Connect 4

Simon Bihlmaier, Thomas Monninger, Julian Obst

Practical Course Robotics | 2020-07-13

Objectives

- Create an environment where robots can play game “Connect 4”
- Perceive the current game state
- Choose from AI game agents to play against
- Integrate human interaction
- Realize different behaviours like pick up, handover and dropping of spheres

Challenges

- Define objectives to make the robot drop spheres at specified positions
- Source code structure to support generic robot behaviour and different states
- Perceive 2D positions of large numbers of spheres and infer grid state
- Add asynchronous, reactive way for the human to interact with the robots

Simulation World Game Setup

- 6x7 game grid opened at the top
- Two robot arms representing the players and executing the moves
- Balls in red and blue dynamically spawned on ramps

Human-Robot / Game Interaction

- AI Strategies self-implemented: No external dependencies or libraries
  - Monte-Carlo agent: Random sampling, choose action with best win ratio
  - Min-Max agent: Depth-first search, maximize own/minimize opponent reward
- Easily understandable: Human can perceive current game state
- Reactive: Change of mind possible while robot is in motion
- Keyboard-based: Press keys from 1-7 to select a column

Robot States & Optimization Objectives

- → Align to ramp → Grasp → Lift → (Handover) → Align to column → Drop →
- Modular implementation applicable to both robot arms
  - Robot arms take turns depending on current players turn
  - Robot arms cooperate in form of handover to increase range of action
- Dynamic manipulation (avoid collision, variable column selection)
- Fully decoupled from simulation space, shooting for realistic implementation

Perception

- Detect region of interest for game grid, segment per player color
- Extract circular objects by Circle Hough or Morphological Opening
- Transform pixel positions into grid cell occupancy and return game state