Final Project Write-up: Bayesian Network based AI for BANG!

The Game:

BANG is a 4 to 7 player card game where each player is randomly given a role. In a 4 player game, there are 2 Outlaws, 1 Renegade, and one Sheriff. In a 6 player game, the role of Deputy is added, and in a 7 player game, there is an additional Outlaw. With the exception of the sheriff, all player roles remain a secret until they are eliminated, or the game ends. The game ends when either the Sheriff or all the Outlaws and the Renegade are eliminated. The goal of the Sheriff is to eliminate the Outlaws. The goal of the Deputy is to protect the Sheriff and eliminate the Outlaws. The goal of the Outlaws is to eliminate the Sheriff. The goal of the Renegade is to first eliminate the Outlaws, and Deputy if present, then eliminate the Sheriff. The deck consists of 22 different types of cards. “BANG!” cards allow you to shoot another player if they are in range of your gun and “MISSED!” cards can be played to protect against shots. There are also status cards, such as the mustang, which increases your distance, or beer, which restores your health. The real complication comes from being able to play as many cards from your hand as you want during your turn, but at the end of your turn, the number of cards in your hand must be no greater than your remaining health points.

The Problem:

The problem we wished to tackle, was the creation of an AI player that will competently play a game of BANG.

Our Approach:

Our approach to solving this problem, was to implement a Bayesian Network. A Bayesian Network is a probabilistic graphical model, formed from a directed acyclic graph which represents a set of variables and their conditional dependencies. We chose this method because BANG is a game of incomplete information. The information is incomplete because all roles, excepting the Sheriff, remain hidden until the player is eliminated. With roles hidden, the AI cannot know for certain the goals of the other players, excepting the Sheriff.

The language we chose to use for this project was python. This was mainly due to the large number of libraries available and the flexibility of the language. For our Bayesian Network, we used the bayespy library.

Our first step was to create the framework for the game. This included constructing classes for the cards, deck, players, boards, and game play. This portion of the project was more time consuming than we had anticipated. The first hurdle was the amount of actions for the game was far more than we had initially expected. The second hurdle was the language barrier. We had both worked with python before, but not to this level of complexity. As a result many minor errors and incorrect assumptions were made, which took time to track down and correct. Once these had been dealt with, we were faced with the challenge of constructing a Bayesian Network for the AI.

We then created a DumbAI, which played as many cards as it could out of its hand on itself and the person to its right. When given options about selecting a card from a general store, or discarding at the end of a turn, the DumbAI picked the first option presented.

Our second step was to physically draw the model of the Bayesian Network we wished to create.

PUT PIC/DIAGRAM HERE

We then broke the diagram down into levels of implementation to make the task manageable

LEVEL DIAGRAMS HERE

Results:

Future Work:

Bibliography