Human-Machine Cooperation Loop in Game Playing

Maciej Świechowski*, Kathryn Merrick†, Jacek Mańdziuk‡§, Hussein Abbass†
*Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland
Email: m.swiechowski@ibspan.waw.pl
†School of Engineering and Information Technology, University of New South Wales, Canberra, Australia
Email: (k.merrick, h.abbass)@adfa.edu.au
‡Faculty of Mathematics and Information Science, Warsaw University of Technology, Warsaw, Poland
§School of Computer Engineering, Nanyang Technological University, Singapore
Email: j.mandziuk@mini.pw.edu.pl

***Abstract*—This paper presents a new concept for human-machine iterative cooperation based on the Upper Confidence Bounds Applied to Trees method. Analysis of such human-computer cooperation may potentially lead to pertinent insights related to performance improvement of both the human subject and an artificial agent (machine) during certain kinds of strategic interactions. While the experiments described in this paper refer to the so-called General Game Playing (being a certain embodiment of multi-game playing) the overall idea of proposed human-machine cooperation loop extends beyond game domain and can, in principle, be implemented in the form of a flexible general-purpose system applicable to cooperative problem solving or strategic interactions of various kinds. The concept proposed in this study is evaluated by means of a direct involvement of human subjects in specifically defined cooperative environment, which provides vast opportunity to learn from and cooperate with an artificial game playing agent under the certain rules of cooperation. The choice of games is seemingly important to this kind of experiment. Although the participants played better with the assistance of the machine in some of the games, they lost the track in subsequent matches when the assistance was (intentionally) no longer available. Possible reasons for such an activity pattern are discussed in the conclusions. Three design iterations showing evolution of the experiment setup are presented in the paper. The analysis of cooperative and non-cooperative matches reveals different patterns for each chosen game characterized by various levels of advantage gained by means of cooperation.**

Keywords–human-machine cooperation; human learning, game playing; UCT algorithm.

I. Introduction

In this paper, we extend the finding and ideas proposed in our previous conference paper [1] devoted to human-machine cooperation in General Game Playing [2]. Based on the outcomes of the initial experimental studies presented in [1], a new refined experimental setup and new extended experimental results are proposed in this paper. The conclusions listed in [1] are revisited, strengthened and better motivated.

General Game Playing (GGP), by some researchers claimed to be one of the 'grand AI challenges' in games and a step towards machine manifestation of possessing human-like intelligence [3], [4], gained recently a lot of attention due to popularization of the annual world-wide General Game Playing Contest organized by people affiliated with Stanford Logic Group at Stanford University [5]. This event is currently the most prominent embodiment of the multi-game playing idea, which aims to create systems capable of playing a variety of games (as opposed to agents that can only play single games). Hence, the design, study and verification of approaches that allow for cooperation between humans and GGP agents is an important research avenue and complementary research stream to the mainstream research activities. We believe that analysis of such human-computer cooperation may potentially lead to pertinent insights related to performance improvement of both the human subject and an artificial agent (machine) during certain kinds of strategic interactions.

In this study, we continue our attempts to building a system that would enable effective cooperation between human subject and machine player by means of iterative strategic cooperation. We borrowed the concept of cooperation from [6] stating that it takes place when two systems cause each other to modify their behavior to achieve some mutual advantage. Since our focus is on human-machine cooperative loop, we do not explicitly discuss in the paper the principles of cooperation between humans. The readers interested in this area may be willing to consult [7], or [8] as a starting point.

The particular type of strategic interaction considered in this research is collaborative game playing. The type of machine cooperator will be a GGP [2] agent based on the Monte Carlo Tree-Search (MCTS) method. The MCTS is used as the main routine of the strongest state-of-the-art GGP players and is also widely applied to other games such as Go [9] or Arimaa [10] as well as other areas of Artificial Intelligence (AI), including decision problems based on Partially Observable Markov Decision Processes [11], [12], [13], Dynamic Vehicle Routing Problems [14] or Risk-Aware Project Scheduling [15], [16].

The paper reports on several human user studies performed to validate the proposed approach to human-machine cooperation. We will start with presentation of two pilot experiments performed with the two following aims: (1) verification (in terms of effectiveness, clarity, user-friendliness) of the experimental setup for human-machine cooperation and (2) providing preliminary verification of our research hypothesis. Next, we will present a large-scale experiment, in which the disadvantages and pitfalls of the preliminary design were (to a large extent) eliminated. Clearly, apart from providing the circumstances for potentially successful cooperation, we are also interested in measuring the effects of such cooperation, i.e., verify to which extent it affects/improves the average quality of play.

While human-machine interaction [17] has been a hot research topic in various domains outside the game area, e.g., in aviation [18], complex products design [19] or surgery [20], surprisingly in games the task of creating strong machine players appeared to be challenging enough *per se* [9], [21], [22]. Consequently, to the best of our knowledge, except for our initial study [1], there has not been any related work concerning human-machine cooperation neither in GGP nor in the MCTS game-playing framework. We believe that the way we approach the problem of cooperation can contribute to the area of general knowledge-free and learning-based methods in games [23] and beyond this area, since we can examine the way humans learn from machines and *vice versa* provide a basis for development of automatic or semi-automatic methods by which machines can learn from humans how to play games (or solve problems in a more general perspective).

The remainder of the paper is organized as follows: the next two sections contain brief descriptions of GGP, MCTS and our cooperation platform within the MCTS framework. In Section IV the research hypotheses are formulated. Section V describes the two particular setups tested in the two pilot studies and summaries the main outcomes and conclusions that stemmed from these studies. Sections VI and VII present the new refined experimental setup and the outcomes of this main experiment, respectively. The last section is devoted to conclusions and directions for future work.

II. Introduction to General Game Playing and Monte Carlo Tree Search

In this section, a brief introduction to the General Game Playing and Monte Carlo Tree Search is provided. In particular, the way MCTS is applied to GGP is discussed in more detail, since the machine cooperator is actually a GGP player that relies on MCTS scheme enhanced by a bunch of local, lightweight heuristics.

A. GGP Preliminaries

GGP is a trend in AI which involves creating computer systems, known as GGP agents, capable of playing a variety of games with a high level of competence. The range of games playable within the GGP framework consists of any finite deterministic game. Unlike specialized playing programs, GGP systems do not know rules of the games being played until they actually start. The concept of designing universal game playing agents is also known as multi-game playing or metagaming, but as stated in the introduction, we refer to the Stanford's definition of GGP [2], which is the most recent one. The official GGP Competition, which is *de facto* the World Championship Tournament, is also part of the GGP specification. The machine player used in this research is our entry in the latest installment of the competition (2014). Borrowing from the GGP terminology, we will use the term *play clock* for the time (in seconds) available to make a move by a player. In addition to *play clock*, there is also *start clock*, which is the time spanning from starting the match until the first move is to be played. During the *start clock* players may think and do any kind of meta-gaming prior to the actual game start. To enable matches between our GGP program and humans, we had to slightly loosen the official specification. For instance, GGP agents are normally penalized for not responding with a legal move in time by having the move chosen for them at random. In our scenario, human participants can think about moves as long as they want to without any penalty and the machine players always respond

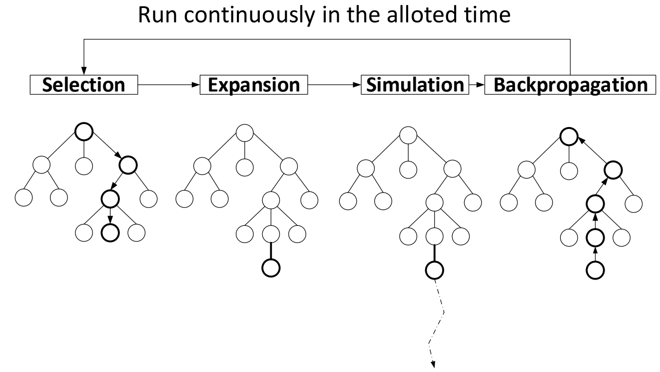

Figure 1. The four phases of Monte Carlo Tree Search algorithm. The algorithm is repeated until the alloted time runs out.

in time. In GGP, the rules of the games to play are a real-time input parameter written in a specialized language called Game Description Language (GDL) [24]. We will not go into details of this language other than it is relatively slow to interpret in a program, therefore, it required to set the *play clocks* to a relatively long time.

B. The Tree-Search Algorithms Used

MCTS is an algorithm for searching a game tree in a quasi-random fashion in order to obtain as accurate an assessment of game states as possible. In general, the assessment is computed statistically as the average score - **Q** - which is defined by the total score of simulations going through a state divided by the number of visits to that state. The total score is a sum of the outcomes of simulations. For all games considered in this article, the value of 1.0 denotes a win, 0.5 denotes a draw and 0.0 denotes a loss in a single simulation. The input to the method is the current game state. Then, the algorithm gradually searches the game tree starting from the current state in a series of iterations adding one node in each of them. An iteration, depicted in Figure 1, consists of the following four steps:

1) **Selection.** Start from the root and go progressively down. In each node, choose the child node according to a given selection policy until reaching a leaf node.
2) **Expansion.** If a state contained in the leaf node is not terminal, choose an action that would fall out of the tree. Allocate a new child node associated with that action.
3) **Simulation.** Starting from a state associated with the newly expanded node, perform a full game simulation (i.e., to a terminal state).
4) **Backpropagation.** Fetch the result of the simulated game. Update statistics (average scores, numbers of visits) of all nodes on the path of simulation, starting from the newly expanded node up to the root node.

In the classical MCTS implementations, the selection policy of child nodes in step 1) is either uniform (each child can be selected with equal probability) or greedy (a child with the highest score hitherto, i.e., the most promising one, is selected). A significant improvement over the pure MCTS is the Upper Confidence Bounds Applied to Trees (UCT) algorithm [25], which allows for maintaining a balance between the exploration and exploitation ratio in the selection step. Instead of sampling each action uniformly or greedily, the following scheme is advised:

1) **Selection.** If there are child nodes not yet selected (in previous simulations) choose one of them at random. Otherwise, select child node a^* according to the following formula:

$$a^* = arg \max_{a \in A(s)} \left\{ Q(s,a) + C\sqrt{\frac{ln\,[N(s)]}{N(s,a)}} \right\} \quad (1)$$

where ***s*** is the current state; ***a*** is an action in this state; ***A(s)*** is a set of actions available in state ***s***; ***Q(s,a)*** is an assessment of performing action ***a*** in state ***s***; ***N(s)*** is a number of previous visits to state ***s***; ***N(s,a)*** is a number of times an action ***a*** has been sampled in state ***s***; ***C*** is the exploration ratio constant.

In summary, the input to the UCT method is the current state. Then, the algorithm gradually searches the game tree starting from the current state in a series of simulations adding one node in each of them. Actions within the built part of the tree (the selection phase) are chosen according to Equation (1) whereas actions outside of the tree (the simulation phase) are chosen quasi-randomly with the help of lightweight heuristics (for the sake of clarity of the presentation, we will not get into details here but interested readers may consult our papers [26], [27]). Nodes store the average players' scores obtained in the process of these iterations.

In contrast to all the variations of min-max alpha-beta search, the MCTS is an aheuristic, anytime and relatively easy to parallelize method. Aheuristic means that there are no game-specific knowledge required (heuristics) so the method can be applied in general domains. The min-max-based search needs the heuristic unless the tree can be search thoroughly. Moreover, the algorithm can be stopped at virtually anytime and still give the currently best answer. It asymptotically converges to the perfect play although the convergence can be very slow. The quality (or accuracy) of the answer is a function of time allocated for the process. Finally, it is considerably easy to parallelize because many simulations can be run in parallel.

In order to illustrate the way the MCTS algorithm works, let us consider a game between two MCTS-based machine players in a well-known game of Connect-4. In each step, each of the players was allotted 7 second for a move. Four interesting steps were selected from the played game. Figure 2 shows snapshots of the game board taken in these four steps (after the move was performed). The statistics computed for each available action (move) just prior to the presented game positions are provided below in the text. Please note that Q and N denote the UCT parameters from Equation (1). The boundary values of Q are equal to 0 (expected loss for red) and 100 (expected victory for red), respectively.

In step 1, two moves putting stones in the middlemost columns, i.e., 4 and 5, were clearly the best for the red player.

```
STEP 1
[Drop in column 1]  Q: 50.0   N: 2319
[Drop in column 2]  Q: 53.3   N: 3294
[Drop in column 3]  Q: 58.4   N: 6743
[Drop in column 4]  Q: 61.7   N: 12829
[Drop in column 5]  Q: 61.4   N: 11896
[Drop in column 6]  Q: 57.4   N: 5677
[Drop in column 7]  Q: 52.5   N: 3017
[Drop in column 8]  Q: 50.8   N: 2522
```

In step 3, the best strategy for red was to continue putting stones in the middle and block the blue player:

```
STEP 3
[Drop in column 1]  Q: 56.1   N: 6263
[Drop in column 2]  Q: 55.9   N: 6036
[Drop in column 3]  Q: 54.4   N: 4882
[Drop in column 4]  Q: 59.3   N: 11395
[Drop in column 5]  Q: 59.7   N: 12208
[Drop in column 6]  Q: 53.1   N: 4094
[Drop in column 7]  Q: 55.4   N: 5601
[Drop in column 8]  Q: 53.9   N: 4552
```

In step 15, the highest Q value was equal to 55.6, which means that the red player was slightly more likely to win, but the MCTS/UCT algorithm does not see any path leading to a very strong position (very likely victory) of this player (and of the other, as well):

```
STEP 15
[Drop in column 1]  Q: 55.6   N: 44908
[Drop in column 2]  Q: 46.0   N: 4945
[Drop in column 3]  Q: 47.1   N: 5788
[Drop in column 4]  Q: 44.8   N: 4241
[Drop in column 5]  Q: 47.0   N: 5737
[Drop in column 6]  Q: 43.7   N: 3679
[Drop in column 7]  Q: 44.8   N: 4205
[Drop in column 8]  Q: 43.4   N: 3547
```

Finally, in step 41, three out of four actions were assigned $Q = 100$ meaning that they lead to a victory for red. Only the last action resulted in an immediate win, but the length of a winning path makes no difference to the MCTS/UCT assessment.

```
STEP 41
[Drop in column 2]  Q: 100.0   N: 2567553
[Drop in column 3]  Q: 26.3    N: 259
[Drop in column 7]  Q: 100.0   N: 2567833
[Drop in column 8]  Q: 100.0   N: 2681408
```

III. COOPERATION IN THE MCTS/UCT FRAMEWORK

The machine cooperator used in this paper is an adapted MiNI-Player [26], [27], [28] - a GGP tournament-class program equipped with additional features to enable cooperation. The cooperation is accomplished mainly by means of statistics provided by the machine to help humans choose which move to play. During both cooperative and non-cooperative plays, it is always a human who makes the final choice. When the statistics are presented, i.e., in the cooperative mode, it is up to the participant whether or not to take advantage of them.

The second means of cooperation is by permitting interference with the MCTS/UCT. At this point, we need to clarify that there are two versions of the program participants operate with: with and without cooperation. Both possibilities are embedded in the same (common) program and the cooperation option is either enabled or switched off. In this way, we propose an interactive process of building the game tree, while playing the game, involving both the machine and human. In the original MCTS/UCT, the same four-phase algorithm is repeated all the

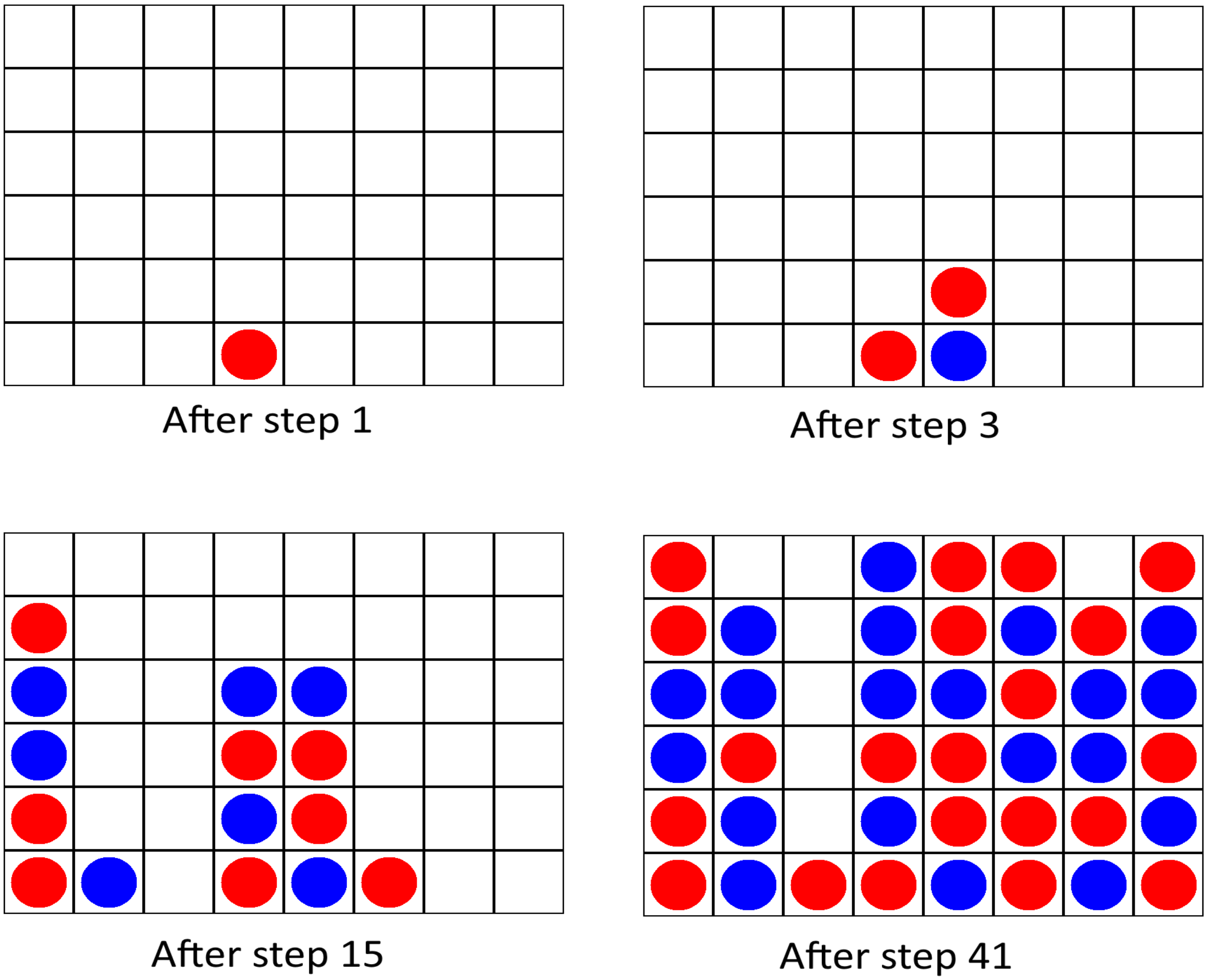

Figure 2. Board states after moves 1,3,15, and 41 extracted from a game between two MCTS-based computer playing programs in Connect Four.

time during the *play clock*. For cooperative purposes we split this time into three equal intervals T1 + T2 + T3 = *play clock*. Between any two consecutive intervals (T1 and T2 or T2 and T3) humans can interact with the MCTS/UCT based on statistics presented to them. The statistics include: each action $\boldsymbol{a}$ available to the player to make a move with the $\boldsymbol{Q(s,a)}$ and $\boldsymbol{N(s,a)}$ values from (1). These values are scaled to the [0%, 100%] interval to be more readable by the participants. The final statistic is the actual number of simulations which ended with a win, draw and loss for the subject, respectively. The MCTS/UCT can be directed by the human in two ways: enabling/disabling actions available in the current state or toggling priorities of the actions on/off. If an action is disabled, the MCTS/UCT will ignore this action in the selection step, which means that no simulations will start with a disabled action. Changing the priority is equivalent to changing the value of the C parameter in (1) from 1 to 10. Participants are allowed to make any number of the aforementioned interventions at each step and once they are done, they click the simulate button to submit all of them in one batch and observe how the statistics have changed. By doing so, they can help the machine to focus on the most promising actions and avoid presumably wasteful computations. In this way, the human→machine cooperation happens. On the other hand, the feedback from the machine supports or questions the above-mentioned human player's choices. The machine→human cooperation is naturally accomplished by means of providing human player with all moves-related statistics calculated by the machine during simulations and allowing human to react accordingly. The reaction can be realized either by setting up particular actions in the simulation phase or making a particular move in the played (real) game. Since there are three simulation phases before each move played in a game, the cooperation happens, *de facto*, in a loop.

Our experimental design is justified by the two following observations. First of all, in many well-established games, it has been found that the experts can intuitively discard unpromising actions and focus on the few best ones. Such behavior is manifested by human playing experience and intuition and is one of the aspects in which humans are better than machines despite having comparably "less computational power". Provided that the human choice is correct, the process can converge faster to the optimal play. The introduction of action priority is a similar, but slightly weaker, modification to the MCTS/UCT algorithm. The second observation (or assumption) we made is that the cooperation has to be easy for participants to understand. Therefore, we avoided asking them to set the internal parameters of the algorithm or to deal with more complicated operations on the tree structure. In fact, the participants do not even need to know that the tree exists.

IV. Research Hypothesis

In order to better focus the study on performance of human-machine cooperation, we formulated the following research hypothesis: **a human cooperating with a machine GGP agent is a better player than human or machine agent individually**. We write this thesis in a shortened form of H + M > M and H + M > H, where H denotes a human player; M denotes a machine player and M + H denotes a hybrid player comprising a cooperating machine and human.

We attempt to verify this hypothesis in a devoted experiment or at least make a step towards such a verification. The main research question is whether a mutually beneficial cooperation can originate and develop between human and machine players. In order to verify the above-listed hypotheses, we gathered samples from people playing without any machine assistance (H vs. M) and with such assistance (H+M vs. M). The first case involves a human simply playing a match against MINI-Player [26], [27], [28]. The second case involves a human playing against the same opponent but this time with assistance of a “friendly” GGP agent (a clone of MiNI Player) running in the background. Before we move to the experiment setup, let us discuss the meaning of all possible outcomes:

- **H + M > M** If also H + M > H holds, then we can say that there are truly mutual benefits of the cooperation.
- **H + M > H** Humans with machine assistance play better than those without.
- **H + M = M** Humans do not improve the level of play of machines.
- **H + M = H** Humans do not make any quantifiable positive use of the machine assistance.
- **H + M < M** Humans not only do not benefit from machine assistance, but also degrades the machine playing performance.
- **H + M < H** The machine assistance is deceptive to humans thus decreasing their quality of play.

V. Pilot Studies

This section reports on the results of two pilot studies that we have run to refine our experimental setup as well as to gather preliminary evidence regarding the research hypothesis. In particular, we present a technical setup and provide the profile of the human players participating in the experiment.

Since a well-played game is time consuming, we limited the number of games a single person can play to three. The experiment was performed separately for each human subject, so no information could be exchanged in the process, e.g., looking how other people play. The program participants used to play, and the opponent program were run on the same computer, both having access to two physical CPU cores. We set the *play clock* for the two machines (the cooperator and adversary) to 30 seconds in the first pilot study and 9 seconds in the second one. In order to avoid time-outs resulting from the human player, we discarded the concept of random moves if a player fails to respond in time. The matches were played only during weekdays anytime from the morning to the late afternoon. The age of participants varied from 21 to 30 with only one exception of 31 to 40. Most of them were PhD students of computer science.

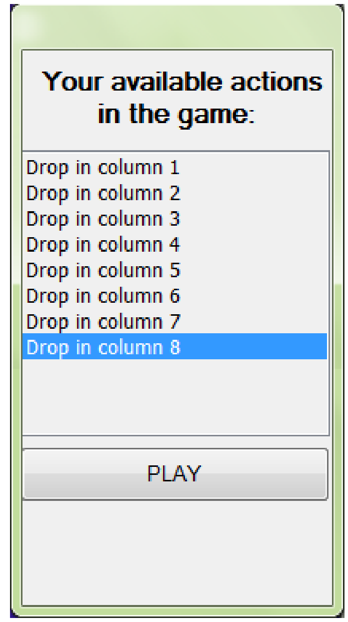

Figure 3. The window used to choose an action during a game.

In the experiment, three games were used but one of them, namely Tic-Tac-Chess, was discarded after the Pilot Study 1 as it appeared to be too biased in favor of the player making the first move. Descriptions as well as screenshots of all the games can be found in the Appendix. Figures 7, 8, and 9 show screenshots of the program operated by participants for Inverted Pentago, Nine Board Tic-Tac-Toe and Tic-Tac-Chess respectively, the three games played in Pilot Studies.

The user interface consisted of three windows: the action window, the board window and the cooperation window. The action window is presented in Figure 3. It displays moves available to the participant and contains a button for making an actual (i.e., not simulated) move in the game. The board window presents the current game state (board) - please refer to the Appendix for some examples. Finally, the cooperation window, depicted in Figure 4, provides the statistics from simulations and enables the cooperation options which were discussed in Section III. The only difference, in terms of the user interface, between cooperative and non-cooperative games is that in the latter case, the cooperation window is not shown.

A. Pilot Study 1

We gathered 6 human participants for the first pilot study. They were divided into two groups of 3 people each. These two groups formed our two samples of data: playing with machine assistance (H+M) and without (H). During the experiment, we started each game with a short training session. We also gave participants a transcript explaining what they are asked to do and how the user-interface works. When participants

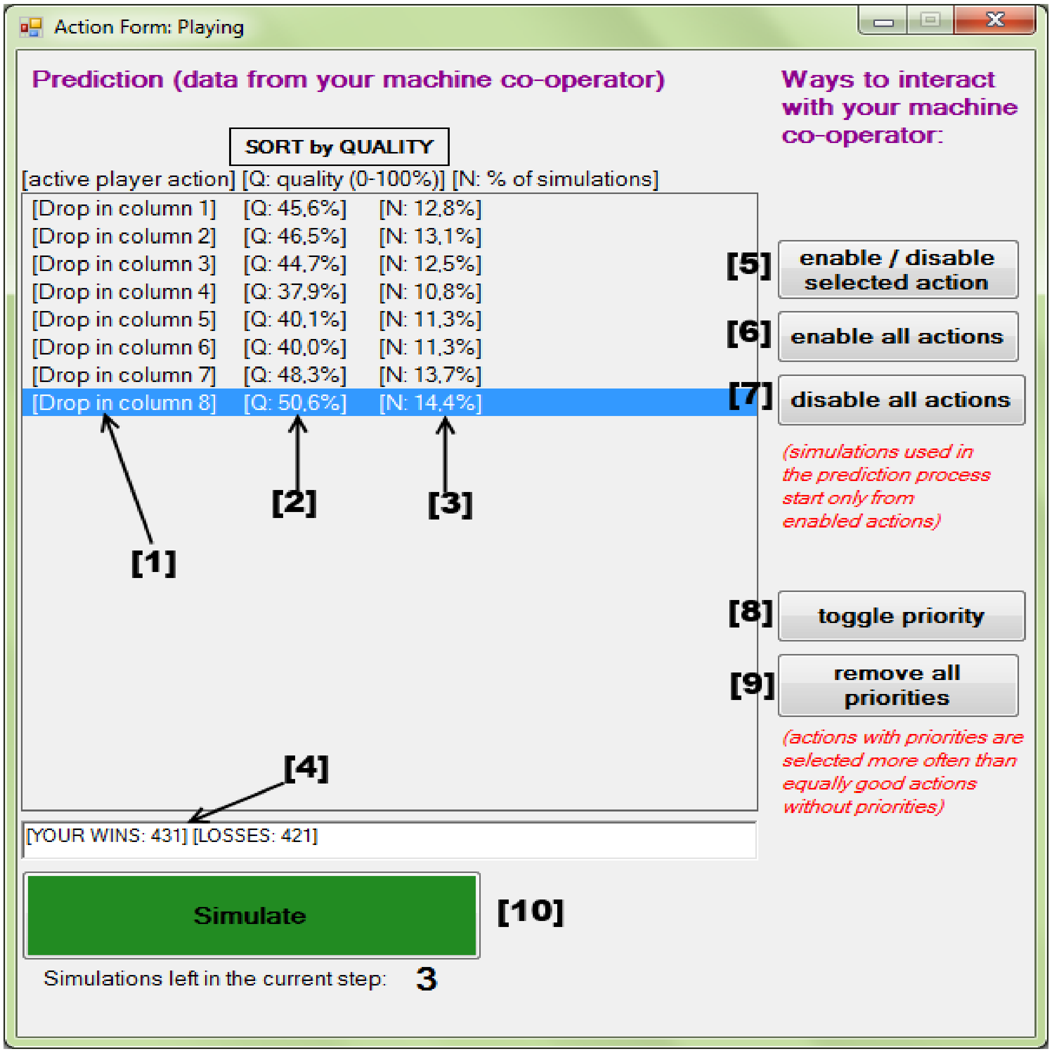

Figure 4. The window human players used to cooperate with the machine player. [1] - action name; [2] - the quality of an action (Q) computed by the MCTS/UCT algorithm, shown as a percentage chance of winning the game by the human player; [3] - the percentage of simulations (relative to the total number of simulations), in which a particular action was investigated; [4] - this field, after selecting an action, it shows the number of simulations which ended with wins, draws and losses, respectively, from the human player's perspective; [5][6][7] - an interface for disabling and enabling actions to be used in the simulations; [8][9] - an interface for managing priorities of actions (to make them used more frequently than they would be used otherwise, i.e. without priority settings); [10] - when participants were satisfied with the selection of the enabled actions and priority assignments, they would click this button to run a batch of simulations.

were ready, they started playing a serious (i.e., not training) game and when they finished all three matches they were asked to complete a short questionnaire to obtain a profile of the subjects. The assignment of human players to games was based on the *Latin square design* [29] with 3 games, 6 participants and two playing modes, i.e., with machine assistance being switched ON or OFF. Using this design, the minimum required number of participants for a full experiment is 12, but in the pilot study we stopped at 6 participants.

B. Pilot Study 2

At this point, we decided to revisit the experimental setup slightly and continue the experiment, called pilot study 2, to mitigate some problems that arose. Instead of asking people to play each game once, we asked them to play one game three times in order to enable learning by experience. The first match played includes a training session. The training session was extended to be a full match to let participants learn from their mistakes in endgames (late phases), which are often the most tricky to play. It is also often the case that people learn how to play better from the way they lost.

As mentioned above, we excluded Tic-Tac-Chess from the set of games for giving too much advantage to the first player to have a turn. As a consequence, each subject lost their match very quickly in the same way leaving us with no relevant data to work on. Although there exist certain strategies to avoid a quick loss, it is unlikely to be seen by players unfamiliar with the game. Having only one type of game per participant, we modified the players' assignment in such way that we have all combinations of participants playing at least one of the three consecutive matches with the co-operation of the machine.

In order to deal with the problem of long experiments, which was mainly caused by the simulation time needed to get meaningful results, we decided to write highly-optimized dedicated interpreters for rules of the chosen games. We were able to reduce the *play clock* just to 9 seconds.

C. Results of Pilot Studies

Numerical outcomes and human players' behavior during the experiments allowed us to make the following observations:

- The score between samples is even.
- All games appear to be very demanding for participants.
- There were no wins for Inverted Pentago and for Tic-Tac-Chess (discarded in the pilot study 2). There were 2 wins for Nine Board Tic-Tac-Toe, one with the cooperation and one without.
- The main reason for poor performance as specified by subjects in the questionnaire and communicated right after the end of the experiment was the lack of experience in playing given games. The rotations in Pentago were commonly mentioned as something being particularly difficult.
- Despite understanding the role of the program and the advice provided to them, the participants often seemed not to have desire to cooperate. If they had an assumption about which action was the best, they just opted to play it instead of investing time for more simulations.
- The participants seemed to enjoy playing the game but some stress was caused by the level of difficulty and the expectation to win.

Figure 5 shows the average scores (0 meaning loss, 50 meaning draw and 100 meaning victory) obtained by the co-operating participants (H+M) and non-cooperating participants (H) against the machine in Inverted Pentago whereas Figure 6 shows the same data for Nine Board Tic-Tac-Toe. Vertical error bars denote 95% confidence intervals. The X axis denotes game step (ply). The error bars overlap so the results cannot be used yet to formally verify the hypothesis.

There were not enough participants in the pilot study to make any statistically significant claims. However, the trend so far is that the participants who did not cooperate played slightly better average games. This is reflected in the **H vs M** curve, starting from step 10, being above the **H + M vs M** one. However, both curves eventually meet at a common point, which means that the average game results of both samples are even and equal to zero (which means a loss). The same properties are valid in the Nine Board Tic-Tac-Toe game. Because in the pilot studies, the participants rarely and quite chaotically used the cooperation possibilities, a conclusion that cooperation does not help would be an overstatement. The sample is too small, the participants would use the provided statistics when already behind in the game and because the cooperation options were shown only every second move, the machine was not able to help with a coherent line of actions.

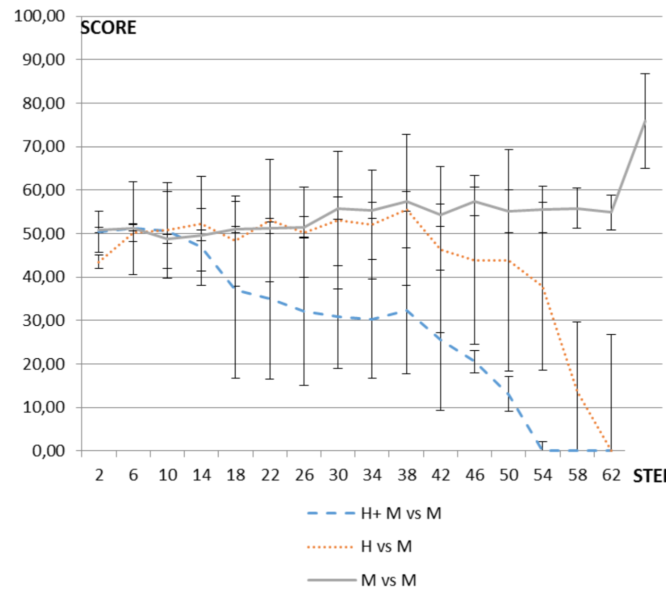

Figure 5. Graph showing the average scores obtained by the cooperating participants (H+M) and not cooperating participants (H) against the machine in Inverted Pentago.

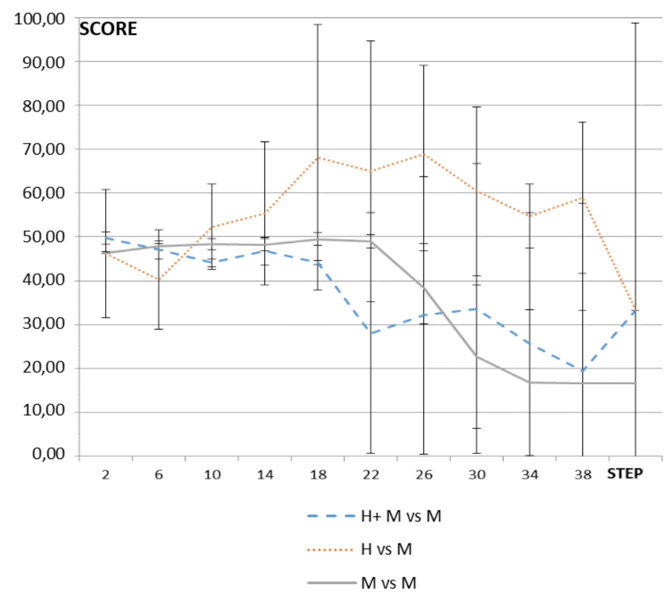

Figure 6. Graph showing the average scores obtained by the cooperating participants (H+M) and not cooperating participants (H) against the machine in Nine Board Tic-Tac-Toe.

VI. REFINED EXPERIMENT

A. Changes after Pilot Studies

Based on the lessons learnt from the pilot studies we have made a list of desirable changes to be introduced before moving to the final phase of the experiment. Among various observations the following four points seem to be the most relevant.

- In order to have a chance to observe any progress in playing, each subject should play a given game more than three times, preferably at least five. We have to make room for more learning possibilities as it turned out that three games were not enough to learn how to play previously unknown game well (e.g., Inverted Pentago or Nine Board Tic-Tac-Toe).

With more repeats we can also slightly reduce (though not eliminate) the effect of personal predispositions.

- The cooperation options should definitely be shown all the time for players playing with the help of a machine.
- Actions' priorities should be removed and only the mechanism of enabling and disabling actions should be left as the latter has more influence on the game tree and should be used more often. We have to make sure that all the participants understand why and when it is beneficial to disable actions.
- Participants should be asked to play two games with the machine cooperation in the middle (e.g., the second and the third ones) to be able to observe, in the remaining games, the effects of learning from those (supervised) games.

We analyzed the average outcomes of matches for the H + M vs. M and H vs. M samples of data as well as the average evaluation observed by the machine in every 4 steps of games. The 95% confidence intervals were computed using the t-student test. The results showed that the number of participants in the pilot study is not enough to make any significant claims regarding the hypothesis. Therefore, our plans shifted towards investigating how participants cooperate and whether they can learn the game faster by playing with machine assistance.

B. *Setup of the Experiment*

All the changes enumerated in the previous paragraph were pursued in the refined setup. In terms of the user interface two changes were introduced: firstly, the toggle priority button was removed (since it was very rarely used in previous experiments) and secondly, the statistics (the cooperation window) were shown permanently in the cooperation mode. Making the statistics always visible had been identified as a crucial requirement in order to support continuous cooperation loop. We have managed to gather 11 people who agreed to participate in three game sessions of our experiment. All of them were computer science or mathematics students, without any competitive backgrounds in games. None of them knew the games chosen for the experiment. Such a candidate profile, i.e., analytic mind capable of learning quickly unknown games, was deemed appropriate for our experiment. Ultimately, the attendance varied from 9 to 10 people at the same time. Consequently, we do not have full statistics for some people.

Each session was carried out in a computer lab starting by a detailed explanation of the experiment and rules of the current game to play. We ensured that there was no communication and information exchange between the subjects during the experiment.

Three games have been prepared, one for each session, i.e., Nine Board Tic-Tac-Toe in session 1, Cephalopod in session 2 (see Appendix for a description of this new game and Figure 10 for its visualization in our system) and Inverted Pentago in session 3. Each game has been tested for opening balance between the playing sides, i.e., whether one the players is favored over the other one because of them making move in the first/second turn. There are many ways how to check such a property and we decided to repeat hundreds of matches between two machine players and observe the average score.

TABLE I. Games used in respective sessions of the refined experiment. The *Participant* column denotes which player (the first turn or the second turn) participants played as.

Session	Game	Participant	Balance
1	Nine Board TTT	2nd	1st slightly favored
2	Cephalopod	2nd	Practically fair
3	Inverted Pentago	2nd	2nd slightly favored

The setup of games and outcomes related to balance were as shown in Table I.

During a session, we asked each subject to play at least five matches. The first five were mandatory meaning that the participants had agreed to play those beforehand. Each match after the fifth one was optional and some participants took this opportunity to play one or two more games.

In the first session, half of the participants played the second and the third game (out of the mandatory five) in the cooperation with the machine. The other matches were played without the cooperation.

In the second session, those who had not played with the cooperation before, were asked to play games two and three with the cooperation. The main idea was to observe whether any learning process occurs. We could compare whether the cooperating people play the respective games better than the non-cooperating ones as well as whether any effects of learning can be noticed in games played after the cooperation (game fourth and onward). We assumed that playing with the machine assistance at some point may speed-up the learning process.

Finally, in the third session all participants played games two and three in the cooperation in order to gather more samples in this mode.

The average game lasted about 35 minutes and the average five-game session took about 3 hours. We expected this time to be long enough to build up some experience in playing particular games.

VII. Results of the Refined Experiment

We present an overview of how each game ended in Tables II, III, and IV for Nine Board Tic-Tac-Toe (session 1), Cephalopod (session 2) and Inverted Pentago (session 3), respectively. The first thing to notice is that players with the machine assistance, therefore, theoretically on a favored position, did not win any match of Nine Board Tic-Tac-Toe whereas there have been wins for players who did not cooperate. Though, nobody was able to win more than two matches.

The only three games won by participants in Cephalopod happened to be in the cooperation mode, however, only in the first assisted games. Nobody out of those three winners was able to repeat a victory, even in the cooperation mode. Since the second assisted games were carried out using the same setup as in the first ones, we consider this "concentration of wins" as a pure coincidence. Overall, this game has proven to be very difficult although it is relatively unbiased when it comes to starting position balance.

In the third session, with Inverted Pentago as the game of choice, participants were playing slightly favored roles, i.e., second to go. In such a setup, in majority of the played moves,

TABLE II. Results of matches of Nine Board Tic-Tac-Toe. In the first column, there are identifiers of participants used cohesively for all games. The respective G_i column denotes i-th played game. 100 denotes a win of the participant, 50 denotes a draw, whereas 0 denotes a loss. Scores with stars to them were obtained in games played with the cooperation of the machine player.

Subject	G_1	G_2	G_3	G_4	G_5	G_6	G_7
1	0	0*	0*	0	0	-	-
2	0	0*	0*	0	0	-	-
3	0	0*	0*	0	0	-	-
4	0	0*	0*	0	0	-	-
5	0	0*	0*	0	0	-	-
6	0	0	100	0	0	-	-
7	100	100	0	0	0	-	-
8	100	0	0	100	0	-	-
9	0	0	0	0	0	0	100
10	-	-	-	-	-	-	-
11	0	100	0	0	0	0	-

TABLE III. Results of matches of Cephalopod. Please refer to Table II for the cells' interpretation.

Subject	G_1	G_2	G_3	G_4	G_5	G_6	G_7
1	0	0	0	0	0	-	-
2	0	0	0	0	0	-	-
3	0	0	0	0	0	-	-
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0*	0*	0	0	-	-
7	0	0*	0*	0	0	-	-
8	0	100*	0*	0	0	-	-
9	0	100*	0*	0	0	-	-
10	0	100*	0*	0	0	-	-
11	-	-	-	-	-	-	-

TABLE IV. Results of matches of Inverted Pentago. Please refer to Table II for the cells' interpretation.

Subject	G_1	G_2	G_3	G_4	G_5	G_6	G_7
1	0	100*	50*	0	0	-	-
2	0	50*	50*	0	0	-	-
3	0	100*	0*	0	0	-	-
4	0	50*	50*	0	0	-	-
5	0	100*	100*	0	0	-	-
6	-	-	-	-	-	-	-
7	0	100*	0*	0	0	-	-
8	0	50*	100*	0	0	-	-
9	0	50*	100*	0	0	-	-
10	0	100*	50*	0	0	-	-
11	-	-	-	-	-	-	-

one does not have to do anything more to win than just to follow the machine cooperator suggestion. The only won or drawn games indeed happened to be in the cooperation mode. The main reason for that was the strength of the machine cooperator. When not played in the cooperation mode, all games were lost.

Analysis of the results shows that the cooperation is significantly helpful in Inverted Pentago. It may be beneficial for Cephalopod as well, but the sample is not big enough to make such a claim. What definitely stands out is that no learning patterns can be observed. Players who won (or drew) once were not able to repeat this result convincingly.

In order to facilitate the detailed analysis of results let us introduce a concept of the rank of a move by means of the machine evaluation, i.e., based on the average score Q computed by the UCT algorithm using Equation (1). The rank of 1 means that the move is the best according to the machine, i.e., has the highest Q. The rank of 2 denotes the second best (the second highest Q) and so on. Those ranks were logged for each played move in all matches, even in games without the cooperation because the machine still computed all the statistics without showing them to players. The average and median ranks were then computed for a match and next aggregated for all matches with and without the cooperation.

Tables V and VI show some aggregated statistics gathered separately for games played with and without cooperation. It could be seen from Table V that players were generally choosing moves that were highly ranked in the cooperation mode, especially in Nine Board TTT and Inverted Pentago. Please recall that the *MedRank* and *AvgRank* indicators are the averages (over subjects) from median and average ranks, respectively, based on moves played in each game. Both these indicators are significantly lower, with 95% confidence, than their counterparts computed for matches without cooperation.

These average scores show that lower ranks do not necessarily translate into better outcomes since the game-related starting bias should also be taken into consideration when analysing these move-ranking statistics. Since the machine opponent is slightly favored at the beginning of Nine Board Tic-Tac-Toe games, a human player should disagree with the machine evaluation, at least at some point, to give themselves a chance to turn the game into their favor. If they do not, the machine playing the slightly favored role against itself has a very high chance for winning. That is why there have been more wins in Inverted Pentago, in which the situation is the opposite.

Table VI presents how many times (in game steps) the chosen move was within the 15% margin of the best move by means of machine evaluation, in which case we call it a *good move*. The average number of game steps is included for reference. We also introduce a concept of a mistake which occurs when the played move is evaluated at least 20% worse than the previously played one. As shown in Table VI, the average number of such mistakes per game varied from 0.3 (Inverted Pentago, cooperation) to 7.8 (Cephalopod, no cooperation), which is - most probably - the main reason for so many losses in this game.

The notion of good moves and mistakes is more accurate for Cephalopod and Inverted Pentago since, as stated above, the machine's assessment of moves is higher in these games than in Nine Board Tic-Tac-Toe, which favors the opponent's role. Nevertheless, in NB TTT, the number of mistakes is significantly higher in the cooperative mode, which means that subjects chose to disagree with the machine helper at some point, but eventually lost all the games, anyway.

Another statistic shows that the cooperative games took more steps to finish what may suggest that they were more balanced until the game reached the conclusion phase.

It is worth underlying that participants used the cooperation options actively, when playing in this mode, almost in every step of every game. The differences in median and average move ranks, which can be read from Table V, suggest that participants also actively used the information provided by the machine player. However, without asking participants explicitly whether a particular play was a result of the machine suggestion, we cannot be sure whether this claim is true. On the

TABLE V. Selected averaged statistics grouped for each game for both cooperation modes. *MedRank* and *AvgRank* denote median and average ranks of moves taken by players, respectively. The ranking is defined by the machine evaluation of moves, e.g., the highest evaluated move has the rank equal to 1. These values are averaged among all matches played for the respective (game, cooperation mode) pair, therefore medians can be non-integer numbers. The last column denotes the average score from 0 (loss) to 100 (win). For each value, the 95% confidence intervals are shown in brackets.

Game	MedRank		AvgRank		AvgScore	
	Coop	NoCoop	Coop	NoCoop	Coop	NoCoop
NB TTT	1.2 *(± 0.2)*	2.3 *(± 0.3)*	1.7 *(± 0.3)*	2.6 *(± 0.2)*	0 *(± 0.0)*	16.3 *(± 11.1)*
Cephalopod	1.3 *(± 0.3)*	3.5 *(± 0.3)*	2.3 *(± 0.6)*	5.1 *(± 0.2)*	30 *(± 29.9)*	0 *(± 0.0)*
Inv. Pentago	1.1 *(± 0.2)*	5.1 *(± 0.4)*	2.0 *(± 0.4)*	6.8 *(± 0.5)*	66.7 *(± 15.8)*	0 *(± 0.0)*

TABLE VI. Selected averaged statistics grouped for each game and for both cooperation modes. *AvgLength* denotes the average number of steps required to complete the games. The *GoodMoves* column denotes the number of steps in which participants played a move with the machine evaluation not lower than the highest evaluated move by more than 15%. The last column denotes the number of steps in which the played move had at least 20% lower evaluation than the previously played one (likely to be a mistake leading to a loss). For each value, the 95% confidence intervals are shown in brackets.

Game	AvgLength		GoodMoves		Mistakes	
	Coop	NoCoop	Coop	NoCoop	Coop	NoCoop
NB TTT	19.1 *(± 1.3)*	17.5 *(± 0.9)*	18.1 *(± 1.3)*	16.5 *(± 1.0)*	3.5 *(± 0.6)*	1.8 *(± 0.3)*
Cephalopod	39.8 *(± 2.4)*	37.2 *(± 0.9)*	37.8 *(± 2.9)*	30.7 *(± 1.1)*	5.0 *(± 2.2)*	7.8 *(± 0.6)*
Inv. Pentago	35.9 *(± 0.1)*	30.8 *(± 1.7)*	35.8 *(± 0.2)*	27.5 *(± 1.9)*	0.3 *(± 0.3)*	3.0 *(± 0.9)*

other hand, posing such a question after each move would have been very disturbing for them, so we had refrained ourselves from that option.

VIII. CONCLUSIONS AND DIRECTIONS FOR FUTURE RESEARCH

In this paper, the MCTS/UCT algorithm is used as a tool for introducing cooperation between humans and machines during strategic interaction. The particular embodiment of such a cooperation is General Game Playing framework, which is represented in our experimental setup and used in the experiments conducted with human subjects. The MCTS/UCT algorithm has a desirable property that it is able to exploit the raw computational power of machines by running a continuous simulation-based assessment of game states. The method can be regarded as a machine learning approach. The algorithm can, in principle, be steered by the human input to improve its learning rate, although, we have not been able as yet to convincingly justify this claim.

Balancing the proper games difficulty is crucial to this kind of experiment. Certainly, for the cooperation to make sense, humans should not dominate over machines. Furthermore, on the one hand, games have to be difficult enough for humans so as human players should consider statistics computed by the machine as a valuable source of knowledge about the game. On the other hand, however, games cannot be too difficult because unexperienced humans will lose in the vast majority of matches played against the machine without gaining any real experience/knowledge about how to play the game, at least in the non-cooperation mode.

The initial goal of our experiment was to verify the H + M > M and H + M > H hypotheses. It turned out that proving any of these two claims with statistical significance requires many more human participants and much more data. Additionally, successful cooperation between human and machine does not necessarily have to be measured solely by means of the average score achieved in played games. One of the interesting factors indirectly supporting the existence of such cooperation could be, for instance, the observable existence of the learning process emerging in the proposed human-machine cooperation loop.

When looking at the games played in cooperation with the machine player in Nine Board Tic-Tac-Toe, Cephalopod and Inverted Pentago, one can see three different activity patterns. In Nine Board Tic-Tac-Toe, the machine assistance does not really help. In Cephalopod, it has some positive impact, because the only games won by participants involved cooperation. In the last game, people played significantly better with machine assistance. However, in all three cases, no effects of learning could actually be spotted. Even in the case of Inverted Pentago, where many of the games in the cooperation mode ended up well, none of the participants was capable to win or even draw afterwards when played without cooperation. Major differences in the results shapes suggest that the hypotheses H + M > M and H + M > H are very sensitive to the choice of game and, therefore, hard to prove or disprove in general case. On a general note, the results show that experimental outcomes which involve human subjects are often inconsistent and drawing any firm conclusion requires much more time and effort than in the case of machine-based experiments.

Our future plans are concentrated on revisiting the experimental environment to make it more learning-friendly for the human subjects. It seems that the basic statistics of actions which are provided to them are not sufficient to *really understand and learn how to play the game*. The participants should learn game-related concepts and patterns and memorize more from the played games than they do in the current setup. They should be able to understand why certain actions are good or bad. In order to learn how to play a game well, the participants probably need more time per session and more sessions separated by a one-two day breaks (i.e., organized in different days). Apparently, it is hard for humans to learn how to play a new non-trivial game during one session only.

Our conclusions, which revolve around the importance of the proper selection of games, in a natural way lead towards the next research steps. It is very likely that not only "pure" difficulty but also other characteristics of games may be decisive in building a successful cooperation framework. Investigation of such features across various games and measuring the extent to which they affect the effectiveness of cooperation is one of the crucial objectives of our future research. The problem is multi-dimensional where "pure" game difficulty is only one of the dimensions. The game difficulty, fairness and interestingness may be measured analytically, e.g., by analyzing the number and the complexity of game rules or game states, or experimentally, based on the outcomes of games with either human players, or specifically designed artificial agents [30]. A similar problem was approached in [31] from the automatic game design perspective, where the authors proposed 57 computable characteristic features of games such as uncertainty, drama, lead change, permanence, completion, duration, momentum, etc. While the goal of [31] was to find *interesting* games, we believe that similar research principles apply in the quest for *cooperation-friendly* games.

Appendix
Games Descriptions

Inverted Pentago is a game played on a 6x6 board divided into four 3x3 sub-boards (or quadrants). Taking turns, the two players place a marble of their color (either red or blue) onto an unoccupied space on the board, and then rotate any one of the sub-boards by 90 degrees either clockwise or anti-clockwise. A player wins by making their opponent get five of their marbles in a vertical, horizontal or diagonal row (either before or after the sub-board rotation in their move). If all 36 spaces on the board are occupied without a row of five being formed then the game is a draw. Participants play as blue and are the second player to have a turn.

Nine Board Tic-Tac-Toe. In nine board tic-tac-toe, nine 3x3 tic-tac-toe boards are arranged in a 3x3 grid. Participants play as 'O' and are the second player to have a turn. The first player may place a piece on any board; all moves afterwards are placed in the empty spaces on the board corresponding to the square of the previous move. For example, if a piece was placed were in the upper-right square of a board, the next move would take place on the upper-right board. If a player cannot place a piece because the indicated board is full, the next piece may be placed on any board. Victory is attained by getting 3 in a row on any board. If all boards are full without any player having a line-of-three, then the game ends with a draw.

Tic-Tac-Chess is a game played on a 7x7 board. Players start with one piece marked by a red or blue square in their respective starting location. Participants are the second player to have a turn. The starting locations are outside the movable area of the board, which is defined by the inner 5x5 square. On their turn, each player may move a piece as though it were a Chess knight or capture with a piece as though it were a Chess king. Capturing is possible only with pieces belonging to the center 5x5 square. Pieces from the starting locations do not disappear when moved, so moving a piece from the starting location effectively spawns a new one on a destination square. The first player to get three pieces in a row, column, or diagonal in the center 3x3 square wins. Participants play as blue and are the second player to have a turn.

Cephalopod is a game played on a 5x5 board. Taking turns, the two players place a die of their own colour (either white or black) onto an empty cell on the board. Immediately after the die is placed, the capturing conditions are checked and enforced if apply. The placement is capturing if and only if: (1) - the currently added die is horizontally or vertically adjacent (so no more than four neighbours are possible) to at least two dice and (2) - the sum of the pip counts on those adjacent dice is six or less. In both cases it does not matter what colours the adjacent dice are. If the die placement is capturing, then all the horizontally and vertically adjacent dice are removed from the board and the new dice is set to show the sum of the pip counts of the captured dice. Capturing are mandatory only when placing a die onto a square where capturing conditions apply. The game terminates when the board is full. The player who controls more dice wins. Draws and ties are impossible. Participants play as black and are the second player to have a turn.

Acknowledgment

M. Świechowski would like to thank the Foundation for Polish Science under International Projects in Intelligent Computing (MPD) and The European Union within the Innovative Economy Operational Programme and European Regional Development Fund. The research was financed by the National Science Centre in Poland, grant number DEC-2012/07/B/ST6/01527. This initial studies and the pilot experiments were performed while Maciej Świechowski and Jacek Mańdziuk were visiting UNSW Canberra. The ethics approval number granted from the university is A14-09.

References

[1] M. Świechowski, K. Merrick, J. Mańdziuk, and H. Abbas, "Human-Machine Cooperation in General Game Playing," in The Eighth International Conference on Advances in Computer-Human Interactions (ACHI 2015). IARIA, 2015, pp. 96–100.

[2] M. R. Genesereth, N. Love, and B. Pell, "General Game Playing: Overview of the AAAI Competition," AI Magazine, vol. 26, no. 2, 2005, pp. 62–72.

[3] J. Mańdziuk, "Towards Cognitively Plausible Game Playing Systems," IEEE Computational Intelligence Magazine, vol. 6, no. 2, 2011, pp. 38–51.

[4] ——, "Computational intelligence in mind games," in Challenges for Computational Intelligence, ser. Studies in Computational Intelligence, W. Duch and J. Mańdziuk, Ed. Springer Berlin Heidelberg, 2007, vol. 63, pp. 407–442.

[5] "Stanford General Game Playing," 2014, URL: http://games.stanford.edu/ [accessed: 2015-11-25].

[6] C. P. Hoc J-M. and H. E., Eds., Expertise and Technology: Cognition & Human-computer Cooperation. Psychology Press, 2013.

[7] M. Argyle, Cooperation (Psychology Revivals): The Basis of Sociability. Routledge, 2013.

[8] K. R. Olson and E. S. Spelke, "Foundations of Cooperation in Young Children," Cognition, vol. 108, no. 1, 2008, pp. 222–231.

[9] S. Gelly et al., "The Grand Challenge of Computer Go: Monte Carlo Tree Search and Extensions," Commun. ACM, vol. 55, no. 3, Mar. 2012, pp. 106–113, DOI: 10.1145/2093548.2093574.

[10] O. Syed and A. Syed, Arimaa - A New Game Designed to be Difficult for Computers. Institute for Knowledge and Agent Technology, 2003, vol. 26, no. 2.

[11] A. Kolobov, Mausam, and D. S. Weld, "LRTDP versus UCT for online probabilistic planning," in Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012.

[12] T. Keller and P. Eyerich, "PROST: Probabilistic Planning Based on UCT," in Proceedings of International Conference on Automated Planning and Scheduling, 2012.

[13] Z. Feldman and C. Domshlak, "On Monte-Carlo Tree Search: To MC or to DP?" in Proceedings of ECAI-14. 21st European Conference on Artificial Intelligence, 2014.

[14] J. Karwowski and J. Mańdziuk, "A New Approach to Security Games," in International Conference on Artificial Intelligence and Soft Computing (ICAISC'2015), ser. Lecture Notes in Artificial Intelligence. Springer-Verlag, 2015, vol. 9120, pp. 402–411.

[15] K. Walędzik, J. Mańdziuk, and S. Zadrożny, "Proactive and Reactive Risk-Aware Project Scheduling," in 2nd IEEE Symposium on Computational Intelligence for Human-Like Intelligence (CHILI'2014), Orlando, FL. IEEE Press, 2014, pp. 94–101.

[16] K. Waledzik, J. Mańdziuk, and S. Zadrożny, "Risk-Aware Project Scheduling for Projects with Varied Risk Levels," in 3rd IEEE Symposium on Computational Intelligence for Human-like Intelligence (CIHLI'2015). IEEE Press, 2015, p. *in press*.

[17] A. Klinger, Human-Machine Interactive Systems. Springer Science & Business Media, 2013.

[18] B. Stevens and F. Lewis, Aircraft Control and Simulation. New York: Wiley, 1992, ISBN: 0-471-61397-5.

[19] Y. H. Yin, L. Da Xu, Z. Bi, H. Chen, and C. Zhou, "A novel human–machine collaborative interface for aero-engine pipe routing," Industrial Informatics, IEEE Transactions on, vol. 9, no. 4, 2013, pp. 2187–2199.

[20] C. G. Eden, "Robotically Assisted Surgery," BJU International, vol. 95, no. 6, 2005, pp. 908–909.

[21] M. Buro, "The Evolution of Strong Othello Programs," in Entertainment Computing, ser. The International Federation for Information Processing, R. Nakatsu and J. Hoshino, Eds. Springer US, 2003, vol. 112, pp. 81–88.

[22] F.-H. Hsu, Behind Deep Blue: Building the Computer that Defeated the World Chess Champion. Princeton, NJ, USA: Princeton University Press, 2002.

[23] J. Mańdziuk, Knowledge-Free and Learning-Based Methods in Intelligenet Game Playing, ser. Studies in Computational Intelligence. Berlin, Heidelberg: Springer-Verlag, 2010, vol. 276.

[24] N. Love, T. Hinrichs, D. Haley, E. Schkufza, and M. Genesereth, "General Game Playing: Game Description Language specification," 2008. [Online]. Available: http://logic.stanford.edu/classes/cs227/2013/readings/gdl_spec.pdf [accessed: 2015-11-25]

[25] L. Kocsis and C. Szepesvári, "Bandit Based Monte-Carlo Planning," in Proceedings of the 17th European conference on Machine Learning, ser. ECML'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 282–293.

[26] M. Świechowski and J. Mańdziuk, "Self-Adaptation of Playing Strategies in General Game Playing," IEEE Transactions on Computational Intelligence and AI in Games, vol. 6, no. 4, Dec 2014, pp. 367–381.

[27] M. Świechowski, J. Mańdziuk, and Y.-S. Ong, "Specialization of a UCT-based General Game Playing Program to Single-Player Games," IEEE Transactions on Computational Intelligence and AI in Games, 2015, DOI: 10.1109/TCIAIG.2015.2391232 *(accepted for publication)*.

[28] M. Świechowski and J. Mańdziuk, "Fast Interpreter for Logical Reasoning in General Game Playing," Journal of Logic and Computation, 2014, DOI: 10.1093/logcom/exu058.

[29] R. A. Bailey, Design of Comparative Experiments. Cambridge University Press, 2008, vol. 25.

[30] A. Jaffe, A. Miller, E. Andersen, Y.-E. Liu, A. Karlin, and Z. Popovic, "Evaluating Competitive Game Balance with Restricted Play," in AIIDE, 2012.

[31] C. Browne and F. Maire, "Evolutionary Game Design," Computational Intelligence and AI in Games, IEEE Transactions on, vol. 2, no. 1, 2010, pp. 1–16.

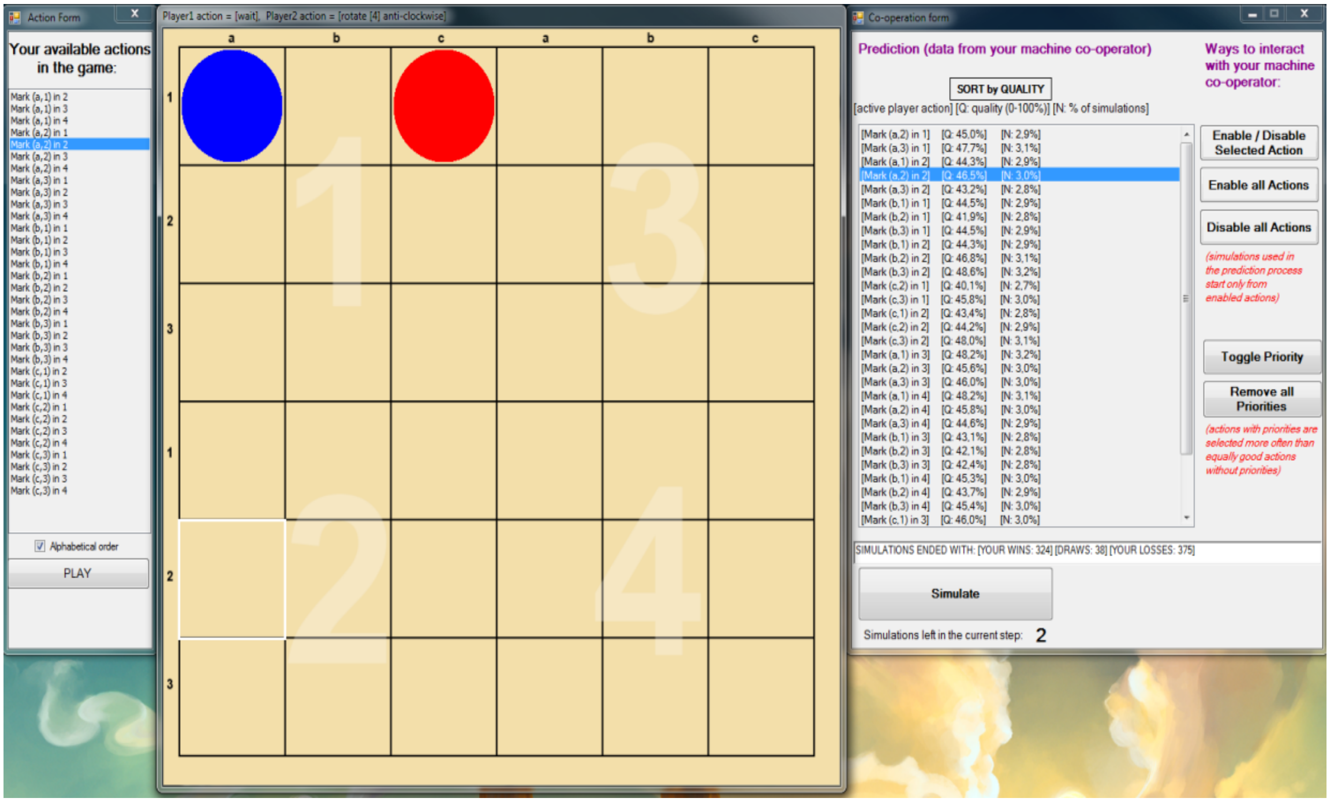

Figure 7. Screenshot of a program used to play Inverted Pentago on Windows 7 operating system (version with the cooperation).

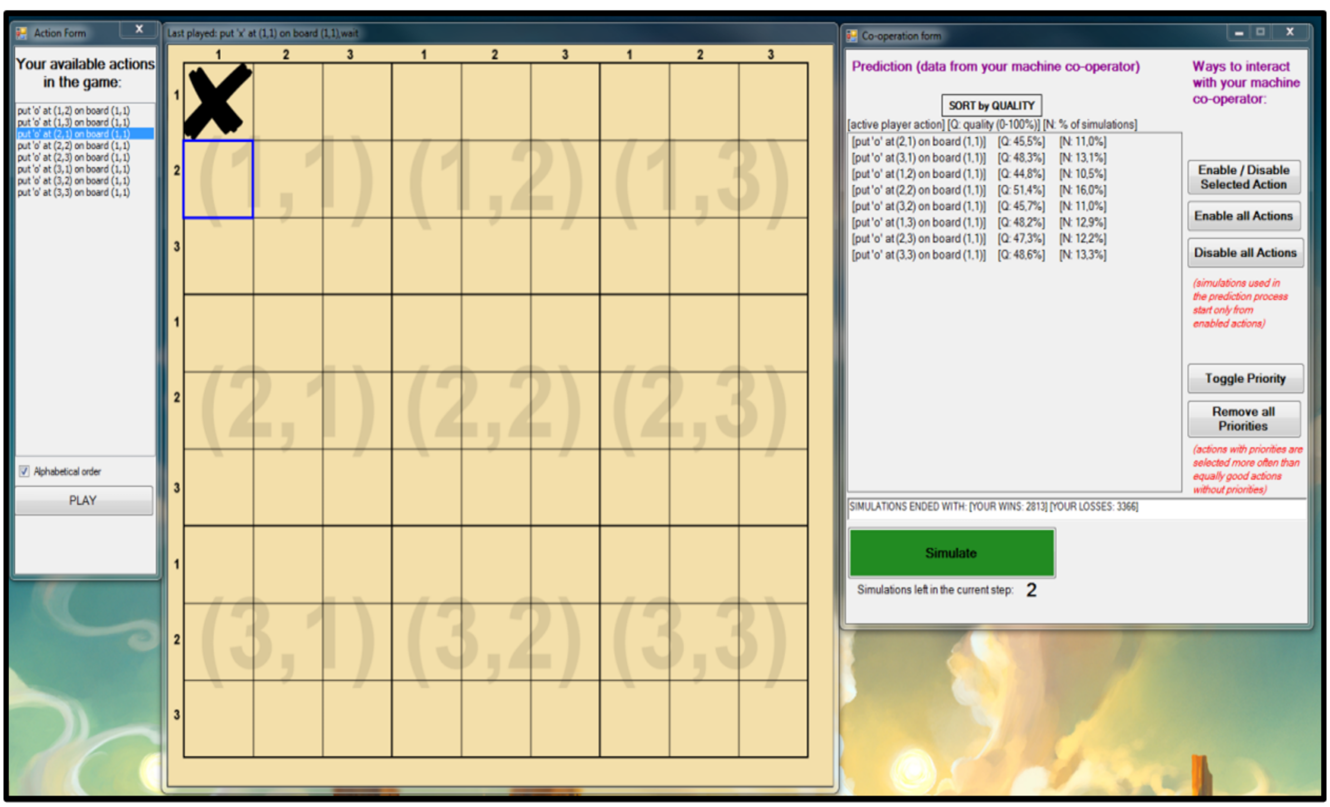

Figure 8. Screenshot of a program used to play Nine Board Tic-Tac-Toe on Windows 7 operating system (version with the cooperation).

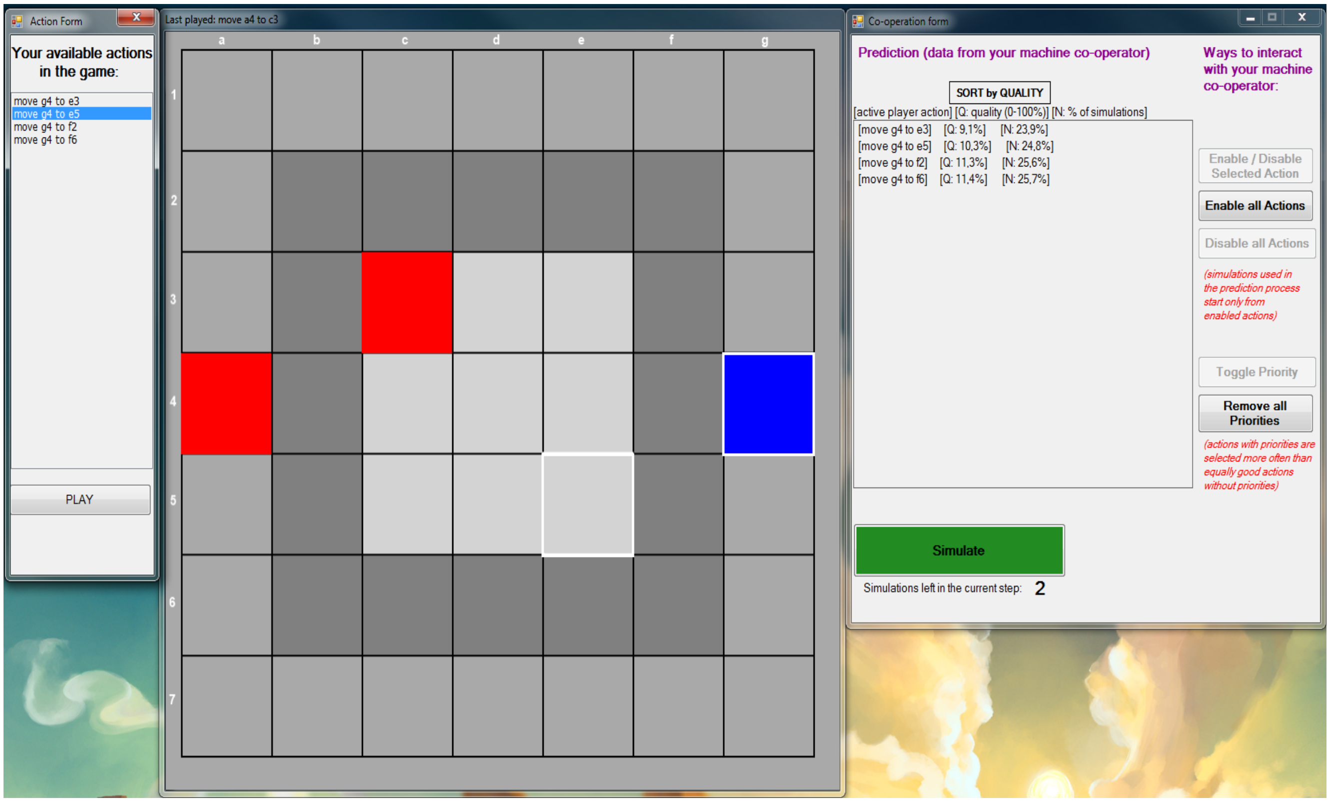

Figure 9. Screenshot of a program used to play Tic-Tac-Chess on Windows 7 operating system (version with the cooperation).

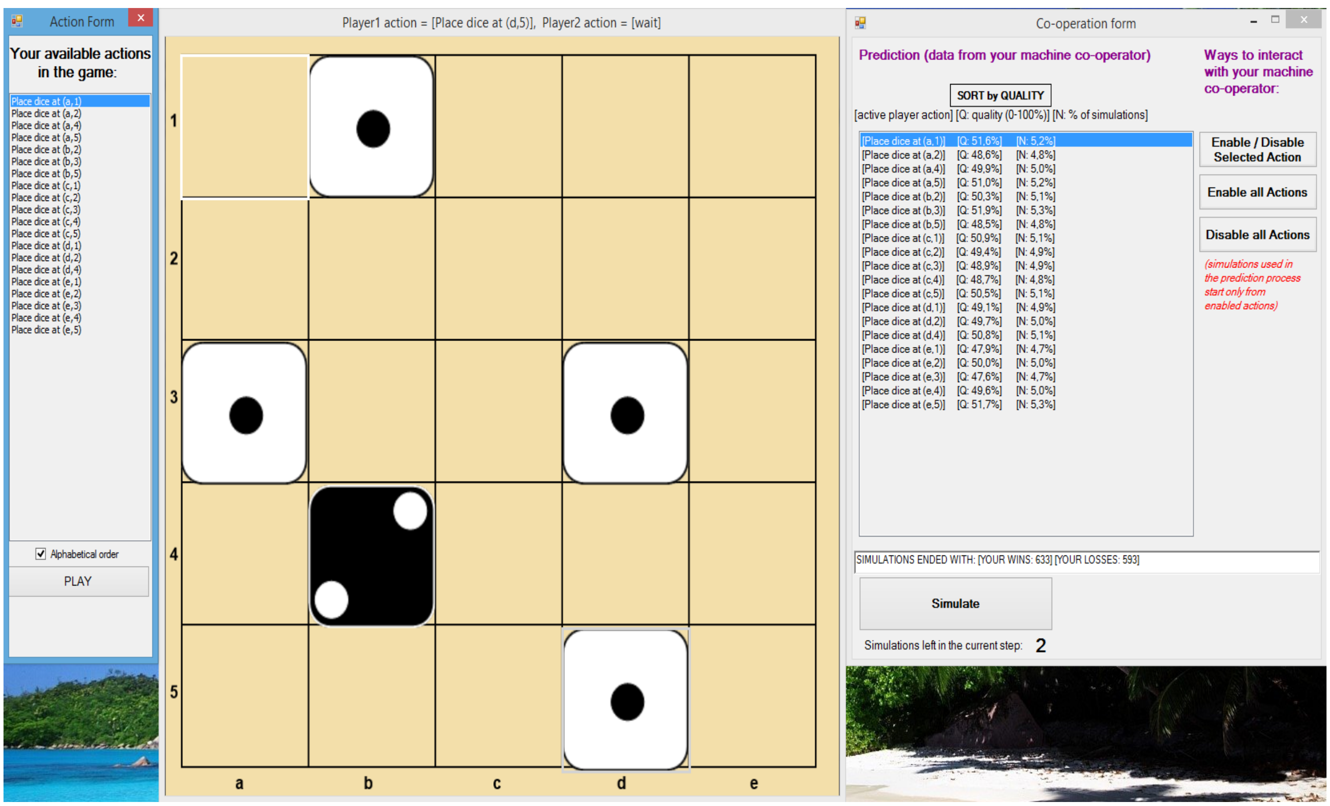

Figure 10. Screenshot of a program used to play Cephalopod on Windows 8 operating system (version with the cooperation).