A Browser-based Interface for the Exploration and Evaluation of Hanabi AIs

Extended Abstract

Markus Eger
Principles of Expressive Machines Lab
NC State University
Raleigh, NC
meger@ncsu.edu

Chris Martens
Principles of Expressive Machines Lab
NC State University
Raleigh, NC
crmarten@ncsu.edu

ABSTRACT

The card game Hanabi is partially observable, cooperative and strictly limits how players can communicate. This makes it an interesting target for AI researchers, and as such various different AIs for it have been developed. We developed our own AI based on Grice's maxims of communication, and implemented a browser-based version of the game to have human players play with a number of different AIs. As a result of this experiment we have over 2000 logs of players interacting with 3 different AIs, that are publicly available. To make exploring these game logs easier and enable others to investigate AI behavior we present a modified version of our implementation which can be used to step through the games and display AI information and participant information. We believe the data set itself is interesting for future research, and that this user interface enables an easier way to interact with it, and that the implementation provides a good framework for the development of new AIs.

CCS CONCEPTS

•**Computing methodologies** → *Reasoning about belief and knowledge;*

KEYWORDS

Hanabi, Game AI, Data Set

ACM Reference format:
Markus Eger and Chris Martens. 2017. A Browser-based Interface for the Exploration and Evaluation of Hanabi AIs. In *Proceedings of Foundations of Digital Games, Cape Cod, MA, USA, August 2017 (FDG2017),* 4 pages.
DOI: 10.475/123_4

1 INTRODUCTION

Cooperative games with partially observable information and limited communication present an interesting challenge for AI research [7]. One such game is the award-winning [6], cooperative card game Hanabi [1]. In Hanabi, players cooperate to build fireworks, represented by cards in five colors and ranks 1 to 5. Players are dealt hands of 4 or 5 cards, depending on the number of players, that they hold with the cards' faces away from themselves, such that every player sees every other player's cards, but not their own. The goal of the game is to play the cards of each color in ascending order onto five stacks on the table. On their turn, players have to either play a card, give a hint or discard a card. Playing a card requires the player to choose a card from their hand, and if it is the next card in ascending order for the stack corresponding to its color, it is placed on top of that stack. Otherwise the card is removed from the game and a mistake is noted. To give a hint, a player tells another player about all cards in their hand that have a particular color or a particular rank, such as telling them which cards are all their red cards, or all their 3s. Giving a hint comes at the expense of a hint token, of which there are initially eight. Discarding a card removes that card from the game, but regenerates one hint token, if there are currently fewer than eight available. After playing or discarding a card, the player draws a replacement card from the deck. Play proceeds until either 3 mistakes in total have been made by the players, or the deck runs out of cards, plus one extra turn for each player. The score at the end of the game is equal to the number of cards that were successfully played, for a maximum score of 25. Figure 1 shows a typical game state during a 2 player game of Hanabi.

Figure 1: A typical game of Hanabi (Source: BoardAgain Games)

Hanabi is an interesting challenge for AI research because it involves communicating information using a limited communication channel, which requires deciding which hints to give and interpreting hints that are received from other players. Several AIs have been developed that play the game reasonably well using different approaches to this communication problem. Cox et al view Hanabi as a hat guessing game, which allows their AI to pass information to each of four other players by giving a single hint [2]. However, this scheme only works when all players are using it, which is particularly a problem when playing with human players. Osawa presented several AIs that are more in line with how humans play the game [5]. His agents follow a predetermined heuristic which prefers playing cards that are known to be playable over discarding cards that are known to be discardable, which it prefers over giving hints about playable cards, which it in turn prefers over giving hints about random cards. This approach works reasonably well in two-player games with another AI cooperator of various types. One limitation of this approach is that, while it follows the general outline of how humans play Hanabi, human players expect goal-directed behavior, which can not be observed when the AI gives random hints. Furthermore, the hints that are given often violate Grice's maxims [4], which state, among other things, that communication is expected to be relevant and unambiguous. In a previous project, we have developed an AI that is based on one of the AIs presented by Osawa, called the Outer State Strategy, but improves upon it by adopting intentions when deciding which hints to give [3].

Our tech demo consists of two major parts: One part is our implementation of the game, with several AIs, including implementations of Osawa's AIs and our own. The implementation lets users play games in a standard web browser, but also offers capabilities to replay previous games and obtain additional information about the AI's behavior. The second major part of our tech demo consists of data that we collected for an experiment in which we compared our AIs with Osawa's Outer State AI when human players play with each of them. Participants were asked if they agreed to have their game logs and survey answers made publicly available, and we obtained over 2000 game logs from 240 participants this way that we can share with the research community. We believe that these two parts together are a great basis for future research, by providing a framework for the development of new AIs, an environment to test and improve these AIs as well as the means to evaluate them and data to compare to and/or learn from.

To see how our demo works in practice, it can be accessed online under **https://go.ncsu.edu/hanabi**. We have also recorded a short video demonstrating it in practice, which can be found online under https://go.ncsu.edu/hanabivideo.

2 HANABI IMPLEMENTATION

We implemented Hanabi to be playable in a web browser to be able to develop and test different AIs. Our system consists of two main components: An implementation of the game rules of Hanabi including multiple AIs, and a user interface component that presents the game state to the user on an HTML page. The latter component can also be used to watch replays of previously completed games. The complete code for our implementation is available on github [1].

2.1 Game play and AI

Our implementation of the game play itself is straightforward. The game keeps track of the deck, the board, cards that have been discarded and players' hands. Additionally, for each card in each player's hand it tracks what that player knows about that card, as a list of possible values. Each of these lists starts out containing all 25 possible card values, but will be updated according to the hints the player receives. For example, if a player is told that a card in their hand is red, that list will only contain (*Red*, 1) through (*Red*, 5). We keep track of this information in the game object because it is required across many different types of AIs.

Players in the game are represented by objects that support what we call the *player protocol*. This protocol consists of two methods: One, `get_action` that is called when it is that player's turn, and that should return what that player's action is on their turn, and another `inform` that is called to tell each player which actions are performed by the players. Different AI agents can be developed easily by implementing these two methods. Our code base includes several different AIs to demonstrate this:

- One AI performs random actions every turn
- Implementations of all AIs developed by Osawa [5]
- A version of Osawa's self-recognition player more suitable for real-time use, by drawing random samples of possible hands until a time limit is reached, rather than enumerating all possibilities
- An AIs developed by us, based on Osawa's Outer State AI, but more explicitly goal-direct and based on Grice's maxims of communication
- Another AI developed by us that improves on the goal-directed AI by attempting to guess intentions behind hints it receives
- One "AI" that actually writes the game state to the console and asks the user to decide which action to perform

The last AI on this list can be used to play the game with AI cooperators on the console. Because this mode of interaction is cumbersome and not very intuitive, we have subsequently implemented a graphical version that displays the game state in a browser window and allows humans to play this way.

2.2 GUI and Replay viewer

Our GUI implementation currently supports a two-player game with one human player and one AI player. It is implemented as a web server that presents an HTML page that allows users to start, play and record, and subsequently watch games. When playing a game, the server keeps the game object in memory and renders an HTML page representing the current state of the board, with available actions for the player represented by links on the cards. When the user clicks such a link, for example to play a card, the web server constructs the appropriate action and advances the game by two turns: One for the human player with the action they selected and one for the AI player, and then returns the updated state. To make it easier for players to track what changed, any card affected by these two turns is outlined in red.

The game will log all actions performed by players in a file associated with the game. Instead of playing a game, users can also select such a previously recorded game and replay it. This selection can be filtered by AI type, score reached or deck used, to make finding a replay of interest easier. The replay itself is presented in the same way as the actual game, with the addition of controls in the bottom left corner that allow users to move one full turn

[1] https://github.com/yawgmoth/pyhanabi

Card	W 4	B 4	W 5	Y 5	G 1
Intention	Optional	Play	Keep	Play	Discard
Hint Yellow	Keep	Keep	Keep	Play	Keep
Hint White	Play	Keep	Play	Keep	Keep
Hint Green	Keep	Keep	Keep	Keep	Discard
Hint Blue	Keep	Play	Keep	Keep	Keep
...					

Table 1: Partial explanation our AI gives for the action performed in Figure 2.

forward and backward, meaning one action by each player. Figure 2 shows a screenshot of watching a replay. As can be seen, when watching the replay, the cards can be interacted with to perform an action, which allows users to take control of the game from any state that they currently see. When they choose to do so, they can either use the same AI that was used in the recording of the replay or choose a different one. This makes it easy to test how different AIs behave in the same game state.

Another feature of our GUI that aids with the development and evaluation of new AIs is the option for AIs to provide explanations for their decisions. After every AI move, the developer can click on an *Explain* link, that will open a window showing the information provided by the AI. The AIs we developed use this feature to display tables with estimated probabilities for the cards in their hand, what actions they want the human player to perform, and their prediction for what would have happened had they performed another action. For example, Table 1 shows how our AI would explain its action in Figure 2, by showing which cards the human player has in their hand (with abbreviated color names), which intentions the AI has for what the human player should do with each of these cards, and for each possible hint it could give, a prediction for what the human player might do with each card. In this case, the AI predicts that, if it hints the human player about all their yellow cards, they will play the Yellow 5, which matches the intentions of the AI, but if the AI hints the human player about all their white cards, they would play either the white 4 or the white 5, neither of which matches the AI's intentions. The AI will then perform one of the actions that match its intentions. When running an experiment with human subjects to compare different AI agents, this functionality could of course provide an unfair advantage to the player, and can be disabled for such applications.

Finally, when we used the GUI for an experiment in which human players played with different AIs, we also asked the participants some basic demographic information and to rate their experience with the AI. Because one of the use cases for the GUI is the exploration of the data set we obtained, it also provides a link to see the survey responses of the participant, if available.

3 THE DATA SET

To test how well our AI does compared to Osawa's when playing with human players, we ran an experiment in which participants played with different AIs. Log files for all games played during this experiment were recorded and participants could choose to make them publicly available. The complete data set can be found on github [2].

3.1 Experiment design

In our experiment, every participant was asked to play one game with an AI chosen at random from Osawa's Outer State AI and our two intentional AIs. Because the initial order of the deck can have an effect on how hard it is to get a high score, we also randomly assigned every participant one of five possible orderings of cards in the deck by setting the seed of the random number generator to 1 through 5 before shuffling the deck. To simulate real game play, players did not have access to the "explain" functionality described above during the experiment. At the conclusion of the game we told players which score they obtained, according to the game rules, and then asked them some basic demographic information, including age range, familiarity with board games in general and Hanabi in particular, recency of play and an estimate on how often they reach the maximum score. They were also asked to rate the AI in terms of skill, goal-directness and how much they enjoyed playing with it. After answering the survey, participants were presented with the option of playing additional games, that were also recorded. For these additional games, participants would get a truly random deck, and a newly randomly assigned AI from the three options.

3.2 Available data

Of all participants that finished at least one game, 240 agreed with the publication of their game logs and survey answers. Because many of these players played several games, we have over 2000 game logs available for analysis. The data set presents an opportunity for analysis beyond the results discussed in our previous paper [3]. Each game game log contain the seed for the random number generator, which can be used to determine the order of the cards in the deck at the beginning of the game, as well as every action taken by every player. For example, games played with random seed 1 start with the AI player holding a green 4, red 5, 1 and 3 and a yellow 4, in that order. The order of cards in the deck and the player's hand can also be determined. Each action then represents a deterministic change of the game state, and they can be applied in order to see the game state after every turn. Our system demonstrates how this can be used to replay the game, but another way this information is useful is to identify interesting scenarios during game play, for example realistic end-games, and develop AIs specifically for these scenarios.

A major challenge in Hanabi is to determine how hints are interpreted by a human cooperator. Because our game logs contain the hint information, as well as how the human cooperator reacted to the hint, it could provide a basis to investigate this process. For example, by using machine learning techniques it would be possible to predict which action a human will perform after receiving a particular hint. This could then be used to improve upon our AI or use it as the basis for a new one. The data set also contains all demographic information obtained during the experiment, associated with each game log. This could be used in addition to the raw game log data to determine any correlation between a particular

[2]https://github.com/yawgmoth/HanabiData

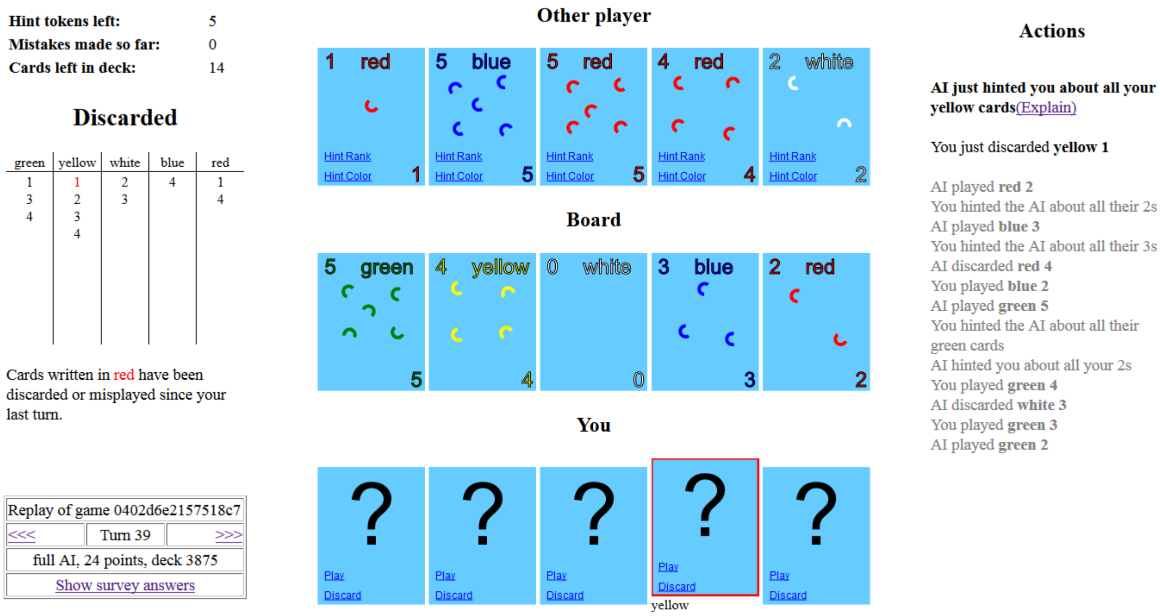

Figure 2: A screenshot of our UI being used to replay a recorded game. A full demo can be accessed under https://go.ncsu.edu/hanabi

behavior and player information, to e.g. develop AI agents geared towards playing with novices or experts.

4 EXTENSIBILITY

Our framework can significantly aid other researchers with the development of agents for Hanabi in multiple ways. Since it is written with extensibility in mind, adding a new agent is a matter of subclassing the `Player` class and implementing the desired behavior. We also provide several agent implementations to test a newly developed AI with, either as cooperators, or to have a baseline to compare to. Additionally, the "explain" functionality helps during development to have information about the AI's internal working available during game play. The take-over functionality also enables developers to refine AI behavior in certain game situations, such as the end-game, without the need to painstakingly recreate previous games. Finally, our framework also includes our entire experiment setup, including the surveys, and can be used to run similar studies with minimal setup time.

5 CONCLUSION

We presented our implementation of the cooperative card game Hanabi, which allows users to play the game with a variety of AIs in a web browser. Additionally, our GUI has the capability to watch replays of previously completed games, and to take over control of these replays at any point. Additionally, we have obtained over 2000 game logs for the use in an experiment that we can share with the research community to aid with the development of future AIs. We believe that Hanabi presents an interesting challenge for AI research and that our framework provides an ideal environment to develop, test and evaluate new AIs.

REFERENCES

[1] Antoine Bauza. 2010. Hanabi. (2010). https://boardgamegeek.com/boardgame/98778/hanabi

[2] Christopher Cox, Jessica De Silva, Philip Deorsey, Franklin HJ Kenter, Troy Retter, and Josh Tobin. 2015. How to make the perfect fireworks display: Two strategies for Hanabi. *Mathematics Magazine* 88, 5 (2015), 323–336.

[3] Markus Eger, Chris Martens, and Marcela Alfaro Córdoba. 2017. An Intentional AI for Hanabi. IEEE Conference on Computational Intelligence and Games (CIG), (2017).

[4] H Paul Grice. 1975. Logic and conversation. In *The Logic of Grammar*, G. Harman D. Davidson (Ed.). Dickenson, 64–75.

[5] Hirotaka Osawa. 2015. Solving Hanabi: Estimating Hands by Opponent's Actions in Cooperative Game with Incomplete Information. In *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence.*

[6] Spiel des Jahres. 2013. Spiel des Jahres Award. (2013). http://www.spieldesjahres.de/en/hanabi

[7] Piers R Williams, Diego Perez-Liebana, and Simon M Lucas. 2016. *Cooperative Games with Partial Observability.* Technical Report. IGGI.