Taleblazer vs. Metaverse: a comparative analysis of two platforms for building AR location-based educational games

Alexandros Kleftodimos*

Department of Communication and Digital Media,
University of Western Macedonia,
52100, Kastoria, Greece
Email: akleftodimos@uowm.gr
*Corresponding author

Georgios Lappas

Department of Communication and Digital Media,
University of Western Macedonia,
52100, Kastoria, Greece
and
Hellenic Open University,
Patra, Greece
Email: glappas@uowm.gr

Michalis Vrigkas

Department of Communication and Digital Media,
University of Western Macedonia,
52100, Kastoria, Greece
Email: mvrigkas@uowm.gr

Abstract: Location-based AR games are becoming increasingly popular in education. With location-based AR games, learners can obtain knowledge by visiting places of educational value through informative digital content that is activated and displayed on their mobile devices when specific locations are reached. To create location-based AR games, there are several available authoring tools. Taleblazer and Metaverse Studio are two popular platforms that are used nowadays by many educators. This study aims to perform a comparative analysis between these platforms to provide educators interested in developing location-based AR experiences with all the information needed to make an informed decision on which platform to use. The analysis examines the designer environment and its available features, the end-user interface, the documentation that accompanies each platform, and third-party applications that are developed by these tools. Furthermore, two game prototypes have been developed to better understand the two platforms' functionality.

Keywords: location based games; augmented reality; education; Taleblazer; Metaverse Studio.

Reference to this paper should be made as follows: Kleftodimos, A., Lappas, G. and Vrigkas, M. (2022) 'Taleblazer vs. Metaverse: a comparative analysis of two platforms for building AR location-based educational games', *Int. J. Entertainment Technology and Management*, Vol. 1, No. 4, pp.290–310.

Biographical notes: Alexandros Kleftodimos is an Assistant Professor at the Communication and Digital Media Department, University of Western Macedonia, Greece. He holds a PhD in Computer Science from the Department of Applied Informatics, University of Macedonia, Greece. His research interests and published work are in the areas of educational technologies, multimedia, web mining, social informatics, e-government, and e-politics. He has also participated in several national and EU-funded research and development projects. In the past, he also worked as an informatics secondary education teacher, software developer, and analyst.

Georgios Lappas is a Professor in Social and Political Informatics at the Communication and Digital Media Department, University of Western Macedonia, Greece. He served as the Chair of Department, Dean of Digital Media and Strategic Communication Lab, Dean and Vice-Dean of the MA 'Public Relations and Marketing with New Technologies' and Vice-Dean of the MSc 'Public Discourse and Digital Media'. He published more than 100 articles and served as PC member at more than 100 international conferences. He participated in many national and EU funded projects. His research interests include social informatics, e-marketing, web mining, opinion mining, e-government, and e-democracy.

Michalis Vrigkas is an Assistant Professor at the University of Western Macedonia at the Department of Communication and Digital Media, Greece. He received his PhD from the Department of Computer Science and Engineering, University of Ioannina, Greece, and his MSc and BSc in Computer Science from the same institution. For the academic year 2018–2019, he was an Adjunct Lecturer at the Department of Computer Science and Engineering, University of Ioannina. In the past, he has spent two years (2016–2018) at the University of Houston, TX, USA where he worked as a Postdoctoral Fellow at the Computational Biomedicine Lab. His research covers a wide range of topics such as computer vision, augmented and virtual reality, image and video processing, and machine learning.

1 Introduction

Games have been traditionally used in education for increasing learner motivation and satisfaction. Location-based AR games (LBARGs) are a relatively recent trend and are gaining increasing popularity since the launch of Pokémon by Niantic in July 2016 (https://pokemongolive.com/). Since then, numerous companies have attempted to recreate the consumers by developing their own location-based AR Games. Undeniably, LBARGs have been commercially deployed and proved to be successful in capturing players' interest. Other popular LBARGs that have gained millions of installations besides Pokemon Go are Ingress, Draconius GO, Jurassic World: Alive, The Walking Dead: Our World, and Landlord Tycoon-Real Estate (Laato et al., 2019).

Besides entertainment LBARGs are also used in other fields such as education and tourism. In tourism, for example, LBARGs are used to enhance the tourism experience by adding a gaming dimension. These types of games have also attracted the interest of many tourism researchers (e.g., Lacka, 2020; Nóbrega et al., 2017; Weber, 2017; Weber and Dickinson, 2018). In education, LBARGs promote active learning through the parallel use of a mobile application and the exploration of the physical environment (e.g., Mozelius et al., 2019; Söbke et al., 2019). Location-based games (LBGs) increase learner satisfaction by adding excitement and stimulation to the learning process (Avouris and Yiannoutsou, 2012). With LBARGs learners can obtain knowledge by visiting places of educational value through informative digital content activated and displayed on their mobile device when specific locations are reached. The learner's location is tracked through the GPS sensor of the mobile device, and the scenario of the game can take place in areas of historical, cultural, and environmental interest. It is understandable that applications for tourism can also have educational challenges. Educators can organise scavenger hunts or create 'learning journeys' by setting up geographical location points where digital content is activated. The digital content can be in various multimedia forms such as text, images, videos, audio, 3D models, 2D and 3D animations. Learners can be presented with information, and the knowledge gained can be tested by interactive activities such as multiple-choice questions. In these games, learners can gather points and compete with one another by answering questions correctly or by collecting virtual artifacts. According to many studies, educational activities involving physical activity have proved to be efficient in learning (Norris et al., 2020).

This study aims to find and compare popular platforms used for implementing mobile AR LBGs and, specifically, platforms intended for designers with little or no technical expertise. Such platforms would be suitable for most educators. Specifically, two authoring tools will be examined and compared: Taleblazer and Metaverse Studio. The reason for selecting these two platforms is that they are both popular and are currently being used by many educators worldwide for designing location-based AR learning experiences. Furthermore, these platforms do not require advanced technical expertise, and hence they are suitable for novice and amateur game designers. Both tools can also be used for delivering applications for both IOS and Android mobile devices.

Finally, it can be said that the study aims to provide all the necessary information to educators interested in developing location-based AR games with these two platforms, so they can make informed decisions in selecting the tool that best suits their needs.

Taleblazer (http://taleblazer.org/) is an open-source platform developed by MIT (Massachusetts Institute of Technology) and Metaverse Studio (https://studio.gometa.io/discover/me) is a platform created by GoMeta an American software company headquartered in San Diego.

The comparative analysis focuses on issues such as the designer environment of both platforms and the available functionality, the game mechanics that are supported, as well as the end-users environment.

2 Authoring tools and comparative analysis studies

Various solutions are encountered in the literature regarding the authoring tools for developing LBARGs. Platforms such as COLLAGE, Games Atelier, ROAR, and TOTEM were mainly developed in the context of research projects in the past and successfully contributed to the creation of innovative location-based learning experiences (Xanthopoulos and Xinogalos, 2017). Most of these platforms, however are no longer available. ARIS is also a platform used by designers of location-based AR learning games. ARIS, however is only available for IOS devices, leaving out a significant proportion of end-users with Android devices.

Before moving on to examine the capabilities of the two platforms, it is worth mentioning that there have been other studies in the past that perform a comparative analysis between authoring tools used for creating LBAR games. For example, in Metikaridis and Xinogalos (2021), the authors perform comparative analysis for Taleblazer, Aris and Unity, and Mapbox. In this study, the authors based the analysis on evaluation metrics that were derived from the literature and the practical experience gained through the development of location-based AR prototype applications. The metrics on which the comparative analysis was based on have mostly to do with the affordances of the developer environment (e.g., capabilities for visual authoring and visual programming, re-use and re-editing, editor customisation, simulation mode etc.). In Fidas et al. (2015), the authors provide an overview of LBGs authoring tools ARIS, CHEF, Hoppala, LoCloud, and TaggingCreaditor from the viewpoint of implementing cultural heritage experiences. The authors briefly present aspects of the user interaction design as well as the architectural and technological design of the authoring tools. In Siakavaras et al. (2018), TaleBlazer, ARIS, 7scenes, and Wherigo were briefly presented and compared. Their research aimed to define the role of mobile games in teaching the subject of information and communications. The comparison was based on seven features. These features were role-playing support, assessment support, multimedia support, the ability to incorporate QR codes, the ability to store data that players may record such as photos and videos, the use of open-source technologies, and finally, the ability to create multiplayer games. It is worth mentioning that the level of analysis was different between the above studies. Metikaridis and Xinogalos (2021) provide a much more detailed analysis when compared to the other studies. Furthermore, the authors of this study also implement prototype applications to understand the tools' functionalities better and construct a reliable comparison framework. Metikaridis and Xinogalos (2021) also mention that although several LBG authoring tools have been developed and analysed in the literature, the studies that comparatively examine such tools are scarce.

In the studies mentioned above, the authors present the advantages and disadvantages of each authoring tool rather than declaring a winner. For example, Siakavaras et al. (2018) findings showed that all of the examined tools support the assignment of roles to players. Furthermore, ARIS and seven scenes provided the ability to develop multiplayer games, incorporate QR codes, and store user data. Metikaridis and Xinogalos (2021) also present the advantages and disadvantages of the authoring tools examined. For example, according to the authors, a disadvantage of the ARIS tool is the existence of minor editor bugs, which are sometimes visible during game development. On the other hand, the advantages of ARIS are:

a being an open source tool that publicly exposes a part of its code base in GitHub repositories

b having an active community

c having a large number of video tutorials on YouTube to support the educators in the development process.

Thus, it is evident that the studies mentioned above aim to provide information to anyone interested in developing LBAR games (e.g., educators and developers) so they can make informed decisions in selecting a tool that best suits their needs rather than declaring a winner.

The current study performs a comparative analysis that takes into consideration another popular platform that hasn't been included in past studies. Specifically, this study examines Metaverse Studio, a popular platform that requires no programming knowledge and, therefore, a platform that is user-friendly for many educators. Metaverse Studio as mentioned in some general statistics reported on a tutorial video (Metaverse Teacher Professional Development Session) released in January 2019 on the Metaverse Studio YouTube channel (https://www.youtube.com/watch?v=MLeZo7X5rnA&t=2495s) was used for creating 165,000+ experiences, across many subjects and grade levels and used for developing experiences in 179 countries around the world till that date.

3 Research methodology

As it becomes evident from the previous section, the few studies that perform a comparative analysis of LBAR authoring tools follow different methodologies. All of these studies however base the comparative analysis on the tool affordances provided to the developer, and the features of the end-user interaction interface. Furthermore, as Metikaridis and Xinogalos, (2021) mention in their research, most comparative analysis studies that were conducted in the past were based on the literature and/or the manuals of the tools and not on empirically studying the tools through the development of games. The current study also attempts to follow a similar approach. Furthermore, the presence of similar functionality features such as the ones mentioned in previous studies was also examined. For example, this study also examines the presence of visual authoring or visual programming capabilities, the ability to support role playing games, the ability to re-use & re-edit games created by other users, the ability to perform game simulation tests without the need to be present at the actual game site, the ability to perform data collection and game analytics, the ability to store player data such as photos and videos and so on. However, the study also examines other features that are specific either to Metaverse Studio or Taleblazer to perform the comparative analysis (e.g., types of multimedia elements that can be incorporated into the applications, use of custom maps, need for internet connection, etc.).

More specifically, a series of steps were followed to perform the comparative analysis between the two platforms. First, the designer and end-user environments were examined thoroughly by the authors of this study. Then the documentation manuals of two platforms were studied. The documentation manuals were found on the respective websites. Furthermore, several publications from the relevant literature were also studied (e.g., MacCallum and Parsons, 2019; Metikaridis and Xinogalos, 2021). The next step was to download and test third-party applications built using these two platforms. This step was carried out to explore the end-user environment's capabilities further.

At last, two prototype AR LBGs were created using the Taleblazer and the Metaverse Studio platforms. This was again done in order to obtain a broader understanding of the software functionalities. The prototype experiences unfold in a historical region with cultural heritage monuments in the city where the Department of Communication and Digital Media, of the University of Western Macedonia, is based, named Kastoria.

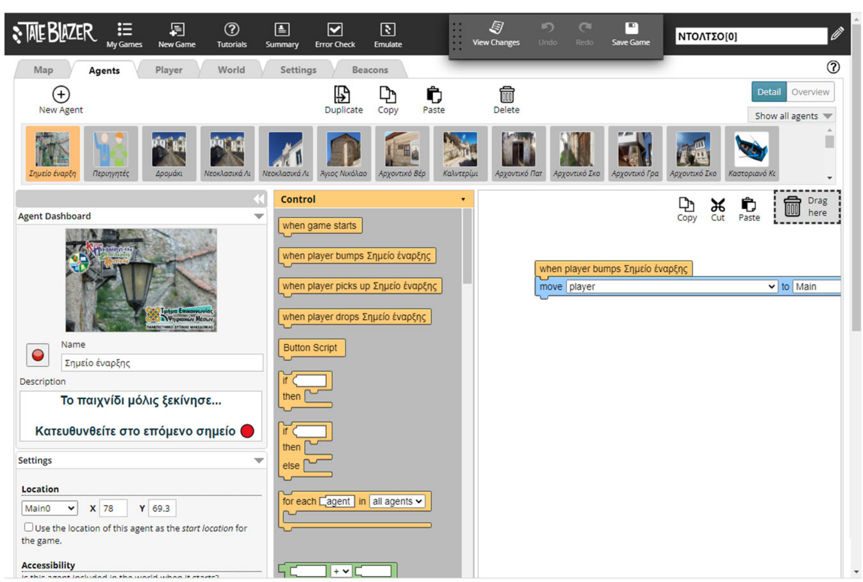
4 Comparative analysis

Both Taleblazer and Metaverse provide the mechanics to develop AR location-based interactive games. Nevertheless, significant differences exist in the designer and end-user environments, the learning curve of each platform, and the possible features that may be present in the games developed with these platforms. Before moving to the details of each platform, we must point out that TaleBlazer is exclusively designed for location-based augmented reality games. Metaverse Studio, on the other hand, is not solely designed for building ARLBGs but can also support the development of mobile apps that are not location-based. In Metaverse Studio, the term 'experience' is used for the developed applications, so this term will also be used when referring to applications that are designed with this tool.

4.1 Designer environment

Before we move on to the details of each platform, it is worth mentioning some common features that both platforms have. First, the designer environment in both platforms is browser-based, meaning there is no need to download a software package and perform an installation to start designing games. However, an internet connection is needed to access the platform website and use the development environment. Both applications allow interactive game designers to share their work with other designers/educators. In other words, interested educators can copy existing projects and remix them to produce new and unique experiences. Furthermore, both platforms can be used to create LBAR experiences for both IOS and Android operating systems.

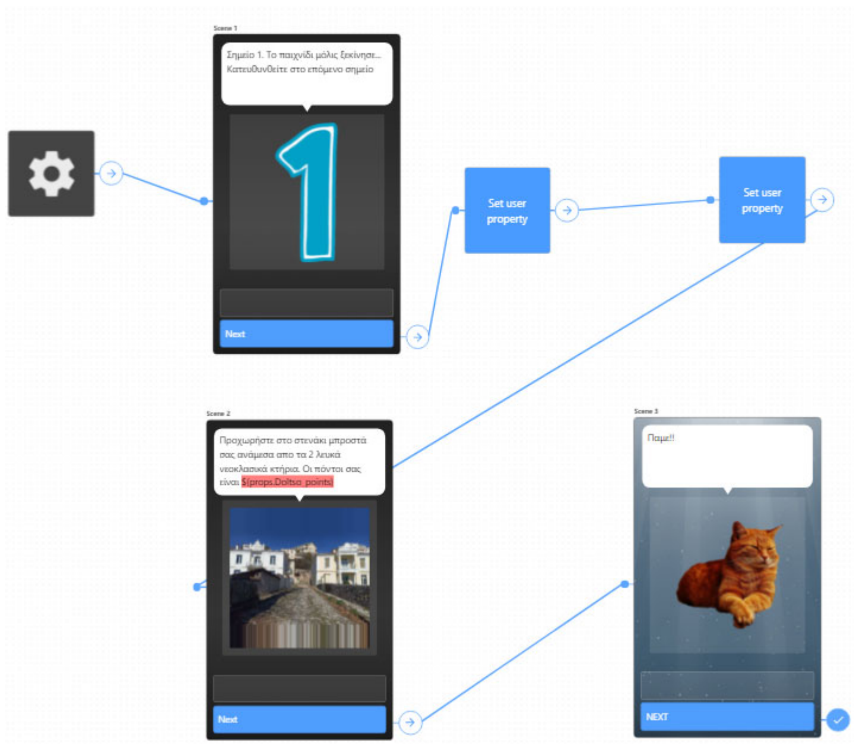
Figure 1 Taleblazer visual block-based scripting language (see online version for colours)



Note: The programming blocks are on the left panel of the environment, while the game scenario is developed on the right panel by using the appropriate blocks.

The Taleblazer development environment relies on a visual block-based scripting language (Figure 1). To program a location-based learning game the educator must be (or willing to get) acquainted with a visual block-based language which is very similar to Scratch (https://scratch.mit.edu/), another known product of MIT. The visual block-based programming environment of Taleblazer is suitable for people with no prior knowledge of programming. Visual block-based programming environments are extensively used for introduction to programming and computational thinking and are also ideal for young kids. So, it is feasible for an educator without previous programming experience to get started on designing interactive mobile applications with Taleblazer and its visual block-based programming environment.

Figure 2 Metaverse Studio experience elements: scenes and blocks (see online version for colours)

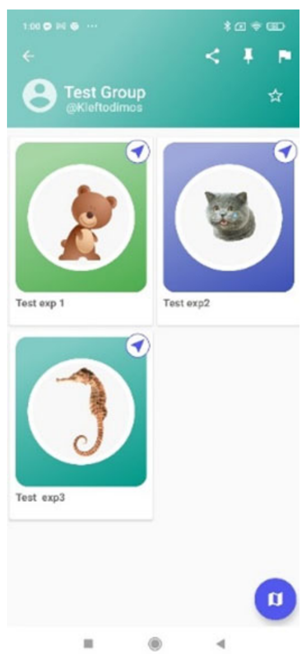


Note: Scenes and blocks are linked together using transition lines, to determine the flow of the game.

In Taleblazer the designer creates ‘regions’ and ‘agents’. Regions are the physical areas on a map where the game takes place. The first thing that a designer must do is to determine the region of the game on a digital map with the use of a selection tool. After the region is set then ‘agents’ can be introduced. ‘Agents’ are associated with GPS locations. Taleblazer ‘agents’ and Metaverse Studio ‘experiences’ are more or less the same thing in the context of LBGs. Agents and Experiences are digital content associated with a GPS location and activated when a learner (or player) ‘bumps’ into this location. This digital content can take many forms, such as a multiple-choice question, an avatar that starts a conversation with the player, a video or audio file, etc. The two platforms support different options regarding digital content. The digital content can be associated with what the learners see in their physical environment (e.g., a point of interest such as a building, a bridge or other construction, a cultural heritage monument, a museum exhibit, a street sign, etc.). Alternatively, it can be content imagined by the designer, and therefore it is not associated with the players' surroundings.

On the other hand, the Metaverse Studio development environment does not require programming but instead relies on visual authoring and logical thinking. Experiences in the Metaverse AR environment are created in the Experience Storyboard. The Experience Storyboard is a drag-and-drop canvas where the designer combines elements called Scenes and Blocks. Scenes are the visual elements of the experience, and Blocks are logic elements that perform actions while an experience is running. The user defines the flow of the experience, or otherwise the experience scenario, by creating lines (links) between the elements. Scenes and Blocks are linked together using transition lines (Figure 2). Conditional branching is also available.

Figure 3 A group of Metaverse experiences in the designer's environment (see online version for colours)

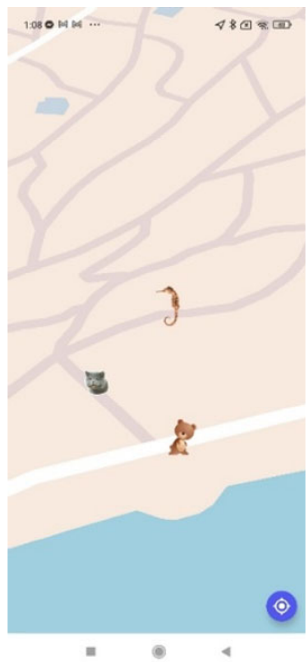


Note: Each experience has its own icon

As already mentioned, Metaverse Studio does not support only AR location-based experiences. It is designed for AR mobile applications in general and can also support the implementation of AR location-based experiences. A certain procedure must be followed to create a unified AR location-based experience that involves many geographical locations in Metaverse Studio. In this procedure, the experiences for each location must be created one by one, and each experience must be assigned to specific GPS coordinates using a map. Moreover, the created experiences must be organised in a 'group'

(Figure 3). The designer can obtain a QR code for the Group and distribute it to interested parties. By visiting the group of experiences, a map appears, and the end-user can visit the GPS locations for each experience (Figure 4). When the user gets close to a target location, the point on the map can be tapped, and the experience will be activated. The experience designer chooses the images (or icons) representing each experience. GPS locations can be visited in any order.

Figure 4 The experiences are depicted on a map when the game starts (see online version for colours)



4.2 Multimedia elements

As mentioned previously, the digital content that is associated with GPS locations and triggered when the users reach these locations can contain various multimedia elements. The multimedia elements that are supported by both platforms vary. Taleblazer interactive game scenarios can include 2D images, audio, and video. On the other hand, Metaverse Studio has a rather extensive set of options. More specifically, the multimedia elements that can be inserted in a Metaverse interactive experience can be 2D images and 3D models, video, and audio files, 360° videos and 360° images, internet webpages, polls, and games. The Metaverse Studio library contains many 2D and 3D images to choose from as well as 360 images and videos. The designer can also use multimedia elements that can be found in open web repositories. For example, 2D images (bitmap or vector), can be found on sites like Pixabay, Freepick, Freevector, and 3D images can be found on sites like Sketchfab, Free3d.com, etc. The designers can also use their own multimedia creations. The designers also have the option to create their 360° images and 360° monoscopic videos if the appropriate equipment (e.g., 360° camera) is available.

YouTube 360° videos will not render in Metaverse, so this option is unavailable. In a Metaverse experience, the end-user can also be asked to input text (e.g., open-ended questions), and different actions can be carried out according to the text given. The users can also be asked to take videos or pictures of their surroundings as well as selfies. Pictures and videos can be stored on a media wall and viewed by the player and the experience creator on demand.

Figure 5 3D image in AR view (see online version for colours)

Note: The text in the label says, 'Mission 2 accomplished!!! Your score is 2'.

Figure 6 3D image in AR view (see online version for colours)

Note: The text in the label says, 'Lets Go'.

GoMeta, the company that created Metaverse Studio, also has a platform for creating interactive games called Koji Games. These games can also be incorporated into a Metaverse experience. Google vision can also be utilised in a Metaverse experience. For example, the designer can determine what Google should look for in an image (an emotion, an animal, an object, etc.), and the Google vision algorithm will check if this exists. Furthermore, various interactive scenarios can be played out according to the result returned by the algorithm. Another available option that is present in the Metaverse studio platform is the ability to view objects in AR view (Figure 5 and Figure 6). That means 2D images and 3D models can appear in the natural environment through the mobile device's camera lens.

Another way to incorporate multimedia elements in experiences created by both platforms is via links that redirect the user to external web pages (or platforms). In this way, the designer can also incorporate various multimedia elements in a Taleblazer game, such as 360 videos and polls, but this will happen outside the application in a separate web browser window.

4.3 Inventory and variables

Both development environments support the notion of 'inventory'. Users can be asked to collect digital items during the game and keep them in an inventory. Various game scenarios can play out depending on the items collected. In Taleblazer and Metaverse, players can check the objects in their Inventory at any time during a game.

Both platforms also support variables. In Taleblazer, these are called 'traits' and in Metaverse Studio, they are called 'properties'. These variables can be assigned specific values, and values can be increased or decreased. In this way, various game mechanics can be supported such as scores, and branching scenarios. By branching scenarios, it is meant that the flow of the experience will be different depending on a value stored in a variable. In Taleblazer, 'traits' can be assigned to agents or roles (players). A trait can be common for all objects of the same type (agent, role, or team) or only for a specific object (agent, role, or team). World traits are global settings shared by all players in a game. For example, the designer can use a world trait to specify and display the current temperature in the virtual world.

In Metaverse Studio, there are three kinds of properties, user properties, experience properties, and environment properties. User properties are tied to a user account and are carried in-between experiences. Experience properties on the other hand are temporary properties that live within the life of an experience. Experience properties are created and destroyed each time the experience opens and closes. At last, environment properties are global values that are affected by every user that interacts with them. Environment properties can be also carried in-between experiences. An example of an environment property could be the total number of times an experience has been activated.

4.4 Role-playing games

Taleblazer is built for creating role-based activities. One can easily create different roles and different scenarios can be built around these roles. The game designer can configure a single role for a game, in which case all players experience the same game, or he/she can configure multiple roles, in which case the scripting language can be used to specify different interactions for players taking on these roles. Like an agent, the role can have a name, description, and image associated with it. Also, like an agent, the role can have traits and actions associated with it.

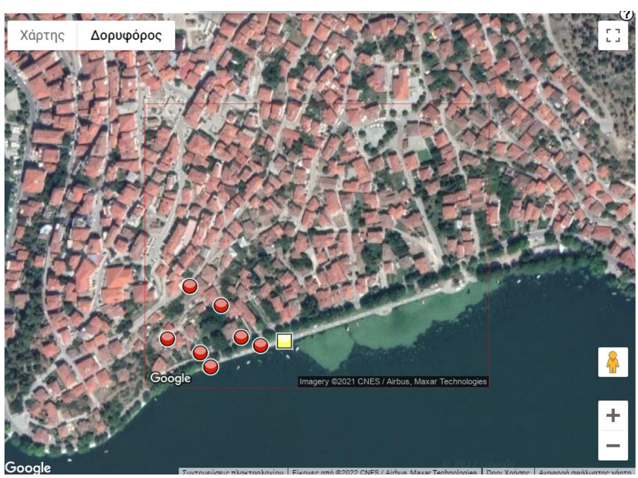
Metaverse Studio is not designed to support different roles the way Taleblazer is. However, role-playing scenarios could also be implemented in Metaverse with the use of different user variables. For example, the user can answer a multiple-choice question to choose a role, and according to his/her answer, a different value can be assigned to a user variable. Then different scenarios can be created depending on these values.

4.5 Building experiences where target locations are visited in a predefined order

It may be a design decision to develop games that contain target locations that have to be accessed sequentially and in a predefined order or games that the end-users visit the target locations in the order that they desire.

Taleblazer supports both design decisions, that is LBGs where Agents are accessed sequentially and in a specific order and games where the order in which the agents are visited is left to the end-user to decide. In non-sequential games, the visited target locations can be given a different colour or can even be deleted from the map, indicating in this way the agents that remain to be visited. In games where the agents are to be visited in a specific order the same technique can be used. The designer can decide at any time which agents become visible or invisible and therefore, only the agent that needs to be accessed next according to the game scenario will be visible. In Figure 7, all the agents that are present in the prototype application are shown in the Designer environment. The designer then determines which agent will appear next using the Taleblazer's visual scripting language

Figure 7 Agents introduced in the designer environment (see online version for colours)



Different colours can also be used (e.g., grey, red) to indicate the agent to be visited next. Figure 8 depicts the start of the game in the player's UI and the first agent to be visited according to the game scenario.

Metaverse, on the other hand, is not designed for games where locations are accessed sequentially. It is mainly designed to support games where the order of the experiences to be visited is not set. In a group of location-based experiences where target locations are to be visited in a specific order, there is no concern if these targets are on an obvious path, and the order in which one should visit the points is straightforward. However, this is not often the case. In this case, it can be possible to create experiences where the locations are accessed in a specific order, but the designer has to be inventive and use his/her imagination to achieve this.

Figure 8 Next agent to be visited by a player according to the game scenario (see online version for colours)

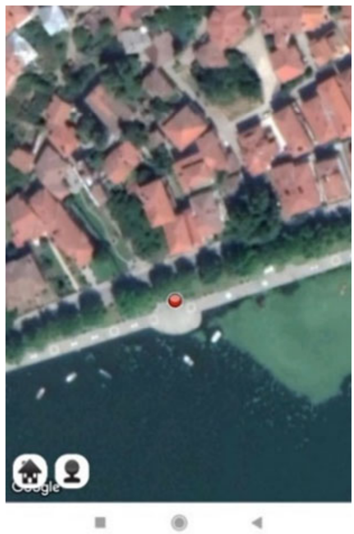


Figure 9 Experiences to be visited in a predefined order (see online version for colours)

In the prototype application, we tried just this. To create an experience where locations are visited sequentially. Luckily, third-party videos on YouTube and the Metaverse website provide ideas on how to accomplish this.

So, to create such type of game, images with numbers were used for each experience (Figure 9). The numbers indicate the order in which the experiences should be visited. Also, by utilising a user or experience property that increments every time the correct experience is visited, the designer can forbid users from activating the wrong experience. For example, at the beginning of the game, a variable has the value 0, and then in target location 1 (or experience 1) the variable's value is incremented by 1. In location 2, this variable is then checked to see if it has the appropriate value (value 1). If yes, the experience can be executed. If not, the user is asked to visit location 1 on the map. And the variable will continue to be incremented if the target locations are visited in the right order.

4.6 'Autobump' and 'tap to bump' settings

When the user's location comes close enough to the target GPS locations defined in the game scenario, then the Metaverse experience or the Taleblazer agent is activated. According to the Taleblazer terminology, the player is said to 'bump' into the Agent. In Taleblazer an agent is activated either automatically (autobump) or by tapping on the red dot of the agent on the map (tap to bump). For the agent to be activated automatically, the player has to be within a certain distance from the target location. This distance can be defined in Taleblazer, in the 'bump settings', and the same holds for the 'tap to bump' option.

The designer, in some cases, may want the player to get very close to a physical location to observe something small such as a street sign or a sign on a door or wall. In these cases, the designer must set the 'autobump' and 'tap to bump' distance to a small value. However, distances smaller than three metres can be problematic because the accuracy of the GPS location can be affected by various factors such as tall buildings, atmospheric conditions, etc. Sometimes finding a target can be a confusing task in case the 'distance value' is small, and the designer will have to test the game to ensure the smooth flow of the experience. However, there are also cases where the physical target is something big such as a monument, building, or other construction, that is easily visible from a distance. In these cases, the 'autobump' and 'tap to bump' distance can take a larger value.

On the other hand, in Metaverse Studio this distance is fixed to 30 metres. This means that the player must be within a 30 metre radius to be able to tap on the dot of the 'experience' in order to activate it. Metaverse supports only the 'tap to bump' option, meaning that experiences will not be activated automatically but only when the player taps on the red icon of the target location on the map. However, 30 metres radius can be a long distance for games where the scenario requires the player to get close to a target location to complete a task, such as observing a relatively small object (e.g., a sign on the wall) and perhaps answering questions regarding this object.

In conclusion, Taleblazer is more appropriate for games where the 'autobump' and 'tap to bump' distance settings need to be adjusted.

4.7 Map details

Taleblazer platform has more detailed maps when compared to the Metaverse platform. Both applications use dynamic maps downloaded from application servers and updated as the player changes location. More specifically, Taleblazer uses the Google Map API by default to display the player's position in the real world during the game. Figure 8 (Taleblazer map) and Figure 9 (Metaverse map) show the same location. The snapshot is taken from the prototype applications. It is clear from the images that Taleblazer is more appropriate for cases where a more detailed map is needed.

Another useful option in Taleblazer is the ability to upload 'custom maps'. The designer can upload custom maps and use these custom maps instead of the dynamic map. For example, the game can contain many agents (e.g., 20 agents) that are distant from one another and placed in a large region. The designer can use map images that are cropped and 'zoomed in' when the game is temporarily limited to a smaller part of the region that contains a subset of the agents (e.g., 5 agents). A custom map can also be used to show custom details on the map during gameplay, such as historical or geographic information or even fictional information (e.g., future constructions). This option is also useful when the designer wants to develop a game that evolves in an area not covered by the Google Map API, such as paths in nature (hills, forests, and mountains). In this case, custom map images can be created with editing software or derived from other resources and used in the Taleblazer application. In this case, caution must be taken to match the custom map and GPS targets. Taleblazer also allows the designer to capture a satellite image of a game area within the designer environment of Taleblazer with a click of a button, the 'capture image button', which can be found in the map settings section. Then the image can be downloaded, edited with the appropriate software (add layers of constructions, pedestrian paths, text to give instructions, etc.), and then uploaded again to the application. In Figure 10, one can see a custom image map that is edited to contain a path in nature (i.e., red line).

Figure 10 A snapshot of Taleblazer game depicting a path in nature using a custom map (see online version for colours)

Custom image maps can also be used if a game scenario involves multiple regions. Multiple regions can be used to define different physical regions where the game evolves, such as two different areas of a school campus. But regions can also be used to create different layouts on the same physical location. Multiple regions that are mapped to the same real-world location can be used to represent different game levels, time periods, or different outcomes for the player's in-game decisions. Another advantage of Taleblazer's option to upload custom map images is its ability to support experiences played in indoor locations where the GPS signal is out of range. Indoor experiences will be covered in the section 'indoor games'.

Concluding we can say that Taleblazer is more appropriate in cases where a more detailed map is needed and in cases where the designer wants to develop games where custom map images are used. By using custom images, the designer can add additional details to the physical region using image editing programs. These details can be text or icons that help player navigation, real or fictional objects, and nature or city paths that are not depicted in a Google dynamic map.

4.8 Internet connection

Both Taleblazer and Metaverse Studio need an active internet connection for the dynamic maps to be updated when players move about in the real world. More specifically, with Taleblazer when the Google map API is used, the player must remain within Wi-Fi or cell tower coverage for the game map to be displayed. On the other hand, if custom map images are used, Taleblazer games can be downloaded and played without an internet connection. The ability to play games designed with Taleblazer without an internet connection is a significant advantage of Taleblazer when compared to Metaverse Studio. Designers that use Metaverse to build experiences do not have the option to use custom image maps, so an active internet connection will be needed all the time for an experience to run.

4.9 Indoor games

Indoor games are required to use a custom map of the physical indoor region (e.g., Figure 11), since indoor spaces cannot be included on a dynamic map. Furthermore, the GPS signal cannot be active in interior spaces.

Taleblazer has mainly two ways for supporting games in locations where the GPS signal is not active. One way is to activate the "tap to bump – regardless of the player location" option in the 'bump settings' and then use password-protected agents. In other words, agents are placed on the custom map, and passwords are associated with these agents. The passwords will have to be placed somewhere in the physical environment, and the game designer can think of various ways to achieve this (e.g., signs, stickers, etc.). The player will have to tap on the appropriate icon on the map and then insert the password found in the physical environment close to the target location to activate the agent. This way, it is guaranteed that the player has reached the required location in the indoor region.

The other way to support indoor games is with the use of beacons. Beacons are small Bluetooth devices that use a technology called Bluetooth low energy (BLE). Beacons regularly broadcast packets in the iBeacon format. iBeacon packets contain the beacon's identifying information and the power level at which the beacon is transmitting. The iOS and Android operating systems both provide libraries for determining three proximity levels based on signal strength: immediate, near, and far.

According to Apple, 'immediate' is physically very close to the beacon and very likely being held directly up to the beacon. 'Near' is 1–3 metres with a clear line of sight, and 'far' is given when the device is not confident enough in the accuracy of the reading to claim immediate or near (Getting Started with iBeacon, 2014; Finch, 2015). Multiple precision levels can indicate the player's position and support different game mechanics.

Moving to Metaverse Studio, we can say that the idea of password-protected experiences can also be implemented in the Metaverse platform but not as efficiently and easily as in Taleblazer since custom maps are not supported in Metaverse the way they are in Taleblazer. The option to use beacons is also found in the designer's environment but there are no instructions in the documentation on how to implement a game with beacons. In general, there is a lack of information in the Metaverse documentation regarding the development of indoor experiences.

Concluding we can say that Taleblazer's capabilities (e.g., custom maps) and documentation are more efficient in supporting the development of indoor games.

Figure 11 Custom map of an indoor game (Kastoria city Aquarium) (see online version for colours)

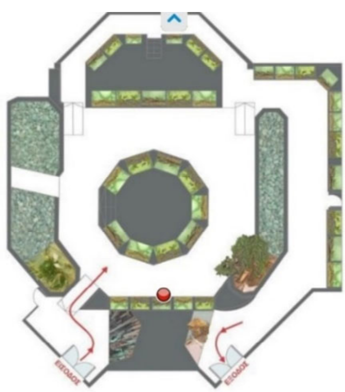
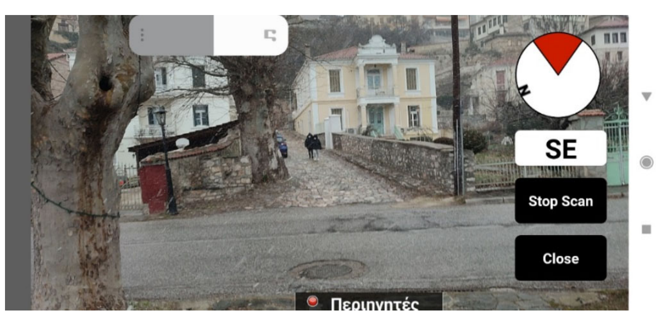


Figure 12 Head up navigation mode (see online version for colours)



4.10 Alternative features that aid navigation

Taleblazer has an additional feature to support player navigation. The 'head up navigation option' activates a live camera view containing a compass that indicates the direction to be followed to reach the next agent's location according to the game scenario (Figure 12).

4.11 Testing an experience in simulation mode

Both platforms provide the designer with ways to test the produced experience. However, there are significant differences in the available options that the two platforms provide for testing the created games.

Starting with Metaverse, as already mentioned in order to create LBGs, 'experiences' have to be gathered in a group and assigned to GPS locations. In Metaverse, individual experiences can be tested one by one, and this is accomplished by pressing the button 'test' and obtaining a QR code. Metaverse however does not give the option to test the Group of experiences (as an integrated experience) without being in the physical region of the game and this does not make things easy for the designer. In other words, the tool does not provide the ability to perform game simulation tests from the developer's office.

Taleblazer, on the other hand, provides a simulation mode option. More specifically, in the settings tab there is an option called 'tap to visit', and when this option is activated, the game can be played, and tested at any time and without the need for the designer to be within the physical region of the game. This 'tap to visit' option is also available to the end-users, and this is something which is not always desired. In many cases, the designer may decide that the game can only be played by visiting the physical location, and in these cases, the Taleblazer platform provides the option to password-protect the 'tap to visit' option.

4.12 Game learning analytics

Game learning analytics are useful in understanding how the learner (or player) navigates through the game and other aspects of the learning experience. Analytics can reveal aspects of the end-user performance, problems associated with the game scenario, and hidden patterns in learner behaviours. Analytics can also predict the learning or the game's outcome (e.g., predict if the end-user will complete or abandon the game).

In Taleblazer, there is no way to store user data in global variables or a database so that they can be accessed after the game is over, making it impossible to extract metrics. Metaverse, on the other hand, supports variables that can be altered by all end-users called 'environment properties' as already mentioned. Thus, by introducing 'environment properties' simple statistics can be derived (e.g., number of times an experience is activated, total time spent in the experiences, etc.).

4.13 Scores and competition amongst players

Metaverse Studio provides a feature that can introduce competition amongst players: the LeaderBoard block. Leaderboards track scores of users for experiences, and points can be added or subtracted to and from specific users. In Taleblazer, on the other hand, there is no obvious way for a designer to introduce competition among players based on scores. According to Taleblazer documentation (http://taleblazer.org/files/docs/TaleBlazerDocumentation.pdf), multiplayer features are under development. According to the documentation, players participating in a multiplayer game will experience a single shared game world in which changes to the virtual world are propagated to all players playing the same game. It is expected that in a multiplayer game environment, user scores could be shared amongst users, introducing in this way competition among players.

4.14 Support

Both Taleblazer and Metaverse Studio have detailed online documentation to support designers in the creation process. More specifically, Taleblazer has detailed documentation in the form of pdf documents. The pdf documents can of course, be downloaded and viewed offline. These documents cover different aspects, such as a brief introduction to creating LBGs, a detailed guide to the game mechanics, best practices, etc. Taleblazer also provides example games that designers can download in order to see how a game is implemented and experiment by altering its source code in the block-based programming environment.

On the other hand, Metaverse Studio also provides the designer with detailed online documentation (https://studio.gometa.io/learn) covering all aspects of the platform (e.g., environment overview, scenes, blocks, etc.). Metaverse also has a significant number of tutorial videos in its official YouTube channel (https://www.youtube.com/channel/UCum7uPJBXug0HfqNi4AfQmQ) covering every aspect which might concern the designer. The channel also hosts several third-party 'how-to' tutorials covering various aspects of the development of AR experiences.

5 Conclusions

Location-based AR games promote active learning through the parallel use of a mobile application and the exploration of the physical environment. Taleblazer and Metaverse Studio are two popular platforms that are currently used by educators to design location-based AR games. This paper performs a comparative analysis between the two platforms to provide educators and other parties interested in building location-based AR learning games with the necessary knowledge to make an informed decision on which platform to use based on their requirements. To perform this comparative analysis, a certain procedure was followed. First the designer and the end-user environments were examined thoroughly. Then the relevant documentation of both platforms was studied as well as several papers from the related literature. The next step was to test third-party applications that were developed on both platforms in order to further explore the end-user interaction environment. As a last step, two prototype AR LBGs were developed using the Taleblazer and Metaverse Studio in order to obtain a better understanding of the developer environment and the capabilities supported by both platforms. The findings show that both platforms have advantages and disadvantages over the other.

Taleblazer might be more suitable for educators that want to implement more advanced functionality through a visual block-based programming environment. However, Taleblazer's learning curve is steeper, even though the block-based programming environment used to develop games is similar to the MIT Scratch environment and thus appropriate for people with no programming experience.

Taleblazer is also better for developing role-playing games, games where the target locations must be visited in a specific order, games that take place in nature or in regions where the dynamic map is not able to depict the required information for the player to navigate safely (e.g., hiking paths), and games that take place in indoor spaces where there is no GPS signal. Games made by Taleblazer can also be played offline if custom map images are used. In this case, an internet connection will only be needed to download the Taleblazer app and the game. Games using a dynamic map (Google Map API) however need an active internet connection. Testing the whole LBG functionality without visiting the game region is also feasible in Taleblazer.

On the other hand, Metaverse requires no programming knowledge, so it might be more appealing to many educators who do not want to deal with programming. Metaverse Studio has a broader range of multimedia elements that can be incorporated into a location-based AR game, it can support competition amongst players as well as basic game learning analytics. Finally, the designer must keep in mind that games developed in Metaverse Studio can only be played with an active internet connection, and there is no way to test a game without being in the game region.

As a bottom line, it can be said that both platforms are popular and support location-based AR games, and it's up to the educators to decide on which platform to use according to their requirements.

Acknowledgements

This work has been co-funded by the European Union and Greek national funds through the Priority axis 'Development of Entrepreneurship Support Mechanisms' of the Operational Program 'Competitiveness, Entrepreneurship, and Innovation', Proposal in the framework of the action entitled 'Supporting Regional Excellence' (project code: 5047196).

References

Avouris, N.M. and Yiannoutsou, N. (2012) 'A review of mobile location-based games for learning across physical and virtual spaces', *J. Univers. Comput. Sci.*, Vol. 18, No. 15, pp.2120–2142.

Fidas, C., Sintoris, C., Yiannoutsou, N. and Avouris, N. (2015) 'A survey on tools for end user authoring of mobile applications for cultural heritage', in *2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA)*, July, IEEE, pp.1–5.

Finch, E.Y. (2015) *TaleBlazer: Using iBeacons for Indoor Location-Based Augmented Reality Games*, Doctoral dissertation, Massachusetts Institute of Technology, DSpace@MIT, http://hdl.handle.net/1721.1/100594.

Getting Started with iBeacon, Version 1.0 [online] https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf (accessed November 2021).

Laato, S., Pietarinen, T., Rauti, S., Paloheimo, M., Inaba, N. and Sutinen, E. (2019) 'A review of location-based games: do they all support exercise, social interaction and cartographical training?', in *CSEDU*, May, No. 1, pp.616–627.

Lacka, E. (2020) 'Assessing the impact of full-fledged location-based augmented reality games on tourism destination visits', *Current Issues in Tourism*, Vol. 23, No. 3, pp.345–357.

MacCallum, K. and Parsons, D. (2019) 'Teacher perspectives on mobile augmented reality: the potential of metaverse for learning', in *World Conference on Mobile and Contextual Learning*, September, pp.21–28.

Metaverse Studio YouTube Channel [online] https://www.youtube.com/channel/UCum7uPJBXug0HfqNi4AfQmQ (accessed November 2021).

Metaverse Teacher Professional Development Session (2019) [online] https://www.youtube.com/watch?v=MLeZo7X5rnA&t=2495s (accessed November 2021).

Metikaridis, D. and Xinogalos, S. (2021) 'A comparative analysis of tools for developing location based games', *Entertainment Computing*, Vol. 37, p.100403, ISSN 1875-9521, https://doi.org/10.1016/j.entcom.2020.100403.

Mozelius, P., Jaldemark, J., Bergström, S.E. and Sundgren, M. (2019) 'Augmented education: location-based games for real-world teaching and learning sessions', in *Augmented Reality Games I*, pp.217–235, Springer, Cham.

Nóbrega, R., Jacob, J., Coelho, A., Weber, J., Ribeiro, J. and Ferreira, S. (2017) 'Mobile location-based augmented reality applications for urban tourism storytelling', in *2017 24º Encontro Português de Computação Gráfica e Interação (EPCGI)*, October, IEEE, pp.1–8.

Norris, E., van Steen, T., Direito, A. and Stamatakis, E. (2020) 'Physically active lessons in schools and their impact on physical activity, educational, health and cognition outcomes: a systematic review and meta-analysis', *British Journal of Sports Medicine*, Vol. 54, No. 14, pp.826–838.

Siakavaras, I., Papastergiou, M. and Comoutos, N. (2018) 'Mobile games in computer science education: current state and proposal of a mobile game design that incorporates physical activity', in *Research on e-Learning and ICT in Education*, Springer, Cham., pp.243–255.

Söbke, H., Baalsrud Hauge, J., Stefan, I.A. and Stefan, A. (2019) 'Using a location-based AR game in environmental engineering', in *Joint International Conference on Entertainment Computing and Serious Games*, November, Springer, Cham., pp.466–469.

Weber, J. (2017) *Designing Engaging Experiences with Location-Based Augmented Reality Games for Urban Tourism Environments*, Doctoral dissertation, Bournemouth University [online] https://eprints.bournemouth.ac.uk/27176/ (accessed November 2021).

Weber, J. and Dickinson, J. (2018) 'Designing location-based augmented reality games for urban tourism: a conceptual framework', in *e-Review of Tourism Research* [online] https://ertr-ojs-tamu.tdl.org/ertr/article/view/128 (accessed November 2021).

Xanthopoulos, S. and Xinogalos, S. (2017) 'An overview of location-based game authoring tools for education', in *Interactive Mobile Communication, Technologies and Learning*, November, Springer, Cham., pp.201–212.