An Authoring Tool for Location-based Mobile Games with Augmented Reality features

Carleandro Nolêto, Messias Lima, Luís Fernando Maia, Windson Viana, Fernando Trinta

Federal University of Ceará, Group of Computer Networks, Software Engineering and Systems (GREat), Brazil

Abstract

This paper presents an authoring tool for building location-based mobile games, enhanced with augmented reality capabilities. These games are a subclass of pervasive games in which the gameplay evolves and progresses according to player's location. We have conducted a literature review on authoring tools and pervasive games to (*i*) collect the common scenarios of current location-based mobile games, and (*ii*) features of authoring tools for these games. Additionally, we have also used the focus groups methodology to find new scenarios for location-based mobile games, in particular regarding how augmented reality can be used in these games. Both literature review and focus groups provide us a set of requirements to design a software architecture for our authoring tool, a web-based application where games are created and executed. In our approach, games are designed as a set of missions that can be ordered or not. Players use mobile devices to perform these missions in order to complete each game. Our main objective is to provide a software solution to enable non-programmers users to design, build and run location-based mobile games. In order to evaluate our tool, we present a game design called "Battle for Fortaleza", and how this game is implemented in our solution.

Keywords: Pervasive Games, Location-based Games, Augmented Reality, Authoring Tools

Author's Contact:

{carleandronoleto,fernandotrinta}@great.ufc.br
{luis.maia}@ifma.edu.br
{windson}@virtual.ufc.br
{messiaslima.03}@gmail.com

1 Introduction

In recent years, Location-Based Mobile Games (LBMG) (also known as "Urban Games") have gained attention in both research and industry communities. LBMGs are a subtype of pervasive games where player's location is a fundamental key to the game design. Players use mobile devices such as smartphones to provide player's location using specific positioning techniques (such as GPS sensors). A typical LBMG maps a real world area into a virtual world, where players should move themselves in real life in order to accomplish tasks related to the game itself. According to the game objectives, players may collect virtual objects such as weapons, ammunition or treasures. These objects can be used to achieve benefits in the virtual world, or even in the real world. These game features offer a great potential for touristic and educational purposes, where users will be encouraged to visit specific places, and further information about these locations are provided to them.

In the case of tourism and educational games, LBMG require a careful design to achieve goals beyond the entertainment itself. For instance, the order in which tasks are planned may affect users' interaction and game experience. Such planning ends up being delegated to professionals, such as tourism specialists or educators, who in general do not have experience in developing applications, especially a LBMG. This scenario creates a dependency on IT specialists in order to implement location-based games. A more interesting scenario would be that one where tools allow the design and implementation of location-based games in a quick and easy way, even by people with no software development background.

This paper addresses this topic by presenting an authoring tool for location-based mobile games, where people with no-prior knowledge of programing languages or application development skills can design and execute these applications. This authoring tool uses a visual notation to create games based on a set of quests. Each quest represents a task (or group of tasks) that a player or a group of players must accomplish to reach the game objective. Each task is associated with activities such as "Moving to a specific point", "Collecting a virtual object", and so on. In our solution, these tasks are known as Game Mechanics. Quests and Game Mechanics are concepts in which our authoring tool is based upon, and reflect common activities that most LBMGs require, and that were collected according to a survey on both academic and industry LBMGs. We also used the Focus Groups technique to refine attractive ideas and scenarios for location-based mobile games, especially regarding how augmented reality features could be integrated with LBMGs. Both LBMGs survey and the focus groups provided us requirements to our application model, which were used to design a software architecture to support game development and execution.

The rest of this paper is organized as follows. After this introduction, section 2 presents some LBMGs and authoring tools related to our proposal. In section 3 we present the research methodology used to define the games' styles and features to be developed using this work. Section 4 introduces our game application model and describes the architecture of our solution. In Section 5, we present a study case, a game called "Battle for Fortaleza", where most features of the authoring tool are presented and discuss an evaluation of the tool made with two users with no programming background. Finally, in section 6 we conclude our paper and present future directions for our research.

2 Related Work

LBMG represents the most widespread category of pervasive games. Unlike most of digital games, which take place in a virtual world, pervasive games merge virtual and physical worlds by integrating information concerning the user's situation (location, device, activity, object proximity) into the game experience [Mello Viana et al. 2014].

In order to understand the main features of pervasive games, we have conducted a literature review of authoring tools and pervasive games. Games that rely on location were analysed to provide a better comprehension of their gameplay, features, popularity, interaction, usage and mechanics. Moreover, an authoring tools study was required to provide advantages and drawbacks, as well as knowledge from the state-of-art on the area.

2.1 Location-based Mobile Games

From a set of keywords (e.g, "pervasive games", "location-based games", "urban games", ...), we have conducted searches on scientific bases (e.g., Google Scholar, IEEE Xplorer) aiming to find scientific papers on LBMGs' support. Additionally, a search for mobile applications with LBMG features has been done in the App Store (iOS) and PlayStore (Android).

Among the games found, the following stand out: MusA[Rubino et al. 2013], FreshUP[Köhlmann et al. 2012], Geocaching[Ihamäki and Luimula 2013], Urban Match[Celino et al. 2012], ExCORA[Linaza et al. 2013], Far Quest[1], GeoBoids[2], Parallel King-

[1] https://play.google.com/store/apps/details?id=com.openrevman.farQuest&hl=pt_BR

[2] https://play.google.com/store/apps/details?id=com.hitlabnz.geoboids&hl=pt_BR

dom MMO[3] and Ingress[4]. The last five selected games are LBMGs that present great popularity in Android's Play Store. For instance, Ingress has more than ten million downloads. In our analyses, we searched for specific characteristics, such as GPS and Augmented Reality usage, game type, mechanics and supported platforms.

The majority of games uses GPS for tracking player's location, except MusA that uses image processing for this purpose. MusA relies on a database of images bound to places, so the player has to take pictures of objects in the environment and the application queries the image database to determine player's location.

Augmented Reality(AR) is becoming increasingly popular among pervasive games for its ability to increase immersion and provide a new means of interaction. AR was used in games like X-Rift, GeoBoids, Gunman and Far Quest to deploy virtual objects inside player's environment. Those games use GPS tracking to determine where an AR object must be positioned, thus working as markerless AR browsers.

Another important aspect of a game is its type, which define whether a game is collaborative or competitive. Collaborative games seem to be more successful among LBMGs since the most downloaded games we have explored were cooperative. For instance, Parallel Kingdom MMO, X-Rift and Ingress are founded on conquest of virtual territory by teams.

A key part of this work is the definition and implementation of game mechanics present in LBMGs, therefore a detailed analysis of the mechanics was carried out. A widely used mechanics is to require the player to move to a determinate location and perform a single task. Additionally, only MusA and ExCORA used AR objects on their mechanics.

Table 1 summarizes the games and their features. Regarding platforms, most games are developed for Android and iOS devices, only Geocaching runs also on Windows Phone. Additionally, [Moebert et al. 2011] and the game FreshUP used web browsers as their platform of execution.

2.2 Authoring Tools

There are numerous authoring tools for designing games available on the Internet. However, we were interested on authoring tools that can generate LBMGs, especially those with augmented reality features. The main purpose of this investigation is to map their features, workflow, common mechanics, advantages, and drawbacks.

Currently, one of the most full featured authoring tools is Aris Game. It is an open-source platform that allow users to create mobile games using GPS tracking and QR Code recognition[ari]. Aris Game provides support for numerous media files, such as images, sounds, texts and videos. The platform generates games that merge virtual and real world by placing these virtual media content based on the player's location and QR code detection. Moreover, Aris Game presents a sharing edition feature, that allow multiple users to edit the same game simultaneously. However, the software presents a great disadvantage since it only generates single player games.

Another popular authoring tool is fAr-Play. It generates games that run on web browsers of mobile devices and allows the integration with AR objects[Gutierrez et al. 2011]. AR integration is achieved via Layar, an augmented reality tool that plots 3d objects using GPS coordinates.

The mechanics of LBMG play an important role in game design, and the ALRA[do Prado Rafalski et al. 2013] allows the creation of public and private mechanics. This characteristic extends the flexibility of outputted games as they may contain mechanics visible to all players or specific groups. However, it also allows outsider players to change the games' flow by altering the sequence of mechanics, which is a great flaw.

[3] https://play.google.com/store/apps/details?id=com.silvermoon.client&hl=pt_BR

[4] https://play.google.com/store/apps/details?id=com.nianticproject.ingress&hl=pt_BR

A limited but original authoring tool is Tidy City Scout. The tool only supports the use of pictures and texts in its mechanics[Wetzel et al. 2012]. However, it allows users to create mechanics from a mobile application by capturing the state of phone's sensors. After defining the mechanics, the user has to finish and publish the game via an online platform in which the mechanics can be edited. Despite using the mobile device to aid in the process of game designing, Tidy City Scout does not allow users to correctly define the game flow once it prevents mechanics from being ordered.

Finally, SILO is a tool that provides means to order game mechanics. However, it yields limited support for media content, thus not being as popular as other authoring tools[Wake 2013].

Regarding the authoring process, all of the above mentioned authoring tools allow users to graphically define tasks on a map. However, unlike this work, none of the tools provide a visual language or a graphical mechanism to order game mechanics or control the game flow.

Table 2 summarizes the features of the LBMG authoring tools. Most authoring tools present similar mechanics and a strong limitation. None of them provided support for the design of social or collaborative games since in all cases only a single device could execute the game at a time, and the use of virtual 3d objects and AR concepts is quite limited when present. Based on this scenario, we proposed an authoring tool to address such limitations while extends the most popular and useful features of the previously introduced authoring tools.

3 Research Methodology

A popular approach for data collection and analysis in qualitative researches is the Focus Groups methodology[Aschidamini and Saupe 2004]. It has received great attention from researchers for their ability to perceive features and compile ideas about specific topics and activities. This kind of research method is widely used on academic works because it provides a fast setup and for being cheap to organize. Focus groups are also used in interactive systems like games to highlight the challenges and features users expect to face when playing.

Focus groups allow researchers to efficiently collect qualitative data through short meetings. These meetings are guided by a researcher, who observes the participants' comments on specific topics for a specified period. Nevertheless, not all data gathered at a meeting can be generalized or are easy to process. Hence, focus groups must be properly conducted to provide the necessary information.

3.1 Focus Groups for LBS Games with AR

We have used focus groups to formulate scenarios, specify the game mechanics for location-based mobile games and enhance the integration with AR. This approach has been successfully used in other game design researches[Ballagas and Walz 2007].

The main information sought on the focus groups' meetings concerned the types of tasks present in location-based mobile games, the mechanics present in those tasks, the use of augmented reality to increase immersion, and whether such games present scenarios for playing individually or in groups, thus raising the possibility to create competitive or collaborative games.

After deciding the main questions to be answered by the focus groups, a group of professionals was formed to join the first meeting, which lasted sixty three minutes. Since the focus groups aimed at specific LBMG features, the group was composed by three digital media professors, a computer science professor and a digital designer.

Additionally, we have organized a second meeting to ensure the correctness of the data collected in the first reunion, and to gather supplementary information. This second session was composed by four digital media students that work with digital games, a PhD student in computer science with a master's degree in computer graphics, and a game design professor. The meeting focused on the same topics discussed by the first group, and lasted about eighty minutes.

Table 1: *Examples of LBMG found on scientific papers and on mobile application stores.*

Game Name	Platform	3D Augmented Reality	Features
FreshUP	Web-based	No	Answer geolocated questions about some specific issues
Musa	Android	No	Singleplayer game that enhances Museum Visit experience. It uses a Marker-based Indoor Position System. Observation of the artworks and their surrounding environment are necessary to progress in the game
GeoCaching	Android, iOS, Windows phone	No	Multplayer game allowing users to find something hidden by other players by using a GPS-based mobile device and Map Interface
ExCORA	iOS	No	Once the player reaches a place, the application propose mini games to be accomplished: Puzzle, multiple choice questions, and other enigmas
X-Rift	Android	Yes	Players have by objective to destroy (shooting) AI characters (monsters), which are distributed geographically
Ingress	Android and iOS	No	The mobile application is the pervasive game's most successful, has a capture-the-flag style. Two global teams vying for territory (portals). A player has to go to the location to conquer neutral portals; or to strengthen protective shields. The player can also attack enemies portals moving to the location and leaving objects (bombs).
Far Quest	Android and iOS	Yes	The player has to find rewards in sites nearby the player's location. Traps are also geolocalized and have a certain time to explode. The player must go, in the real world, to the position of traps in order to disarm them.
Parallel Kingdom MMO	Android, iOS and PC	No	This LBMG is a massively multiplayer RPG. Users clam their territories based on their location (GPS) and can interact with NPCs (e.g., monsters) and other players.

Table 2: *Examples of Authoring Tools*

Authoring Tool	Objective	Mechanics	Platform	Multimedia Content
Aris Games	Tourism and Education	Move to specified places to interact with virtual content	iOS	Texts, images, videos and sounds
fAR-Play	Tourism and Education	Move to specified places or use a QR Code to view 3D virtual objects	Web browser	Texts, images, videos, sounds and 3D objects
ALRA	Education	Move to specified places to interact with virtual content	Web browser	Texts, images and videos
Ticy City Scout	Tourism	Move to specified places to interact with virtual content	Android	Texts and images
SILO	Education	Move to specified places to interact with virtual content	Symbian	Text and images

In both meetings, members were free to manifest their opinion, discuss the subjects and provide suggestions about the theme.

Both meetings provided similar concepts, especially related to the mechanics LBMGs should present. In general, participants claimed that such games should be composed by complex tasks built from a set of simple interactions. We called simple mechanics these simple interactions, and composite mechanics the complex tasks built from a collection of simple mechanics. For instance, a simple mechanic can be understood as a simple activity or interaction within the game, like moving to a specific place and viewing or dropping a digital content (text, images, videos, etc.). The game Ingress, a LBMG developed by Google that is currently the most downloaded LBMG on Play Store has proven the simple mechanics' approach is effective. It requires players to perform simple tasks continually, such as conquer a portal and capture virtual objects as means to advance in the game.

Moreover, at the first meeting participants stated the games should propose challenges between players, thus allowing users to compete and specialize in particular tasks. The game Ingress was again quoted as a successful example, since it widely explores such features.

Other significant information raised by the focus groups relates to playability, once games with a few types of mechanics were defined as tedious. According to the information gathered at the focus groups' meetings, the more varying mechanics a game presents, the more fun and entertaining it becomes. Therefore, participants suggested mechanics could be ordered or arranged in such a way the game flow could present changes in playability.

Finally, both meetings discussed scenarios where location-based mobile games could be played competitively or collaboratively. The majority of participants defended both styles present great acceptance among players, and traditional gameplay, such as capture the flag, hide-and-seek and tag are still widespread.

4 The Proposed Application Model

Both LGMGs' survey and the focus groups provided us relevant ideas that were considered when designing an authoring tool for these applications. We created an application model that represents which LBMG's features our authoring tool should support. In our view, LBMGs are designed as a set of quests to be accomplished. These quests can be performed by one player alone or by a group of players. One important feature mentioned by participants in focus groups is that missions could be sorted or not. Our authoring tool provide support for both sorted and unsorted missions, or even partially sorted. This last possibility represents a game design where only part of missions are sorted.

Each quest in our application model represents one or many actions that a player or group of players must perform. The actions performed by players refer mainly to their movement in the real world, reflecting their moves in the virtual world. In addition, players typically are induced to view and collect virtual objects in the virtual world, according to the objectives of the game. Once collected, these objects can be used and even relocated on the map,

according to the movement of the player who captured each item. The above mentioned actions represent the basic activities of a mobile game based on location. According to [Lundgren and Bjork 2003], a game mechanic is simply any part of the rule system of a game that covers one, and only one, possible kind of interaction that takes place during the game, be it general or specific. Therefore, in our application model, a game mechanic represents the most basic kind of interaction that a player could perform. The basic game mechanics includes (*i*) moving to a specific point on the map, (*ii*) visualizing a particular type of object, (*iii*) collecting or (*iv*) leaving a virtual object according to the player's location. Objects include a text message, an audio or video sample, and 3D elements presented using augmented reality. Basic game mechanics can be combined in order to produce more complex interactions, called then composite mechanics.

Figure 1 presents a class diagram that represents the concepts previously introduced. The class *SimpleMechanics* represents a basic game mechanic in our solution. It has an associated item that represents the object to be used in this mechanic. *CompositeMechanics* uses the Composite Design Pattern[Gamma et al. 1994] to indicate that a composite mechanic may be composed by simple mechanics or other composite mechanics. Each mechanic has a list of priorities, containing the mechanics list that must be executed before the mechanic itself. This list is used to provide the idea that mechanics can be ordered, and that some mechanics should not start until its predecessors tasks have been completed.

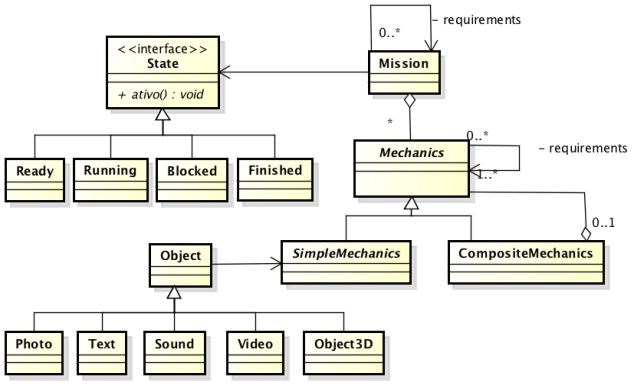


Figure 1: *Class diagram presenting how game concepts are modeled.*

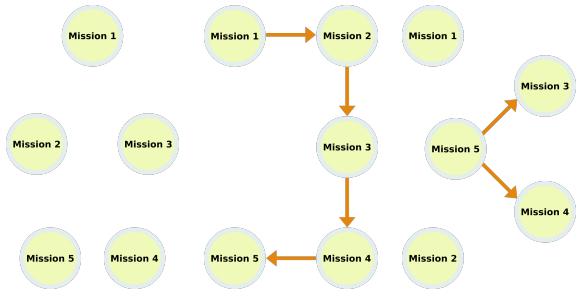
The class *Mission* represents a set of mechanics that are semantically grouped. Missions also have a list of priorities, in a similar approach as for mechanics. At runtime, each mission may assume one of the following states: ready, running, blocked and finished. A quest on a ready state means that it can be executed, since all its predecessors had already finished their tasks. Otherwise, a quest will be at the blocked state. A quest changes to the running state when at least one player starts the execution of one of its mechanics. When all mechanics are accomplished, the quest changes its state to finished.

The proposed model accomplishes mission composite, groups management, mission sharing and allows for virtual context extension. We have designed an authoring tool that addresses most location-based game plots, designs and features. However, this system presents some constraints, such as the need to create groups to include players. In this case, a single player game will have to define one group composed by one player. Moreover, the numerous features presented on the authoring tool can confuse some users, since even to build simple games a wide variety of concepts has to be understood.

4.1 Mission Modelling

In this section, we present examples on how to model different types of missions to better illustrate the authoring tool's concepts and features. For instance, Figure 2 depicts three games with varying game flow. For instance, in Figure 2 (a), there is no priority or dependency between missions, meaning players can start the game by choosing whatever mission they want to play, and move to any other they please.

Figure 2 (b) shows a game where each mission has a priority level, therefore players have to follow the specified sequence since they can only access missions after completing its dependencies. In this case, players have to finish mission 1 to unlock mission 2, and so on. In the last example, Figure 2 (c) shows a game with partially sorted missions. According to the diagram, players can start the game from missions 1, 2, and 5, while missions 3 and 4 will only be available after completing mission 5. Mission modelling is a key part of the authoring tool since it can prevent the design of many games by preventing the specification of the game flow. The proposed system of defining priority queues for missions offers the necessary flexibility to model countless types of game flows, thus making our approach compatible with the vast majority of LBMGs.



(a) *Game 1* **(b)** *Game 2* **(c)** *Game 3*

Figure 2: *A sample game with missions and priority list of execution*

After investigating the currently available authoring tools, we have noticed no systems allowed the creation of collaborative and multiplayer games. Therefore, this work proposes a new approach for binding players to missions. The platform enables users to define groups of players and to bind missions to many groups. Hence, players in the same group collaborate to finish the game. A competitive game can be designed by assigning players to different groups. The tool allows the creation of multiple groups, therefore a game can be designed for players of a group to compete with players of other groups. Moreover, missions can be bounded to different groups, so groups can compete with each other even when having different missions, such as capture the flag games.

The authoring tool also provides players tags to help define rules for chasing games. The system allows players to capture or be captured only if they have the same location. There are four player tags (i) neutral (used to disable capture verification), (ii) catchable (a player that can be captured but can not capture other player), (iii) catcher (a player that capture others, but can not be captured), and (iv) dual (players that can both capture and be captured). This feature is useful for modelling numerous chase games like cops and robbers, where a group of users can be tagged as Catchers to play cops, while other group of users can be tagged catchable to play robbers.

4.2 Software Architecture

Based on the application model described on Section 4, a software architecture aiming at support these applications was designed. This architecture is based on the Client/Server model of distributed computing. Figure 3 depicts this architecture and its three main components: The Game Editor, a mobile application and the Game Server.

The Game Editor and the mobile application act as clients of the game server, saving and retrieving information about games, missions and players. Game designers use the game editor to build their games using a graphical notation that will be explained in section 5. Each game must be explicitly published in order to be available to possible players. The Game Editor is a web-based application that lets authors to configure the scenario where the game will be held. Authors define the set of missions and their associated game mechanics based on locations where they must be executed

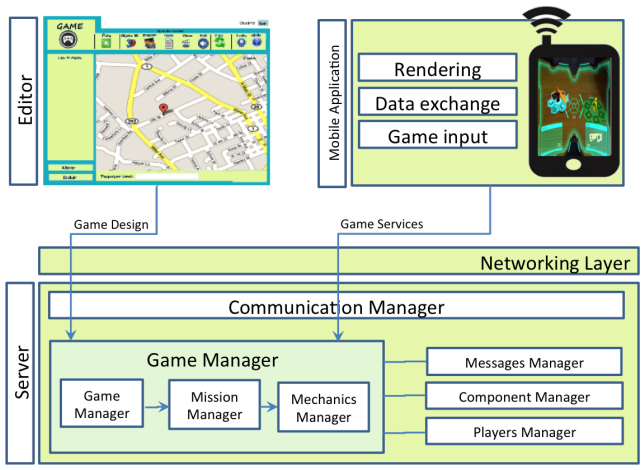


Figure 3: *Software Architecture of the Proposal*

in the map. Players use the mobile application to retrieve information about quests, and also to send its location based on the device's GPS sensor. The mobile app renders each game uniformly, based on which quests are active, finished, blocked, and so on.

When receiving data with the missions and its mechanics, the mobile app renders use standard icons to represent them on the map. Each icon has a different color, that represent its current state. A green flag means that a mission has not been performed yet. A red flag icon is used for blocked missions, since they have other missions that need to be finished first. Finished missions are represented by blue flags. This strategy allows the same mobile application to be used in different games, without requiring any modification. Each flag location represents places where the player must move to activate each mission. Figure 4 shows the mobile application screen with various mechanical spread over a territory which a game is going on.

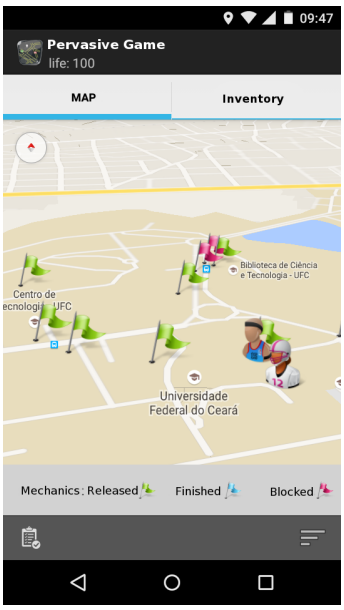


Figure 4: *Mobile Application showing the missions*

The mobile application also tells the server which actions were taken by the player, such as the conquest of objects. The communication between the application and the mobile game server can use both 3G / 4G connections, as well WiFi, if available in the coverage area of the game.

The third component is the game server, which is a central element responsible for managing information about games and players. The game server is compound by 6 components: (*i*) Communication Manager, (*ii*) Game Manager, (*iii*) Message Manager, (*iv*) Player Manager, (*v*) Component Manager and (*vi*) Authors Manager. The communication manager encapsulates how the exchange of information between players and the server is made. All games in a running state and their processes are coordinated by the Game Manager, which receives players' requests and return to the current state of associated game which the player is enrolled.

Games manager is responsible for managing the states of several games built and stored in our tool. This component receives players requests, and forwards them to correct instance of a running game. Therefore, it ensures that each action requested by players take effect in the correct game session that every player is connected. Similarly, the game manager must provide the correct game state for all players connected in a running gaming session. It will filter information according to the group that the player is bound. For example, for each player, only his objects, related missions and data about his characters should be returned.

Quest Manager configure missions and their pre-conditions, according to each game. This manager also controls the state of the quests according to the current state of the game. Mechanics Manager controls logic sequences where each game mechanic was defined. It assures that they are correctly executed, and also controls which player is responsible for them. Message manager allows players to exchange text messages while playing the game. Player Manager controls player's profile and their statuses according the games sessions they are connected. Virtual Objects represented by different types os media (audio, video or AR models) are managed by the Component Manager. At last, the Author Manager controls controls the accounts of game designers in our tool.

The game server uses a relational database to store all data about games and their configuration (mission and mechanics), game objects as well as players and authors, except media files such as MP3 or video files. These files are stored in a regular file-system. The Database keeps only the information about the local where each file is kept.

4.3 Technology

A mobile application was developed to execute the games designed on the authoring tool. The software runs on Android devices and uses JSON to exchange data with the server. For each game, the tool builds a JSON containing all game data, which is sent by the server to the mobile application to be rendered.

A challenge when running multiplayer games is the need to keep the game state synchronized between server and client. Therefore, the system uses GCM (Google Cloud Messaging) push messages to notify mobile applications whenever the game state is changed. When the mobile device receives the notification it sends a request for and updated game state to the server, hence repeating the JSON data exchange.

A novel feature of our system is the use of Augmented Reality virtual objects. The rendering of these objects is made using the libGDX library. LibGDX is an open-source framework used on both 2D and 3D game development. The device uses its camera view AR objects, that are positioned according to the GPS location. A gyroscope sensor is also necessary to handle the camera turns and correctly implement AR immersion.

5 Evaluation

We have developed a game to exemplify the use of the system, and yield a better understanding of the authoring tool. The game was designed to exploit the majority of resources provided by the tool, such as types of mechanics, sharing missions, competition between groups, collaboration within groups, etc. The main idea behind the game is to allow two groups to explore numerous places according to a set of missions. Since such groups compete with each other we allowed players to interfere directly on their opponents' game.

5.1 Case Study - Battle for Fortaleza

The game involves a battle between two teams (cangaceiros and jagunços) in conflict for land ownership, in this case, the city of Fortaleza. Each team has a headquarter identified by a fort. The goal of each team is to capture an object in the headquarters of the opposing team, and bringing it to its own headquarters. Its game design is based in the traditional "Capture the Flag" style, enhanced with a combat mode between players.

Each player, being a jagunço or cangaceiro, has a common property that represents a limited amount of vital energy (life), and the ability to capture their enemies. The city of Fortaleza have several pitfalls, such as mines. When players are hit by these pitfalls, a certain amount of their vital energy is decreased, which may even eliminate them from the game. In fact, there are other objects spread in the map that could be used by players:

1. Medication kits: They are used to increase player's vital energy;
2. Barricade: A barrier that protects the headquarters of each team;
3. Missiles: weapons used to destroy the barricade of the enemy's headquarters;
4. Explosives: they are used to destroy the enemy's headquarters;
5. Enigmas: These objects represent quizzes, challenges for each team. By solving these quizzes, new weapons and objects are unveiled;
6. Flag: It represents the object to be captured by each team.

Each one of these elements are captured or activated when players come close to them in the map. Some objects are triggered implicitly (such as pitfalls), but other may require an explicit interaction. Players should stand on the object he wants to capture and explicitly notify to load the object. Each team has a limited number of objects.

Players may also interact with each other. One possible interaction is the idea of one player capturing an enemy, which is done when they are close enough to perform a "capture" action. When the enemy captures a player, he is automatically eliminated from the game.

The dynamics of the game involves teams setting traps and solving puzzles to get weapons and other objects of the game. These puzzles typically refer to capture different kind of information such as text, images or videos around the map. At the startup phase, each team gets only the location of its headquarters, as well as a set of clues to puzzles that need to be resolved for other objects to be unlocked. The main objects are missiles and explosives. They are required to destroy the barricade and the headquarters of the enemy team, respectively. In fact, each can only capture the enemy flag after destroying the enemy's headquarters. In turn, headquarters may only be destroyed when its barricade is demolished. After that, the enemy flag can be captured and players should take it back to their own headquarters. Players can use a chat to define their strategies. The first team to capture the enemy's flag to your headquarters is the winner.

5.1.1 Game Editor

The Game Editor proposed in this research uses the application model described in section 4.1 to configure the missions and tasks to be accomplished in the game. Game authors must authenticate through their usernames and passwords to access the game editor. Once an author is logged in, he has access to the list of games previously created where he can edit the missions or even create a new game.

An overall view of game creation starts by defining groups of users, then creating groups configurations, followed by the design of missions for each group configuration. After all missions are defined, the author creates the mechanics for each mission, and finally publishes the game. A detailed description of the process is presented in the rest of this section.

The first step to create a new game is to define the number of groups and where the game will take place. Figure 5 shows the screen where the author makes these initial settings of the game.

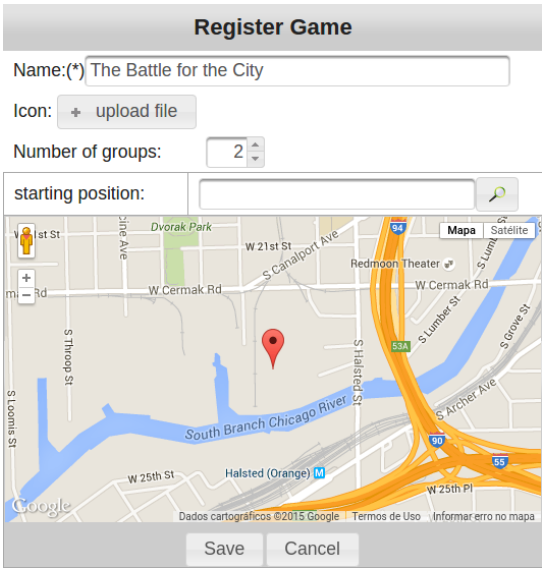


Figure 5: *Screenshot - Choosing the number of groups and the location map*

The second stage of game creation is where the names of the groups is set, as well as set of missions that each group must accomplish. By default, there will be always a configuration set for each team or group and another set shared among all teams. You can add new teams, as well as new sets of missions.

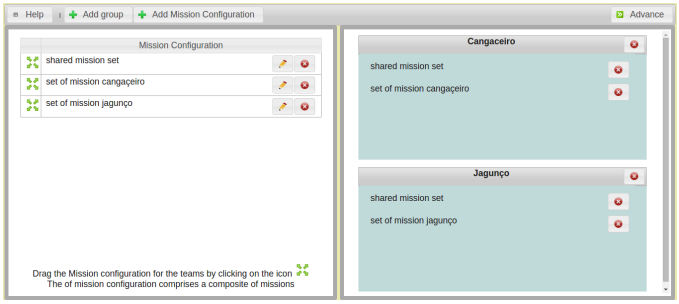


Figure 6: *Screenshot - Configuration for players groups, shared and specific set of missions*

The third step defines the names of missions for each set defined previously. For our study case, the shared missions set has the following missions, as shown in Figure 7: Jagunço Headquarters, Cangaceiro Headquarters, Jagunço Barricade, Cangaceiro Barricade, Explosives and Medication Kits. Only the missions name must be defined at this point. Each mission configuration will be seen ahead. The set of missions for a specific group defines the tasks that will be accessed by this group only. Although some missions have similar goals (such as capture missiles), each mission should be activated differently by each group.

After the first configuration steps, the author is led to the main screen for setting missions and game mechanics, as shown in Figure 8. Mission settings are shown on the right side of the editor screen. A toolbar with the main actions the game designer can perform is displayed at the top of the game editor. On the left side are shown specific information about the mission currently being edited, as the list of tasks that must be completed to enable it.

The Game Editor uses the notation described in the mission modeling section. Missions and mechanics are illustrated as ellipses,

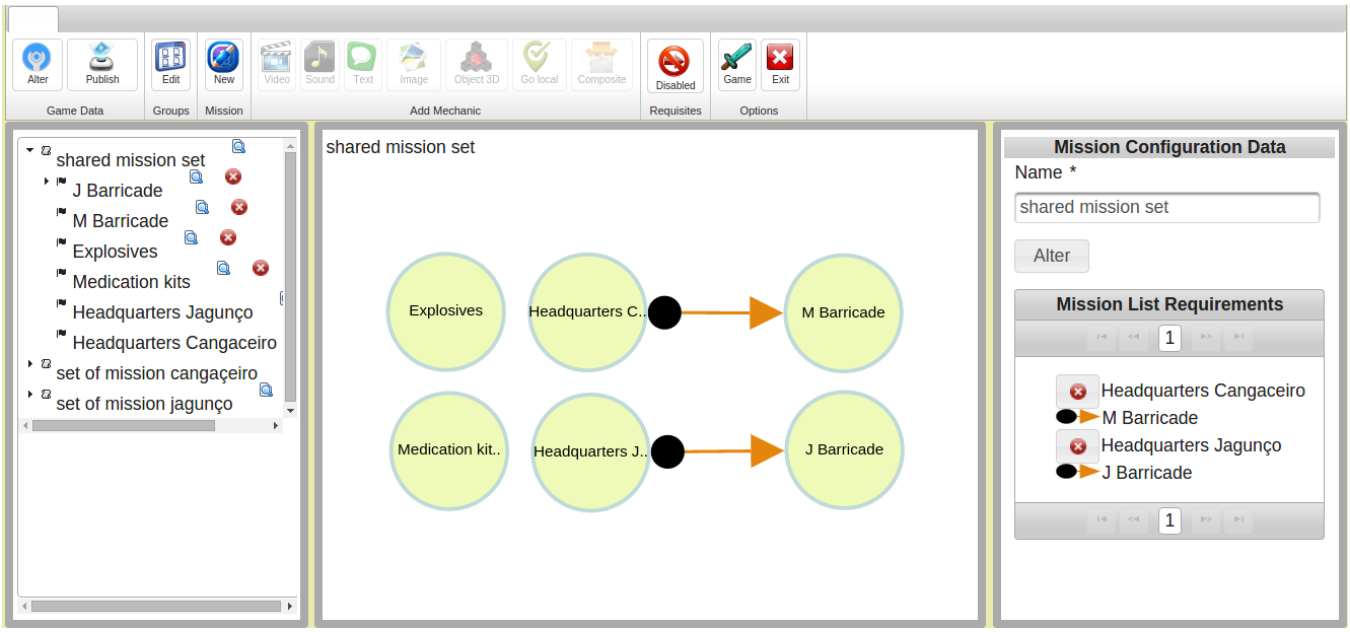


Figure 8: *Screenshot - Main Screen for Shared Missions*

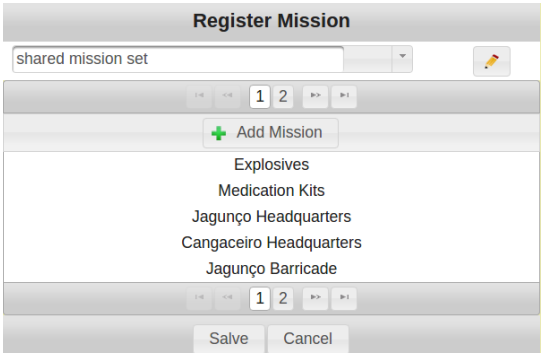


Figure 7: *List of shared missions*

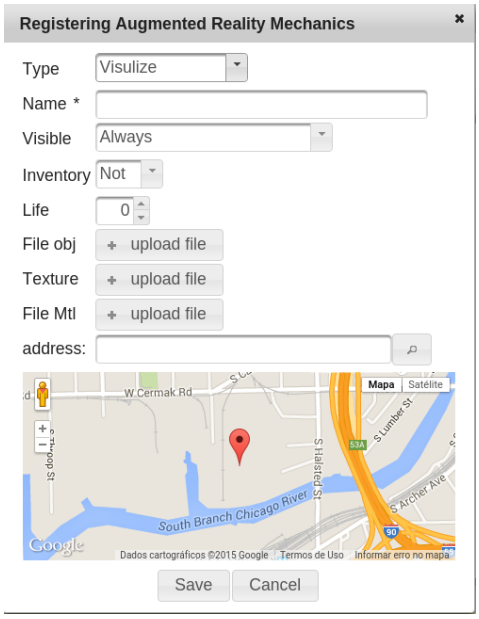


Figure 9: *Inserting an AR object*

which can be interconnected with arrows to illustrate the order that they must be performed. In the Figure 8, the barricade missions have precedence over those related to the headquarters. In a mission definition, game designers can indicate which groups can perform it. If a mission is shared for more than one group but only one can accomplish it, the other groups will only view the status of this mission. This is the case for "General Headquarters" of each respective group. Although each group is responsible for building their respective barricades, the opponent group you can see when this mission was completed.

For each mission, the game designer may include game mechanics as those part of the our application model. The Toolbar offers a button for mechanics related to different types of media: text, audio, video and 3D objects. Toolbar also offers a button for a mechanic related to the shift to any point on the map. Choosing one of these options opens a dialog where additional information box must be provided to complete the mechanic configuration. For example, Figure 9 shows the setting for viewing an AR object, which requires a file in the Wavefront OBJ File Format, and other texture file. It is possible to indicate if performing this mechanic will capture, display or release an object.

The inventory options and life respectively indicate whether objects linked to the mechanics in question will be attached or removed from player's inventory, and if there will be gain (positive values) or loss (negative values) of life energy when the mechanics is performed. The visibility option allows four choices: (i) a mechanic may be always visible, no matter in which state it is, (ii) never visible, which will only be activated when a player comes close to the associated location to the mechanics, but it will never be visible, (iii) unlocked, which means that it will be visible only when in the "ready" state, ie, all its requisite missions were accomplished, and (iv) unrealized, that indicates that it must be turn invisible when it comes to the finished state. Toolbar also offers options to create composite mechanical and set precedence between missions and mechanics, as well as self-publishing the game.

The mission to destroy the barricades will be composed of five mechanics ordered sequentially, as depicted in Figure 10. The first mechanics called "Leave missile" requires that one player uses an inventory item representing a bomb or missile, which necessarily is achieved with the resolution of puzzles or clues. It will enable the second mechanic - "Destroy Barricade". The third and fourth mechanics have the same idea of the previous, but regards to the headquarters destruction. Both second and fourth mechanics use a video that is presented to the player to represent barricades demolition and enemy headquarters' conquest. The "Catch the flag" is represented by the fifth (and last) mechanic, which uses an augmented reality object. According to Figure 10, it will be enabled only when all previous missions were accomplished.

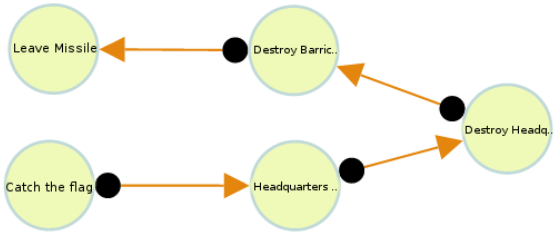


Figure 10: *Modelling Mission - Destroy Barricade*

A remaining feature of the authoring tool is related to composite mechanics, which was used in two missions: "Question 1" and "Answers". The former is composed by four mechanics, being one of them a composite mechanic, which is represented by a blue ellipse in Figure 11.

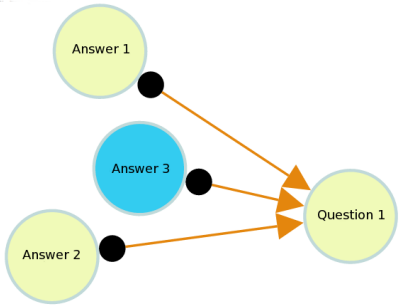


Figure 11: *Example of a Complex Game Mechanic*

When selecting a composite mechanics, the game designer can view its details, just like an ordinary mission model of its basic mechanics, as shown in Figure 12. This feature will help authors to save space and provides an alternative for complex diagrams when modeling mechanics.

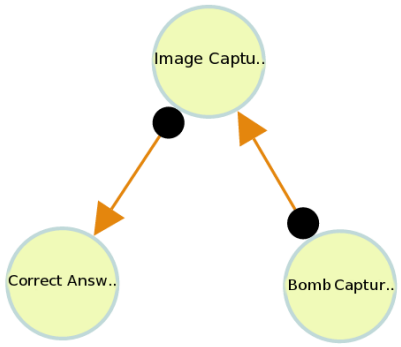


Figure 12: *Example of a Complex Game Mechanic*

5.2 Evaluation with architecture students

In this section we present an evaluation conducted aiming to verify the usability and usefulness of the authoring tool for users with no programming background. We have observed users while designing a previously specified game to later apply an evaluation questionnaire. In this section we show details about the experiment, the material used in the evaluation and discuss the results of the analysis.

5.2.1 Students Profile

The participants were two graduate students chosen for not having prior programming background and having never used an authoring tool to design games. Furthermore, these users are members of a research group that aims to develop location-based applications for numerous purposes.

5.2.2 Materials and Instruments

The questionnaires were composed of inquiries about the features, usefulness and easiness of the tool to create games. For instance, users responded questions about how straightforward the application was to add mechanics, to define the game flow, to order the missions, and to insert augmented reality objects into the game. The evaluation questionnaire used the Likert scale [Barbosa and da Silva 2010].

During the evaluation, the authoring tool executed on a notebook connected to the Internet and the mobile application executed on a Motorola Moto E smartphone with Android operating system.

5.2.3 Procedure

The evaluation was divided into four phases. In the first step a prior questionnaire was applied to ensure users do not have programming background. The second phase consisted of a short presentation of the tool, its features, and the main concepts employed. In the third step, users were given a notebook running the authoring tool to design a previously specified game. Finally, the evaluation questionnaire was applied in the last phase to assess the authoring tool.

The game participants had to design was composed of two groups of players, four missions and eleven mechanics. The missions were named introduction, track, selfie and treasure collection. The first mission contained a simple mechanic to view a video, the second mission was composed by a simple mechanic to capture a picture and a composed mechanic formed by four simple mechanics that required players to visit specific places. The simple mechanics required the composite mechanics to be unblocked. The third mission included a mechanic in which the player had to move a determinate place and take a picture to be stored in user's personal content. The last mission was composed by two mechanics that require players to drop items from user's personal content at specific places to unlock the last mechanics that guided the player to rescue the treasure. Despite being simple, the game explored the main concepts of the authoring tools, like mission ordering, digital content interaction and mission sharing between groups of players.

5.2.4 Results

The first user completed the development of the proposed game in sixteen minutes while the second completed the development in fourteen minutes. Answers of the post-test questionnaire showed all the participants have developed the requested game successfully. Both users evaluated the tool as easy to use. However, one of the participants has required technical assistance to apply the tool for certain concepts and configurations.

One of the questions was related to the tool's ability to provide easy means of ordering the missions in priority lists.It was asked whether the tool allows the correct ordering of missions. According to the participants, it was easy to create an orderly and simple way game by following the steps imposed by the tool. Another item assessed by the participants was related to the sharing of tasks between groups. In this case, we aimed to assess whether this feature presents easy to be set in the tool. The two evaluators agreed fully with the ease of use of this function. In addition, users considered relevant the mechanical related with images, sound, video, and 3D object to create location-based mobile games.

The results indicate the proposed authoring tool is easy to use and allows users with no programming background to design location-based games. However, we attested the tool's user interface requires improvements, since during user evaluation, participants were confused with some icons, concepts and configurations.

6 Conclusion

This work presents an approach for designing and managing location-based mobile games. We proposed an authoring tool that allows users with no programming skills to create singleplayer and multiplayer games. A novel feature of this work is the ability to order missions and mechanics, thus providing flexibility when defin-

ing the game flow. Moreover, the tool allows the use of multiple digital content, including augmented reality objects, that can be inserted in the game, hence increasing player's immersion.

A key contribution of this work is the development of an authoring tool that can be used by users with no programming background, like professors, game designers, etc. to design a wide variety of games.

In this work, we have used the focus groups methodology, which proved efficiency in providing the necessary information about game scenarios. Focus groups' meetings highlighted the importance of an authoring tool that could successfully build a diversity of games, including traditional ones, such as capture-the-flag and tag. The meetings also were important to model a collection of mechanics to be used in location-based games and to establish the guidelines for the design of the tool and the games it generates.

As future work, we intend to incorporate into the authoring tool features for monitoring the games' execution. For example, the tool can generate reports of individuals who are participating and the places they have visited so far (a user trajectory on a map).

A usability evaluation of the authoring tool will also be implemented with a larger group of users. Our aim with this new assessment is to understand the advantages and disadvantages of creating games with the proposed approach. This evaluation will also include acceptance tests of the created games.

Additionally, we intend to conduct a focus groups with tourism professionals and teachers to gather a specific feedback on characteristics and features that can enhance the authoring tool in generating touristic and educational games.

References

Basic Structure of ARIS - ARIS - The Manual. `http://manual.arisgames.org/documentation/basic-structure-of-aris`. Accessed: 2014-02-12.

ASCHIDAMINI, I. M., AND SAUPE, R. 2004. Grupo focal: estratégia metodológica qualitativa: um ensaio teórico. *Cogitare enferm 9*, 1, 9–14.

BALLAGAS, R., AND WALZ, S. 2007. Rexplorer: Using player-centered iterative design techniques for pervasive game development. *Pervasive Gaming Applications 2*.

BARBOSA, S. D. J., AND DA SILVA, B. S. 2010. *Interação humano-computador*. Elsevier.

CELINO, I., CONTESSA, S., CORUBOLO, M., DELL'AGLIO, D., DELLA VALLE, E., FUMEO, S., KRÜGER, T., AND KRÜGER, T. 2012. Urbanmatch-linking and improving smart cities data. In *LDOW*.

DO PRADO RAFALSKI, J., DOS SANTOS, O. L., AND DE MENEZES, C. S. 2013. Um editor colaborativo para descrição de aventuras pedagógicas locativas com realidade ampliada. 283–288.

GAMMA, E., HELM, R., JOHNSON, R., AND VLISSIDES, J. 1994. *Design Patterns: Elements of Reusable Object-Oriented Software*. Pearson Education.

GUTIERREZ, L., NIKOLAIDIS, I., STROULIA, E., GOUGLAS, S., ROCKWELL, G., BOECHLER, P., CARBONARO, M., AND KING, S. 2011. far-play: A framework to develop augmented/alternate reality games. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, IEEE, 531–536.

IHAMÄKI, P., AND LUIMULA, M. 2013. Let's go geocaching: Understanding users enjoyment in geocentria application.

KÖHLMANN, W., ZENDER, R., AND LUCKE, U. 2012. Freshup—implementation and evaluation of a pervasive game for freshmen. In *Pervasive computing and communications workshops (PERCOM workshops), 2012 IEEE international conference on*, IEEE, 691–696.

LINAZA, M. T., GUTIERREZ, A., AND GARCÍA, A. 2013. Pervasive augmented reality games to experience tourism destinations. In *Information and Communication Technologies in Tourism 2014*. Springer, 497–509.

LUNDGREN, S., AND BJORK, S. 2003. Game mechanics: Describing computer-augmented games in terms of interaction. In *Proceedings of TIDSE*, vol. 3.

MELLO VIANA, J., PONTE VIANA, N., MOTA TRINTA, F., AND VIANA, W. 2014. A systematic review on software engineering in pervasive games development. In *Computer Games and Digital Entertainment (SBGAMES), 2014 Brazilian Symposium on*, 51–60.

MOEBERT, T., ZENDER, R., AND LUCKE, U. 2011. A pervasive educational game on pervasive computer networks. In *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, vol. 2011, 1543–1552.

RUBINO, I., XHEMBULLA, J., MARTINA, A., BOTTINO, A., AND MALNATI, G. 2013. Musa: Using indoor positioning and navigation to enhance cultural experiences in a museum. *Sensors 13*, 12, 17445–17471.

WAKE, J. D. 2013. *Mobile, location-based games for learning: Developing, deploying and evaluating mobile game technology in education*. The University of Bergen.

WETZEL, R., BLUM, L., AND OPPERMANN, L. 2012. Tidy city: a location-based game supported by in-situ and web-based authoring tools to enable user-created content. In *Proceedings of the international conference on the foundations of digital games*, ACM, 238–241.