A Model-based Approach for Designing Location-based Games

Cristiane Ferreira[1]* Luís Fernando Maia[1,3] † Carlos Salles[2] ‡ Fernando Trinta[1] § Windson Viana[1] ¶

[1]Federal University of Ceará, GREaT-MDCC, Brazil
[2]Federal University of Maranhão, Computer Science Department, Brazil
[3]Federal Institute of Maranhão, Campus Caxias, Brazil

ABSTRACT

Location-Based Games (LBGs) are a subclass of pervasive games that make use of location technologies to consider the players' geographic position in the game rules and mechanics. This research presents a model to describe and represent LBGs. The proposed model decouples location, mechanics, and game content from their implementation. We aim at allowing LBGs to be edited quickly and deployed on many platforms. The core model component is LEGaL, a language derived from NCL (Nested Context Language) to model and represented the game structure and its multimedia contents (e.g., video, audio, 3D objects, etc.). It allows the modelling of mission-based games by supporting spatial and temporal relationships between game elements and multimedia documents. We validated our approach by implementing a LEGaL interpreter, which was coupled to an LBG authoring tool and a Game Server. These tools enabled us to reimplement a real LBG using the proposed model to attest its utility. We also edited the original game by using an external tool to showcase how simple is to transpose an LBG using the concepts introduced in this work. Results indicate both the model and LEGaL can be used to foster the design of LBGs.

Keywords: Location-based Games, Game Modeling, Multimedia Document.

1 INTRODUCTION

Digital games have evolved significantly over the past two decades, specially due to the influence of technological advancements in hardware and software. Games for PCs and console have become a dominant form of entertainment. They provide a high level of attractiveness by creating imaginary and interactive virtual environments. These games use realistic graphics and sounds to immerse the player into a virtual world [13].

The development of mobile technology has broadened the digital platforms on which games can run. Its peculiar characteristics (e.g., embedded sensors, communication capacity, and omnipresence) enabled the popularization of pervasive games. In fact, before the digital era, traditional games were designed and played in the physical world, relying on real-world properties such as objects, physical space, etc. [21]. Pervasive games are capable of mixing these two genres by integrating virtual and real environments using mobile and ubiquitous technologies [13]. Nowadays, most pervasive games use smartphones' sensors to infer the player's context (e.g., location, nearby objects) and introduce this knowledge in the gameplay. This research focuses on modelling a popular subtype of Pervasive Games called Location-Based Games (LBGs). LBGs use the players' location to modify the game state during runtime. Pokémon GO and Ingress are notorious examples of LBGs. In these games, players have to move in the real world to progress and reach goals in the game.

The process of developing an LBG is a multidisciplinary activity involving distinct professionals, such as artists, interface and sound designers, and also mobile developers. A successful game project requires a clear communication among team components, specially regarding its game design. In addition, many tools are used by this multitude of professionals and the integration between them is not always an easy and harmonious process. Some digital game modeling approaches use UML (Unified Modeling Language) for this task [29]. Other research goes further and proposes DSLs (Domain-Specific Languages) [11] [8] for game modeling and representation. Another alternative to this challenge is the use of LBG authoring tools. This kind of software offers a set of resources to both game designers and developers for integrating game content with gameplay (e.g., text, image, mission workflow, game map, etc.). Despite these examples (some of them applied to pervasive games), there is no well-defined model for representing LBGs. Some LBG authoring tools use their proper notations for the visual representation of game workflow, its mechanics, and related multimedia. However, there is no uniformity between these representations.

The absence of an explicit model for LBGs (i.e., the game structure and its media) impairs their understanding by the professionals involved in the development of these games. In the case of the LBG authoring tools, this absence may generate doubts and misunderstanding of the game mechanics by the authors. It also hinders its evolution and integration with third-party tools (e.g., testing tools, game-checking tools, transposition tools). For instance, games developed using specific tools can't be deployed to other platforms easily, include features from other authoring tools, and even be transposed to multiple places. Therefore, this work addresses these issues by introducing a fully featured model-based method for designing LBGs. The core element of this model is a declarative language inspired by the multimedia language NCL (Nested Context Language). The developed language inherits NCL concepts that guarantee the representation of both temporal and spatial aspects of these games. We maintain compatibility with NCL tools, which allows edition, interpretation, and testing of games described in the new model.

The proposed language is called LEGaL (Location-based Game Language). It allows intuitive and explicit modeling of LBGs mechanics and rules. Moreover, LEGaL includes the media used in each mechanic of such games (e.g., images, videos, animations, 3D objects), with support for temporal aspects, as well as spatial relations of these multimedia documents. To validate this work, LEGaL was coupled to an LBG authoring tool called LAGARTO [32] using a LEGaL XML representation, and an XML parser that interprets LEGaL documents and generates games on the LAGARTO's Game Server [24]. We evaluated the proposed model with the following research questions in mind:

(Q1): Is it possible to rewrite an LBG using the game modeling introduced in this work?

(Q2): Can the proposed approach be coupled to external algo-

*e-mail: cristianeferreira@great.ufc.br
†e-mail: luis.maia@ifma.edu.br
‡e-mail: csalles@deinf.ufma.br
§e-mail: fernandotrinta@great.ufc.br
¶e-mail: windson@great.ufc.br

rithms and tools?

In order to answer such research questions, we have structured the evaluation of the proposed model in two phases. First, we rewrite an audio-based treasure hunt game, called AudioRio by using a textual representation of LEGaL. Thus, we would like to demonstrate its usefulness in describing a real LBG. Second, we use the LEGaL code representing the game to apply an existing algorithm and transpose the LBG to a different city. The evaluation assesses both the usefulness of the presented model and expressiveness of LEGaL to address key challenges in the development of LBGs. Results indicate the proposed model is able to describe most LBGs and can successfully be used by many authoring tools and platforms to enhance the development of such games.

The reminder of this paper is organized as follows. Section 2 presents the concept of LBGs. We present the principles and objectives of our model and the LEGaL language in Section 3. In Sections 3.4 and 4, we consider the elements reused and those embedded by the NCL extension. Section 5 and 6 present the proof of concept and the experiments conducted to validate this work. Finally, in Section 7, we present our final remarks and future work.

2 Background

In order to conceive a general approach for designing LBGs, we conducted a study about the games and their patterns. Also, we investigated existing authoring tools used to build LBGs. This section describes the results of this research.

2.1 Location-Based Games

Location-Based Games (LBG) are a subtype of pervasive games. These games use location-based technologies and include the players' position in the rules of the game [7]. The game rules often demand that players move to certain locations in the real world, which may be an absolute position (such as GPS coordinates) or a relative one. At last, LBGs promote a "double perception" of the game space: the physical or real world, and the digital or virtual world of the game, but establishing a link between both spaces.

2.1.1 LBG Patterns

The gameplay in most LBGs requires tracking player's absolute location or their movement. However, the purpose and sequence of player's movements to achieve their goals in the game defines distinct game patterns. In this case, LBGs implement one or more of these patterns, eventually combining some of them to create more complex gameplay. The game patterns present in LBGs were classified by [17] into four types: (P1) *Search-and-Find*, (P2) *Follow-the-Path*, (P3) *Chase-and-Catch*, and (P4) *Change-of-Distance*. Figure 1 illustrates them.

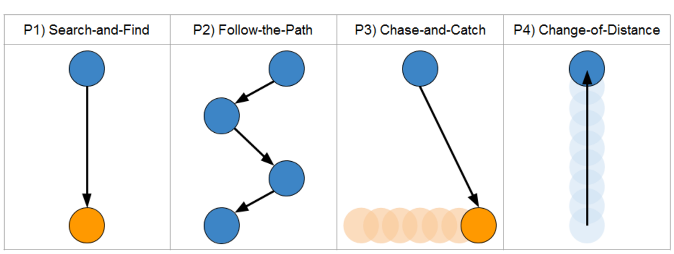


Figure 1: LBG patterns. Source: Adapted from [17].

Search-and-Find (a.k.a Seek and Find) is a popular pattern that requires players to find a target location on the game map, and move to this location. Usually, LBGs implement this pattern by providing tips for each player guiding them to the target location. In this case, the searched location is fixed, therefore, players establish a mapping between locations of both virtual and real worlds. It is common, for LBGs to implement the *Search-and-Find* pattern for collecting items, such as boxes in *Geocaching* [25] and *pokémons* in *Pokémon Go* [16].

The *Follow-the-Path* pattern is similar to *Search-and-Find*, however, it focuses on following a predefined route, thus encouraging players to visit several consecutive places before reaching a final location. In some cases, players can receive penalties if they diverge from the specified path. For instance, in the game *Tourality* [39], players can race each other, and taking shortcuts is punishable. Due to its characteristics, this game pattern is very popular among cultural and touristic LBGs.

Chase-and-catch is also a popular LBG pattern, where players must follow a moving object in the virtual world. This object may be another player or an ordinary game object, such as an NPC. In other words, the target's location changes frequently. The pattern was implemented in a famous multiplayer LBG called *Shadow Cities* [3] whose players were divided into two different groups aiming at chasing each other.

The pattern *Change-of-distance* differs from the others due to its focus on the players' displacement, regardless of fixed locations or predefined directions. In this case, players must move from any place and towards any direction as the important aspect is the movement. Consequently, *Change-of-distance* is very popular among fitness and health games, such as *Zombies, Run!* [38] and The Walk [35].

It is important to highlight that LBGs sharing the same game pattern are structurally similar, hence presenting equivalent gameplay, interaction, features, and challenges. For instance, Table 1 shows some popular LBGs and the game patterns they use. Consequently, analyzing each game pattern is fundamental to provide a game modeling approach capable of supporting all classes of LBGs available in the market. The proposed model was designed to support all patterns, including a fusion between them. As a result, all of the LBGs surveyed in this work can be described using our model.

2.1.2 LBG Main Features

We found some LBGs in our literature survey. Table 1 resumes some features present in these games. These features guided us in the design of a custom model to describe LBGs, which was used to specify LEGaL. According to the features found in the survey, an LBG can be defined as a set of quests (or missions). In theses games, an individual player or a group of players (they can be arranged in a team), must accomplish missions to win or to progress in the game. For each quest, players interact with different medias, such as audio, video, 3D objects, or text messages. A few LBGs have Augmented Reality features (AR). According to the media type, players may collect, create or even drop these media content in several locations of the game map. These actions are better described at Section 4.2.

Furthermore, LBGs may use one or more patterns presented in Section 2.1.1. The majority of LBGs has map-based visual interfaces. In this case, map markers indicate the quests' location, regions of interest, collectible items, or NPCs. Games may run on singleplayer (SP) or multiplayer (MP) modes. Additionally, we can classify an LBG as a collaborative or competitive game. The former occurs when a group of players needs to help each other to accomplish missions and finish the game. On the other hand, in a competitive game, players compete to perform missions better or faster than others. The winner is the one who completes the game first. In both modes, players can have the same set of missions or different, but equivalent, quests to accomplish.

The game definition described above defines the structure and the mechanics that compose an LBG in our research, i.e., how players interact with the virtual world while moving around in reality. We want to represent this structure and also to model the mechanics related to the multimedia documents of the game and their activa-

Table 1: LBGs surveyed.

LBG	P1	P2	P3	P4	MP	Teams	AR
Geocaching [25]	x						
Parallel Kingdom [27]		x			x	x	
Tourality [39]		x	x		x		
Shadow Cities [3]			x		x	x	
The Walk [35]				x			
SpecTrek [10]		x					
Ingress [15]	x			x	x	x	
Tidy City Scout [37]		x					
Barbarossa [14]	x		x				x
Gossip at Palace [30]			x				
Woody [34]		x					
TARX [18]		x					
BattleSuit Runner [26]				x			
Zombies, Run! [38]				x			
Pokémon GO [16]	x		x	x	x	x	x

tion events (i.e., spatial and temporal triggers). However, our LBG model, at this time, does not address the game visual interface on the mobile user device. Our intention is to create a model that expresses the game behavior only. So, our LBG model does not define screen positioning, colors, text fonts, layout, icons, or map themes used in the game visual interface. The mobile game app must handle these features.

2.2 Game Modeling Approaches

Regarding the development of LBGs, recent surveys [2, 36] researched software development methodologies to create pervasive games and LBGs. The studies concluded that an approach considering all variables and features of pervasive games is needed. However, some works focused on the design of pervasive games by proposing a Domain Specific Language (DSL) [8][11][31] or a meta-model for representing digital games [19][29][23].

For instance, in [11], the authors propose a DSL that allows a user to specify a mind map with the rules of the game. From this map, the DSL engine generates the game code automatically. Another example is the multi-platform gaming DSL proposed in [31]. The games are modeled graphically and then converted to the Haxe language, which can be compiled for different platforms. In [23], the authors propose a gamification framework. It makes use of GaML, a declarative modeling language represented by a meta-model. In this approach, after the modeling, the game is converted into code automatically.

In [19], authors propose a meta-model for assisting the development of educational simulation games. It provides a set of mechanics for modeling the game domain and generates an XML that specifies the game features. The meta-models and DSLs found in our survey are not appropriate to describe all the characteristics present in Table 1. It is worthwhile to mention that [12] is the closest research to the LBG domain. The authors propose the PerGO ontology to serve as the basis for the creation of DSLs to be used in the development of pervasive games. However, the approach does not support the time synchronization required for specifying the game media execution, nor does it support the spatial relationships needed to define the LBGs patterns we have listed in Section 2.1.1.

Since none of the game modeling approaches mentioned above supports all the features and resources demanded by modern LBGs and were designed to be used by specific tools, this work proposes a model-based approach that addresses these issues and can be implemented by any authoring tool and platform. The key component of the proposed approach is a description language designed to be platform independent and support the features like time synchronization, spatial relationships, and the LBG patterns presented in [17].

2.3 Multimedia Languages for Game Representation

We decided to investigate languages for representing multimedia documents aiming at finding a model or a language capable of describing the behavior of LBGs and its media. As a requirement for our approach, this language must be able to model both the missions concept and the temporal presentation events as first-class entities. HTML was the first option, but it is not satisfactory because it can not specify an LBG exclusively in declarative code. A mission should be specified in an imperative code, as a script, increasing the document complexity. Both NCL and SMIL (Synchronized Multimedia Integration Language) are hypermedia languages that state the temporal synchronism as structure-based documents, but they have an important difference. SMIL specifies its synchronism implicitly using the time containers *par* and *seq*. NCL determines the temporal relationships among media contents explicitly using the link and connector elements. Because of that, NCL is more natural to model both missions concept and temporal synchronism as first-class entities.

2.3.1 NCM Model and NCL Language

NCM [33] is a conceptual model with expressive power focused in the representation and handling of multimedia documents. The model represents the abstract concepts and concrete rules about the relationships of these concepts. The model is structure-based and not media-based. In other words, it has an hierarchical organization and it is not just based in the media contents. The main concept of the model is the composite node that characterize the nesting of information. The NCM definition of the hypermedia document is based on the nodes (pieces of media information) and the synchronization relationships of these nodes.

Nested Context Language (NCL) is a hypermedia language that was created based on NCM. NCL is a declarative language for hypermedia documents authoring specified in a modular approach in XML. It is the default language of the SBTVD (Brazilian Standard of Digital TV, in portuguese) and ITU-T recommendation for IPTV. NCL documents specify how media objects are structured and synchronized in spatial-temporal relationships. There are some important requirements that NCL offers to authors, like reuse of contents and structures, high-level abstractions and the separation of multimedia relationships.

2.3.2 NCL restrictions for LBGs support

Although NCL addresses some of the features found in LBGs, it does not fully support them. The most important LBG feature unsupported by NCL is the handling of spatial relationships and events. LBGs require support for spatial relationships between game elements. For instance, each game mission needs real-world coordinates in order to be discovered by game players. Each mission defines a geographical area (aka action area) where a mission is defined. This mission is triggered when players are near to this map area. Other important unsupported features by NCL regarding LBGs are:

- Missions properties: such as, if a mission is mandatory or not to accomplish the whole game;
- Teams and Players features: where groups of players are defined when the game is created;
- Game Mode: which may be single-player or multiplayer, competitive or collaborative games, as explained previously (Section 2.1.2);
- Game Mechanics: i.e., actions that each player must perform to complete a mission;

- Game Scores: i.e., how points are gathered by players;
- Location places: i.e., specific location points in the map where game missions and events are triggered;
- Augmented Realty Support.

3 Location Based Game Model

This work proposes a model-based approach capable of coping with the challenges and requirements demanded by modern LBGs. This game model acts as a mediator between conception and execution, thus working to bridge game design, development, and platform deployment. A key advantage of this method is the possibility to use a single model capable of merging features from existing tools and running the game in multiple platforms. Figure 2 depicts how the proposed game model interacts with other tools and platforms. In this case, an LBG concept can be modeled in a specific document (LEGaL Document) using a formal language. The model document can be created, edited and processed using any tools (e.g., a visual authoring tool), transposition algorithms, and applications, provided they support the proposed language. Furthermore, the model can be deployed on multiple platforms, thus ensuring portability and interoperability. For instance, with a model transformation approach, an Android map-based application can be generated from the game model description.

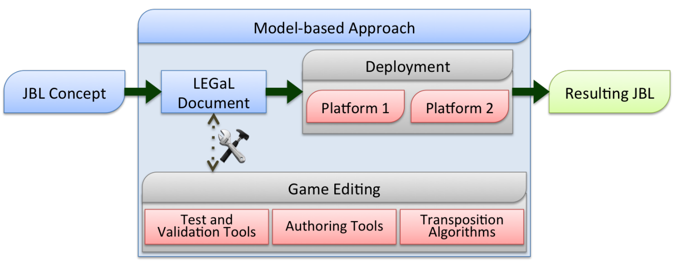


Figure 2: Architecture of the proposed game modeling approach.

Additionally, our game model introduces traits that were neglected in previous works, such as separation between mechanics, content, media, and location data. This modular structure is necessary to allow third party applications and tools to handle only specific parts of an LBG. For instance, in Section 6, we showcase how this work furthers the use of an external algorithm designed to transpose LBGs. In this case, the transposition algorithm will edit only the information related to geographic coordinates, therefore keeping content and media unmodified.

The approach for developing LBGs introduced in this work is built on top of a general purpose game model and hence defining this model is key to the success of the method. In order to provide a reliable game model, we designed a fully featured description language called LEGaL that is responsible for accurately defining multiple LBG patterns while supporting both temporal and spatial properties. Next, LEGaL is detailed along with its features, concepts, media, representation and usage.

3.1 LEGaL

LEGaL (Location-based Games Language) is an extension of the NCL multimedia language that includes specific elements to support the design of LBGs. The proposed language allows the description of rules and mechanics of LBGs. Moreover, LEGaL specifies the modeling of game missions in a text document. This modeling includes the game structure, its components, and specifications about the media used in game mechanics. LEGaL inherits the structure and entities from NCL, as shown in Section 3.4. NCL focuses on digital TV applications and allows the organization of both spatial and temporal aspects of hypermedia elements. As a result, LEGaL benefits from NCL features, such as temporal synchronism used to handle media and a language structure connecting the LBGs' items. In resume, the LEGaL permits the creation of documents defining game missions, temporal and spatial events, and its media content.

We used NCL as the basis for the design of the LEGaL structure and components. Especially due to its ability to describe interactions between users and media. Also, NCL supports basic content such as images, videos, texts, etc. Furthermore, NCL allows media synchronization, ordering media execution, and the use of conditional triggers. This last feature enables the definition of actions to be executed as responses to certain conditions. LEGaL allows the game designer to separate how the game is structured of its execution and its visual interface. In this way, LEGaL is a standard format for exchange the game specification among multiple tools. So, in theory, with LEGaL, a game designer may create an LBG in a particular tool, and run it in another environment that interprets and executes its LEGaL representation. Moreover, LEGaL is the first step to enable syntax and semantic analysis of the game model, which can be extended for model checking it[1][4].

3.2 Language Conception

LEGaL is a declarative language. It is based on the NCM entity concept and has an XML representation. It also inherits the NCL flexibility for temporal synchronism definition. The layout of LEGaL components resembles an NCL document structure. A LEGaL document can be translated into a directed nested graph, where nodes and edges are used to describe LBGs concepts presented in Section 2.1.2. As we stated, LEGaL is designed for extending NCL documents enabling them to model LBGs. In fact, it describes the structure of these games. Thus, the resulting graph of this model represents missions that players must accomplish in an LBG. The graph also includes the composition of missions (media with which the player interacts) and the relationships between the missions and media files. It introduces the game flow, from its inception to its accomplishment. LEGaL graphs use two types of nodes: **context nodes** (composite nodes), and **media nodes**. The former represents nesting of nodes in the graph and the latter describes the associated media files. The edges of the graph represent the relationships between the nodes, i.e., the game flow. In LEGaL, ordering and time synchronisms between nodes are defined by the connectors and links.

3.3 Node Relationships

Relationships between composite nodes, as well as simple nodes, are constructed by causal relationships, in which a condition must be satisfied for an action to be performed. Relationships are multipoint, containing one or more source points, which will trigger an action, and one or more target points, which will be affected by the action. In LEGaL, points in a relationship may describe game missions or media that players must interact with.

For mission nodes, relationships reflect the correct order that missions must be accomplished. In other words, some missions may be marked as prerequisites to other missions, defining the game flow. Synchronism between the media into LEGaL documents, as well in NCL, is done according to Allen relations [1]. Allen proposes a linear model of relations between two sequences that can be used with time intervals.

3.4 NCM Components reused by LEGaL

Entities of NCM 3.0 represent the components of LEGaL documents. These entities are better detailed in [33]. The following components are key to model LBGs using LEGaL: *(i)* the context

[1]Model Checking is an automated technique which systematically exploits states of a system and exposes potential design errors.

Table 2: LEGaL Media types.

type/subtype	File format	type/subtype	File format
text/plain	txt	image/png	png
image/jpeg	jpg, jpeg	audio/mp3	mp3
video/mpeg	mpeg, mpg	video/3gpp	3gp
text/plain	obj, mtl	application/gml+xml	gml

nodes representing game missions; *(ii)* the media nodes representing the media to be displayed and the geographic location of missions; and *(iii)* the connections used to establish relations between nodes. The connections also specify time synchronism using cause and effect sentences and define how links are activated and which actions should be executed.

The media node specifies the media to be displayed when a player executes a mission. The media types supported by LEGaL are text, image, audio, video, and 3D objects, as shown in Table 2. All types conform to the Multipurpose Internet Mail Extensions (MIME) rules [5]. The composed nodes, called **context nodes**, are subgraphs of the modeling graph. LEGaL defines these nodes as a set of media nodes and their relationships. Therefore, context nodes can represent the missions that players have to execute.

The LEGaL **connector** is a component responsible for defining spatial-temporal relations that are synchronized according to a cause. In this case, a condition has to be satisfied to execute an action, as shown in Section 3.3. The links represent the relationship between mission nodes or media nodes. A link is responsible for associating nodes by connectors and establishes a time synchronism between nodes. Synchronism is useful to define an execution ordering for nodes. Therefore a game flow can be specified using this feature.

In the model, nodes and connectors communicate using **interface points**. Anchors and properties are interface points of media nodes that can be accessed by links. Ports are mission nodes. Roles are connectors of points. The ports are used to define where the game starts and to indicate the next internal node to be executed. Defining one or more starting missions for a game is possible. To do that, the game designer defines one or more ports inside the context of a document's body.

4 NCL EXTENSION

4.1 Mission Properties

We add some properties to the context node for representing mission information. For instance, the number of times a mission can be played, which missions are required to be played before playing the others, etc. Table 3 contains four of these properties and its possible values. Listing 1 illustrates a context node defined using the new properties.

```
<context id="mission3">
   <port id="port" component="mdAudio"/>
   <property name="mandatory" value="true"/>
   <property name="occurrence" value="unbounded"/>
   <property name="visibility" value="true"/>
   <property name="requirements" value="mission1
       mission2"/>
   <media id="location" type="application/gml+xml"
        scr="media/location.gml"/>
   <media id="mdAudio" type="audio/mp3" src="media
       /audio.mp3"/>
</context>
```

Listing 1: Context node with additional properties.

4.2 Game Action and Score

There are four types of actions supported by LEGaL: `execute`, `create`, `collect`, and `drop media`. The `execute` action consists of exhibiting one or more media, such as playing an audio or video, displaying a text or visualizing a 3D object. The `create` action allows players to produce game media, like images and videos. Additionally, the `drop media` action enables players to place a media in a determined location and the `collect` action allow players to collect media placed in a specific location. The `action` parameter was created to store the desired action when defining a media. The parameter receives an integer value between 0 and 3 representing the corresponding action.

During runtime, a player is rewarded with a score for every mission or action completed. The same actions presented in Section 4.2 can have a matching reward. To implement such system, a `score` property was added to missions and media, thus defining the reward for each action executed by the players. This parameter can assume positive integer values.

4.3 Location Data of Game Missions

In the proposed language, a media node has information about the location of game missions. The node type is `application/gml+xml` and consists of a GML (Geography Markup Language) with ".gml" extension. GML [6] is an XML extension developed to express geographic features. The extension uses points, lines, polygons and geometric shapes defined by Cartesian coordinates and associated with spatial reference systems. LEGaL uses GML to specify activation areas for a game mission, thus being able to describe a polygon representing a mission's location. Listing 2 illustrates a media node representing a mission location.

```
<media id="location" type="application/gml+xml"
    scr="media/location.gml"/>
```

Listing 2: Mission Location.

4.4 Spatial Relations in Games

The location of players and missions' activation areas is key to the gameplay of LBGs. In this work, an activation area is a planar region defined bi-dimensional c oordinates. These areas can be defined as regular or irregular polygons, and c ircles. The spatial relation between activation areas is the foundation to the RCC (*Region Connection Calculus*). The RCC is a vital model to define topological relations in bi-dimensional space. The model defines eight basic relations between two areas: *disconnected, externally connected, equal, partially overlapping, tangential proper part, tangential proper part inverse, non-tangential proper part, non-tangential proper part inverse* [28].

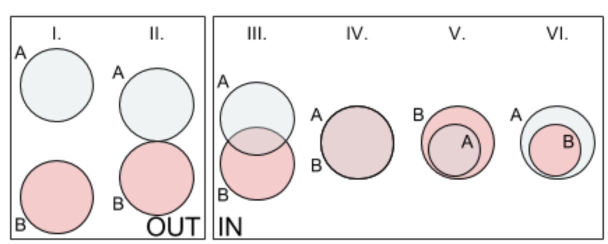


Figure 3: Spatial relations between missions and players. Figure adapted from [28].

Recently, RCC has been used to establish relations between presenting regions in an NCL document, thus defining media positioning in space [9]. Figure 3 illustrates the spatial relations used in the

Table 3: Node properties

Property	Values	Description
mandatory	"true" or "false"	Defines whether a mission is mandatory
occurrence	Positive integer	Sets how many times the mission can be executed
visibility	"true" or "false"	Indicates if a mission can be executed
requirements	List of values	Stores a list of required missions

LBG model presented in this work. Regarding game elements, the circles represent activation areas for missions (A) and player (B). Each column depicts a positioning state for these elements, and the relations are based on these states. In states I and II, the player is considered out of an activation area for a mission, thus the mission cannot be executed. In states III and IV, there are intersections between activation areas, therefore the player can start the mission.

4.5 Location Events for LBGs

In order to handle the relations between space and players' actions, we added a set of events to LEGaL. As a result, the following events can be linked to media nodes: `onEntering`, `onLeaving`, and `onStaying`. `onEntering` is triggered when a player enters the activation area of a mission. Conversely, `onLeaving` is launched when a player exits an activation area. Finally, `onStaying` is triggered if a player remains in an activation area during a determined time. Listings 3 and 4 show examples of connectors and links, respectively. In this case, a media is executed when a player enters an activation area.

```
<casualConnector xconnector="onEnteringStart">
   <simpleCondition role="onEntering"/>
   <simpleAction role="start"/>
</casualConnector>
```

Listing 3: Connector using the onEntering event.

```
<link xconnector="onEnteringStart">
   <bind role="onEntering" component="location"/>
   <bind role="start" component="audio"/>
</link>
```

Listing 4: Relation using the onEntering event.

4.6 Augmented Reality Media

LEGaL supports the use of media nodes to represent Augmented Reality (AR) content in games. In this case, it is necessary to specify a ".obj" file containing the points of a 3D object, a ".mtl" file describing information about the surface of the object, and optionally a ".png" file containing a texture to be applied to the 3d model. Listing 5 illustrates a media node defining an AR object in the game. The media node uses the `src` parameter to specify the ".obj" file and the other files are specified using two distinct properties.

```
<media id="objeto3d" type="text/plain" scr="media/
    objeto3d_polygon.obj">
   <property name="mtlFile" value="
       objeto3d_material.mtl"/>
   <property name="texture" value="
       objeto3d_texture.png"/>
</media>
```

Listing 5: Augmented Reality definition.

4.7 XML Representaton

The textual representation of LEGaL is an XML document. The block structure defines the grouping of language components. As in an NCL application, the document must have a definition header (`<ncl>`), a program header (`<head>`), a program body (`<body>`), and the closing of the document (`</ncl>`). The elements `<head>` and `<body>` must be declared as children of the `<ncl>` element. Definitions of descriptors and connectors are made in the document header, in their respective code blocks. Context and media nodes, links, and other components are defined in the body of the game document. NCL language tokens are used in the definition of the LEGaL document to describe the components and the behavior of the game. Table 4 summarizes the document components, belonging to NCL, which are used in our extension.

First, the game developer must specify the GML document, which contains geolocation points related to the missions. Then, he should define the descriptors, which detail how the game media will execute. After this step, the connectors and links are identified, and then the media must be specified. The next step is to set the ports for the flow composition of the game missions. Listing 6 exemplifies the basic document structure that LEGaL uses to specify an LBG.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ncl id="" xmlns="http://www.ncl.org.br/NCL3.0/
    EDTVProfile">
   <head>
      <descriptorBase ... > <!-- descriptors -->
          </descriptorBase>
      <connectorBase ... > <!-- connectors --> </
          connectorBase>
   </head>
   <body>
      <port ... /> <!-- initial ports -->
      <context ... > <!-- missions -->
         <port ... />
         <media ... > ... </media> <!-- media
             objects -->
         ... <!-- relationships between media -->
             ...
      </context>
      ... <!-- relationships between missions -->
          ...
   </body>
</ncl>
```

Listing 6: Basic code structure of LEGaL.

4.8 LEGaL Parser

We have developed a *parser* for LEGaL documents and added it to the LAGARTO tool (*LocAtion-based Games AuthoRing TOol*) [24]. LAGARTO allows the visual modeling of LBGs and generates a representation of the games in a database using the Hibernate framework. The tool includes a mobile application that runs LBGs and connects them to the LAGARTO's Game Server, which, in turn, accesses Hibernate.

We then discard the visual part of the tool and created a new way to generate games in the LAGARTO database by using a LE-

Table 4: LEGaL's main elements and attributes.

Category	Element	Token	Parameters
Content	Media node	<media>	id, type, scr, descriptor
	Context node	<context>	id, mandatory, times, visibility, requirements
Relationship	Connectors base	<connectorBase>	id
	Connector	<connector>	id, condition, action
	Link	<link>	id, xconnector
	Bind	<bind>	component, interface, role, delay
	Port	<port>	id, component
	Anchor	<area>	id, begin, end
Presentation	Descriptors base	<descriptorBase>	id
	Descriptor	<descriptor>	id, duration, opacity, volume, fontSize, fontColor, fontWeight

GaL document, the associated media, and a GML document as input. Figure 4 illustrates the new tool configuration. The LEGaL parser receives the game model (a ".ncl" file) and the media files described in that document as input. Then, it parses the XML document and generates the mechanics and associations that define the game in the LAGARTO's Game Server by inserting objects into the database using the Hibernate framework. The output of the process produces an LBG ready to be executed by the LAGARTO Scout mobile application. In the next section, we present an evaluation conducted with these tools and extensions to validate them.

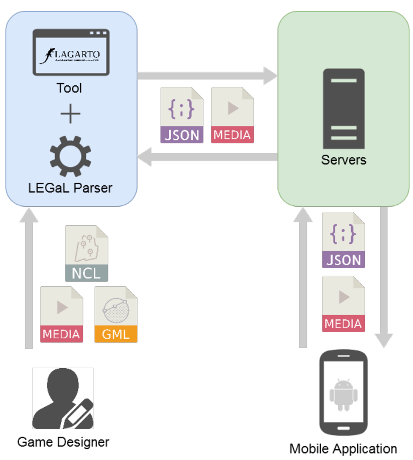


Figure 4: LEGaL Parser integration.

5 Evaluation

We tried to evaluate whether LEGaL is compatible with NCL tools and if experts in this language could understand the concepts introduced by LEGaL. To do that, we conducted a qualitative research with 5 NCL experts (graduate students with NCL skills). These users were unaware of LEGaL, although they knew the central concepts of LBGs. The evaluation was divided into three steps. First, we presented LEGaL concepts to them (30 minute training). In the second step, the testers were invited to create two predefined "simple" LBGs by using NCL editing tools. The last step was conducted after the testers concluded the activity, they were encouraged to discuss LEGaL's expressivity and its compatibility with NCL.

5.1 Expressivity and NCL compatibility

In general, as a result of the test, the developers evaluated LEGaL positively. All testers were able to successfully generate the games proposed by the evaluation activities. Although the group of samples is small, all of them evaluated LEGaL positively, thus indicating the language has a good acceptance. Moreover, testers claimed using LEGaL is simple for anyone who has prior knowledge of NCL. One of the developers stated: "Someone with prior knowledge of NCL can quickly develop a game with this extension. Also, NCL is an easy-to-learn language and is very accessible". Additionally, a key feature that simplifies the development of LBGs with LEGaL is the ability to insert location data into the game as media abstractions using GML.

Additionally, we expect people with no programming background -e.g. game designers and game enthusiasts- can learn simple descriptive languages such as NCL and LEGaL. Therefore, this possibility can increase the number and types of LBGs, as well as, promote the design of new tools and methods that can use LEGaL ease the development of LBGs by people with no programming skills.

5.2 Proof of Concept

Once we were sure that the language had good expressiveness and was compatible with NCL, we decided to create a Proof of Concept (PoC). The main objective of this evaluation was to answer the first research question (Q1). We have them rewrote the game AudioRio[2]. This cultural game was developed earlier without the use of LEGaL, and includes the second game pattern, *Follow-the-Path*, in which a player must follow a pre-established route for playing. We chose AudioRio because of its small size and the access to its multimedia documents. These characteristics allowed us to rewrite the game in a textual representation using LEGaL, without any visual modeling tool. The game consists of an audio-guided tour by Points of Interest along a river crossing the center of a metropolis. In the game, the player must visit these points and listen to an informative audio each time he enters areas previously defined by the game designer. Figure 5 illustrates the main screen of the game.

Each game mission is a geolocated map point. The user's avatar and the mission point have a radius of action. Thus, the user must be at a minimum distance from the point to be able to hear the audio. When entering this area (a circular region) the sound is automatically played. The information considered for the activation are latitude, longitude, and a radius. At AudioRio, there is no mandatory visitation order. This way, the user chooses where to start the

[2]AudioRio Web Site: www.excursaopajeu.com

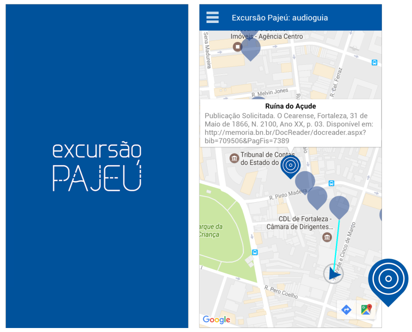


Figure 5: AudioRio for Android.

tour and which points to visit first. The game ends when the user visits all the predefined location points.

5.2.1 Game Modeling

This game consists of 18 missions[3]. Each mission is composed of two media, which is a text and a location file, and the relationship between them. The mobile game application uses the TTS (Text-to-Speech) feature to play audio from these text files defined in each mission.

We rewrote the game with LEGaL using the following elements:

1. **Context nodes** that are composed of media nodes;
2. **Media nodes** containing the location and the media to be executed;
3. **The input ports** that define by which nodes the game can start;
4. **The communication ports** of the nodes that define the game flow.

All media in the game, including those that specify the geolocated points, must be in a directory specified in the XML document (i.e., the game representation using LEGaL). Listing 7 shows an excerpt of the game, which models one of the AudioRio's missions.

```
<!-- game start port -->
<port id="pEntrance1" component="msWaterReservoir"
     interface="pWaterReservoir"/>

<context id="msWaterReservoir">
   <port id="pWaterReservoir" component="
       locWaterReservoir"/>
   <property name="mandatory" value="true"/>
   <property name="occurrence" value="unbounded"/>
   <property name="visibility" value="true"/>

   <media id="locWaterReservoir" type="application
       /gml+xml" scr="media/waterReservoir.gml"/>
   <media id="mdText1" type="text/plain" scr="
       media/text1.txt"/>

   <link xconnector="onEnteringStart">
      <bind role="onEntering" component="
          locWaterReservoir"/>
      <bind role="start" component="mdText1"/>
   </link>
</context>
```

Listing 7: AudioRio Mission described with LEGaL

[3]The complete modeling of the AudioRio game is available at https://tinyurl.com/khgeusz

In the modeling, the mission is identified by `msWaterReservoir` and the port that allows access to the mission content is `pWaterReservoir`. LEGaL allows game developers to define ports that indicate the first node to be executed (i.e., by which mission the player should start the game). For instance, in the AudioRio model, the port `pEntrance1` leads to the execution of the `msWaterReservoir` mission. In the game, the player can start on any mission and the missions do not have an order of accomplishment. Thus, for the modeling of AudioRio, we declared a start port for each mission node. It was not necessary to create links for the relationships between them. Figure 6 shows the LEGaL model of AudioRio in a structural view of its components.

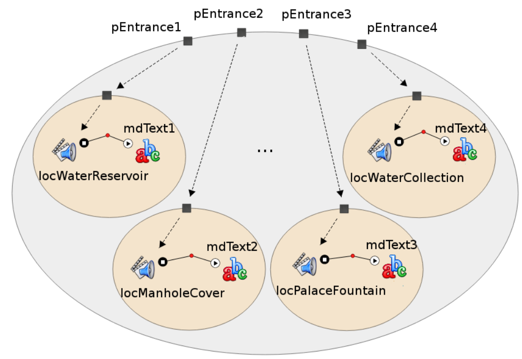


Figure 6: Structural view of AudioRio components.

5.2.2 AudioRio Generation

After the AudioRio modelling by using LEGaL, we used the LeGaL parser to regenerate the AudioRio game. Once the textual description of the game was specified, the parser took 1 599 milliseconds to generate the Hibernate instances and to transfer the media associated with the game. The game ran in exactly the same way as its previous version.

6 TRANSPOSITION

In the previous section, we showed that is possible to rewrite an ordinary LBG using LEGaL. In this case, a set of descriptive files (LEGaL documents) representing the model of the game were created. Therefore, any changes made to these documents affect the game. In this section, we benefit from the simplicity of LEGaL to apply a third-party transposition algorithm to AudioRio, and hence answer the second research question (Q2).

A great challenge for current LBGs is the difficulty to be available in any region users want to play. In this case, most LBGs using game patterns like *Search-and-Find* and *Follow-the-Path* are designed to be played in a specific area, requiring players to explore a fixed region, or move between predefined pl aces. This is a well know problem first addressed in [20] by introducing an algorithm called *Location Translation*. Recently, authors of [22] proposed an algorithm for balancing worldwide LBGs like Pokémon GO and Ingress. We noticed this algorithm is able to allow the transposition of LBGs, therefore an adaptation was performed during our tests.

The key advantage of using a transposition algorithm is to allow LBGs designed for specific locations to be ported and played elsewhere. However, algorithms like the ones presented by [20] and [22] are often designed to work with a specific tool or platform. Fortunately, LEGaL decouples the game model into different documents, thus allowing an easy integration with other algorithms and tools. Figure 7 shows how an original LBG is represented by specific documents, and illustrates the specific parts of the model

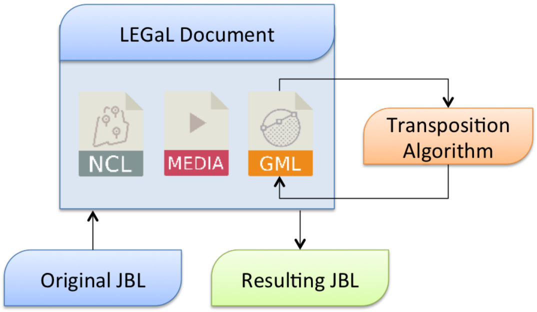
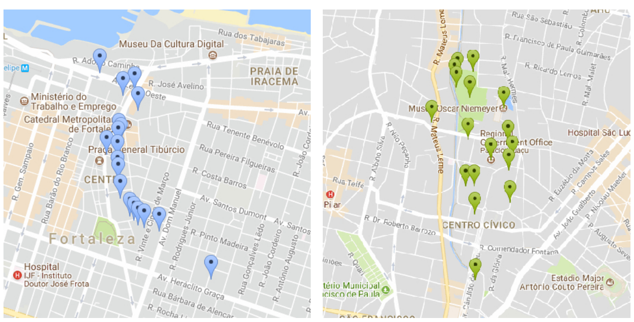


Figure 7: Transposition applied to the proposed game model.

a transposition algorithm has to interact with. In this case, the resulting LBG is given by a new ".gml" file containing the geographic coordinates of the new area, whereas both the ".ncl" and media files remain unchanged.

To proof our model can be used successfully with other algorithms and tools, we adapted the algorithm presented in [22] to take ".gml" files as input and output. Later, we tested this integration by applying this algorithm to transpose the game AudioRio to Curitiba. Consequently, the resulting LBG is composed of new places and can now be played there. Figure 8 shows the places that compose the original game (a) and the places that compose the resulting transposed LBG (b), thus showing how an LBG could benefit from LEGaL's integration with other tools to solve common LBG issues.



(a) Original LBG. (b) Resulting LBG.

Figure 8: Figure depicting POIs of the original LBG (a) and the resulting LBG (b) after transposition.

The transposition of an LBG consists in adapting one or more instances of the game to another geographical region. Moreover, the game balancing relates to the difficulty level presented by the games. In summary, to perform a balanced transposition of an LBG, it is not enough to select locations in a new geographic region where the game should take place. Additionally, it is necessary to define points that best represent the difficulty level present in the original game. Our transposition algorithm queries POIs (Point-of-Interest) from Google Places, and then selects points that deliver a game workflow with similar distance to be walked by the players. Using this strategy and a given initial geographic location, many well known places of Curitiba were selected by the algorithm to compose the transposed version of the game AudioRio, namely "Praça Rio Iguaçu", "Palácio Iguaçu", "Paranacidade", "Bosque João Paulo II", "Museu Oscar Niemeyer", "Palácio das Araucárias", etc.

Finally, this algorithm can now be used by any LBG described using LEGaL, thus ensuring the same tools and algorithms can be reused regardless of the platform or editing tools. This is a key advantage of adopting a universal modeling approach. In the future, we expect even more methods and applications will support LEGaL, thus fostering game designers to use this model during the development of LBGs.

7 Final Considerations

The absence of an explicit model capable of representing LBGs impairs its development and hinders the work of many professionals. In this paper, we presented the first effort to foster the modeling of LBGs. We proposed the LEGaL language (based on NCL) that allows a simple and precise representation of mechanics and rules of an LBG. LEGaL includes the media used in the mechanics of these games and supports temporal and spatial aspects. Our goal is to provide a canonical model to be supported by most authoring tools and algorithms used in the development of LBGs. With the structure and media of LBGs represented as NCL documents, in the future, game developers would take benefit from tools that edit, test, and check NCL documents.

The proof of concept presented in this work ensures that the first research question was successfully answered (Q1). However, new games need to be rewritten or developed from scratch to demonstrate the coverage that LEGaL offers regarding the variety of LBGs and game patterns that can be represented by it. Furthermore, the transposition algorithm used in Section 6 shows the potential of the proposed model to include new programs and methods capable of aiding and advancing the design and development of LBGs. Consequently, we conclude LEGaL can easily be used by external tools and applications given its simplicity and flexibility, thus answering to (Q2). We believe that LEGaL has the potential to enhance the development of a broad range of LBGs, and serve as a test bed for novel approaches and algorithms, such as the transposition method, designed to improve LBGs in general.

As future work, we want to support model checking mechanisms to avoid possible inconsistencies in the game modeling or the mechanics and rules created by the game's author. Another interesting point would be to contemplate spatial relations considering the proximity of points, permanence in regions, and the number of players. Finally, we want to propose a visual language for defining LBGs, and from this representation, implement the generation of NCL documents compatible with LEGaL.

References

[1] J. F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843, Nov. 1983.

[2] J. Arango-López, C. A. Collazos, F. L. Gutiérrez Vela, and L. F. Castillo. *A Systematic Review of Geolocated Pervasive Games: A Perspective from Game Development Methodologies, Software Metrics and Linked Open Data*, pages 335–346. Springer International Publishing, Cham, 2017.

[3] G. Area. Shadow cities wiki — fandom powered by wikia, 2016.

[4] C. Baier and J.-P. Katoen. *Principles of Model Checking (Representation and Mind Series)*. The MIT Press, Cambridge, MA, USA, 2008.

[5] N. S. Borenstein and N. Freed. Multipurpose internet mail extensions (mime) part one: Format of internet message bodies, 1996.

[6] S. Cox, A. Cuthbert, P. Daisey, J. Davidson, S. Johnson, E. Keighan, R. Lake, M. Mabrouk, S. Margoulies, R. Martell, et al. Opengis® geography markup language (gml) implementation specification, version, 2002.

[7] A. de Souza e Silva and D. Sutko. *Digital Cityscapes: Merging Digital and Urban Playspaces*, volume 57 of *Digital Cityscapes: Merging Digital and Urban Playspaces*. Peter Lang, New York, NY, 2009.

[8] E. F. do Prado and D. Lucrédio. A flexible model-driven game development approach. In *Components, Architectures and Reuse Software (SBCARS), 2015 IX Brazilian Symposium on*, pages 130–139. IEEE, 2015.

[9] J. A. dos Santos, C. Braga, D. C. Muchaluat-Saade, C. Roisin, and N. Layaïda. Spatio-temporal validation of multimedia documents. In *Proceedings of the 2015 ACM Symposium on Document Engineering*, DocEng '15, pages 133–142, New York, NY, USA, 2015. ACM.

[10] Games4All. Spectrek, 2016.

[11] V. Guana, E. Stroulia, and V. Nguyen. Building a game engine: A tale of modern model-driven engineering. In *Proceedings of the Fourth International Workshop on Games and Software Engineering*, pages 15–21. IEEE Press, 2015.

[12] H. Guo, H. Trætteberg, A. I. Wang, and S. Gao. Pergo: an ontology towards model driven pervasive game development. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 651–654. Springer, 2014.

[13] G. Hong. *Concepts and Modelling Techniques for Pervasive and Social Games*. PhD thesis, Norwegian University of Science and Technology, 2015.

[14] V. Kasapakis, D. Gavalas, and N. Bubaris. Addressing openness and portability in outdoor pervasive role-playing games. In *2013 Third International Conference on Communications and Information Technology (ICCIT)*, pages 93–97, Beirut, Lebanon, June 2013. IEEE.

[15] N. Labs. Ingress, 2016.

[16] N. Labs. Pokémon go, 2016.

[17] L. Lehmann. *Location-Based Mobile Games*. GRIN Verlag, Munich, Germany, 2012.

[18] M. Lochrie, K. C. Pucihar, A. Gradinar, and P. Coulton. Designing seamless mobile augmented reality location based game interfaces. In *Proceedings of International Conference on Advances in Mobile Computing & Multimedia*, MoMM '13, pages 412:412–412:415, New York, NY, USA, 2013. ACM.

[19] C. S. Longstreet and K. Cooper. A meta-model for developing simulation games in higher education and professional development training. In *Computer Games (CGAMES), 2012 17th International Conference on*, pages 39–44. IEEE, 2012.

[20] A. Macvean, S. Hajarnis, B. Headrick, A. Ferguson, C. Barve, D. Karnik, and M. O. Riedl. Wequest: scalable alternate reality games through end-user content authoring. In *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*, page 22. ACM, 2011.

[21] C. Magerkurth, A. D. Cheok, R. L. Mandryk, and T. Nilsen. Pervasive games: bringing computer entertainment back to the real world. *Computers in Entertainment (CIE)*, 3(3):4–4, 2005.

[22] L. F. Maia, W. Viana, and F. Trinta. Using monte carlo tree search and google maps to improve game balancing in location-based games (to appear). In *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8, Aug 2017.

[23] A. Matallaoui, P. Herzig, and R. Zarnekow. Model-driven serious game development integration of the gamification modeling language gaml with unity. In *System Sciences (HICSS), 2015 48th Hawaii International Conference on*, pages 643–651. IEEE, 2015.

[24] C. Nolêto, M. Lima, L. F. Maia, W. Viana, and F. Trinta. An authoring tool for location-based mobile games with augmented reality features. In *Computer Games and Digital Entertainment (SBGames), 2015 14th Brazilian Symposium on*, pages 99–108, Teresina, Brazil, 2015. IEEE, SBGames.

[25] K. O'Hara. Understanding geocaching practices and motivations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1177–1186, New York, NY, USA, 2008. ACM.

[26] L. Olive Seraph. Battlesuit runner fitness, 2016.

[27] A. Patro, S. Rayanchu, M. Griepentrog, Y. Ma, and S. Banerjee. The anatomy of a large mobile massively multiplayer online game. *ACM SIGCOMM Computer Communication Review*, 42(4):479–484, 2012.

[28] D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. *KR*, 92:165–176, 1992.

[29] E. M. Reyno and J. Á. C. Cubel. Model driven game development: 2d platform game prototyping. In V. J. Botti, A. Barella, and C. Carrascosa, editors, *GAMEON*, pages 5–7, Vaencia,Spain, 2008. EUROSIS.

[30] I. Rubino, C. Barberis, J. Xhembulla, and G. Malnati. Integrating a location-based mobile game in the museum visit: Evaluating visitors' behaviour and learning. *Journal on Computing and Cultural Heritage (JOCCH)*, 8(3):15, 2015.

[31] K. Sánchez, K. Garcés, and R. Casallas. A dsl for rapid prototyping of cross-platform tower defense games. In *Computing Colombian Conference (10CCC), 2015 10th*, pages 93–99. IEEE, 2015.

[32] L. F. M. Silva, C. Noleto, M. Lima, C. Ferreira, C. Marinho, W. Viana, and F. Trinta. LAGARTO: A location based games authoring tool enhanced with augmented reality features. *Entertainment Computing*, 22:3–13, 2017.

[33] L. F. G. Soares and R. F. Rodrigues. Nested context model 3.0. part 1–ncm core. Technical report, Pontífica Universidade Católica do Rio de Janeiro, RJ, Brasil, May 2005.

[34] P. Spiesberger, F. Jungwirth, C. Wöss, S. Bachl, J. Harms, and T. Grechenig. Woody: A location-based smartphone game to increase children's outdoor activities in urban environments. In *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*, MUM '15, pages 368–372, New York, NY, USA, 2015. ACM.

[35] A. N. Start, S to. The walk game, 2016.

[36] J. R. M. Viana, N. P. Viana, F. A. M. Trinta, and W. V. d. Carvalho. A systematic review on software engineering in pervasive games development. In *2014 Brazilian Symposium on Computer Games and Digital Entertainment*, pages 51–60, Nov 2014.

[37] R. Wetzel, L. Blum, and L. Oppermann. 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*, FDG '12, pages 238–241, New York, NY, USA, 2012. ACM.

[38] E. Witkowski. Running from zombies. In *Proceedings of The 9th Australasian Conference on Interactive Entertainment: Matters of Life and Death*, page 1. ACM, 2013.

[39] C. Workline. Tourality – the ultimate outdoor gps game, 2016.