HALMSTAD UNIVERSITY

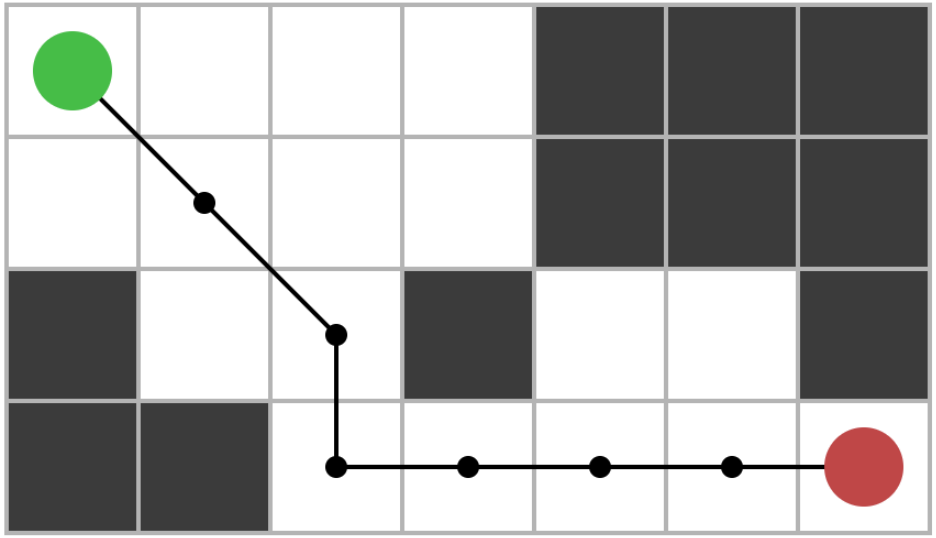
Artificial Intelligence Laboratory 2: A* (A star) Search Algorithm

DT8012 (HT16) Halmstad University

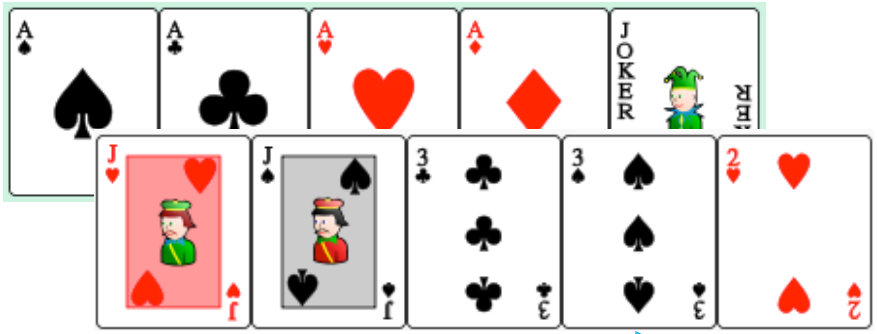
Nov 2016

Lab 2

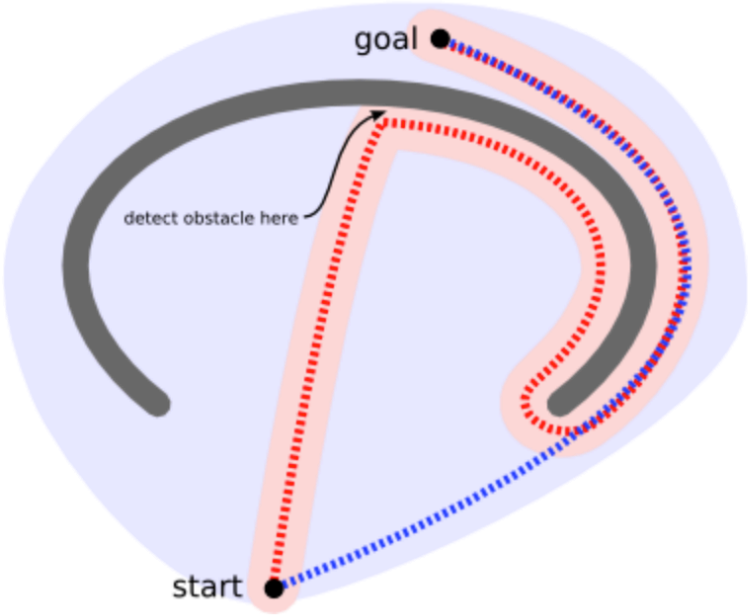
- Path Planning
  - Find a shortest path

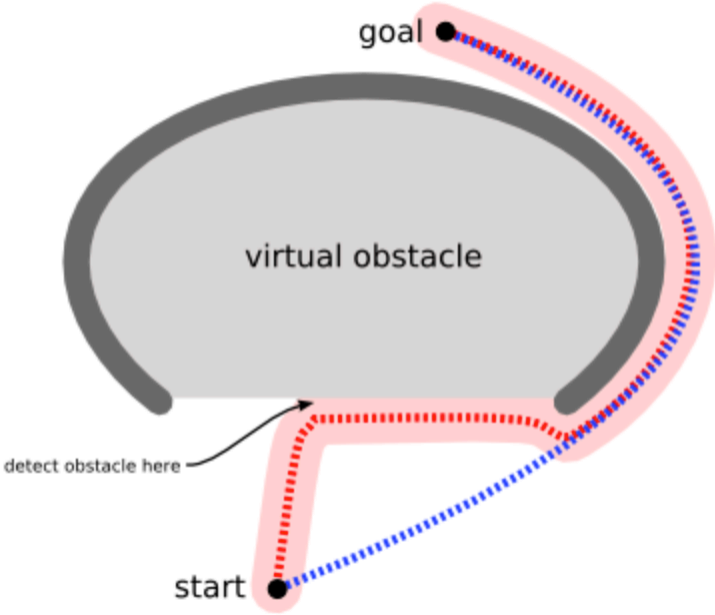


- Simplified Poker game
  - Find optimal sequence of actions



Reduce search space using A* algorithm

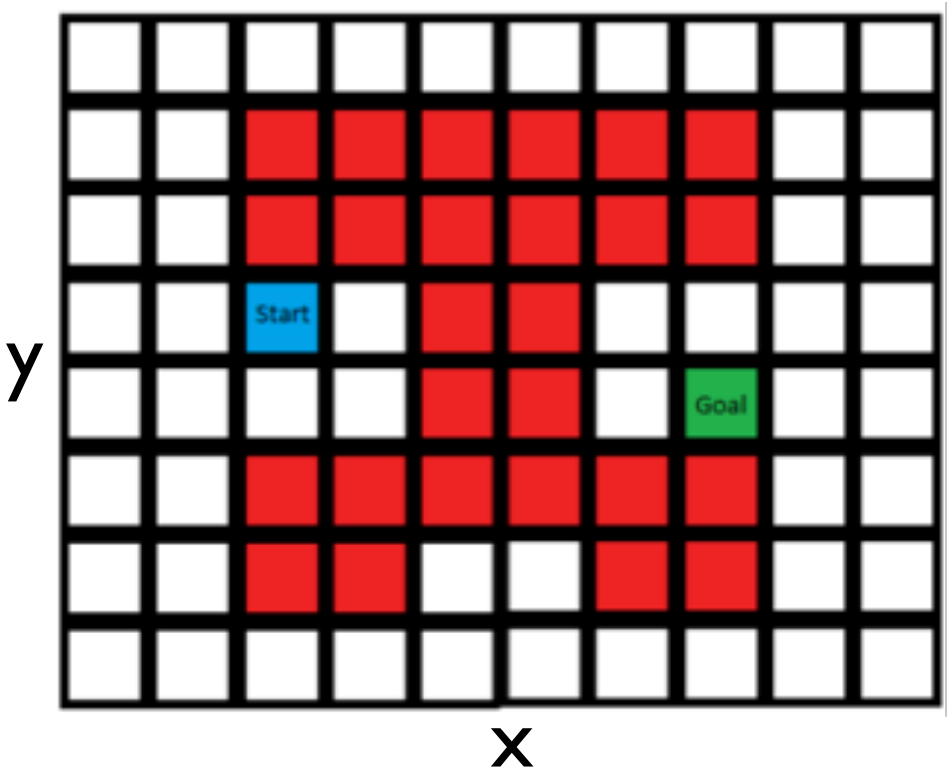




General Objective

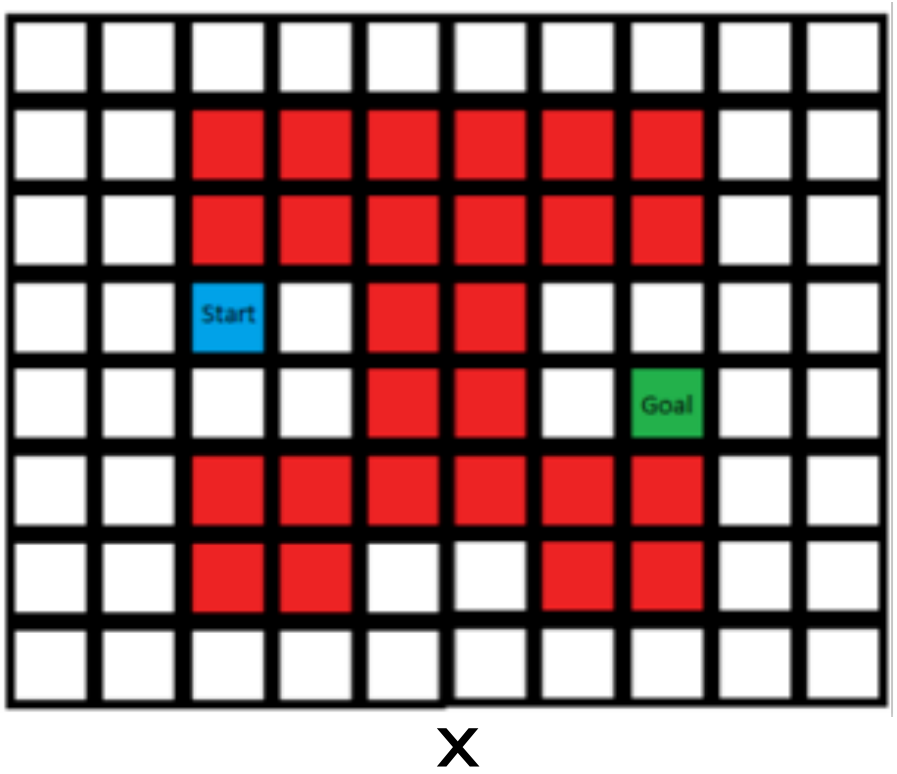
- Implement A* algorithm
  - Investigate different heuristic functions
  - Design heuristics
    - Reduce the search space
    - based on available information of the problem

Path Planning



- Red block: obstacles
- Starting point
- Goal

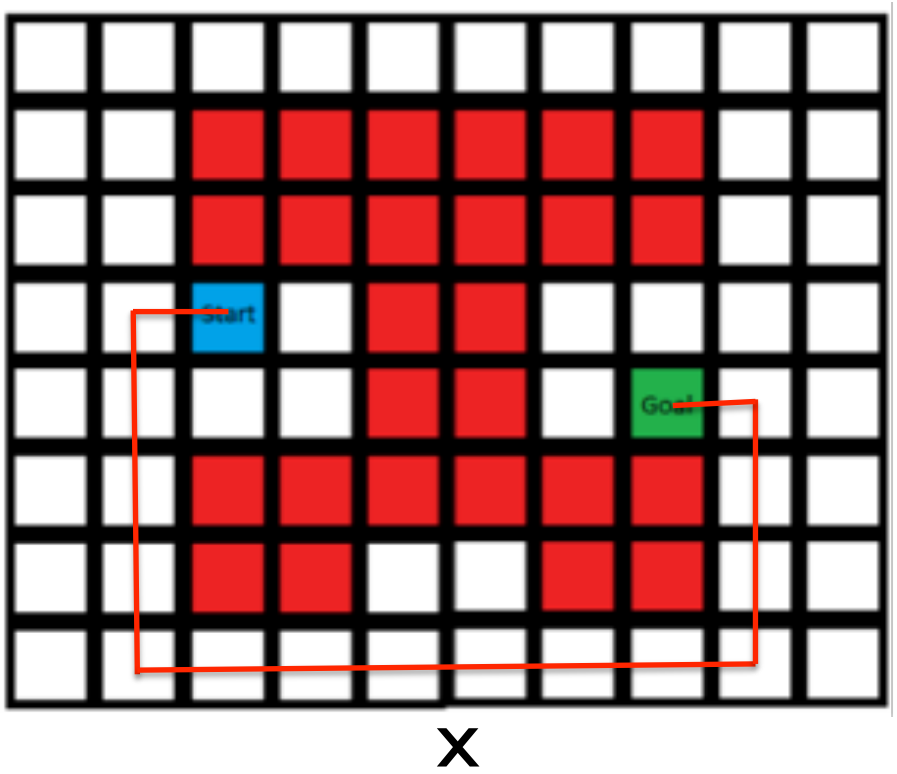
Path Planning



- Red block: obstacles
- Starting point
- Goal

1	1	1	1	1	1	1	1	1	1
1	1	-1	-1	-1	-1	-1	-1	1	1
1	1	-1	-1	-1	-1	-1	-1	1	1
1	1	-2	1	-1	-1	1	1	1	1
1	1	1	1	-1	-1	1	-3	1	1
1	1	-1	-1	-1	-1	-1	-1	1	1
1	1	-1	-1	1	1	-1	-1	1	1
1	1	1	1	1	1	1	1	1	1

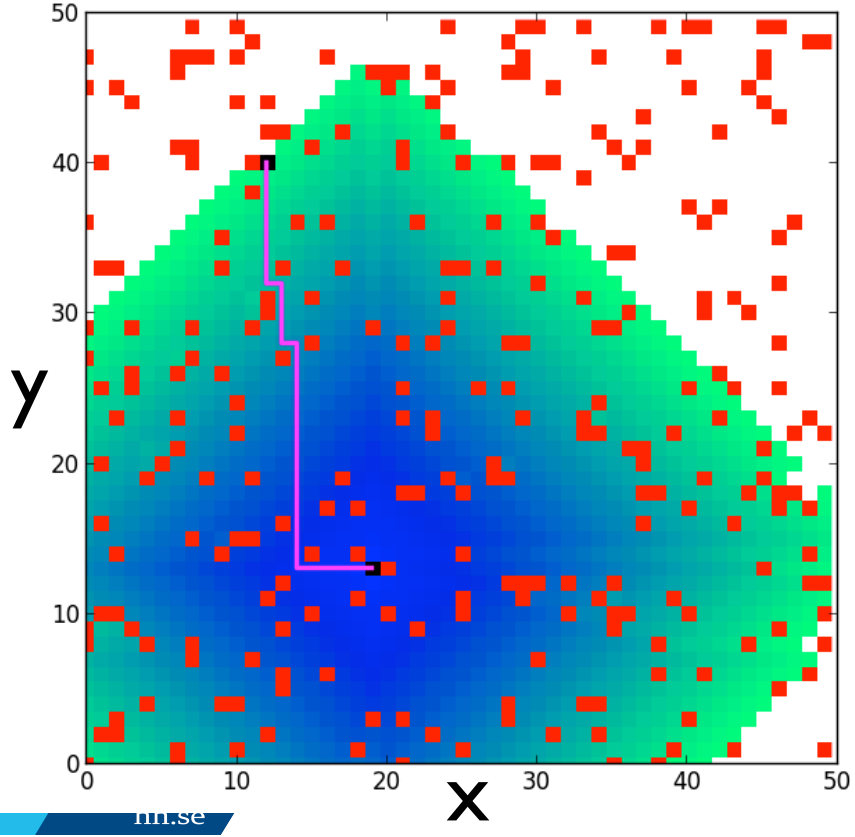
Path Planning



- Red block: obstacles
- Starting point
- Goal

- Find short path

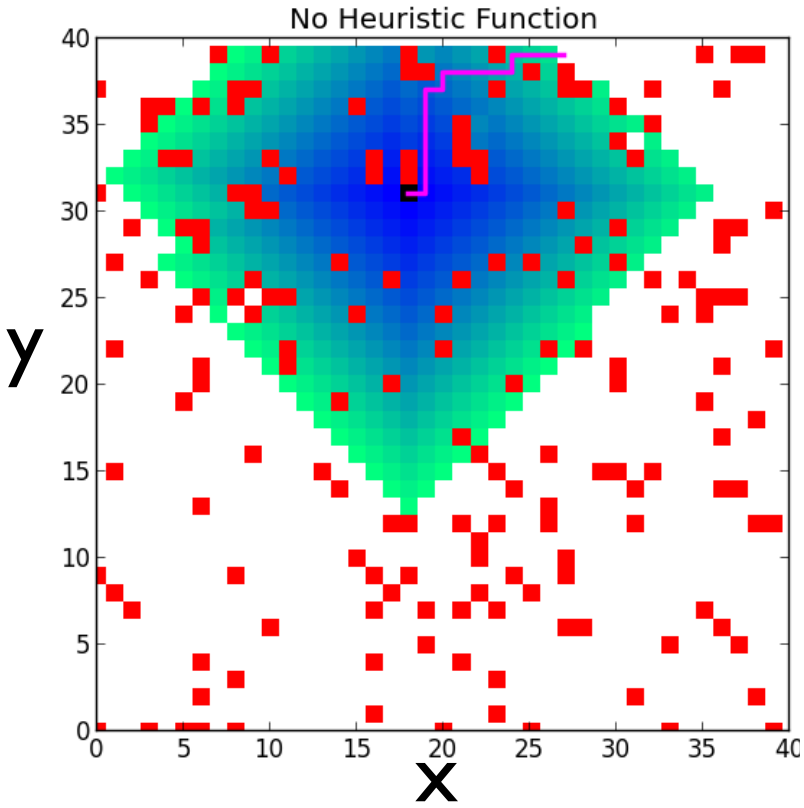
Path Planning



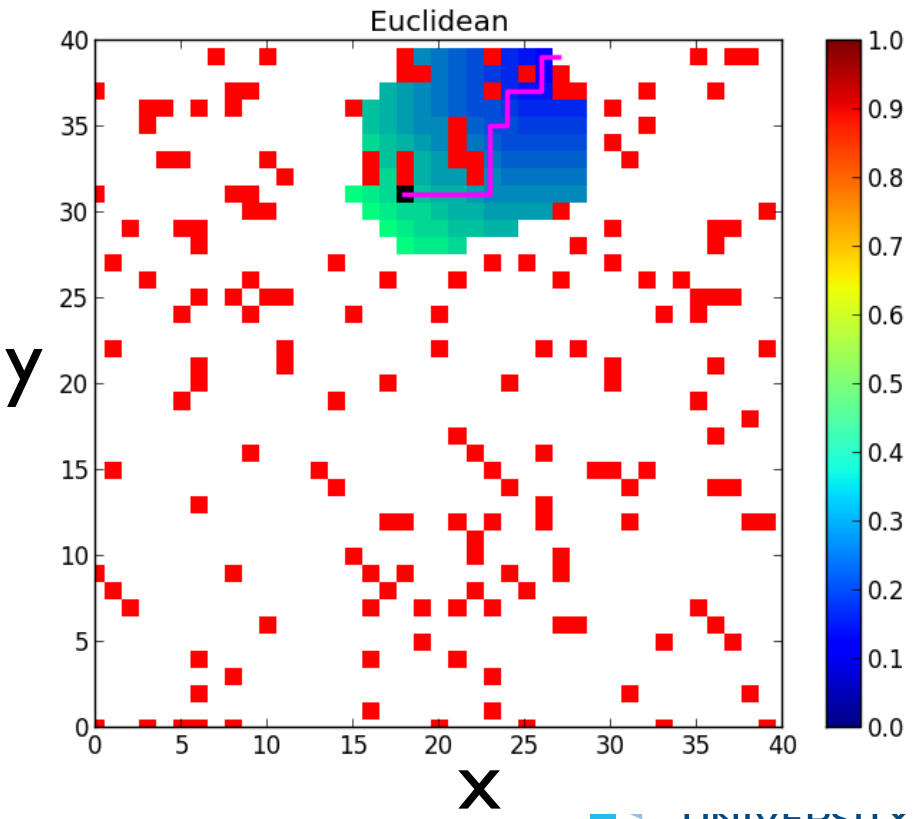
Red block: obstacles
Starting point
Goal

Find short path
– Reduce search space!

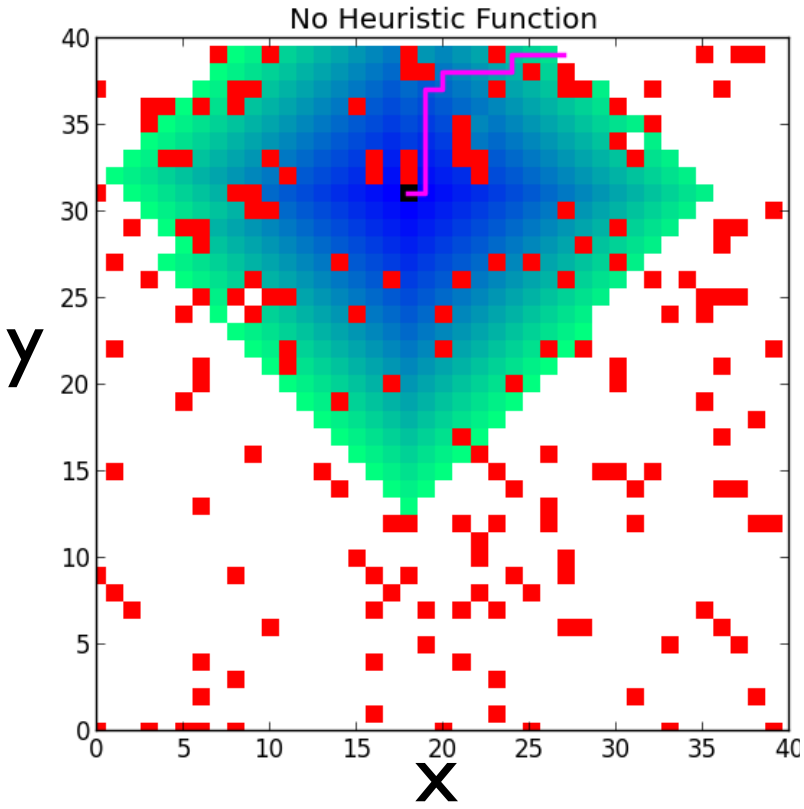
Breadth-first search



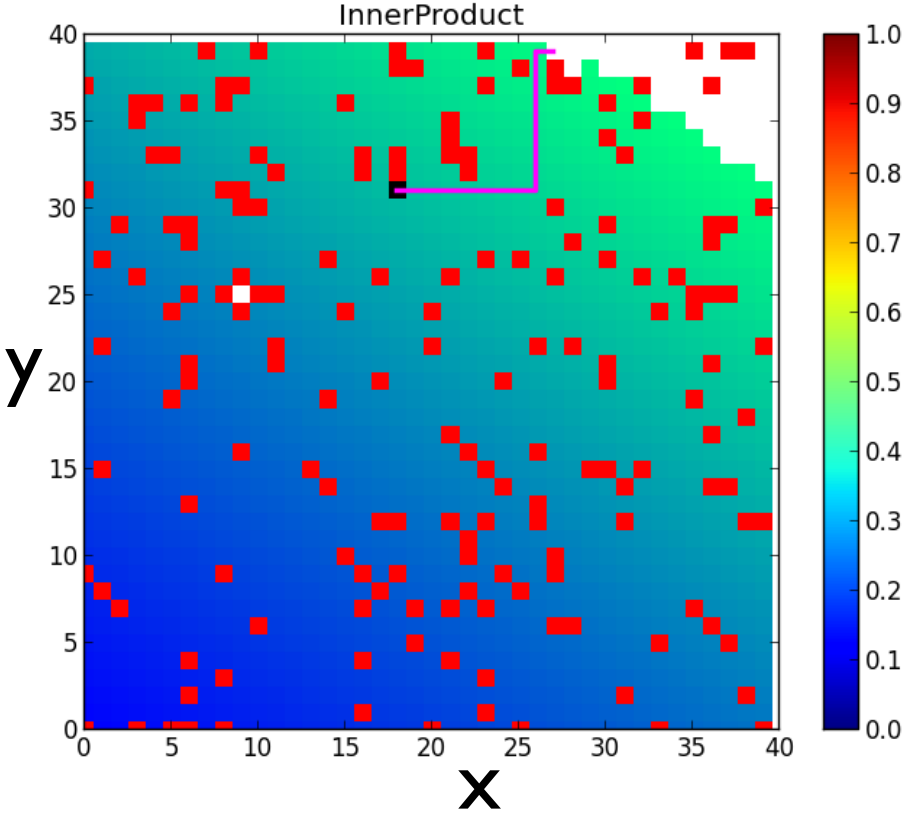
A* with Manhattan distance as heuristic



Breadth-first search

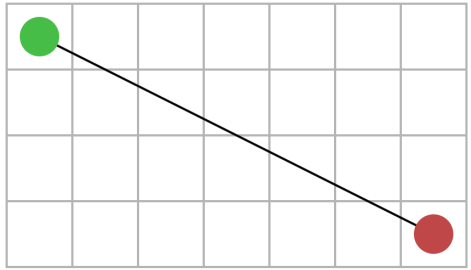


A* with Inner Product distance as heuristic

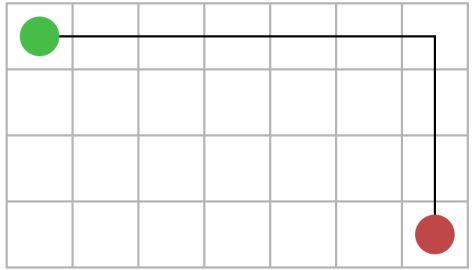


Path Planning

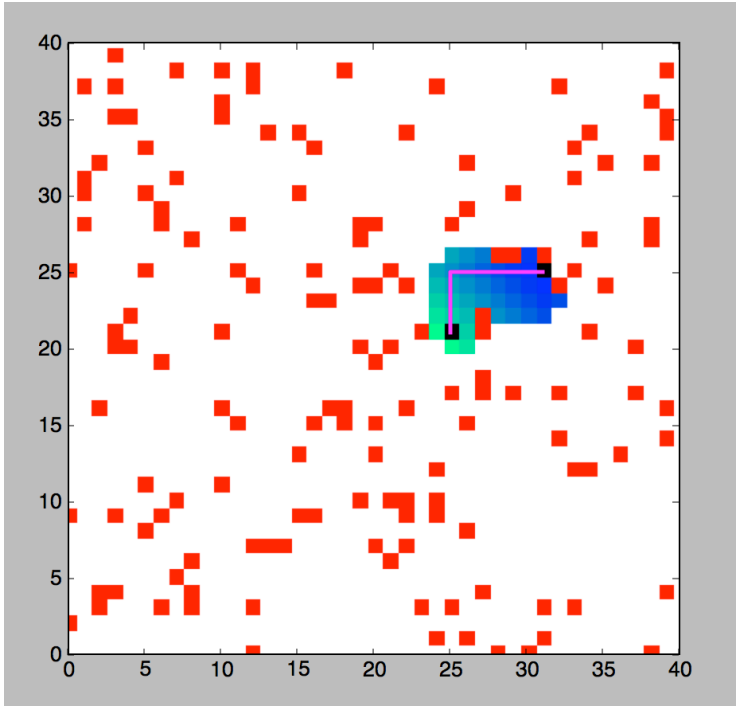
$$h(n)^2 = (n.x - goal.x)^2 + (n.y - goal.y)^2$$



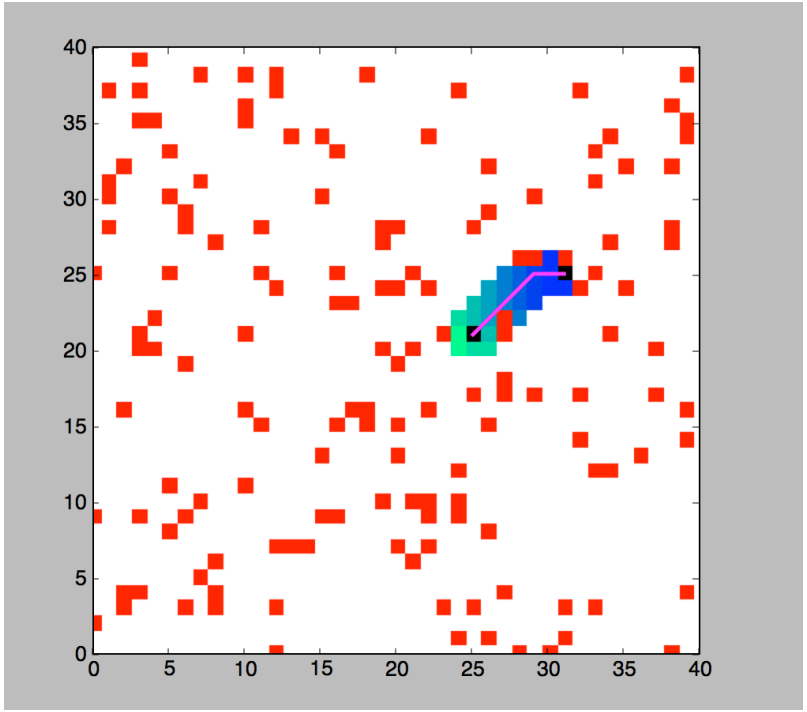
$$h(n) = |n.x - goal.x| + |n.y - goal.y|$$



- Heuristics
  - Euclidean distance
  - Manhattan distance
- General-purpose heuristics for 2d grid map

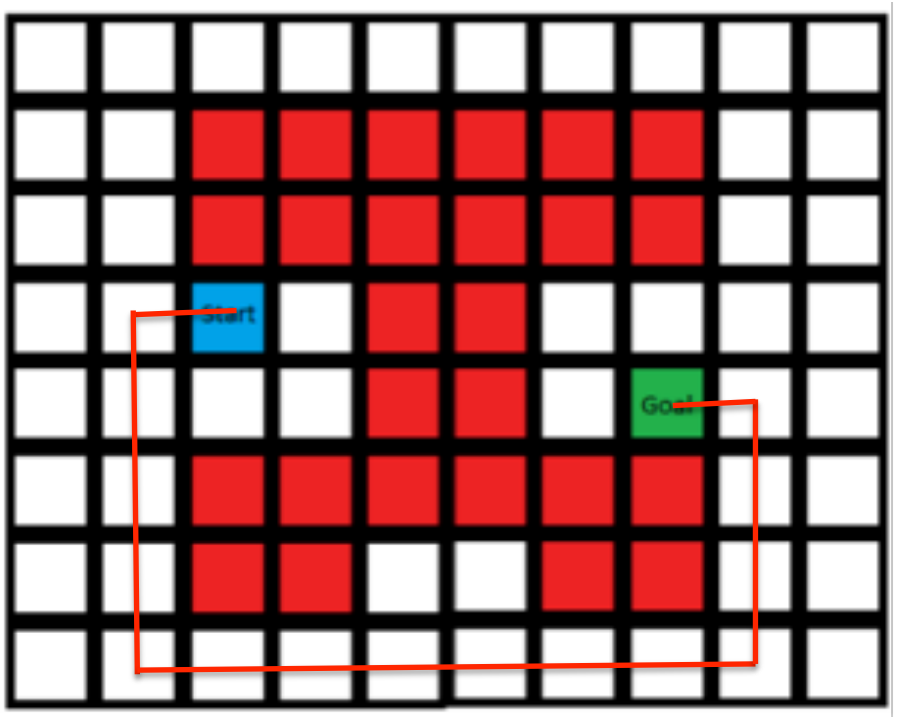


Manhattan distance as heuristics



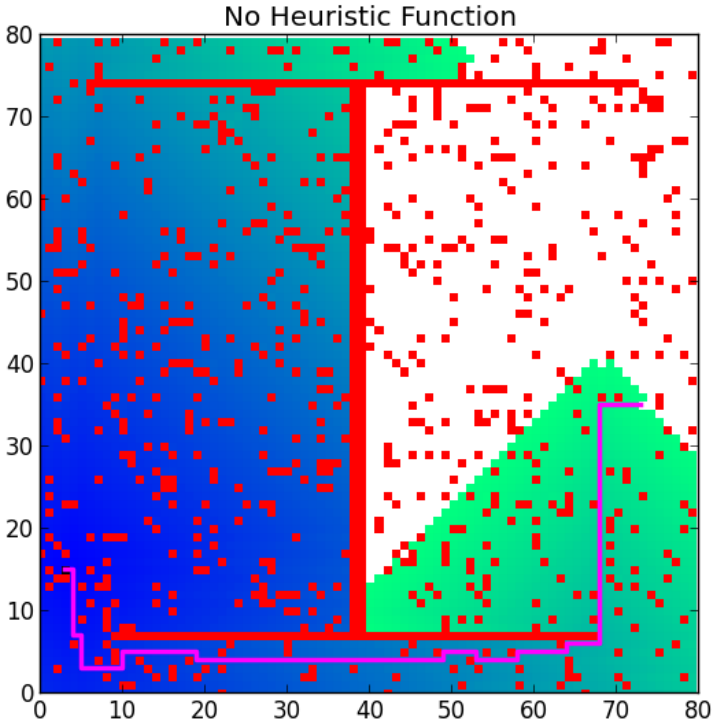
Euclidean distance as heuristics

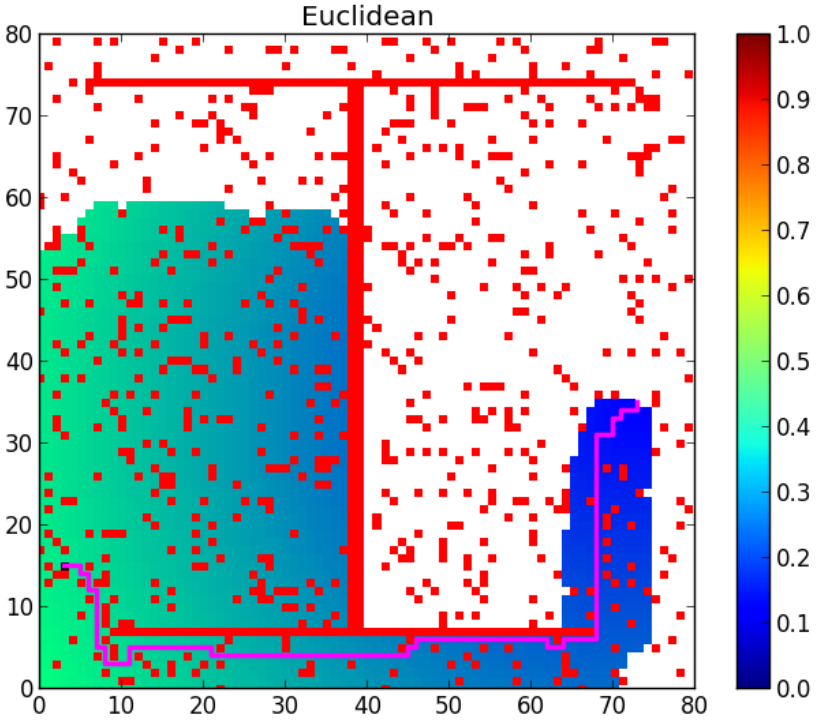
Environment-specific Heuristic



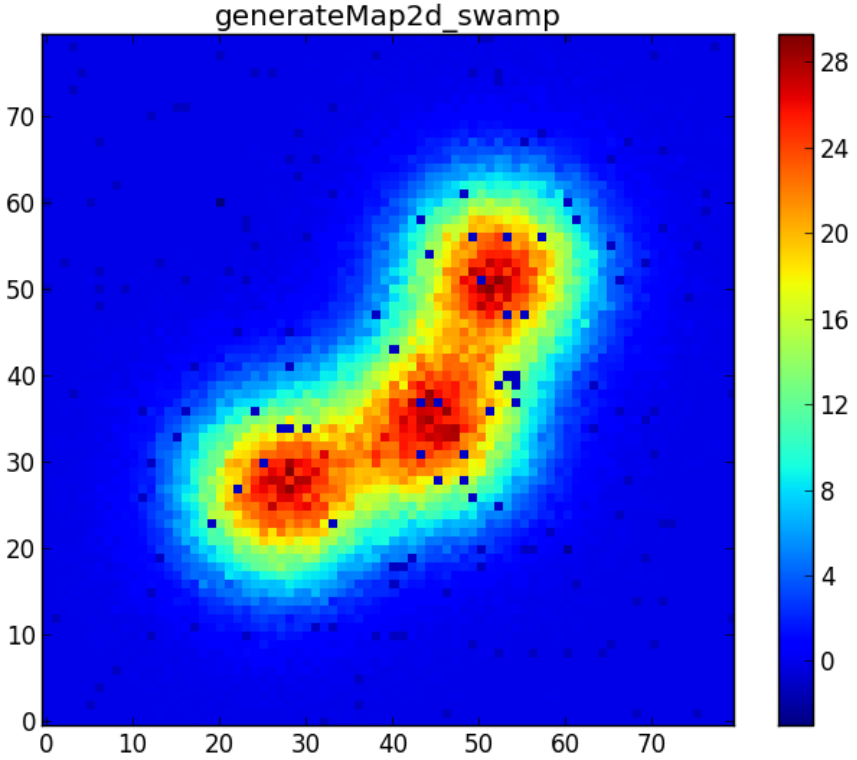
- Env. Information
  - Starting point at left side and ending point at right side
  - The environment contains a 'ㅗ' shaped obstacle
    - Y coordinate of upper and lower edges of 'ㅗ'

Path Planning in ‘工’ environment





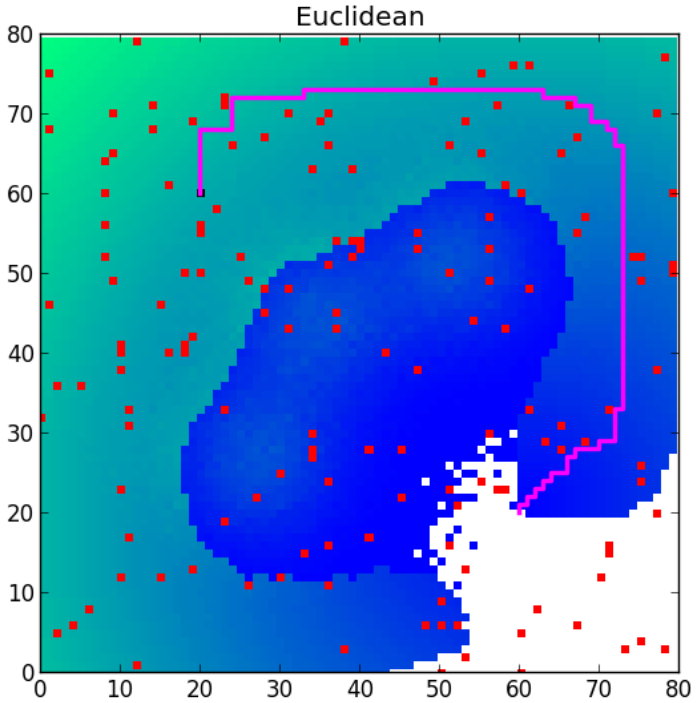
Path Planning in Env. with swamp

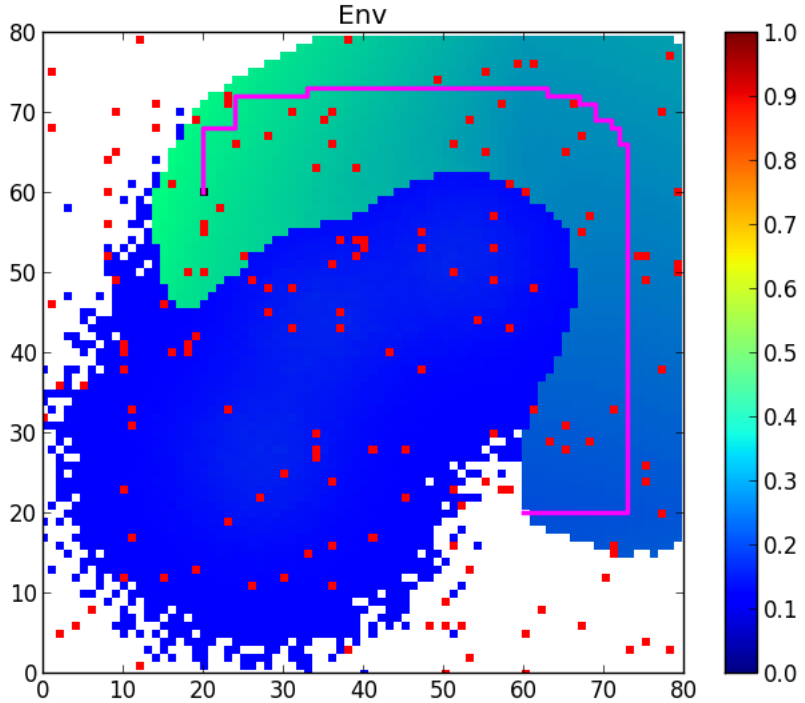


- Env. Information
  - Consists of 3 repulsive fields
  - Center of the fields are available

HALMSTAD UNIVERSITY

Path Planning in Env. with swamp





Expectation

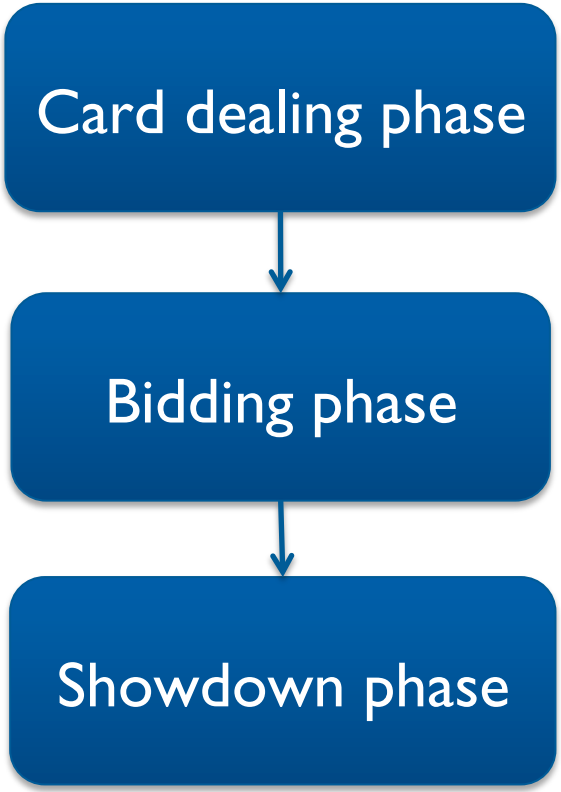
- A* Star algorithm
- Try different type heuristics
- 3 types of environment
  - Map with obstacles randomly placed
  - Map with ‘工’ obstacles and randomly placed obstacles
  - Map with ‘swamp’

Task 2 Poker game

- Rules: slightly more complex than the first lab!
- Search optimal solution
  - Breadth-first search
  - A* Star algorithm with heuristic
- Objective
  - Design a special heuristic function that reduces the search space

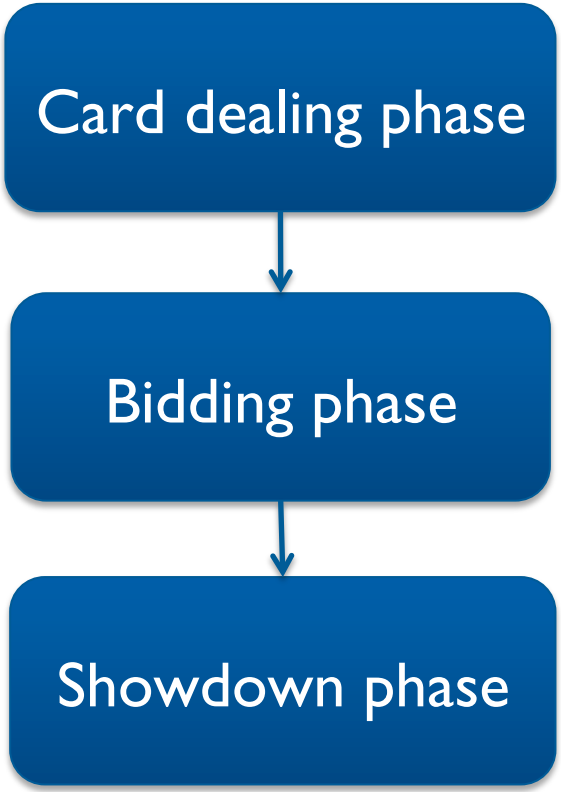
Game flow (1st lab)

- Card dealing phase
  - Assign 3 cards to agents
- Bidding phase
  - Amount $0-50
  - Regardless of how much other players bet
- Showdown phase



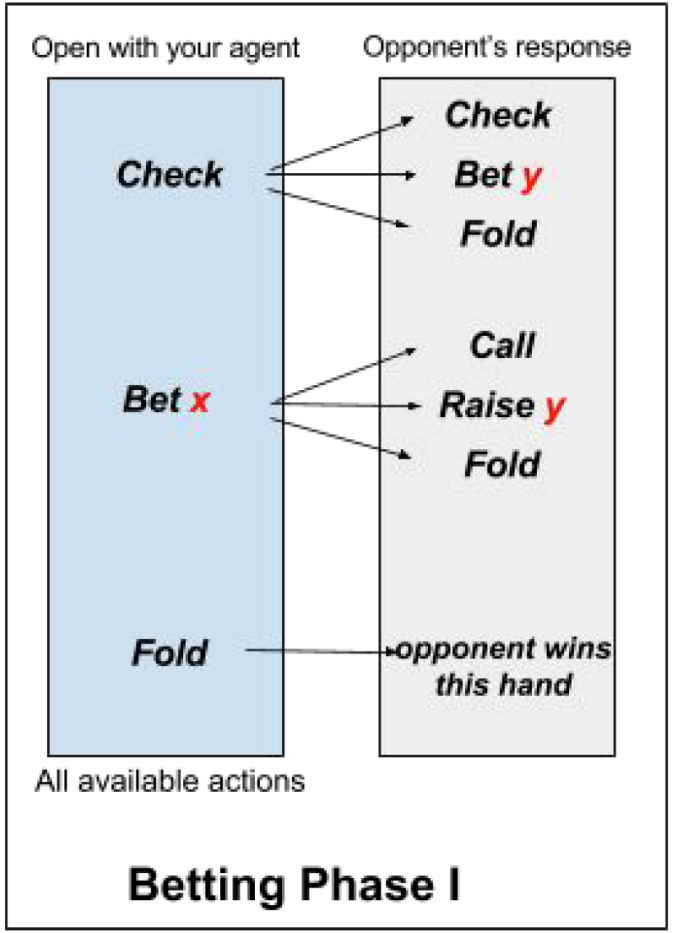
Game flow

- Card dealing phase
  - Assign 5 cards to players
- Bidding phase
  - Search sequence of actions
- Showdown phase



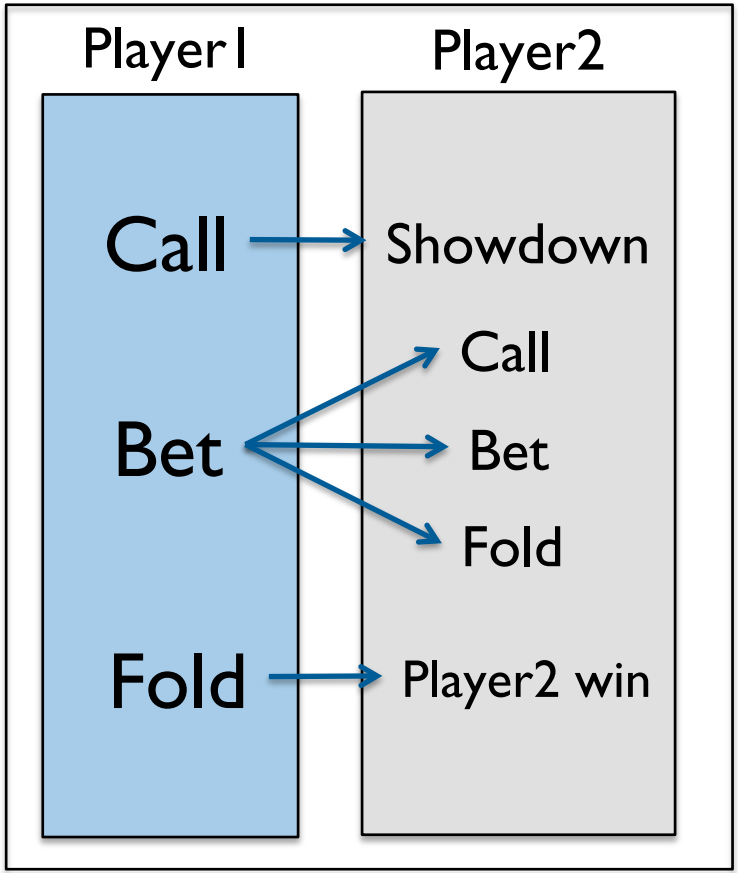
Traditional Poker game

- Actions Available
  - Bet
  - Raise
  - All In
  - Check
  - Call
  - Fold
  - Showdown



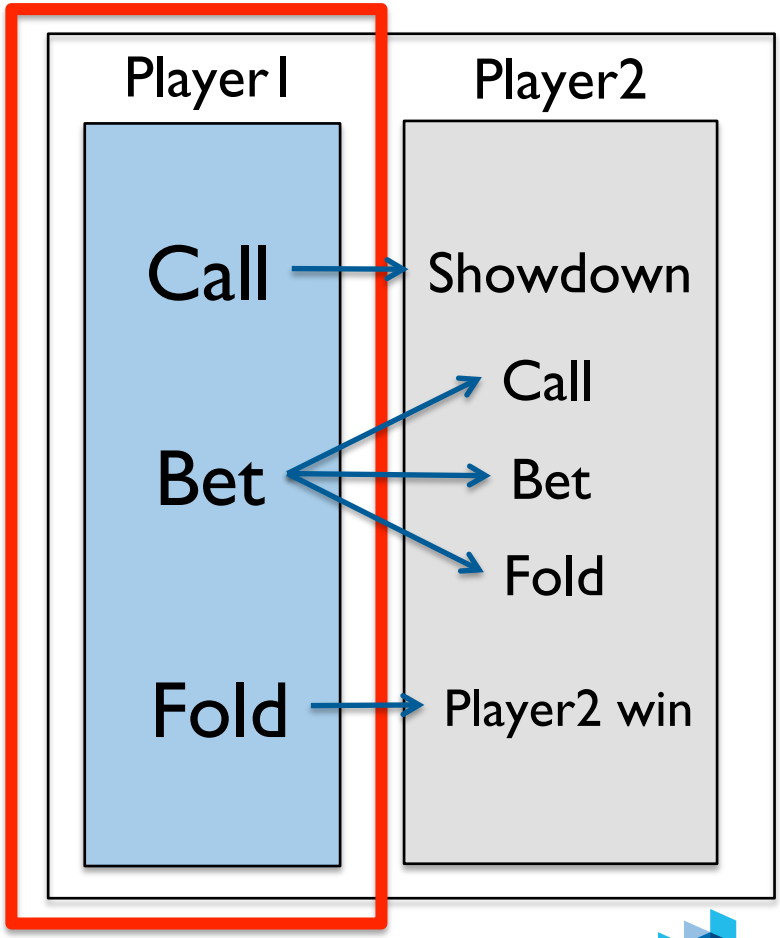
Simple Poker Game

- Action Available
  - Bet x coins
    - Regardless of how much opponent bet
  - Call
    - Putting 5 coins in the pot and show hand
  - Fold



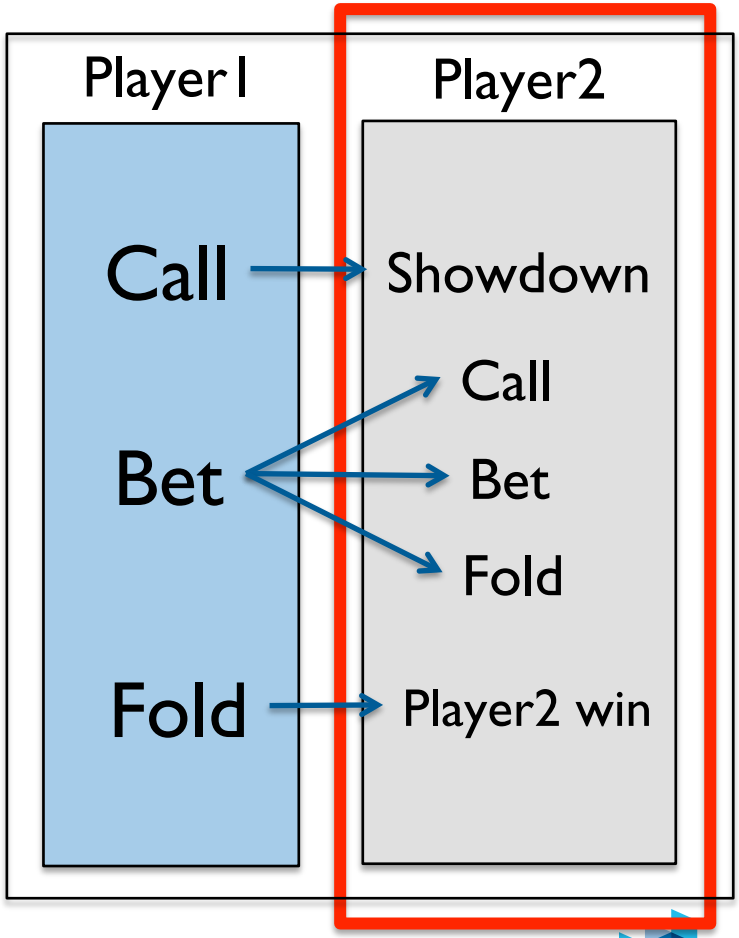
Simple Poker Game

- Always start with your agent
- Action Available
  - Bet x coins
  - Call
  - Fold



Simple Poker Game

- Always start with your agent
- Action Available
  - Bet x coins
  - Call
  - Fold



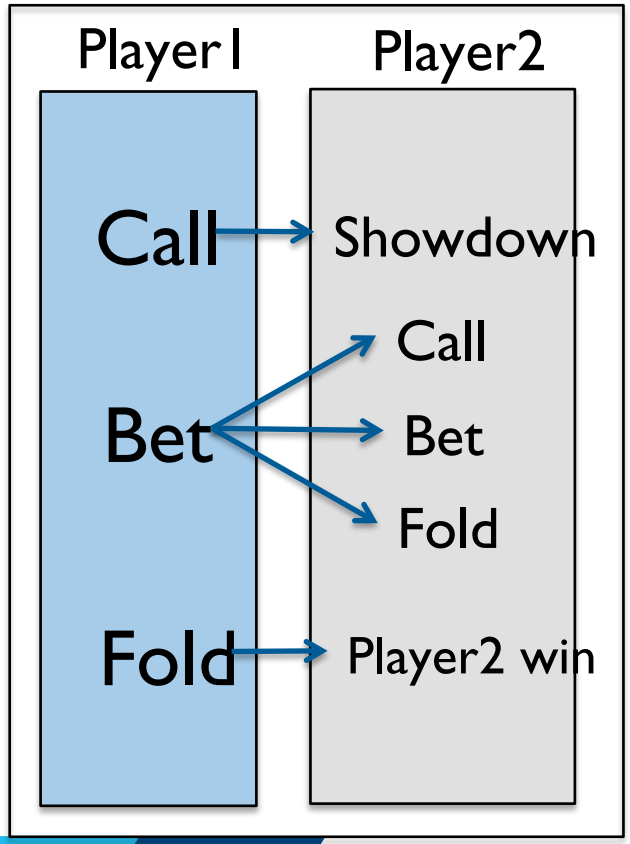
Tasks

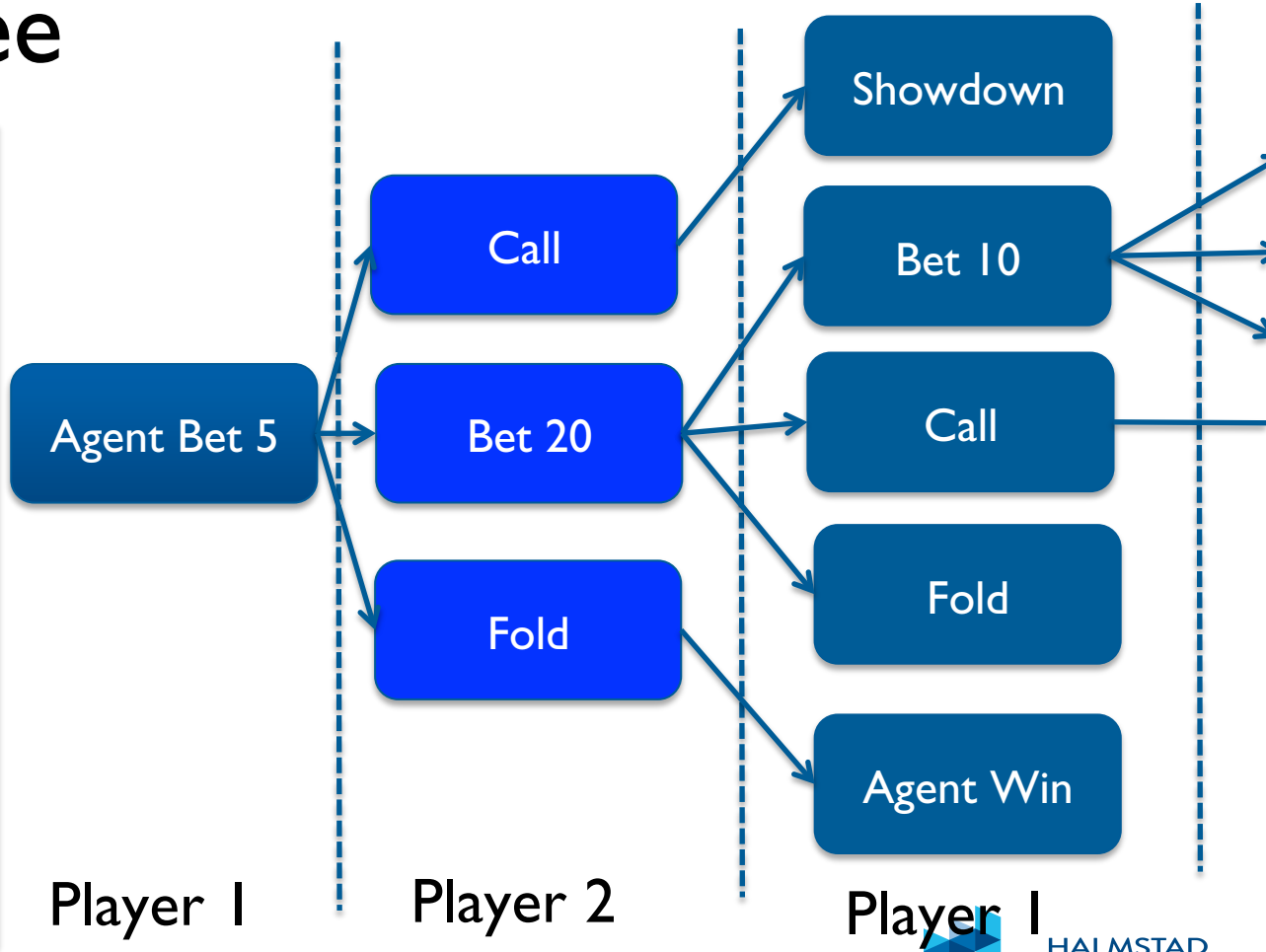
- Build environment of the game
  - Hand evaluation function for 5 cards
  - Function for updating the state of the game
    - Number of hands played, coins left for both agent, coins in the pot, current hand for both agent etc.
- Implement two fixed agents playing against each other

Expectation / Tasks

- Implement environment of the game
  - Evaluation function
  - Update state of the game
- 2 fixed agent playing against each other
- Breadth-first search
- A* algorithm
  - Heuristics: Find optimal solution as breadth first search provided and expand less nodes

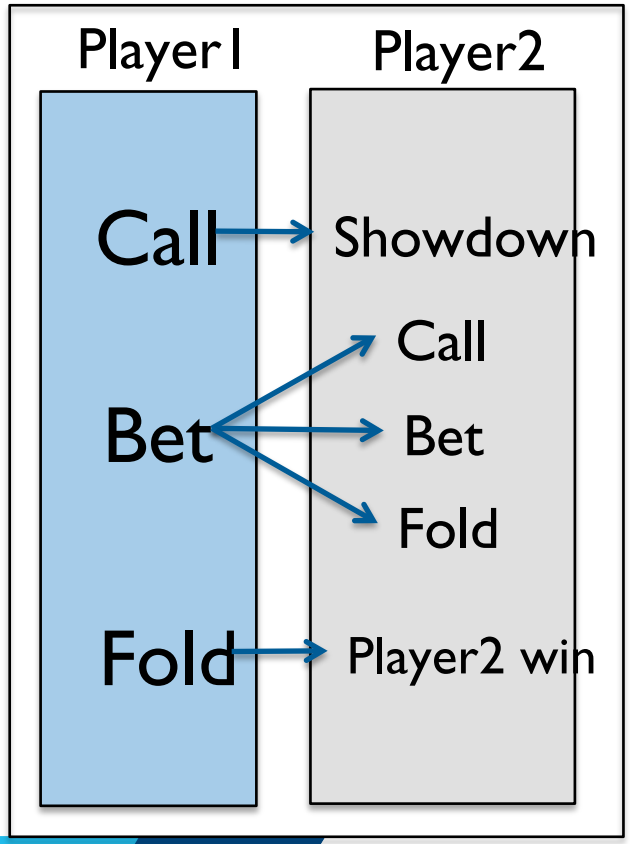
Decision Tree

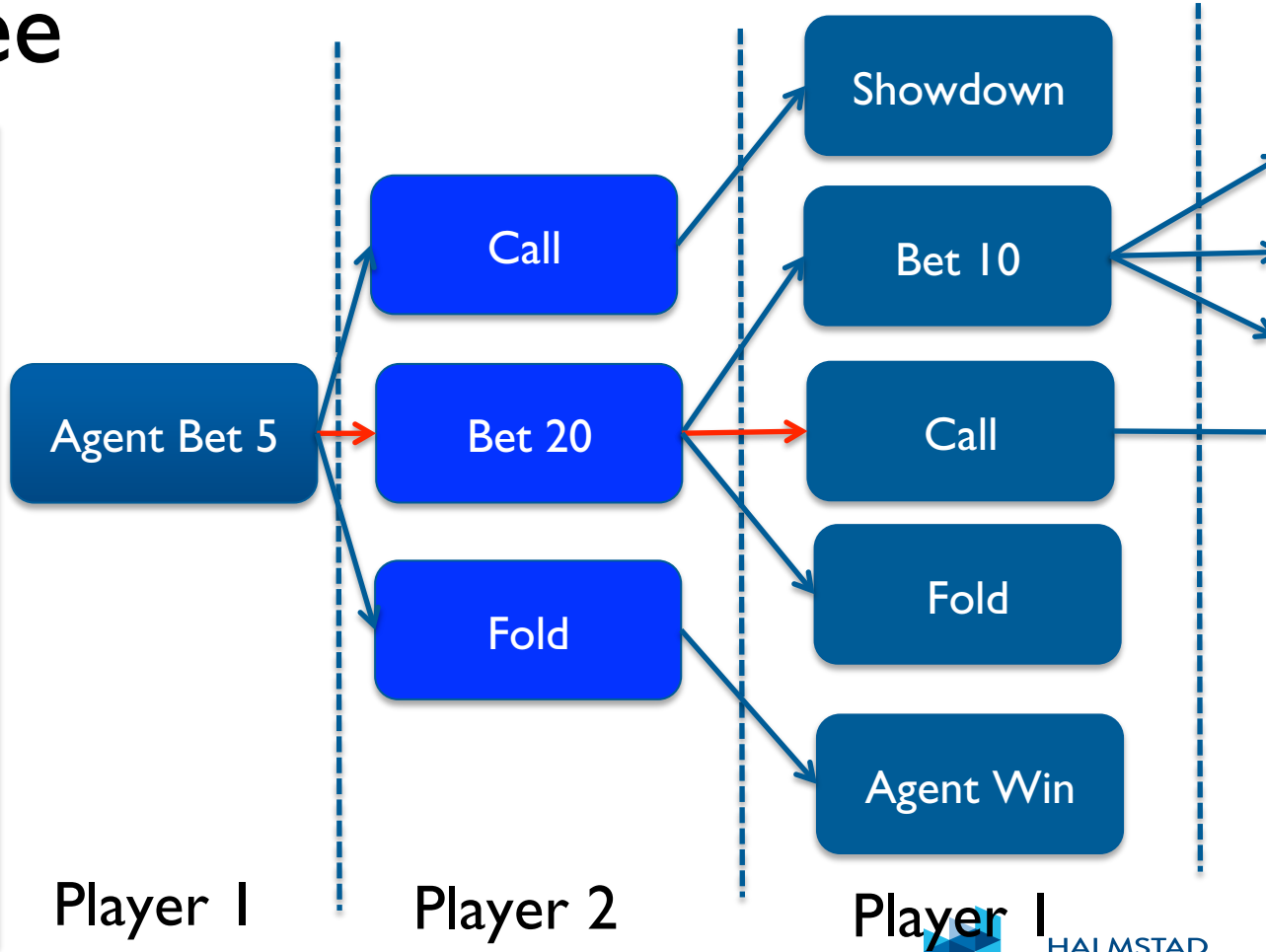




HALMSTAD UNIVERSITY

Decision Tree





pokerStrategyExample(...)

Information/input	Details
playerAction	Bet, Call or Fold
playerActionValue	the amount of coins used to Bet or Call
playerStack	total amount of coins your agent have
agentHand	current opponent's hand type
agentHandRank	current opponent's hand rank
agentStack	total amount of coins your opponent have

Expectation / Tasks

- Use A* to generate a series of actions that maximize your player's winning
  - given known strategy of the opponent and the complete information of the game
- Start with breadth-first search
- Design heuristic function that reduce the search space

Expectation

- Implement environment of the game
  - Evaluation function
  - Update state of the game
- 2 fixed agent playing against each other
- Breadth-first search
- A* algorithm
  - Heuristics: Find optimal solution as breadth first search provided and expand less nodes

Grading

- Pass/fail/extra credits
- Submit your lab on Blackboard
  - a short report about what you have done
  - Code
- Yuantao Fan
  - yuantao.fan@hh.se
  - E513