

# General Game Playing

N00Bs presenting their game player... ;-)

# Architecture

- main application written in Java
- Java interface of ECLiPSe prolog system
- use prolog to derive states etc.
- searches, heuristics etc. written in Java

# Game Analysis

detect:

- single player games
- multiplayer games divided into:
  - synchronous games
  - turn taking games divided into:
    - zero sum games
    - non zero sum games

# Search Algorithms

- single player: BFS followed by IDS
- turn taking, zero sum: alpha beta pruning
- turn taking, non zero sum: max search
- all others: minimax search

# Heuristics

- primary use of hand crafted goal heuristic
  - evaluates to which degree goals are fulfilled
- supplemented by a mobility heuristic
- no way to automatically figure out which heuristics is good for which game
- therefore we use a static combination of these heuristics

# Optimizations

- hashing of visited states
- IDS of alpha beta pruning and max search reuses the game tree
  - leads to a kind of anytime algorithm
- randomization of equally good moves
  - won't always choose paper in roshambo ;-)

# Summary

- stucked to basic algorithms, due to lack of time
- did several optimizations to increase speed