Computers and Electrical Engineering 65 (2018) 250–260

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Augmented reality mobile app development for all☆

José Miguel Mota[a,*], Iván Ruiz-Rube[a], Juan Manuel Dodero[a], Inmaculada Arnedillo-Sánchez[b]

[a] *Department of Computer Engineering, University of Cádiz, Puerto Real (Cádiz), Spain*
[b] *School of Computer Science and Statistics, Trinity College, Dublin, Ireland*

A R T I C L E   I N F O

Article history:
Received 7 March 2017
Revised 23 August 2017
Accepted 23 August 2017
Available online 1 September 2017

Keywords:
Learning activities
Augmented reality
Visual language
Mobile learning
Programming skills

A B S T R A C T

Lack of programming skills is a barrier to the engagement of teachers in the development and customisation of their own applications. Visual Environment for Designing Interactive Learning Scenarios (VEDILS), a visual tool for designing, customising and deploying learning technologies, provides teachers with a development environment with a low entry threshold. Current mobile devices are equipped with sensors and have sufficient processing power to use augmented reality technologies. Despite the heavy use of mobile devices in students' lives, the use of augmented reality mobile applications as learning tools is not widespread among teachers. The current work presents a framework comprising the development tool and a method for designing and deploying learning activities. It focuses on the augmented reality components of the authoring tools, which allow users to create their own mobile augmented reality learning apps. It also present the results of the evaluation of the framework with 47 third-level educators, and two case studies of classroom implementations of mobile augmented reality apps developed by these educators. The results illustrate the suitability of the framework and authoring tool for supporting users without programming skills in developing their own apps.

1. Introduction

In Technology Enhanced Learning (TEL) research, ubiquitous and mobile technologies and serious games, Augmented Reality (AR) and Learning Analytics (LA) provide a means of improving users' experiences and satisfaction in enriched, multimodal learning environments [1]. While TEL research takes advantage of technological innovations in mobile hardware and software, significant developments in user modelling and penalisation techniques have placed students at the center of the learning process [2], and AR has had the effect of increasing the motivation of students [3].

AR technology refers to the inclusion of virtual elements in views of actual physical environments, in order to create a mixed reality in real time. It supplements and enhances the perceptions humans gain through their senses in the real world. AR provides various degrees of immersion and interaction, which can help to engage students in e-learning activities. For instance, in AR learning environments, motivational factors related to attention and satisfaction are rated more highly than for slide-based environments [3].

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Associate Editor Dr. R. Varatharajan.

* Corresponding author.
E-mail addresses: josemiguel.mota@uca.es (J.M. Mota), ivan.ruiz@uca.es (I. Ruiz-Rube), juanma.dodero@uca.es (J.M. Dodero), macu.arnedillo@scss.tcd.ie (I. Arnedillo-Sánchez).

http://dx.doi.org/10.1016/j.compeleceng.2017.08.025
0045-7906/

Society increasingly demands efficient and skilled professionals. In view of this, a particularly valuable skill for educators is the ability to create their own e-learning experiences [4]. Doing so endows educators with agency over the technology; that is, the ability to place this at their service rather than being driven by it [5]. However, the full potential of technology is often underutilised in teaching. Barriers include the need for programming skills to develop customised AR experiences [6] and the lack of education-specific authoring tools [7]. The translation of concepts and instruments into end-user computer systems is a complex task, requiring the involvement of computer specialists; the development of education-oriented programming environments therefore remains a challenge [8].

Visual block-based programming environments, where users drag and drop blocks together to write code, provide novices with an alternative for learning programming. This has proven successful in classrooms and informal learning settings [9,10]. It is also suitable for early-stage programming teaching, for instance the *One Hour of Code*[1] initiative or with tools such as *Scratch*[2] or *App Inventor*.[3]

Based on the potential that block-based visual programming languages have shown in overcoming the barriers faced by educators when developing mobile applications, we developed VEDILS [11]. This is built on top of App Inventor, and is extended with modules that allow the integration of AR and LA features into mobile applications. VEDILS is embedded in the VEDILS Framework, which is a four-phase method comprising: 1) the design and development of new components; 2) training; 3) iterative design; and 4) assessment. To evaluate this authoring tool, a series of workshops were conducted with 47 third-level educators. During these, the participants completed tasks aimed at improving their knowledge of the visual block-based programming language, and developed an app. In addition, two case studies were undertaken of classroom implementations of the mobile augmented reality learning apps developed by the participants.

This paper presents the VEDILS Framework and focuses on the AR components of the VEDILS tool, which allow users to create their own mobile augmented reality learning apps. It also presents the initial results from the evaluation by educators. The remainder of this paper is structured as follows: Section 2 describes the state of the art; Section 3 presents the VEDILS Framework, outlining the stakeholders and stages of this system; Section 4 describes the authoring tool and explains the AR components; Section 5 presents the results of the workshops; Section 6 reports on two case studies of apps developed with this tool; and finally, Section 7 presents the conclusions and describes future work.

2. Related work

New instructional environments based on multimodal interaction (for instance, gesture interaction, voice recognition, AR, tactile interaction or artificial vision), have emerged as delivery formats which are readily available to teachers. It is easier to motivate [12] and involve students when e-learning experiences are based on these types of interaction [13]. In particular, AR enhances the perception of users and improves their interactions with the real world, displaying information that users cannot detect directly using their senses [14].

There are two taxonomies of AR application in terms of development: 1) marker-based tracking, which requires labels containing a coloured or black and white pattern; and 2) marker-less tracking, which uses the mobile device's GPS or image recognition systems to identify a location. Both make use of sensors such as accelerometers and gyroscopes to determine the location, orientation and direction of the mobile device. AR systems look for a predefined pattern in order to identify a match and a reference position, and the virtual information (sounds, 2D images, 3D models, etc.) is then superposed on the live image. Researchers have highlighted the potential of AR in engaging and motivating learners, for instance when explaining or evaluating topics, conducting lab experiments, playing educational games and augmenting information, among others [3].

When developing e-learning activities with AR, teachers can use various AR authoring tools. A classification according to the schema outlined in [15] is presented below:

- *Low-level libraries:* These provide only basic computer vision integration. Some of the best-known libraries and frameworks are ARToolKit [16], Aruco [17], Wikitude[4] and Vuforia.[5] All of these require coding/scripting skills and the use of graphical utilities to project and manipulate virtual objects.
- *High-level programming environments:* These simplify the development process by providing the infrastructure required to build applications with AR. Examples of these tools are Studierstube[6] and osgART [18]. They include all the functions needed to develop AR applications, such as scene graph rendering, networking, window management, support for input devices and so forth. These environments usually require programming abilities; developing AR applications with these is therefore a time-consuming and demanding task.
- *GUI-based tools for non-programmers:* These do not require programming skills in order to generate AR applications. Examples of these are APRIL [19], AMIRE [20] and DART [21]. Several other high-level authoring tools exist, such as Layar,[7]

[1] https://code.org/learn.
[2] https://scratch.mit.edu/.
[3] http://appinventor.mit.edu.
[4] http://www.wikitude.com/.
[5] https://developer.vuforia.com/downloads/sdk.
[6] http://studierstube.icg.tugraz.at/main.php.
[7] https://www.layar.com/.

Aumentaty,[8] Augment,[9] Aurasma[10] or ARCrowd,[11] which only support one-to-one mappings between physical markers and virtual objects. The resulting interactions are easy and intuitive, although somewhat limited.

Regardless of the authoring tool used to create AR educational applications, a number of difficulties regarding their implementation are well documented [2]:

- *Student monitoring:* This is an essential component of high-quality education, and is "one of the major factors differentiating effective schools and teachers from ineffective ones" [22]. However, most mobile learning applications do not provide teachers with built-in monitoring features to view statistical information on students' interactions.
- *Assessment:* Authoring tools do not usually include features for creating questionnaires or other assessment tools, and instructors must use external instruments to evaluate students.
- *Personalisation:* High- and low-level tools can provide customisation of activities; however, they require substantial programming skills in order to build AR applications.
- *Leveraging mobile capabilities:* Many authoring tools do not take full advantage of mobile devices, such as sensor data acquisition, information storing and connectivity with external systems, among others.

3. The VEDILS framework: supporting teachers in designing and deploying mobile learning activities

The VEDILS framework aims to support users without programming skills in designing and deploying mobile technologies. Since it is content independent, it can be used to support professionals from many different domains. Although the current work focuses on how VEDILS supports teachers in creating their own e-learning activities, it is also applicable to many other contexts, such as e-health, design and others. For instance, the e-health market is populated by apps created by non-clinical developers [23].

The framework comprises a four-stage method based on Design-Based Research (DBR) [24] and the VEDILS [11] authoring tool. It is characterised by an iterative methodology, development of interventions in situ, and a commitment to teaching and learning. The stakeholders in the framework are:

- *Leaderes:* These form a link between the teachers, programmers and researchers, to define new technologies that can be incorporated, and are accountable for the quality of the final process.
- *Programmers:* These are knowledgeable professionals in programming languages and low-level/high-level libraries.
- *Researchers:* These research and propose the design of new components. They are knowledgeable about research methodologies and DBR, and can therefore design processes to test these designs in natural learning environments.
- *Teachers:* These create their own learning activities with VEDILS, and can propose teaching approaches and technologies to be integrated with these. They are trained in VEDILS and understand the possibilities of the authoring tool.
- *Students:* These are the end-users. Their background, context, needs and desires should be taken into account by teachers when designing new learning apps or improving existing ones.

The four stages of the method for designing and deploying learning activities are as follows: 1) design and development of new components; 2) training; 3) iterative design; and 4) assessment (Fig. 1). These can be described as follows:

1. *Design and development of new components:* This only takes place when leaders, teachers and researchers decide to use a new technology, for instance gestural interaction, voice recognition or AR, or to improve an existing feature. Then, developers with expertise in the code base of VEDILS incorporate the new technology or feature into the platform by defining a component which includes a set of visual properties and several programming blocks.
2. *Training:* Leaders, programmers and researchers can adopt the role of trainers by introducing teachers to the concepts and technologies that can be used with the authoring tool, in order for the teachers to understand their potential and benefits.
3. *Iterative design:* Teachers, using a block-based visual language and the tools provided by VEDILS, develop ready-to-deploy mobile apps for incorporation into their teaching. This phase can be repeated several times to develop new e-learning activities or to improve existing ones, based on the interaction data generated as students complete tasks.
4. *Assessment:* Once the students have installed the application, the teacher can obtain real-time information on users' interactions; for example, information on which screens are being viewed, where, when and so forth. In addition, the assessment of the educational activity can be developed within the application itself. All of this information is made available to the teachers, enabling them to analyse the activity and to return to the previous phase to improve the app if necessary.

[8] http://www.aumentaty.com/.
[9] http://www.augment.com/.
[10] https://www.aurasma.com/.
[11] http://arcrowd.com/.

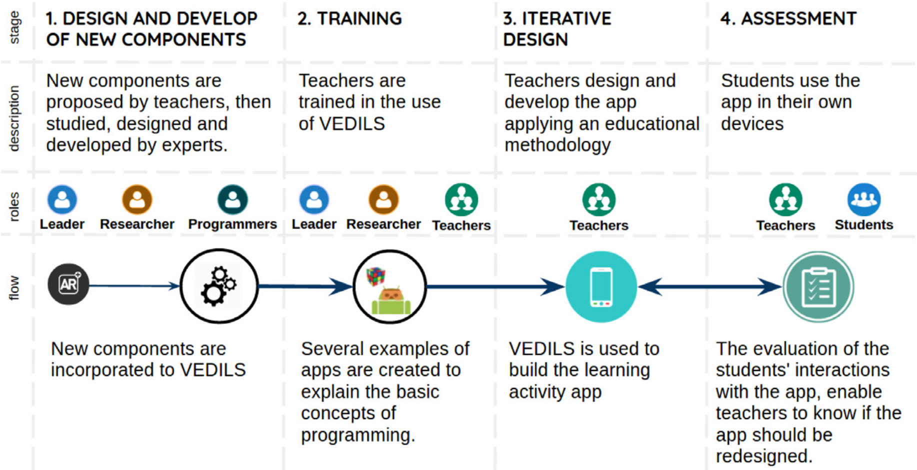


Fig. 1. Method to design and deploy learning activities.

Fig. 2. Screen to design the app and define the properties of the components.

4. VEDILS: the authoring tool

Rather than developing a new programming environment, we examined successful programming environments to assess whether any of these could be extended. Each environment was evaluated based on a number of criteria, including programming simplicity (children should be able to create rich applications in it), spatial simplicity (minimal load in terms of spatial cognition, with a preference for 2D over 3D), and ease of expansion (capability of being expanded with AR functionality).

VEDILS is an authoring tool based on App Inventor; it was designed by Google under an open source license, and is currently maintained by the Massachusetts Institute of Technology (MIT). It is a block-based programming tool that empowers users, including novices, to start programming and to build fully functional applications for Android devices. The applications use common features provided by Android devices, for example sensors, such as a GPS receiver, gyroscope or accelerometer; multimedia elements, such as a photo camera, microphone or video player; drawing and animation; data sharing with social networks; and various web services.

VEDILS's architecture consists of several modules: a GWT application for designing the user interface of the new apps (Fig. 2); a Blocky editor for programming the behavioural logic of the app (Fig. 3); a build server, which turns the design and the logic above into an exportable file (apk); an interpreter, which runs on the mobile device to debug the apps; and

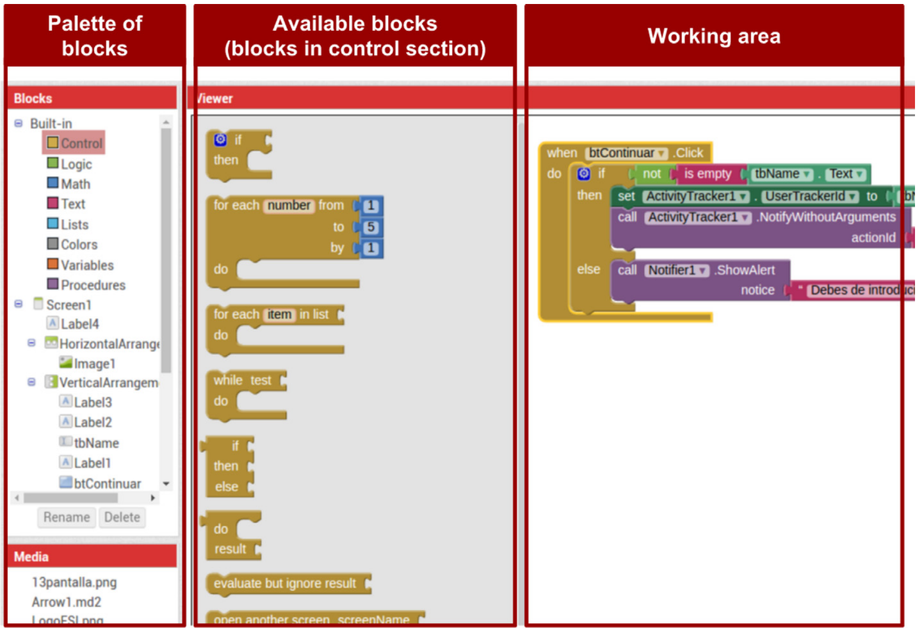
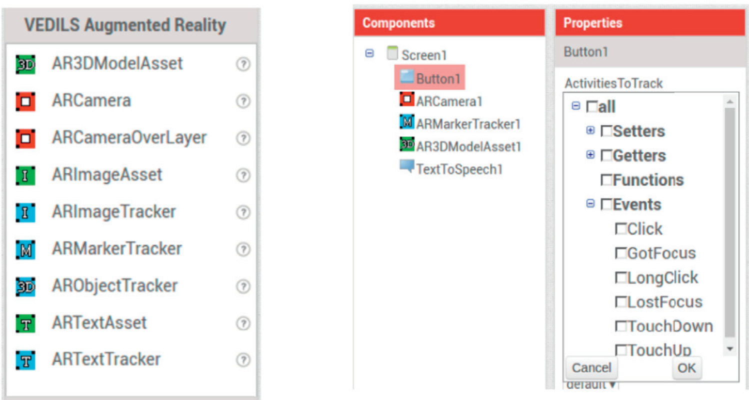


Fig. 3. Use of a visual language based in blocks to define the app's behaviour.



(a) (b)

Fig. 4. Image (a) show all augmented reality components of VEDILS, and image (b) the tracking options, in this case of button component.

finally, a module with all the built-in components (visuals and non-visuals) needed by the other modules and available to the end users for developing their applications.

4.1. Augmented reality components

In order to address the issues with current AR authoring tools, namely student monitoring, assessment, personalisation and leveraging the capabilities of mobile devices, the following components (Fig. 4(a)) were developed in order to implement AR features in the VEDILS authoring tool:

- *ARCamera:* This component represents the AR scene, which shows the real image directly captured by the device's camera. The user can configure aspects such as the screen orientation, the camera (front or rear), an optional external database of physical targets to be recognised, and whether stereoscopic rendering is required. Additionally, using an ARCameraOverlay component associated with the ARCamera, a navigation bar can be displayed over the screen.
- *ARTrackers:* This component corresponds with the physical elements used to trigger actions in the final application. This physical element can be an AR marker, an external object (image, cuboid or cylinder) or text. This component manages events, such as when the physical element is recognised in the field of view of the camera, when its position changes or when it disappears.

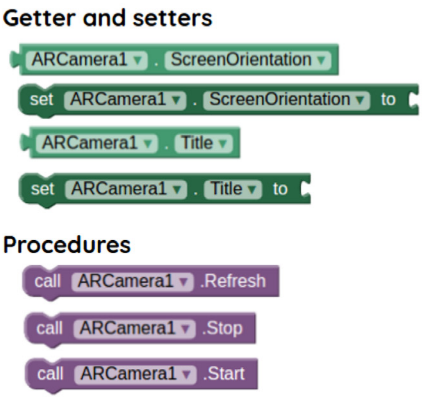


Fig. 5. Blocks for programming component ARCamera.

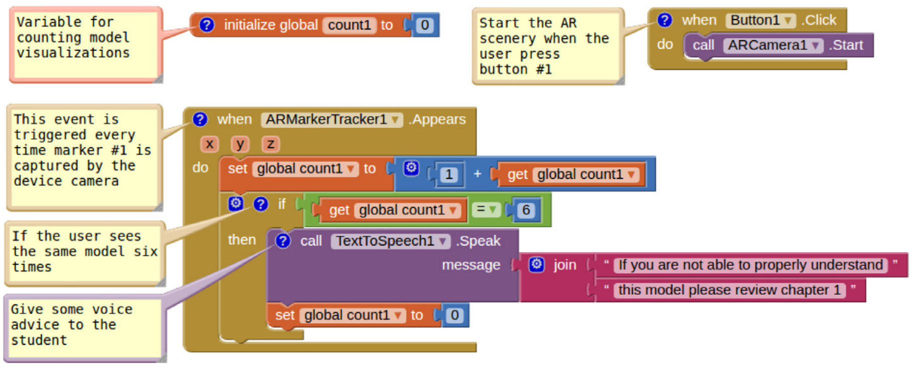


Fig. 6. Program using AR Blocks.

- *ARAssets:* This component enables users to define the virtual objects to be rendered in a given position on the screen. These elements can be text, 2D images or 3D models in various formats (3DS, OBJ, MD2 or 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 which set the position, rotation and translation for that binding. The virtual objects can also be rotated, by swiping with one finger on the 3D models, or zoomed in/out using two fingers.

Java-based Android libraries were used to develop the new components. Vuforia SDK is used to recognise and track flat images and simple 3D objects in real time. Although Vuforia includes a visual recognition algorithm, it does not provide the functions required to display 3D objects on the screen in a straightforward way; to achieve this, the jPCT-AE 3D engine was used. A set of libraries above the OpenGL API were used to render 3D models in Android.

In addition, all components of VEDILS were extended using a special property to select the interaction data to be collected (see Fig. 4(b)). This is a non-intrusive way of automatically intercepting component data when a component function is invoked, a certain event is received or a property value is accessed or modified.

4.2. Block-based programming

A set of new programming blocks are available in VEDILS which enable users to include AR features in their apps. These can be divided into the following categories:

- *Getter/setter blocks:* The properties of the components can be modified or obtained from the design view as well as from the blocks view.
- *Procedure blocks:* These enable users to start or stop the AR recognition.
- *Behaviour blocks:* These are produced in response to events that arise from the interaction of students with the AR elements, such as when an object appears or disappears.

Examples of blocks for the ARCamera component are illustrated in Fig. 5; an example of their use is shown in Fig. 6, and further examples can be found on the VEDILS channel on YouTube.[12]

[12] http://vedils.uca.es/web/index.html#documentation.

Table 1
Questionnaire based on TAM.

Perceived usefulness	Perceived ease of use	Intention to use
Is easy ...	Is useful ...	I'll use to ...
Q1: to learn	Q6: to create e-learning activities	Q9: create e-learning applications
Q2: to use the programming language	Q7: to explain a topic with AR	Q10: create e-learning activities with AR
Q3: to create e-learning activities	Q8: to analyze the interactions of my students	Q11: to analyze my e-learning applications
Q4: to create e-learning activities with AR		
Q5: to know the students interactions		

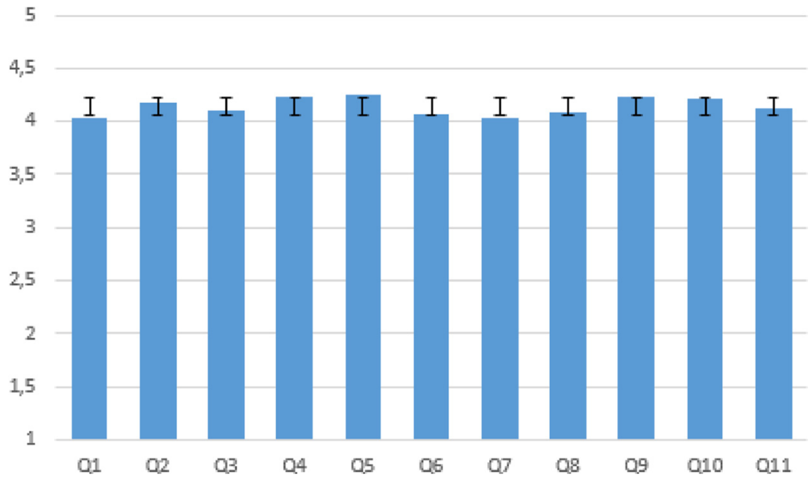


Fig. 7. Overview of the technological acceptance results.

5. Training workshop evaluation

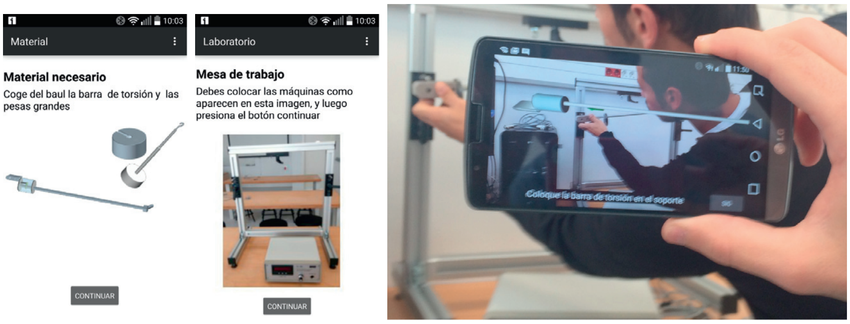
As part of the training phase of the method proposed in this research, three workshops were conducted with 47 university lecturers. The duration of each workshop was five hours, and they began with a brief introduction to the educational applications of mobile devices and the foundations of AR. Following this, the use of the authoring tool was explained. To reinforce and consolidate what was learnt, participants developed educational mobile apps by themselves.

In order to determine whether VEDILS is a user-friendly solution for designing and deploying mobile applications for educational purposes, a questionnaire was administered which evaluated the technological acceptance and usability of the tool (Table 1) based on the Technology Acceptance Model (TAM) [25]. All the questions, except those concerning profiling information, were rated using a five-level Likert scale.

On average, the participants had 14 years of teaching experience, and their backgrounds were as follows: arts and humanities (10.64%), engineering and architecture (46.81%), health sciences (19.15%), natural sciences (17.02%), and law and social sciences (6.38%). In terms of their use of technology, where M is the average: they used mobile devices to support their learning activities (M=1.98); they used on-line questionnaires for evaluation (M=2.1); they could deploy learning activities with AR (M=1.05); they could develop applications for mobile devices (M=1.57); and they had sufficient programming skills (M=2.19). These results indicate that lecturers attending the workshops did not have enough knowledge to deploy e-learning activities by themselves.

TAM [25] provides a method to identify the Perceived Usefulness (PU), the Perceived Ease of Use (PEU) and Intention to Use (ITU) of a certain technology. The questions used in this study (Table 1) were adapted to match our requirements. To check the reliability of answers, Cronbach's alpha was calculated, obtaining an acceptable value of 0.870. As illustrated in Fig 7, the results show high values, indicating a high overall acceptance of VEDILS among the participants. These results also reveal that participants were able to design and execute a mobile learning scenario using AR, although none had a computing background or prior programming skills. This therefore provides initial evidence of the suitability of block-based programming languages with AR extensions in introducing teachers to the design and deployment of their own mobile learning scenarios. In order to guarantee the reproducibility of the study, all questionnaires, results and graphics can be found on the VEDILS web site.[13]

[13] http://vedils.uca.es/web/index.html#workshops.



(a) (b) (c)

Fig. 8. Application of assembly manual with AR. Image (a) indicates the required elements, image (b) shows how to position the transformer and image (c) shows a student using the app.

6. Case study evaluation

This section presents two AR-enriched mobile learning applications developed by lecturers using the VEDILS framework and its authoring tool. In addition to describing the apps, we provide details of the classroom implementations and the overall acceptance and user experience evaluation. The first case study was conducted with 33 engineering students, who used an AR-enriched app to support an Assembly Manual for a Gunt Strain Gauge Training System. The second was carried out with 41 students using the *WerBinIch* (Who am I?) language learning app. The evaluation of this case study focused on the analysis of the students interactions in order to monitor their progress, as well as the appropriateness of the learning tasks.

6.1. AR Assembly Manual App for a Gunt Strain Gauge Training System

This application was designed by the lecturers of a Material Resistance course, taught as part of the Industrial Design Engineering degree course at a university. Its aim was to support the students in learning and practising how to set up the torsion bar in the lab. Currently, the lectures face two main problems: 1) the high number of students, which affects the amount of hands-on practice students can get, and has a knock-on effect on their learning; 2) the need to guide students in setting up the lab equipment, to ensure they do this properly and do not damage it.

The design of the AR Gunt Strain Gauge Training System Assembly Manual App followed a learner-centered approach, and enabled users to engage in guided practice at their own pace and as often as required. The app was used in the first 15 min of the class in the laboratory. The manual contains the following steps for setting up the equipment: 1) turn off the transformer; 2) mount the torsion bar over a frame; 3) connect the cable from the torsion bar to the measuring equipment; 4) turn on the transformer; 5) adjust the potentiometer; 6) hang the weight set on the lever arm; and 7) increase the weight gradually. These steps are illustrated by rendering 3D models on the screen and are enhanced with text and audio information (Fig. 8).

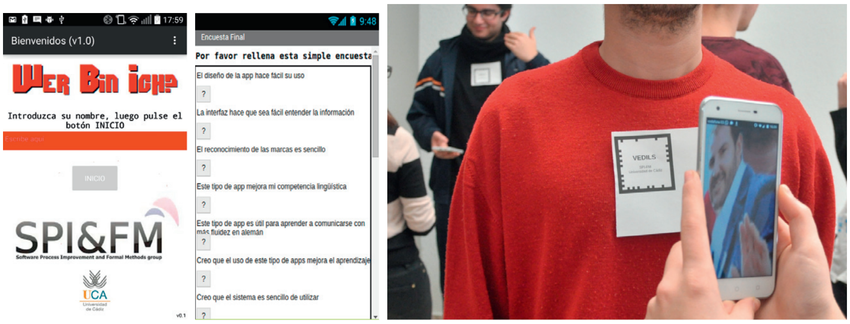
In order to evaluate the overall acceptance and user experience of the application, we conducted an online survey with the students who had used the app. In total, there were 33 responses. All the questions on the survey were rated using a seven-level Likert scale. The complete survey, results and charts can be found on the project's web site.[14] Students gave positive answers to the question “I think using these types of apps improves learning”, with an average of M=6.12. Our results indicate that AR is a simple, intuitive, and engaging technology, even for students who have not used this type of learning technology before.

6.2. Werbinich (Who am I?) language learning app

The *WerBinIch* application was developed in order to build a collaborative mobile learning scenario to help in learning German as a foreign language. This application, designed by lecturers in German as a foreign language, is based on the popular game “Who am I?”; it aims at a dynamic reinforcement of the knowledge acquired by students when practicing how to describe in German the physical characteristics of people, such as hair colour, height and so forth, as well as their non-physical characteristics, for instance profession, personality and so on. This activity was performed during a two-hour class.

The steps of the activity are as follows: 1) each participant randomly selects an AR mark and displays it to the others; 2) they launch the application and log in (Fig. 9(a)); 3) they capture their own marker using the app to associate the popular

[14] http://vedils.uca.es/web/index.html#portfolio.



(a) (b) (c)

Fig. 9. Photos from the activity in class.

character from the marker with themselves; 4) they start interacting with their peers, asking and answering questions about their own characters to find out who they are. In order to answer the questions properly, the students can see the characters of their peers by capturing their markers (Fig. 9(c)), although they can never visualise the character in their own marker; 5) when they know who they are, the students can select their character from a drop-down list; 6) if the answer is correct, the app returns to the camera function to enable users to start again. If the answer is incorrect, the student continues asking questions until they correctly guess who they are. At the end of the game, the participants are prompted to complete a satisfaction survey (Fig. 9(b)).

During the pilot study, a large amount of data on the students' interactions was captured. This data was analysed by the lecturer of the module in order to understand how the students were performing. The complete survey, results and charts can be found on the project's web site.[15] The analysis revealed that 78% of the participants were able to guess the secret character the first time around and without any mistakes. Interestingly, the analysis also highlighted three characters that involved a higher average number of failed attempts. This may indicate some difficulty on the part of the students in describing the characteristics associated with those characters. Regardless of the actual issue, which remains for the teacher to interpret, our approach enables relevant information to be collected which can detect such situations, thus enabling teachers to identify learning problems and then to modify the app to address these appropriately.

7. Conclusions and future work

In this paper, we present the VEDILS Framework, a development method and tool which enables teachers to design and deploy learning activities. The framework comprises a four-stage method: 1) design and development of new components; 2) training; 3) iterative design; and 4) assessment. The authoring tool provides a block-based programming language for building Android apps. We focus here on the AR components features of the tool. These were developed to design and deploy AR-enriched mobile learning apps. The AR components also aim to address the shortcomings of current AR tools for learning, namely student monitoring, assessment, personalisation, and the underutilisation of certain features of mobile devices.

Through an evaluation of the framework with 47 third-level educators, we found that block-based programming languages such as the one used in the VEDILS authoring tool can help teachers to overcome their lack of programming skills, thus enabling them to develop their own mobile applications. The TAM evaluation indicates a high level of acceptance for VEDILS among the lecturers surveyed.

With respect to the evaluation of two case studies, based on the classroom implementation of AR mobile learning apps developed by lecturers, the results also show a high level of acceptance from students. The second case study illustrates how the student interaction data captured by the app developed in VEDILS can provide valuable feedback in terms of student performance and can also inform the instructional design of learning activities and contents.

The next step in the development of the VEDILS framework is to support new technologies in the authoring tool. To this end, we are currently working to integrate gesture recognition and brain activity detection with devices such as Leap Motion and EMOTIV Epoc+. In doing this, we expect to involve lecturers in health sciences in the VEDILS project.

Acknowledgements

This research has been partially funded by the Spanish Ministry of Economy and Competitiveness through the EmPhasys project grant (ref. RTC-2016-5095-1).

[15] http://vedils.uca.es/web/index.html#portfolio.

References

[1] . NMC horizon report, games and gamification. The New Media Consortium; 2014.
[2] Bacca J, Baldiris S, Fabregat R, Graf S. Augmented reality trends in education: a systematic review of research and applications. J Educ Technol Soc 2014;17:133–49.
[3] Di Serio A, Ibáñez MB, Kloos CD. Impact of an augmented reality system on students' motivation for a visual art course. Comput Educ 2013;68:586–96.
[4] Santos MEC, Yamamoto G, Taketomi T, Miyazaki J, Kato H. Authoring augmented reality learning experiences as learning objects. In: 2013 IEEE 13th international conference on advanced learning technologies; 2013. p. 506–7.
[5] Rushkoff D. Program or be programmed: ten commands for a digital age. O/R Books; 2010.
[6] Kurkovsky S, Koshy R, Novak V, Szul P. Current issues in handheld augmented reality. In: 2012 International conference on communications and information technology (ICCIT); 2012. p. 68–72.
[7] Martin S, Diaz G, Sancristobal E, Gil R, Castro M, Peire J. New technology trends in education: seven years of forecasts and convergence. Comput Educ 2011;57(3):1893–906.
[8] Miglino O, Walker R. Teaching to teach with technology - A project to encourage take-up of advanced technology in education. Procedia - Soc Behav Sci 2010;2(2):2492–6.
[9] Price TW, Barnes T. Comparing textual and block interfaces in a novice programming environment. In: Proceedings of the eleventh annual international conference on international computing education research. ACM; 2015. p. 91–9.
[10] Aivaloglou E, Hermans F. How kids code and how we know: an exploratory study on the scratch repository. In: Proceedings of the 2016 ACM conference on innovation and technology in computer science education; 2016. p. 53–61.
[11] Mota JM, Ruiz-Rube I, Dodero JM, Figueiredo M. Visual environment for designing interactive learning scenarios with augmented reality. In: Proceedings of 12th international conference on mobile learning; 2016. p. 67–74.
[12] Rodgers DL, Withrow-Thorton BJ. The effect of instructional media on learner motivation. Int J Instr Media 2005;32(4):333.
[13] Di Serio Á, Ibáñez MB, Kloos CD. Impact of an augmented reality system on students' motivation for a visual art course. Comput Educ 2013;68:586–96.
[14] Azuma RT. A survey of augmented reality. Presence: Teleoper Virtual Environ 1997;6(4):355–85.
[15] Seichter H, Looser J, Billinghurst M. ComposAR: an intuitive tool for authoring AR applications. In: 2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality; 2008. p. 177–8.
[16] Kato H, Billinghurst M. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In: Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on. IEEE; 1999. p. 85–94.
[17] Muñoz-Salinas R. ARUCO: a minimal library for augmented reality applications based on OpenCV. Universidad de Córdoba; 2012.
[18] J. Looser, R. Grasset, H. Seichter, M. Billinghurst, OSGART-A pragmatic approach to MR, 2006.
[19] Ledermann F, Schmalstieg D. APRIL a high-level framework for creating augmented reality presentations. In: Proceedings of the 2005 IEEE conference 2005 on virtual reality. IEEE Computer Society; 2005. p. 187–94.
[20] Haller M, Stauder E, Zauner J. AMIRE-ES: Authoring Mixed Reality once, run it anywhere. In: The 11th International conference on human-computer interaction; 2005. p. 22–7.
[21] MacIntyre B, Gandy M, Dow S, Bolter JD. 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. ACM; 2004. p. 197–206.
[22] Mazza R, Dimitrova V. CourseVis: A graphical student monitoring tool for supporting instructors in web-based distance courses. Int J Hum Comput Stud 2007;65(2):125–39.
[23] Lewis TL, Wyatt JC. mHealth and mobile medical apps: A framework to assess risk and promote safer use. J Med Internet Res 2014;16(9):1–8.
[24] Barab S, Squire K. Design-based research: putting a stake in the ground. J Learn Sci 2004;13(1):1–14.
[25] Davis FD, Bagozzi RP, Warshaw PR. User acceptance of computer technology: a comparison of two theoretical models. Manage Sci 1989;35(8):982–1003.

José Miguel Mota Macías is an assistant lecturer at the University of Cadiz (UCA). He received his Master's degree in Computer Science at the Universitat Oberta de Catalunya (Spain). He is also a PhD candidate researcher in technology-enhanced learning. In particular, his research areas are mobile learning, augmented reality and learning analytics.

Iván Ruiz Rube is an assistant lecturer at UCA, and received his PhD from the UCA. His fields of research are software process improvement, linked open data and technology-enhanced learning, and he has published several papers in these fields. He has previously worked as a software engineer for consulting companies such as Everis Spain S.L. and Sadiel S.A.

Juan Manuel Dodero has a Computer Science PhD from the Carlos III University of Madrid. He is currently an associate professor at UCA. He has been a lecturer of the Carlos III University of Madrid and has worked as a R&D engineer for several Spanish ICT companies. His main research interests are web science and engineering, and technology-enhanced learning.

Inmaculada Arnedillo-Sánchez is a professor of Learning Technologies at Trinity College Dublin. Her research explores theories and practices of teaching and learning to design, implement and evaluate technology-enhanced learning experiences. Her research includes: 1) mobile lifelong learning; 2) autism: the design and development of personalised adaptive systems and multiplayer 3D games; 3) privacy and ethics: the Big Brother phenomenon of automated data collection.