



Organic Pathfinding

EAAI-17

Joshua Eckroth

Klaus Humpert
Lauf-Spuren
edition esefeld
& traub
2007



CSCI 431 / Artificial Intelligence

csci431.artifice.cc

Joshua

Notes

- Syllabus
- Turing Test
- Searching for a solution
 - Uninformed search
 - Informed search
- Finding a plan
- PDDL
- Beating an adversary
- Evolving solutions
- Representing expert knowledge
 - Drools rules engine
- Logic programming with Prolog
 - Prolog unification
 - Prolog resolution
 - Prolog examples

Assignments

- A01: Organic pathfinding**
due Sep 7, 11:59pm
- A02: Git planner**
due Sep 12, 11:59pm
- A03: Connect Four AI**
due Sep 19, 11:59pm
- A04: Wedding seat assignment**
due Sep 26, 11:59pm
- A05: Student advisor**
due Oct 3, 11:59pm
- A06: Prolog Pokédex**
due Oct 26, 11:59pm
- A07: Automated jury**
due Nov 9, 11:59pm
- A08: NFL play-by-play**
due Nov 21, 11:59pm
- A09: Recipe classification**
due Nov 30, 11:59pm

CSCI 431

Fall 2016

jeckroth@stetson.edu

Office hours:
Mon/Wed 12-2:30

```
closedset←[]
openset←[starting-state]
parents←{}

while(openset != [])
    state←pick-one(openset)
    openset←openset - [state]
    closedset←closedset + [state]
    if(state == goal)
        return reconstruct-path(parents)
    else
        foreach(new-state←reachable-states(state))
            if(new-state not in closedset and new-state not in openset)
                parents[new-state]←state
                openset←openset + [new-state]

(if we made it here, oops, couldn't find goal!)
```

```
closedset←[]
openset←[starting-state]
parents←{}
```

```
while(openset != [])
    state←pick-one(openset)
    openset←openset - [state]
    closedset←closedset + [state]
    if(state == goal)
        return reconstruct-path(parents)
    else
        foreach(new-state←reachable-states(state))
            if(new-state not in closedset and new-state not in openset)
                parents[new-state]←state
                openset←openset + [new-state]
```

(if we made it here, oops, couldn't find goal!)

only interesting bit

pick-one(openset)

Suppose openset is ordered (ascending) according to $f(s) = g(s) + h(s)$.
Thus, first state in openset is “best.”

$g(s)$: actual cost of path from start-state to s

$h(s)$: heuristic (guessed) cost of path from s to goal

max openset size: keep only best N states in openset

pick-one(openset)

Depth-first search

$g(s) = -1 + g(\text{parent})$

$h(s) = 0$

$\text{max openset} = \infty$

Breadth-first search

$g(s) = 1 + g(\text{parent})$

$h(s) = 0$

$\text{max openset} = \infty$

Best-first search

$g(s) = 0$

$h(s) = \text{heuristic}$

$\text{max openset} = \infty$

A* search

$g(s) = \text{path cost so far}$

$h(s) = \text{admissible heuristic}$

$\text{max openset} = \infty$

Beam search

$g(s) = \text{path cost so far}$

$h(s) = \text{admissible heuristic}$

$\text{max openset} = N$

“Human” search

$g(s) = ?$

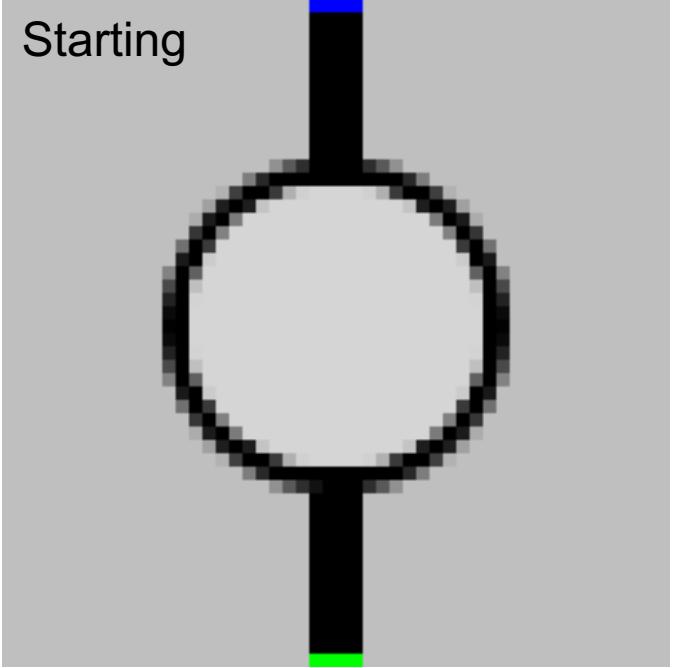
$h(s) = ?$

$\text{max openset} = ?$

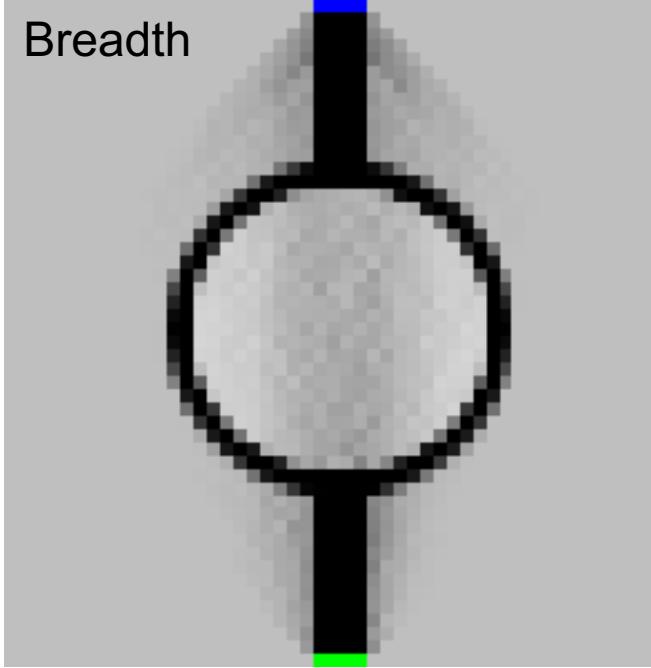


Klaus Humpert
Lauf-Spuren
edition esefeld
& traub
2007

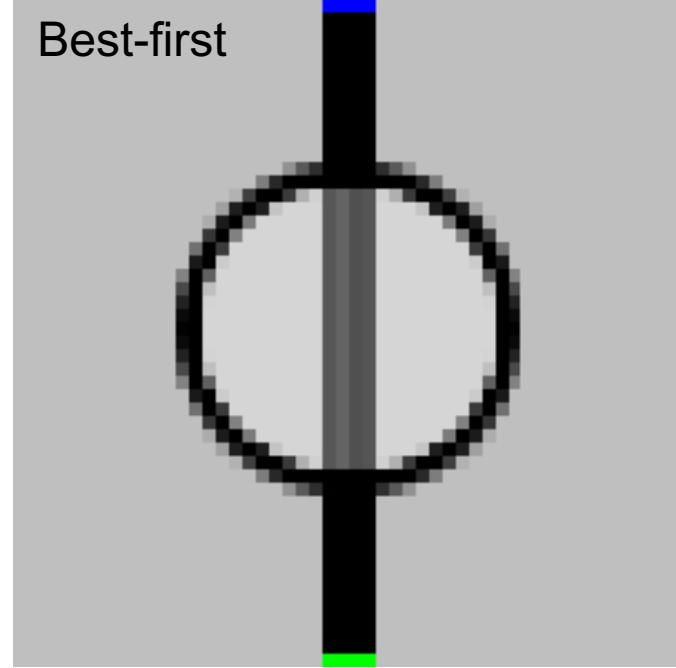
Starting



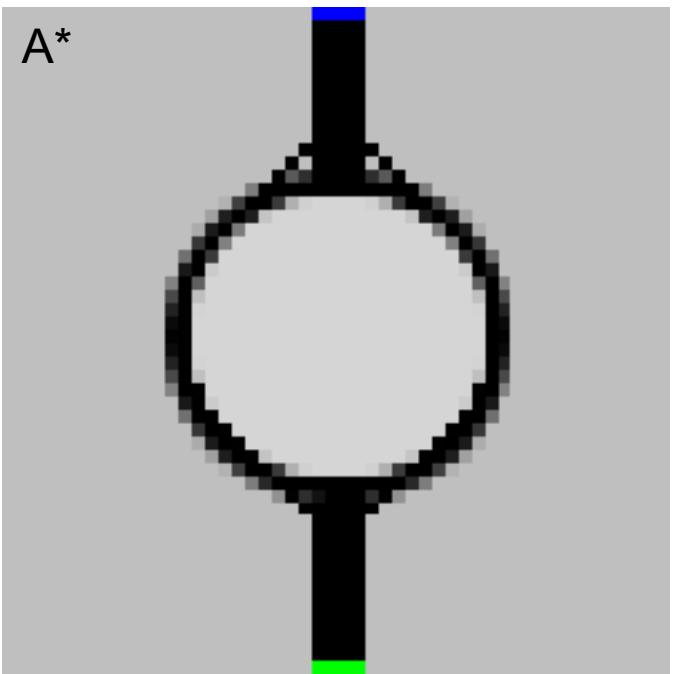
Breadth



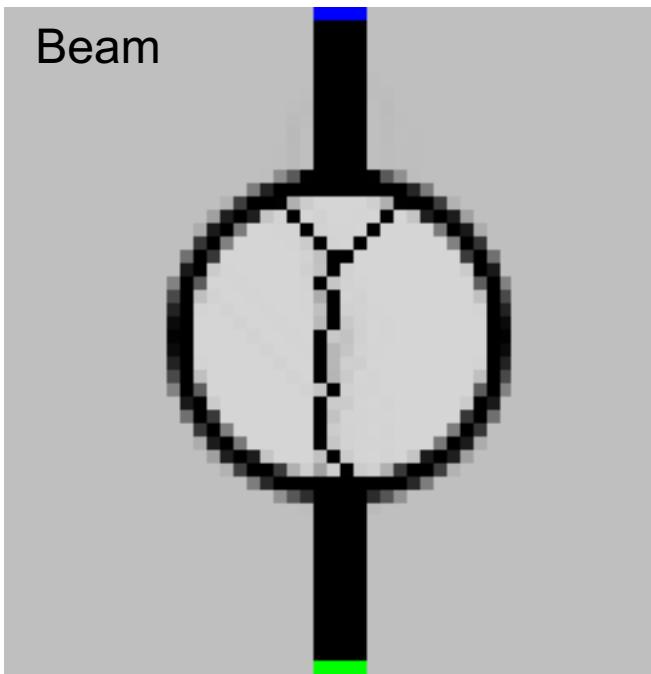
Best-first



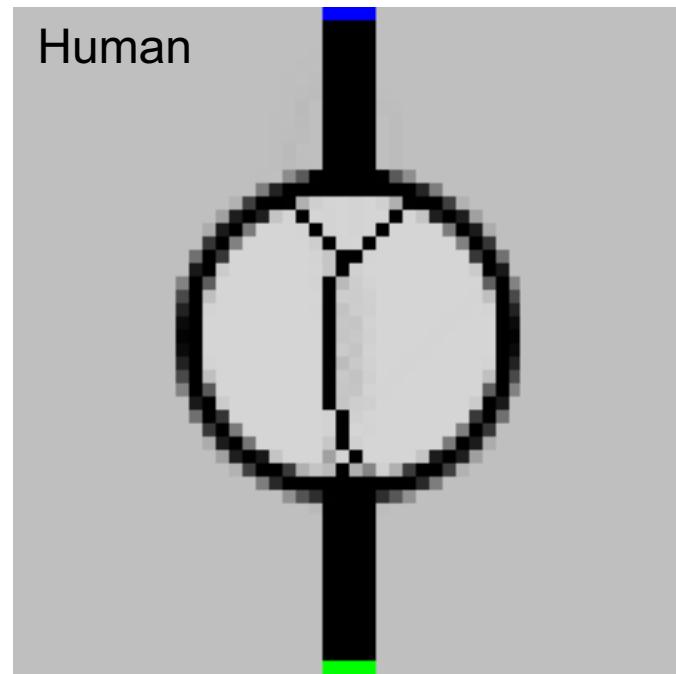
A*



Beam

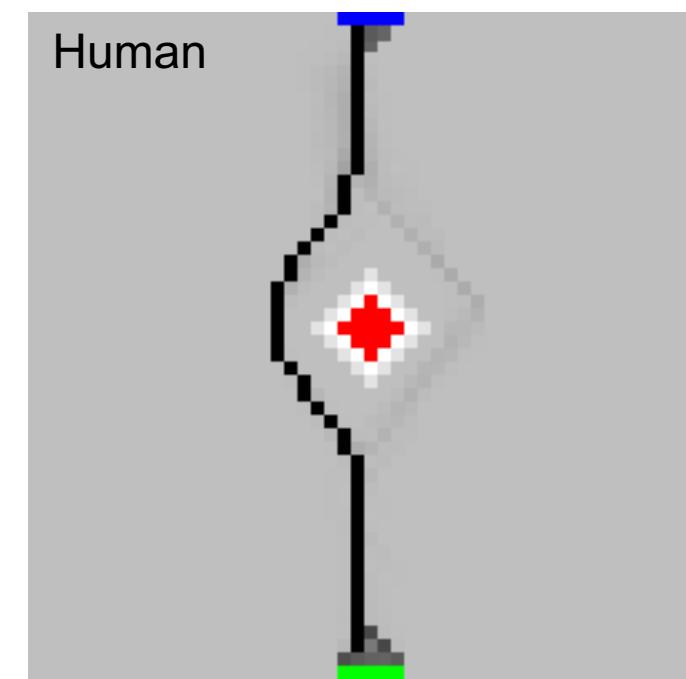
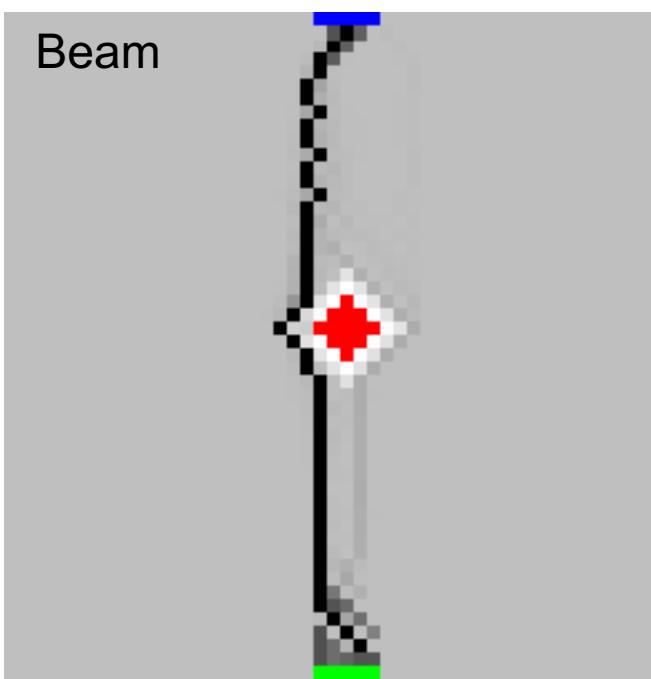
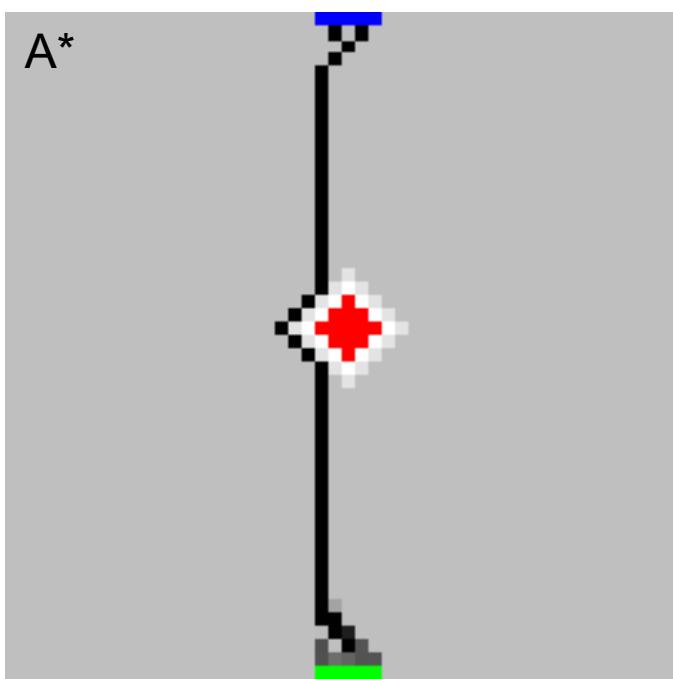
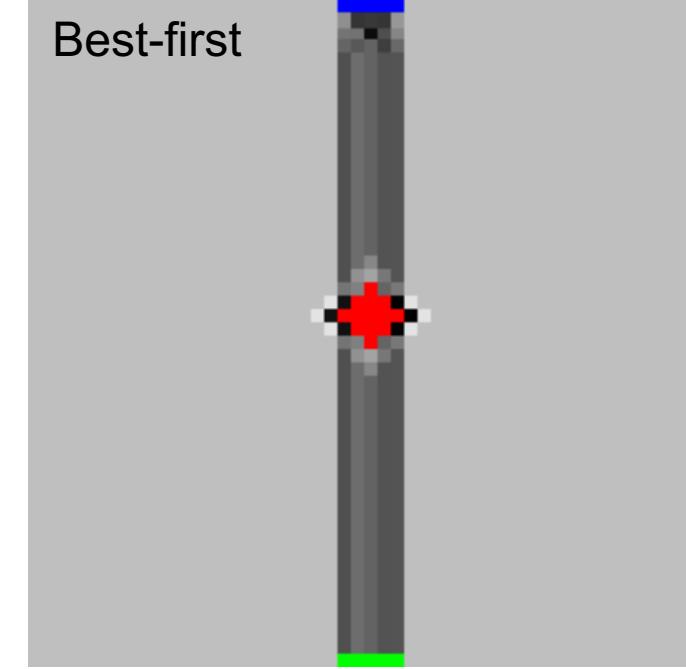
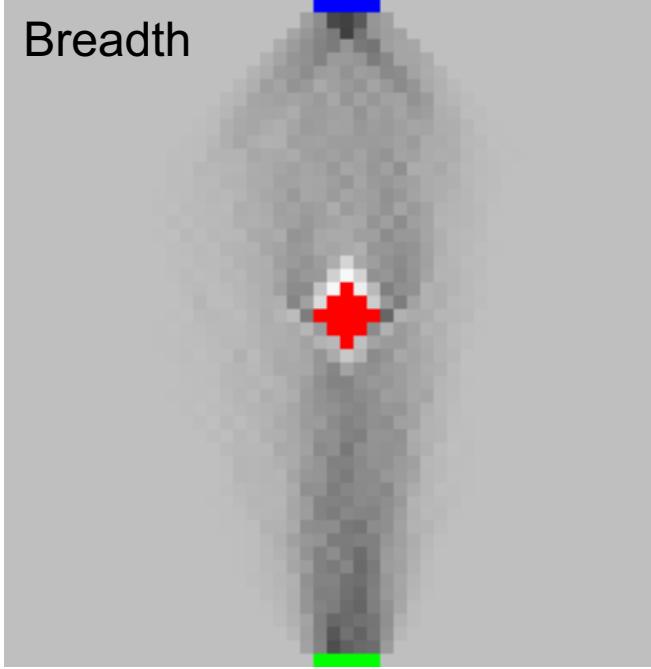
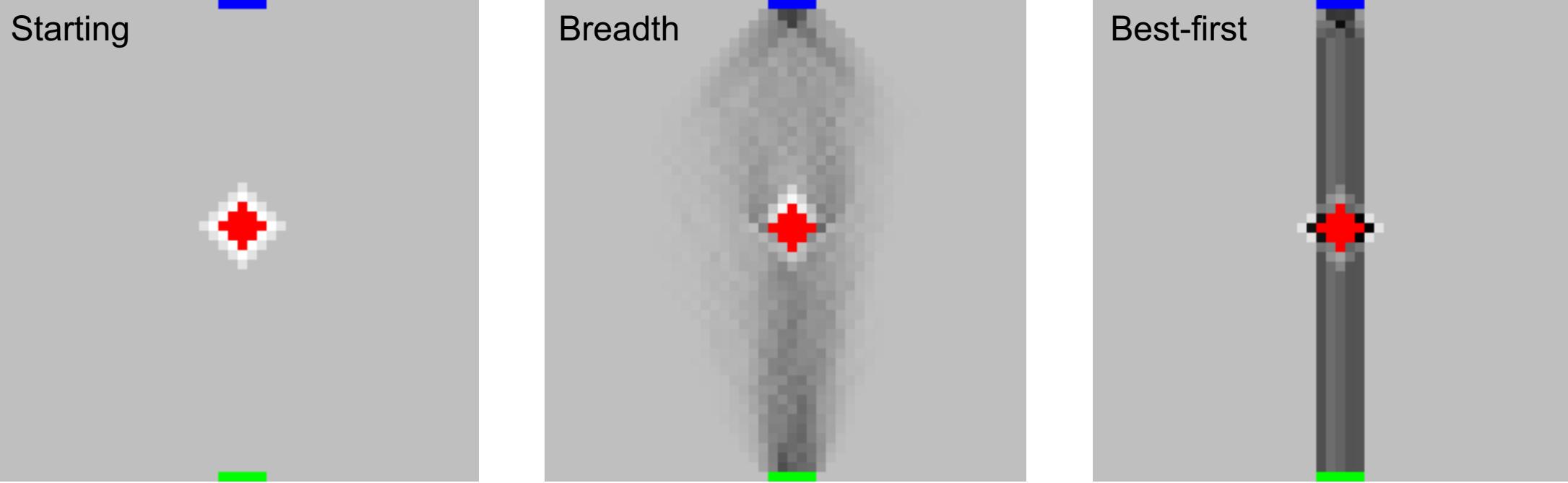


Human

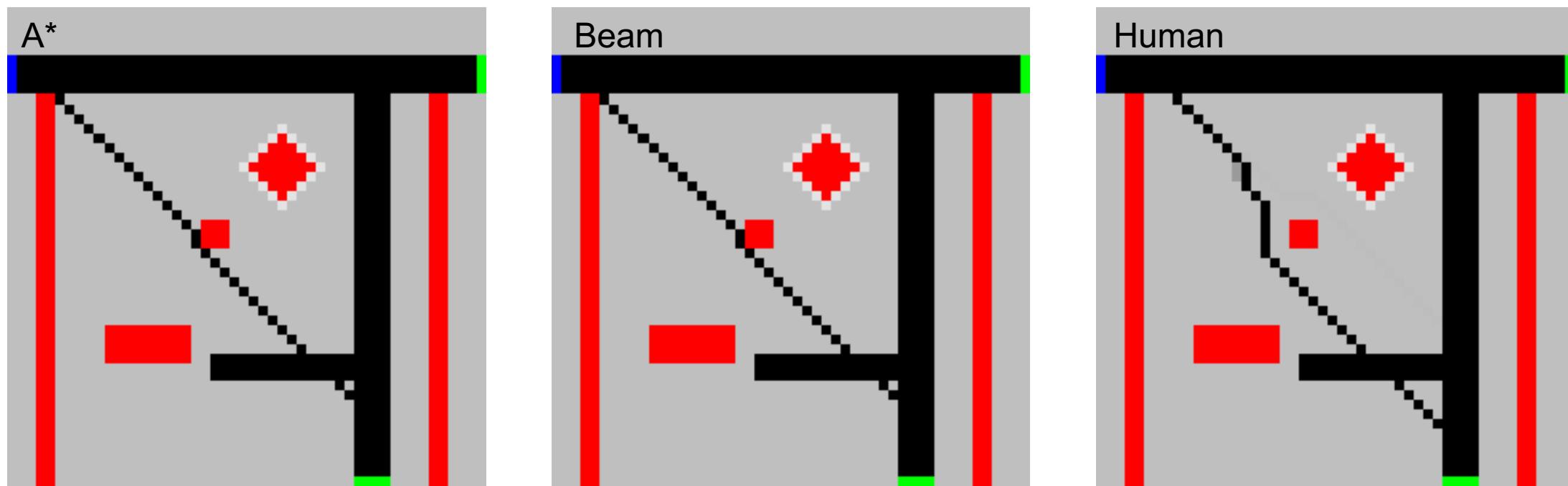
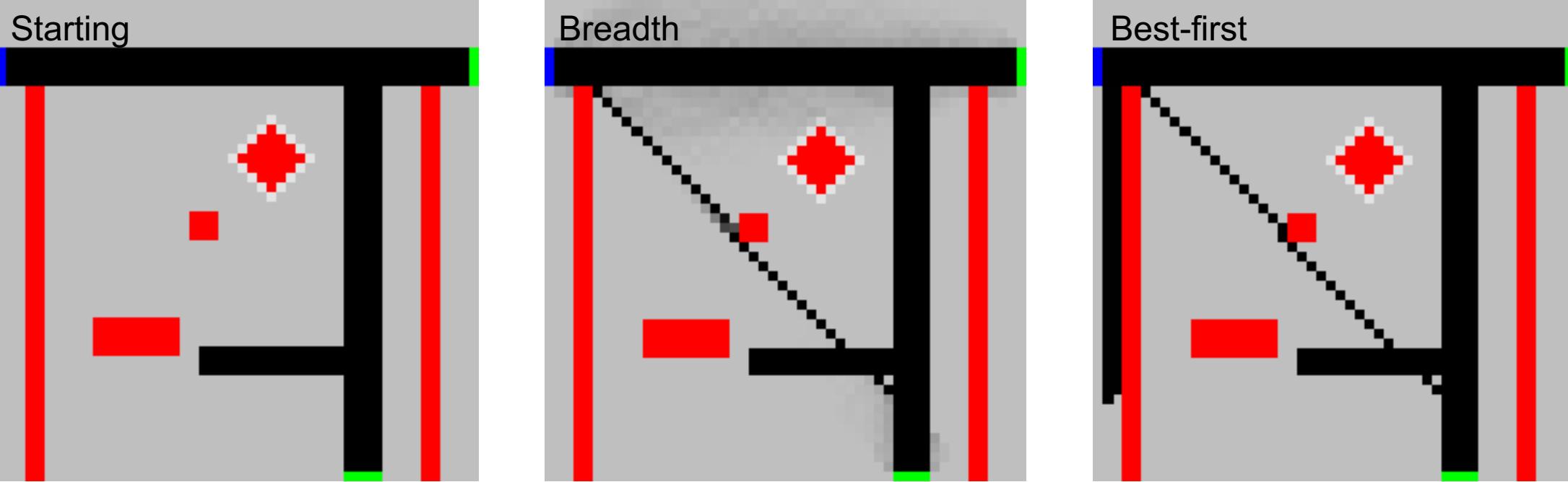


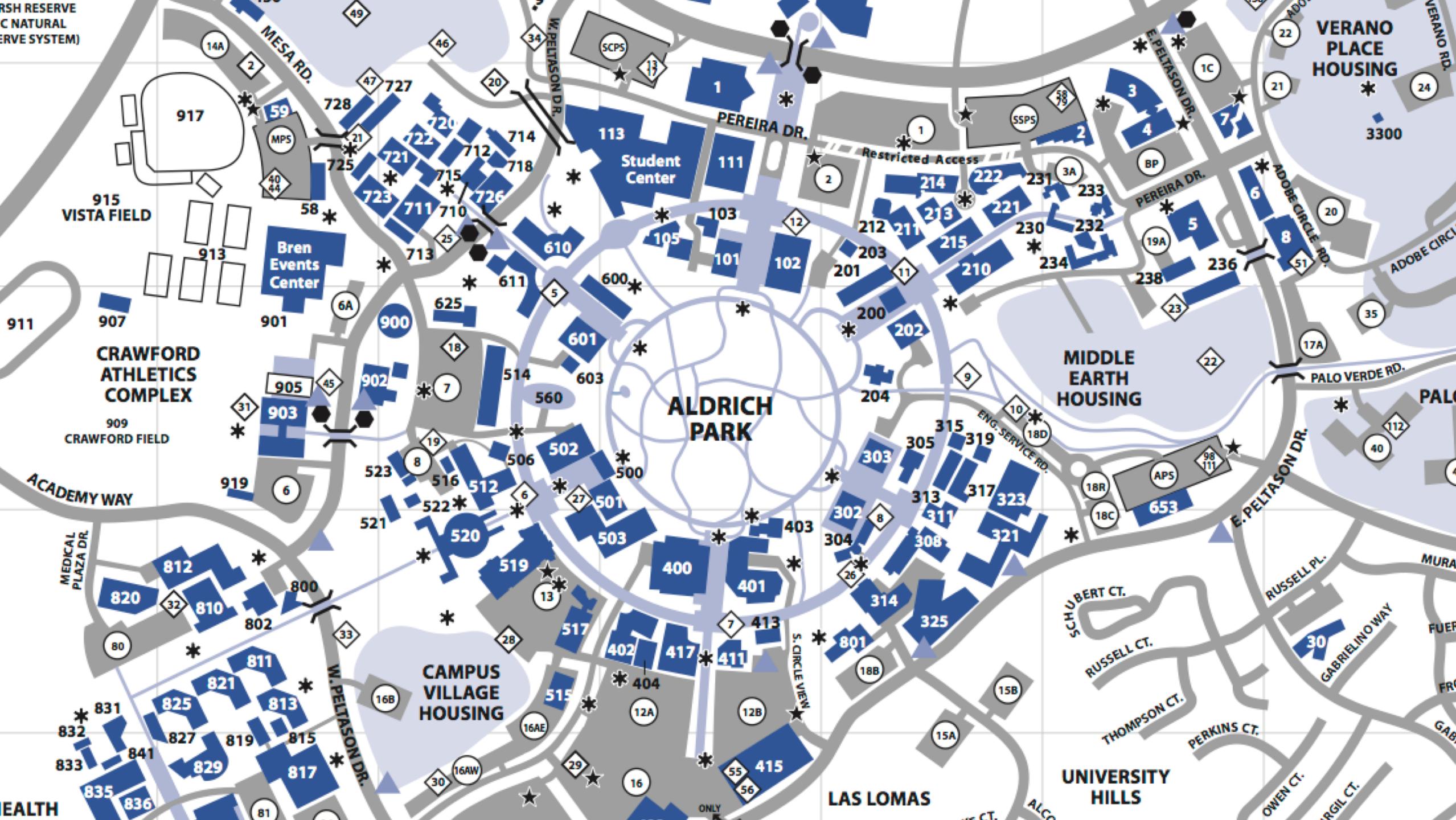


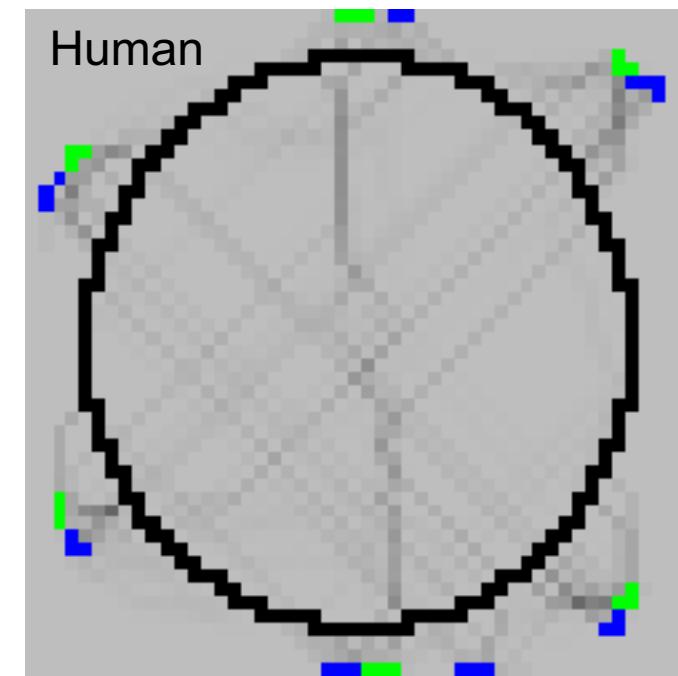
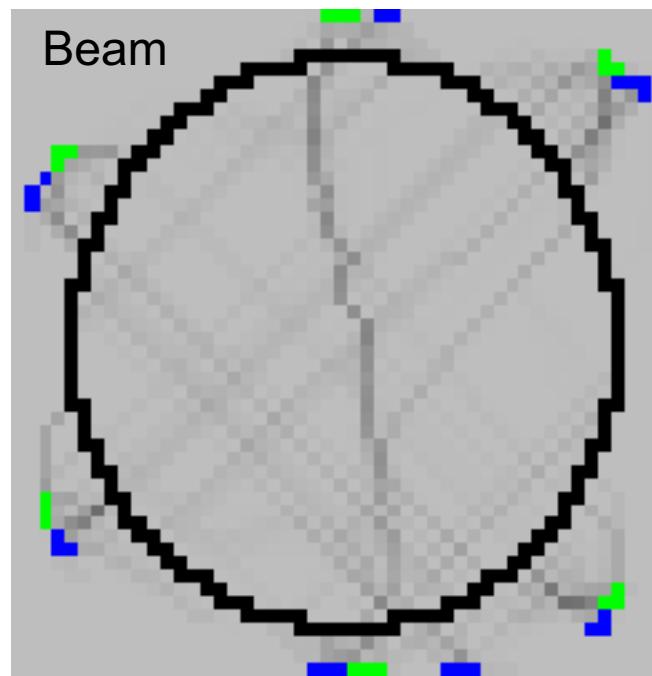
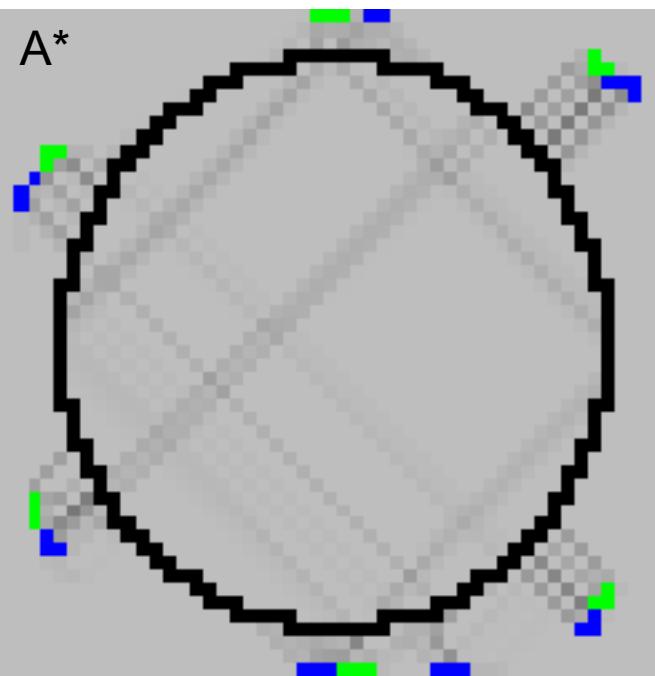
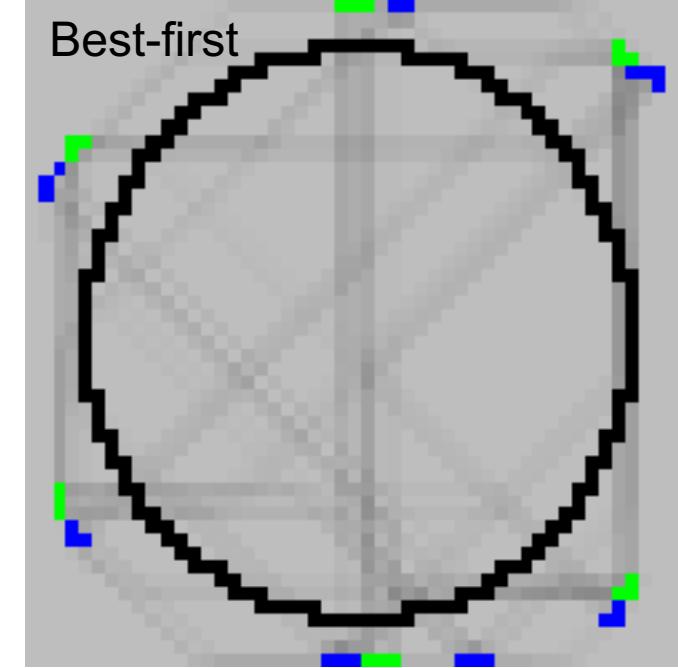
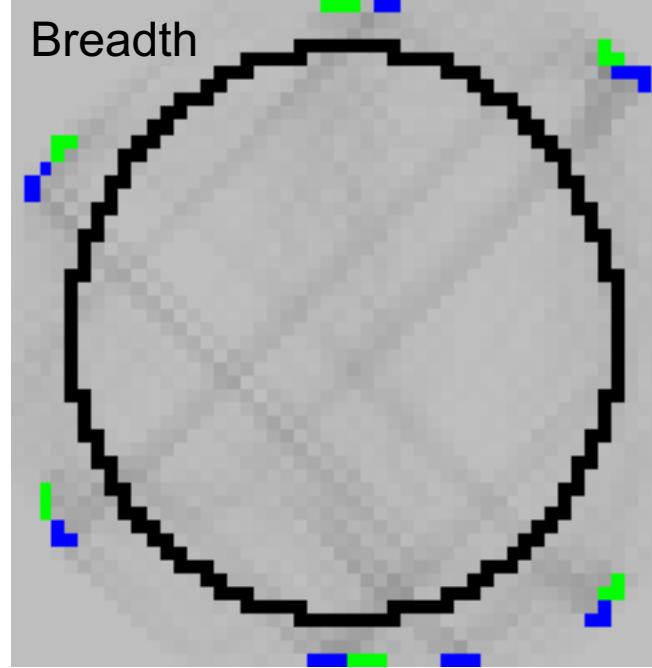
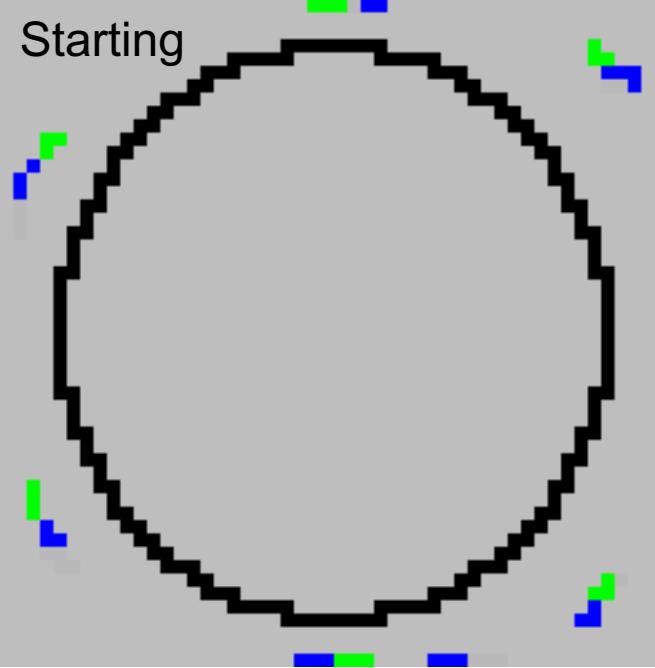
Klaus Humpert
Lauf-Spuren
edition esefeld
& traub
2007











Experience in the classroom

9 of 11 students completed it

One student had a logic error (“human” technique preferred approaching obstacles)

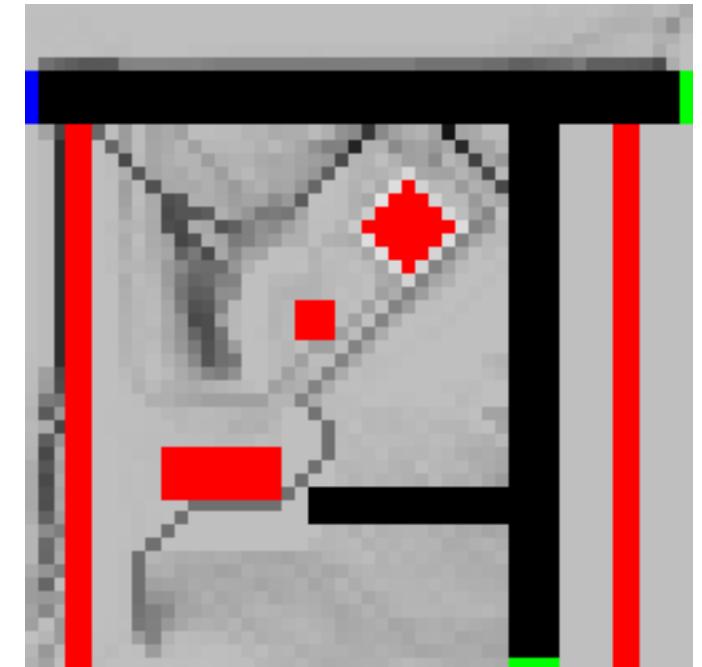
One student did not submit any work

Experience in the classroom

9 of 11 students completed it

One student had a logic error (“human” technique preferred approaching obstacles)

One student did not submit any work



Questions?