

# Handlungsplanung und Allgemeines Spiel

„Einführung“

Peter Kissmann

# Ablauf der Vorlesung

- ▶ Termine:
  - Mittwochs 10 bis 12 Uhr
  - Donnerstags 10 bis 12 Uhr
- ▶ Vorlesungszeit: 14 Wochen → 28 Termine
- ▶ Davon etwa
  - die Hälfte Handlungsplanung (zumeist mittwochs)
  - ein Viertel Allgemeines Spiel (donnerstags)
  - Rest Übungen (donnerstags)

# Ablauf der Übungen

- ▶ Konstruktion eines allgemeinen Spielers
  - Bildung von Gruppen (2-4 Studierende pro Team)
  - Aufgaben, um Fortschritt zu steuern
    - Ausgabe mittwochs, Abgabe Montag vor Übung (bei Stefan, mir oder in unserem Briefkasten im Sekretariat)
  - Alle zwei Wochen doppelstündige Übung
    - darin, Besprechung der erzielten Ergebnisse
  - erste Übung: Do., 11.11.
  - Am Ende Turnier zwischen allen erstellten Spielern
    - evtl. erst am Ende der vorlesungsfreien Zeit Ende März

# Themen Allgemeines Spiel

- ▶ Einführung
- ▶ Game Description Language (GDL)
- ▶ Spielen allgemeiner Spiele
- ▶ Heuristiken im allgemeinen Spiel und Verbesserungen
- ▶ Lösen allgemeiner Spiele
- ▶ Instanziierung
- ▶ Ausblick: Unvollständige Information und Zufall

# Spiele, die Deep Blue spielen kann

## 1. Schach

# Spiele, die Chinook spielen kann

## 1. Dame

# Spiele, die ein allgemeiner Spieler spielen kann

1. Schach
  2. Dame
  3. Halma
  4. Vier Gewinnt
  5. Tic-Tac-Toe
- ...

# Spiele, die ein allgemeiner Spieler spielen kann

...

6. Reversi
7. Mühle
8. 15-Puzzle
9. Nimm
10. Sudoku
11. Pentago
12. Blocker
13. Breakthrough
14. Lights Out
15. Amazons
16. Knightazons
17. Blocksworld
18. Cephalopod

19. Zhadu
20. Qyshinsu
21. Pancakes
22. Quarto
23. Knight's Tour
24. Dameproblem
25. Blob Wars
26. Bomberman (vereinfacht)
27. Catch a Mouse
28. Chomp
29. Gomoku
30. Hex
31. Cubicup

...



# Spiele, die ein allgemeiner Spieler spielen kann

- 95. Bidding Tic-Tac-Toe
- 96. 9-Brett Tic-Tac-Toe
- 97. Labyrinth lösen
- 98. Schere-Stein-Papier
- 99. Iteriertes Gefangenendilemma

und viele, viele mehr...

# General Game Playing

- ▶ Allgemeine Spieler können
  - beliebige Spielbeschreibungen einlesen und verarbeiten
  - auf deren Basis effizient spielen
- ▶ Unterschied zu spezialisierten Spielern (DeepBlue, Chinook etc.):
  - haben kein spezielles Wissen
  - müssen Wissen über aktuelles Spiel selbständig erarbeiten

# General Game Playing

- ▶ wichtige Technologien für Allgemeine Spieler:
  - Wissensrepräsentation
  - Resoning
  - Lernverfahren
  - Rationales Verhalten

# Unterschied zur Handlungsplanung

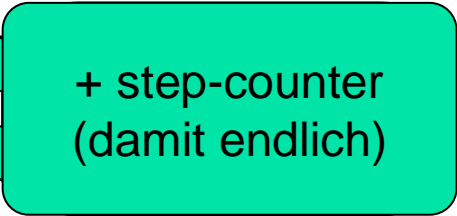
- ▶ Klassische Handlungsplanung entspricht 1-Personenspielen
- ▶ Nicht-deterministische Planung als 2-Personenspiel ansehbar
- ▶ Handlungsplanung:
  - Planlänge meist relevant
  - Aktionskosten
  - Evtl. Bewertung am Ende (Preferences, Net-Benefit)
  - Kann Kreise enthalten
- ▶ Allgemeines Spiel:
  - Nur erreichte Bewertung in Terminalzustand relevant
  - beliebige Spieleranzahl
  - gleichzeitige oder abwechselnde Züge möglich
  - Spiele on-the-fly spielbar
    - vorgegebene Zeit für einzelnen Zug

# General Game Playing

- ▶ Einschränkungen des derzeitigen Formalismus: Spiele sind
  - endlich
  - diskret
  - deterministisch
  - vollständige Information für alle Spieler
- ▶ Ein- und Mehrpersonenspiele
  - Spielerzahl endlich und fest (von Start bis Ende)
- ▶ Gleichzeitige und abwechselnde Züge
  - Immer  $\geq 1$  Zug pro Spieler
  - standardmäßig nur simultane Züge
  - nicht-simultan: sog. “noops”
    - Züge, in denen die nicht-aktiven Spieler nichts tun

# Mögliche Spiele

- ▶ Vier Gewinnt
- ▶ Tic-Tac-Toe
- ▶ Solitaire
- ▶ Türme von Hanoi
- ▶ Halma
- ▶ Schach
- ▶ Sokoban
- ▶ etc.



+ step-counter  
(damit endlich)

# Unmögliche Spiele

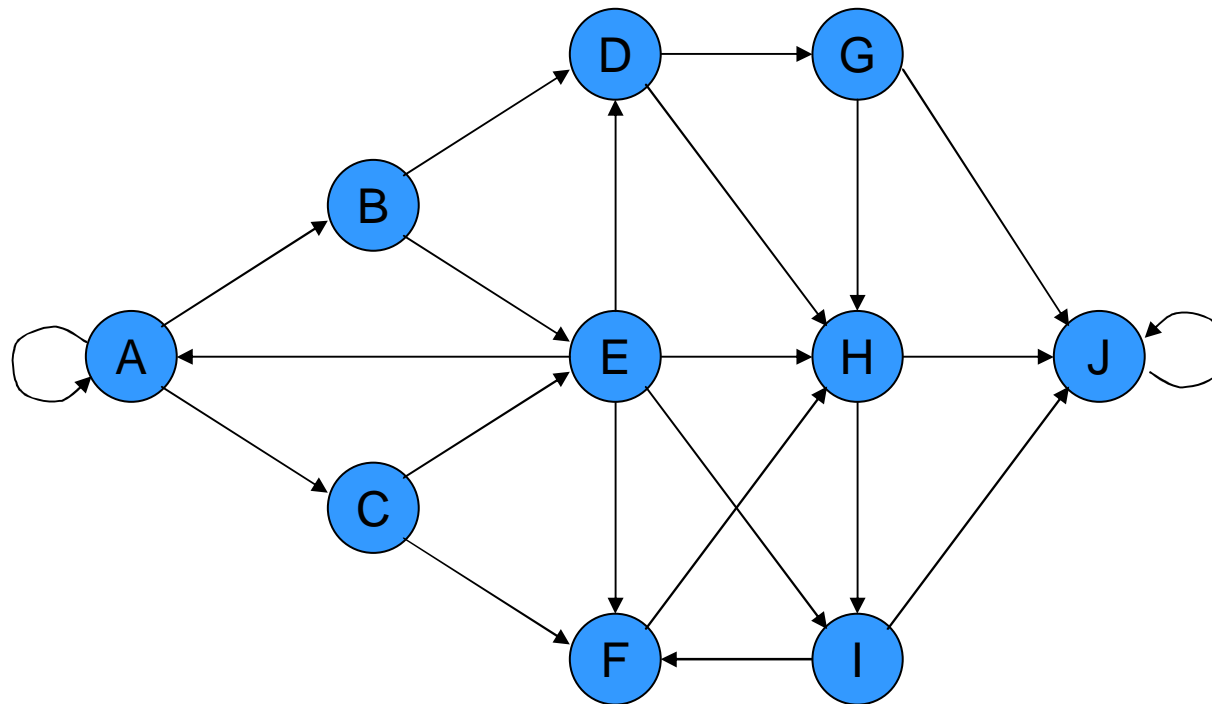
- ▶ Kartenspiele (Skat, Poker, Doppelkopf etc.) → unvollständiges Wissen
- ▶ Schiffe versenken → unvollständiges Wissen
- ▶ Würfelspiele (Kniffel, Backgammon, Mensch ärgere Dich nicht etc.) → Zufall (nicht deterministisch)
- ▶ Computerspiele → Zufall, unvollständiges Wissen, evtl. kontinuierlich, nicht endlich (?)
- ▶ Pen-and-Paper Rollenspiele → Zufall, unvollständiges Wissen, nicht endlich
- ▶ etc.

# General Game Playing

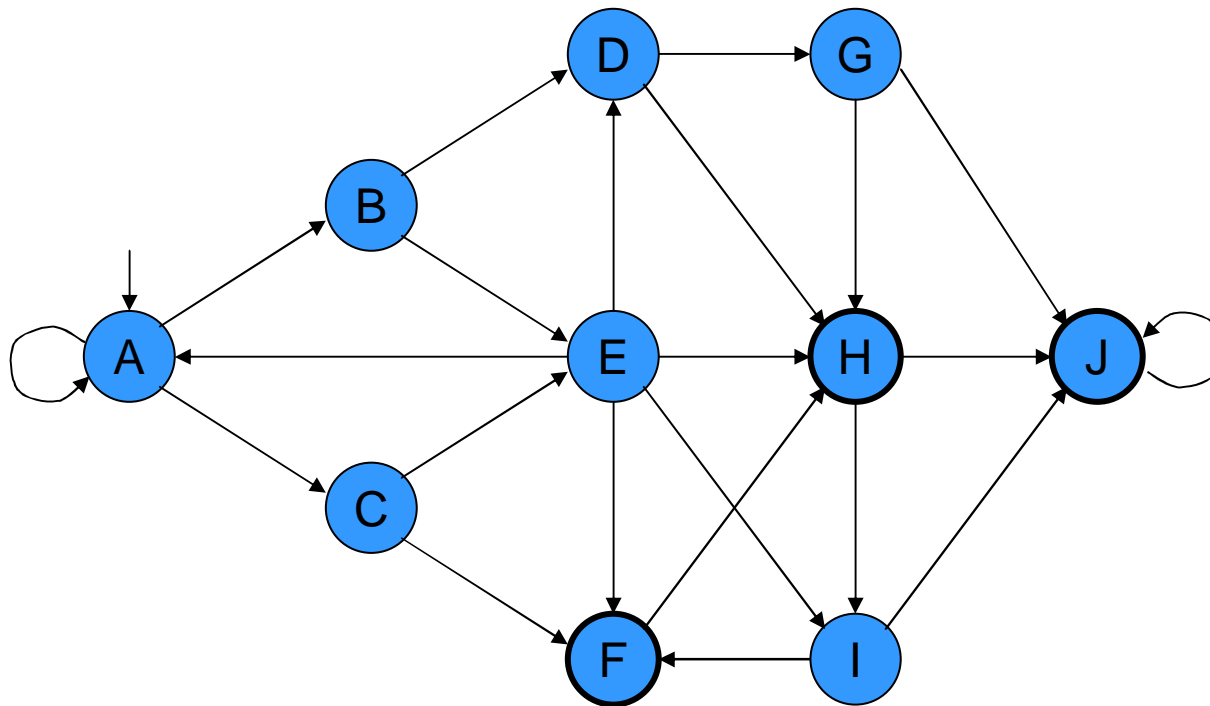
- ▶ Spielverlauf = Abfolge von Zuständen
- ▶ speziell:
  - ein Initialzustand
  - ein oder mehrere Terminalzustände
  - Bewertung der Terminalzustände (Ziele o.ä.)
- ▶ Jedes Spiel
  - startet an Initialzustand
  - endet an Terminalzustand
  - Spieler erhalten Gewinn gemäß Bewertung



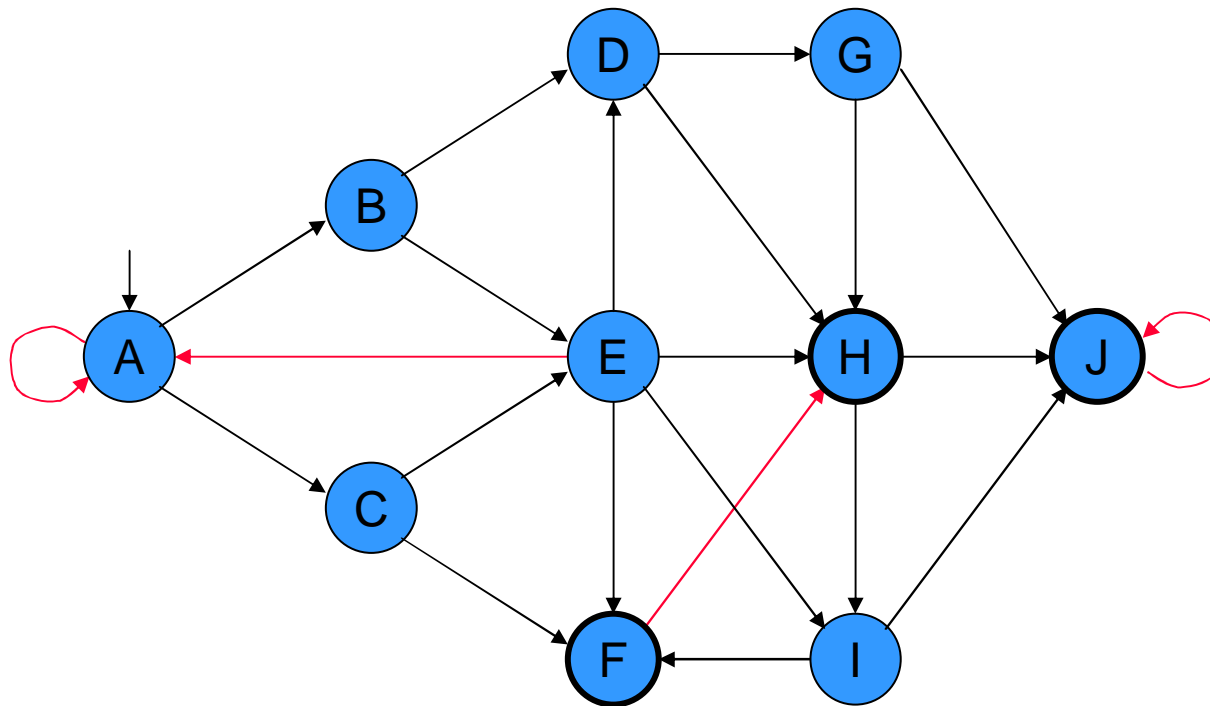
# Spiele als Automaten



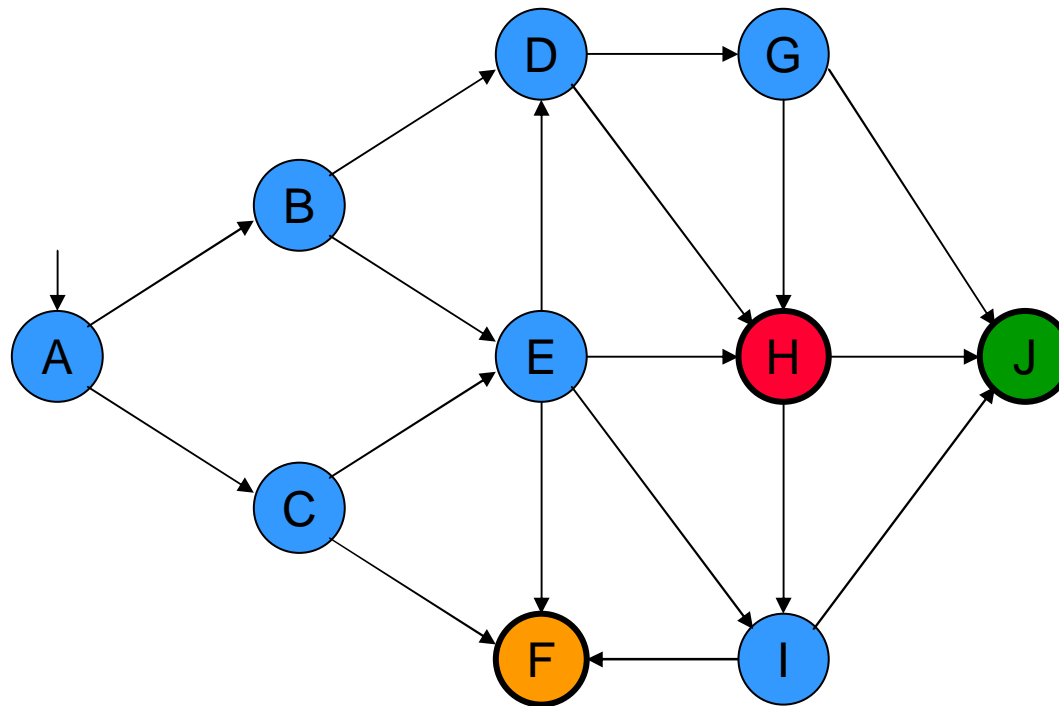
# Initialzustand und Terminalzustände



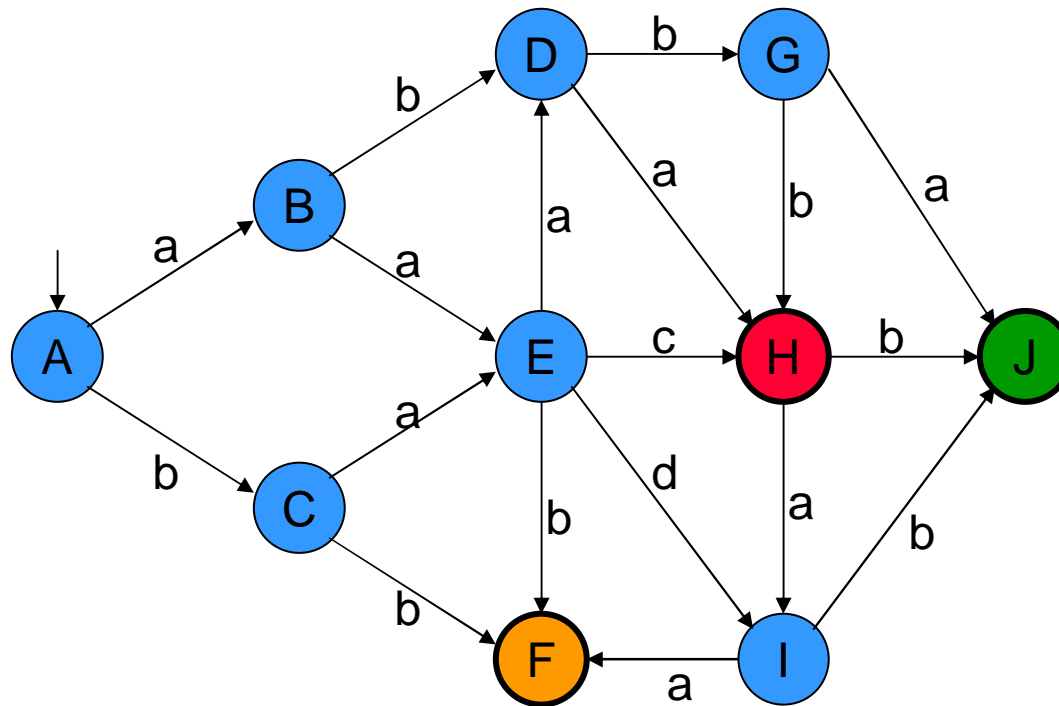
# Terminierendes Spiel



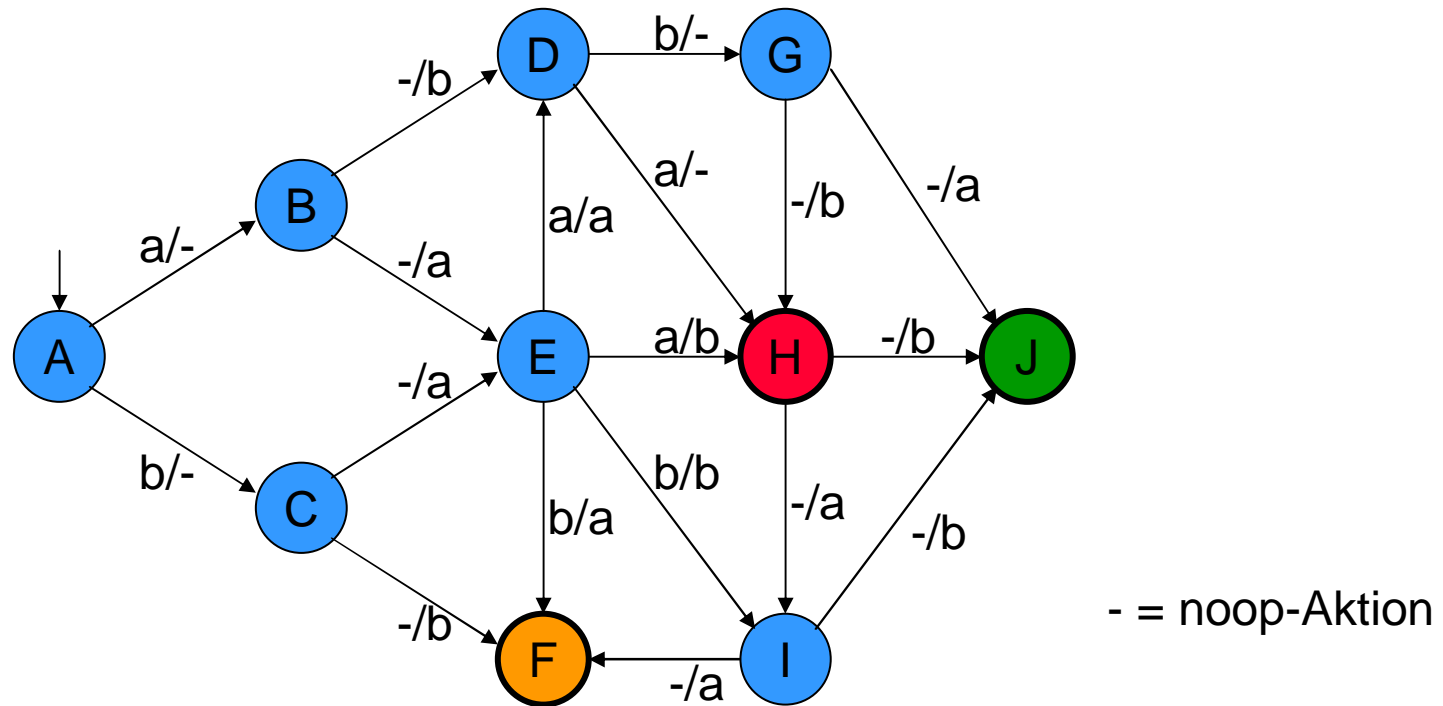
# Zielzustände (untersch. Qualität)



# Züge (Eiersonenspiel)



# Züge (Zweipersonenspiel)



# General Game Playing - International

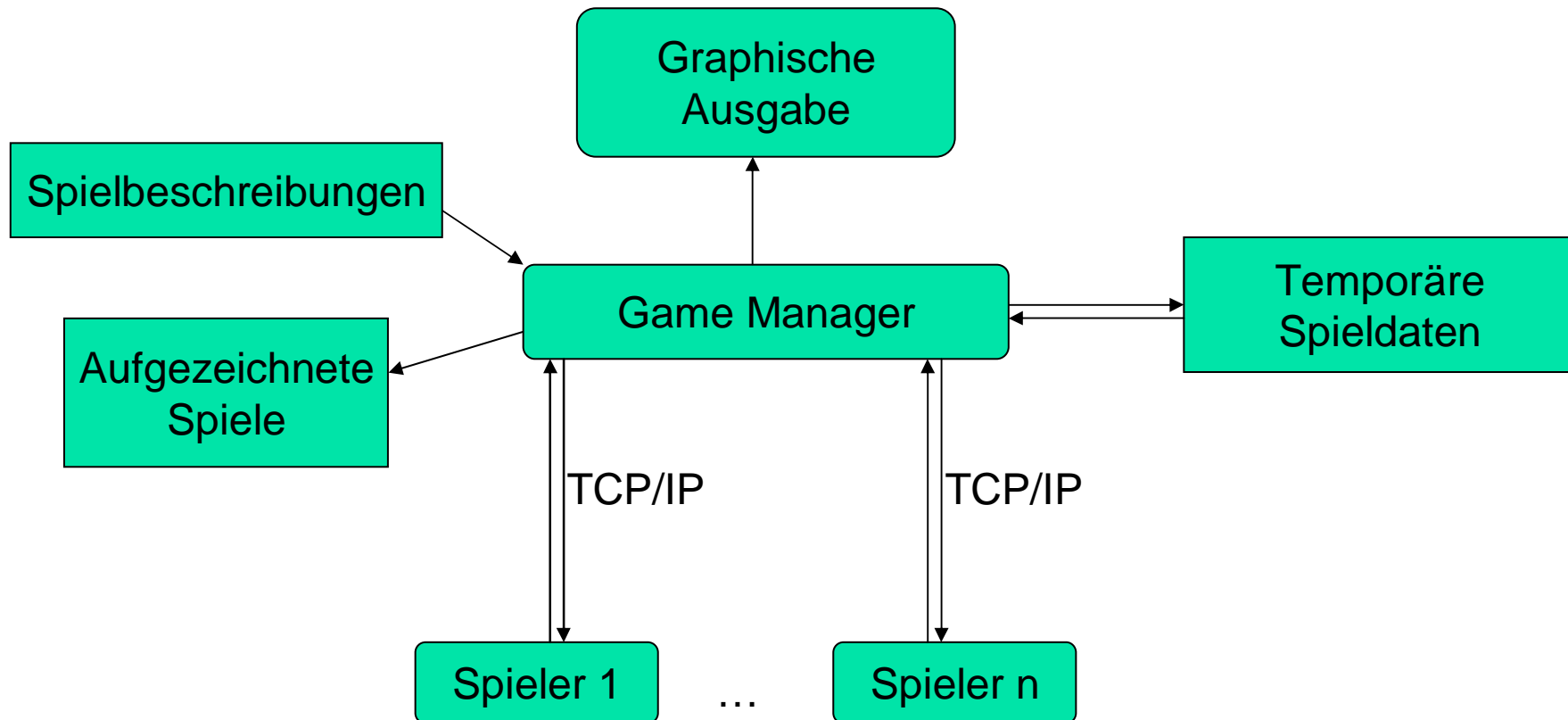
- ▶ Ursprünglich Projekt der Games-Gruppe der Uni Stanford
  - <http://games.stanford.edu>
- ▶ Weltmeisterschaften seit 2005 (bei AAI-Konferenz)
- ▶ Deutsche Meisterschaften seit 2009
- ▶ Viele Informationen (englisch) unter
  - <http://www.general-game-playing.de>
    - Literatur
    - Referenzspieler (Prolog, Java, C++)

# GGP-Server der TU Dresden

- ▶ <http://ggpserver.general-game-playing.de>
  - Große Zahl an Spielen (derzeit 219)
  - Jeder kann sich anmelden
  - Möglichkeit, selbst zu spielen
  - Eigenen Spieler anmelden und spielen lassen
  - Round-Robin
  - Turniere



# Game Manager



# HTTP Kommunikation

- ▶ Netzwerk-Modell:
  - Spieler als Server
  - Game Manager als Client
- ▶ Game Manager erzeugt TCP/IP Verbindung zu Spieler
- ▶ Spieler kann nur auf Anfragen des Game Managers antworten

# Kommunikationsprotokoll

- ▶ Alle Game Manager Nachrichten der Form

```
HEADER  
<NEWLINE>  
CONTENT
```

- ▶ Standard HTTP Header:

```
POST / HTTP/1.0  
Accept: text/delim  
Sender: GAMEMASTER  
Receiver: <PLAYER NAME>  
Content-type: text/acl  
Content-length: <LENGTH OF CONTENT>
```

- Inhalt ACL; nicht zwingend durch <NEWLINE> beendet
- Details in der GDL-Spezifikation:
  - Love, Hinrichs, Haley, Schkufza & Genesereth: "General Game Playing: Game Description Language Specification "

# Kommunikationsprotokoll

- ▶ Spieler antwortet auf ähnliche Art, aber verwendet HTTP

Reply Header:

```
HTTP/1.0 200 OK
```

```
Content-type: text/acl
```

```
Content-length: <CONTENT LENGTH>
```

- ▶ Sobald Nachricht auf TCP/IP Socket geschrieben, Verbindung schließen

# Erste Nachricht: START

▶ Spiel startet mit der START Nachricht

```
(START <MATCH ID> <ROLE> <GAME DESCRIPTION> <STARTCLOCK>  
<PLAYCLOCK>)
```

▶ Bedeutungen:

- MATCH ID: Eindeutiger Name des aktuellen Spieles
- ROLE: Rolle, die uns zugewiesen wird
- GAME DESCRIPTION: Logische Beschreibung des Spieles
- STARTCLOCK: Vorverarbeitungszeit, bevor erster Zug startet
- PLAYCLOCK: Zeit pro Zug

# Antwort auf START: READY

- ▶ Wenn ein Spieler bereit ist, sollte er READY an Game Manager schicken
  - signalisiert, dass Spieler lauffähig und bereit ist
  - Spiel startet spätestens nach der STARTCLOCK, auch wenn ein Spieler kein READY geschickt hat

# Spiel-Schleife: PLAY

- ▶ Zu Beginn jedes Zuges schickt Game Manager PLAY Nachricht:

```
(PLAY <MATCH ID> <PRIOR MOVES>)
```

- ▶ PRIOR MOVES:

- Liste von Zügen, einer pro Spieler
- Reihenfolge entsprechend Rollen in Spielbeschreibung
- z.B. ((mark 1 1) noop) in Tic-Tac-Toe:
  - Spieler 1 markiert Zelle (1, 1)
  - Spieler 2 macht nichts
- Spezialfall: Beginn des ersten Zuges → “nil”

# Antwort auf PLAY

- ▶ Nach spätestens PLAYCLOCK Sekunden muss Antwort bei Game Manager ankommen
- ▶ Antwort der Form  
MOVE
- ▶ MOVE: Gewählter Zug (etwa “(mark 1 1)” oder “noop”)
  - wenn Zug ungültig oder nicht in PLAYCLOCK, zufälliger Zug



# Spielende: STOP

- ▶ Sobald Terminalzustand erreicht, sendet Game Manager  
(STOP <MATCH ID> <PRIOR MOVES>)
- ▶ Informationen über Bewertungen müssen selbst berechnet werden (wenn Interesse)
- ▶ Antwort: “DONE”
  - nicht zwingend, aber nett

# Warum General Game Playing?

- ▶ Auch klassische Spieltheorie beschäftigt sich mit Spielen im Allgemeinen
- ▶ Aber: Programmierer
  - kennt das Spiel vorher
  - kann spezielles Wissen einfügen
- ▶ Im General Game Playing wird Spielbeschreibung erst zur Laufzeit übermittelt

# Warum General Game Playing?

- ▶ Allgemeine Spieler könnten als intelligenter angesehen werden
  - Klassische Spieler
    - denken nicht selbst
    - nutzen nur bereitgestelltes Wissen
  - Allgemeine Spieler müssen alles selbst erarbeiten