

A Pattern-based Framework for Designing Location-based Games

Thomas Heinz
University of Bamberg
An der Weberei 5, Bamberg, Germany
thomas.heinz@uni-bamberg.de

Christoph Schlieder
University of Bamberg
An der Weberei 5, Bamberg, Germany
christoph.schlieder@uni-bamberg.de

Abstract

This research identifies a core set of location-based operators needed for building complex spatio-temporal game mechanics. An analysis of popular design frameworks shows that they provide limited support for fundamental spatial game patterns. Patterns involving the concept of locality are supported only in part, while those involving the notion of proximity are not supported at all. We describe a placed-based event system grounded in established geographic core concepts that permits a non-expert designer to create games using any of the fundamental spatial game patterns. Finally, we present and evaluate an implementation of a design framework for location-based games based on the place-based event system.

Keywords: Location-based-games, spatio-temporal events, framework, agent-based-simulation

1 Introduction

The design process of a location-based game typically requires a considerable amount of expertise in game mechanics as well as spatial analysis skills (El-Nasr et al. 2013). However, for educational games it is desirable to let educators, who generally lack this kind of expertise, create their own location-based games. There are several reasons for involving educators. They possess the relevant local geographic knowledge, they are acquainted with the players, who are their students, and, most importantly, they are in a position to integrate the game into a comprehensive learning experience (Mathews and Holden 2018). Location-based game design frameworks assist non-experts to create games by providing, for instance, visual editors, which permit to georeference resources. Although design frameworks have been successful at attracting educators to location-based games, they were much less successful in supporting the creation of a variety of diverse games. Many of the games implement a simple spatial search pattern, not very different from Geocaching (Schlieder et al. 2018).

In our research, we ask whether the missing diversity of games can be tied to limitations of the frameworks. The paper makes the following main contributions: (1) we analyse three major design frameworks and identify fundamental spatial game patterns which are not supported, (2) we describe a placed-based event system grounded in established geographic core concepts that permits a non-expert designer to create games using any of the fundamental spatial patterns, (3) we present a design framework based upon the event system, and (4) we finally evaluate the framework by showing that a test game, which makes use of the locality and proximity patterns, can be modelled by the framework.

2 Related work

Spatial game patterns

Game patterns are established tools in video- and board game design. They consist in textual “descriptions of recurring interaction elements relevant to gameplay” (Davidson et al. 2004). Designer use game patterns, to discuss, document and analyze existing game mechanics (Kreimeier 2002). Patterns are also used as building blocks for creating new and interesting game mechanics (Bjork and Holopainen 2004).

Two pattern inventories are of special interest for the study of mobile, and more specifically, location-based games. Davidsson et al. (2004) list 74 different patterns, focusing on games played on mobile devices, while Sintoris (2015) covers 41 patterns that relate to location-based games exclusively. Both pattern inventories were created by a systematic analysis of existing games in workshops with expert game designer. The patterns cover different aspects of game play, ranging from spatial event chains to complex forms of social user interaction.

Ahlqvist and Schlieder (2018) have studied the two pattern inventories to identify those patterns that have a clear spatial content (e.g. obtaining a benefit for entering a spatial region) and those that do not (e.g. letting players compete by providing a high-score list). Interestingly, only 30% of the patterns were found to be connected to at least one of the geographic core concepts described by Kuhn (2012) and Janelle and Goodchild (2011), while the others cover non-spatial aspects of the games. In other words, location-based games seem to share a large number of patterns with other types of games that are not location-based. Among the spatial patterns, Ahlqvist and Schlieder (2018) identify two groups of fundamental spatial game patterns, namely locality and proximity patterns, both of which can be defined in purely spatial terms without referring to social interactions. These

patterns are considered fundamental in the sense that any location-based game engine should support at least one of each group.

A frequently found example of a *locality pattern* has been called Strategic Locations ("locations where players receive special benefits") by Davidsson et al. (2004). Sintoris (2015) refers to this pattern as Spatial Structure ("locations ... may award points") by. The pattern assigns a reward to a player who is located in a specific geographic region. For locality patterns being located in a region is a binary property. Location-based games, however, also rely on continuous spatial distinctions. These are covered by the *proximity patterns*. An example is the pattern Player-Location Proximity ("The distance between the player and a location can trigger events") from the inventory of Davidsson et al. (2004).

Location-based game design frameworks

In addition to the difficulties of video game development, the designers of location-based games face new challenges like determining the player positioning in the real-world and the need to support a multitude of spatial operations (Crooks et al. 2008). In video game development there is a strong trend towards game engines, which already provide most of the required core functionalities (*Unity*, *Unreal Engine*) (Thorns 2011). A similar development can also be observed in location-based games. Especially, three location-based frameworks have become popular among educators interested in designing game-based learning experiences.

Actionbound. An "app for playing digitally interactive scavenger hunts to lead the learner on a path of discovery"¹. Leans towards games, in which search patterns dominate such as the classic Geocaching. (Rittel 2017)

Aris Editor. An "experimental platform to expand what is possible"². Promoted game genres and activities that are listed on their informational website include: Interactive Stories, Scavenger Hunts, Tours, Data Collection. (Holden 2015)

TaleBlazer. An "augmented reality (AR) software platform" that "allows users to play and make their own location-based mobile games."³. The framework ties to support for a variety of different genres ranging from roleplaying to arcade games⁴. (Medlock-Walton 2012)

While the frameworks are popular among non-designer who wish to create a location-based game (Perez Colado et al. 2017), they suffer from limited expressiveness as we will show in section 3.

3 A comparison of frameworks

All three design frameworks mentioned in section 2, produce games played on a smartphones or tablets. Game relevant data is either downloaded to the device before the play session or transmitted in real time to a game server. With the frameworks, non-expert designers create games by defining

events, which relate to one or more game entities. The frameworks offer web-based editors that do not require programming knowledge but allow setting up game events via visual, block-based programming languages.

We studied the expressiveness of the frameworks by analyzing to what extent they support their users in creating the fundamental locality and proximity patterns. In a first step, we determined the game entities supported (Table 1). All three frameworks permit to represent players, virtual items (resources) as well as computer controlled characters (agents) as point features. Only places are represented as two-dimensional objects. However, the geometries of places are restricted to circles, that is, a point location with a capture radius. In other words, the frameworks have obvious limitations in representing geo-data.

Table 1: Framework entity support

Entities	Actionbound	Aris	Taleblazer
Players	1D	1D	1D
Places	2D*	2D*	2D*
Resources	-	1D	1D
Agents	-	-	1D

All three frameworks provide a block-based programming language that assists the designer in defining game events. Table 2 shows the types of events that are supported. The On-Enter event, for instance, occurs when a player enters a geographic region. As this event is supported by all frameworks, designers can easily create the locality patterns Strategic Locations and Spatial Structure described in the previous section.

Table 2: Framework event support

Events	Actionbound	Aris	Taleblazer
Game Start/End	✓	✓	✓
On Enter	✓	✓	✓
Scheduled Events		✓	✓
User Input		✓	✓
Bluetooth Beacons		✓	✓
Entity Attribute Changes		✓	✓

Not all locality patterns are triggered by On-Enter events. A more complex pattern is Co-Locality, which requires the simultaneous presence of players in a bounded geographic region. None of the framework provides support for this pattern. Even in combination with the other events it is not possible to model a mandatory dwelling time, or a mechanic related to entity leaving a specific area.

¹ <https://en.actionbound.com/>

² <https://fielddaylab.org/make/aris/>

³ <http://www.taleblazer.org/about#make>

⁴ <http://www.taleblazer.org/files/docs/>

TaleBlazerGameWorksheet.pdf

Table 3: Framework pattern support

Game Pattern	Actionbound	Aris	Taleblazer
Strategic locations (Locality)	(✓)	(✓)	(✓)
Spatial structure (Locality)	(✓)	(✓)	(✓)
Co-locality (Locality)	-	-	-
Player-Location (Proximity)	-	-	-
Artifact-Location (Proximity)	-	-	-
Player-Player (Proximity)	-	-	-
Artifact-Artifact (Proximity)	-	-	-

As can be seen from Table 2, there is no support for distance-based events and thus no possibility to realize any of the proximity patterns. Table 3 summarizes the support for the patterns.

4 A pattern-based framework

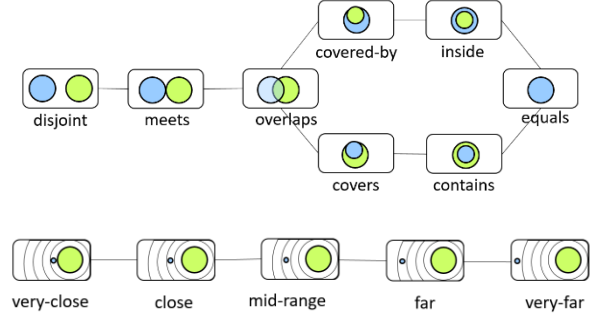
Similar to the editors provided by other location-based framework, in our configuration tool, users may create game mechanics by setting up events between game entities. With non-experts in mind, the underlying event model is based on qualitative spatio-temporal information, which abstracts from raw sensor data. Game mechanics are created as a sequence of states, which must occur in a specific temporal order to trigger a designated outcome.

The spatial event model is based on a few central concepts, which we briefly define in the following. A *game state* (E, R) consists of a set E of game entities (players, places, resources, agents) and a set R of qualitative spatial relations on E . To keep the event model conceptually simple, we include only two types of binary spatial relations. Formally, both are relation algebras. For the technical details of qualitative spatial reasoning formalisms we refer the reader to the textbook of Ligozat (2012). The first relation algebra is a system of topological relations widely used in geo-information processing: RCC-8 (Randell et al. 1992). See Fig. 1 for an illustration of these relations. In the event model, the topological relations serve for modelling locality game patterns. The second system of relations is used to describe proximity patterns. We use the system of qualitative distance relations originally defined by Hernández et al. (1995). These relations are also shown in Fig. 1.

Note that changes of qualitative spatial relations occur in a systematic way described by conceptual neighborhoods (Ligozat, 2012). For instance, if at a first point in time a disjoint relation holds between two spatial regions, a meet relation has to occur at a second point in time before an overlap relation can occur at a third point in time (Fig.1). In this sense, we define a *spatial event* (s, t) as a pair of game states s and t , where t is a temporal successor of s and the

spatial relations holding in s are conceptual neighbors of the spatial relations holding in t .

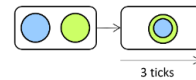
Figure 1: Conceptual neighbourhoods of topological relations and qualitative distance relations



An event described in terms of only topological relations is called a *locality event*. Similarly, a *proximity event* can be defined by only using distance relations.

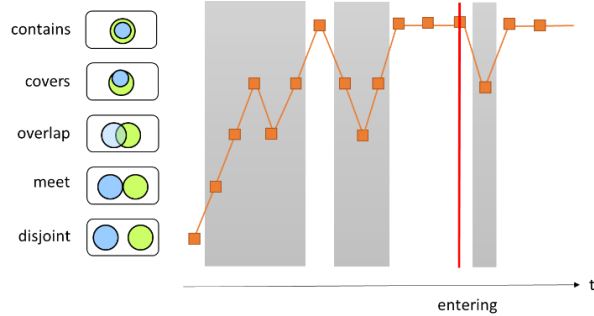
This spatial event model, though conceptually simple, is considerably more expressive than what is provided by the game design frameworks discussed in section 3. With the editors of those frameworks, a user can only define a single type of locality event: On-Enter. Note also that the frameworks use much coarser topological relations distinguishing only between disjoint and inside. In our event model, a single On-Enter event in a framework such as TaleBlazer corresponds to a sequence of events involving the disjoint, meets, overlaps, covers, and inside relation).

In our system the user may specify an arbitrary sequence of states that have to occur before the event is triggered. Additionally, a state can be specified with a duration, for which it has to persist. Time is measured by game-specific ticks in this case. Figure 2 gives an example of a locality events that can be defined within the event model. The event describes the process, by which a player captures a place: the player has to enter the place and stay within the place for at least 3 ticks of time.

 Figure 2: Example of a locality pattern: *Capture*


The framework continuously monitors the positioning data from the players to detect events. Figure 3 shows the timeline of a player entity in interaction with a place entity. The Capture pattern is triggered (red line) when the beginning and ending conditions of the pattern are met and the duration constraint is satisfied. While the Capture pattern is expressed in the proposed event model, it exceeds the expressive capabilities of the frameworks discussed in section 3.

Figure 3: Capture event - state transitions



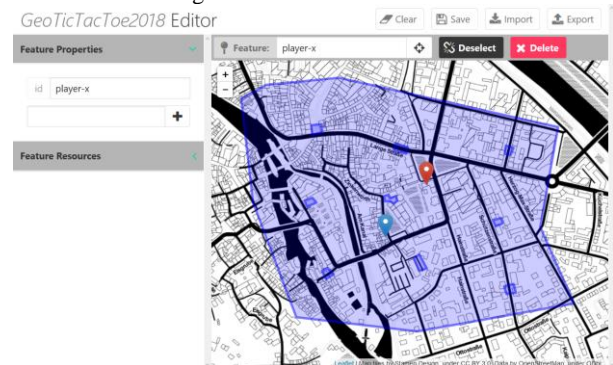
We based the implementation of the framework on a client-server architecture, not very different from the approach taken by the frameworks *Actionbound*, *Aris* and *TaleBlazer*. Game designers can develop games with a visual web-based editor. Visual, block-based programming languages with a drag and drop interface have proven to be very successful in this kind of approach (*ActionBound*, *Aris*, *TaleBlazer*), but also in developer frameworks for non-location-based games (*Unity*, *GameMaker*) (El-Nasr & Smith 2006). Upon completion, game configurations are stored into a central database and are immediately playable on any smartphone with state of the art positioning technology. The focus is on competitive or cooperative multiplayer games, but single player games are supported as well. It supports all of the entities listed in Table 1, but additionally provides support for arbitrary two-dimensional geometries. The player model that is utilized in this framework was previously also used to support agent-based simulation (Heinz & Schlieder 2015). To support co-designers, agent-based simulation is also included in the framework and described in more detail in section 5.

5 Evaluation of the event model

The purpose of the evaluation consists in providing a proof of concept for the event model. We want to demonstrate that it is possible to handle a realistic game design task with the framework that implements the event model. The game design task consists in creating a designed a variant of the location-based game *GeoTicTacToe* (Schlieder et al 2006). This variant, named *GeoTicTacToe2018* to distinguish it from the original. The underlying game mechanic is based on the well-known pen and paper game *TicTacToe*, using multiple locality and proximity based patterns (see Table 3). This variant has a more complex spatio-temporal game flow than the original game and therefore provides a challenging test case. The game board is split into nine separate geographic locations (Fig. 5). In order to place an X or O token on this geographic game board, the player has to visit the place as specified by the Capture pattern. *GeoTicTacToe2018* is played as a real-time game without turn-taking. However, the game does not reveal the exact game board locations to the players. Like in the children’s game "Hot & Cold", players are only given feed-back (very far, far, close, and very close) on their distance to a location (cp. *Player-Location Proximity*). Once a player has entered a cache, she can capture it by staying a mandatory dwell time but only whilst her opponent is not trying to do the same (cp. *Strategic Locations*, *Co-Locality*).

Figure 4 portrays a game board constructed in the frameworks editor.

Figure 4: Game field creation



At the current state of development, there is no user interface for specifying the game end and winning conditions. Therefore, creating a game still requires writing some code. Once the interfaces have been scripted, the game can be relocated to any location by placing game entities on a map and setting up the place-based events.

The framework features an agent-based modelling toolkit that can be used to test the balancing of created game fields by simulation. This component is built on top of the *MESA*⁵ framework but was significantly extended to support geographical spaces. Like the winning conditions, game specific agent behavior has to be scripted in the Python programming library. A variety of features is included, like sophisticated pedestrian navigation. Individual simulation runs can be observed via a web-based interface (See figure 5). Running simulations permits designers to obtain valuable insights into different type of games that emerge from different spatial layouts without having to rely on user-tests.

Figure 5: GeoTicTacToe2018 Simulation



The game was already put into practice with a prototypical game client. Simulation agents with different basic strategies

⁵ <https://github.com/projectmesa/mesa>

were implemented and provide users with the opportunity to test and balance their own game fields, as already shown in figure 4.

6 Conclusion and future work

In this paper, we analyzed design frameworks for location-based games finding that they only support the creation of very simple spatial game patterns. Support is limited to locality patterns and, more specifically, to the On-Enter pattern. To assist designers who want to create more complex patterns we described a spatial event system. This event system permits to model locality events as well as proximity events, two event types that were shown to be fundamental to location-based games. We implemented the event system in a design framework and provided a proof of concept by addressing a realistic game design task with the framework.

While our framework provides more spatial modeling capabilities than the design frameworks discussed in section 3, its expressiveness has limitations. The event system focuses on games with clear spatial and temporal boundaries. In its current form, the framework does not support pervasive games such as Pokémon Go or Ingress, with an ongoing gameplay. Designers who wish to create such games have to extend the framework via programming, for instance, by adding gameplay rules and winning conditions. Although the framework improves considerably upon the state of the art, it still only supports a subset of location-based games.

Future work is going to concentrate on improving the event system beyond the support for fundamental spatio-temporal gameplay patterns. A natural next step consists in covering as many of the known location-based game design patterns as possible. Other planned improvements regard the framework's user interface, which could give users more feedback during the game design process.

7 References

- Ahlqvist, O. and Schlieder, C., 2018. Geogames and Geoplay: Game-based Approaches to the Analysis of Geo-information. Springer.
- Bjork, S. and Holopainen, J., 2004. Patterns in game design (game development series).
- Crooks, A., Castle, C. and Batty, M., 2008. Key challenges in agent-based modelling for geo-spatial simulation. *Computers, Environment and Urban Systems*, 32(6), pp.417-430
- Davidsson, O., Peitz, J. and Björk, S., 2004. Game design patterns for mobile games. Project report to Nokia Research Center, Finland.
- El-Nasr, M.S. and Smith, B.K., 2006. Learning through game modding. *Computers in Entertainment (CIE)*, 4(1), p.7.
- El-Nasr, M. S., Drachen, A., & Canossa, A., 2013. Game analytics: Maximizing the value of player data. Springer Science & Business Media.
- Heinz, T., Schlieder, C., 2015. An Agent-Based Simulation Framework for Location-Based Games.
- Hernández, D., Clementini, E. and Di Felice, P., 1995, September. Qualitative distances. In *International Conference on Spatial Information Theory* (pp. 45-57). Springer, Berlin, Heidelberg.
- Holden, C., 2015. Mobile media learning: Innovation and Inspiration. ETC Press, Pittsburgh, PA, USA.
- Janelle, D.G. and Goodchild, M.F., 2011. Concepts, principles, tools, and challenges in spatially integrated social science. The SAGE handbook of GIS and society. Thousand Oaks, CA: SAGE, pp.27-45.
- Kreimeier, B., 2002. The case for game design patterns.
- Kuhn, W., 2012. Core concepts of spatial information for transdisciplinary research. *International Journal of Geographical Information Science*, 26(12), pp.2267-2276.
- Ligozat, G. (2012). Qualitative Spatial and Temporal Reasoning, Wiley: Hoboken, NJ.
- Mathews, J. and Holden, C., 2018. The Design and Play of Geogames as Place-Based Education. In *Geogames and Geoplay* (pp. 161-176). Springer, Cham.
- Medlock-Walton, M.P., 2012. TaleBlazer: a platform for creating multiplayer location based games (Doctoral dissertation, Massachusetts Institute of Technology).
- Rittel, J., 2017. Actionbound at School: An Introduction to Library Use with Apps & Co. *School Libraries Worldwide*, 23(2), p.29.
- Randell, D. A., Cui, Z., & Cohn, A. G. 1992. A spatial logic based on regions and connection. KR-92, Conference on Knowledge Representation, 165-176.
- Perez Colado, V.M., Perez Colado, I.J., Martínez Ortiz, I., Freire Moran, M. and Fernandez Manjon, B., 2017. Simplifying location-based serious game authoring.
- Schlieder, C., Kiefer, P. and Matyas, S., 2006. Geogames: Designing location-based games from classic board games. *IEEE Intelligent Systems*, 21(5), pp.40-46.
- Schlieder, C., Kremer, D. and Heinz, T., 2018. Teaching Geogame Design: Game Relocation as a Spatial Analysis Task. In *Geogames and Geoplay* (pp. 111-130). Springer, Cham.
- Sintoris, C., 2015. Extracting game design patterns from game design workshops. *International Journal of Intelligent Engineering Informatics*, 3(2-3), pp.166-185.
- Thorn, A., 2011. Game engine design and implementation. Jones & Bartlett Publishers.