authoring/arspot/> (visited on 01/08/2016).
- ARToolKit (2016). URL: <https://www.hitl.washington.edu/artoolkit/> (visited on 01/08/2016).
- ArUco (2016). URL: <http://www.uco.es/investiga/grupos/ava/node/26> (visited on 01/08/2016).
- Augment (2016). URL: <http://www.augmentedev.com/es/> (visited on 01/08/2016).
- Augment (2016). URL: <http://www.augment.com> (visited on 01/20/2016).
- Aulablog X Encuentro (2016). URL: <http://www.aulablog.com/blog/encuentro10/> (visited on 01/20/2016).
- Aumentaty (2016). URL: <http://www.aumentaty.com/> (visited on 01/08/2016).
- Aurasma (2016). URL: <https://www.aurasma.com/> (visited on 01/08/2016).
- Chromville (2016). URL: <https://chromville.com/> (visited on 01/08/2016).
- Droidar (2016). URL: <https://github.com/bitstars/droidar> (visited on 01/08/2016).
- Hour of Code (2016). URL: <https://code.org/> (visited on 01/08/2016).
- Layar (2016). URL: <https://www.layar.com/> (visited on 01/08/2016).
- MIT App Inventor (2016). URL: <http://appinventor.mit.edu/explore/> (visited on 01/08/2016).
- Scratch (2016). URL: <https://scratch.mit.edu/> (visited on 01/08/2016).
- Vedils (2016). URL: <http://vedils.uca.es/> (visited on 01/20/2016).
- Vedils web site (2016). URL: <http://vedils.uca.es/web> (visited on 01/20/2016).
- Vuforia (2016). URL: <https://developer.vuforia.com/downloads/sdk> (visited on 01/08/2016).
- Wikitude (2016). URL: <http://www.wikitude.com/> (visited on 01/08/2016).