

# Reinforcement Lernen für mobile Systeme

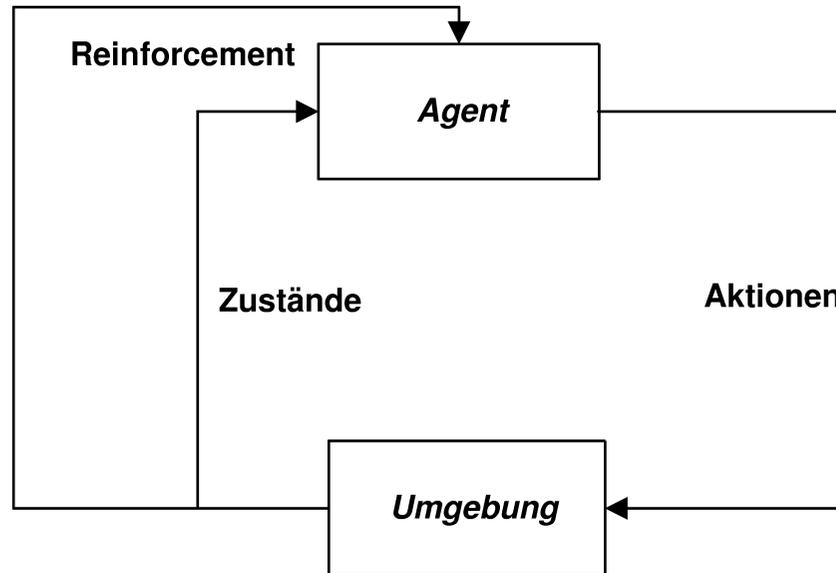


# Aufgabenstellung

Zielstellung des Themas ist die Evaluierung des Reinforcement Lernen (speziell Q-Lernen) zur Anwendung in mobilen Systemen.

Hierbei sollen die verschiedenen Arten der Methode theoretisch und experimentell verglichen und beispielhaft im Simulator sowie im realen Roboter umgesetzt werden. Die Anwendungsdomäne sei dabei das Erlernen eines Verhaltens zur Hindernisvermeidung mittels Sonarsensoren. Die Implementierung soll besonderen Wert auf die Wiedernutzbarkeit legen.

# Reinforcement-Lernmodell (theoretisch)



- Reinforcement Lernen (verstärkendes Lernen)
- Interaktion von Agent (Roboter) und Umgebung
- Steuerung mit Hilfe von Sensoren
- Bestandteile des RL: Zustände, Aktionen und Reinforcement

# Reinforcement-Lernmodell (praktisch)

- **Zustände** (Anzahl 14)

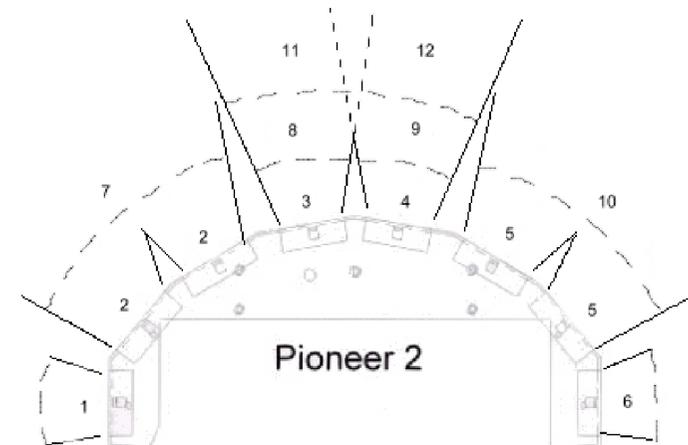
-> Aufteilung in Entfernungszonen

- **Aktionen** (Anzahl 10)

-> Einteilung gemäß Drehrichtung, Geschwindigkeit, Winkel und Fahrtrichtung

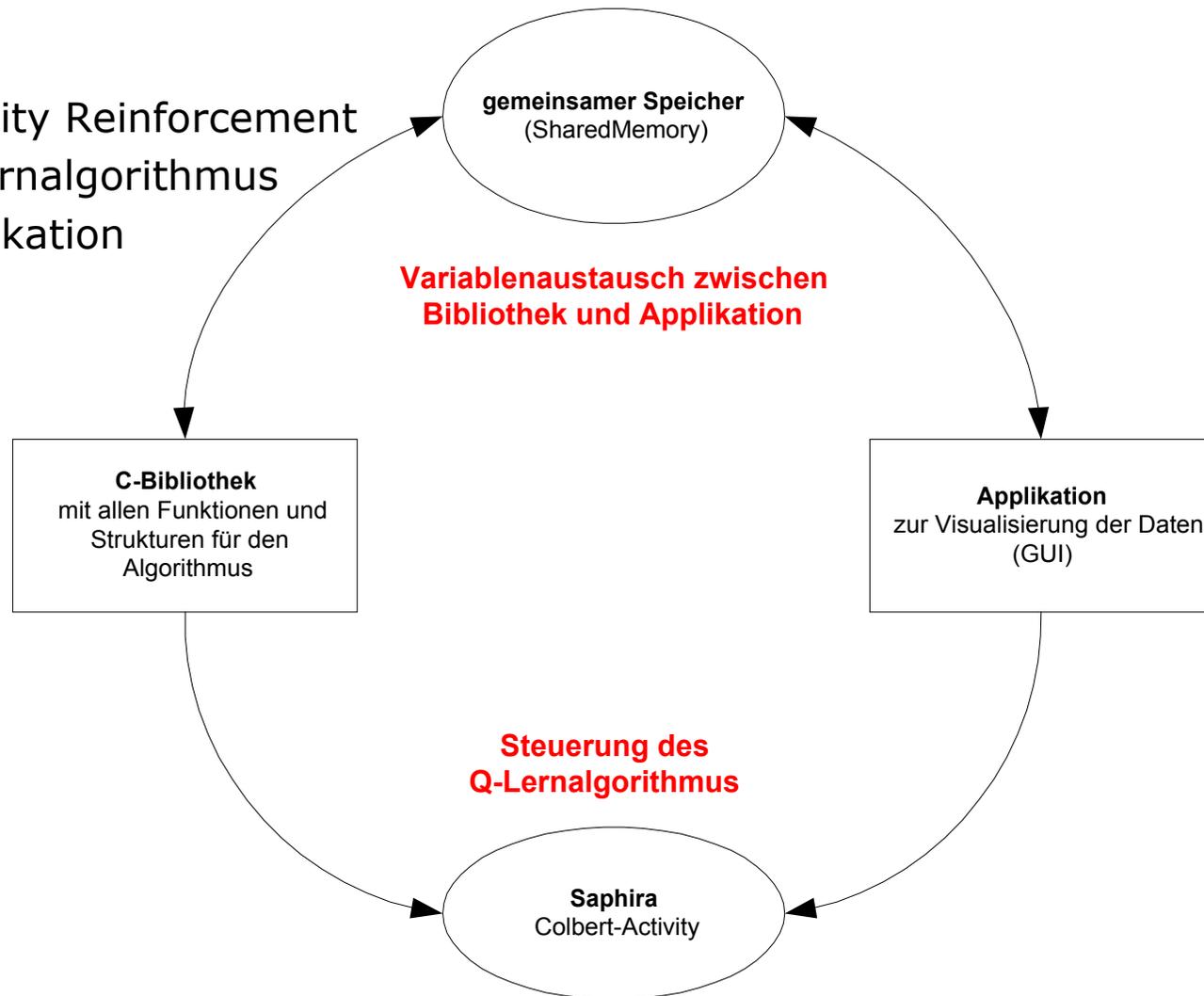
- **Reinforcement**

-> Zuordnung von Zahlenwerten je nach Zustand



# Systemarchitektur

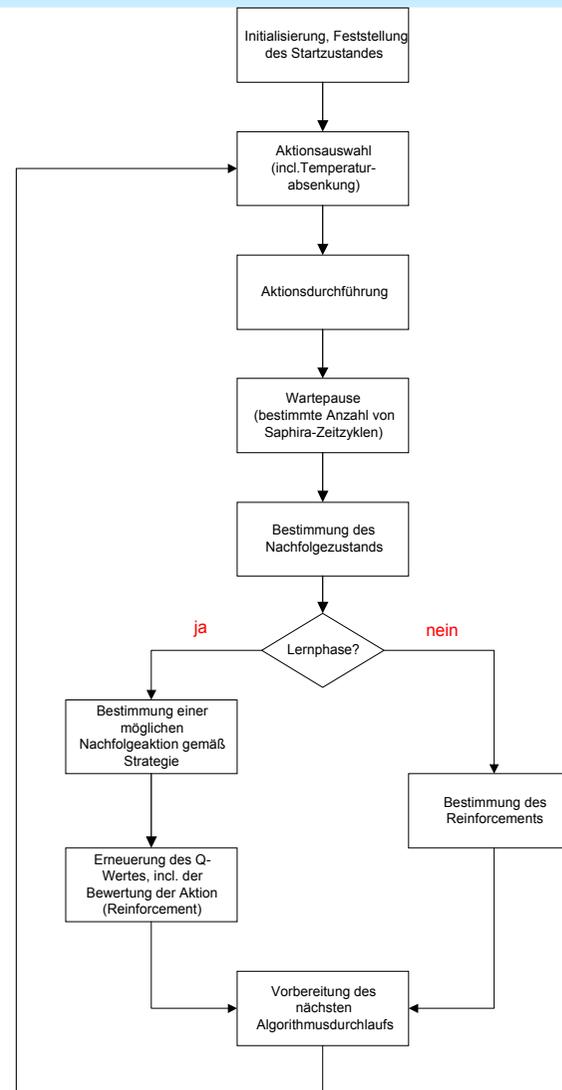
- **Modul 1:** Activity Reinforcement
- **Modul 2:** Q-Lernalgorithmus
- **Modul 3:** Applikation



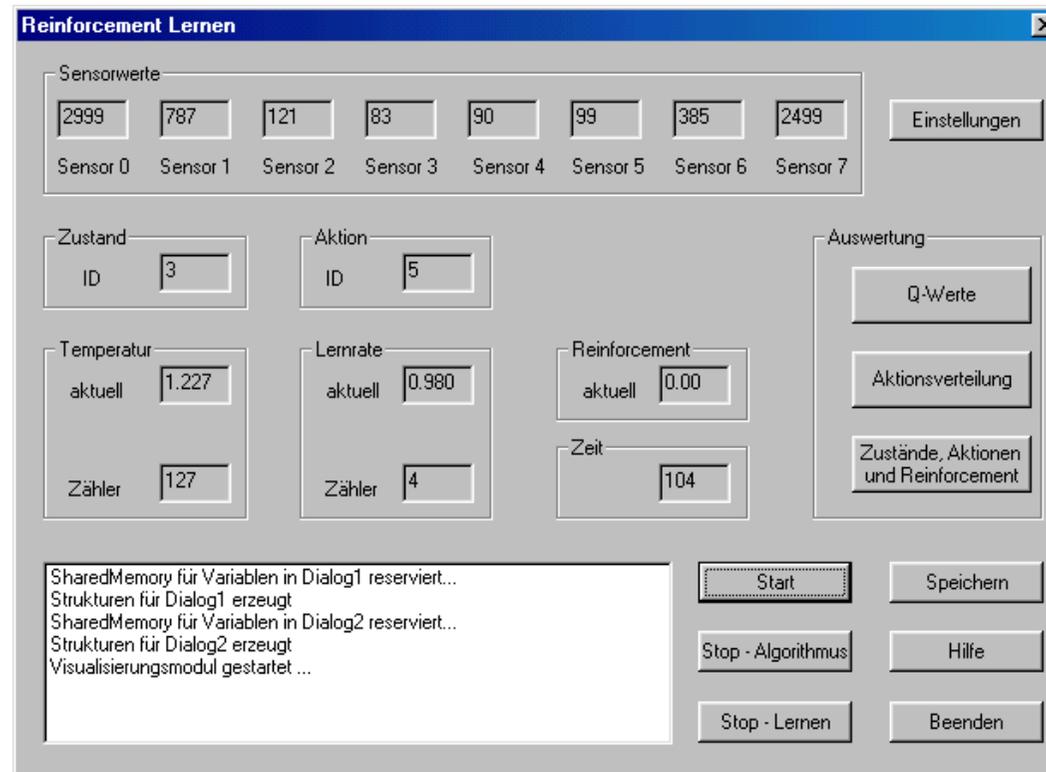
# Activity Reinforcement (Modul 1)

- Einbindung der externen Bibliothek
- Bekanntmachung von Funktionen, Variablen und Strukturen
- Steuerung der Schritte des Q-Lernalgorithmus
- Aufruf der einzelnen Funktionen

# Q-Lernalgorithmus (Modul 2)



# Applikation (Modul 3)



- Visualisierung der Daten mittels GUI
- Veränderung der Parameter im Einstellungsdialog
- Auswertung durch verschiedene Dialoge

# Testparameter

- Sensorbereiche
- Geschwindigkeit
- Winkel
- Discountfaktor  $\gamma$
- Reinforcement
- Temperatur
- Lernrate  $\alpha$
- Befehl wait im Activity Reinforcement

# Auswertung

- kein Einfluss der Rechnerperformance auf die Ergebnisse
- Bestrafung des Wandkontakts mit dem Reinforcementwert  $r = -1$
- empfehlenswerte Einstellungen für die Parameter:
  - hohe Werte für den Discountfaktor  $\gamma$
  - mittlere Wartezeit (Befehl `wait(x)`) nach Ausführung der Aktion
  - geringer Wert für kleine Geschwindigkeit
  - kleiner Winkel für eine enge und genaue Navigation
  - niedriger Startwert und langsame Absenkung für Temperatur und Lernrate

# Zusammenfassung & Ausblick

## ■ Zusammenfassung

- Hindernisvermeidung wird mittels Versuch und Irrtum gelernt
- Einsatz in Online-Navigation möglich
- Wahl der Parameter wichtiges Kriterium

## ■ Ausblick

- Verwendung anderer Lernverfahren
- Ausbau des GUI
- Online-Auswertung grafisch
- Einsatz in Schule und Lehre