

NavMeshes

CMSC425.01 Spring 2019

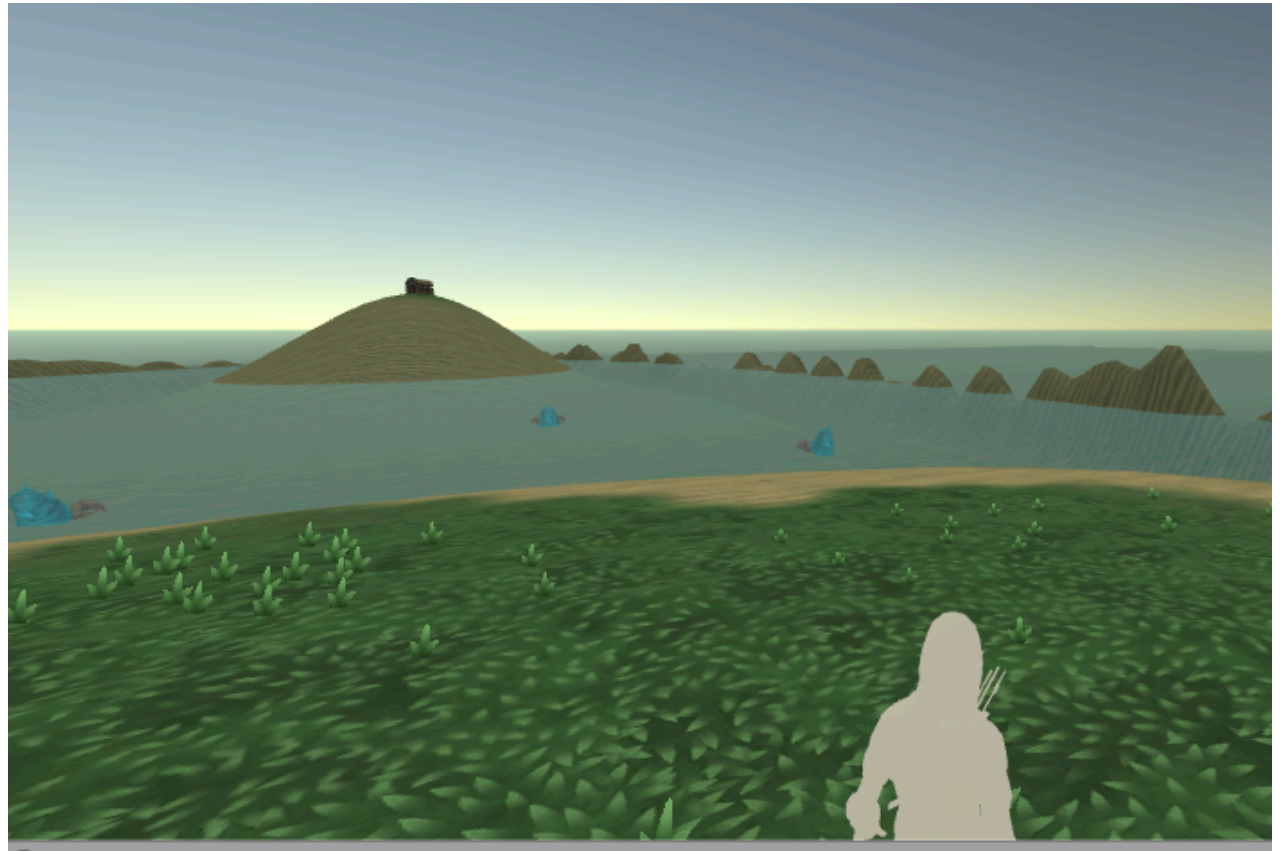
Still at tables ...

Administrivia

- Hw2 posted
- Project 2 outline ready, full description soon
- Practice exams linked
- Any choices on what we do after break?
 - Game design big subject, lots of topics we can choose from

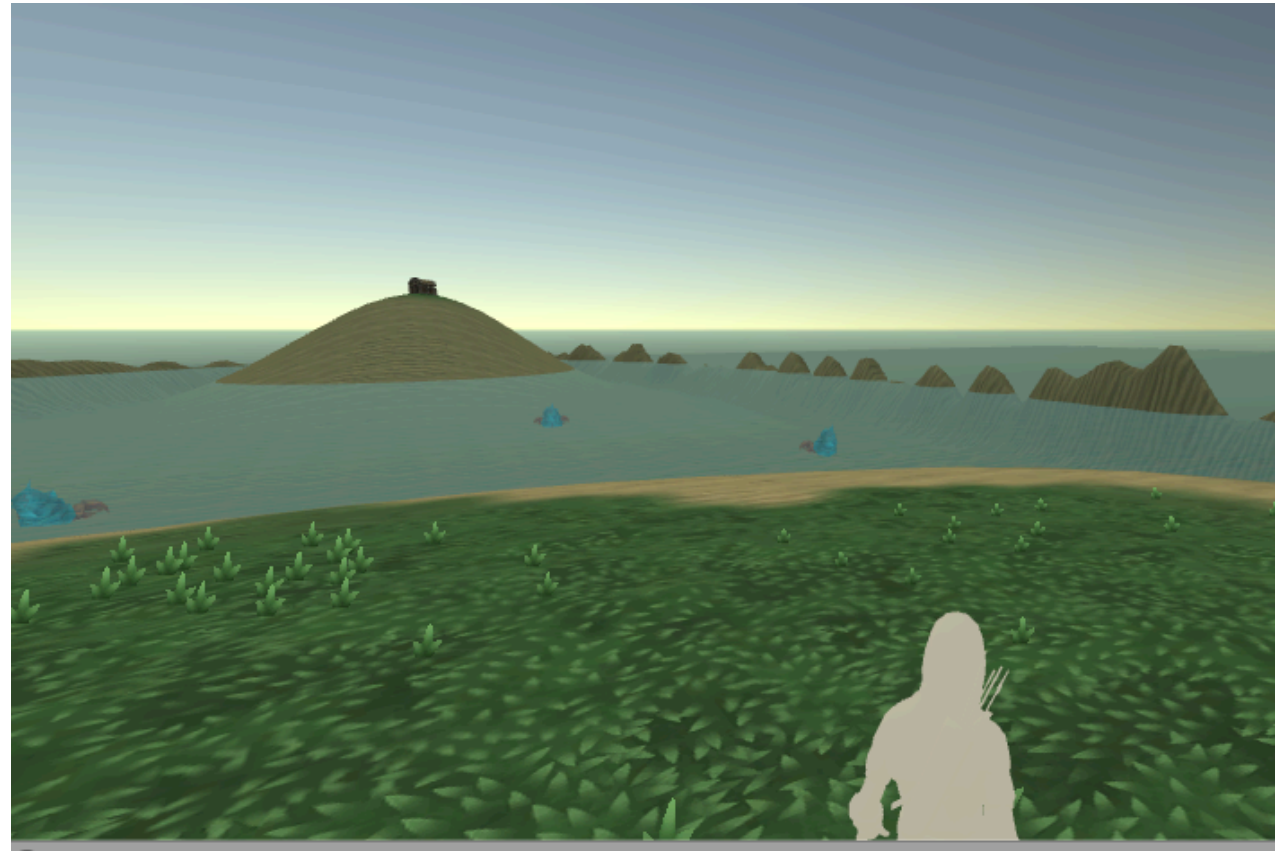
Project 2: Crab Soup, Anyone?

- Goal: Player must retrieve treasure from chest on far island
- Problem: Mutated Crabs will attack



Project 2: Crab Soup, Anyone?

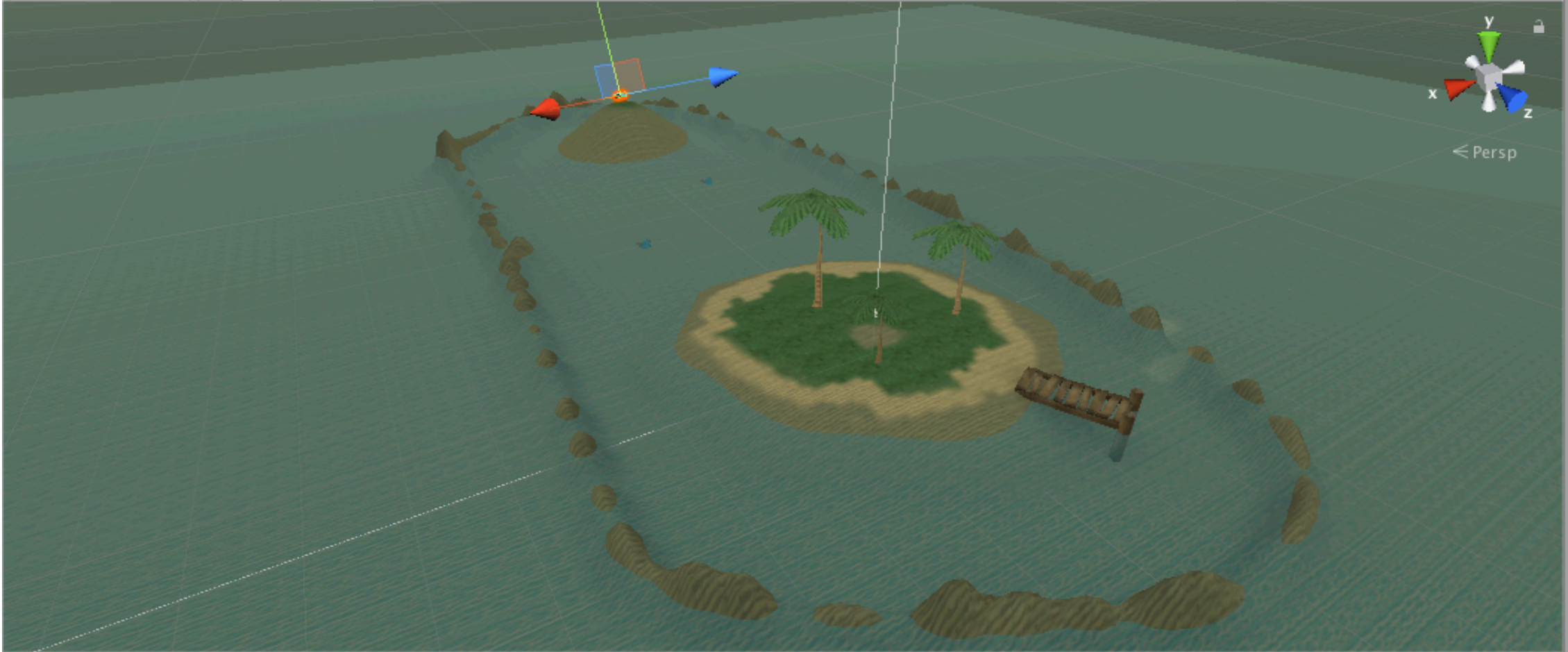
- Goal: Player must retrieve treasure from chest on far island
- Problem: Mutated Crabs will attack
- New Unity elements:
 - Terrain
 - NavMesh
 - Animated figures for Player and Crabs



Scene Console Asset Store Animator

Shaded 2D [Light] [Speaker] [Screenshot]

Gizmos [Search] All



Animation Game

Your instructor plays ...



Details

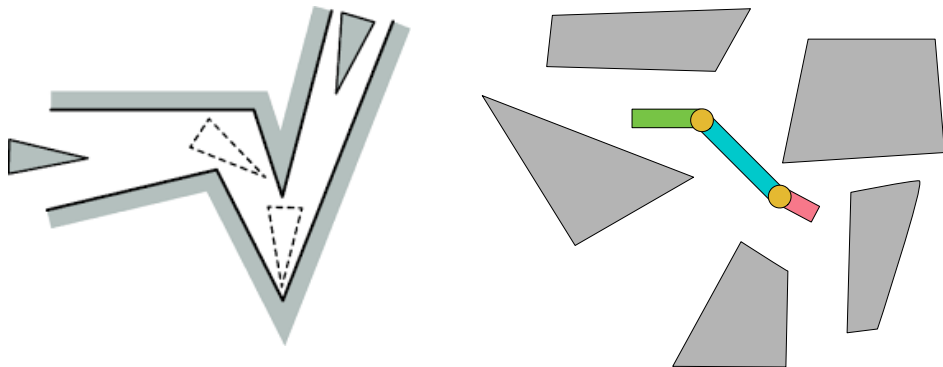
- Part 1: Make it to the treasure! Make a terrain with a NavMesh, navigate to the Treasure, and have the Crabs chase. Player humanoid, Crabs blocks. Win if you make it to the treasure without being “crabbed”
- Part 2: Animate Player and Crabs with Mixamo animations. Add a key that must be found to open chest, return to dock to win. Add UI graphic to show that key is picked up.
- Reach?: Make it underwater with swimming animation
- Free to: add elements to make it more playable, change theme if spirit is kept

Today's question

How NPCs move around:
NavMeshes

Navigation problems

- Navigating from place to place
- Dense crowd navigation
- Coordinated team movement
- Pursuit
- Moving complex/articulated shape
 - Piano movers problem(rigid)
 - Skeleton (articulated)



@UMD: Dinesh and Ming

- Dense crowd simulations
- How to move many agents naturally
- [Video](#)

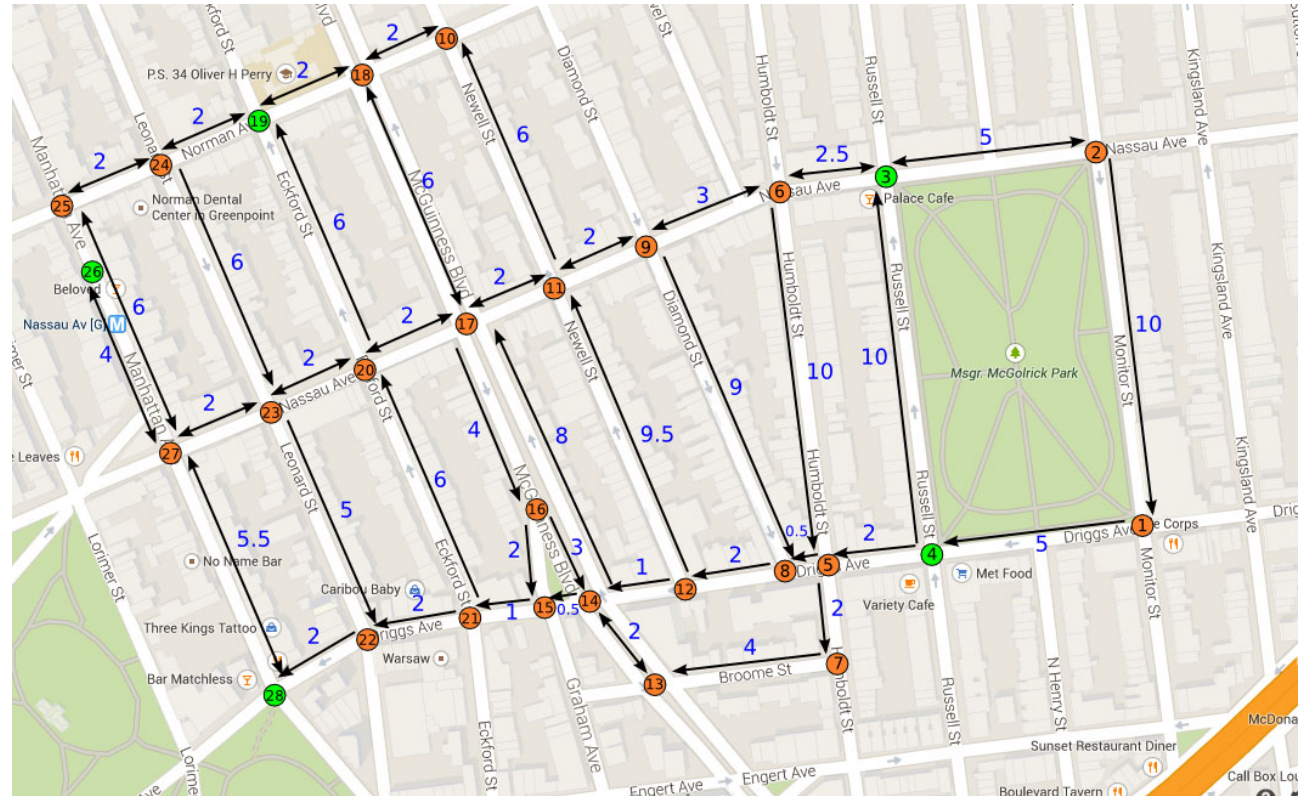
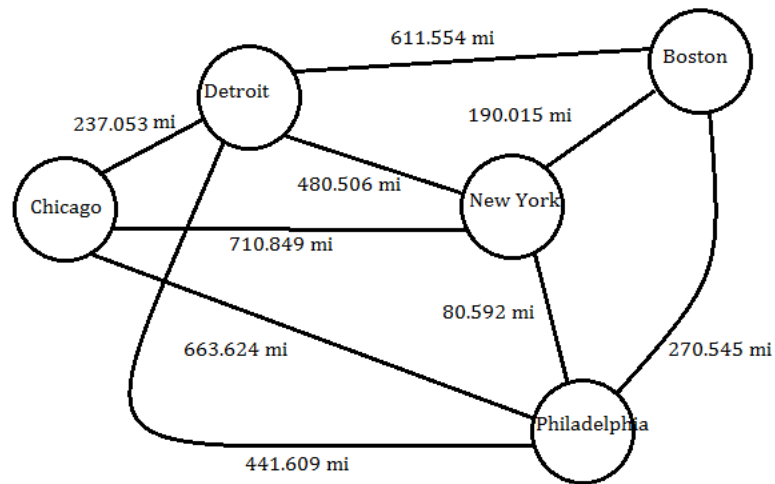


Navigation Version 1: Use a map

Create graph structure

Use Dijkstra's algorithm

Shortest path



Assumption: World is rigid,
limited lanes for movement

What is "shortest" path?

- Distance?

What is "shortest" path?

- Distance?
 - Speed?
 - Energy cost?
 - Exposure to enemy?
-
- Hex map
 - 6 directions
 - Terrain types => speed costs



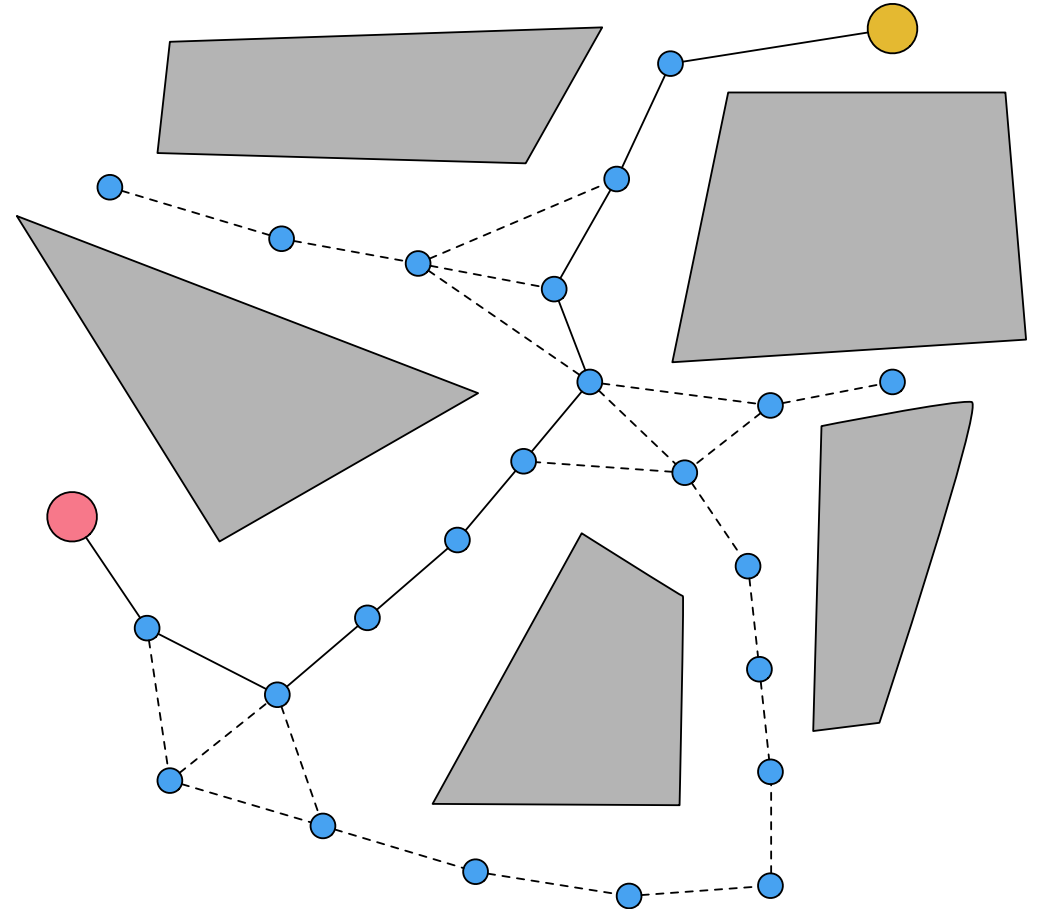
Navigation Version 2: Open Terrain

- Mix of obstacles and open spaces: "free space geometry"
- More options for direction
- How pick path?



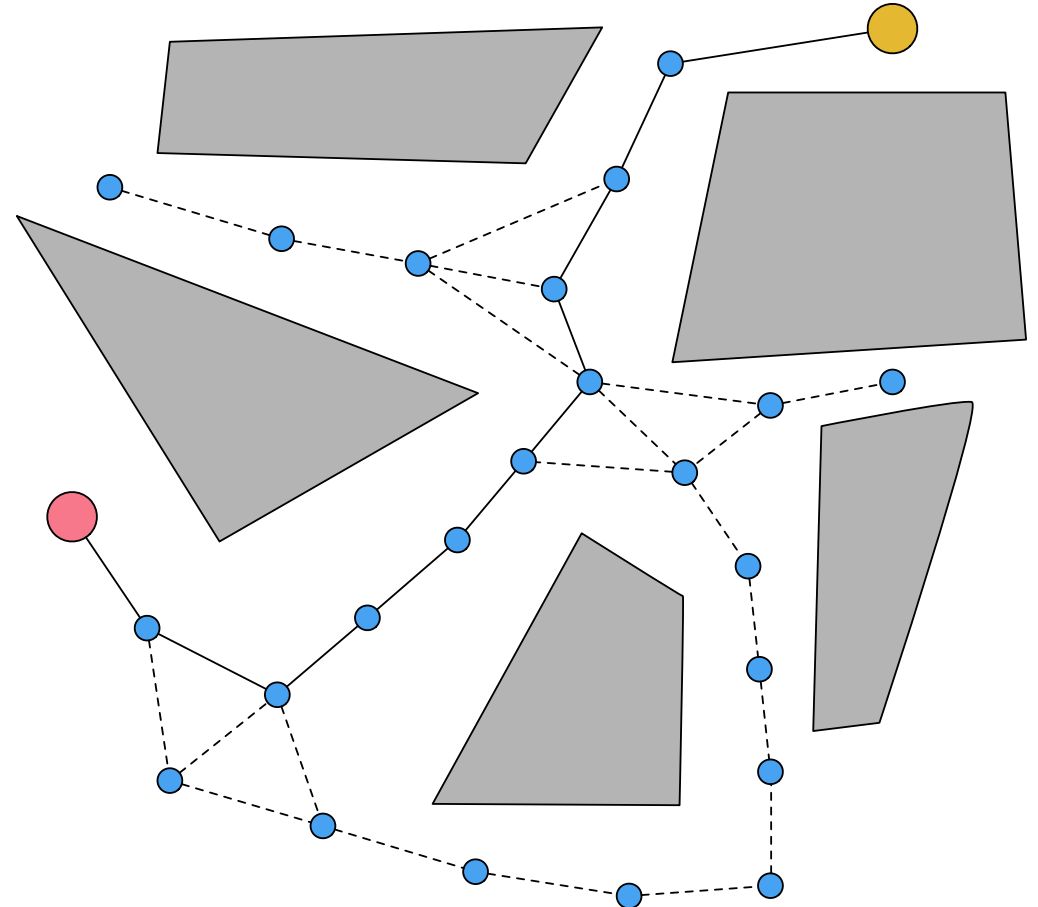
Solution 1: Waypoint roadmap

- Preprocess space into graph of waypoints
- Place waypoints along natural corridors



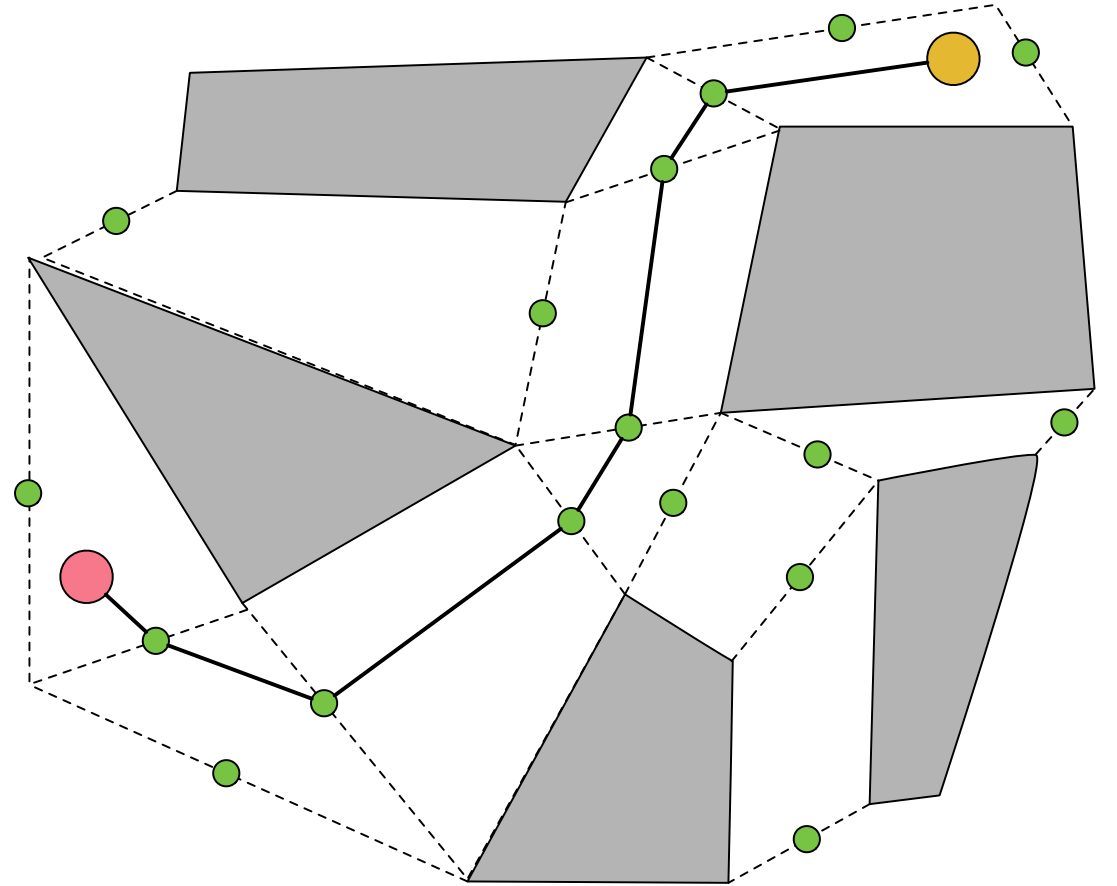
Solution 1: Waypoint roadmap

- Preprocess space into graph of waypoints
- Place waypoints along natural corridors
- Drawbacks
 - Could need lots of way points
 - Harder to plan for coordinated team movement



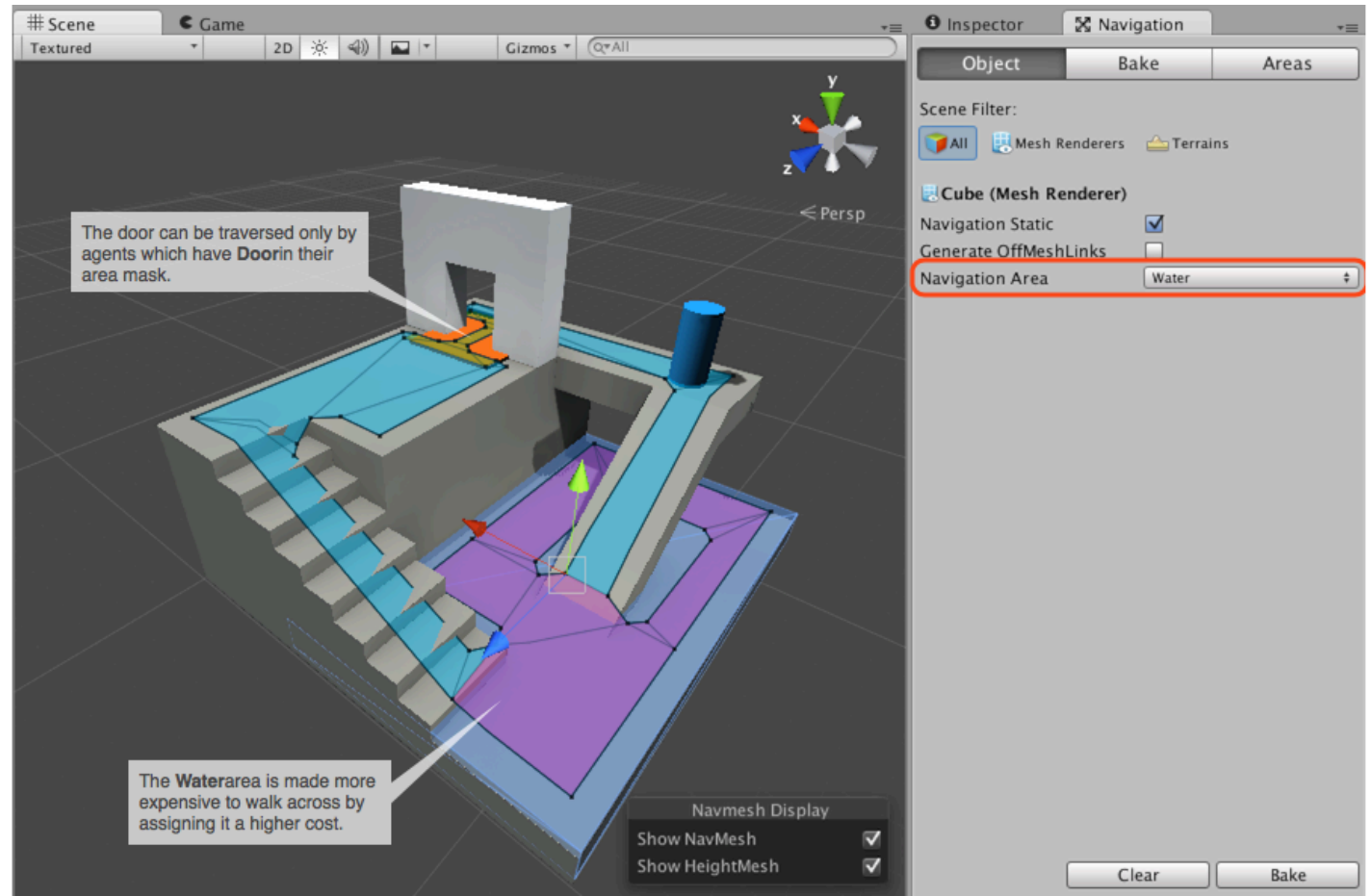
Solution 2: Navmesh

- Preprocess space into mesh of free areas as polygons
- Plan movement between polygons
 - Between edges
 - Between centers
- Multiple level graph search
 - First between regions
 - Then pick entry and exit points



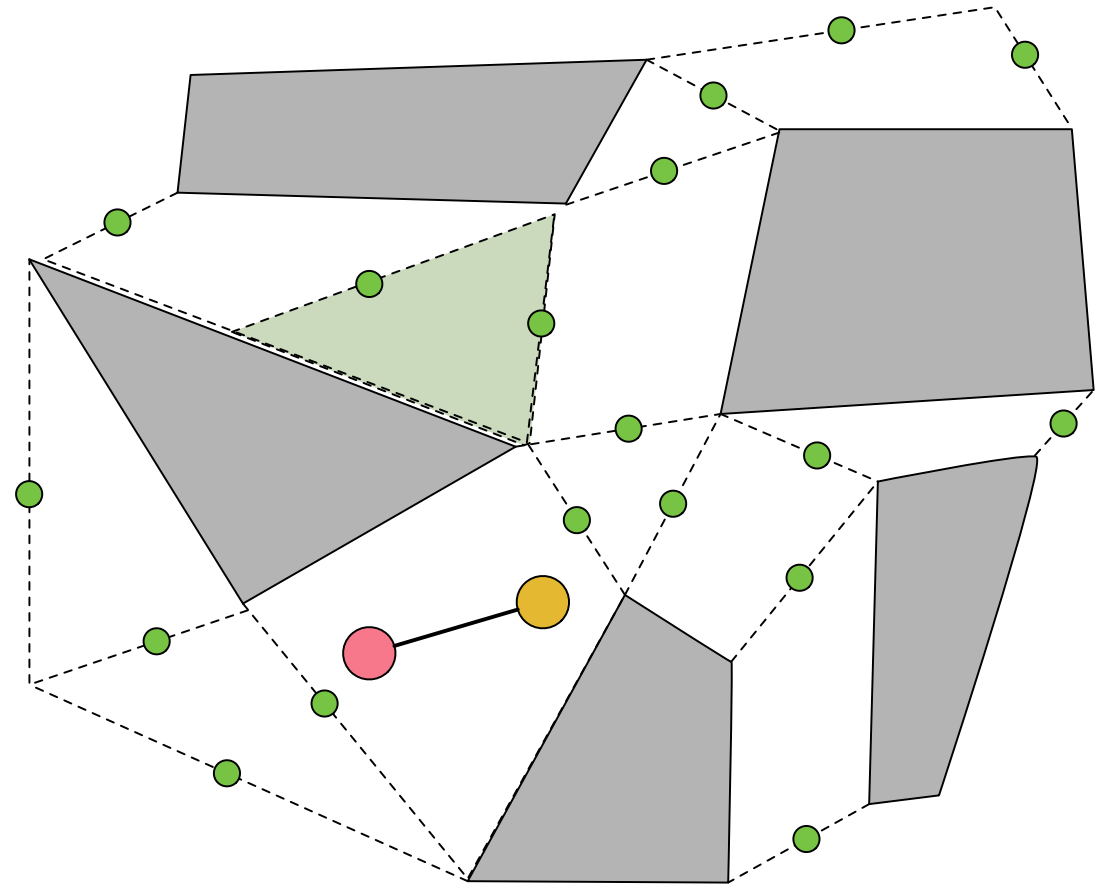
Navmeshes can be

- Labeled with different terrain types
- Set with "gates"
- 2D manifolds which are topologically complex (eg, non-planar)



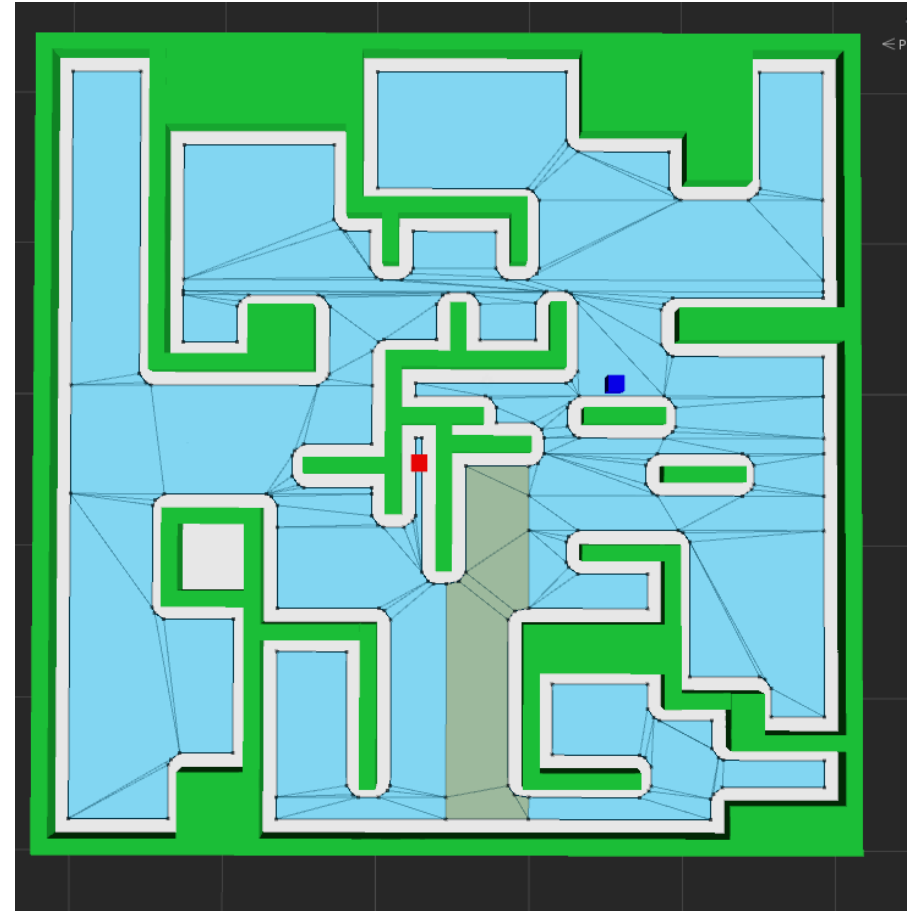
Using Navmesh

- Inside region use direct path
- Regions (and subregions) can be labeled with different terrain types and costs



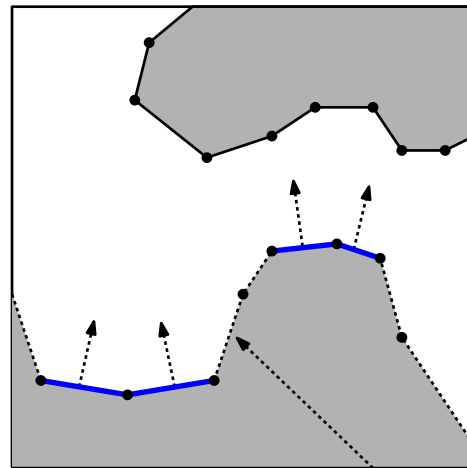
Creating Navmeshes from terrain

- Step 1: Find walkable surfaces
 - As large polygon "map"
- Step 2: Simplify boundaries
 - Simplify polygon "map"
- Step 3: Triangulate "map"
 - Cover with set of triangles

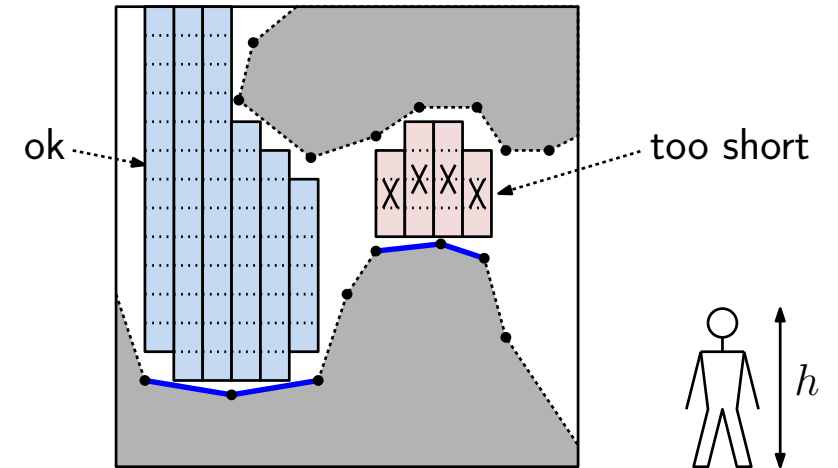


Creating Navmeshes from terrain

- Step 1: Find walkable surfaces based on agent height, width, slope ability
- Variation on piano movers problem



(a)



(b)

Creating Navmeshes from terrain

- Step 2: Simplify boundaries
 - Simplify polygon "map"
- Recursive refinement of straight line

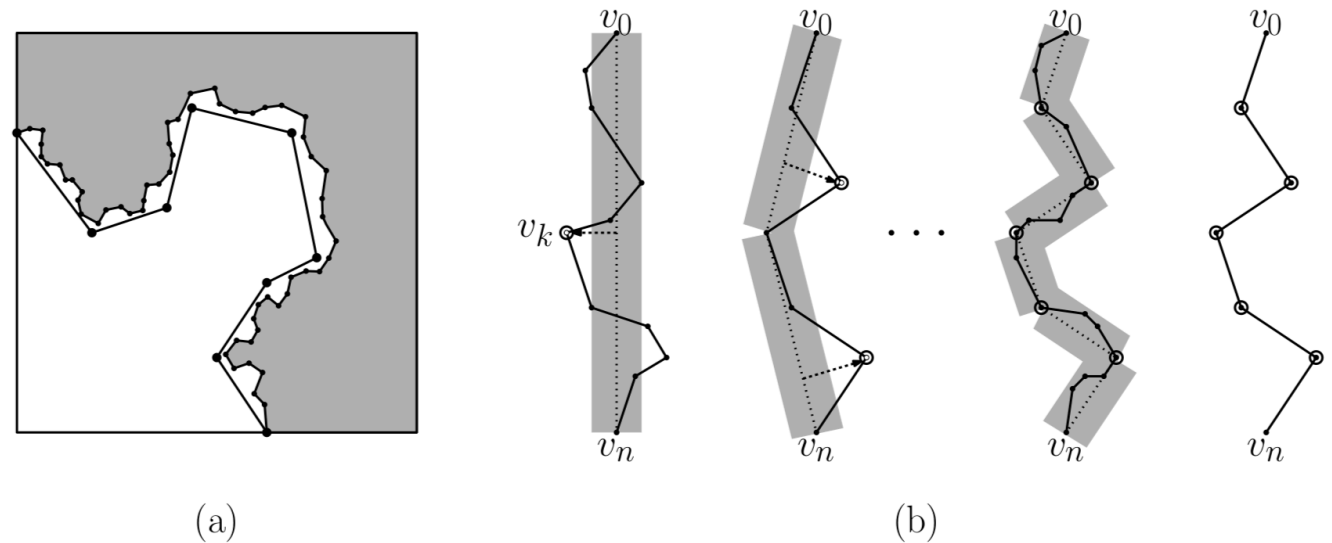
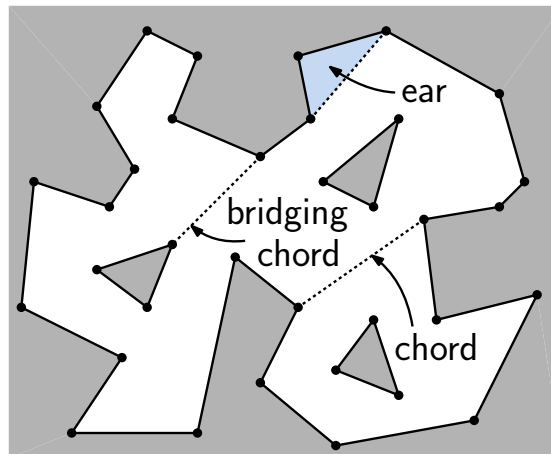


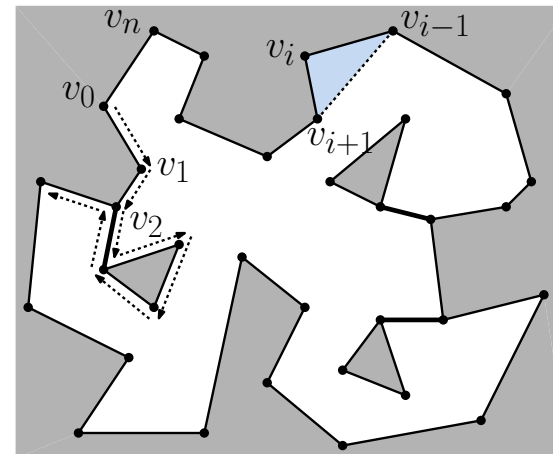
Fig. 3: The Ramer-Douglas-Peucker Algorithm.

Creating Navmeshes from terrain

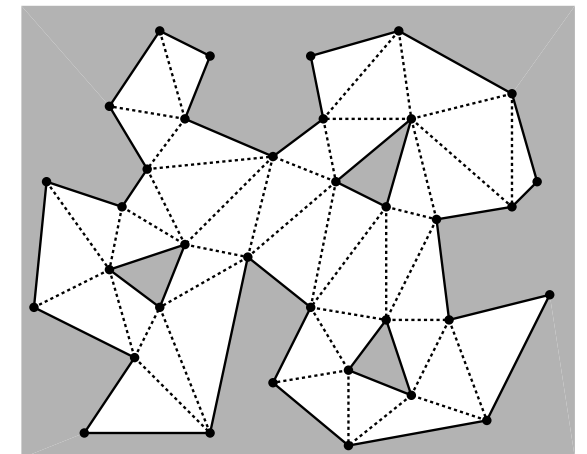
- Step 3: Triangulate "map"
 - Cover with set of triangles
- Bridge holes
- Cut ears (!)



(a)



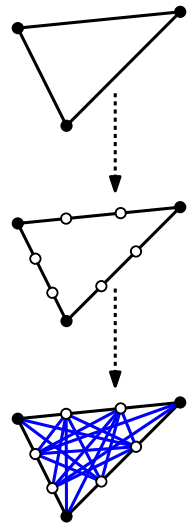
(b)



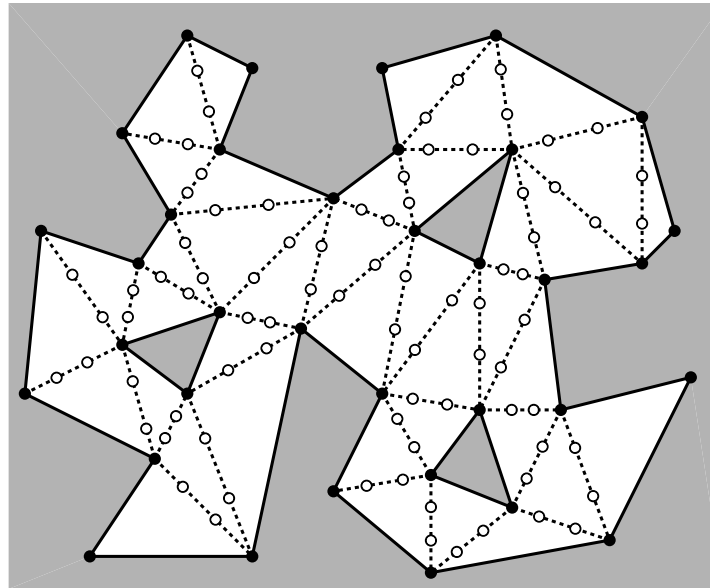
(c)

Use Navmesh: find path

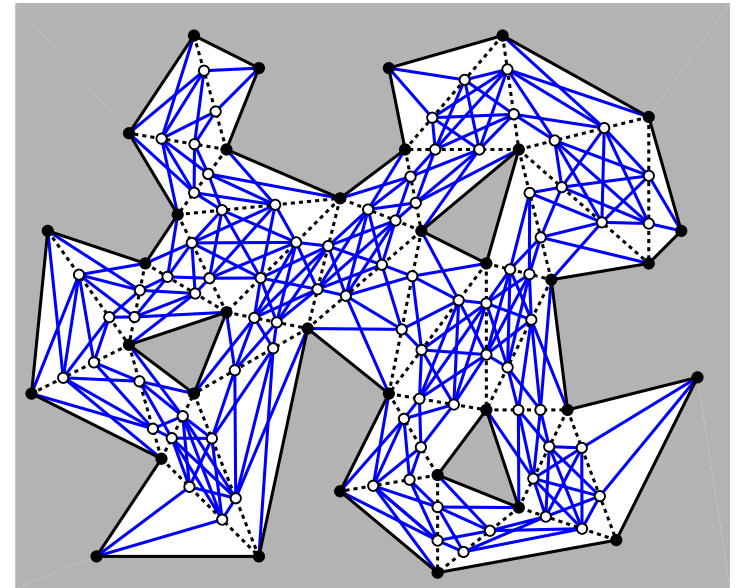
- Discretize by adding points
- Find shortest path



(a)



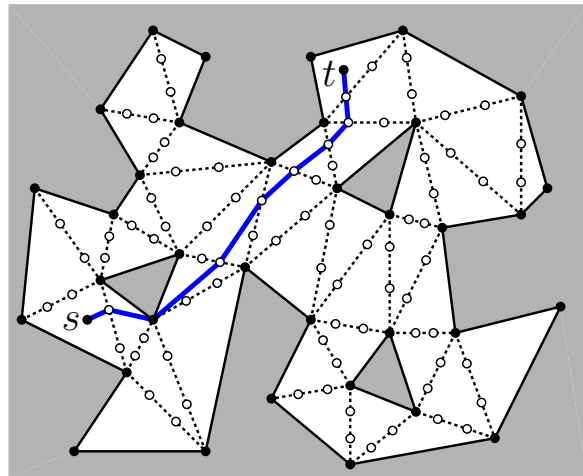
(b)



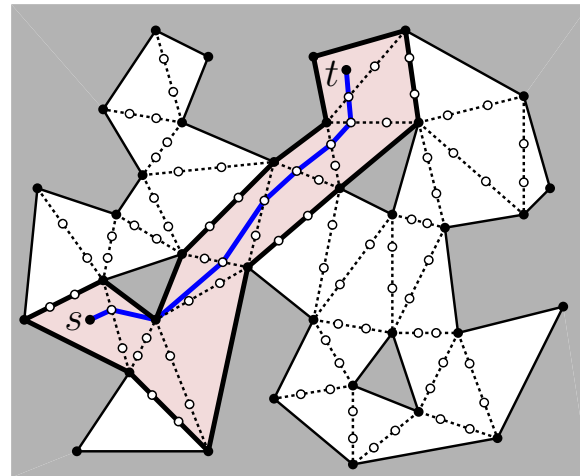
(c)

Use Navmesh: refine

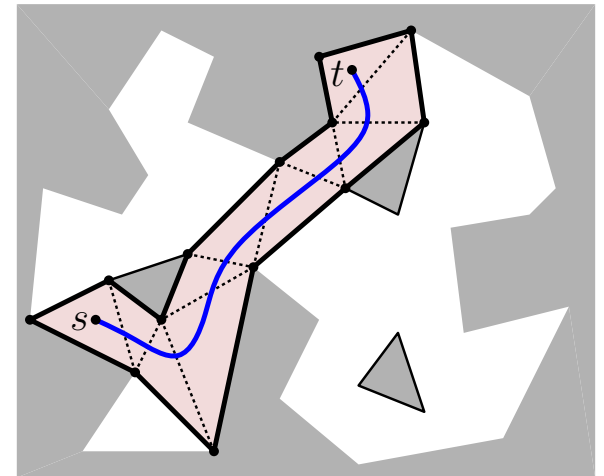
- Smooth and clean path



(a)



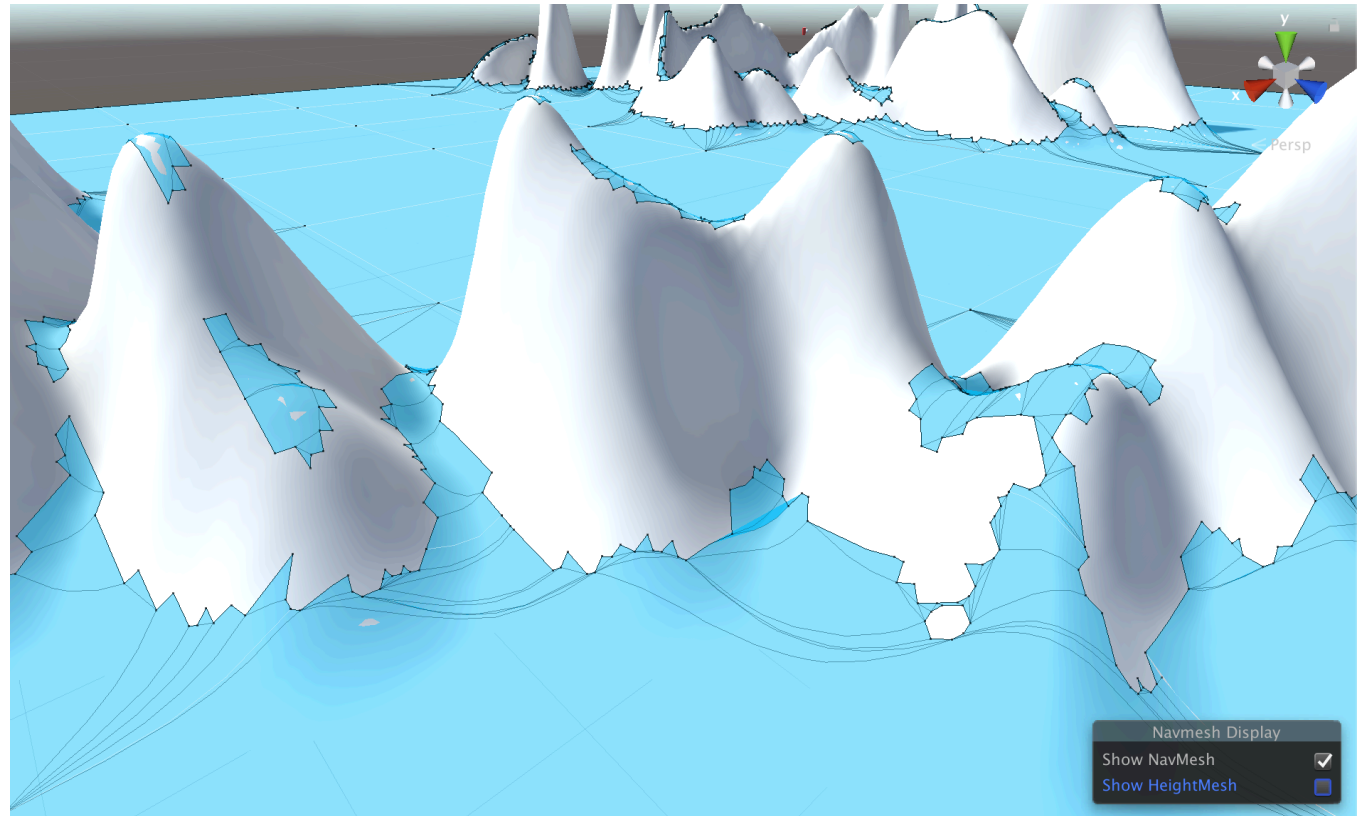
(b)



(c)

Navmeshes in Unity

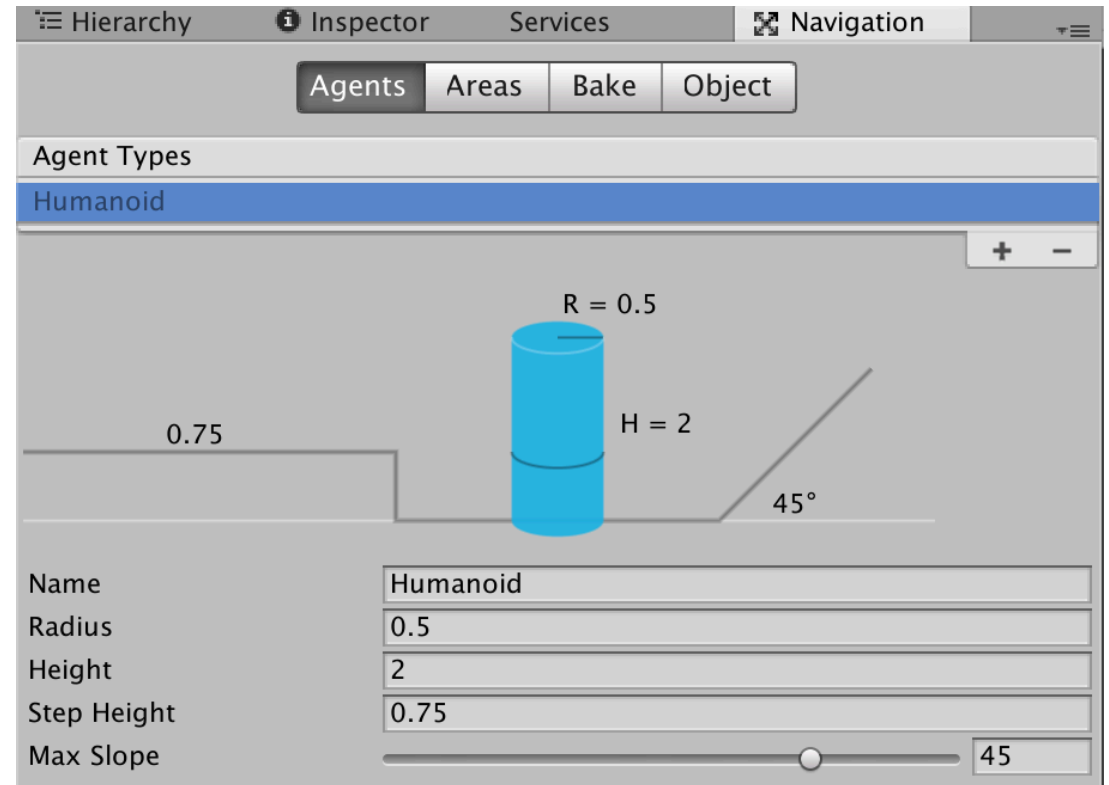
- Create terrain
 - Terrain editor
- Set agent navigation properties
 - Height, width of agent
 - Slope capability
- Bake Navmesh
 - Finds navigable regions
 - Creates mesh



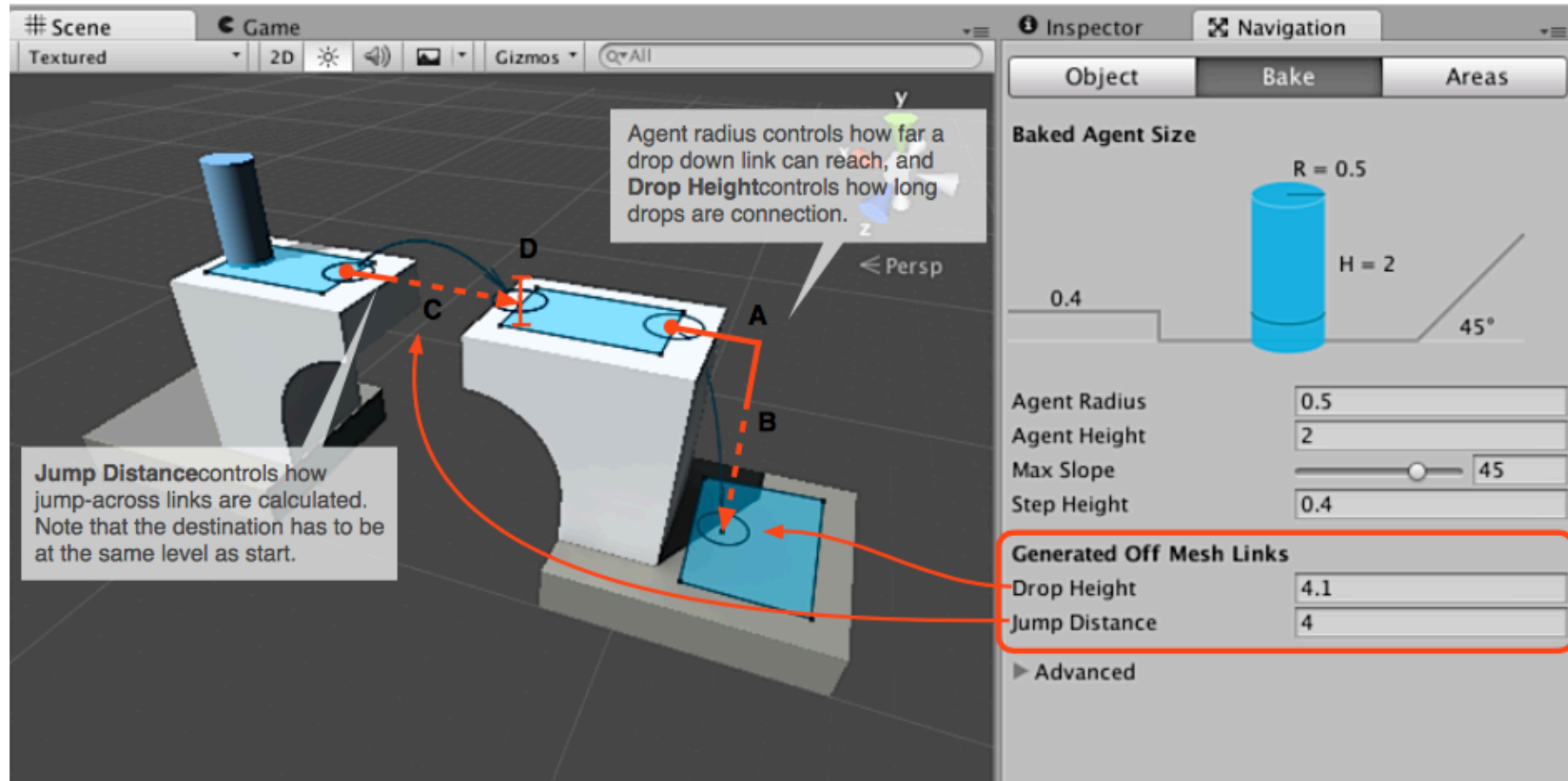
Agent properties

- Radius, Height
- Step height
- Max slope

- Can have multiple agent types



Off mesh links



Question

- How does this all relate to colliders?

Readings

- Look at Unity manual and tutorials
- Terrain
 - <https://docs.unity3d.com/Manual/terrain-UsingTerrains.html>
- NavMesh
 - <https://unity3d.com/learn/tutorials/topics/navigation/navmesh-agent>
 - <https://docs.unity3d.com/Manual/nav-BuildingNavMesh.html>
- Animation
 - <https://unity3d.com/learn/tutorials/topics/animation/animate-anything-mecanim>
 - <https://unity3d.com/learn/tutorials/s/animation>