Geogames: Designing Location-Based Games from Classic Board Games

Christoph Schlieder, Peter Kiefer, and Sebastian Matyas, *University of Bamberg*

Location-based games introduce an element missing in interactive console games: the physical effort of sports. Using classic board games as templates, challenging location-based games add strategic reasoning to sportive locomotion.

The traditional image of interactive entertainment—games that reduce players' physical involvement to moving a joy stick—is obsolete. Games researchers and designers are already integrating complex movement into games; Nintendo's Donkey Konga, Sony's EyeToy, and Konami's Dancing Stage are good examples. This kind of game might use simple sensor contact, such as hitting a drum or stepping on a dancing mat, or more intricate forms of movement captured by video or infrared, such as waving gestures. They involve moving parts of the body but only limited displacement of the body as a whole.

Cognitive psychology's classification of spatial scales helps clarify the issue of player movement. Daniel Montello[1] distinguishes

- *figural space,* which is smaller than the body and accessible to haptic manipulation or close visual inspection;
- *vista space*, which is as large as or larger than the body but can be apprehended visually from a single place without locomotion; and
- *environmental space*, which is larger than the body and can't be experienced without considerable locomotion.

Location-based games involve body movements beyond figural space—that is, beyond the space of computer screens and small 3D objects. They focus on locomotion in vista space, typically a single room or sports field, or in environmental space, such as a neighborhood or city. Several location-based games have been designed for environmental space using localization technologies such as GPS. Some are adaptations of popular computer games, and others are rather straightforward chase games (see the sidebar "Location-Based Games").

A third type of game—what we call *challenging location-based games*—combines the intellectual and strategic appeal of classic board games with the real-time locomotion and physical involvement of location-based games. Although the idea of directly mapping classic board games onto the real world isn't new,[2] we know of no general framework for designing such games. Furthermore, a straightforward approach to the spatial mapping of classic board games doesn't work. Instead, we developed the Geogames framework, which uses search techniques to identify a single temporal parameter for balancing sportive versus strategic elements.

Spatial versions of board games

Not all types of spatial mapping are interesting. Players consider a location-based game challenging if it addresses both their reasoning skills and their motor skills. Neither a chess tournament with outdoor chess pieces nor a 100-meter sprint would constitute a challenging location-based game in this sense.

Board games come in many variants, not all of which are intellectually as demanding as chess or Go. Many of them involve two players, are deterministic (no random element such as dice), and provide full information about the game's state to each player (no hidden elements such as cards in the opponent's hand).

1541-1672/06/$20.00 © 2006 IEEE
Published by the IEEE Computer Society

Location-Based Games

A *mobile game* is simply a game that runs on a mobile device—for instance, Tetris on a smart phone.

Some mobile games use environmental space as playing ground. In these *location-based games*, the players' positions—and sometimes locomotion—constitute key game elements. Some sort of position-tracking technology, typically GPS, wireless local area network, or Cell-ID in general packet radio service or universal mobile telecommunication system networks, is required to play the game. An early and rather simple example of a location-based game is *geocaching*, in which players use geographic coordinates to find hidden objects using a handheld GPS receiver as the mobile device (www.geocaching.com). Games with more advanced computational support include Can You See Me Now![1] where players on the streets chase online players, and Pirates![2] where players interact with other players through their avatars in the virtual world.

Augmented-reality games integrate the virtual world of the game with environmental space at a perceptual level. A well-known example is Human Pacman,[3] where human players role-play the main game characters.

Pervasive or *ubiquitous games* such as the Songs of North[4] introduce the concept of an omnipresent gaming world that lets players connect with the game anywhere and anytime.

Through all genres, from location-based to ubiquitous games, traditional chase games or adaptations from well-known desktop computer games dominate. Although novel game concepts exist, for instance Uncle Roy All Around You,[5] none of these concepts seems to focus on finding a challenging balance of sportive and reasoning elements. This is exactly what geogames are about.

References

1. M. Flintham et al., "Where On-Line Meets On-the-Streets: Experiences with Mobile Mixed Reality Games," *Proc. Conf. Human Factors in Computing Systems* (CHI 03), ACM Press, 2003, pp. 569–576.
2. S. Björk et al., "Pirates!–Using the Physical World as a Game Board," *Proc. IFIP TC.13 Conf. Human-Computer Interaction* (Interact 01), IFIP Technical Committee on Human-Computer Interaction, 2001; www.viktoria.se/fal/publications/play/2001/pirates.interact.pdf.
3. D. Choek et al., "Human Pacman: A Mobile, Wide-Area Entertainment System Based on Physical, Social, and Ubiquitous Computing," *Personal and Ubiquitous Computing*, vol. 8, no. 2, 2004, pp. 71–81.
4. P. Lankoski et al., "A Case Study in Pervasive Game Design: The Songs of North," *Proc. 3rd Nordic Conf. Computer-Human Interaction* (NordiCHI 04), ACM Press, 2004, pp. 413–416.
5. M. Flintham et al., "Uncle Roy All Around You: Mixing Games and Theatre on the City Streets," *Proc. Level Up: 1st Int'l Conf. Digital Games Research Assoc.* (DIGRA 03), 2003.

We concentrate on this rich class of games as a source of inspiration for the strategic elements in our game design and use the term *board game* in this narrow sense.

Our running example is a structurally simple, well-known board game: tic-tac-toe. Two players move alternately, placing tokens (the first player to move uses "X," the second "O") on a 3×3-square game board. The player who first places three tokens in a row, a column, or one of the two diagonals wins the game.

Physically, board games are played in figural space. To obtain a location-based version of the game, we must map the game onto vista or environmental space such that each move requires the player's locomotion. This introduces time as a new dimension in the game. We call the result a *spatial* version of the board game and refer to the process of producing it *spatialization*.

The straightforward approach to spatialization consists of assigning coordinates, or *geographic footprints*, to each board position. A geographic footprint can be a point, a polyline, or an area. For tic-tac-toe, we assign a coordinate to each of the nine board positions (and to simplify matters, we'll use points). In the spatial version of a board game, players must physically move to a board position to place a token (an "X" or "O" in tic-tac-toe). The time it takes to complete a move therefore depends on the distance between the board positions in vista or environmental space.

The spatial version of a classic board game is particularly interesting if it satisfies the following requirements:

- *Balance of sportive and reasoning elements*. A marginal difference in speed shouldn't lead to a winning strategy (as it does in a 100-meter sprint). Also, the game design should handle the problem that we don't know in advance who is faster and who is slower (although in practice there's always a difference in speed between two players). In setting up tournaments with multiple unknown players, it's impossible to counterbalance speed differences by giving the slower player a head start.
- *Unconstrained choice of geographic footprints*. The game design should constrain spatialization as little as possible. Ideally, the game designer should be able to map board positions onto any set of geographic locations—for example, to cultural heritage sites or places of outstanding natural beauty—to increase players' interest in the game.

Because of the designer's freedom to assign geographic footprints, the spatial version of a board game generally won't preserve the distance relationships that hold on the game board in figural space. For instance, tic-tac-toe's board positions don't need to be arranged in a regular 3×3 square, as in figure 1a; the arrangement in figure 1b is just as possible. The latter set of footprints isn't just a rotated version of the previous set: the three top-most positions form a row, as do the three bottom-most positions and the three middle positions.

The synchronization problem

Unfortunately, a straightforward approach to spatialization (one that ignores the synchronization problem) results in unchallenging location-based games. The problem is that board games' logical appeal is linked to the complexity of a game's state space, which changes drastically if the two players don't move in alternation.

Consider the game illustrated in figure 1a. The players' spatial trajectories reveal that the X-player moves faster than the O-player. Obviously, in this case the X-player has a simple winning strategy: take the shortest path from top to bottom through the closest column and win the game after placing the third X-token. This spatial version of tic-tac-toe amounts to a race between the two players, which lacks any elements of strategic rea-

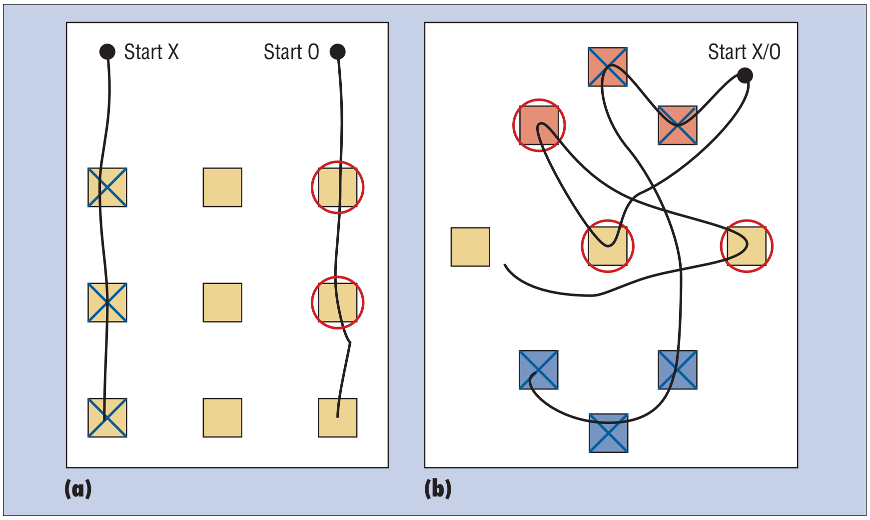


Figure 1. Two spatial versions of tic-tac-toe: (a) with an equidistant mapping, and (b) with a distorted mapping.

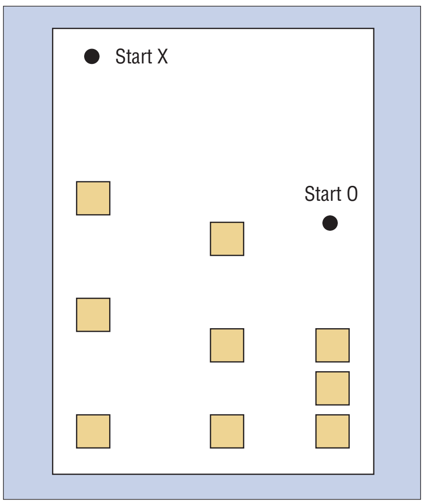


Figure 2. Looking for a spatial solution to the synchronization problem.

soning. So, we can't consider this a challenging location-based game. This kind of race will occur even for a marginal difference in speed. On the other hand, in a game with strictly alternating moves (for example, outdoor chess), a player's speed has no impact at all—also an unchallenging board game.

Figure 1b illustrates a challenging location-based game. Here, moves don't always alternate. To put it differently, the task of designing a challenging location-based game from a board game involves limiting the occurrence of multiple moves by the faster player. We call this the *synchronization* problem because it somewhat levels differences between the two players' speed, resulting in a deliberately non-perfect synchrony of moves.

The synchronization problem isn't a problem of the specific choice of geographic footprints and the starting position, as in figure 1a. It appears in figure 1b as an effect of speed difference between players in much the same way as in figure 1a. Nevertheless, we might be tempted to look for a spatial solution to the synchronization problem, one that somehow handicaps the faster player. We could try to distort the tic-tac-toe game board so that the faster player has to cover larger distances, as in figure 2. Such an approach, however, violates both design requirements listed earlier, with the accompanying consequences. On the one hand, it forces game designers to determine which player moves fastest; on the other hand, it prevents them from using geographic points of interest in the spatialization.

Fortunately, the synchronization problem has a surprisingly simple temporal solution: after reaching a board position and placing a token, players must spend a certain synchronization time at that position before moving on.

Typically, the game designer will integrate synchronization time into the game by having the player perform some time-consuming tasks, such as solving a puzzle or searching for an item. We'll see later that the length of the synchronization time interval constitutes the parameter that determines whether the spatial version of a board game becomes a challenging location-based game or deteriorates into either a race-style game (in which the sportive element dominates) or a classic board game (in which the reasoning element dominates).

The Geogames framework

Although a spatial version of tic-tac-toe is somewhat interesting on its own, designers need a general method for reusing board games as templates for location-based games. Our descriptive framework for location-based games should handle other spatial versions of board games or other possible games, such as CityPoker.[3] Typically in such games, a fixed number of players (often but not always two) move between a fixed number of board positions, called locations, taking up and putting down resources when they reach a new position. A resource is anything that players can transport and deposit in locations, such as an X-token in tic-tac-toe or a playing card in CityPoker. The framework we created, Geogames, defines the state of a game by the players' locations and by the distribution of resources among players and locations. The framework describes actions as transitions between states, including the combined effect of moving from one location to another and of taking up or putting down resources there.

The following definitions describe the Geogames framework, in which the designer can express a temporal solution to the synchronization problem.

Definition: Let P denote a set of players, L a set of locations, and R a set of resources. A *state* $s = (location, resources)$ is a tuple of two mappings, *location*: $P \to L$ and *resources*: $R \to L \cup P$. S denotes a set of states, usually a game's states. An *action a on S* is a mapping $a: S \to S$. A set of actions is denoted by A.

The first basic constraint for actions is *spatial coherence*: a player can pick up or dispose of a resource only at the player's current location, and no resource can appear or disappear at a location without a player's involvement. To describe the temporal aspect that differentiates location-based games from board games, the designer assigns a duration to all actions.

The designer also assigns each game state a value that expresses a player's interest in that state. In tic-tac-toe, the values of interest are *open*, *X-wins*, *O-wins*, and *draw*, with the intuitive semantics that *open* is assigned to non-end-game states. The designer specifies the synchronization interval's length (`syncTime`) as a positive real number.

The second basic constraint for actions is *temporal coherence*: every action consumes time equivalent to the sum of its duration and the synchronization interval.

Taking Turns in Games

In most classic board games, players alternate moving. Nevertheless, a typical board game involves more strategic reasoning than most location-based games that have been proposed so far. Not surprisingly, board games have made a digital reappearance in the form of computer-augmented tabletop games.[1] Some noncomputerized variants of chess, such as Progressive Chess or Double Move Chess (www.chessvariants.org), try to enrich the game experience by lifting restrictions on turn-taking to some degree.

The geogames described in this article go one step further by replacing turn-taking restrictions with real-time locomotion in a real-world environment. Traditional AI techniques for turn-based games such as the minmax algorithm can be used in this context but must be adapted to handle players' concurrent moves or simultaneous actions. For real-time strategy games, Alexander Kovarsky and Michael Buro developed a randomized minmax algorithm combined with a sampling-based evaluation strategy to search the corresponding state space under real-time restrictions.[2]

Two of us (Peter Kiefer and Sebastian Matyas) presented another variant of the minmax algorithm for balancing a geogame at design time.[3] This algorithm considers concurrency in location-based games by taking into account the game's spatiotemporal configuration.

References

1. C. Magerkurth, T. Engelke, and M. Memisoglu, "Augmenting the Virtual Domain with Physical and Social Elements," *Proc. Int'l Conf. Advancements in Computer Entertainment Technology*, ACM Press, 2004, pp. 163–172.

2. A. Kovarsky and M. Buro, *Heuristic Search Applied to Abstract Combat Games*, LNCS 3501, Springer, 2005, pp. 66–78.

3. P. Kiefer and S. Matyas, "The Geogames Tool: Balancing Spatio-Temporal Design Parameters in Location-Based Games," *Proc. 7th Int'l Conf. Computer Games* (CGAMES 05), Q. Mehdi and N. Gough, eds., Univ. of Wolverhampton, 2005, pp. 216–222.

Definition: A *geogame* $G = (S, A, time, value, \mathsf{syncTime})$ consists of two mappings, *time*: $A \to R^+$ (the set of positive real numbers) and *value*: $S \to V$, where V denotes the value space for state evaluation, and a constant $\mathsf{syncTime} \in R^+$.

The *time* mapping doesn't necessarily need to be proportional to the geodetic distances in the geographic environment, because players will rarely be able to move as the crow flies. Furthermore, the possibility of using physical objects to obstruct players' movements offers an additional degree of freedom that's supported by the Geogame definition.

Although the definitions together describe a large class of games, not all location-based games are geogames. Most important, games that don't satisfy the spatiotemporal coherence constraints—in which resources magically jump around the board—aren't geogames.

GeoTicTacToe

We can easily describe a spatial version of tic-tac-toe, which we call GeoTicTacToe, in a way that fits with the two Geogames definitions given earlier. Two players, $P = \{P_X, P_O\}$, play the game on a board with locations $L = \{L_{11}, \ldots, L_{33}, \mathrm{Start}_X, \mathrm{Start}_O\}$, where X and O are the tokens; that is, $R = \{X_1, \ldots, X_6, O_1, \ldots, O_6\}$. The game's states are described by their distribution of resources. For instance, *start* = (*locationStart*, *resourcesStart*) with *locationStart*(P_X) = Start_X, *locationStart*(P_O) = Start_O, and *resourcesStart*(X_1) = … = *resourcesStart*(X_5) = P_X, and similarly for the resources denoting the O-tokens. In practice, the framework explicitly describes only the starting state and constructs the other states by applying actions to the starting state.

Analysis

The synchronization time interval specified by a geogame's **syncTime** constant is crucial to the game. In principle, we could find suitable values for the parameter by playing and evaluating numerous games with different parameter settings. Although we improved early versions of the CityPoker game that way, the trial-and-error approach is hardly satisfactory as a method for game design.

However, we can find adequate values for the **syncTime** parameter by systematically exploring the game's state space. To do this, we use a Geogames analysis tool that relies on a simple player behavior model. A player first decides which location to move to next (several possibilities), then moves toward that location, arriving some time later (no lost players). Then, the player decides which actions to take upon which resources, and takes those actions. The player waits **syncTime** and then moves to the next location. The Geogames analysis tool makes several additional assumptions—for instance, that players move as fast as they can and that they don't waste time by waiting longer than the synchronization interval. Finally, the tool assumes that each player will try to win.

With these assumptions, the Geogames analysis tool explores the state space using a generalization of the minmax algorithm. The generalized algorithm handles multiple players and determines the next player to move according to the time units players need for their actions. This requires several modifications of minmax (see the sidebar "Taking Turns in Games"). Consider, for example, two or more players arriving at a location in the same instant of time (concurrent action upon resources), necessitating the incorporation of randomized elements. Furthermore, appropriate pruning strategies become essential for state spaces larger than that of GeoTicTacToe.

We can evaluate game-end states using an evaluation function that is derived from the geogame's value mapping. The result propagates from the game tree's leaf nodes to its root as the standard minmax would. Finally, the values at the starting state induce a player ranking and thus represent the game's outcome. This ranking gives the game designer a preliminary idea about the fairness of the game he or she has created: a completely fair game ends with all players having the same rank. Note that fair games aren't necessarily challenging location-based games.

The architecture of the Geogame analysis tool features four functional layers: a search mechanism, the Geogames engine, a concrete geogame, and a parameterized geogame. This architecture lets us easily model and analyze any geogame and to experiment with different search mechanisms and parameterizations (such as different starting points and spatializations). This includes the implementation of advanced search algorithms such as maxn or paranoia[4] for multiplayer games or pruning strategies such as null move pruning[5] for

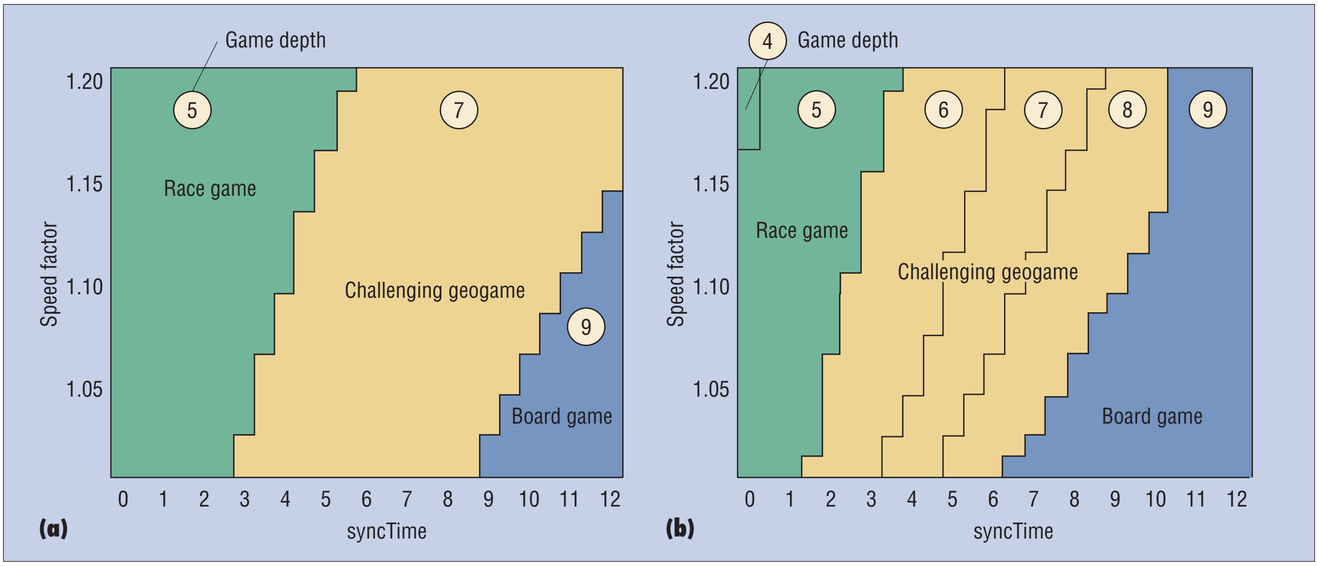


Figure 3. GeoTicTacToe results (a) for the game board shown in figure 1a, and (b) for the game board shown in figure 1b.

more complex games. Only the Geogames engine layer is fixed, as it reflects our general framework.

Evaluation

We can evaluate the game's state space for different syncTime values as well as for different values of the speed factor—that is, the ratio of the average speed of the X-player divided by the average speed of the O-player. To illustrate differences between various spatializations, we evaluated the two spatial versions of tic-tac-toe depicted in figure 1. Values for syncTime ranged from 0 to 12 in steps of 0.5, and values for the speed factor ranged from 1.01 to 1.20 in steps of 0.01. The result of the evaluation is described by three parameters:

- *Player ranking*. Because the slower O-player can't win, only two outcomes are possible: the X-player wins or the game ends in a draw.
- *Game depth* (the number of X- and O-tokens placed at game end). Each end state has a depth value between 3 (one player could place three tokens) and 9 (all tokens have been placed), which is propagated through the tree along with the corresponding evaluations. Winners prefer the end state with smaller depth. Obviously, depth correlates with ranking: a depth smaller than 9 always means a win for the X-player, and a depth of 9 could result in either a win or a draw. Our study didn't have any wins at depth 9.
- *Optimal path* through the game tree corresponding to the game in which both players act optimally. Usually, more than one optimal path exists.

Figure 3a shows the results for GeoTicTacToe played with the geographic footprint configuration shown in figure 1a. The result clearly shows the effect of the length of the synchronization interval. For small values of syncTime, the game's depth doesn't exceed 5. The faster X-player wins these games by racing. The O-player can't prevent the X-player from setting three X-tokens in a straight line.

On the other hand, high values for syncTime lead to the two players alternating moves, as in the board version of tic-tac-toe. The interesting range for the syncTime parameter lies, for example, between 3 and 9 if the players' speeds differ no more than two percent (the *speed factor* is less than 1.02), leading to games that end at depth 7. Games with depth 6 through 8 are a winning situation for the X-player, as are race games at depths 4 and 5.

We consider a location-based game with greater depth more interesting, mainly for two reasons. First, discovering the winning strategy is cognitively more difficult, and second, implementing the winning strategy is harder because of real-world obstacles. For instance, the time needed to move in an urban environment between two board positions might vary because of traffic lights. This unpredictability increases with greater depth, so there's good reason to call these games challenging geogames.

Figure 4 illustrates an optimal path (the game course) for a syncTime of 5, a speed factor of 1.02, and depth 8. At the beginning of this game, it looks race-style. Player X starts running through the upper horizontal line, while player O occupies the middle spot 5. But because of the syncTime interval, player X is forced to wait; meanwhile, player O can prevent a fast win by taking location 1. Player X in return blocks player O from winning by moving to location 9. This in turn forces player O to move to location 6 so that player X can't get three in a column. Finally, player X now can take advantage of his speed and takes locations 8 and 7 before player O can reach either of them. This type of move sequence, blending logical reasoning with physical locomotion, is generally found at depths 6 through 8 and creates what we consider a challenging geogame.

Using the Geogames analysis tool, we can also study the effect of different choices of geographic footprints, including the starting locations. Figure 3b shows a depth-versus-syncTime plot comparable to that of figure 3a but for the geographic footprint configuration shown in figure 1b. We found different boundary values delimiting race-style games, challenging geogames, and classic board games. The footprint configuration generating figure 3b promises more interesting games than that generating figure 3a, because all depth values from 3 through 9 actually occur.

Designing a geogame

So, to design a geogame, choose a synchronization interval within the value range corresponding to challenging geogames—for GeoTicTacToe, we find this at a game depth between 6 and 8. Within this range, you can emphasize speed or reasoning. You can analyze any other geographic footprint or other assumptions (for instance, about speed ratios), the same way. In general, then, you can

1. select a classic board game with interesting strategic elements,
2. choose alternative sets of geographic footprints in vista or environmental space for the board positions,
3. model the resulting location-based game within the conceptual framework, and
4. use the analysis tool to derive interesting values for the syncTime parameter.

Our main goal in describing our conceptual framework and analysis tool was to show how to determine a synchronization interval numerically so that you can implement challenging geogames in geographic space. Knowing the game area's real dimensions, you can easily translate the syncTime parameter from abstract time units to seconds. For instance, when choosing the spatialization illustrated in figure 1a for a city-size game board of 2 × 3 km and pedestrians with a speed of 3 meters per second, one time unit would correspond to approximately 83 seconds. In other words, a syncTime of 5 time units corresponds to approximately 7 minutes of waiting.

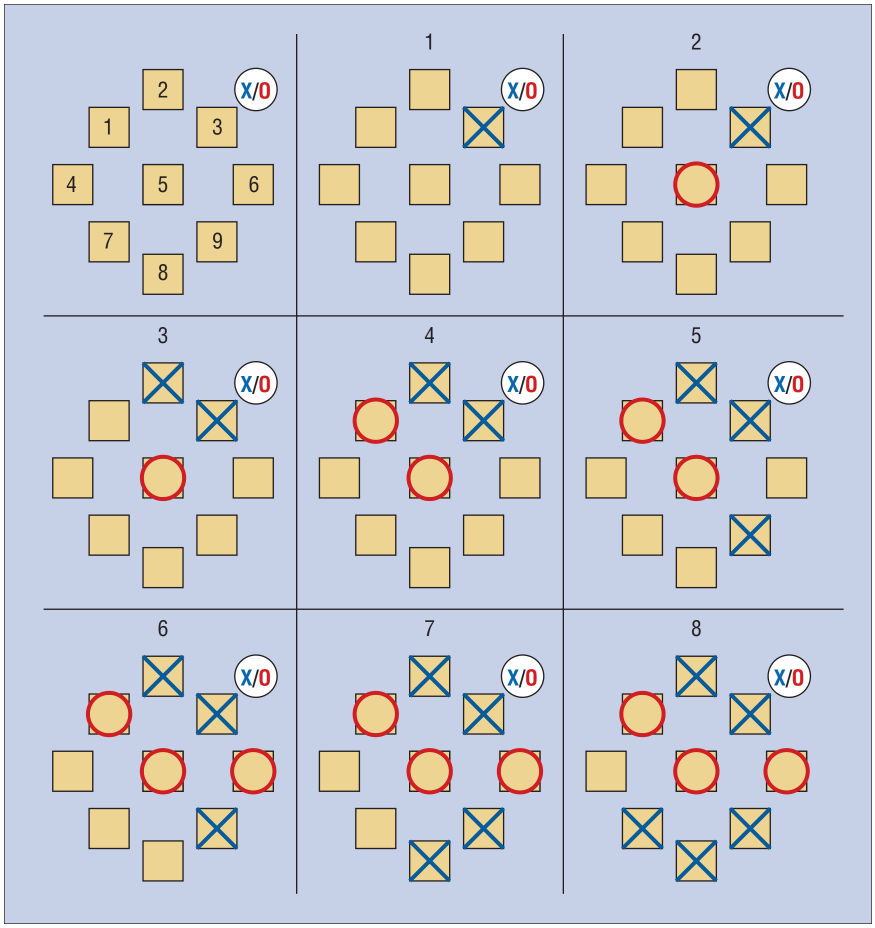


Figure 4. Game flow of a challenging geogame.

We are currently exploring how to best fill the syncTime interval by presenting interactive content (mini games, quizzes, or puzzles). These elements can add extra value to a geogame in tourist or educational scenarios. Edutainment, most recently that of learning with mobile games, is a growing research topic.[6,7] Geogames with their intrinsic syncTime interval constitute an ideal medium for transporting edutainment content.

syncTime analysis, as we've implemented it thus far, uses some simplifying assumptions. For instance, we assumed that players don't wait longer than the synchronization interval. In test-playing our games such as GeoTicTacToe, we never encountered player behavior that contradicted that assumption. However, it's well known from games with more complex state spaces that sometimes the best move is not to move but to let the opponent move twice. Obviously, this could lead to deadlocks in which the players wait for each other to move. Therefore, complex games such as chess possess *zugzwang* rules, which enforce making a move.

Because of the simplifying assumptions, the Geogames synchronization logic is almost trivial: move to the next position, wait syncTime, move to the next position, and so on. With this logic, players can't wait indefinitely. It could be interesting to permit players of geogames with a complex state space to wait longer than the synchronization interval. We'll then need some sort of *zugzwang* rules to avoid deadlocks. A natural framework for describing more complex synchronization mechanisms is temporal logic. But from an empirical point of view, we must first explore the practical usefulness of different synchronization mechanisms. Which mechanisms can human players manage cognitively? Which are adequate for edutainment applications? Once we've done that work, we can then describe candidate synchronization mechanisms, including *zugzwang* rules, in a sufficiently expressive temporal logic (such as propositional linear temporal logic or first-order linear temporal logic) that we could, for instance, prove safety and progress properties.

Another area for future research is comparing different geogame spatializations. This includes studying the scalability of the Geogames design approach by creating and comparing spatializations at different scales in vista and environmental space—for example, scaling a game from a sports field to a whole country. Furthermore, we plan to spatialize board games with more complex state spaces as well as multiplayer games.

Another line of research that originally led us to designing geogames is the recognition of user intentions from motion behavior. The problem of intention recognition consists of assigning meaning and goals to a person by analyzing his or her motion in a spatial environment. The location-based game CityPoker, for example, was originally devised as a scenario to study intention recognition.[3] Being able to interpret and predict a user's intentions will help game designers present information automatically on mobile devices in situations with limited interaction possibilities—for instance, playing GeoTicTacToe on bikes and with smart phones, as in our current implementation (see the "The Game Interface" sidebar).

Although social interaction plays an important role in geogames, single-player versions are also desirable. The design of a virtual smart opponent with appropriate AI methods running on the limited capabilities of a mobile device is yet another object of future research.

Acknowledgments

This is an extended version of our article, "Geogames: A Conceptual Framework and Tool for the Design of Location-Based Games from Classic Board Games," *Proc. 1st Int'l Conf. Intelligent Technologies for Interactive Entertainment* (INTETAIN 05), M. Maybury, O. Stock, and W. Wahlster, eds., LNAI 3814, Springer, 2005, pp. 164–173.

References

1. D. Montello, "Scale and Multiple Psychologies of Space," *Proc. Conf. Spatial Information Theory* (COSIT 93), LNCS 716, Springer, 1993, pp. 312–321.
2. D. Nicklas, C. Pfisterer, and B. Mitschang, "Towards Location-Based Games," *Proc. Int'l Conf. Applications and Development of Computer Games in the 21st Century* (ADCOG 21), L.W. Sing et al., eds., 2001.
3. C. Schlieder, "Representing the Meaning of Spatial Behavior by Spatially Grounded Intentional Systems," *GeoSpatial Semantics*, LNCS 3799, A. Rodríguez et al., eds., Springer, 2005, pp. 30–44.
4. N. Sturtevant, "Current Challenges in Multi-Player Game Search," *Computers and Games: 4th Int'l Conf., CG 2004*, LNCS 3846, Springer, 2006, pp. 285–300.
5. O.D. Tabibi and N.S. Netanyahu, "Verified Null-Move Pruning," *ICGA (Int'l Computer Games Assoc.) J.*, vol. 25, no. 3, 2002, pp. 153–161.
6. K. Facer et al., "Savannah: Mobile Gaming and Learning?" *J. Computer Assisted Learning*, vol. 20, no. 6, 2004, pp. 399–409.
7. G. Schwabe and C. Göth, "Mobile Learning with a Mobile Game: Design and Motivational Effects," *J. Computer Assisted Learning*, vol. 21, no. 3, 2005, pp. 204–216.

The Game Interface

Geogames are intended to bring game playing back to the physical world. Although game areas for geogames can scale from vista to environmental space, our current implementation of GeoTicTacToe is designed for a citywide game board with bicycles as the means of transportation (see figure A1). Players are localized using a GPS receiver (not necessarily installed on the handlebar) that communicates with the smart phone via Bluetooth. A map of the game area appears on the phone's screen. Figure A2 shows the map, enlarged. Each player can see his or her current location marked with a small pink cross (1) on the map. As soon as one of the players reaches an unoccupied location (one of the yellow squares), a small X or O appears, indicating that the location is reserved. After the syncTime interval has passed, the mark enlarges, and the player is free to move on. If a player leaves a reserved location, he or she loses that location and the small mark disappears.

Because of the high speed of locomotion, possibilities for interaction with the mobile device are severely limited. A main design objective for the GeoTicTacToe assistant system consists in minimizing the need for user interaction. Our current research focuses on the context-aware presentation of game information (for example, automatic zooming).

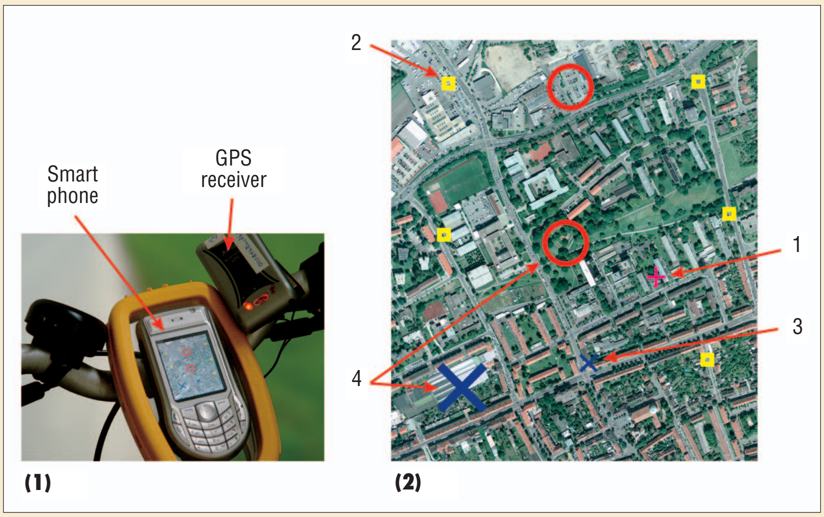
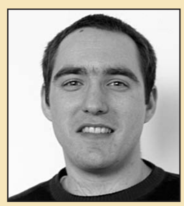


Figure A. Playing GeoTicTacToe: (1) A player's bicycle handlebar. (2) The game area map. (figure courtesy of the State Office for Survey and Geographic Information, Bavaria, Germany, 2005)

The Authors

Christoph Schlieder is a professor of computing in the cultural sciences at the University of Bamberg. His research focuses on developing and applying methods from semantic information processing to problems from the cultural sciences, including qualitative spatial reasoning, preferred mental models, and geospatial semantics. He received his PhD in computer science from the University of Hamburg. Contact him at Otto-Friedrich-Universität Bamberg, Feldkirchenstr. 21, D-96045 Bamberg, Germany; christoph.schlieder@wiai.uni-bamberg.de.

Peter Kiefer is a research assistant and PhD candidate in applied computer sciences at the University of Bamberg. His research interests are in knowledge representation, especially intention recognition from motion patterns. He received a graduate degree in information systems from the University of Bamberg. Contact him at Otto-Friedrich-Universität Bamberg, Feldkirchenstr. 21, D-96045 Bamberg, Germany; peter.kiefer@wiai.uni-bamberg.de.



Sebastian Matyas is a research assistant and PhD candidate in applied computer sciences at the University of Bamberg. His research focuses on semantic data integration in geographic information processing. He received a graduate degree in information systems from the University of Bamberg. Contact him at Otto-Friedrich-Universität Bamberg, Feldkirchenstr. 21, D-96045 Bamberg, Germany; sebastian.matyas@wiai.uni-bamberg.de.