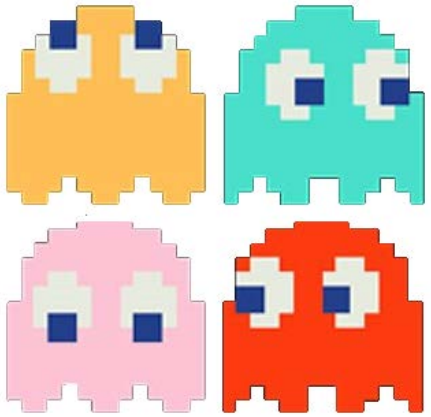
Motion planning in mobile robots

Britta Schulte
3. November 2014

Motion planning in mobile robots

- Introduction
- Basic Problem and Configuration Space
- Planning Algorithms
  - Roadmap
  - Cell Decomposition
  - Potential Field
- Problem Variants
- Example of Use: AI System of Left4Dead

Introduction



Applications of motion planning

The Basic Problem

- The robot is the only moving object
- The robot is a rigid object
- The workspace includes obstacles
- The geometry of the robot and the obstacles is known
- There are no motion constraints

Given a start and goal position
generate a continuous path without collision.

Configuration Space

- 2D represantation
- Robot is represented as a point
- Obstacles are mapped
- Inflation of objects by the robot‘s radius

Planning Algorithms

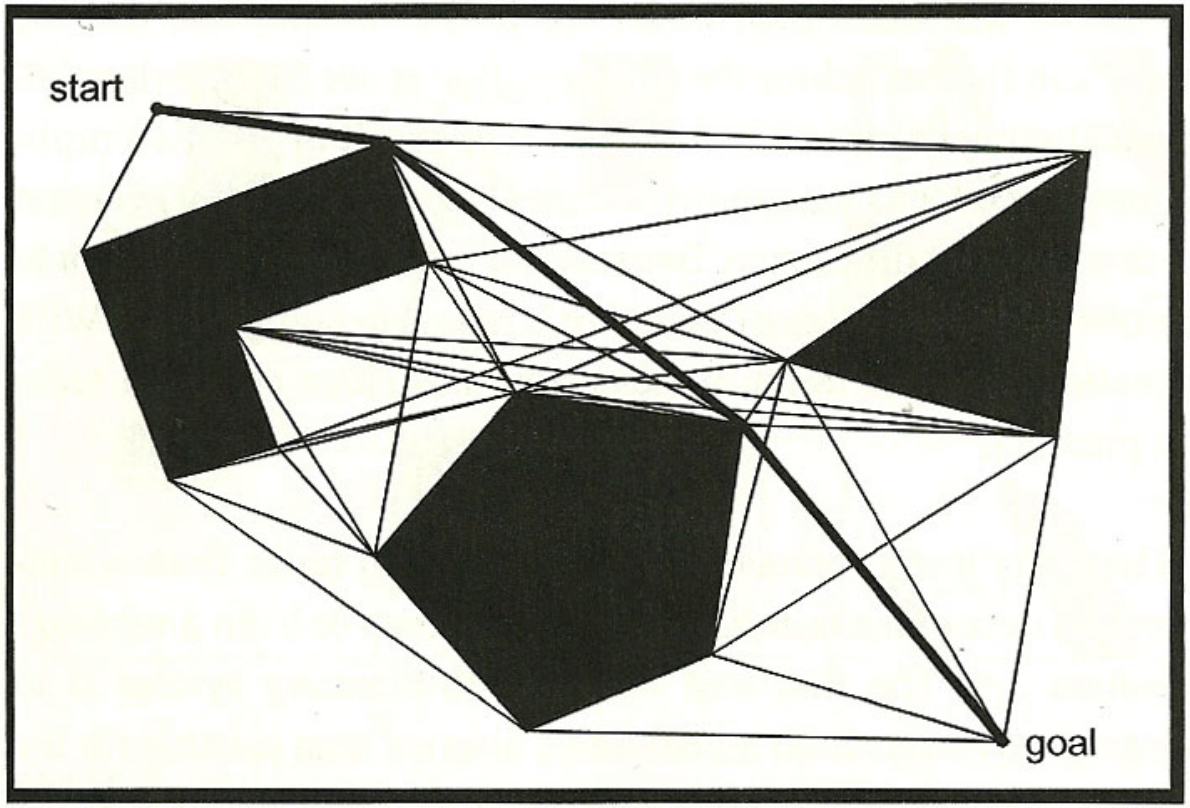
Strategies to transform the continuous environment into a discrete map

Complete algorithm: returns a path if one exists, returns failure otheriwse

Roadmap method

- Representation of space:
  1D line (road) network
- Path:
  Series of roads that conenct start and goal position

Roadmap: Visibility Graph

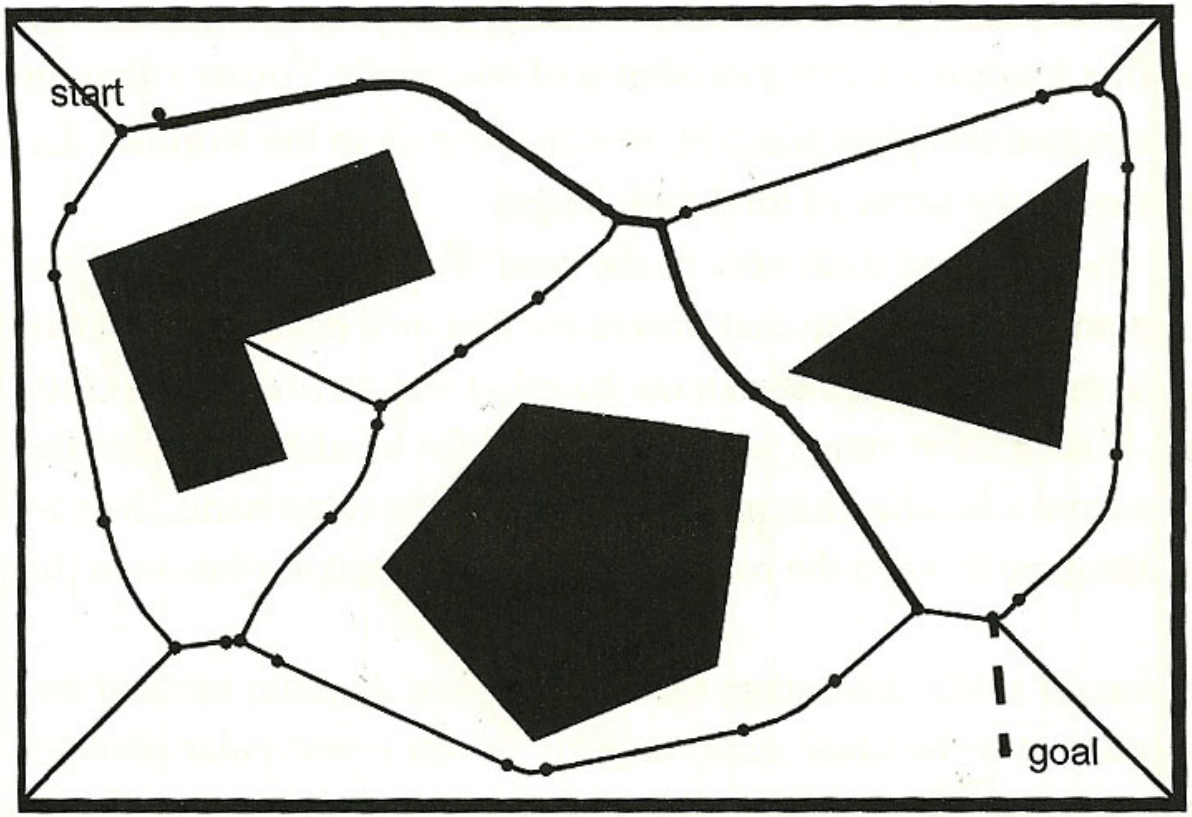


Roadmap: Visibility Graph

+ Simple implementation + Efficient in sparse environments + Creates a path of optimal length	– Size increases with number of obstacles – Slow in populated environments – Robot is as close as possible to obstacles → safety is sacrificed solution: grow obstacles by more than the robot's radius or modify the path away from obstacles

Complete

Roadmap: Vonoroi Diagram



Roadmap: Vonoroi Diagram

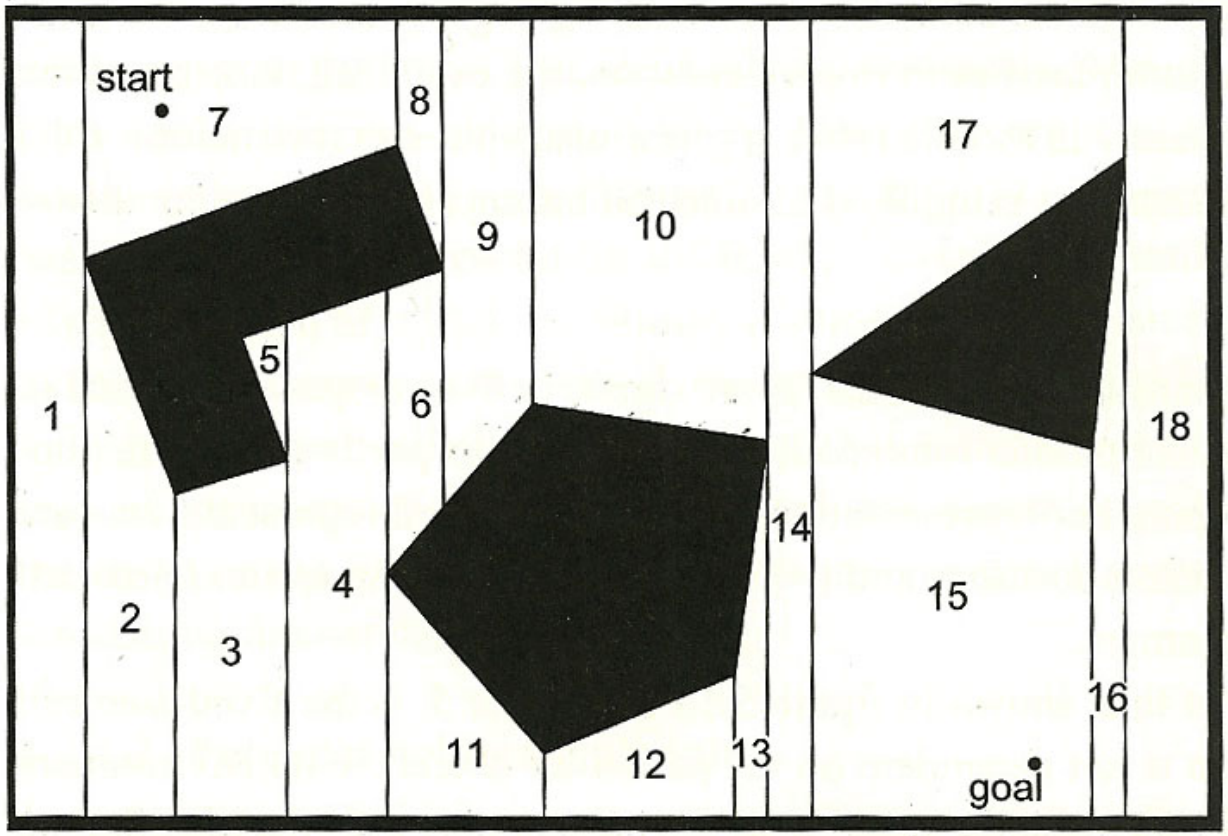
+ Paths can be followed easily with range sensors + Automatic mapping of the environment via moving along unknown edges	– Much less than optimal length – Poor localization of robots with short ranged sesors solution: use visibility graph to keep robot close enough to obstacles

Complete

Cell Decomposition

- Discriminate between free and occupied areas
- Divide space into cells
- Adjacent free cells constitute the connectivity graph
- Find a series of cells that connect start and goal
- Connect cells through their mid points

Exact Cell Decomposition

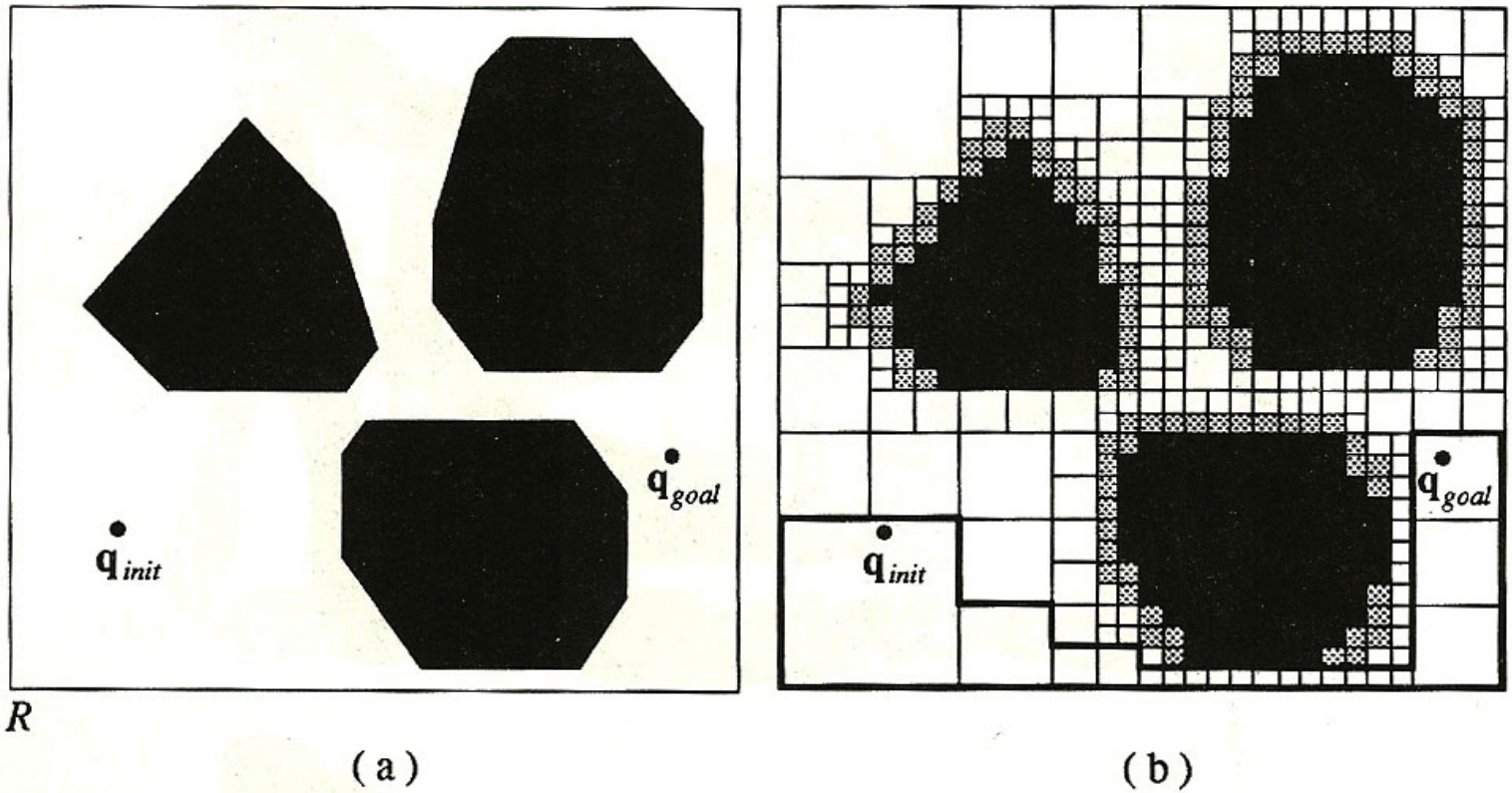


Exact Cell Decomposition

+ Sparse/large environment → few cells → high efficiency	– Efficiency decreases with density of obstacles – Complex implementation

Complete

Approximate Cell Decomposition



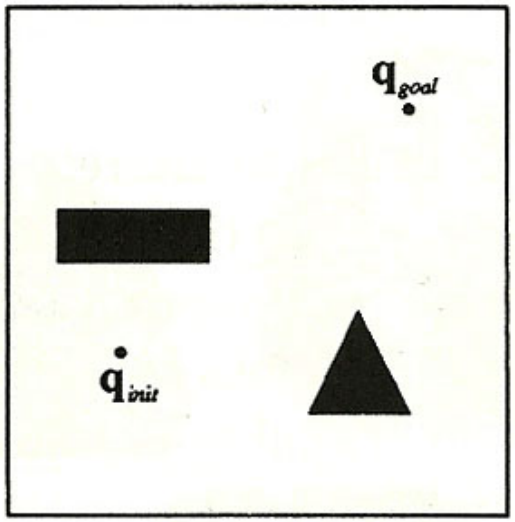
Approximate Cell Decomposition

+ Adapts to complexity of environment (sparse → fewer cells → less memory needed)	– Representing a complete environment is memory intensive

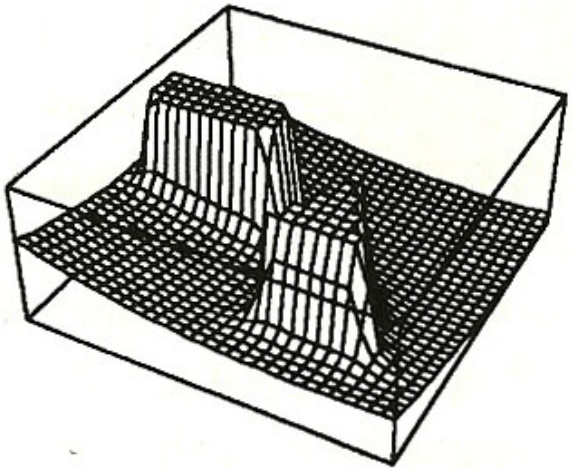
Not Complete

(possibly „Resolution Complete“)

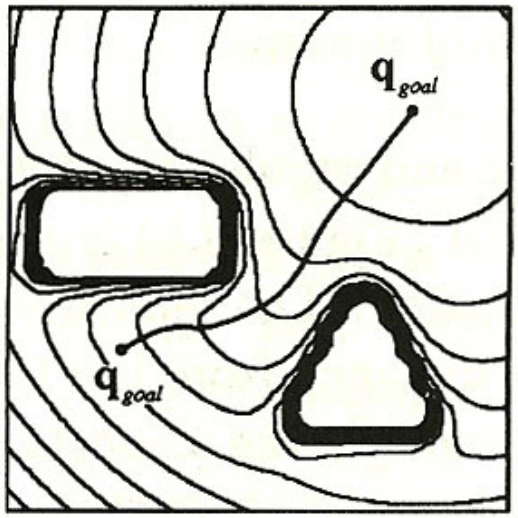
Potential Field



(a)



(b)



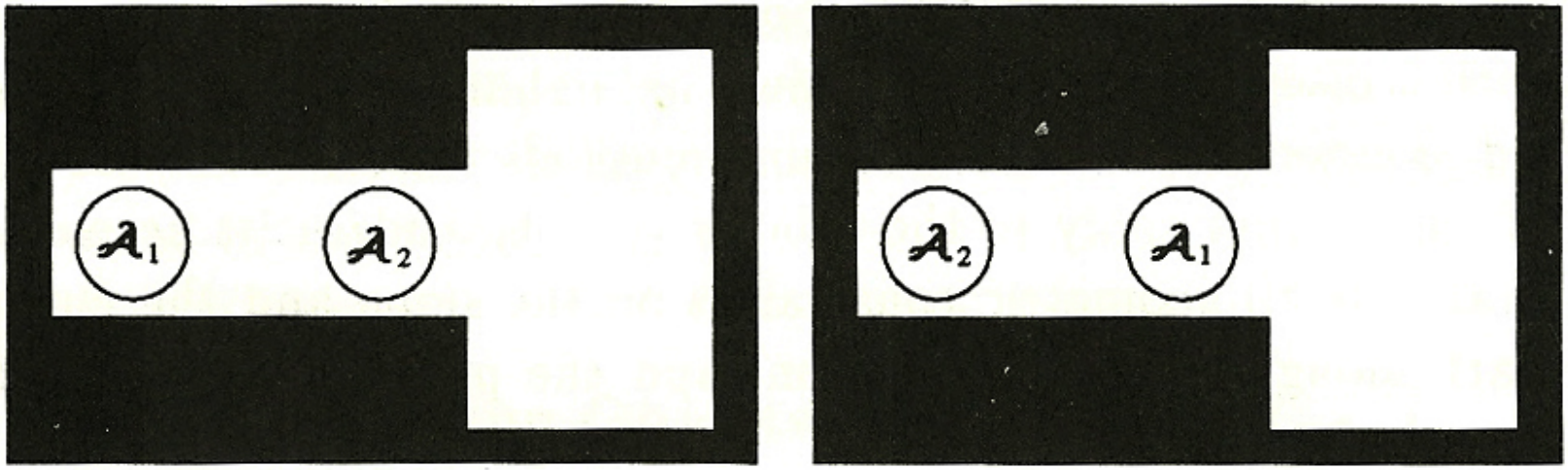
(c)

Potential Field

+ Very efficient + Easy to implement	– Potential to get trapped in local minima solution: design a function that avoids local minima or design a mechanism to escape local minima

Usually incomplete

Problem Variants



Initial configuration

Goal configuration

Multiple robots in one workspace

Moving Obstacles

- Time as a dimension in configuration space

  → „configuration-time-space“

- Specifies robot‘s position for each point in time

Difficulty depends on kinematic constraints and knowledge about the object‘s movement

Multiple Robots

- Difference from moving obstacles:
  not under control vs. Planned
- Centralized planning:
  multiple robots as one multibodied robot
  configuration space = product of each robot‘s config. space
- Decoupled planning:
  plan wach robot‘s motion independently
  second planning phase for interaction
  less computation vs. less completeness

Artculated Robots

- E.g. robot arm → objects/links connected by joints
- Config. space = product of object‘s config. Spaces
- Angle of joint = added parameter
- Config space grows with number of objects

Kinematic Constraints

Holonomic constraints:

- Dependency of parameters of config. Space

Nonholonomic constraints:

- Restrict possible motions at any configuration

Uncertainty

- Little/no knowledge about workspace
- Sensors needed to optain information
- Exploration required
- Less knowledge → planning less important

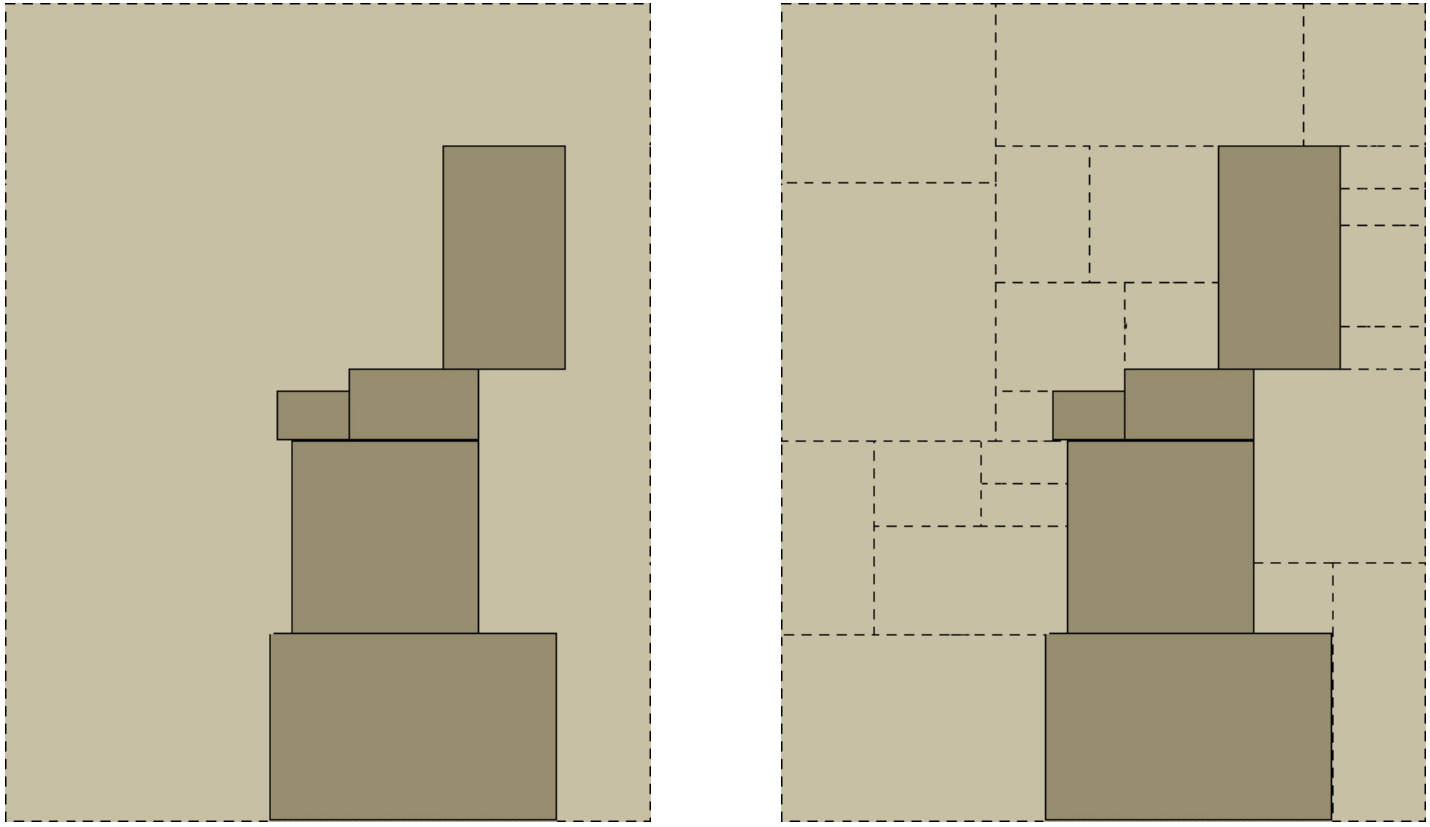
Movable objects

- Using motion commands to solve intermediate problems
- Transit happens in config. space
  Transfer happens in the geometry

Exmple of Use: AI System of Left4Dead

Exmple of Use: AI System of Left4Dead

Navigation Meshes



Exmple of Use: AI System of Left4Dead

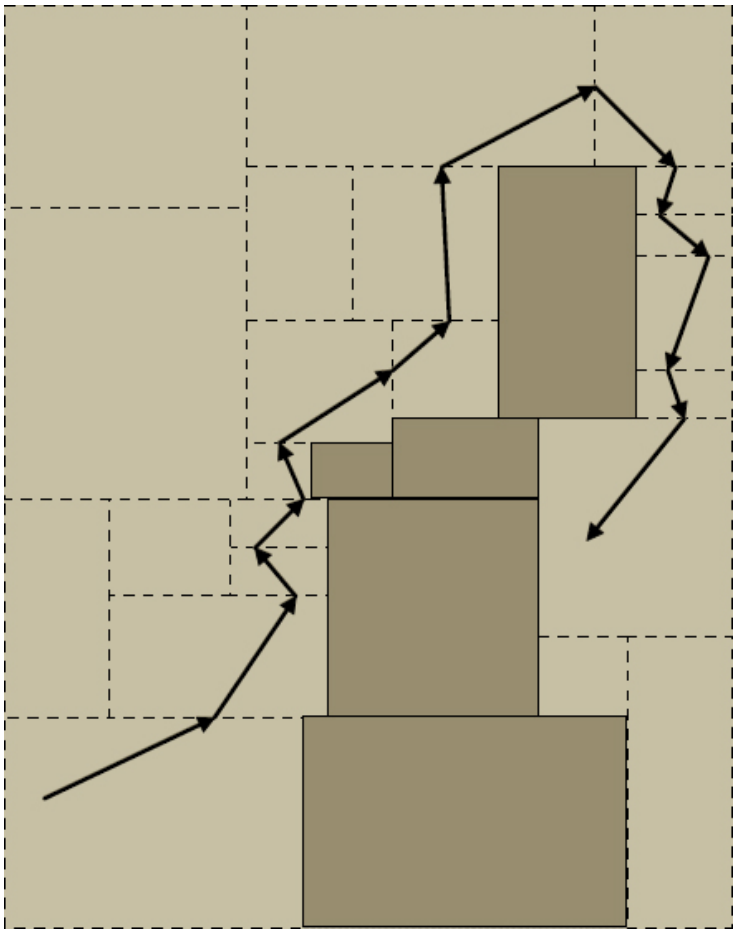
Planning Algorith A*

- Function: $f(n) = g(n) + h(n)$
- Process:

1. mark initial node
   expand adjacent nodes
2. calculate function value for each node
   sort nodes
   identify node of minimum cost
3. repeat for subsequent nodes until target node is reached

Exmple of Use: AI System of Left4Dead

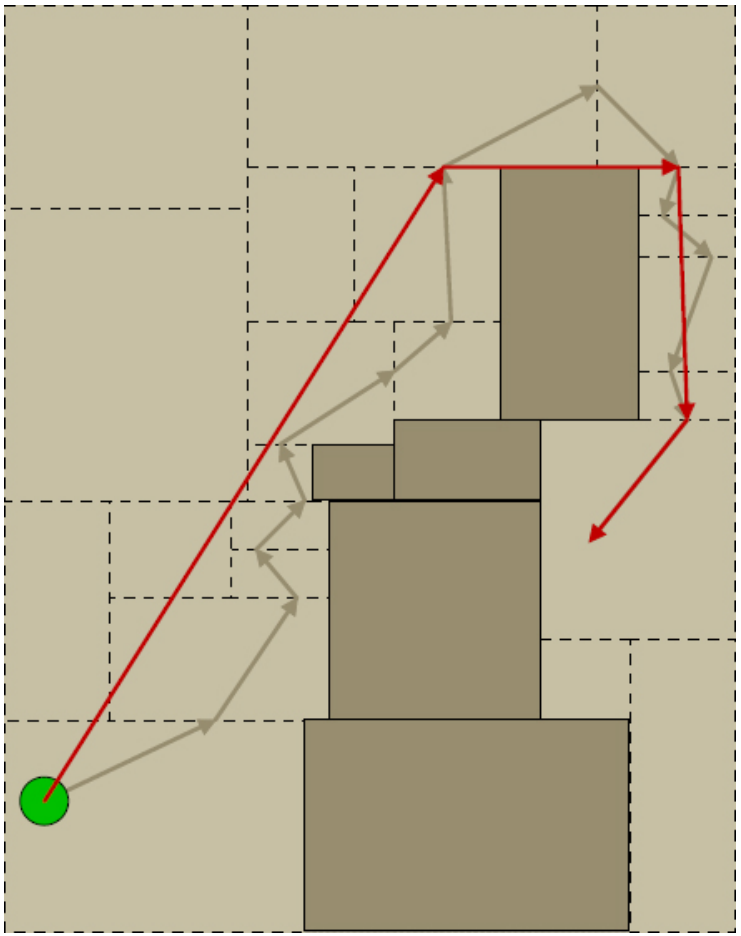
- A* creates jagged path
- fluid motion is desired



Exmple of Use: AI System of Left4Dead

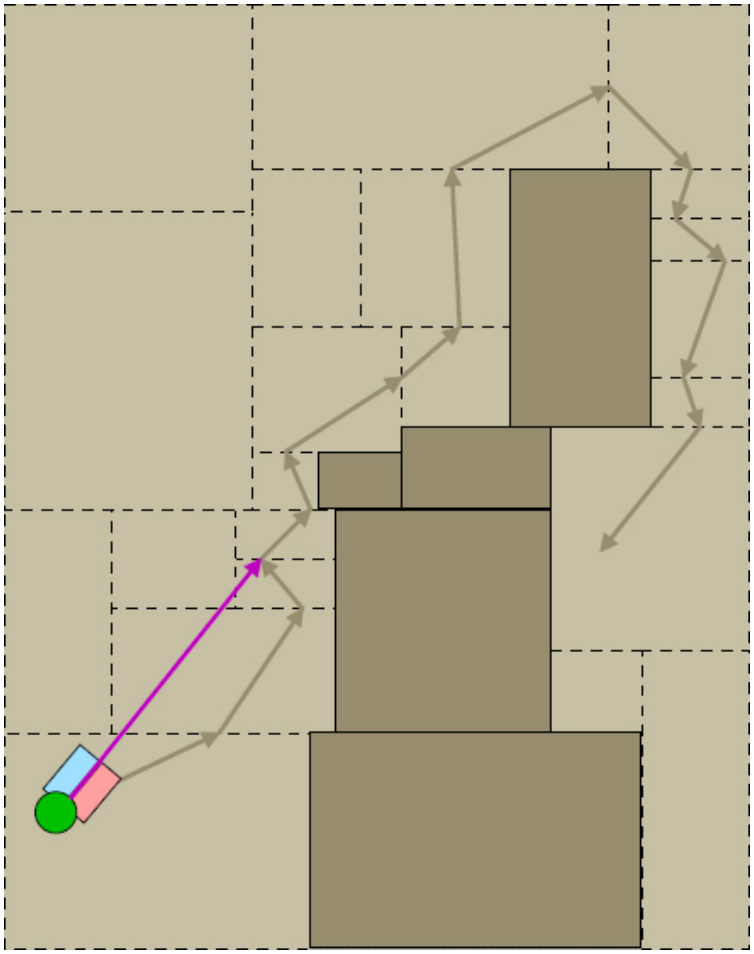
Possible solution:

- Path optimization
- Creates a direct path
- Needs to be recomputed often
- Still jagged



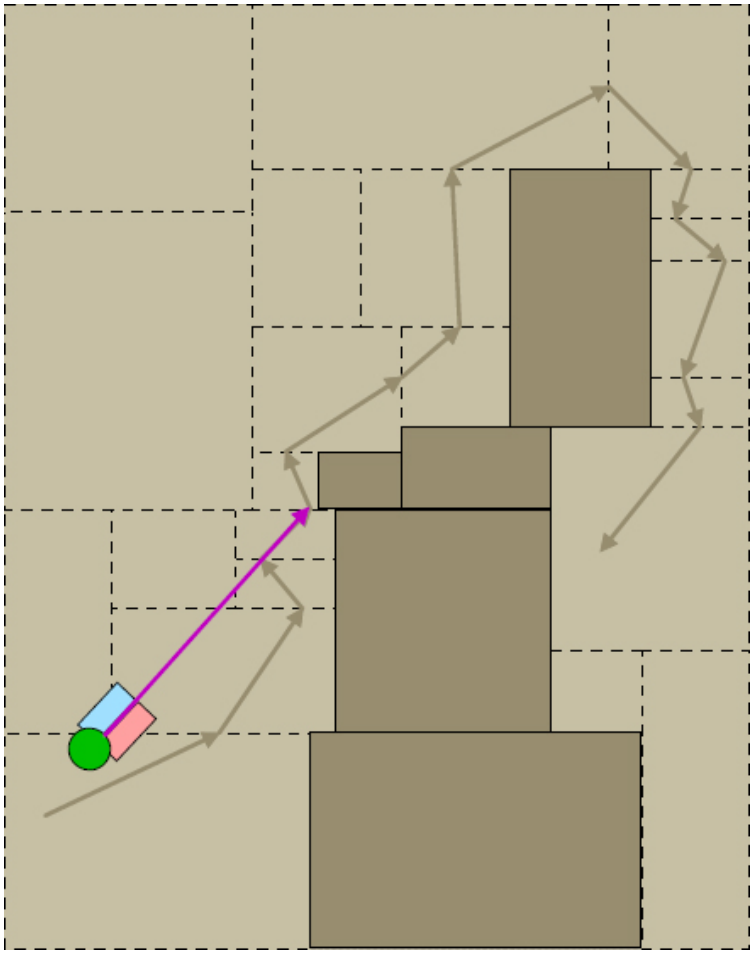
Exmple of Use: AI System of Left4Dead

Actual solution:



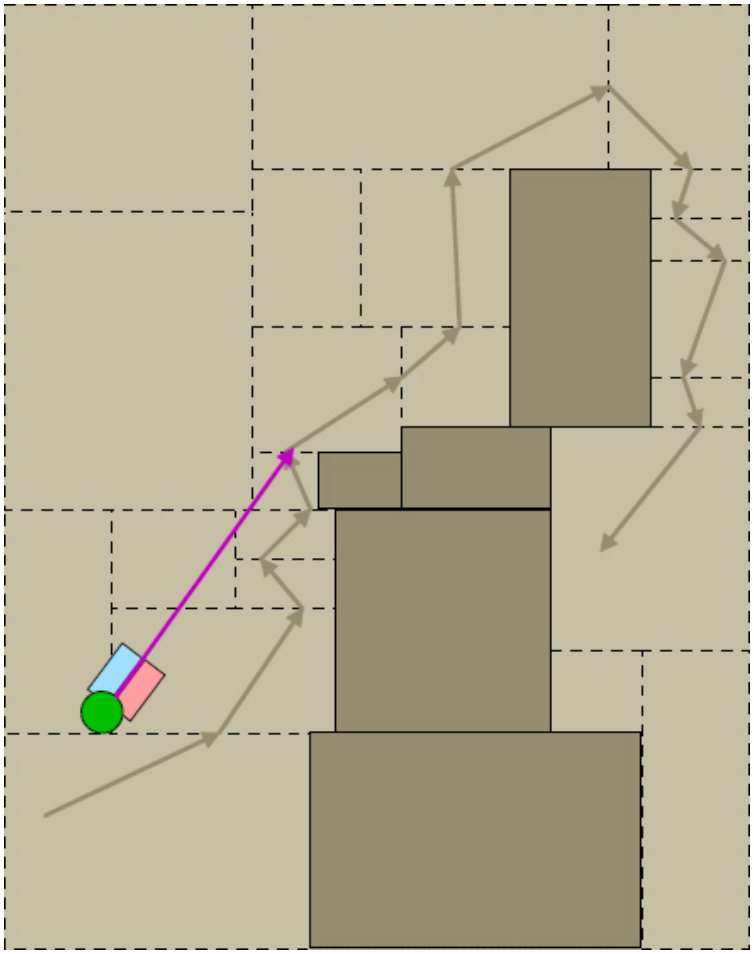
Exmple of Use: AI System of Left4Dead

Actual solution:



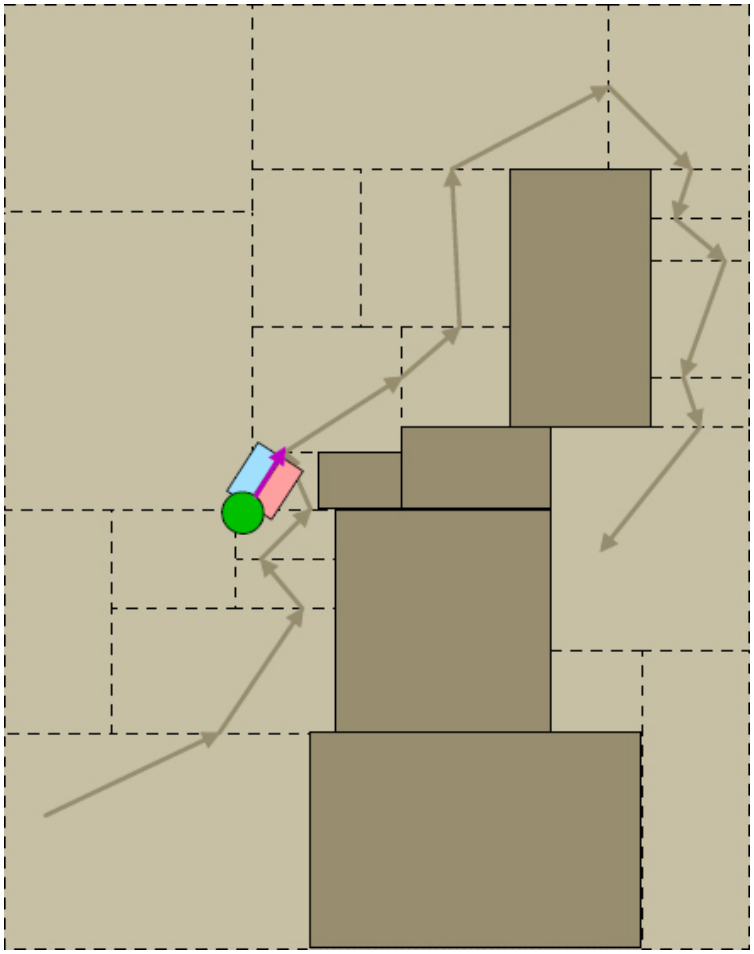
Exmple of Use: AI System of Left4Dead

Actual solution:



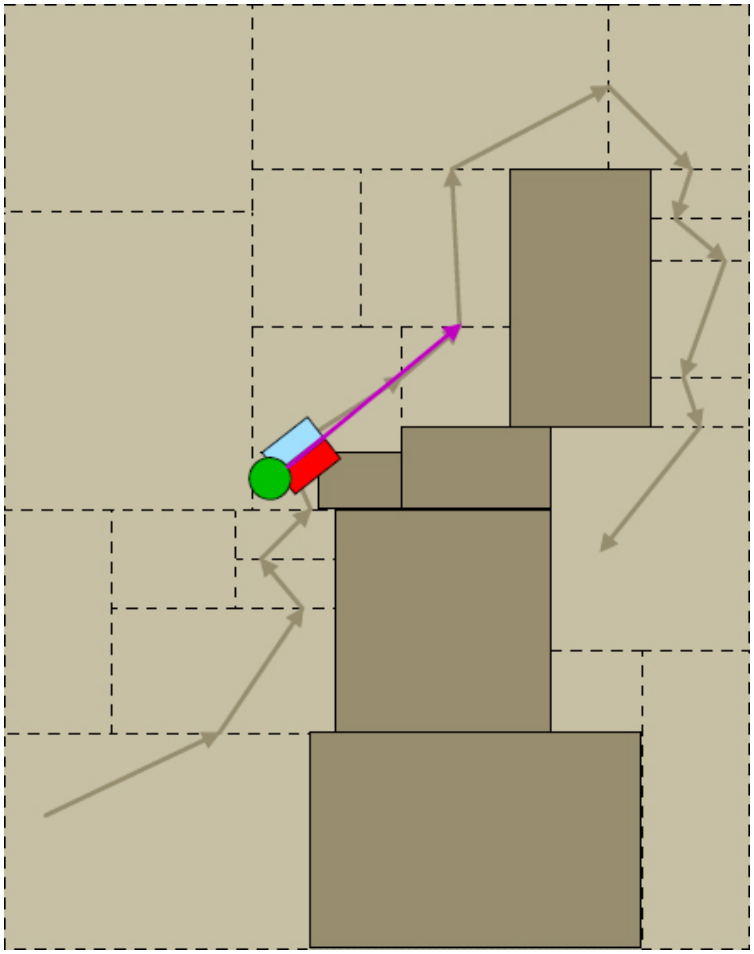
Exmple of Use: AI System of Left4Dead

Actual solution:



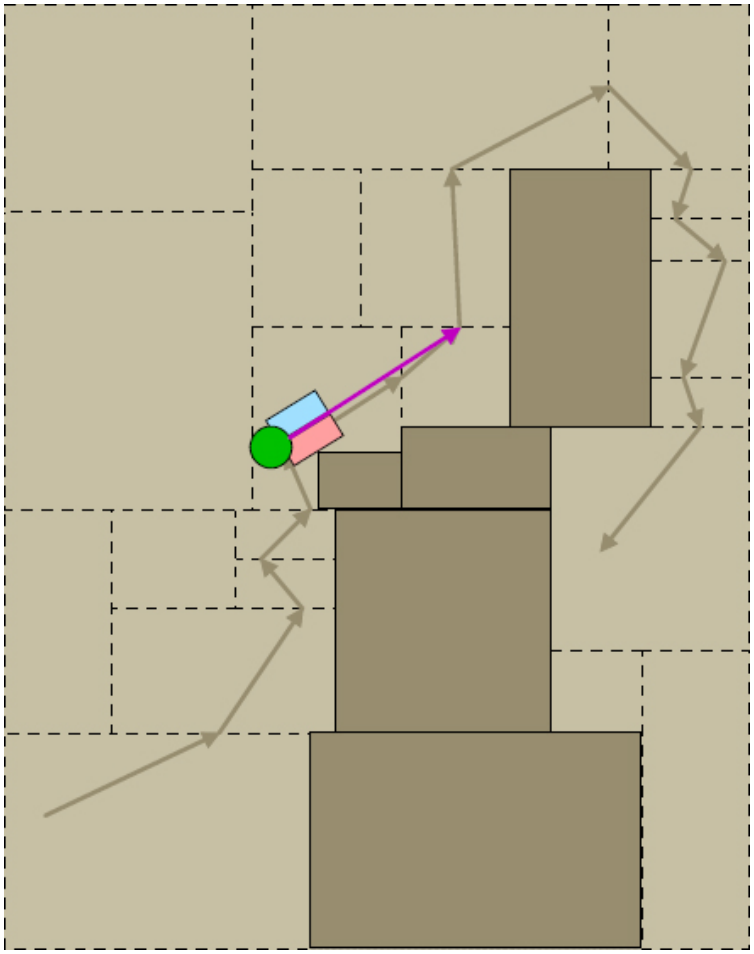
Exmple of Use: AI System of Left4Dead

Actual solution:



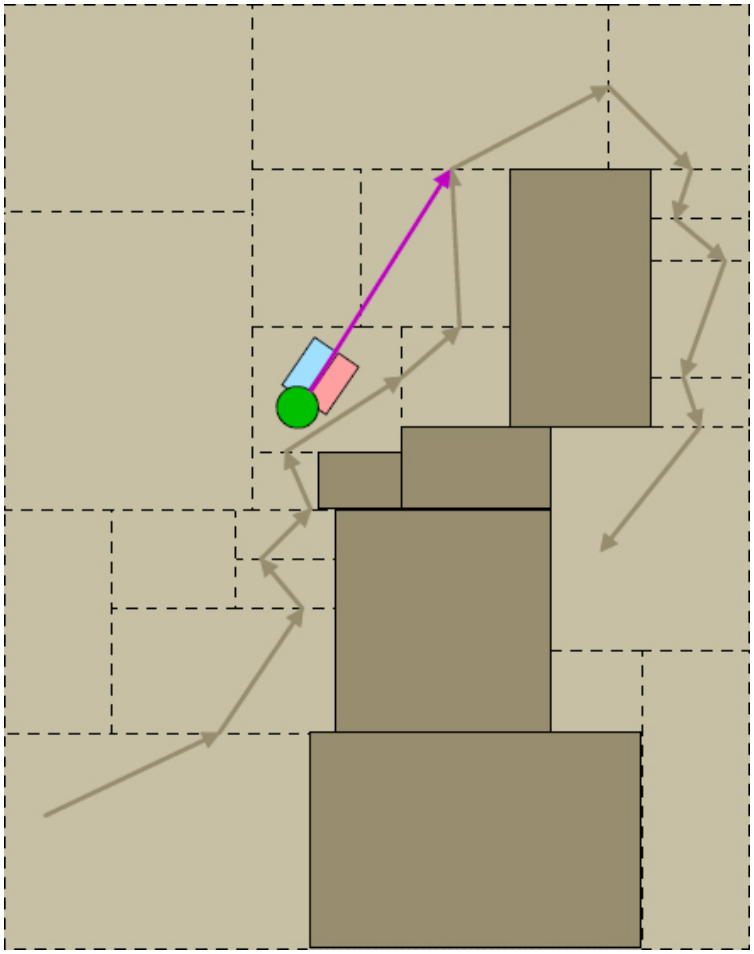
Exmple of Use: AI System of Left4Dead

Actual solution:



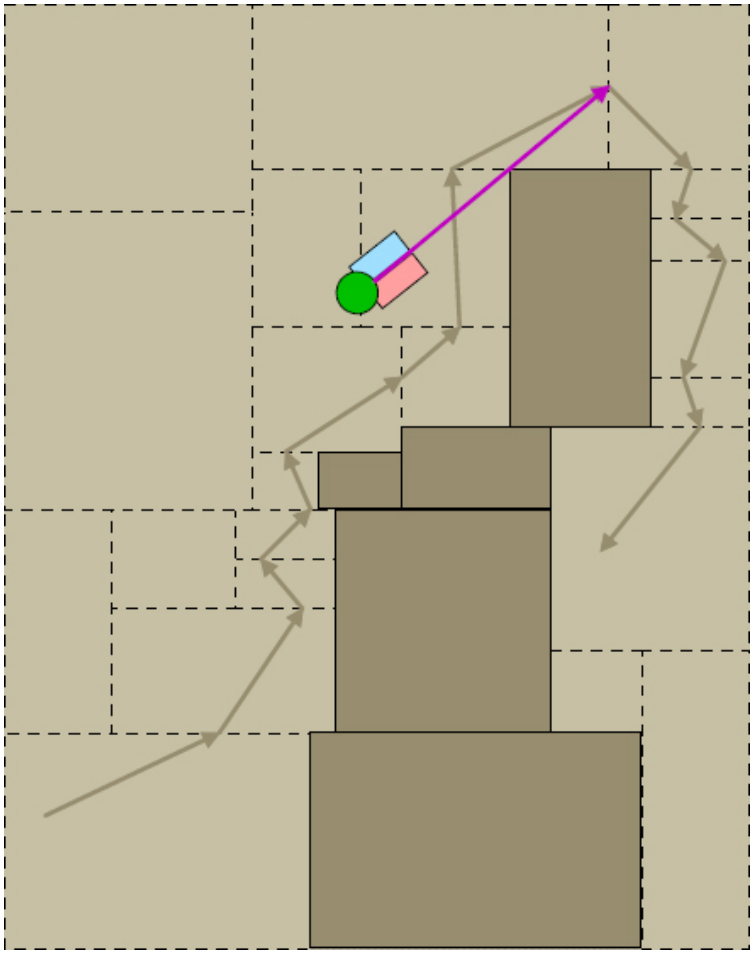
Exmple of Use: AI System of Left4Dead

Actual solution:



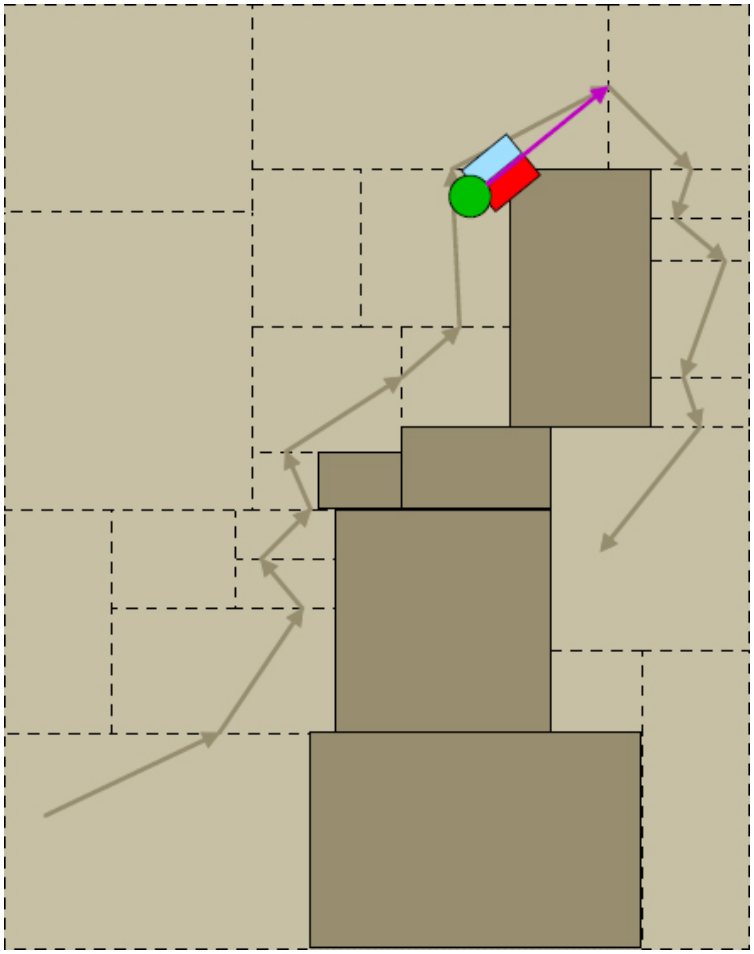
Exmple of Use: AI System of Left4Dead

Actual solution:



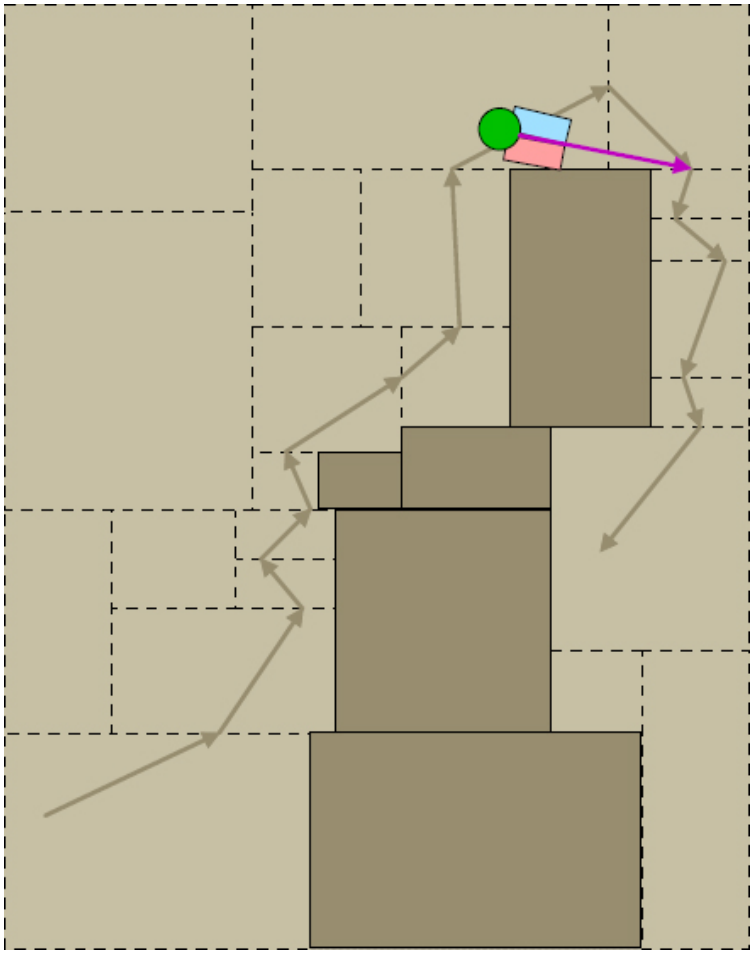
Exmple of Use: AI System of Left4Dead

Actual solution:



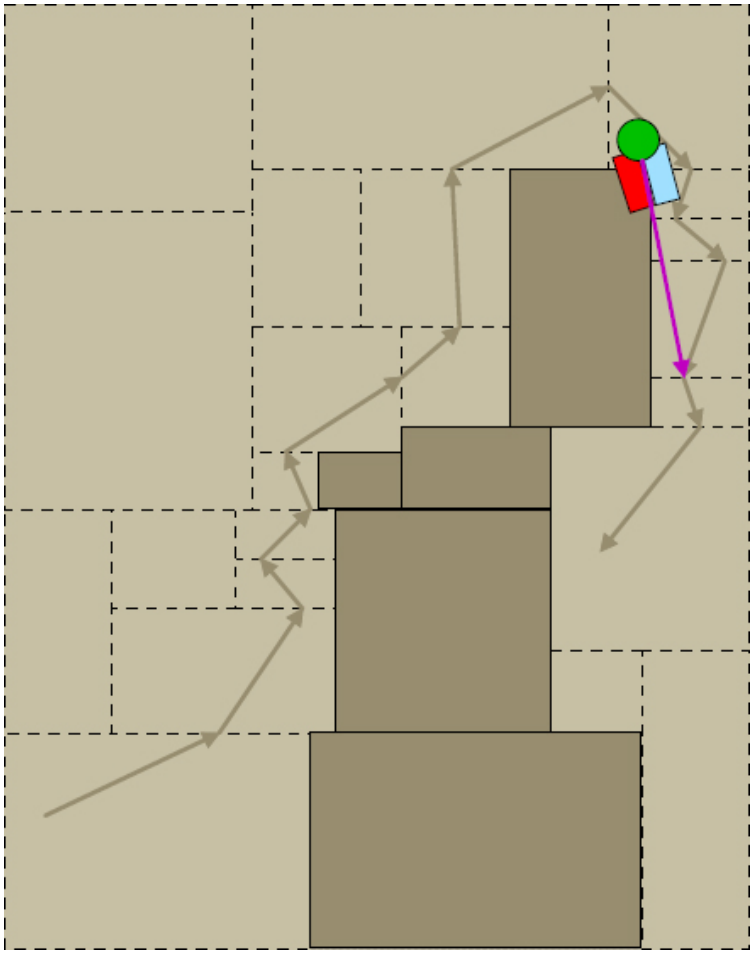
Exmple of Use: AI System of Left4Dead

Actual solution:



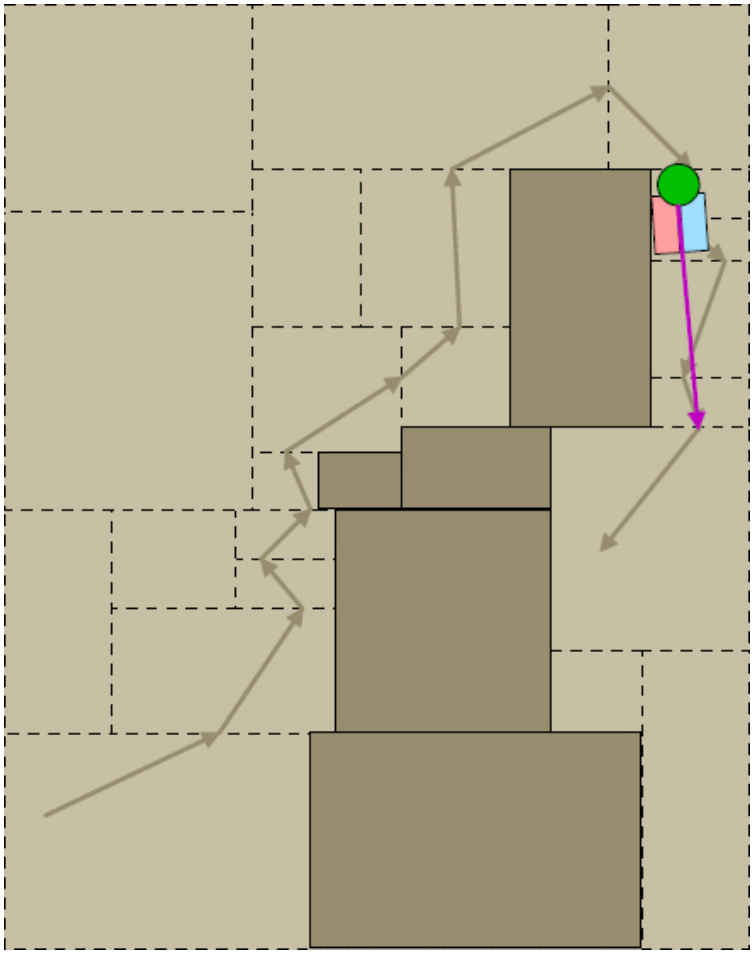
Exmple of Use: AI System of Left4Dead

Actual solution:



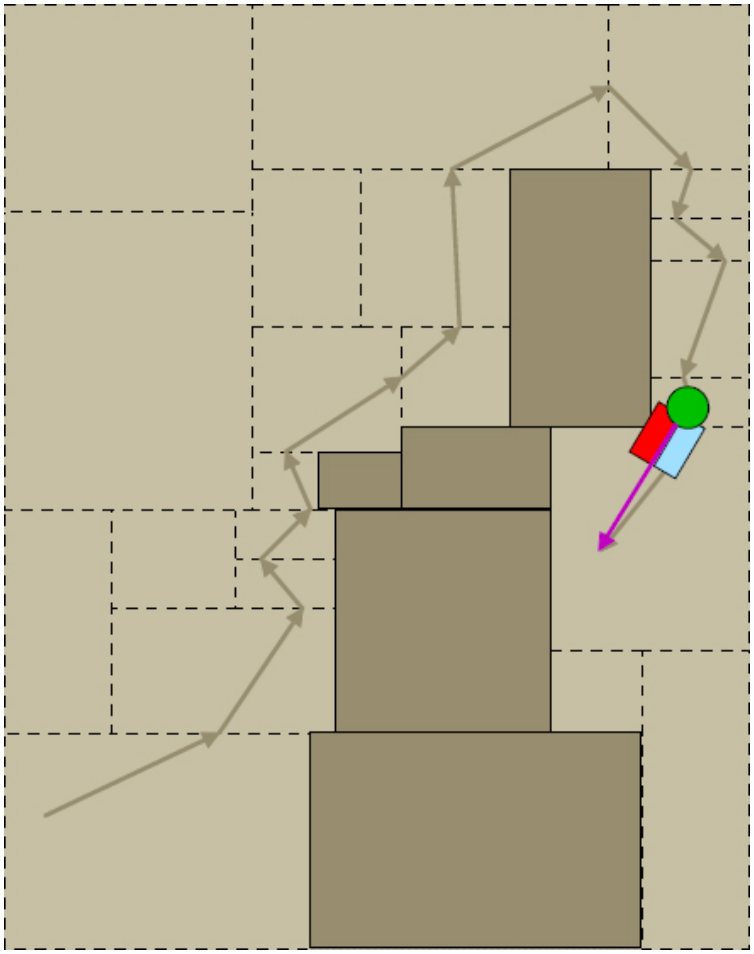
Exmple of Use: AI System of Left4Dead

Actual solution:



Exmple of Use: AI System of Left4Dead

Actual solution:



Exmple of Use: AI System of Left4Dead

Actual solution:

- Movement towards a point of the path in the direction of view
- Cheap re-pathing
- Avoiding obstacles can lead into different area
  → re-pathing

Exmple of Use: AI System of Left4Dead

- Fluid Motion
- Close to optimized path