

Project Proposal: Dark Hex

Main Features

I have designed and implemented a framework for running games of Dark Hex between various agents. It handles interactions between two agents, tracking the information of both players and the game state itself. The server has a variety of settings and is highly configurable to test various rule sets and agents. There are of course the standard settings for hex-like games, board dimensions and time limits. However for Dark Hex, there are a variety of different rules that can be considered in playing a game. For example, how does one deal with a time limit on moves. Do you set a time limit for the game or have a time limit for each move? If there is a collision, do you reset the timer or does the turn time limit remain? My server can handle all of these possibilities with configurable time settings and time enforcement. To deal win imbalance, one could want to start with pieces. Do both players get to know the initial board state, or simply how many pieces are on the board somewhere? My server can handle any initial board state and information for each player. As well, it has various settings for how much information should be sent as feedback to the terminal itself, allowing for human versus machine practice games. A central goal was to provide significant customization to be able to test various configurations for Dark Hex with various agents and I think this goal was achieved.

Agent Interface

To connect to the server, an agent must be presented as a Python3 class with a single required function, “genmove”. The server will send to an agent the current board state (to the knowledge of the player), the amount of “missing” stones from the board, how much time they have left, and the “status” of the game. If a player responds with a move that is currently occupied by the opponent, the server will once again query the genmove function but with the updated board state information and with the status notification that they have uncovered a hidden enemy piece. While this server requires a Python3 class, this can be easily extended to other programs through a small compatibility function. There are four players included in the repository, a “Human” agent that just queries the terminal for input, a simple randomized agent, and a simulation based player. The simulation player simulates several random games, and tries to correct for the missing information by guessing the positions of the unknown enemy pieces and the simulating games after that point. Also included is an interface for Mo-

hex, the University of Alberta Hex Research Group's premier standard Hex agent. This was tested with the Mohex version [here](#). Experimentation with this Mohex agent that just tries to play Hex on the Dark Hex board shows the issues transitioning from native Hex to Dark Hex. Mohex can easily play perfectly on small board standard Hex but it can't consistently beat the simple simulation opponent in Dark Hex since it assumes that cells are always open, even when there could be a hidden opponent piece that blocks a connection. Overall, this interface is effective for allowing a variety of agents to interact and experiment with different strategies.

Future Improvements

There are two components that I would like to continue on in the future but was not able to at this time. The first addition would be a graphical interface for Dark Hex. I had attempted to work out a way to integrate the Dark Hex server into HexGui but it proved too complex an addition to be accomplished within the time I had for this project. I don't have the skills currently to make a useful graphical interface beyond the text based terminal but in the future a proper GUI would be a nice feature for the server. The second addition would be an endgame solver for Dark Hex. The imperfect information aspect of the game makes a conventional solver more restricted in what it can solve at all and exceptionally slow. Given more time, building towards a solver that can find forced wins even with unknown pieces would be an interesting and challenging addition to the tools of this project.

Final Thoughts

The core of the project was to build an interface and host to handle the non-trivial task of running the game of Dark Hex. That aspect was clearly accomplished with a highly configurable server and an interface that can be adapted for many different types of agents. There is a lot more that could have been done but there is always time to continue in the future.

Github

<https://github.com/josephmeleshko/CMPUT497-Dark-Hex-Project>