Arimaa

Arimaa, the Game of ¿Real Intelligence?

Nicolás A. Barriga Richards
nbarriga@inf.utfsm.cl
Arimaa: The Game of Real Intelligence

- Easy rules for humans to learn
- Computer ineptitude
  - Large branching factor
    - Multi-step moves
  - Variable opening position
- Finals with many pieces left
- Positional(arimaa) vs Materialistic(chess) evaluation
Rules

- Variable opening position
- 1 move = 4 steps
- Push move
- Pull move
- Capture
- Immobilization
- Goal
Portada

Arimaa
The Challenge

- Program that can defeat the top human players
- Before 2020
- Reward of US$10.000
The Championship

- **Computer championship** (US$500 & US$200)
  - Floating triple elimination
  - 40 days playing before wcc
- **Human championship** (US$500 & US$200)
  - Floating double elimination
- **Challenge match** (US$10,000)
  - 3 3-game matches against top three humans
Developing a Bot
The Environment

- Simple interface
- gamestate file
- position file
- move file
- move output
- Bot kit (on-line and off-line)
Input files

- running/matchGamestate
- running/matchMove
- running/matchPos
Output

- \{Piece\}\{Pos\}\{Direction\}
- Rain Me2s de3s de4x Ef2e
Matchoffline script

- match test1 test2
- test1 ./getMove getMove -d 4
- test2 ./getMove getMove -d 4
Developing a Bot
The Search

- Variations on minimax
  - Alphabeta
  - MTD(f) (zerowindow)
  - Killer heuristic
  - Nullmove Heuristic
  - Transposition tables
  - Search extensions

- Horizon effect
Alpha-Beta Search

evaluate (node, alpha, beta)
  if node is a leaf
    return the heuristic value of node
  if node is a minimizing node
    for each child of node
      beta = min (beta, evaluate (child, alpha, beta))
      if beta <= alpha
        return alpha
    return beta
  if node is a maximizing node
    for each child of node
      alpha = max (alpha, evaluate (child, alpha, beta))
      if beta <= alpha
        return beta
    return alpha
Developing a Bot
The Evaluation

➢ Feature extraction
  ▶ Hand made

➢ Types
  ▶ Linear Functions
  ▶ Neural Networks (experimental)

➢ Tunning
  ▶ Hand tuned
  ▶ Genetic algorithms (experimental)
Developing a Bot

Possible Improvements

- Search via coevolutionary GA
- Train evaluation function via TD(Lambda)
- Automatic characteristic extraction
- Montecarlo Search
Bitboard handling and other technical stuff