Jason Chalom

Honours Research Project:

An Investigation of AI Tree Search Methods and Their Effectiveness at Playing the Card Game Gwent

The Game

- Gwent is a 2-player turn-based game, with three rounds.
- The game is the best of three rounds and turn based.
- Each player must have a chosen deck which consists one leader card, twenty-two unit cards minimum, and ten special cards maximum. Unit cards which are hero cards, cannot be revived or effected by any other card.
- Ten random cards from the chosen deck are drawn at the start of the three rounds for both players. Each player takes a turn placing one card on the board.
- The board has three rows on both sides.

The Game

- Certain units can only be placed on certain rows.
- A player can also at any turn decide to pass the remainder of the round.
- The winner of a round is decided when either both players run out of cards, both players pass the round or a combination of the two.
- A round win is calculated by adding up the strength of all unit cards whilst taking into account the effects of the special cards.

Some Images


15
Geralt of Rivia
"If that's what it takes to save the world, it's better to let that world die."


7
Yennefer of Vengerberg
"Magic is Chaos, Art and Science. It is a curse, a blessing and a progression."

Rule List

- Unit — Base Card With Attack Points
- Horn — Doubles a rows AP
- ~~Scorch~~ — ~~Kills most powerful Cards in play~~
- Frost, fog, rain, clear — Weather cards reduces row to 1
- Medic — Revives a card in the discard pile also has ap
- ~~Decoy~~ — ~~I have left it out~~
- Spy — Placed on opponents side but get 3 new cards in hand
- Hero — Similar to unit card
- Bond — A unit card but if other bond cards in same row doubles ap

- I Have assumed both players know both decks, discard piles and what is currently on the board. The current players' hands are unknown to each other.

What I need to make

Minimax
Monte-Carlo Tree Search
Possible Hybrid?

Reward Function for Minimax:

- **RF = 0.2(CurrentAttackPoints – EnemyAttackPoints) + 0.2(NumberCards – EnemyNumberCards) + 0.4(RoundWIn - EnemyRoundWin)**

Problems and Possible Solutions

- Rules have been very hard to implement
  - Weather rules are kind of broken .....
  - Tests are also unreliable
  - I plan to make some of the rules simpler like weather just being the count of cards in a row
    - Change the gamestate to make the implementation easier
- I have already left out many rules
  - I plan to remove scorch to make things a bit easier for me
- Reduce the scope of my project to 2 Ais
- Minimax has issues with the unknown player hand
  - Im going to use random roll-outs over a sample space of possible cards
  - Not going to try more complicated approaches.

Problems and Possible Solutions

- I have statically set the players and decks of the game
  - i.e. player 1 is one set deck and player 2 is another
  - I have json objects which describe both decks
  - The configuration of the game state is mostly built
  - Have to change some of the cards whose rules I have ignored … Probably make them low level unit cards
- Dealing with strange rules
  - Passing a round is treated as just another card for the Ais
  - Medic
  - Sends another decision to the same AI where it is searching the discard pile rather than the current hand
    -

FIN