Comp 790-058 :
Multi-Agent Simulation for Crowds & Autonomous Driving

Sahil Narang

Sept 19, 2017

University of North Carolina at Chapel Hill

Structure

- Crowd Simulation
  - Multi-agent simulation basics
  - Menge

- Global Planners
  - Environment representation
  - Probabilistic Roadmaps (PRM)
  - Other global planners

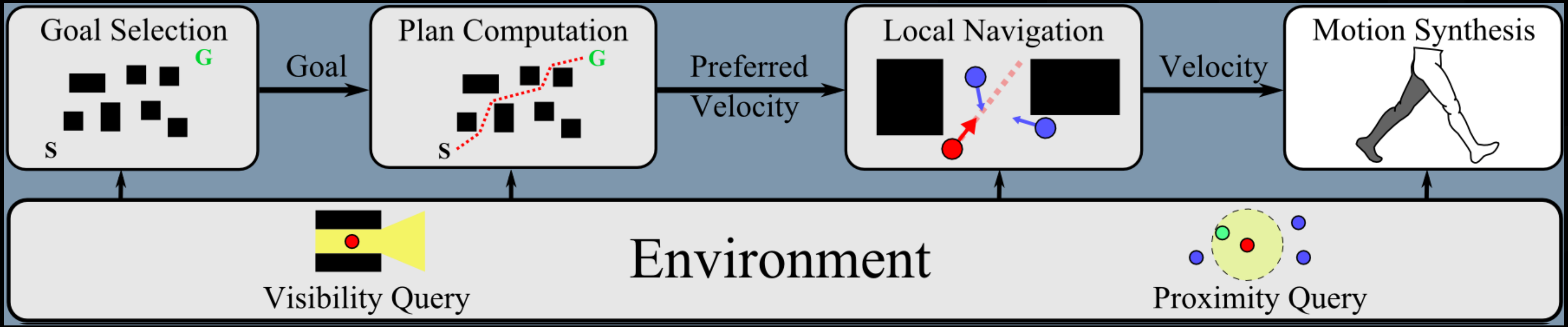
- Assignment 1

WHAT IS MENGE?

- Menge is a modular, pluggable framework for crowd simulation developed at UNC.
- Menge is Open-Source and publicly available.

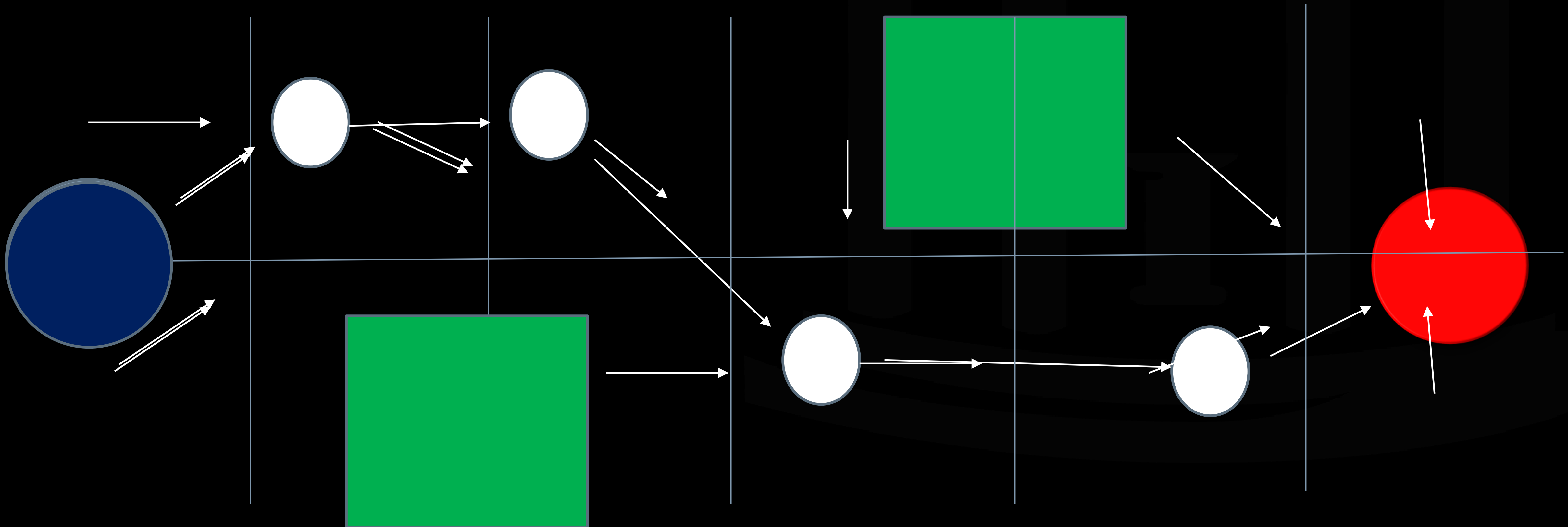
MULTI-AGENT SIMULATION PIPELINE

- Abstract pipeline for multi-agent simulation
- Goal selection: What does the agent want to do
- Plan computation: How does the navigate the environment
  - Preferred velocity: The velocity the agent takes to optimally proceed along its path
- Plan Adaptation: How the Agent responds to local conditions
- Motion synthesis: Animation / Output stage



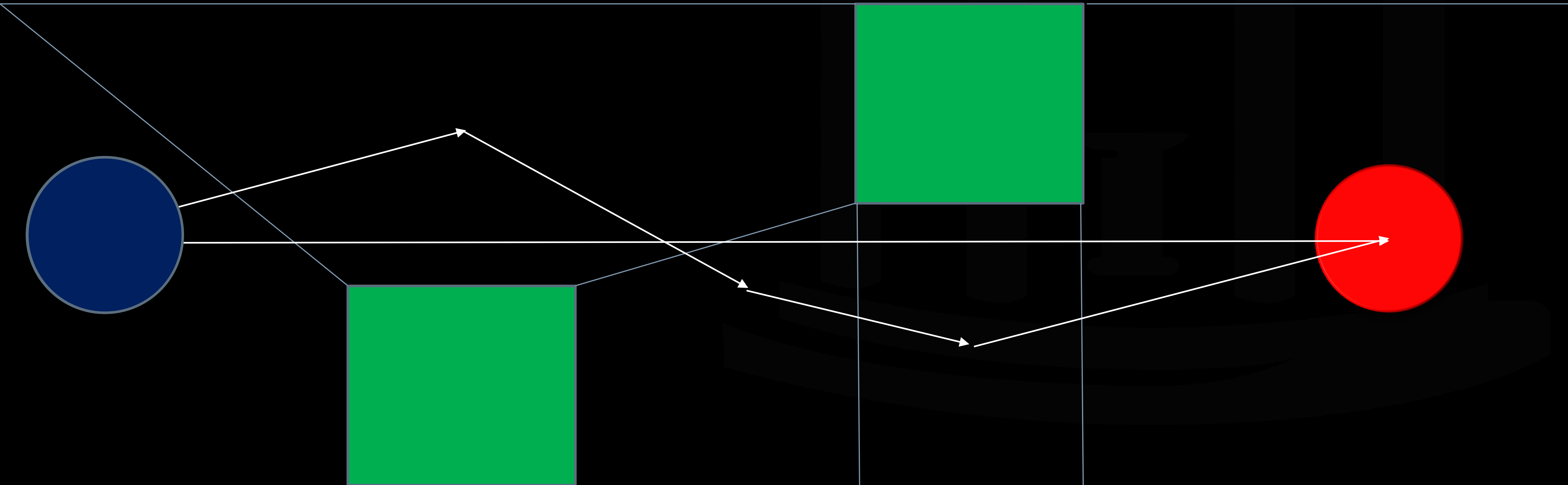
PLAN COMPUTATION

- Computes a path from Agent to Goal
  - VelocityComponent Element
    - Roadmap, Guidance Field, Navigation Mesh, Straight Line



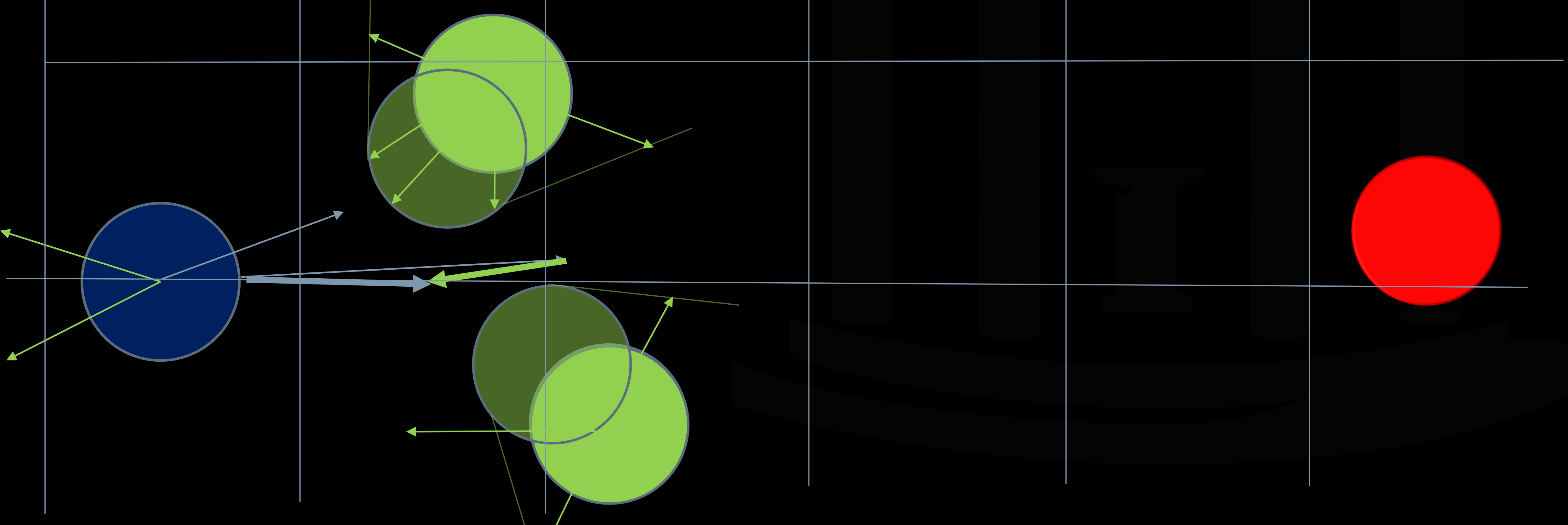
PLAN COMPUTATION

- Computes a path from Agent to Goal
  - VelocityComponent Element
    - Roadmap, Guidance Field, Navigation Mesh, Straight Line

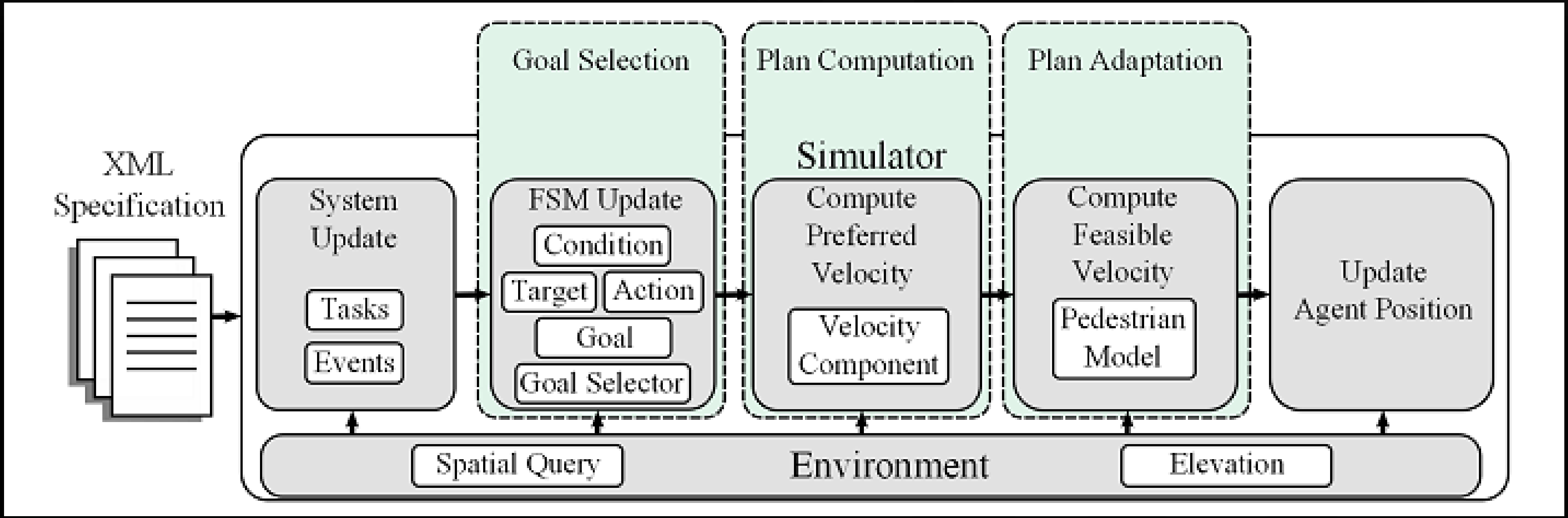


PLAN ADAPTATION

- How does the agent react to local conditions?
  - Pedestrian Model
    - Social Force, RVO, Cellular Automata

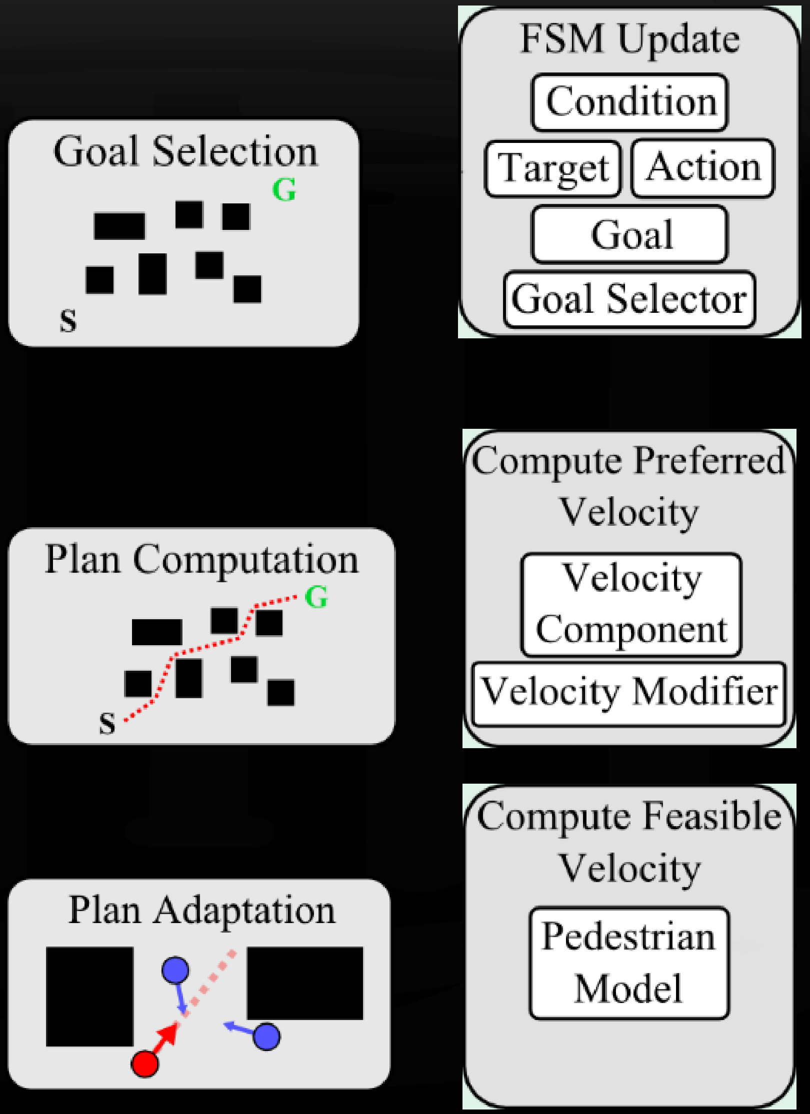


MENGE FRAMEWORK



MENGE FRAMEWORK

- Goal Selection
  - Nearest, farthest, biggest, least populous
- Transitions
  - Goal reached, timers, probabilistic
- Velocity Components
  - Roadmap, Navigation Mesh, Guidance Field
- Pedestrian Model
  - Local Navigation



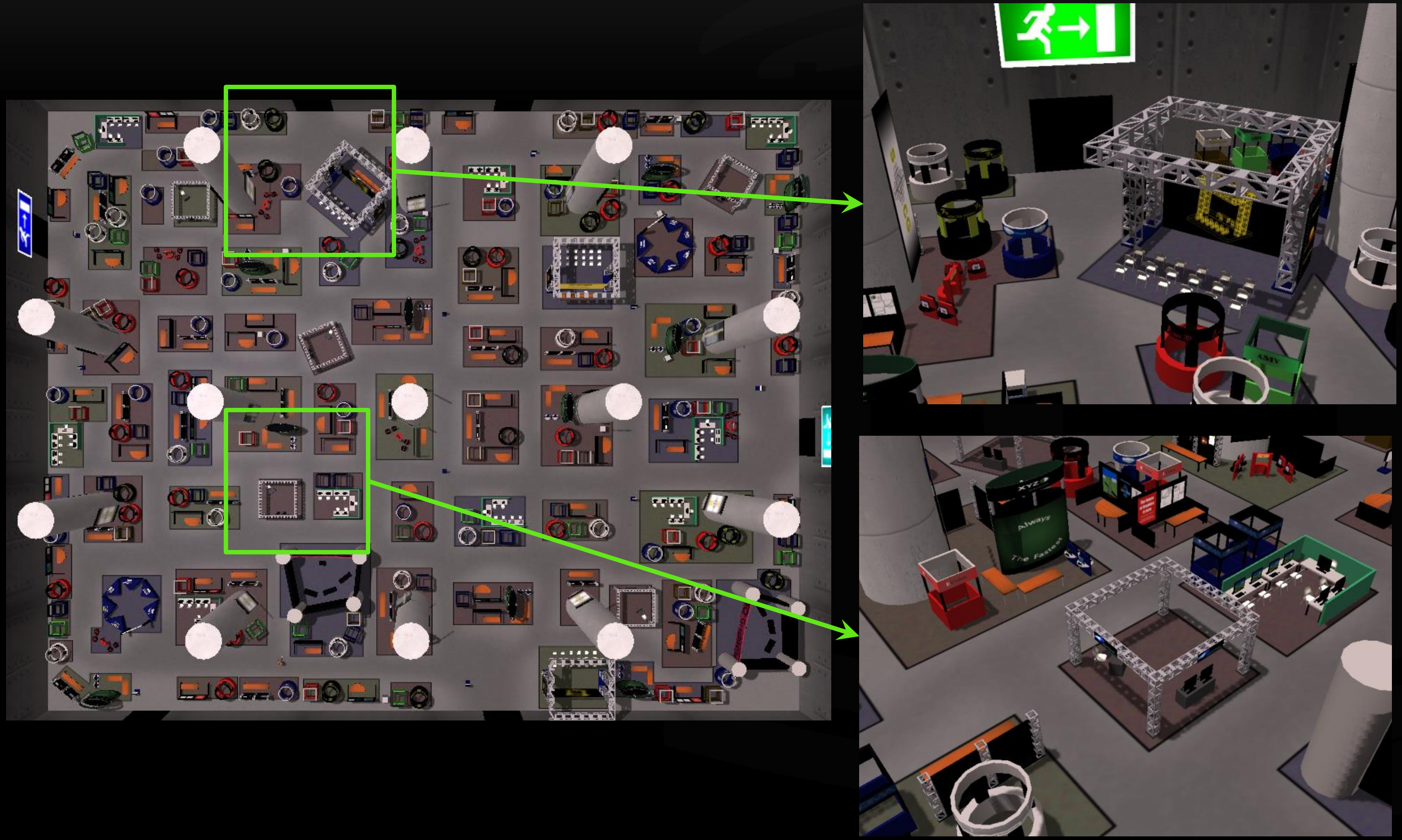
MENGE TUTORIAL

- Live demo
  - Project files
  - Scene specification
  - Run time parameters
  - Visualizer
- Documentation:
  - http://gamma.cs.unc.edu/Menge/

Structure

- Crowd Simulation
  - Multi-agent simulation basics
  - Menge
- Global Planners
  - Environment representation
  - Probabilistic Roadmaps (PRM)
  - Other global planners
- Assignment 1

ENVIRONMENT REPRESENTATION



ENVIRONMENT REPRESENTATION

- Visual representation more detailed than necessary
  - Very common for dynamics simulation
  - Typically true for navigation as well
- The more complex the representation, the more expensive

ENVIRONMENT REPRESENTATION

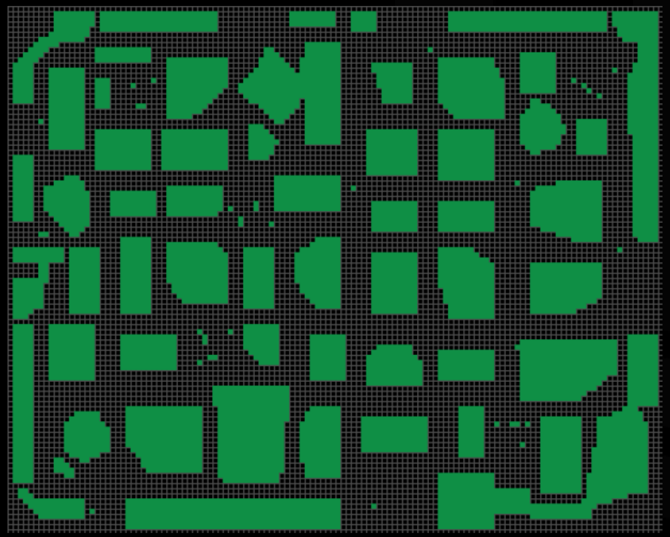
- Full 3D polygonal representation
  - Quite expensive
  - Details smaller than ~0.2 m probably don't matter.
  - Floor plan matters more than vertical space
    - (vertical clearance)

ENVIRONMENT REPRESENTATION

- 2D footprint
  - Saving an entire dimension
  - How much detail?
    - Coarse bounding volumes
      - Visually clear regions are no longer clear

ENVIRONMENT REPRESENTATION

- Keep polygons or rasterize to grid?
  - Grid offers simple “is colliding” query
  - (Compatible with potential field methods)



GLOBAL NAVIGATION

- Solving requires two things
  - Represent the navigable space and its relationships
  - Search the navigable space for optimal paths

Structure

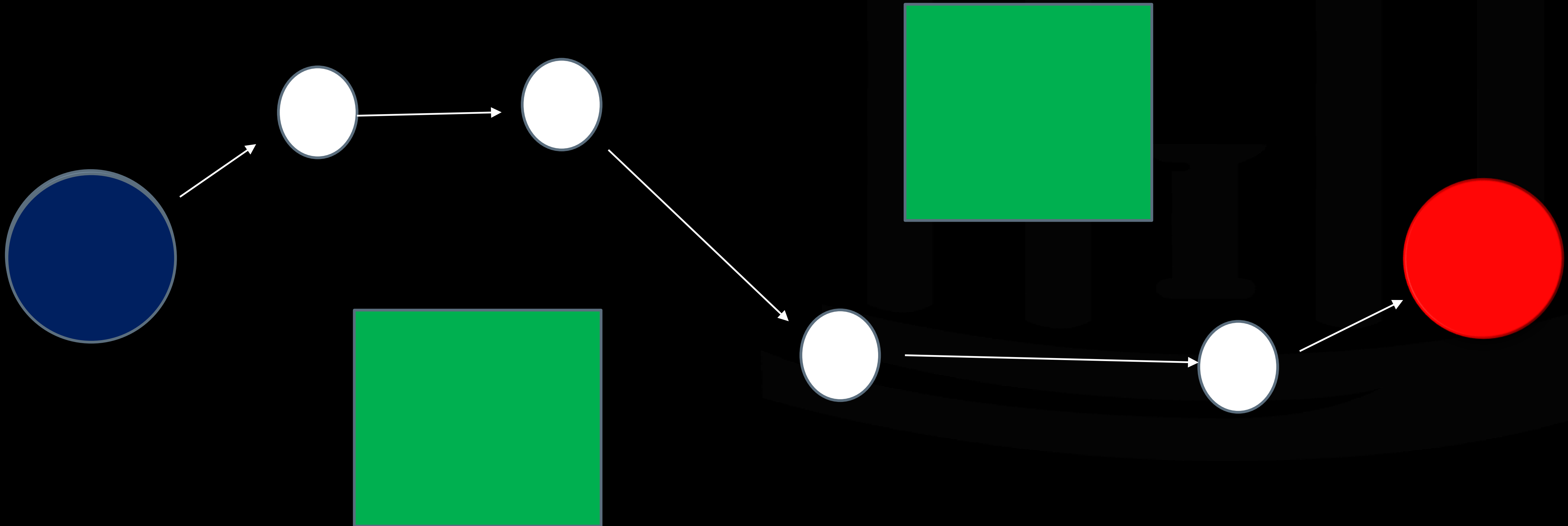
- Crowd Simulation
  - Multi-agent simulation basics
  - Menge

- Global Planners
  - Environment representation
  - Probabilistic Roadmaps (PRM)
  - Other global planners

- Assignment 1

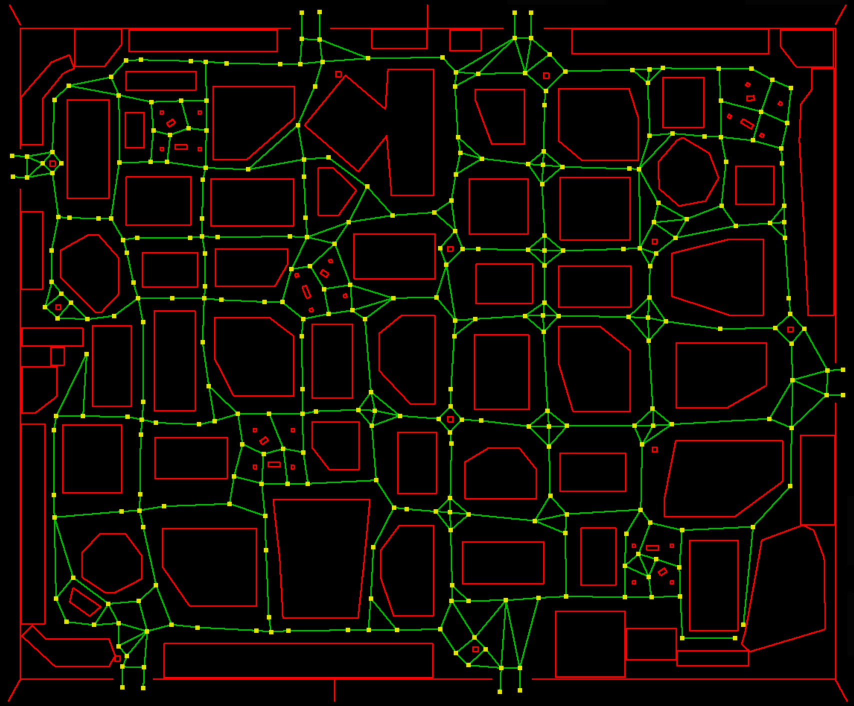
ROADMAPS

- Path composed of waypoints or milestones

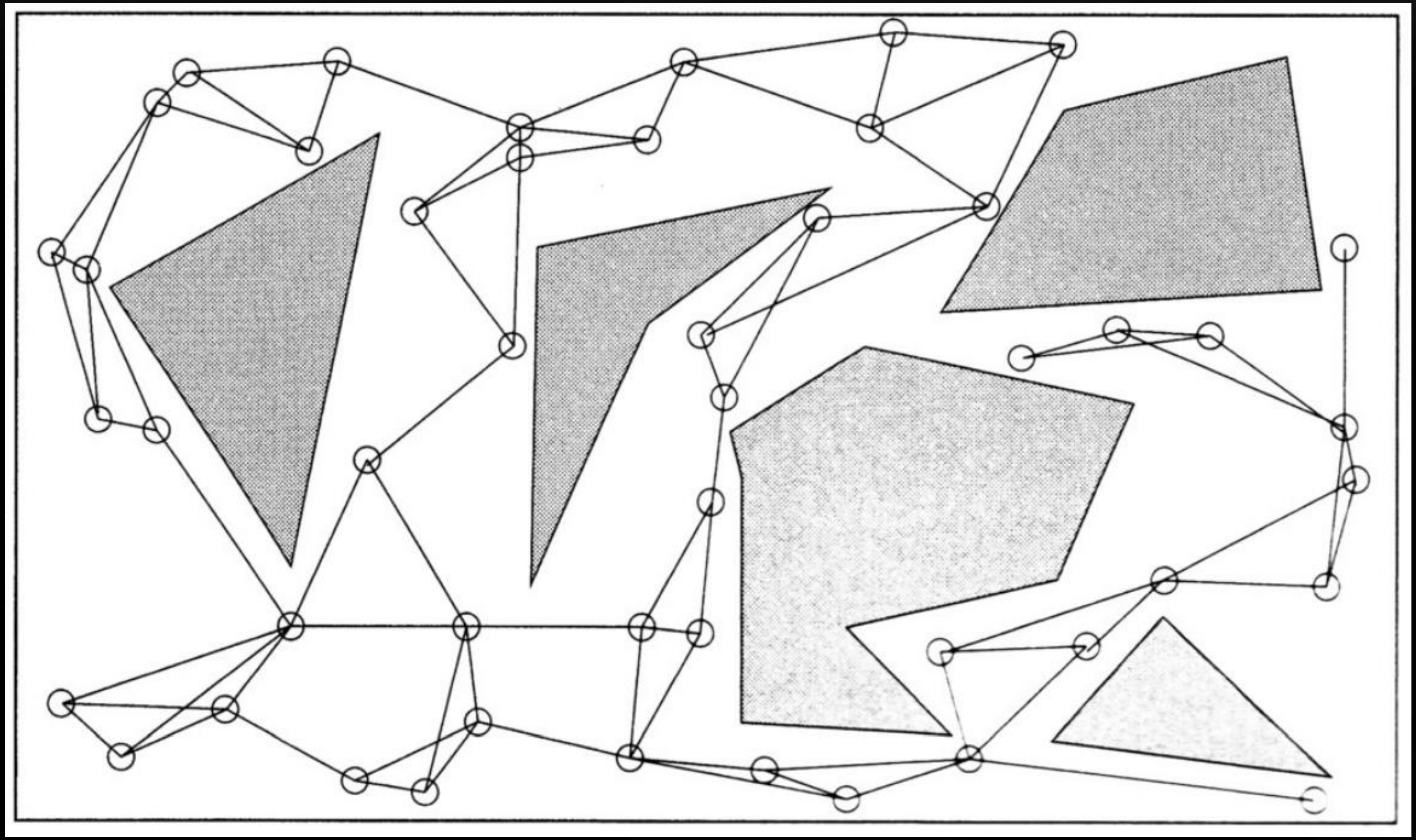


ROADMAPS

- A discrete *sampling* of free space
- Each sample is guaranteed to be collision free (CLEAR(q))
- Links between samples is guaranteed to be a collision free trajectory (LINK(q, q'))



ROADMAPS: CONSTRUCTION



ROADMAPS: CONSTRUCTION

Algorithm 6 Roadmap Construction Algorithm

Input:

n : number of nodes to put in the roadmap

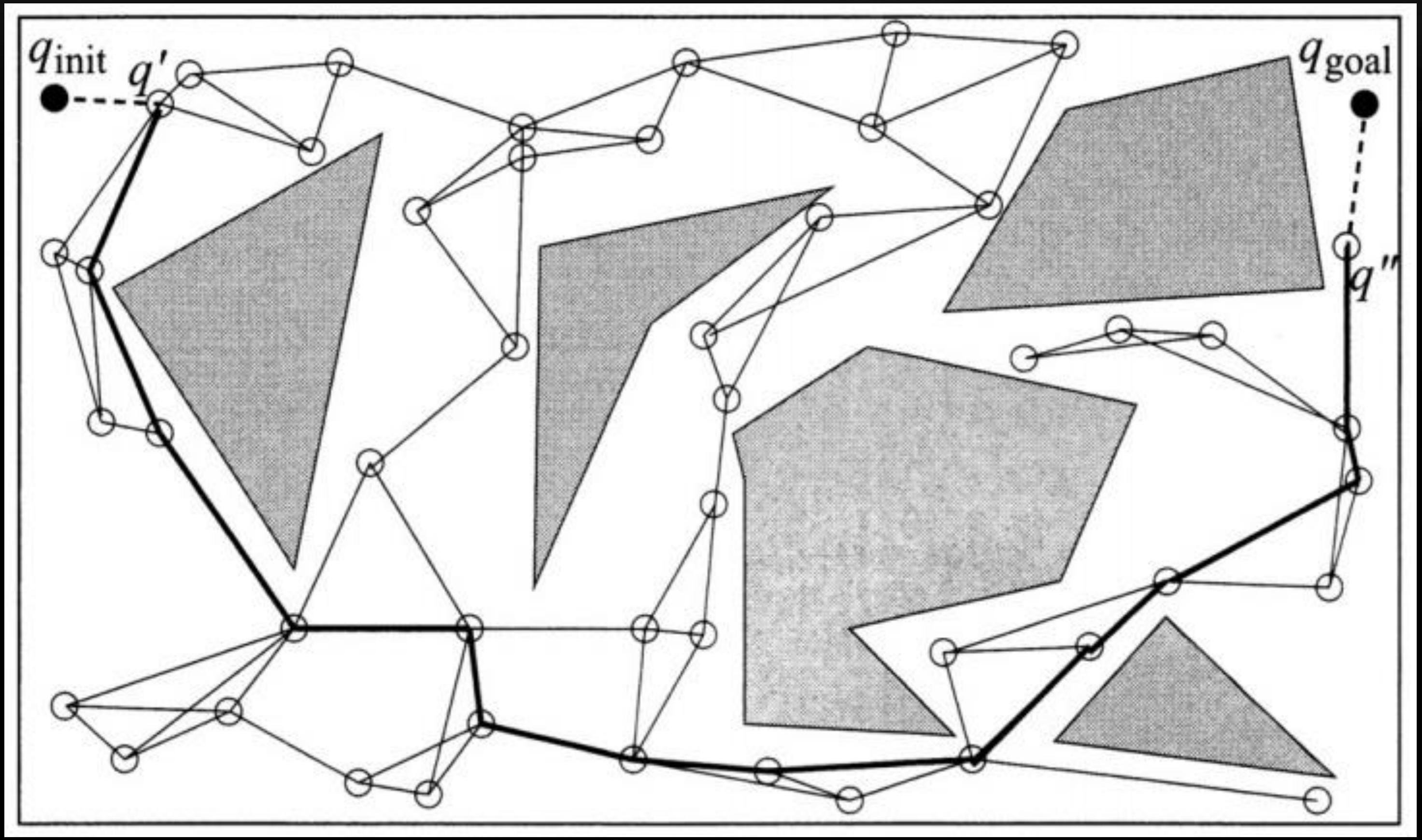
k : number of closest neighbors to examine for each configuration

Output:

A roadmap $G = (V, E)$

$V \leftarrow \emptyset$
$E \leftarrow \emptyset$
while $|V| < n$ **do**
  repeat
    $q \leftarrow$ a random configuration in $\mathcal{Q}$
  until q is collision-free
  $V \leftarrow V \cup \{q\}$
end while
for all $q \in V$ **do**
  $N_q \leftarrow$ the k closest neighbors of q chosen from V according to *dist*
  for all $q' \in N_q$ **do**
    if $(q, q') \notin E$ **and** $\Delta(q, q') \neq \texttt{NIL}$ **then**
      $E \leftarrow E \cup \{(q, q')\}$
    end if
  end for
end for

ROADMAPS: QUERY



ROADMAPS: QUERY

Input:

q_{init}: the initial configuration
q_{goal}: the goal configuration
k: the number of closest neighbors to examine for each configuration
$G = (V, E)$: the roadmap computed by algorithm 6

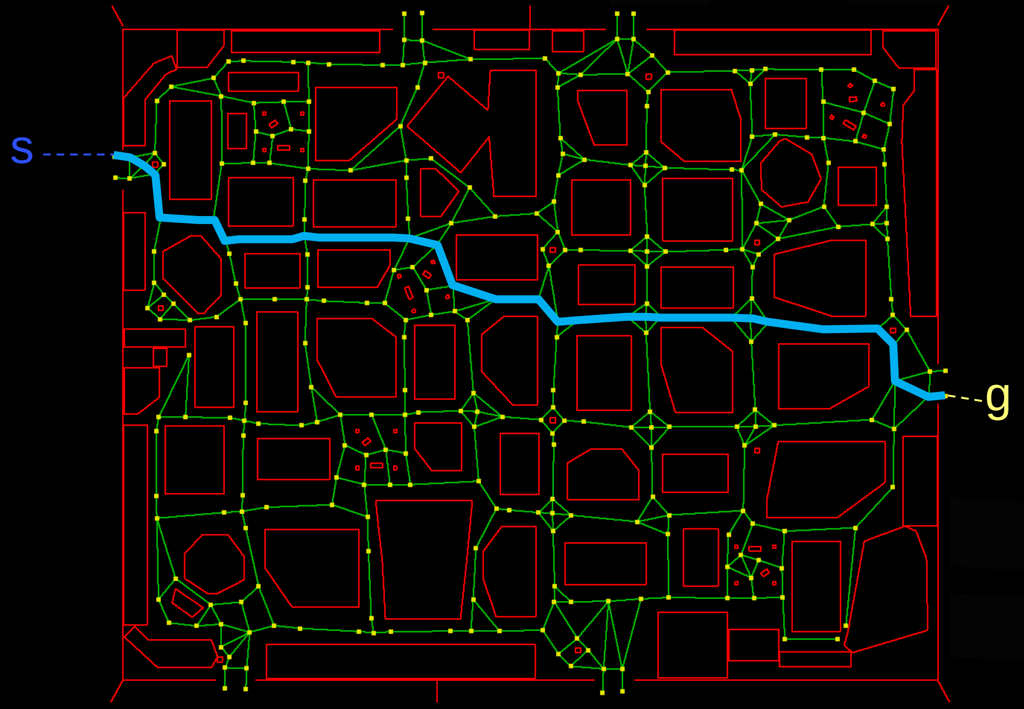
Output:

A path from q_{init} to q_{goal} or failure

$N_{q_{\text{init}}} \leftarrow$ the k closest neighbors of q_{init} from V according to *dist*
$N_{q_{\text{goal}}} \leftarrow$ the k closest neighbors of q_{goal} from V according to *dist*
$V \leftarrow \{q_{\text{init}}\} \cup \{q_{\text{goal}}\} \cup V$
set q' to be the closest neighbor of q_{init} in $N_{q_{\text{init}}}$
repeat
  if $\Delta(q_{\text{init}}, q') \neq \texttt{NIL}$ **then**
    $E \leftarrow (q_{\text{init}}, q') \cup E$
  else
    set q' to be the next closest neighbor of q_{init} in $N_{q_{\text{init}}}$
  end if
until a connection was succesful or the set $N_{q_{\text{init}}}$ is empty
set q' to be the closest neighbor of q_{goal} in $N_{q_{\text{goal}}}$
repeat
  if $\Delta(q_{\text{goal}}, q') \neq \texttt{NIL}$ **then**
    $E \leftarrow (q_{\text{goal}}, q') \cup E$
  else
    set q' to be the next closest neighbor of q_{goal} in $N_{q_{\text{goal}}}$
  end if
until a connection was succesful or the set $N_{q_{\text{goal}}}$ is empty
$P \leftarrow$ shortest path$(q_{\text{init}}, q_{\text{goal}}, G)$
if P is not empty **then**
  return P
else
  return `failure`
end if

ROADMAPS: QUERY

- Given start (s) and goal (g) positions
  - Link to roadmap
  - Find path on roadmap



ROAD MAP - USE

- Path
  - $P = [p_1, p_2, p_3, \ldots, p_n, g]$
    - Ordered list of waypoints
  - Preferred direction is direction toward “next” waypoint – the *target* waypoint
  - When do you change which waypoint is the target waypoint?
  - What if the target waypoint is lost?

ROAD MAP - USE

- When do you advance the target waypoint?
  - Simply measure distance (d) – d < D → reached
    - D – threshold
      - Big enough to be robust
      - Small enough that the next waypoint is reachable
  - What if the crowd keeps me from reaching the waypoint?
  - What if the crowd sweeps me PAST the waypoint along my path, but I don’t get close?

ROAD MAP - USE

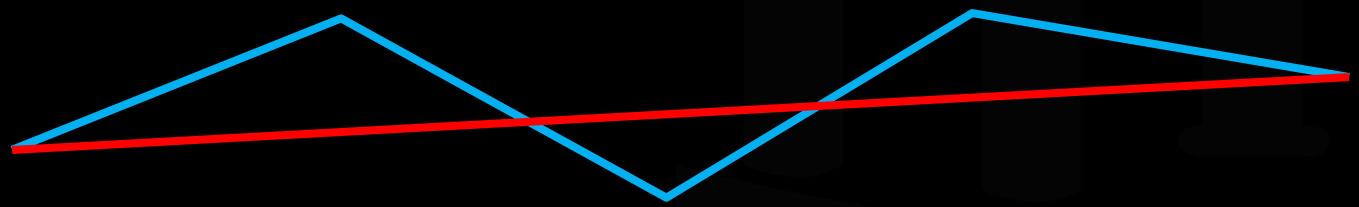
- When do you advance the target waypoint?
  - Visibility tests
    - Set the target waypoint to be the most advanced waypoint that is *visible*
    - This keeps the waypoint as far in “front” as possible
    - Also detects if the agent is pushed from the path

ROAD MAP - USE

- What if you lose sight of the target waypoint (pushed off the path)?
  - Replan
    - Create a new path
  - Rewind
    - Try testing previous waypoints (or successive)
    - Replan if all else fails
  - Remember
    - Remember where you were when you last could see it and work toward that

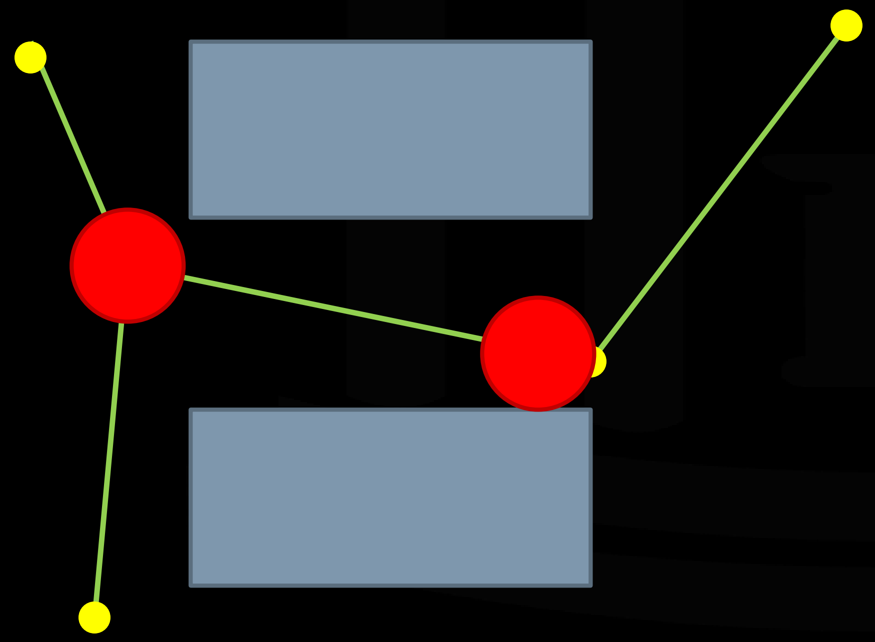
ROAD MAP - ANALYSIS

- Paths are dependent on sampling and connectivity
  - Path is only “optimal” w.r.t. the graph – not the environment
  - “Smoothing” the path helps
  - Earlier visibility query implicitly smooths the path
    - All but the last visible nodes are culled



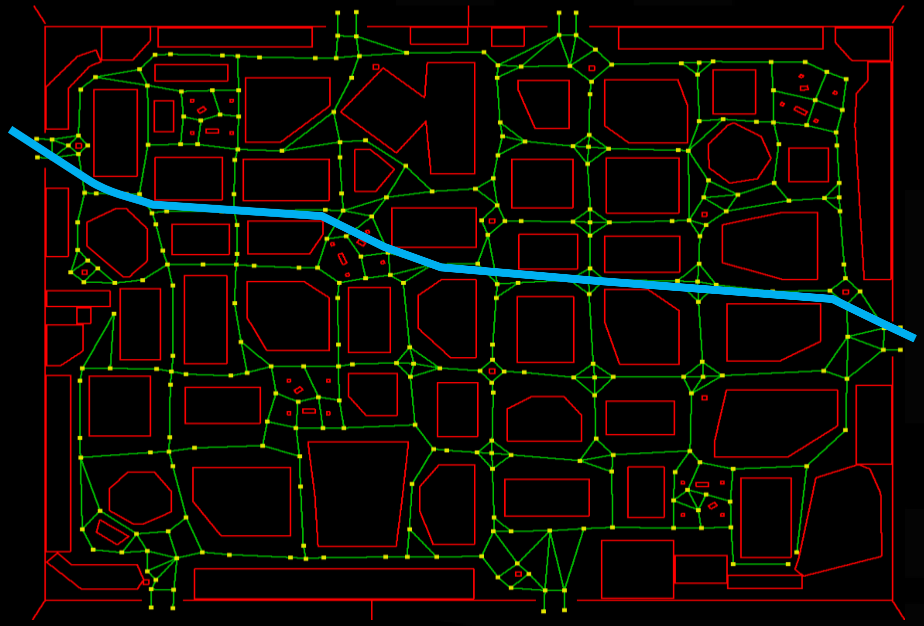
ROAD MAP - ANALYSIS

- That form of smoothness depends on the roadmap



ROAD MAP - ANALYSIS

- Paths are dependent on sampling and connectivity
  - How close it is to optimal depends on how close the roadmap samples come to the optimal path
  - No link → no path



ROAD MAP - ANALYSIS

- Clearance
  - Roadmaps are computed with one clearance in mind
    - What if there are entities of varying size?
    - Big agents will attempt to travel links with insufficient clearance on a small-agent map
    - Small agents will skip valid paths when using big-agent maps
  - Encode each link with maximum clearance

ROAD MAP - ANALYSIS

- More choices → more complexity
  - The only way to give agents more paths to reach their goal is to increase the complexity of the map
  - Search algorithms are worse than linear in the length of the optimal path (length = # of links)
    - Double the # of links, more than double the computation time
  - Also increase memory footprint

ROAD MAP - ANALYSIS

- Pros
  - Easy to create
  - Graph search straight-forward and generally effective
  - Pre-computed
  - Allows for non-planar topologies
- Cons
  - Hard to create a *good* roadmap
  - Paths non-optimal and non-smooth
  - Requires acceleration structure and visibility query to link to the graph

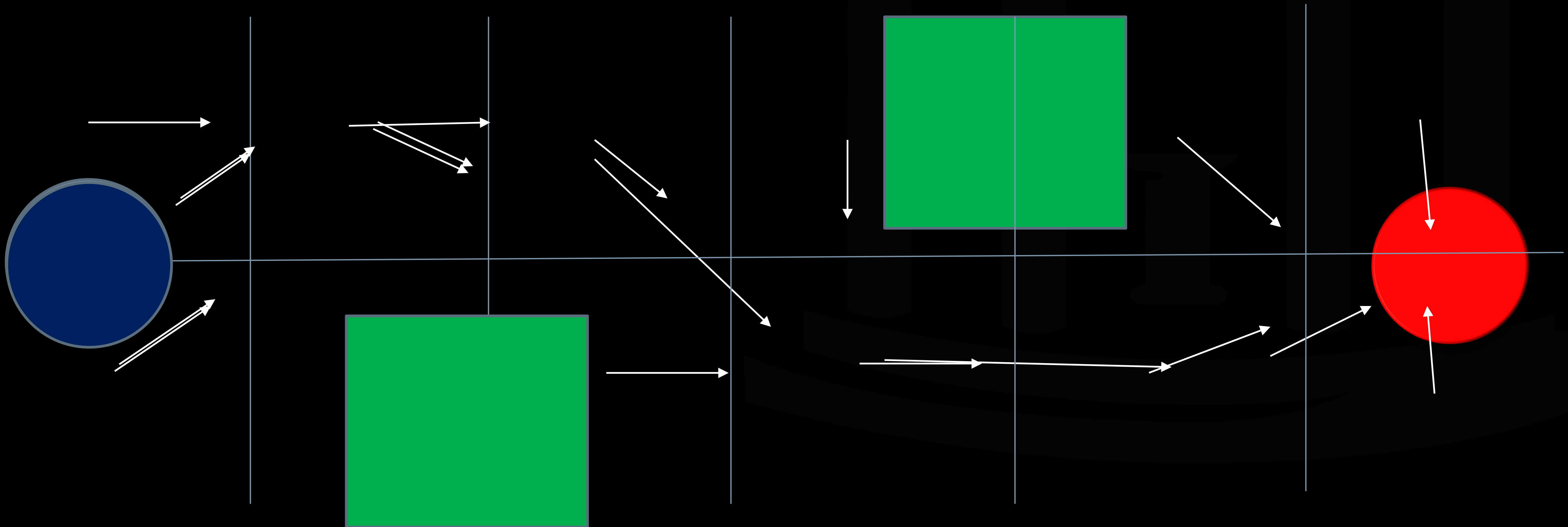
OTHER GLOBAL NAVIGATION METHODS

- Navigation Grids
  - Guidance field
  - Potential field

- Navigation Meshes

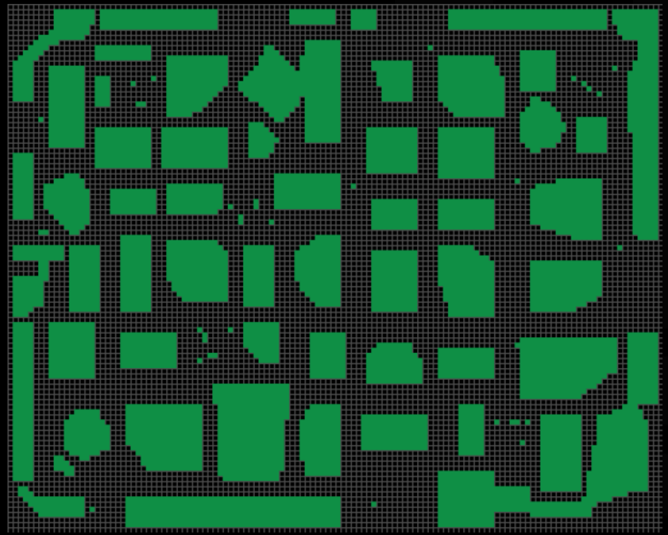
NAVIGATION GRID

- Discretize Free space into cells
- Plan optimal velocity at each cell



NAVIGATION GRID

- Discretization of space
  - Cells don't have to be uniform or square
    - Rectangle, hex, etc.
  - Cells are either marked as free or occupied
    - Non-boolean values possible



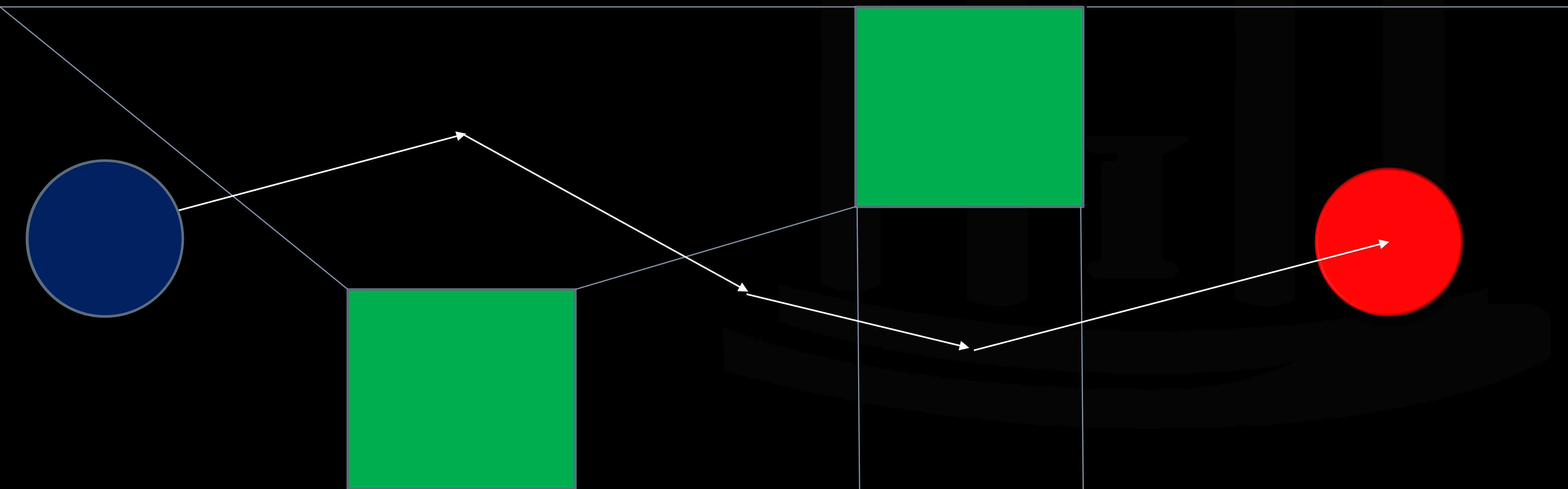
OTHER GLOBAL NAVIGATION METHODS

- Navigation Grids
  - Guidance field
  - Potential field

- Navigation Meshes

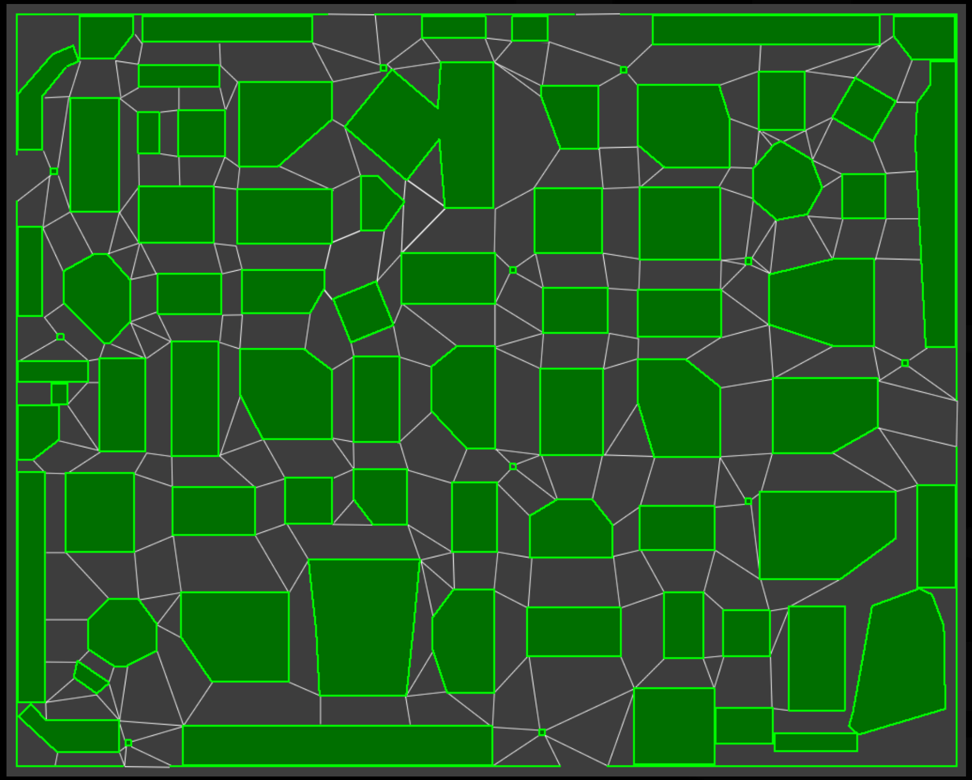
NAVIGATION MESH

- Discretize space as a set of connected convex polygens
- Graph search to find “envelope”
- Path planning through envelope



NAVIGATION MESH

- Discretization of free region into a mesh of convex polygons



GLOBAL PLANNERS

- How do roadmaps compare to navigation grids, or navigation meshes?

Structure

- Crowd Simulation
  - Multi-agent simulation basics
  - Menge

- Global Planners
  - Environment representation
  - Probabilistic Roadmaps (PRM)
  - Other global planners

- Assignment 1

ASSIGNMENT 1

- Revisit project specification
  - Changing properties
  - Changing global simulator
  - Changing local simulator
- Format for roadmaps

ASSIGNMENT 1

- Comparing simulations
  - Cost: the computational time taken to evaluate a single simulation step.
  - Stability: simulation model’s ability to take large time steps and still produce “accurate” results.
  - Efficiency: how much compute time is required to produce one second of simulated results?
    - = stability / cost
- How do you define accuracy?