Increasing Accessibility of Online Board Games to Visually Impaired People via Machine Learning and Textual/Audio Feedback: The Case of "Quantik"

Giorgio Gnecco[1,2](✉), Chiara Battaglini[3], Francesco Biancalani[1], Davide Bottari[3], Antonio Camurri[4], and Barbara Leporini[5]

[1] AXES Research Unit, IMT School for Advanced Studies, Lucca, Italy
{giorgio.gnecco,francesco.biancalani}@imtlucca.it
[2] Game Science Research Center, IMT School for Advanced Studies, Lucca, Italy
[3] MoMiLab Research Unit, IMT School for Advanced Studies, Lucca, Italy
{chiara.battaglini,davide.bottari}@imtlucca.it
[4] DIBRIS Department, University of Genoa, Genoa, Italy
antonio.camurri@unige.it
[5] ISTI Institute, National Research Council, Pisa, Italy
barbara.leporini@isti.cnr.it

Abstract. Playing board games is commonly recognized as an effective way to promote the integration and socialization of their participants. However, visually impaired people may encounter accessibility issues when playing online versions of board games, for instance, because such versions may have been designed having initially sighted people in mind. Given this premise, the aim of this work is to design an interface aimed to help visually impaired people play board games online, via an improved interaction with a normal or touch screen. This goal is achieved by means of automatic recognition of the portion of the screen one's finger or the cursor is pointing to, its classification via machine learning, and the use of either textual or audio feedback. In this way, a visually impaired person could explore the screen in quite a natural way, obtaining information, e.g., about the positions of the various pieces on the board. As a case study, a preliminary version of the interface is developed to address accessibility of the online version of a carefully selected pure strategy abstract board game, namely "Quantik" from Gigamic.

Keywords: Visual Impairment · Board Games · Accessibility · Machine Learning

1 Introduction

Playing board games is usually recognized as an effective way to improve the integration and socialization of their participants. In particular, for the case of visually impaired people (i.e., persons affected by any functional limitation of their vision, which cannot

M. Clayton et al. (Eds.): INTETAIN 2023, LNICST 560, pp. 167–177, 2024.
https://doi.org/10.1007/978-3-031-55722-4_12

be corrected by means of either corrective glasses or contact lenses, see Bailey et al. [1]), it is well-known that an increase of their autonomy in playing board games has positive effects on their quality of life and in general, on their personal fulfillment (da Rocha Tomé Filho et al. [13]). However, to promote their effective inclusion, it is important that visually impaired people are put in the conditions of playing together with other people, with similar winning chances, notwithstanding their visual abilities.

Unfortunately, on one hand, the recent Covid-19 pandemic has severely limited social interaction in the real world, making it quite difficult for visually impaired people to play board games by sitting together at the same table and interacting with the board, e.g., in a tactile way[1]. On the other hand, the pandemic has also greatly increased online interaction via the Internet.

Nevertheless, online interaction is often designed having sighted people in mind, so it can present severe accessibility issues to visually impaired people and especially to blind people, in case it is mainly based on visual information, and this is not replaced by either textual or audio feedback. Although some systems already exist to increase accessibility to these categories of people of some digital games (Morelli and Folmer [10]) and particularly of a few online board games[2], in general, their possibility to play board games online is still quite limited. This issue is particularly relevant for either less known or recently developed board games.

In this context, the goal of this work is to design an interface aimed to increase the accessibility of online versions of board games to visually impaired people, forming an additional layer between existing screen readers (mostly focused on making textual content accessible) and online board games, via automatic recognition and either textualization or sonification of their main visual elements. As a case study, instead of considering a general-purpose version of the interface, we limit the focus to its preliminary version aimed at addressing the accessibility of the online version of a carefully selected pure strategy abstract board game, namely "Quantik" from Gigamic.

The article is structured as follows. Section 2 describes the game "Quantik", analyzes it, and reports some accessibility issues of its online version, discussing how they could be solved using machine learning and either textualization or sonification techniques. Section 3 outlines the design of the first version of a possible accessible interface for the online version of "Quantik". Section 4 concludes the work with a discussion and delineates its possible future extensions.

[1] See the following hyperlink: https://www.youtube.com/watch?v=pT8vZZS7hZo for a demonstration of the relevance of this modality of interaction, with reference to the case of a chess tournament involving blind people.

[2] See, e.g., the case of the chess server Lichess: https://lichess.org/blog/U5AX_DcAADkAz-L5/accessibility-for-blind-players.

2 "Quantik": Description, Analysis, and Accessibility Issues

The two-player board game "Quantik" (https://en.gigamic.com/game/quantik) is a recent pure strategy abstract game, which was published by Gigamic in 2019. It was inserted in the list of Mensa Recommended Games in 2021. Its online version is available on the Board Game Arena gaming platform, at the following hyperlink: https://en.boardgamearena.com/gamepanel?game=quantik. An illustration of its board and pieces is provided in Fig. 1.

Fig. 1. "Quantik" board (image taken from https://en.gigamic.com/game/quantik).

2.1 Description

The following is a description of the game "Quantik". Each of the two players has a set of eight game pieces in the color associated with that player (light for one player, dark for the other player). The pieces have four different shapes (ball, cone, cube, and cylinder), and, for each player, there are two identical pieces for each shape. Players take turns placing one piece per round on an empty space of the board according to the following single rule: a player is not allowed to place a shape in a row, column, or quadrant on which the opponent has already placed a piece of the same shape. The first player to place the fourth different shape in either a row, column, or quadrant wins the game. For this winning condition, it does not matter if the game pieces already present in the specific row, column or quadrant belong to one player or to her/his opponent. A player wins also if she/he can move in the current turn in such a way that the other player has no admissible move at the next turn. Hence, the game never terminates with a draw.

2.2 Analysis

From a theoretical point of view, it is worth highlighting that "Quantik" is a two-person sequential zero-sum finite game with perfect information[3] (Maschler et al. [9]). Interestingly, one can apply Zermelo's theorem[4] (see, e.g., Peters [11]) to analyze any game belonging to this class. In the specific case of "Quantik", one can prove in this way that one of the two players has a winning strategy, i.e., a strategy that enables such player to win no matter which sequence of moves is chosen by her/his opponent. In principle, finding which player has a winning strategy and constructing that winning strategy can be achieved by using backward induction, relying on the so-called extensive form of the game (i.e., its representation based on a game tree and a list of players' payoffs on the terminal nodes), and constructing the so-called value function, which provides, for every node of the tree, the best payoff that can be guaranteed to the player who has to choose a move in that node. In practice, to apply backward induction, one would need to examine all the nodes of the game tree, moving from its terminal nodes (leaves) to its initial node (root), searching each time for an optimal move for the player whose turn is associated with the specific node under examination. Although finding the value function exactly is computationally intractable for many two-person sequential zero-sum finite games with perfect information, in the specific case of "Quantik" there are actually some computational savings, due to the following reasons:

[3] Two-person sequential zero-sum finite games with perfect information model a quite broad class of board games characterized by the following issues: there are two players, who move alternatively, and with opposite goals (e.g., when one wins, the other loses, and vice versa); at any turn, each player has a finite number of choices, and any run of the game always terminates after a finite number of turns (finiteness of the game); each player is able to observe all the previous moves (perfect information). A classic example of a game belonging to this class is provided by chess under any finite termination rule. Although several theoretical results are known for this class of games, their application to the analysis of board games is quite limited. This is motivated, e.g., by: a) the bounded rationality of real-world players, which prevents them to apply too complex strategies (i.e., choices of their moves as functions of the currently available information); b) the fact that solving in closed form such games (i.e., finding exactly their so-called subgame-perfect strategies) can be computationally intractable when these games are characterized, e.g., by a too large board, a too large number of different pieces, and/or a too complex set of rules (associated, e.g., with the possibility of moving the pieces on the board after their first positioning). In the case of "Quantik", these issues are likely reduced, for the following reasons: (i) "Quantik" has a 4×4 board, which is much smaller, e.g., than the 8×8 board used in chess. (ii) The game is characterized by a quite small number of pieces for each player (eight), which are indistinguishable in groups of two. (iii) Once positioned on the board, each such piece cannot be moved from its current position. (iv) Positioning each piece on the board restricts – often severely – the set of admissible positions for the next pieces.

[4] Zermelo's theorem states that, for any two-person sequential zero-sum finite game with perfect information, one (and only one) of the following cases occurs: (i) The first player has a winning strategy. (ii) The second player has a winning strategy. (iii) Both players have a drawing strategy. It is recalled here that a winning strategy for a player is a strategy that enables such a player to win no matter which sequence of moves is chosen by her/his opponent, whereas a drawing strategy for a player is a strategy that guarantees that she/he does not lose under that strategy, whatever is the sequence of moves chosen by the other player. In the specific case of "Quantik", only one of the two cases (i) and (ii) occurs, since the game never terminates with a draw.

– Some moves can be seen immediately to be suboptimal for one player because they lead to the other player's win in a few moves. Hence, starting from a specific node in the game tree, some moves could be immediately excluded from the search for optimal moves for that node.
– The game is characterized by some symmetry: e.g., rotating the board by 90 degrees along a vertical axis produces a node in the game tree that is equivalent to the original node, in the sense that the value function assumes the same value on these two nodes. This depends on the fact that the game subtrees originating from those two nodes have the same structure, and each leaf in one of the two subtrees has the same pair of payoffs for the two players as the corresponding leaf in the other subtree.

Nevertheless, in case finding the value function exactly for "Quantik" turned out to remain computationally intractable in spite of the computational savings above, one could apply suitable advanced machine-learning techniques for its approximation, such as approximate dynamic programming/reinforcement learning (Platt [12], Xenou et al. [15]). Finally, starting from such approximation, one could construct an approximation of a winning strategy.

2.3 Accessibility Issues

For what concerns accessibility issues, first, it is important to remark that the presence of a single (and simple) rule in "Quantik" allows the player to understand easily how the game proceeds and makes it unnecessary to resort to a complicated rulebook (e.g., one containing either several images or a single complex image, difficult to be interpreted by standard optical character recognition systems), which would make the game inaccessible to people with visual impairment (Bolesnikov et al. [2]). Second, it is worth observing that "Quantik" belongs to the category of games that, according to Thevin et al. [14], are deemed to be complicated to make accessible by means of hand-crafted adaptation solutions, since its board configuration is extremely relevant to play that game. However, due to the small size of that board and its low number of different pieces, "Quantik" (particularly, its online version) looks particularly suitable for either a textualization or a sonification of its rows, columns, and quadrants, which would provide visually impaired people with precious information to construct a mental map of the content of the game board. It is expected that, in this way, such players could play competitively with sighted people. Indeed, both the single rule of the game and its winning condition depend on information about pieces located in the same rows, columns, or quadrants. Nevertheless, before applying either textualization or sonification, automatic recognition of the pieces is needed, e.g., to associate each piece with a specific text or sound. In the case of online playing of the game, this could be achieved quite easily, when the gaming device has direct access to the internal state of the game. As an alternative, machine learning can be used to identify the pieces. Compared to the previous solution, this has the advantage of not requiring the additional prior knowledge of the internal state of the game, nor the use of possibly advanced computer programming skills to access that state. In this way, the same machine-learning techniques applied for one game are likely to be applicable to other games, by making only minor modifications to the code. Moreover, machine learning could be used to suggest a specific sonification to the user, depending on her/his needs. For instance, distinct users may have different preferences about timber, pitch, and volume. Hence, one could exploit users' similarities to optimize the selection of the sonification for a specific user.

3 Design of the First Version of a Possible Accessible Interface for the Online Version of "Quantik"

Taking into account the issues highlighted in the section above, a first version of an accessible interface for visually impaired people playing the game "Quantik" online was implemented in MATLAB R2023a, then run on a notebook Intel® Core™ i7850U CPU@1.80 GHz–1.99 GHz under the Windows 10 Enterprise LTSC environment. The interface was thought for use in the two following consecutive phases:

- In the first phase, a sighted individual collects a subset of images directly from the web page of the game and labels them according to their shape. Then, a machine-learning model is trained/validated by means of an augmented dataset obtained by random translation and the addition of white Gaussian noise with varying variance to each element of the subset above. More precisely, the union of the training/validation sets is made of 500 images per class (corresponding to 50 corrupted images for each of the 10 images initially collected and labeled per class). Validation is achieved by using the holdout method, giving the same size to the training/validation sets. Then, a convolutional neural network (composed of the following consecutive layers: 2-D convolutional layer; relu layer; max pooling 2-D layer; fully connected layer; softmax layer; classification layer) is trained/validated to classify objects as belonging to one of four classes, each corresponding to one of the four shapes of pieces that are used in the game "Quantik": ball, cone, cube, and cylinder. The choice of a neural network with a feedforward structure is motivated by its excellent approximation capabilities, a property that is valid already in the case of a single hidden layer (Gnecco et al. [4–6], Gnecco and Sanguineti [7]). Being the specific learning task multi-class classification (with the four classes being represented by one-hot encoding), cross-entropy is used as the loss function (Goodfellow et al. [8]). Training is performed based on stochastic gradient descent with momentum. The resulting validation accuracy turns out to be 94.5% (see Fig. 2). Finally, being it easier to distinguish between the two different colors of the pieces used by the two players, color classification of an image is achieved first by finding the color at its center, then attributing it to the nearest of the colors associated with the two players.
- In the second phase, the same individual or another individual (possibly with visual impairment) navigates the web page, then the trained/validated machine-learning model is tested on the images generated in real time by that user.

All the training/validation/test images have the same size (315×317 pixels, reduced to 26×26 pixels before sending them as inputs to the convolutional neural network), and are possibly obtained also by zooming in/out around the current position of the

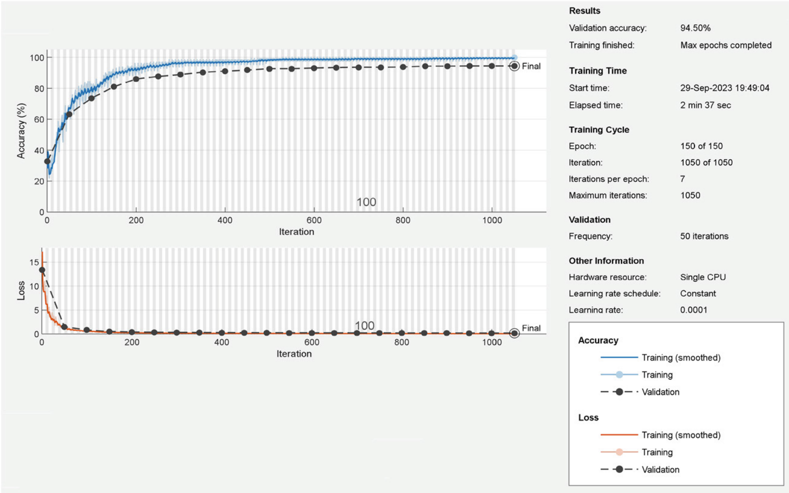

Fig. 2. Details on training/validation of the learning machine.

cursor/finger on the screen. Each image is centered on the position of the cursor/finger on the screen at the time of generation of that image.

Navigation on the screen can be performed in several different ways, all based on the same MATLAB interface: using the mouse, moving a finger on a touch screen, or using Leap Motion (https://www.ultraleap.com), which is an optical hand-tracking device able to capture movements of the hands. The last modality is expected to be more natural for a blind person, who may be not accustomed to the first two modalities.

The screen is split into two equally-sized parts (see Fig. 3):

– In the left part, the user can navigate the online version of "Quantik", which is opened automatically by MATLAB. This is achieved by accessing its associated hyperlink (https://en.boardgamearena.com/gamepanel?game=quantik) with the default web browser.
– The right part shows one figure ('*Interface.fig*'), which contains several subfigures. Subfigure a) represents a scaled version of the left part of the screen, where the user can move the cursor/place her/his finger. This redundancy is due to the fact that, at least under the operating system used, MATLAB can easily access the content of the whole screen, but it looks that it cannot directly access the content of a window associated with another program (e.g., with the default web browser). Subfigure b) presents a zoom of a rectangular subregion of the image around the current position of the cursor/finger in Subfigure a). Subfigure c) reports the classification of Subfigure b) obtained by the trained/validated machine-learning model, by showing a prototype image corresponding to the recognized class and color[5]. The names of the recognized class and color are also reported at the top of Subfigure c)[6]. For each of the

[5] This visual feedback is useful only for sighted people, for a direct evaluation of the accuracy of the classifier. An analogous remark holds for the visual feedback in Subfigures d) and e).

[6] This textual feedback is potentially useful for screen readers, as an alternative to sound feedback.

four classes, a continuous sound is produced (a different sound for each class). This is changed in case of a change in the classification (due to a successive movement of the cursor/finger). Pieces with the same shape but associated with different players are sonified using the same kinds of sounds, played with a different pitch. Finally, Subfigures d) and e) show, respectively, the horizontal and vertical trajectories of the cursor/finger in Subfigure a) during the last s seconds (being s a user-configurable parameter). Such trajectories are potentially useful for a multiscale spatial or temporal analysis of user behavior, at a higher level than simply detecting cursor/finger movement.

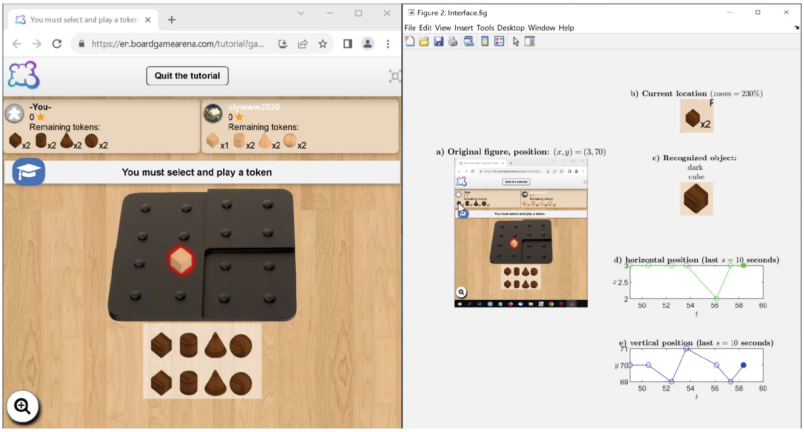

Fig. 3. Screen splitting in the current version of the interface.

The core of the interface relies on event-based MATLAB programming, applied to the figure '*Interface.fig*'. In particular, the occurrence of events associated with three properties ('*WindowButtonMotionFcn*', '*WindowButtonDownFcn*', and '*KeyPressFcn*') of that figure is captured automatically. More specifically, the activation of one of these three properties is associated with the execution of the respective callback function. For instance, in the case of a mouse interface, the first two properties refer, respectively, to mouse movement and mouse clicking. The last property refers to key pressing. The three callback functions are described as follows:

– When the user moves the cursor/finger on the figure '*Interface.fig*', the first callback function is executed. The position of the cursor/finger is detected, and its horizontal/vertical coordinates – expressed as percentages of the total width/height of Subfigure a) – are reported near that subfigure. These coordinates are suitably thresholded when the cursor/finger overcomes one of the horizontal/vertical boundaries of Subfigure a), in such a way as to get always numbers between 0 and 100. Then, the contents of Subfigures b), c), d), and e) are updated accordingly.
– When the user produces a left click, the second callback function is executed. The current image in Subfigure b) is saved in the dataset used for successive training/validation of the machine-learning model (after its suitable data augmentation).

A first persistent variable is used to count the number of figures saved so far, to avoid overwriting. In the current version of the interface, labels for those figures are provided manually by the user.

– When the user presses the left/right arrow key or the '*p*' (play) key, the third callback function is executed. The first two cases correspond with a decrease/increase (by 10%) of the zoom applied to Subfigure b), always remaining in the range [10%, 1000%]. A second persistent variable is used to keep track of the current zoom. The third case corresponds with the activation of the sonification of the currently recognized object, shown in Subfigure c). Persistent variables are used to keep track of the color/shape of that object. According to the current implementation, the sounds used are: '*gong*' (ball), '*splat*' (cone), '*handel*' (cube), '*train*' (cylinder). They are played at two successive octaves (higher for the opponent's pieces, to increase the attention of the player towards those pieces, as their detection is relevant for the application of the single rule of the game).

4 Discussion and Possible Extensions

It is worth comparing the proposed interface with the advanced electronic device designed by Caporusso et al. [3] to provide accessibility to an online version of another board game (specifically, chess). That device was conceived having deaf-blind people in mind. Hence, no sonification is generated by it. Instead, haptic feedback is provided to the player, about both the position of the pointer on the digitalized board and the identity of the object located in that position. However, for people having only visual impairment, a sonification of visual information appears to be a more natural kind of feedback (especially for the case of the game "Quantik", due to the reduced size of its board, compared to that of chess). Moreover, it is worth noting that touch screens were not so widely diffused at the time of publication of Caporusso et al. [3], so an alternative software (rather than hardware) interface seems preferable, being it likely cheaper than the device proposed at the time in that work. Finally, differently from Caporusso et al. [3], the proposed interface makes extensive use of supervised machine learning.

Among possible extensions of this research, we mention:

– *An improvement in the design of the multi-class classifier*. For instance, one could include a fifth class, made of other objects (extracted, for example, from the web page of the specific online board game) not belonging to any of the four main classes. The presence of this fifth class would be motivated by the opportunity of producing no textual/sound output (or, alternatively, producing a low-volume background sound) when the image surrounding the current position of the cursor/finger does not belong to one of the four main classes. This goal could be also achieved by still using only four classes and including an image segmentation step, in such a way as to activate the multi-class classifier in the test phase not for every test image, but only when its segmentation satisfies a suitable constraint (e.g., that the segmented object of interest – the foreground – does not encounter the boundary of the image, or has only a sufficiently small overlap with that boundary).

– *An improvement in the accuracy of the classifier*, achieved, e.g., by using a larger/better-constructed dataset for training/validation, and/or a different architecture for the learning machine.

- *An improvement in the effectiveness of the interface*, motivated by the fact that, in the current implementation, Subfigures a)-e) are updated every time a movement of the cursor/finger is detected, slowing down the execution of the code.
- *An improvement in accessibility of the interface*, motivated by the fact that its current version needs the presence of a sighted individual to generate the dataset used to train/validate the learning machine.
- *An improvement in the choice of the sonification*, possibly based on machine learning and users' feedback. For instance, sonification could be combined with machine learning to suggest/disadvise the choice of specific moves. Moreover, polyphonic/spatialized sonification (using headphones) and/or orchestration techniques could give the user global information about the locations of several pieces on the board, providing her/him additional help to construct a mental map of the content of the game board. Sonification should not require player's learning but provide instead a metaphor that naturally reflects the structure of the board and the single rule of the game. Finally, it should be chosen in such a way as to be pleasant to the player and not to put her/him under stress.
- *The application of the interface to real gaming sessions*, and the collection and analysis of related feedback from users (for instance, initially blindfolded sighted people, then persons really affected by visual impairment).
- *Its extension to the online versions of other board games* (such as "Quarto") *and card games,* based, e.g., on training/validation sets specific for each such game.

Acknowledgment. The authors were partially supported by the Game in Lab 2022 project "Increasing accessibility of online board games to blind and visually impaired people via machine learning", funded by Game in Lab, and by the PRIN 2022 project "Multiscale Analysis of Human and Artificial Trajectories: Models and Applications", funded by the European Union - Next Generation EU program (CUP: D53D23008790006). Giorgio Gnecco dedicates the work to the memory of his mother Rosanna Merlini.

References

1. Bailey, I.L., Hall, A.: Visual impairment: an overview. American Foundation for the Blind (1990)
2. Bolesnikov, A., Kang, J., Girouad, A.: Understanding tabletop games accessibility: exploring board and card gaming experiences of people who are blind and low vision. In: Proceedings of TEI 2022: The 16th International Conference on Tangible, Embedded, and Embodied Interaction, article no. 21, pp. 1–17 (2022)
3. Caporusso, N., Mkrtchyan, L., Badia, L.: A multimodal interface device for online board games designed for sight-impaired people. IEEE Trans. Inf. Technol. Biomed. **14**(2), 248–254 (2010)
4. Gnecco, G., Gaggero, M., Sanguineti, M.: Suboptimal solutions to team optimization problems with stochastic information structure. SIAM J. Optim.Optim. **22**(1), 212–243 (2012)
5. Gnecco, G., Kůrková, V., Sanguineti, M.: Can dictionary-based computational models outperform the best linear ones? Neural Netw.Netw. **24**, 881–887 (2011)

6. Gnecco, G., Kůrková, V., Sanguineti, M.: Some comparisons of complexity in dictionary-based and linear computational models. Neural Netw.Netw. **24**, 171–182 (2011)
7. Gnecco, G., Sanguineti, M.: On a variational norm tailored to variable-basis approximation schemes. IEEE Trans. Inf. Theory **57**, 549–558 (2011)
8. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
9. Maschler, M., Solan, E., Zamir, S.: Game Theory. Cambridge University Press, Cambridge (2013)
10. Morelli, T., Folmer, E.: Real-time sensory substitution to enable players who are blind to play video games using whole body gestures. Entertain. Comput. **5**(1), 83–90 (2014)
11. Peters, H.: Game Theory: A Multi-leveled Approach. Springer Berlin Heidelberg, Berlin, Heidelberg (2015). https://doi.org/10.1007/978-3-540-69291-1
12. Platt, A.: Deep Reinforcement Learning. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-8285-7
13. da Rocha Tomé Filho, F., Mirza-Babaei, P., Kapralos, B., Moreira Mendonça Junior, G.: Let's play together: adaptation guidelines of board games for players with visual impairment. In Proceedings of CHI 2019: CHI Conference on Human Factors in Computing Systems, article no. 631, pp. 1–14 (2019)
14. Thevin, L., Rodier, N., Oriola, B., Hachet, M., Jouffrais, C., Brock, A.M.: Inclusive adaptation of existing board games for gamers with and without visual impairments using a spatial augmented reality framework for touch detection and audio feedback. In: Proceedings of the ACM on Human-Computer Interaction, vol. 5, no. ISS, article no. 505, pp. 1–33 (2021)
15. Xenou, K., Chalkiadakis, G., Afantenos, S.: Deep reinforcement learning in strategic board game environments. In: Slavkovik, M. (ed.) EUMAS 2018. LNCS (LNAI), vol. 11450, pp. 233–248. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-14174-5_16