An overview of location-based game authoring tools for education

Spyridon Xanthopoulos and Stelios Xinogalos
University of Macedonia, Thessaloniki, Greece
spyridon.xanthopoulos@uom.edu.gr, stelios@uom.edu.gr

Abstract—Nowadays, computer literacy is common at a very early age. As mobile devices and new social communication trends are increasingly integrating in the modern digital lifestyle new opportunities arise for harnessing this potential for innovative serious purposes. Location-based games that layer educational activities with game play can help players gain a new perspective through active engagement. In this paper, a range of key issues for implementing location based games for educational purposes using the two most commonly used open source authoring tools ARIS and TaleBlazer are presented. Although the reviewed authoring tools vary in functionality, they share a fairly common conceptual model and in both cases the software architecture is comprised of the same components. Both platforms provide the required game mechanics to create sophisticated gameplay interactions and game logic extension. However, a number of issues should be carefully considered. The conceptual models are analyzed with a critical view, pointing out limitations and challenges that can inform the design and implementation of next generation authoring tools.

Keywords—location-based games, serious games, mobile learning

1 Introduction

Nowadays, computer literacy is common at a very early age as young people have spent almost their entire life interacting with diverse forms of computers and mobile devices. Over the last years technology has become predominantly mobile and ubiquitous; according to Gartner, global sales of smartphones to end users totaled 344 million units in the second quarter of 2016, a 4.3 percent increase over the same period in 2015. Furthermore, electronic games have become increasingly popular especially among young people; according to Google more than 1.5 billion apps and games are downloaded each month. As mobile devices and new social communication trends are increasingly integrating in the modern digital lifestyle, new opportunities arise for harnessing this potential for innovative serious purposes.

There is a considerable interest from researchers in exploiting the unique capabilities of mobile technologies to enhance innovative and engaging forms of learning, such as location-based games for educational purposes. Such games form a subcategory of location-based serious games (intended to support serious activities in a playful way) and the action is taking place in the real world enhanced with a virtual

space where the player's location obtains a central role in configuring game play. Although there is evidence in the literature reporting increased intrinsic motivation through engaging and enjoyable learning experiences, the use of pedagogical games still is controversial and educational practitioners decline to use them. Literature mentions several reasons that can pose barriers, such as missing teacher efficacy in using appropriate software [11, 12, 29], the association of games with extra workload and unclear learning results, and logistical issues such as lack of technical support and equipment [29]. Moreover, remarkably little research exists that directly addresses educational practitioners [12].

This paper aims to explore location-based games implemented for educational purposes from a technical perspective and how free and open source authoring tools can support the implementation of such games. The specific aims of this study are: to review the different types of location-based games described in literature that are applicable to learning; identify common technical requirements for authoring tools; evaluate free and open source authoring tools used in research with a critical view pointing out problems and challenges as well as presenting key issues and guidelines to inform the design of such tools.

The rest of the paper is organized as follows. In section 2, a review of the existing literature focusing on the area of practices, requirements and directives for tools used for authoring location-based experiences and games is presented. In section 3, typical paradigms of location-based games for educational purposes are also presented and a number of common requirements for authoring tools in order to support effectively the creation of games by non-programmers are identified in three groups: authoring functionality; admin tools; and end-user client app functionality. In section 4, the conceptual model adopted by the authoring tools ARIS and TaleBlazer is presented and analyzed. Finally, a comparative analysis of the tools is conducted, pointing out limitations and challenges that can inform the design and implementation of next generation authoring tools.

2 Related work

Although there is a plethora of authoring tools promising to ease the development of locative interacting experiences there is a limited number of surveys focusing on creative practices, requirements and directives for tools to author location-based experiences and games to inform next generation authoring tools.

Paelke et al. [21] provided an overview of the area of mobile location-based gaming and its relation to maps and presented key aspects of exemplary commercial and research location-based games and authoring tools of the past (Mediascape, Caerus). They concluded that in order to be effective, these tools should support an appropriate conceptual model and should be designed so as to operate within a structured process that also integrates with other tools for media production and content management.

Winter [28] conducted a literature review in order to inform the development of a location-based games authoring tool for secondary school children and their teachers.

Regarding educational aspects of location based games authoring the research question was how can location-based games authoring by students support their learning and how can it be integrated in educational practices (e.g. authoring in the context of ICT education emphasizes technical aspects of game development, requiring the use of various technologies to collect, share and manage resources). The discussed functionality and recommendations were synthesized into several generic design guidelines such as support for visual authoring and simulation mode.

In their survey Fidas et al. [8] presented five authoring tools for implementing cultural heritage experiences from a conceptual and architectural design (Hoppala, ARIS, TaggingCreaditor, LoCloud and CHEF). They concluded that a definition of a set of primitive elements is needed that can be used by cultural heritage experts as abstract building blocks without the help of professional developers and identified the need to streamline efforts with the aim of creating meta-models of such primitive elements.

Brundell et al. [5] focused upon the working practices of individuals, small independent artists and researchers, rather than designers or developers in large commercial companies or small to medium sized enterprises. A qualitative analysis of findings from the study was conducted to inform user requirements and design of next-generation authoring tools through a process of co-design.

3 Location based games for education

Location-based games for education form a subcategory of location-based serious games intended to support serious activities in a playful way. Several location-based games within the genre of serious games have been developed mostly to support: *cultural heritage* purposes, such as supporting historical teaching and learning [2, 4, 16]; *museum* visits, experiencing the history of sites and in some cases participate in events and rituals that took place in ancient times [18, 22, 25, 26]; *simulations*, such as flood preparedness and fire evaluation scenarios [14, 19, 20]; *transportation*, exploring the symbiotic potential of social games in combination with location-based games and *tourism*, helping visitors engage with history and culture of their destination [10, 24].

Typical paradigms of location-based games for educational purposes have been developed mostly to support *situational language learning*. Learners are assigned missions (such as finding a specific book in the library) through location based game interaction activities. The learning goal of the games is to enhance English learning (listening and speaking), interest and motivation in a real environment. The conducted activities involve use of Bluetooth, indoor positioning, camera, QR codes attached to walls (revealing web links associated with relevant learning) with the aim to construct mixed reality game learning environments capable of integrating virtual objects with real scenes [6, 9, 13, 15, 17].

Other research efforts focused on engaging in *mathematical activities* and developing conceptual understanding on several topics. In the first case learners are expected to deepen their experiential knowledge of geometrical concepts related to

shapes and orientation/navigation. In the exemplary game MobileMath [27] researchers observed the team activities during the game by accompanying a team outside as a participating researcher or by watching all teams in the game in real time on the website. The game was played by at least two teams with each team creating geometrical shapes on a previously defined playing field using a mobile phone with GPS functionality and an on-screen map. In the second case researchers focused on developing conceptual understanding on topics such as science, problem-solving and collaboration. In the exemplary game Outbreak [23] students used GPS handhelds to work together in groups of three in order to find the antidote to a pretend disease. As the students walk around their school's campus, each person took on the role of a different scientist in order to collect and share data with each other.

The proposed systems are based on the *situational learning approach* [6]. In the general case the system provides clues that learners have to identify (e.g. via listening in the case of language learning) where to go next in order to proceed in the game. The player's location is identified with the use of indoor and outdoor positioning techniques, activating game agents and other game mechanics.

A number of common requirements for authoring tools in order to support effectively the creation of games by non-programmers can be identified in three groups:

Authoring functionality – support for non-linear stories, organizing tasks as questions and answers with conditional branching, visual authoring, re-use of games, customization and personalization of the user interface to individual needs [28]; support for map-based authoring of game content considering also the alternative of on-site authoring within the physical environment, bridging the gap between desktop authoring and outdoor [21]; support for visual programming.

Admin tools – support for simulation mode to test mobile content and game flow; support for customized game analytics for assessing learning performance (e.g. results about a summative test with multiple choice questions [6]).

End-user client app functionality – support for communication (such as messaging or chat), support for multiple media formats (e.g. text, video, images, audio, HTML) [28]; support for the major mobile operating systems Android and iOS; support for augmented reality; support for QR Codes.

In the following section the conceptual model adopted by the most common open source authoring tools is presented and analyzed with a critical view and a comparative analysis is contacted with regard to the synthesized list of requirements.

4 Authoring location-based games

Various research attempts tried to develop tools and technologies in order to encourage the creation of location-based experiences by non-technical users. The platforms COLLAGE[1], Games Atelier[2], ROAR[3] and TOTEM[4] were developed in the

[1] http://www.celekt.info/projects/show/14
[2] http://waag.org/en/project/games-atelier
[3] http://gameslab.radford.edu/ROAR/

context of research projects in the past and successfully contributed towards the creation of innovative approaches of location-based playing and learning experiences. However, the aforementioned platforms are no longer available.

MAGELLAN[5] is a promising 4 year project (2013-2017) aiming at researching and implementing an authoring and gaming platform based on visual authoring principles. The main objective of the platform is to enable non-programmers to cost-effectively author and publish multi-participant location-based experiences, as well as to support the browsing and execution of a massive number of such experiences [1, 7]. Currently, the platform is implemented as a proof-of concept [3]; however, access to documentation and platform functionality is not fully open to the general public.

7Scenes[6] is another platform for creating location-based experiences used in pedagogical settings mentioned in literature that started as a spinoff commercial product incubated at the Waag Society of the Netherlands. The community platform is targeting sightseeing tours, providing simplistic game like elements (time limit and a point system based on multiple choice questions). Finally, popular tools such as App Inventor[7] can support the implementation of multi-purpose mobile apps; however, no off-the-shelf authoring functionality (or other form of community extension) is provided for implementing location-based experiences or games.

Currently, commonly used authoring tools that actively support location-based software development providing open source code and free of charge services are ARIS[8] and TaleBlazer[9]. In the following sections, the conceptual model adopted by ARIS and TaleBlazer is presented and analyzed. ARIS is being developed by Wisconsin-Madison University, and TaleBlazer is being developed by MIT Scheller Teacher Education Program lab. Both platforms provide a way for creating mobile interactive stories and augmented location-based games.

4.1 ARIS

ARIS is an open source, free, cloud-based platform for creating and playing location-based augmented reality experiences on iOS devices. The conceptual model comprises of objects, triggers and scenes.

Objects – are containers for the content that players can see and interact with. Game objects provide support for starting conversations, inspecting items, viewing plaques and visiting web pages.

Triggers – provide a way of connecting actions in the physical world to ARIS objects. There are currently five types of triggers: *location triggers*, fired when the player is close enough to the geospatial coordinates of the trigger; *QR Code triggers*, fired when the players scan the image or enter the corresponding string code; *timer triggers*, fired when a particular period of time elapses; *beacon triggers*, fired

[4] http://www.totem-games.org/
[5] http://www.magellanproject.eu/
[6] http://7scenes.com
[7] http://appinventor.mit.edu/explore
[8] http://www.arisgames.org
[9] http://taleblazer.org

according to the player's proximity to a Bluetooth beacon (iBeacon[10]); *AR View triggers*, fired when specific views or images of the real world are recognized and trigger game objects (like images or video) that replace targets atop of the camera feed.

Scenes – are containers for collection of game objects. Scenes provide support for designing stories and help authors think about the progression of the game. They act as organizational units and represent different parts of the game.

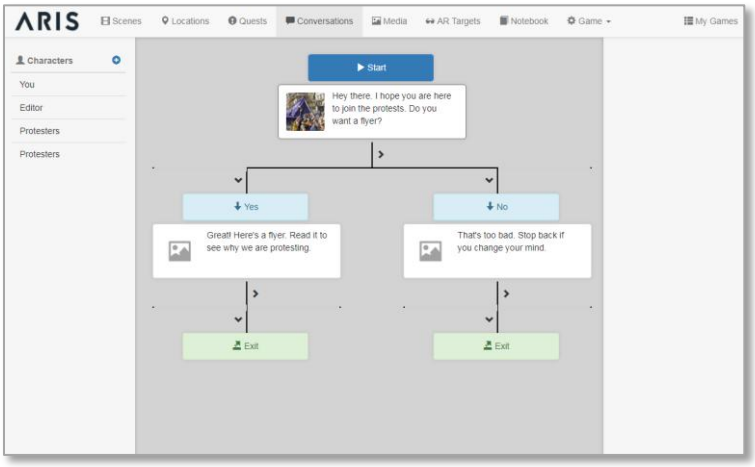

Fig. 1. ARIS Conversations Editor

ARIS has a story-based structure. The ARIS editor is browser-based with no local installation required. The platform allows authors to build *conversations* between the player and visual characters with the use of diagrams that help authors to visualize the flow (figure 1). The author defines the *lines* (text spoken by the character), and the *choices* for the player. Each *line* can also modify the player's environment by taking or giving items or setting the value of custom *attributes* defined by the author. ARIS allows authors to further customize game functionality (e.g. playing an audio clip in the background) based on the use of a JavaScript API. The custom code is executed when the platform renders HTML texts either entered or displayed to the player (e.g. presenting conversation lines).

4.2 TaleBlazer

TaleBlazer is an open-source, free, cloud-based platform for creating and playing location-based experiences on Android or iOS devices. The conceptual model comprises of agents, regions, scenarios and roles.

Agents – represent characters or objects that the player can interact with. The behavior of the agents is configured through *actions* that appear as buttons on the agent's dashboard. There are currently four types of actions that become activated

[10] https://developer.apple.com/ibeacon

when the player bumps into the agent: *text*, in the form of rich text format; *video*, playing an uploaded or YouTube video; *built-in functionality*, pickup and drop actions; *script*, executing a custom blocks-based script.

Regions – are real-world locations where the game takes place (figure 2).

Scenarios – are different versions of the same game that the players can pick from when they start the game (e.g. easy or hard). The game designer can use the player's choice to control the game logic with the use of block-scripting. The game designer can configure also multiple **roles** for the game in order to specify different interactions for players playing different roles through the use of block-scripting. The player picks the role when starting the game in order to access the role specific functionality.

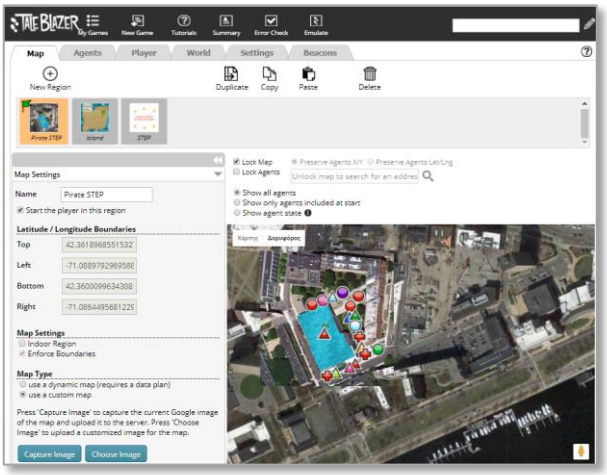

Fig. 2. TaleBlazer Editor

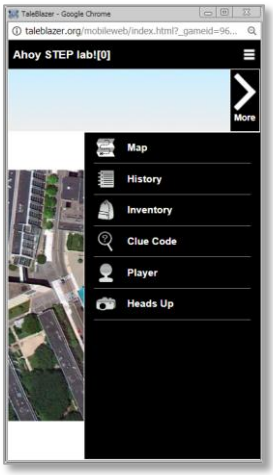

Fig. 3. TaleBlazer Emulator

TaleBlazer has a role-based structure. The TaleBlazer editor is browser-based with no local installation required (figure 2). The platform allows authors to define custom *traits* (for agents, roles or the game) which can be modified through block-based scripting. The editor uses a visual blocks-based scripting language that allows authors to create rich interactivity while helping authors avoid syntax errors. The interaction with the player is configured through *actions* that appear as buttons on the agent's dashboard. The platforms allow authors to modify the visibility of an action during game play through the use of blocks-based scripting. The game can be tested out with the use of an HTML emulator (figure 3) decreasing drastically debugging time.

5 Comparative analysis

In comparison to ARIS, TaleBlazer has a role-based structure. The use of traits provides an intuitive means to extend the data model with custom properties that can be easily modified through the use of blocks-based code. Furthermore, the visual

blocks-based scripting language allows authors to create rich interactivity while helping users avoid syntax errors. On the other hand, ARIS has a story-based structure that helps authors to visualize game flow and provides convenient ways to segment the game.

In Table 1, key issues are synthesized in three groups: admin tools, authoring functionality and end-user client app features.

Table 1. ARIS and TaleBlazer features

Feature	ARIS	TaleBlazer
Authoring functionality		
Re-use games	Yes	Yes
Web visual authoring	Yes	Yes
Map-based authoring	Yes	Yes
Visual programming	No	Yes
Customization and extension	JavaScript API	Blocks–based scripting
In-situ authoring	No	No
Personalization	No	No
Non-linear stories	Yes	Yes
Admin tools		
Simulator	No	Yes
Analytics	No	Yes, only to officially featured organizations
End-user client app		
Mobile OS	iOS	Android, iOS
Connection needed	Yes	No
Augmented reality features	Yes	Yes
Multiplayer	Yes	No
Indoor navigation	Yes	Yes
QR Code	Yes	No
Tap to bump mode	Yes	Yes
On-screen map navigation	Yes	Yes
Communication	No	No
Multiple multimedia formats	Yes	Yes

5.1 Authoring functionality

Both platforms provide authors with visually authoring map-based editors to create and re-use games. Although new games can be created as duplicates of existing ones (TaleBlazer *remixes* games, ARIS *duplicates* games), there seems to be no functionality to create libraries as collections of common scripts (or other resources) that can be imported in new games.

The extension of the game model is allowed by both tools through the use of custom properties which can be modified during game play providing also a way to

control game flow (e.g. enabling or disabling actions according to the value of a property). The TaleBlazer visual blocks-based scripting language allows authors to create rich interactivity between game entities while helping authors avoid syntax errors. Although ARIS has no blocks-based scripting functionality, it provides sophisticated interaction management and various functions without the need for coding. However, the use of a JavaScript API for further customizing game functionality requires technical experience and effort to master.

In-situ authoring is not supported, neither personalization of both editor environments.

5.2 Admin tools

TaleBlazer provides a mobile emulator to test the mobile content quickly and error check functionality to evaluate warnings and errors caused by invalid blocks-based code. TaleBlazer provides also a predefined set of analytics (such as total gameplays, duration times and average game play) and custom analytics events only to officially featured organizations. TaleBlazer *analytics events* can be used to collect specific data such as which choices players make or how many points they gained in the game.

5.3 End-user client app

Both platforms rely on an end-user client-app for playing the games; ARIS supports only the iOS operating system, while TaleBlazer runs on either iOS or Android devices. Augmented reality features are implemented by the platforms in different ways. TaleBlazer provides a “heads up” functionality to show nearby agents as markers overlaid onto the video camera display based on the compass and GPS readings. On the other hand, ARIS uses “AR targets” created by Vuforia[11]. The player chooses views or images in the real world, recognized by the platform through the device's camera. The target views are replaced by the platform with custom media like images of video on the screen atop the existing camera feed.

Multiplayer and QR scanning functionality are available only to ARIS platform while both platforms provide indoor navigation, tap to bump and on-screen map navigation functionality.

6 Conclusions

Learning through mobile devices has been an active research area in recent years. Location-based games are estimated as a promising practice and have become an important strand to interact with the real world and use on learning environments. Various games have been developed in recent years and used to enhance the education and training efficiency of students in several subjects such as situational language learning, mathematical concepts, and conceptual understanding on topics

[11] https://vuforia.com/, https://fielddaylab.wisc.edu/courses/aris-ar

such as science, problem-solving and collaboration. The potential of currently available authoring tools needs careful consideration in order to address effectively the diverse design and technical requirements posed by location-based games for educational purposes.

In this paper, a range of key issues for implementing location based games for educational purposes using two publicly available open source authoring tools, ARIS and TaleBlazer are presented. Although the reviewed authoring tools vary in functionality they share a fairly common conceptual model and in both cases the software architecture is comprised of the same components: a browser-based editor to author the game structure and mechanics, a back-end system to support concurrency and persistence and an end-user client-app for playing the games. Although both platforms provide the required game mechanics to create sophisticated gameplay interactions and game logic extension, a number of issues should be carefully considered.

Cross-platform development – in order to target as many possible mobile operating systems, the development of the end-user client app could be based on cross-platform development frameworks. Frameworks such as Ionic[12] provide free and open source mobile SDK, mobile components, typography, interactive paradigms and an extensible base theme for building and customizing apps for Android or iOS [30].

Interoperability and standardization – the fact that interoperability between authoring tools is not feasible, stresses the need for standardization for packages, libraries, game templates and integration with external systems (e.g. many universities are using single sign on systems for the authentication process).

Extension and customization – although both platforms provide sophisticated interaction management and default functionality, several means for extending the game logic should be provided by the authoring platform with the use of high-level scripting tools, templates and themes, and the provision of interfaces in order to customize and further extend the platform's core functionality and look & feel.

Testing – the integration testing process should provide the required means to evaluate the correct wiring of game container contexts and flow. TaleBlazer provides simulation functionality and evaluation of the custom blocks-based code, but further utilities are needed to support automated runtime unit tests for the entire application.

Analytics and reporting – assessing learning performance should be supported by customized analytics (e.g. results about summative tests in the game), logging and feedback (e.g. about exception or breakpoint conditions).

In conclusion, although several design and technical aspects of the location-based games were considered in this research, further consideration with regard to e-learning standardization and effective ways for embedding learning content must be considered. Future research should also focus on game design patterns specifically for location-based games and how they relate to learning outcomes. Games and tools need to be analyzed with a critical view, pointing out limitations, problems and challenges that can inform the design and implementation of next generation authoring tools.

---

[12] https://ionicframework.com/

7 References

1. Anastasiadou, Despina, and Petros Lameras. "Identifying and classifying learning entities for designing location-based serious games." Semantic and Social Media Adaptation and Personalization (SMAP), 2016 11th International Workshop on. IEEE, 2016.
2. Anderson, E. F., McLoughlin, L., Liarokapis, F., Peters, C., Petridis, P., & De Freitas, S. (2010). Developing serious games for cultural heritage: a state-of-the-art review. Virtual reality, 14(4), 255-275.
3. Balet, O., Koleva, B., Grubert, J., Yi, K. M., Gunia, M., Katsis, A., & Castet, J. (2017). Authoring and Living Next-Generation Location-Based Experiences. arXiv preprint arXiv:1709.01293.
4. Bellotti, F., Berta, R., De Gloria, A., D'ursi, A., & Fiore, V. (2012). A serious game model for cultural heritage. Journal on Computing and Cultural Heritage (JOCCH), 5(4), 17.
5. Brundell, P., Koleva, B., & Wetzel, R. (2016, October). Supporting the design of location-based experiences by creative individuals. In Semantic and Social Media Adaptation and Personalization (SMAP), 2016 11th International Workshop on (pp. 112-116). IEEE.
6. Chen, C. M., & Tsai, Y. N. (2009, July). Interactive location-based game for supporting effective English learning. In Environmental Science and Information Application Technology, 2009. ESIAT 2009. International Conference on (Vol. 3, pp. 523-526). IEEE.
7. Clarke, S., Lameras, P., Dunwell, I., Balet, O., Prados, T., & Avantangelou, E. (2015, October). A Training framework for the creation of location-based experiences using a game authoring environment. In European Conference on Games Based Learning(p. 125). Academic Conferences International Limited.
8. Fidas, C., Sintoris, C., Yiannoutsou, N., & Avouris, N. (2015, July). A survey on tools for end user authoring of mobile applications for cultural heritage. In Information, Intelligence, Systems and Applications (IISA), 2015 6th International Conference on (pp. 1-5). IEEE.
9. Godwin-Jones, R. (2014). Emerging technologies games in language learning: Opportunities and challenges. Language Learning & Technology, 18(2), 9-19.
10. Grüntjens, D., Groß, S., Arndt, D., & Müller, S. (2013). Fast authoring for mobile game-based city tours. Procedia Computer Science, 25, 41-51.
11. Kebritchi, M. (2010). Factors affecting teachers' adoption of educational computer games: A case study. British Journal of Educational Technology, 41(2), 256-270.
12. Ketelhut, D. J., & Schifter, C. C. (2011). Teachers and game-based learning: Improving understanding of how to increase efficacy of adoption. Computers & Education, 56(2), 539-546.
13. Liu, T. Y., & Chu, Y. L. (2010). Using ubiquitous games in an English listening and speaking course: Impact on learning outcomes and motivation. Computers & Education, 55(2), 630-643.
14. Mannsverk, S. J. (2013). Flooded-A Location-Based Game for Promoting Citizens' Flood Preparedness.
15. Meyer, B., & Sørensen, B. H. (2009). Designing serious games for computer assisted language learning–a framework for development and analysis. In Design and use of serious games (pp. 69-82). Springer Netherlands.
16. Mortara, M., Catalano, C. E., Fiucci, G., & Derntl, M. (2013, October). Evaluating the effectiveness of serious games for cultural awareness: the Icura user study. In International Conference on Games and Learning Alliance (pp. 276-289). Springer International Publishing.

17. Mota, J. M., Ruiz-Rube, I., Dodero, J. M., & Figueiredo, M. (2016). Visual Environment for Designing Interactive Learning Scenarios with Augmented Reality. International Association for Development of the Information Society.
18. Nilsson, T., Blackwell, A., Hogsden, C., & Scruton, D. (2016). Ghosts! A Location-Based Bluetooth LE Mobile Game for Museum Exploration. arXiv preprint arXiv:1607.05654.
19. Oliveira, M., Pereira, N., Oliveira, E., Almeida, J. E., & Rossetti, R. J. (2015). A Multi-player Approach in Serious Games: Testing Pedestrian Fire Evacuation Scenarios. Oporto, DSIE15, January.
20. Onencan, A., Kortmann, R., Kulei, F., & Enserin, B. (2016). MAFURIKO: Design of Nzoia Basin Location Based Flood Game. Procedia Engineering, 159, 133-140.
21. Paelke, V., Oppermann, L., & Reimann, C. (2008). Mobile location-based gaming. Map-based Mobile Services, 310-334.
22. Peng, S. T., Hsu, S. Y., & Hsieh, K. C. (2015). An Interactive Immersive Serious Game Application for Kunyu Quantu World Map. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2(5), 221.
23. Rosenbaum, E., Klopfer, E., & Perry, J. (2007). On location learning: Authentic applied science with networked augmented realities. Journal of Science Education and Technology, 16(1), 31-45.
24. Rossetti, R. J., Almeida, J. E., Kokkinogenis, Z., & Gonçalves, J. (2013). Playing transportation seriously: Applications of serious games to artificial transportation systems. IEEE Intelligent Systems, 28(4), 107-112.
25. Rubino, I., Barberis, C., Xhembulla, J., & Malnati, G. (2015). Integrating a location-based mobile game in the museum visit: Evaluating visitors' behaviour and learning. Journal on Computing and Cultural Heritage (JOCCH), 8(3), 15.
26. Sanchez, E., & Pierroux, P. (2015, October). Gamifying the Museum: Teaching for Games Based Informal Learning. In European Conference on Games Based Learning (p. 471). Academic Conferences International Limited.
27. Wijers, M., Jonker, V., & Drijvers, P. (2010). MobileMath: exploring mathematics outside the classroom. ZDM, 42(7), 789-799.
28. Winter, M (2009). Location Based Games Authoring. Literature review to inform the development of a location based games authoring tool for secondary school children. http://www.cem.brighton.ac.uk/staff/mw159/winter_2009_location_based_games_authoring.pdf. Last accessed on 29 August 2017.
29. Williamson, B. (2009). Computer games, schools, and young people: A report for educators on using games for learning. Bristol: Futurelab. https://www.nfer.ac.uk/publications/FUTL27/FUTL27.pdf. Last accessed on 29 August 2017.
30. Xanthopoulos, S., & Xinogalos, S. (2013, September). A comparative analysis of cross-platform development approaches for mobile applications. In Proceedings of the 6th Balkan Conference in Informatics (pp. 213-220). ACM.