Coaching a Machine to Play Othello

A quick walkthrough to the *Coachello*[*] GUI

V. Markos

March 10, 2023

Abstract

In this brief walkthrough, we present *Coachello* (v.0.1.0), a Graphical Interface (GUI) designed to assist users in coaching a machine to play Othello. We start by discussing the basic rules of the game and then proceed to present the interface in detail, using several examples to illustrate its core functionalities.

1 TL;DR

Version:	0.1.0
Web page:	vmarkos.github.io/coachello
GitHub repository:	github.com/VMarkos/coachello
License:	GNU General Public License v3.0

2 Othello in Brief

We start by presenting in short the rules of Othello. In Figure 1 we demonstrate the initial board setup for a typical game of Othello.

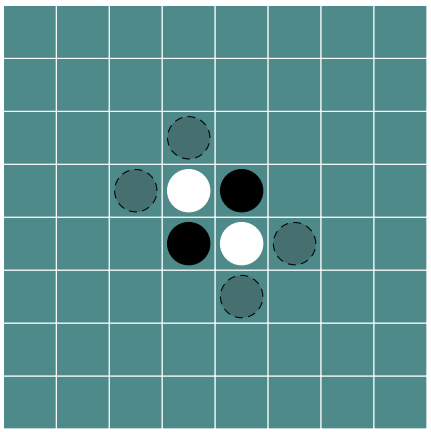


Figure 1: The initial board setup in a game of Othello. Four pieces are placed symmetrically in the center of the board, allowing for four (4) legal moves for the black player — who always plays first.

- The game is played with two (2) players, on an 8×8 board.
- Each player is assigned a color (black/white), whit the black playing always first.
- Whenever a player places a stone on the board, any opposite color stones that are enclosed between two player stones in a straight line (horizontally, vertically or diagonally) are reversed to the current player's color.
- Each player is allowed to place stones in cells that lead to reversing at least one opposite color stone (which explains why there are four legal moves for the black player in the initial board setup).

[*] *Coachello*: From "coach" and "Othello", so no direct reference to the "Coachella" festival.

3 The Interface

We describe the GUI of Coachello through various simple examples. At first, let us have a look at Figure 2. There we demonstrate the starting screen of the interface. There are five (5) important areas in that screen:

1. The first area on the left part of the screen corresponds to the playing area. There is where everything happens. As we have already discussed in Section 2, Othello is played on an 8×8 board. So, why have we decided to use a 10×10 board in our interface? Observe how the first and last rows and columns of the board share the same color, comprising some sort of "frame" around the actual playing board. This frame is intended to be used during coaching to facilitate expressing patterns regarding border cells. More on that a few paragraphs below, in Section 3.1.

2. Right above the main playing area, there are two black and white dots, indicating the score of the current game. As discussed above, the score is simply determined by the number of same color stones each time on the board.

3. On the top right of the screen there is an "Info" icon, which redirects to the Wikipedia lemma for Othello[1].

4. In the middle of the right part of the starting screen, there are two buttons, corresponding to the two modes offered by the interface. Precisely, the "Play" button redirects one to the playing screen, where one can play against an agent or human or even have two agents play against each other.

5. The "Audit" button redirects to the auditing screen, where one can load any previously played game and coach the machine.

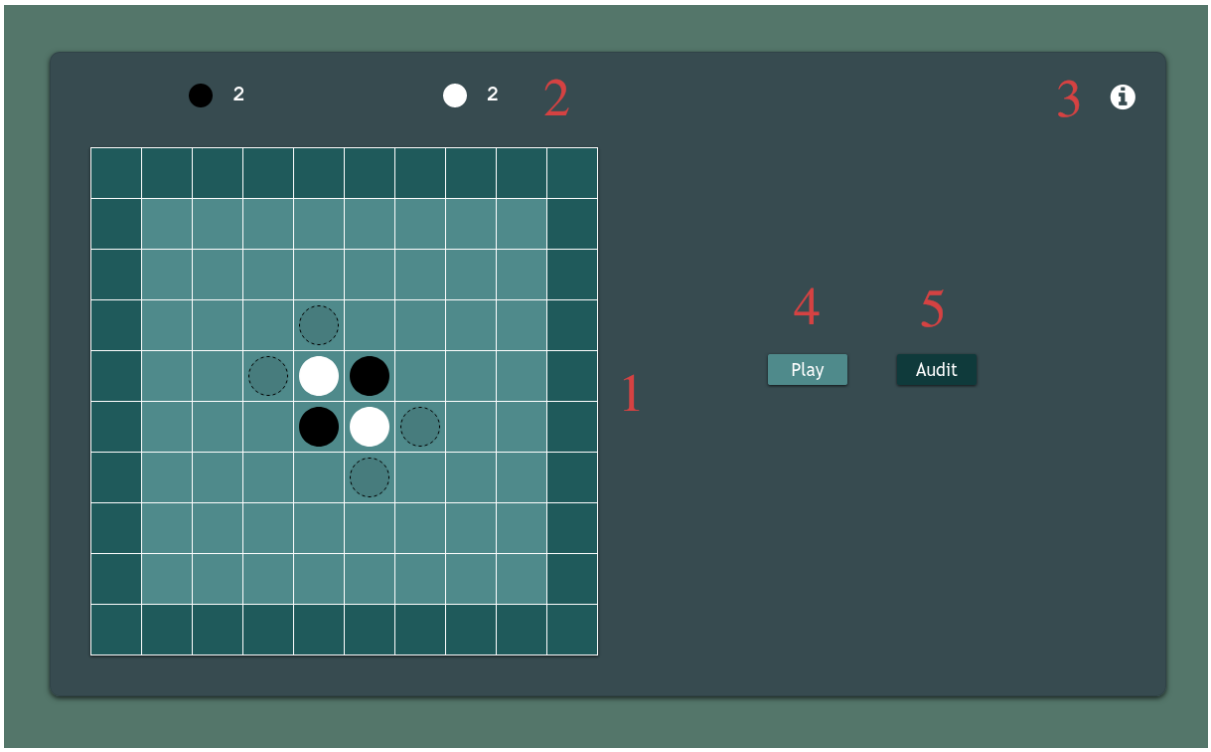


Figure 2: The starting screen of the *Coachello* interface.

[1] https://en.wikipedia.org/wiki/Reversi

3.1 Play mode

Pressing the "Play" button redirects us to the playing screen, as shown in Figure 3. There are eight (8) important features in this page:

1. The "Player" option refers to whether the player of that color is a human or a machine (Prudens) one.
2. The "Policy" option, applicable only to Prudens players, refers to the playing policy used by that player (more on that in the following Section, 3.2). By default, not specifying a policy means that the agent plays at random.
3. The "Wait" option, applicable only to Prudens players, refers to the (minimum) time a Prudens player is required to wait before making its move.
4. The "Animate Moves" option refers to whether moves should be animated or not in case two Prudens players are playing against each other (by default checked).
5. The "# Games" option, again available only in Prudens versus Prudens games, refers to the number of games that two agents should play against each other.
6. The checkmark on the bottom right of the settings panel finalizes game settings and allows for the game(s) to start[2].
7. The "Save game" option, active once a game has started, saves the game and automatically downloads the corresponding game file to your (browser's default) downloads directory.
8. The panel at the bottom of the right side of the screen shows game history of the currently played game.

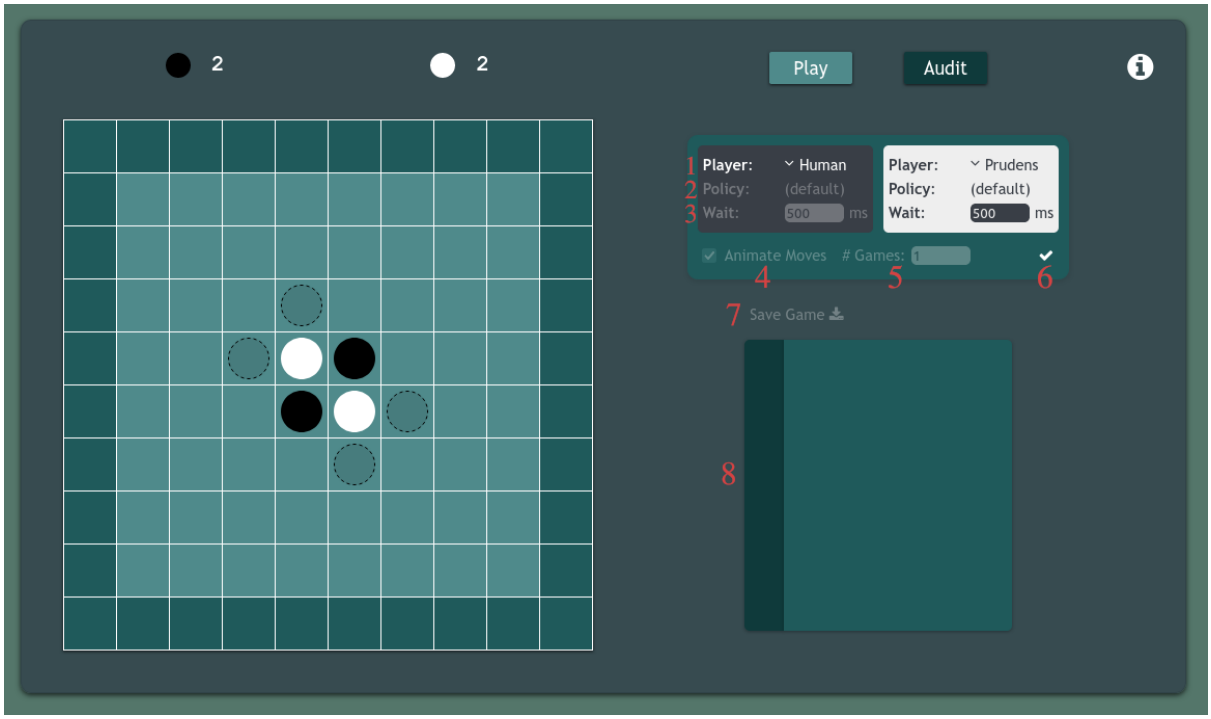


Figure 3: Playing mode, where a user might play a game against another user (over the same screen) or an agent, as well as have two agents play against each other.

There also some remarks regarding Play mode that should be made at this point:

[2] In case the black player is human, there is also the possibility to directly start the game by making the first move

- In case one chooses to change settings while playing a game, the checkmark (option 6 in Figure 3) turns into a "Refresh" button, which, once clicked, prompts user to save the current game and start a new game (with new settings) from scratch.
- While playing, one might still press the "Play" and "Audit" buttons. Precisely, by pressing the "Play" button, the user is prompted to save the currently played game and start a new game (loading default settings). Similarly, when one presses the "Audit" button while playing, there is a prompt to save the current game before the interface switches to audit mode.
- When playing as a human, any move may be made by clicking on any available legal move.

3.2 Audit mode

Pressing the "Audit" button redirects one to the audit mode screen, as shown in Figure 4. There are several actions a user might make in this mode. Namely:

1. "Current Game" indicates the currently loaded game (which is clickable, in case one wants to load another game).
2. The "Player" option indicates, as in play mode, whether the player of that color is a human one or not.
3. The "Policy" option, applicable only to Prudens players, indicates which policy was used when playing.
4. The "Game Date" indicates the date and time the game was saved.
5. The "Offer advice" option is activated in intermediate states (see below, options 8 and 10), in which a Prudens player was playing. Once pressed, the user is redirected to the coaching screen.
6. The two black and white download buttons allow the user to download the policy of any Prudens agent in that game. In case one of the two players is human, this option is deactivated (as with the black download button in Figure 4, option 6).
7. The first column in the game history panel displays current move number.
8. The second column in the game history panel corresponds to the state before the next black move. That is, the state where the player is about to play (but has not yet played) their next move. While in the case of human players these states are of little significance, when it comes to Prudens agents, these are the states when a human coach might provided any advice to the agent.
9. The third column in the game history panel shows the move played by the black player, encoded using the typical board format (columns: a,b,…,h, rows: 1,2,…,8, starting from top left).
10. The fourth column in the game history panel shows the states right before the next white move (as in option 8).
11. The fifth column in the game history panel shows the moves played by the white player, using the same notation (but for, columns are indicated by upper-case letrers).
12. The "fast backward" button corresponds to moving directly to the first move played in this game.
13. The "backward" button corresponds to moving to the previous move or intermediate state.
14. The "forward" button corresponds to moving to the next move or intermediate state.
15. The "fast forward" button corresponds to moving directly to the last move played in this game.

We shall also remark at this point that one might navigate throughout the game by clicking on the moves or intermediate states within the game history panel. Also, note that any navigation options are activated or deactivated depending on whether it is possible to move forward or backward.

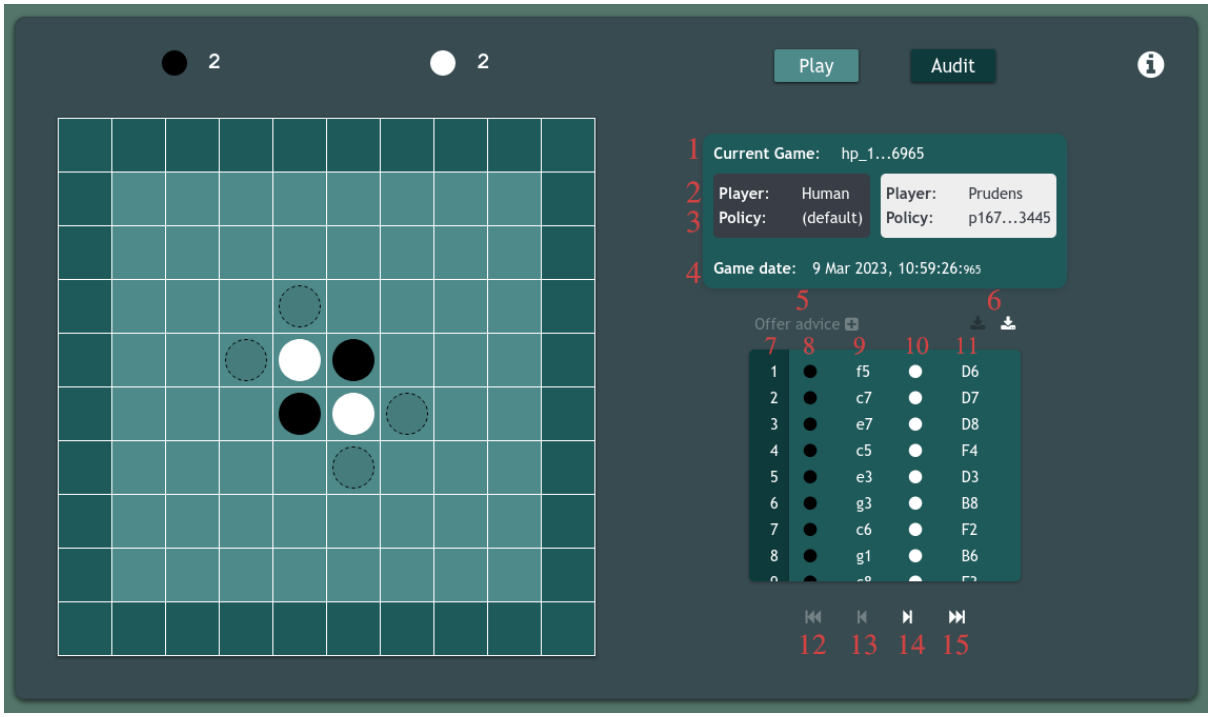


Figure 4: Audit mode, where a user might review an already played game, having access to the internal reasons that have led a Prudens agent to play a certain move and then provide any required further advice to the machine, if needed.

3.3 Coaching

As discussed in Section 3.2, by pressing the "Offer advice" button, one is redirected to the coaching screen, as shown in Figure 5.

1. The top right area on the coaching panel displays any patterns generated by the provided pattern using all square symmetries (i.e., four reflections and three rotations). This is a beta feature, aiming to facilitate the understanding of end-users regarding the internal interpretation of the provided pattern.
2. The "Advise" button might be used to provide single pieces of advice to the agent. Note that this button is activated only when the provided pattern has at least one body and one head cell (see below for more details about that).
3. The "Cancel" button cancels the process without saving any changes.
4. The "Done" button closes the coaching panel, saving any changes made.
5. There is always one highlighted (red) cell on the board, corresponding to the move played by the agent next.

4 A Coaching Example

Next, we shall demonstrate how one can coach an agent with no prior knowledge by providing two pieces of advice. In general, to do so, the user has to click on the cells they want to include in their pattern (which encodes a single piece of advice). There are three types of cells, indicated by color:

- Body cells, denoted by orange color;
- Positive head cells, denoted by blue color;
- Negative head cells, denoted by red color.

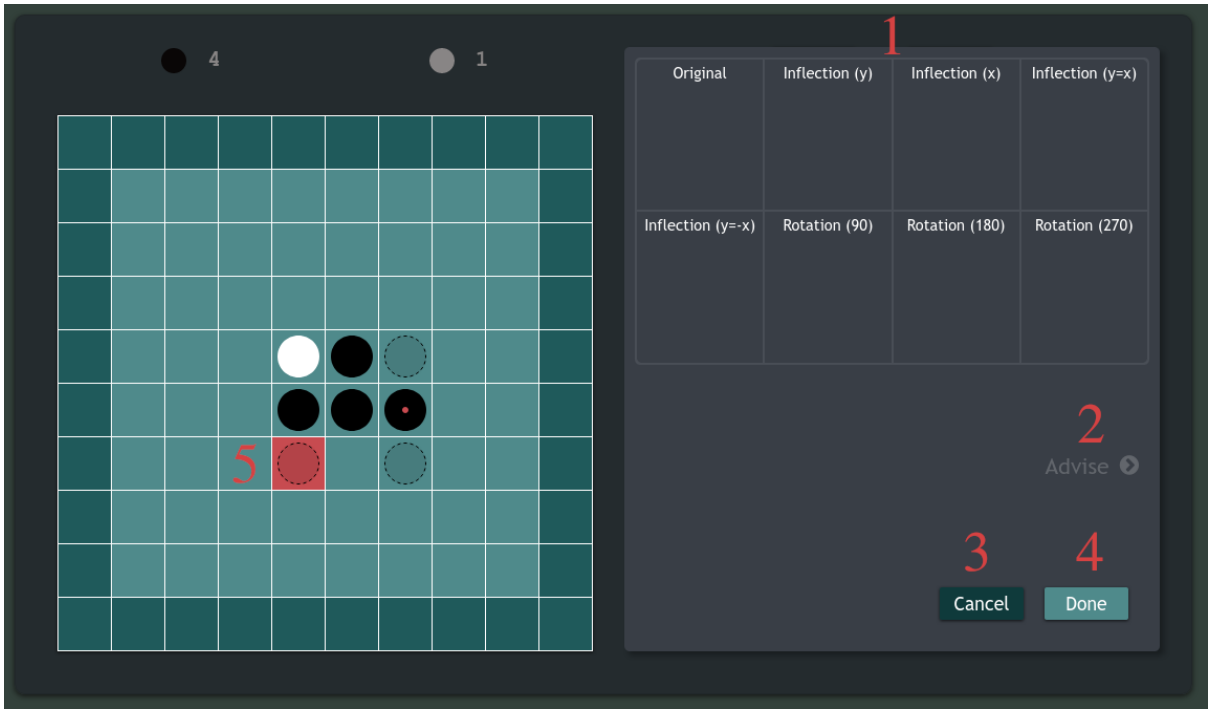


Figure 5: Offering advice to the machine. In this screen, the user might click on the game board and design patterns that are interpreted as cases in which certain moves that should be preferred.

There is also the option to change the semantics of each cell, as shown in Figure 6a. But, let us be more precises. Assume that we want to advise the agent to prefer corners when possible. To do so, we could provide a pattern like the one shown in Figure 6b. Essentially, the provided pattern instructs the agent to consider any cells that adhere to the following pattern:

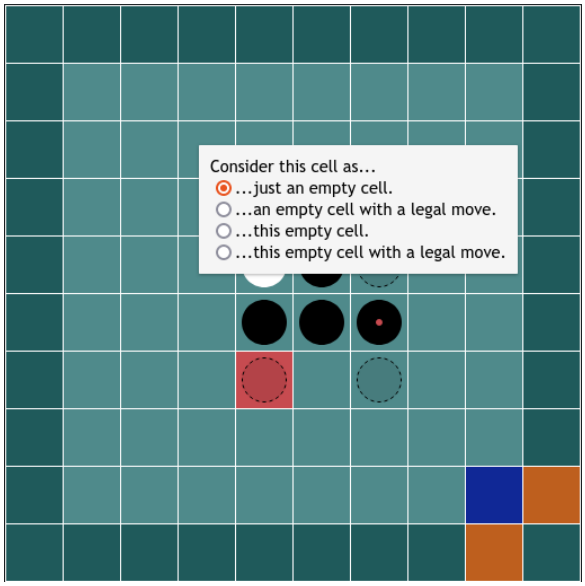
- are next to two cells on the border of the board and;
- these border cells are "vertical" in the sense that they are on different sides of the board.

Then, pressing the "Advise" button, we add that piece of advice to the agent's knowledge base.

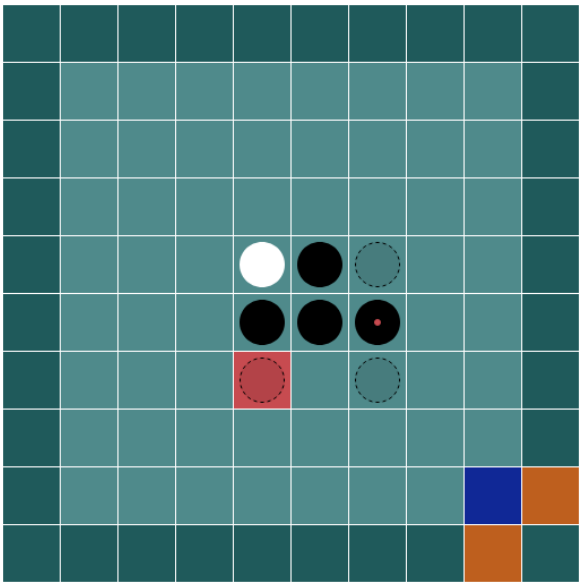
Let us now assume that we would also like to tell our agent to avoid playing to cells next to corners. To do so, we create the "negative" rules shown in Figures 6c and 6d. The rationale behind the patterns is similar, using border cells as points of reference to instruct the agent not to play to certain types of cells. Note that, while we provide a pattern concerning a certain corner of the board, taking into account any possible symmetries, the agent learns the variants of this pattern that correspond to all four corners.

5 Contact

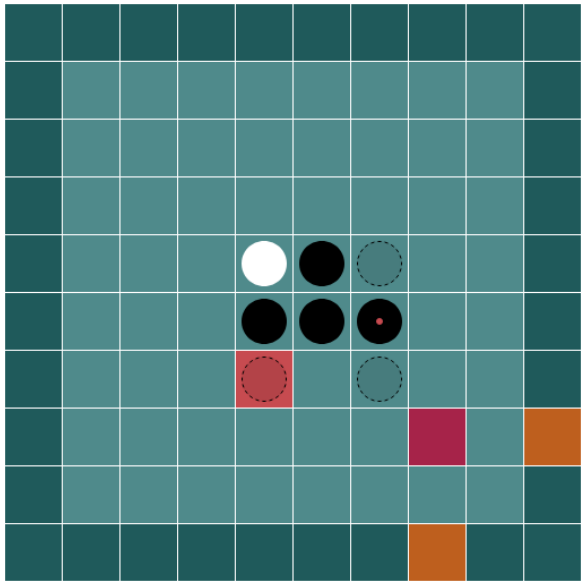
For any bugs, suggestions or any other comment you might have about *Coachello*, you can contact us through email at `vasileios.markos@st.ouc.ac.cy`.



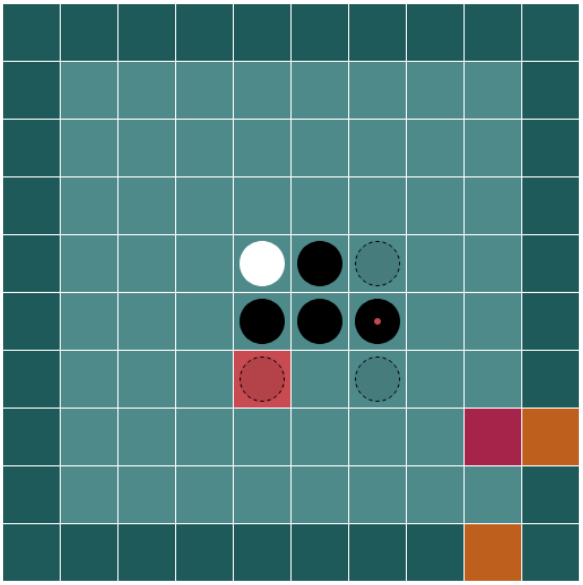
(a) Depending on the cell type (e.g., border cells or board empty/occupied cells), the user might choose among several different semantics, as shown above.



(b) We could describe a corner cell as one that is next to two "vertical" border cells. Using that, one can concretely demand the agent to prefer corners over other moves.



(c) Similarly to the case of a corner cell, in order to describe a diagonally adjacent cell to a corner we can make use of two border cells, as before.



(d) Similarly to the case of a corner cell, in order to describe a diagonally adjacent cell to a corner we can again make use of two border cells.

Figure 6: Using simple patterns as rules that explain basic heuristics to the agent.