Multi-agent Card Game: UNO

*This is the project work for CENG 561 Artificial Intelligence Course

1st Arkin Yilmaz
Computer Engineering Dept.
Middle East Technical University
Ankara, Turkey
2460822

2nd Mert Akkor
Electrical and Electronics Engineering Dept.
Middle East Technical University
Ankara, Turkey
2475549

***Abstract*—This project work tries to implement a well-known card game called UNO by using artificial intelligence (AI). The paper consists of four sections. Introduction section provides first provides a general information about the games and then explains the UNO card game, including the rules and how it is played. Related Work provides a survey and related work about our research, including different game types and different decision mechanisms. Third section is the Problem Definition and Algorithm section which gives the structure of the system and the implementation of the decision mechanisms. The following section provides results and discussions of the experiments. Final section provides conclusions of the research and future work recommendations.**

***Index Terms*—artificial intelligence, game playing, multi-agent systems**

I. Introduction

Humans cannot live without interaction with each other. Sometimes this led to disaster or chaos but mostly it helped people to improve and made strong bonds. Since old times, humans created many games to have fun with each other.

UNO is a very popular game that everybody can play. It is very straight forward and easy to understand. There are 108 cards in a deck and every player starts with seven cards. The rest of the cards are placed in a Draw Pile faced down. When top card in this pile placed to Discard Pile game starts. The first player is normally chosen randomly, and gameplay follow a clockwise direction. Every player looks at their cards and try to match them with the one in Discard Pile. Player either have to match by the number, color, there are red, blue, green and yellow, or the 'Action Card'. As an example, if Discard Pile has a blue card that is a 5, player have to place either blue card or a card with a 5 on it. If the player has a no match or choose not to play any of the cards even though they have a match, they must draw from the Draw Pile. If that card is playable, play it or keep it and game moves on to the next player. The game continues until a player has one card left. The moment a player has just one hard they have to say 'UNO'. If they do not say it until the next player's turn, they get penalized by drawing two cards. Once a player has no cards, game is over. To understand what the most optimal and logical gameplay will be would be, we made our friend play without telling them what to do, and additionally we played with algorithms that we think it will do good.

II. Related Work

Games are natural models for reactive systems. Even tough, there are countless games, all of them are under four main title. These are perfect information and deterministic, imperfect information and deterministic, perfect information and not deterministic(chance) and lastly imperfect information and chance games. A game has perfect information if each player, when making any decision, is perfectly informed of all the events that have previously occurred, including the 'initialization event' of the game(e.g. the starting hand of each player in a card game) [4]. As opposite, if players do not have all the information, it is imperfect information game. For determinism, each player must know the outcomes of the actions made.

TABLE I
Types of Games

	Deterministic	**Non-Deterministic (Chance)**
Perfect Information	Chess, Go, Checkers	Backgammon, monopoly
Imperfect Information	UNO, Battleship, blind Tic-tac-toe	Bridge, Poker, Scrabble

First, perfect information and deterministic games. As stated earlier, in this game type each player can fully observe the environment and they know their actions outcome. Most common examples to this type are chess and GO, since each player can see all the pieces on the board at all times and when a player makes a move, they both know its effect and outcome. Since there is no chance factor and secrets, it is the easiest type to implement.

Secondly, perfect information and chance games. These types of games are widely discussed in literature since chance factor can change the state of the game. This type includes games like Backgammon or Monopoly, which although they have random events (dice rolls), they still do not have any information which is known to one player but not another. (The probabilities of all random events must be known to all players.) [5]. As a result, these games are put into perfect information games but differentiated with chance factor.

Third game type is imperfect information and not deterministic games. Unlike our second type, perfect information and not deterministic, in this type players are simply unaware of the actions chosen by other players. An example for this type can be almost any card game. Players do not know their hands and they do not know how their played cards will affect the game, like Texas hold'em game.

Lastly, imperfect information and deterministic games. Example for this type can be given as Battleship game or blind Tic Tac Toe. However, proposed UNO Card game is different than the other games. First, as the number of players increase, its complexity increases exponentially. This makes it hard to evaluate all the game tree and find an optimal path in game. Because of the game being really dependent on chance factor, drawing every round or the card that previous player threw, makes competitive UNO intractable [1]. So, looking from another perspective, an algorithm has been developed to play this game which discussed in the next chapter.

III. Problem Definition

A. Task Definition

UNO is an example of deterministic game. Each played card leads to completely predictable outcomes. Unlike other deterministic games, UNO is an imperfect information game. Players can not see the others hand and the course of events depends on the chance factor. Since there is a chance factor, it is very hard to generate the search space. Therefore, Minimax algorithm can not be used for this game. Instead of generating the seach space, it is logical to cut off search earlier. Without considering the chance factor, a player can play at most N different cards by looking the last played card, where N is the number of cards in players hand. Since the number is not so big, it is decided to cut off the search earlier. The following subsection explains the proposed evaluation function and algorithm.

B. Algorithm Definition

In order to create a good evaluation function, first the objectives of a player are analyzed:

- Playing the card which maximizes the amount of potential moves in the next round,
- Playing the card which has the maximum point,
- Playing the card which prevents the next player to win

Even if the above objectives are almost same for each player, the priority list of the objectives varies. To understand what is the most optimal and logical play strategy would be, different scenarios are created and asked people what they are going to play. By using the majority voting, the priority list is decided and the evaluation function is created. For each card in player's hand, two different values are computed. The first value corresponds to the amount of potential moves in the next round if that card is played and the second value corresponds to the total score of remaining cards in the hand. The evaluation function decides as follows:

> "If there is a card which maximizes the amount of potential moves in the next round, play it. If there is an equality, break the tie by playing the card which minimizes the remaining hand score."

By using cut-off search and the above heuristic evaluation function, the computer decides which card to play very quickly and makes the move. The following figure shows an example decision situation:

Fig. 1. Example decision situation

From the figure, it can be seen that there is a 'BLUE 3' in the middle. Therefore, player has to play either a blue card or a matching card. In this situation, the player has four possibilities to play: 'GREEN 3', 'BLUE 1','RED 3', and 'BLUE 1'. To decide, player first has to look for possible moves in the next rounds, assuming the color of the middle card is not changed.

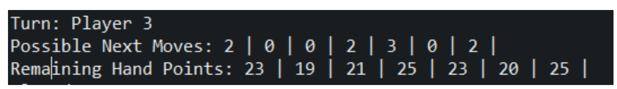

Fig. 2. Example evaluation function

From the evaluation function, it is found that the best option to play for the proposed strategy is, to play 'RED 3' because later in the next rounds, the player can play two more cards from their hand assuming the color in the middle is not changed. However, if there was an equality, the player should play the card which has the biggest value. Assuming that the player does not have 'RED 3', the player would play 'GREEN 3' since it brings more points than 'BLUE 1'.

IV. Experimental Evaluation

A. Methodology

Since UNO does not have any specific game plan and depends on chance, first main objectives of the game have to be listed in importance order. Most important thing in this game is to finish your hand earlier than your opponents, by playing the card which maximizes the amount of potential moves in the next round, playing the card which has the maximum value and playing the card which prevents the next player to win. If these are not an option and another player is finishing earlier, aim is to lower the total sum of number on the cards. When prioritization is made, proposed algorithm is decided by asking to some of our near friends and also what would we do in some situations. As an example, we played with them and observed what would they do; if most cards with same color are played first or least color card will be played first. Some similar applications are made with all random selections and machine learning [3]. However, this is an AI project and not machine learning so in the scope of some articles and help of some friends, we developed this project. Even though it plays with the same algorithm every time, without machine learning, it has high success rate compared to other all random similar applications.

Also, for evaluation metrics, we used memory usage, average time of play of bots and competitiveness.

B. Results

First, project is started with two players, and algorithm is applied to them. After receiving convenient results, project moved on to four player UNO game. After implementing algorithm properly, next step was to design the interface. For visual pleasure and user-friendly interface, everything is used very simple and actual pictures of real UNO cards are used.

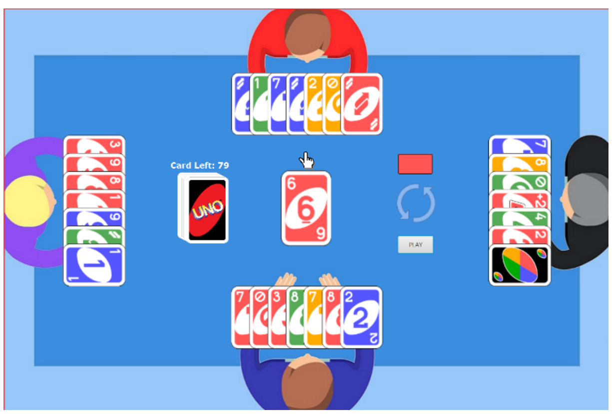

Fig. 3. Example opening status of the game

To understand if the algorithm works correct and to show it more properly, opponents' (right player2, up player3, left player4) cards are also open. Current card color, direction of the game and play button for bots are located on the right side of middle the card. To draw a card, you need to click to Draw Pile located on the left side of the middle card. Also, current player's turn is shown with finger.

As it can be seen from the Figure 5, player cannot win every time, bots can also win very frequently. To test win rate, we played with two different algorithms. First, the proposed one which you discard the most frequent color with the highest value, if you have same frequency between two colors, you also discard the highest value card. Second algorithm is the one which you play the least frequent one from your hand and save the frequent colors, equivalence rule is also same for this one. After playing 100 games with each algorithm, clearly proposed algorithm is better with 36% win rate by winning 36 games out of 100, where second algorithm has only 20% win

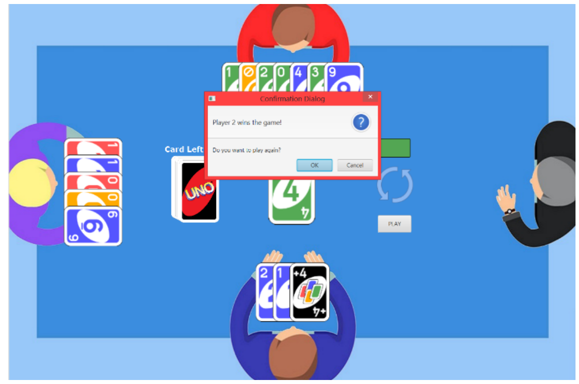

Fig. 4. Example of an endgame situation

rate by winning 20 games out of 100. It is clear that, proposed algorithm works better than the second algorithm.

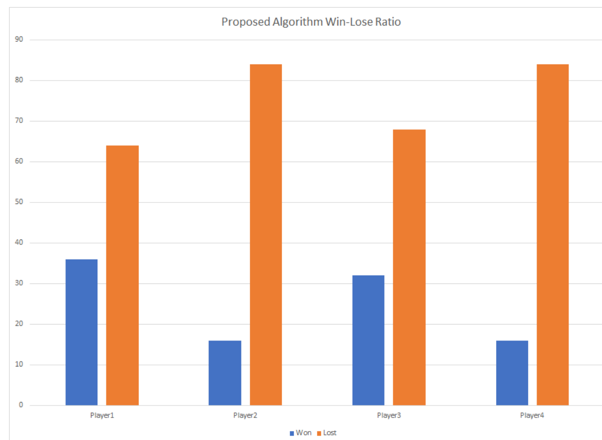

Fig. 5. Example of an endgame situation

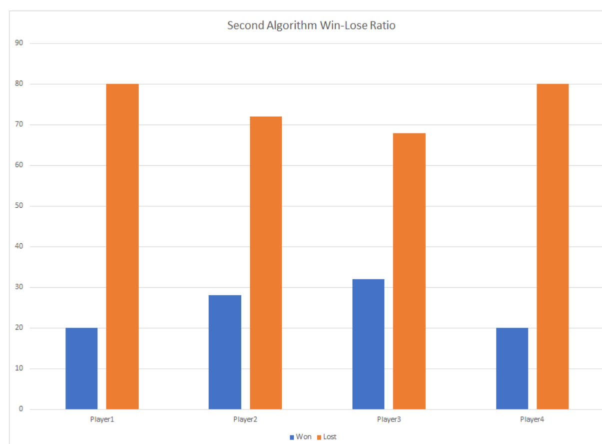

Fig. 6. Example of an endgame situation

C. Discussion

Since UNO is very hard game to make, many different multi-agent games are developed. Similar application is made

[6] but they do not design and develop an algorithm, they just compute UNO's complexity and computing the necessary calculations to make certain player win or make certain player lose. It has been showed that several variants of these problems are difficult, but a restricted single-player version are solvable in polynomial time. On the contrary, it has showed that the uncooperative two-player version is in P. This is somehow surprising in the sense that multi-players' version usually become more intractable than single player's one in many games. [1] In other games, compared to our algorithm, since they are using machine learning as well, their results are much more accurate than ours. Bots winning almost every time against the player.

For our evaluation metrics, we are using very low memory, linear with card size. Since we are just holding 108 card information memory complexity is pretty low. Secondly, average time of play of bots. Bots are making calculations only considering their cards in hand. There are average of 5 cards during one game so calculations and play time of bots are very fast that we made a button to make them play one by one, otherwise we couldn't follow their moves. Lastly, competitiveness measure. As stated before, number of wins are close to each in 100 games, meaning that everyone can win in both algorithm, which show us there exists competitiveness.

V. Conclusion

As it stated before, there are four type of games. Perfect information deterministic, perfect information non-deterministic, imperfect information deterministic and imperfect information non-deterministic. It is found that, algorithms for such as Minimax and Alpha-Beta Pruning for two reasons: the search space is huge and there is a chance factor. Therefore, it is concluded that, there has to be a cut-off search with intelligent evaluation function. UNO game is in type of imperfect information and deterministic games. Proposed algorithm in this paper is much more promising than the second algorithm that we have tested. Our results compared to other one is slightly better. Achieving close number of game winnings shows that UNO is mainly dependent on normally distributed probability chance factor. Even using an unlikely game plan can win 20 games out of 100 against a well-thought game plan. As a final word, project promises a lot of improvement and plenty of good work with pleasant visuals and strong game plan algorithm.

VI. Future Work

In the project, only an algorithm is used to select cards and play them with certain rules. In feature, this can be improved by adding machine learning, as well as some randomness and environment awareness, where algorithm also considers player's number of cards, better use of Action and Wild card, like to prevent someone finishing by throwing Draw Four Wild Card. This can also be achieved by assigning weighted probabilities for each objective. By doing so, the algorithm becomes richer such that players do not follow the same strategy throughout the game. Also, combining different strategies may lead interesting results. Likewise, understand if some player's last card is not matching with middle card's color, try not to play around same color to prevent him finishing. Additionally, the current implementation of the game supports for four players. This could be increased up to eight players. Also for now, there is only one real player. This could also be increased to support multi players. So, with proper improvements in the future this can be great UNO game.

References

[1] D. M. H. N. U. R. U. T. Demaine, E. and Y. Uno, "Uno is hard, even for a single player," *Theoretical Computer Science*, vol. 295, pp. 51–61, 2014.

[2] O. T. S. T. V. Veronique Ventos, Yves Costel, "Boosting a bridge artificial intelligence," 2017.

[3] S. J. Johansson, "On using multi-agent systems in playing board games," *International Conference on Autonomous Agents*, pp. 569–576, 2006.

[4] J. Mycielski, "Games with perfect information," *Handbook of Game Theory with Economic Applications*, vol. 1, pp. 41–70, 1992.

[5] "Complete vs perfect infomation in combinatorial game theory," *Stack Exchange*, 2014.

[6] E. D. R.A. Hearn, "Games, puzzles, and computation," *Handbook of Game Theory with Economic Applications*, 2009.