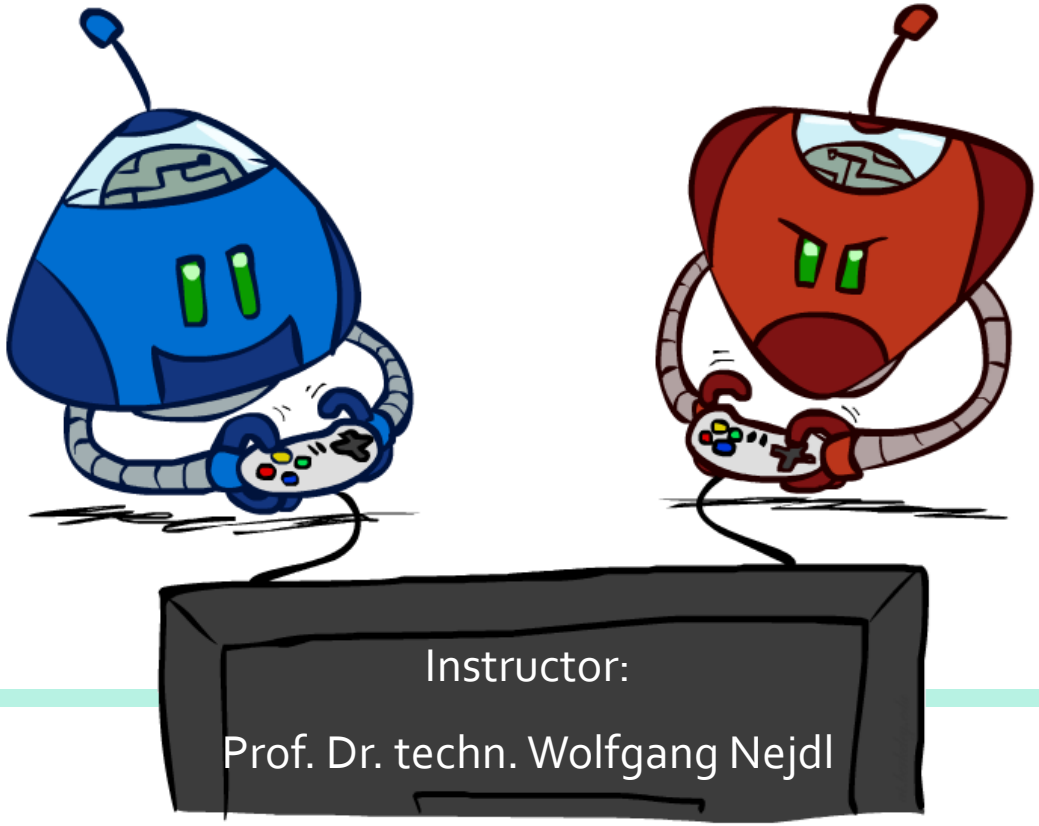
Advanced Topics in AI

Search with Other Agents



Instructor:

Prof. Dr. techn. Wolfgang Nejdl

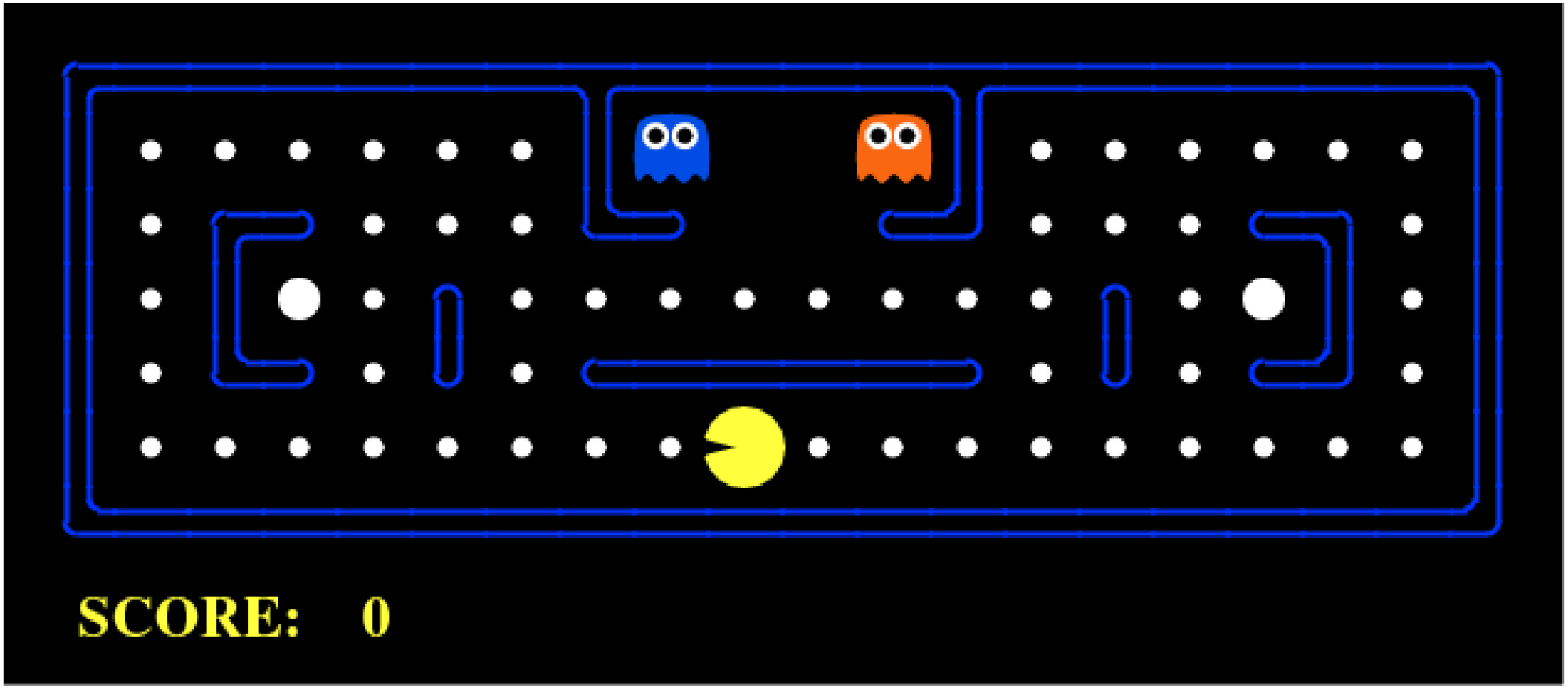
Leibniz University Hannover

Co-financed by the Connecting Europe Facility of the European Union

[These slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All materials are available at http://ai.berkeley.edu.]

Why Multiple Agents?

- Play multi-agent Pacman! (and other games)



Co-financed by the Connecting Europe Facility of the European Union

xAIM

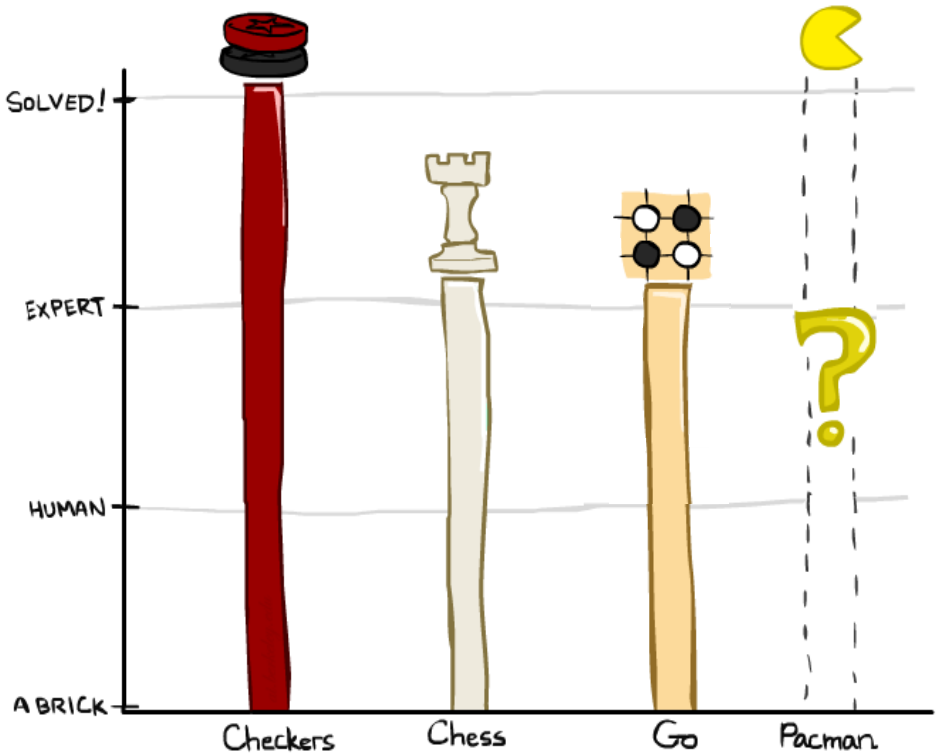
Why Multiple Agents?

- Play multi-agent Pacman! (and other games)
- AI that operates alongside humans
  - or other AIs
- Multiple agents lead to more complex environments/ecosystems
  - Inspired by evolution
  - games, robotics, generative adversarial networks (GANs)
- We'll focus on games, but multi-agent ideas come up in many areas of AI

Co-financed by the Connecting Europe Facility of the European Union

Game Playing Progress

- **Checkers**: 1950: First computer player. 1994: First computer champion: Chinook ended 40-year-reign of human champion Marion Tinsley using complete 8-piece endgame. 2007: Checkers solved!
- **Chess**: 1997: Deep Blue defeats human champion Gary Kasparov in a six-game match. Deep Blue examined 200M positions per second, used very sophisticated evaluation and undisclosed methods for extending some lines of search up to 40 ply. Current programs are even better, if less historic.
- **Go** :2016: Alpha GO defeats human champion. Uses Monte Carlo Tree Search, learned evaluation function.
- Pacman



Co-financed by the Connecting Europe Facility of the European Union

Types of Games

- Many different kinds of games!

- Axes:
  - Deterministic or stochastic?
  - One, two, or more players?
  - Zero sum?
  - Perfect information (can you see the state)?

- Want algorithms for calculating a strategy (policy) which recommends a move from each state

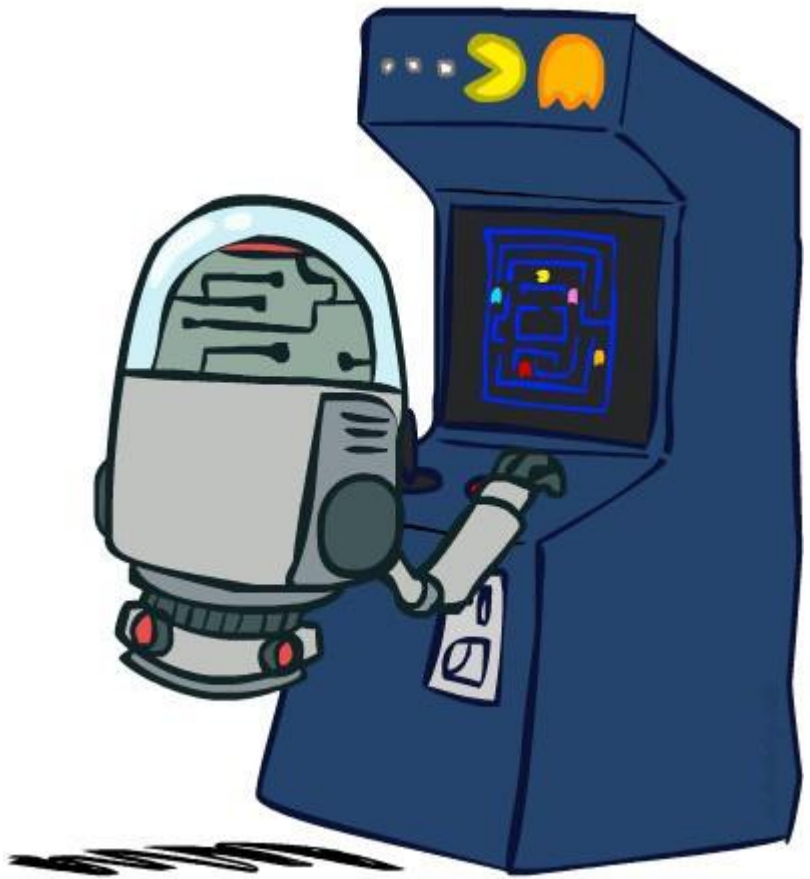
Co-financed by the Connecting Europe Facility of the European Union

Deterministic Games

Many possible formalizations, one is:

- States: S (start at s_0)
- Players: $P = \{1, \dots, N\}$ (usually take turns)
- Actions: A (may depend on player / state)
- Transition Function: $S \times A \rightarrow S$
- Terminal Test: $S \rightarrow \{t, f\}$
- Terminal Utilities: $S \times P \rightarrow R$

Solution for a player is a policy: $S \rightarrow A$



Co-financed by the Connecting Europe Facility of the European Union

xAIM

Zero-Sum Games

Zero-Sum Games

- Agents have opposite utilities (values on outcomes)
- Let us think of a single value that one maximizes and the other minimizes
- Adversarial, pure competition

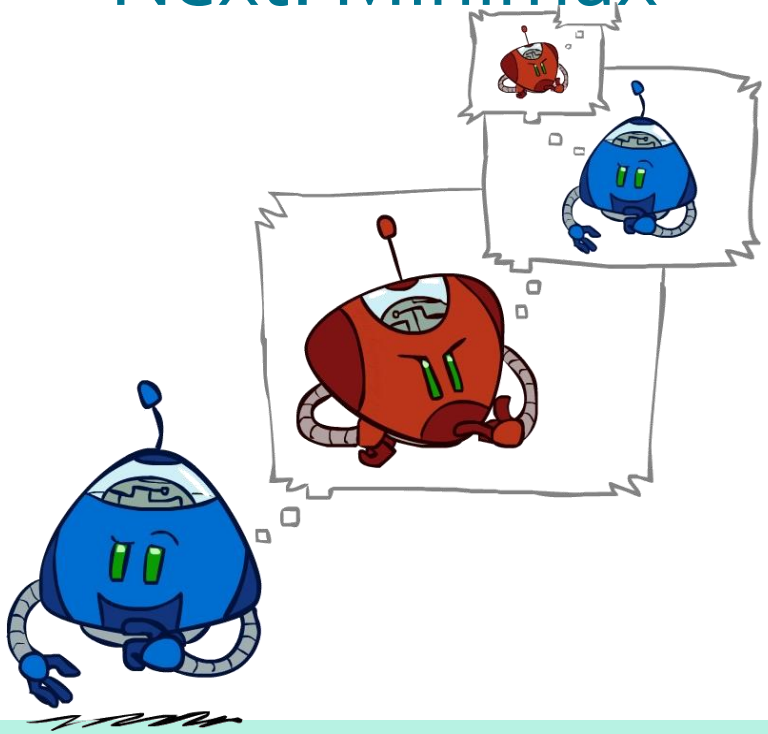
General Games

- Agents have independent utilities (values on outcomes)
- Cooperation, indifference, competition, and more are all possible
- More later on non-zero-sum games

Co-financed by the Connecting Europe Facility of the European Union

Advanced Topics in AI

Next: Minimax



Instructor: Prof. Dr. techn. Wolfgang Nejdl

Leibniz University Hannover

Co-financed by the Connecting Europe Facility of the European Union

[These slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All materials are available at http://ai.berkeley.edu.]