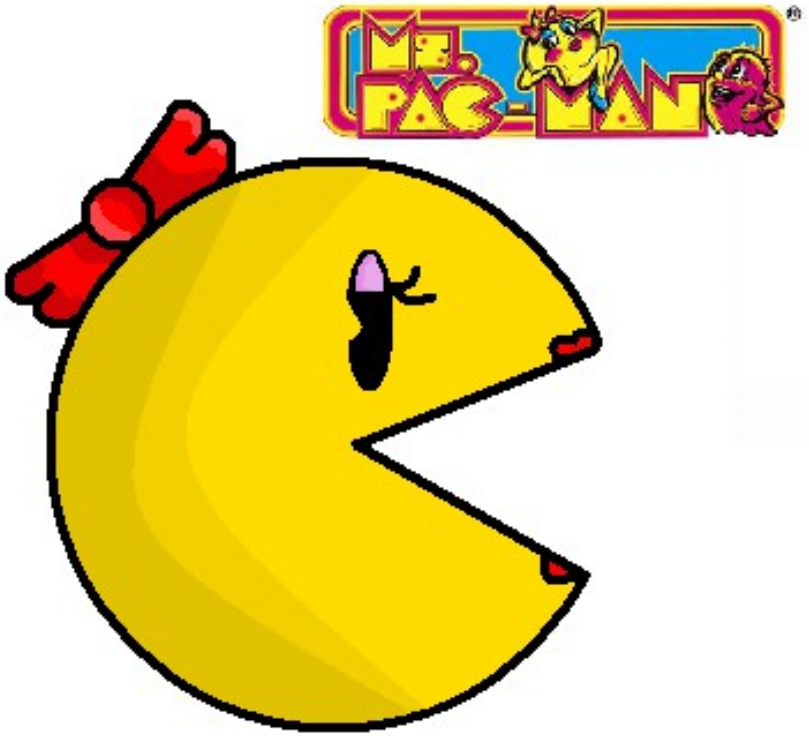
Real-time MCTS (4.7)

TECHNISCHE UNIVERSITÄT DARMSTADT

Seminar aus maschinellem Lernen



Übersicht

TECHNISCHE UNIVERSITÄT DARMSTADT

- Real-time MCTS – Was ist das?

- Beispiele für MCTS in Spielen

  →Tron

  →Ms. Pac-Man

Real-time MCTS – was ist das?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Go, Lines of Action, etc Zugbasierte Spiele
- Real-time games (Echtzeitspiele) sind zeitkritisch
- Agent hat wenig Zeit für Berechnungen
- Auch wenn kein Zug durchgeführt wird, geht das Spiel weiter
- Spiele können mitunter nicht deterministisch sein

Real-time MCTS – wieso MCTS?

TECHNISCHE UNIVERSITÄT DARMSTADT

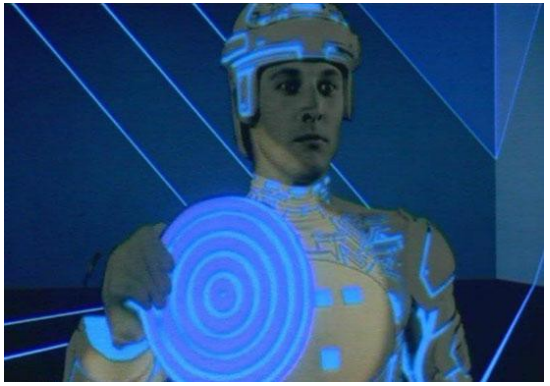
- Bisher meist durch Skripte, Trigger, Routinen gelöst
- Motivation: Gute Ergebnisse mit MCTS bei zugbasierten Spielen übertragbar?
- Ziel: Skripte outperformen, oder ähnliches Level erreichen

Tron – Der Film

TECHNISCHE UNIVERSITÄT DARMSTADT

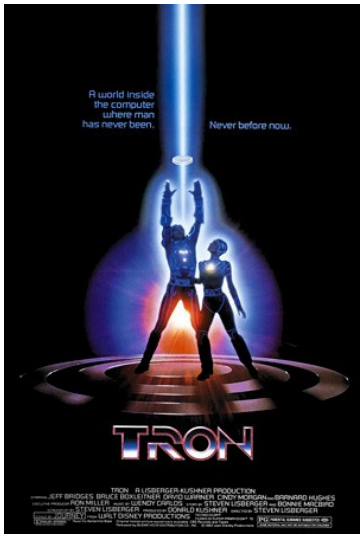
- Spiel basierend auf dem Film Tron
- Exkursion Film:

→1982 von Walt Disney erschienen



→Erster Film mit langen Computersequenzen, gilt als Meilenstein in der Computeranimation

→Kultfilm - Fortsetzung “Tron: Legacy” (2010)

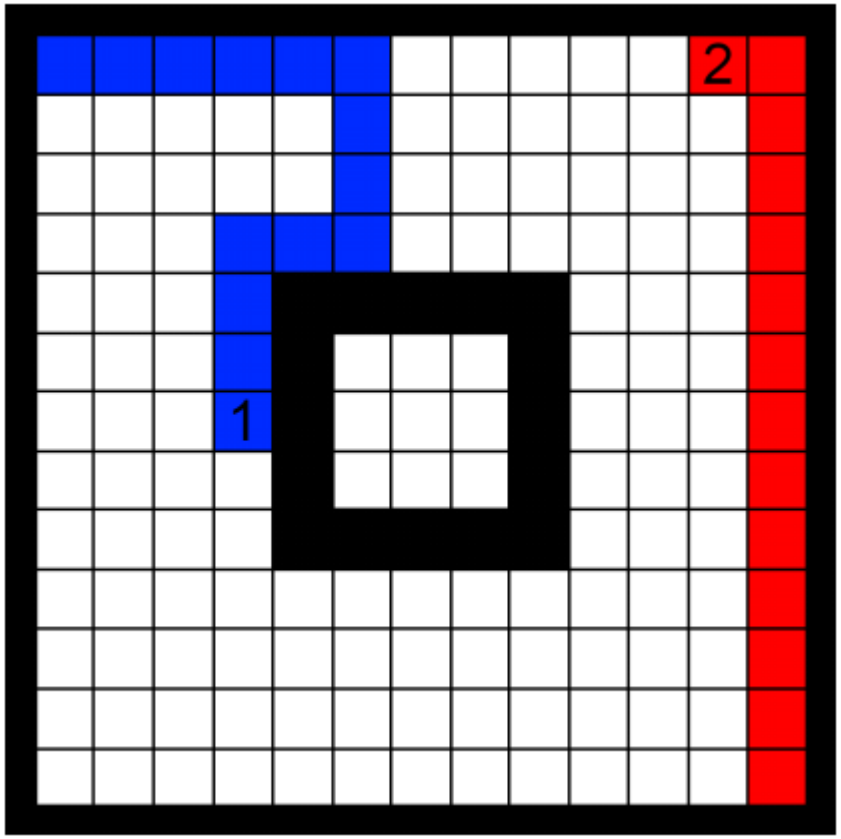


→Handlung: Programmierer will beweisen, dass Code von ihm ist, wird in Computerwelt materialisiert. Tron ist ein Überwachungsprogramm das “installiert” werden soll. Dort werden Kämpfe ausgetragen → dieses Spiel

Tron – Das Spielfeld

TECHNISCHE UNIVERSITÄT DARMSTADT

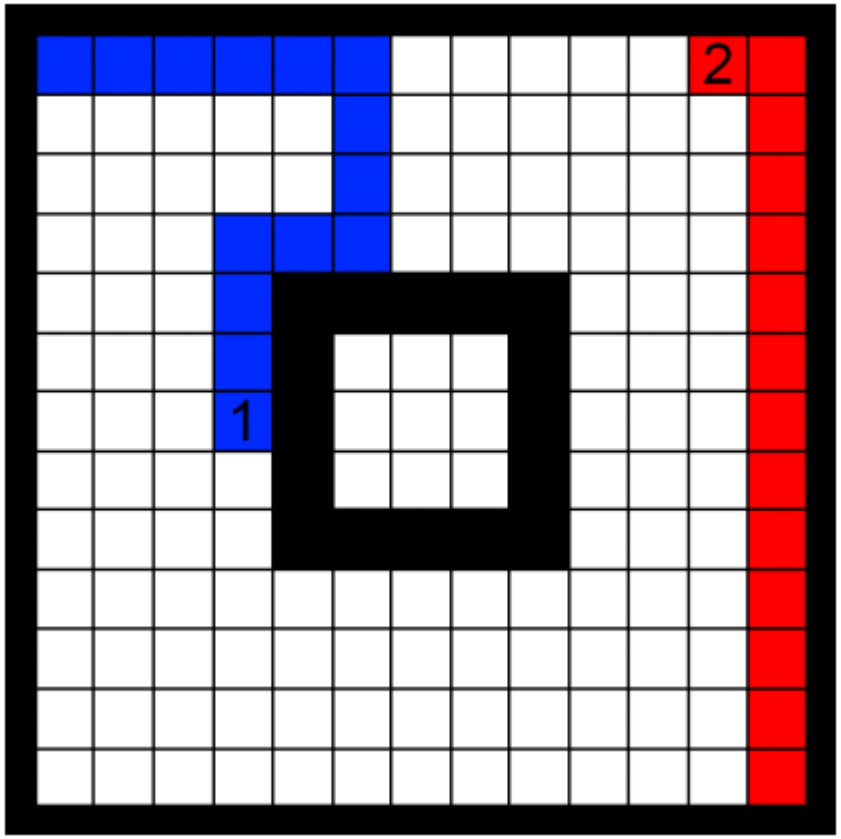
- Hier: Zwei Spieler
- Spielfeld durch Wand begrenzt. MxN groß
- Jedes Feld kann zwei Zustände haben (besetzt, frei)



Tron – Die Spielregeln

TECHNISCHE UNIVERSITÄT DARMSTADT

- Spieler können lediglich 90° Bewegungen ausführen
- Wände ersetzen freie Felder nachdem der Spieler drauf war
- Spieler stirbt wenn er auf ein besetztes Feld kommt
- Sieg, Unentschieden, oder Niederlage sind die Möglichen outcomes



Tron - MCTS

TECHNISCHE UNIVERSITÄT DARMSTADT

- MCTS mit UCT, da Baumstruktur aufgebaut wird
- Es gibt maximal 3 Züge da nur 2x 90° und geradeaus funktionieren. 180° → Niederlage
- Rewards: Sieg 1.0; Unentschieden 0.5; Niederlage 0.0
- Durchschnitt wird zurückgegeben (Backpropagation)

Tron – Game Agent Leitlinien

TECHNISCHE UNIVERSITÄT DARMSTADT

- Spieler können sich nicht selbst einsperren

- “Survival Mode” falls Spieler eingesperrt wurde

  →Verändert das Spiel in “Single Player” Variante

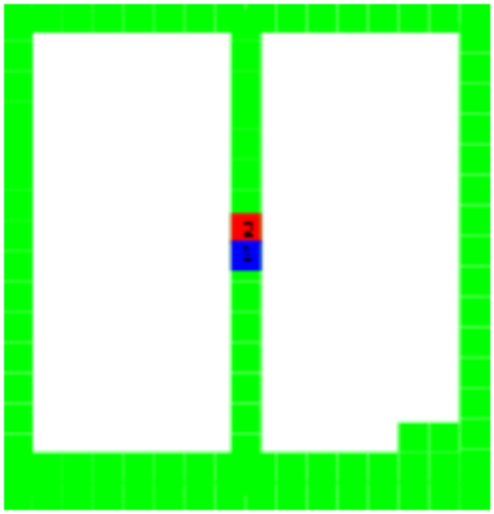
  →Der längste Weg muss gefunden werden

Tron – Experiment Spielfelder (Maps)

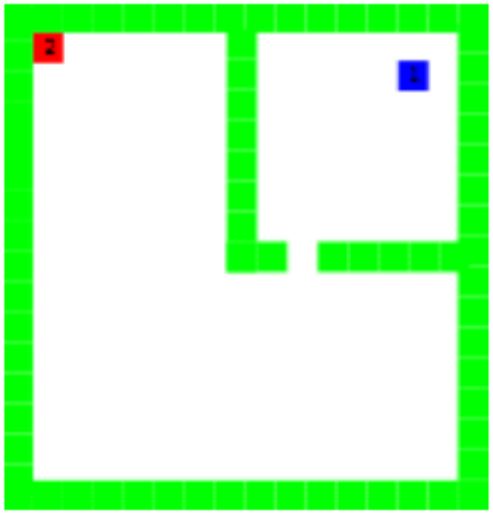
TECHNISCHE UNIVERSITÄT DARMSTADT

- Drei Spielfelder: unterschiedliche Problemstellungen

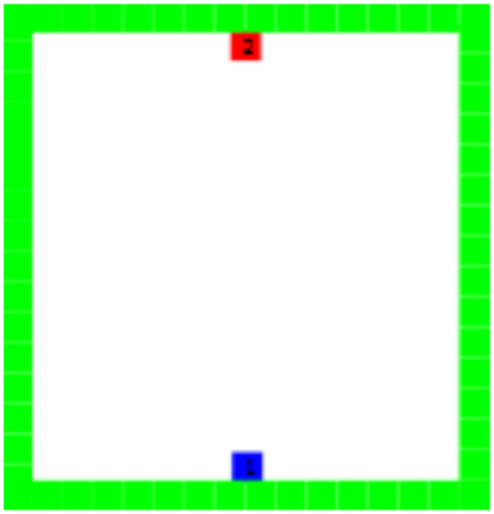
Map A)



Map B)



Map C)

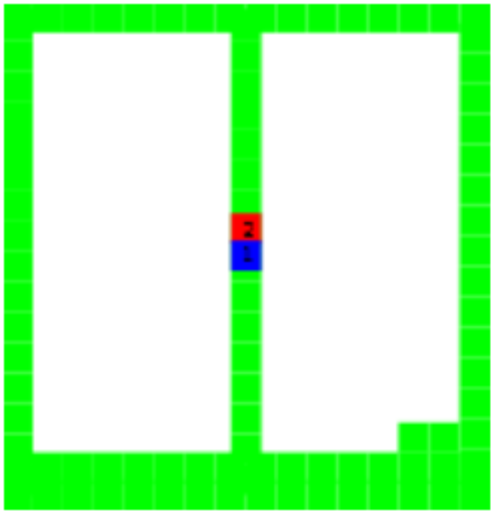


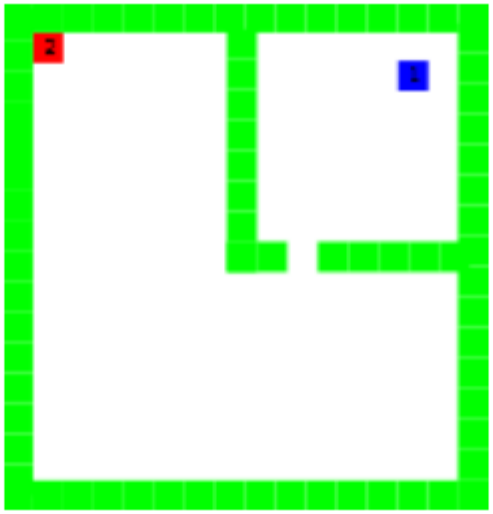
Tron – Map A

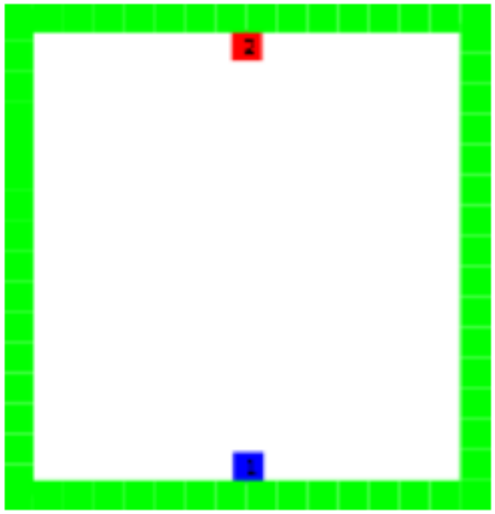
TECHNISCHE UNIVERSITÄT DARMSTADT

- Agent soll erkennen, dass Spieler Eins nach links muss

	Player Two Right	Player Two Left
Player One Right	(1,0)	(0,1)
Player One Left	(1,0)	(0.5,0.5)



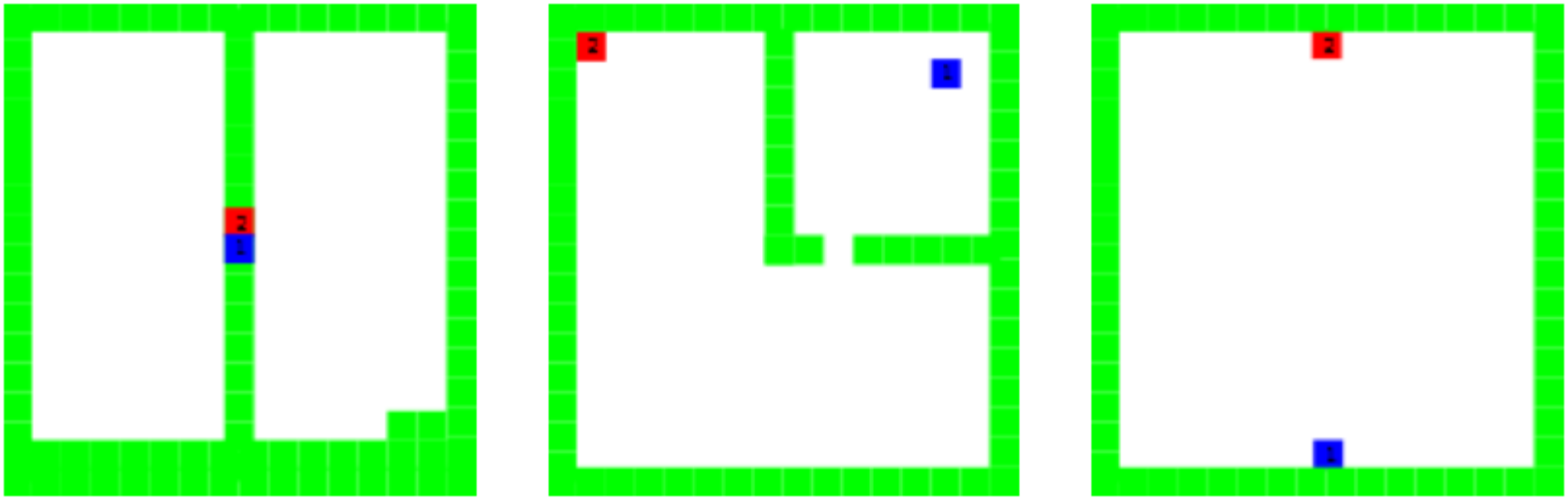




Tron – Map B

TECHNISCHE UNIVERSITÄT DARMSTADT

- Verlassen des “Käfigs” muss erkannt werden
- Ist im Vorteil weil näher am Zentrum, sollte also fast alle Spiele gewinnen

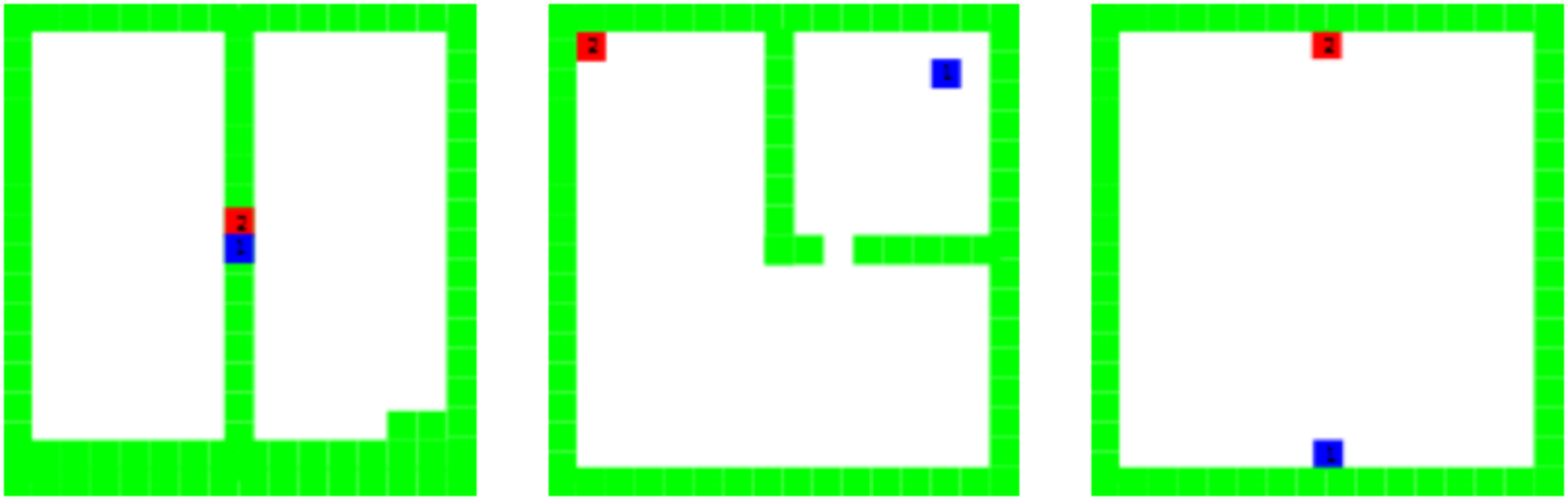


Tron – Map C

TECHNISCHE UNIVERSITÄT DARMSTADT

- “Open Field” Map

- Gleicht real-world Maps wahrscheinlich am ehesten



Tron – Setup Experimente

TECHNISCHE UNIVERSITÄT DARMSTADT

- Mehrere Agents
  - →UCB1 (Upper Confidence Bounds applied to Trees)
    - $\overline{x_j} + C\sqrt{\frac{\ln(n)}{n_j}}$
  - →UCB-TUNED $\bar{x_j} + \sqrt{\frac{ln(n)}{n_j} min\left\{1/4, \bar{x_j^2} - \bar{x_j}^2 + \sqrt{\frac{2ln(n)}{n_j}}\right\}}$
  - →UCB-E (Bandit)
    - $\overline{x_j} + C\sqrt{\frac{\sqrt{n}}{n_j}}$
- Mehrere Zeitintervalle zwischen den “Zügen”
  - →100ms – 900ms mit Erhöhungen um 50ms
- Maps werden 20 mal berechnet
- Gespielt wird gegen “Master Player” UCB1 mit 1000ms Rechenzeit

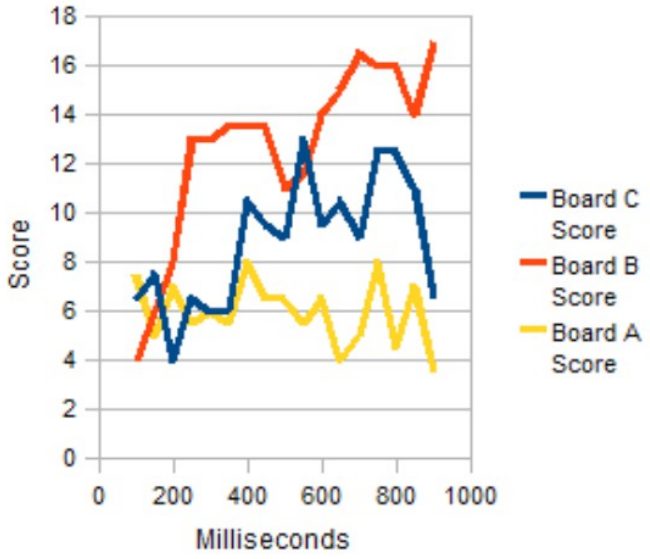
Tron – Experimente UCB1

TECHNISCHE UNIVERSITÄT DARMSTADT

- Reminder: $\overline{x_j} + C\sqrt{\frac{\ln(n)}{n_j}}$

- Agent erreicht nicht die optimale Performance bei Map B

- Ergebnisse für Map A, C sind eher zufällig



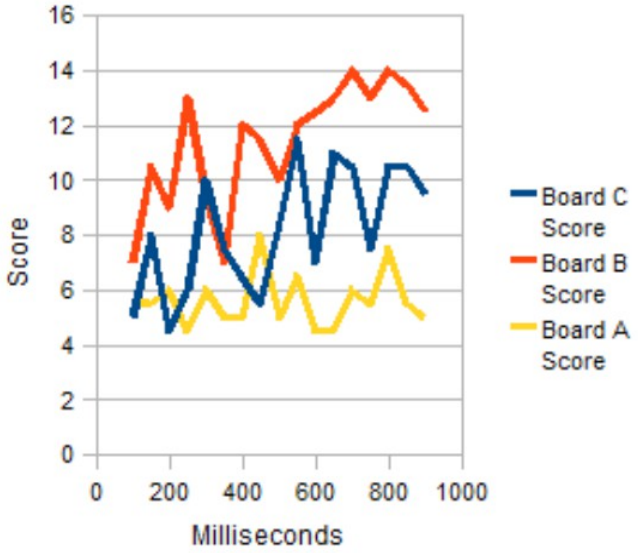
Tron – Experimente UCB-TUNED

TECHNISCHE UNIVERSITÄT DARMSTADT

- Reminder: $\bar{x_j} + \sqrt{\frac{ln(n)}{n_j} min\left\{1/4, \bar{x_j^2} - \bar{x_j}^2 + \sqrt{\frac{2ln(n)}{n_j}}\right\}}$

- Schlechtere Performance als UCB1 für Map B

- Scheint bessere Performance bei Map A, C zu haben (zu kleine Testmenge)



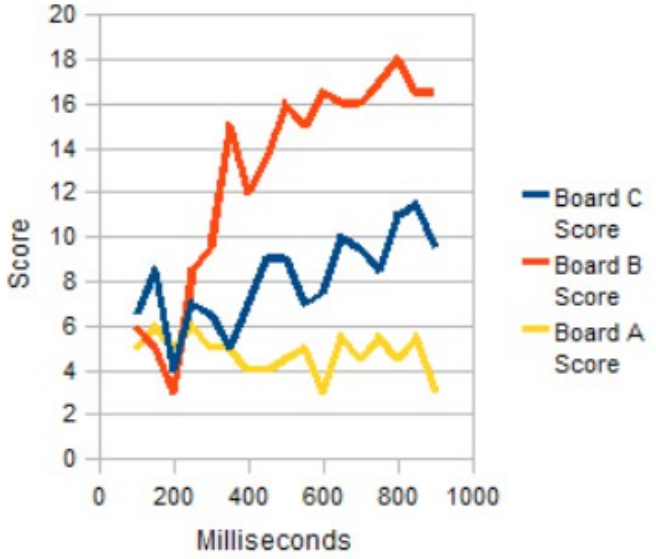
Tron – Experimente UCB-E

TECHNISCHE UNIVERSITÄT DARMSTADT

- Reminder: $\overline{x_j} + C\sqrt{\frac{\sqrt{n}}{n_j}}$

- Richtet sich mehr an Exploration, daher ist die Rechenzeit sehr wichtig

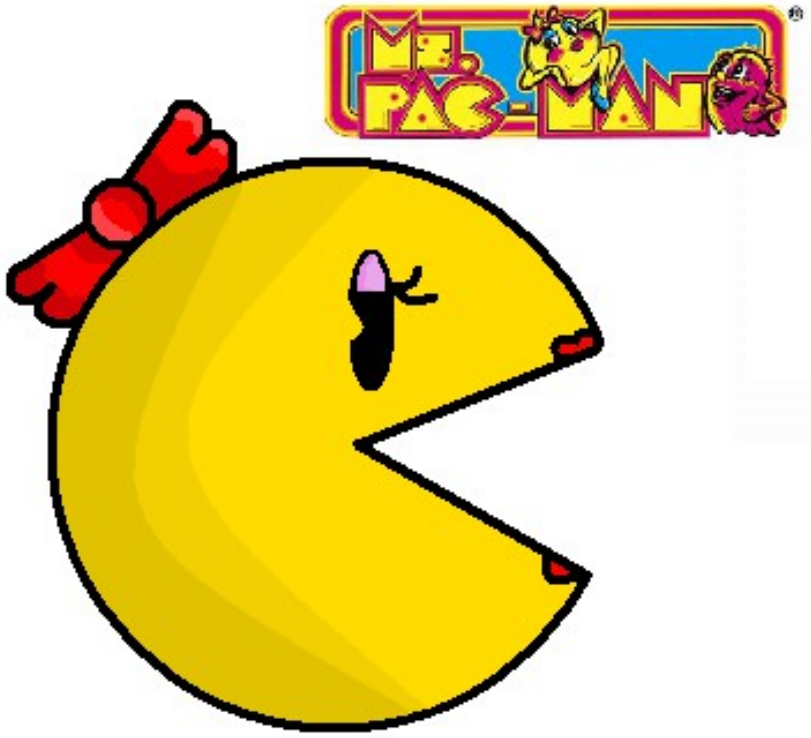
- Map A benötigt tiefe Suche, daher kommt wohl schlechtes Ergebnis



Ms. Pac-Man – Allgemeines I

TECHNISCHE UNIVERSITÄT DARMSTADT

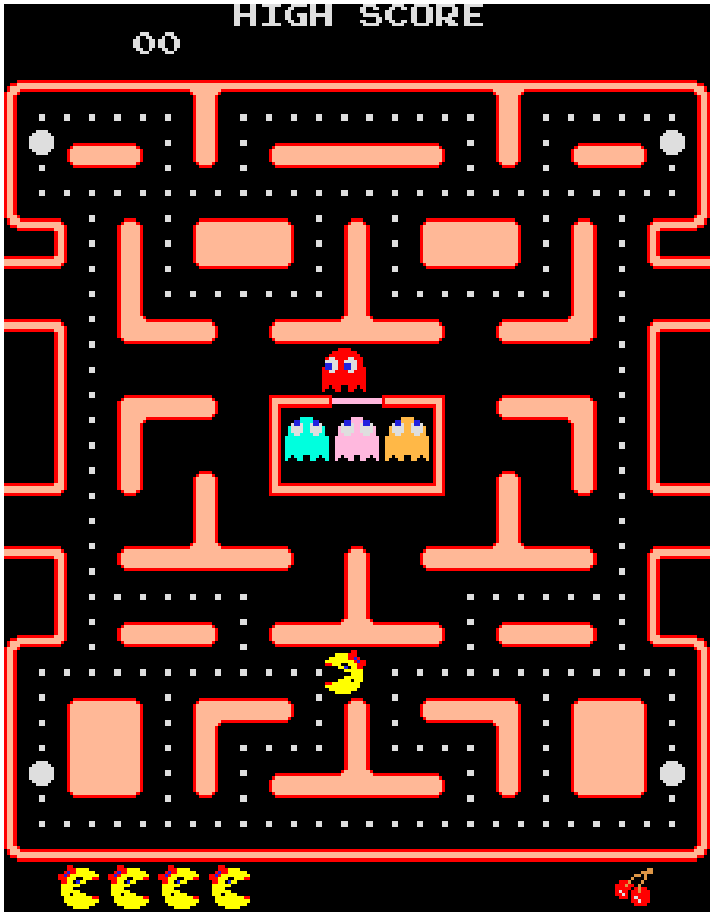
- Wieso Ms. Pac-Man und nicht Pac-Man?
- Unterschiede in den bekannten Spielfakten
- Geschwindigkeit, Laufwege der Gegner unbekannt.
- Beim essen der Pellets, ändern der Richtung verlangsamt Ms. Pac-Man. Faktor unbekannt



Ms. Pac-Man – Allgemeines II

TECHNISCHE UNIVERSITÄT DARMSTADT

- Durch die vielen unbekannten ist es schwer Regeln zu erstellen (skripten)
  →Regeln kamen in letzter Zeit an ihre Limits
- Interne Informationen dürfen nicht verarbeitet werden
- Generierung der Informationen mittels Computer Vision

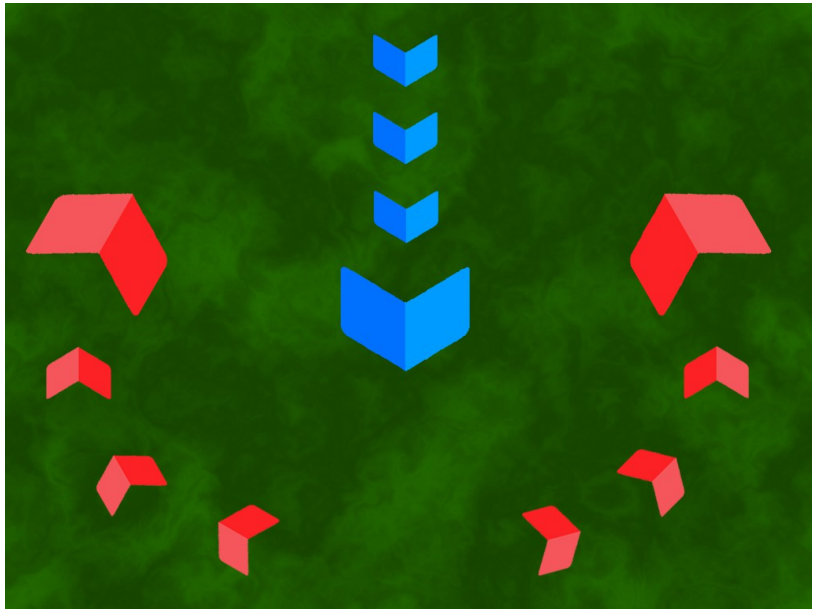


Ms. Pac-Man – Methodik

TECHNISCHE UNIVERSITÄT DARMSTADT

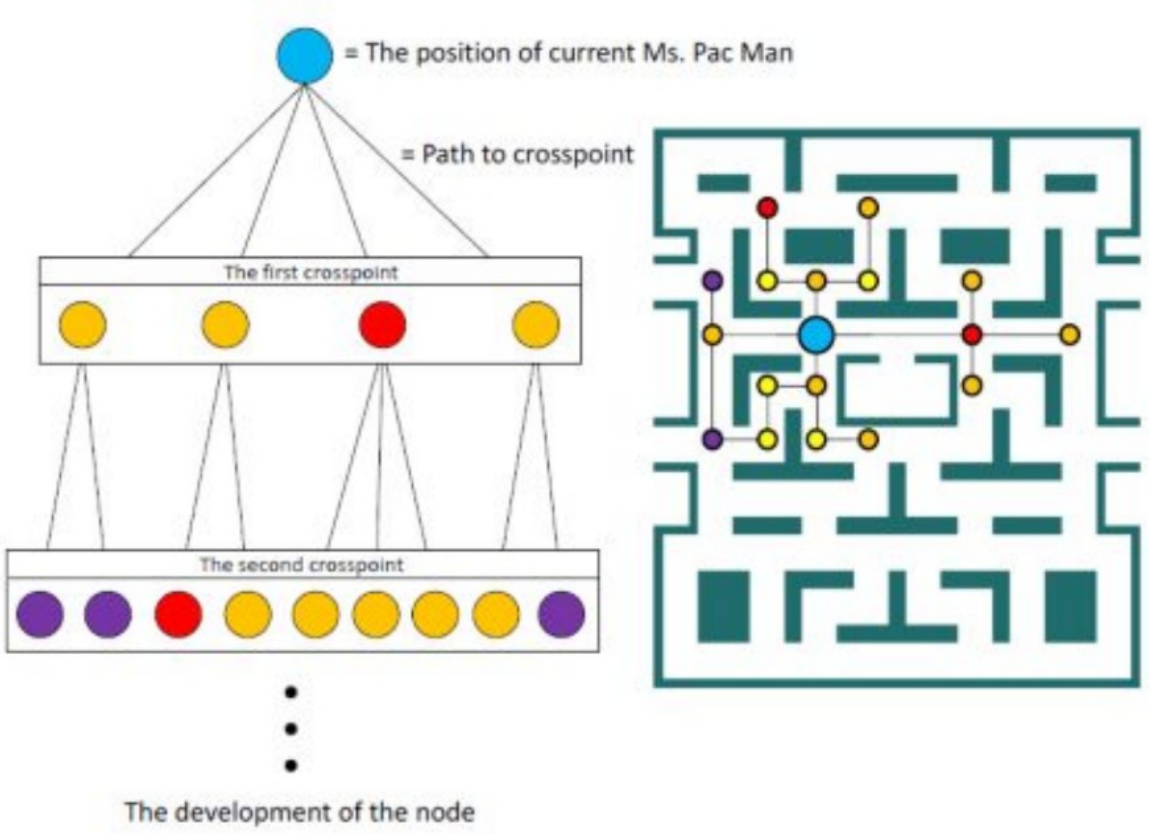
- Ziel ist es einen hohen Score zu erreichen. Sammeln der Pellets oder Früchte, Gegner aufessen

- Vermeidung von sog. Pincer Movements (Flankieren der Gegner)



Ms. Pac-Man – Interne Darstellung I

- Implemtieruung von C-Paths
- Gegnerbewegungen werden nicht mitmodelliert, doch miteinbezogen (c-paths ohne gegner)
- C-Path muss komplett entlang gelaufen werden.



Ms. Pac-Man – Interne Darstellung II

TECHNISCHE UNIVERSITÄT DARMSTADT

- C-Paths bilden nicht die Realität ab
- Realität <> Simulation
- Gegner werden aggressiver auf Ms. Pac-Man zugehen wenn mehr Pellets vom Feld sind - Typabhängig
- Rewardfunktion: Survival 0-1 ; Pellets gegessen 0-1 ; Ghosts gegessen 0-1
  →Normalisiert, daher zwischen 0 und 1
- Survival Reward wird der Max-Leaf zurückgegeben

Ms. Pac-Man - Taktiken

TECHNISCHE UNIVERSITÄT DARMSTADT

- Survival Mode: Davoneilen der Gegner
- Feeding: Essen der Pellets
- Pursuit: Essen der Gegner
- Wechsel der Taktiken abhängig vom Zustand und eines Thresholds

TABLE II
SELECTION OF A MOVEMNET PATH

Current Tactics	Selection Method
Survival	Selects a path in which the survival rate is the highest among the paths connected to Ms. Pac-Man.
Feeding	Selects a path which contains more pellets that can be eaten from the ones connected to Ms. Pac-Man and have a survival rate at or higher than the threshold
Pursuit	Selects a path on which more ghosts can be eaten among the paths connected to Ms. Pac-Man with the threshold survival rate or higher

<Survival> Encounters with a power pellet and a ghost in the blue state are allowed
<Feeding> An encounter with a power pellet is allowed, an encounter with a ghost in the blue state is not allowed
<Pursuit> An encounter with a power pellet is not allowed, an encounter with a ghost in the blue state is allowed

Ms. Pac-Man – Experiment I

TECHNISCHE UNIVERSITÄT DARMSTADT

- Setup: 300 Simulationen, Vergleich gegen ICE Pambush3

- Ergebnis nach Punktzahl

p	min	max	mean	s.d.
System	6840	37630	24926.5	9085.71
ICE	8620	30660	20574.5	6213.21

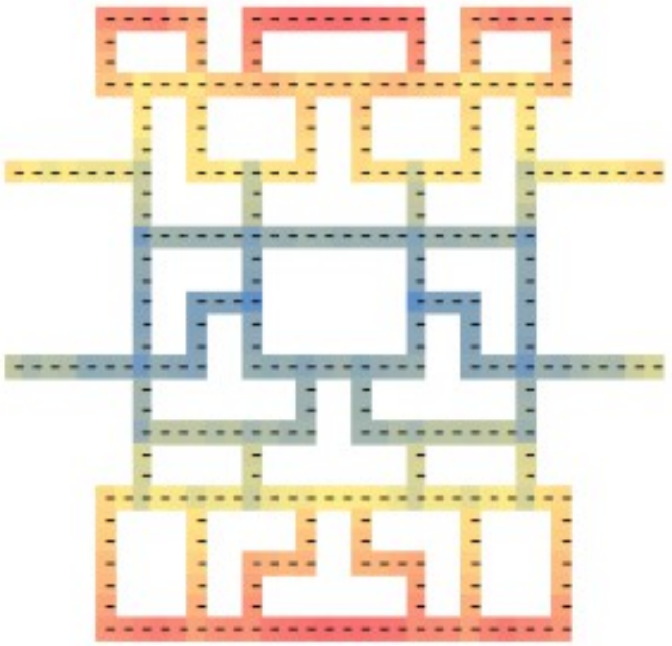
- Ergebnis nach Level

p	min	max	mean	s.d.
System	1	7	4.4	1.497
ICE	1	5	3.4	0.970

Ms. Pac-Man – Beobachtungen und Verbesserungen Experiment I

TECHNISCHE UNIVERSITÄT DARMSTADT

- Erreicht besseren Score
- Nicht stabil im Score
- Heatmap der gefährlichen Wege wurde angelegt
- Anpassung der Rewardfunktion an gefährliche Wege
- Pellets auf gefährlichen Wegen sollten zuerst gegessen werden



Ms. Pac-Man – Experiment II

TECHNISCHE UNIVERSITÄT DARMSTADT

- Evaluation nach Anpassung der Rewardfunktion

- Ergebnis nach Punktzahl

p	min	max	mean	s.d.
Experiment I	6840	37630	24926.5	9085.71
ICE	8620	30660	20574.5	6213.21
Experiment II	16800	58990	31105.6	11605.57

- Ergebnis nach Level

p	min	max	mean	s.d.
Experiment I	1	7	4.4	1.497
ICE	1	5	3.4	0.970
Experiment II	2	7	4.8	1.361

Ende

TECHNISCHE UNIVERSITÄT DARMSTADT

- Vielen Dank für die Aufmerksamkeit
- Gibt es noch Fragen?

Literaturverzeichnis

- N. G. P. Den Teuling, “Monte-Carlo Tree Search for the Simultaneous Move Game Tron,” Univ. Maastricht, Netherlands, Tech. Rep., 2011.
- S. Samothrakis, D. Robles, and S. M. Lucas, “A UCT Agent for Tron: Initial Investigations,” in Proc. IEEE Symp. Comput. Intell. Games, Dublin, Ireland, 2010, pp. 365–371.
- * D. Robles and S. M. Lucas, “A Simple Tree Search Method for Playing Ms. Pac-Man,” in Proc. IEEE Conf. Comput. Intell. Games, Milan, Italy, 2009, pp. 249–255.
- N. Ikehata and T. Ito, “Monte Carlo Tree Search in Ms. Pac-Man,” in Proc. 15th Game Progr. Workshop, Kanagawa, Japan, 2010, pp. 1–8.