Movement for Gaming

From Chapter 3 “Artificial Intelligence for Games”

authored by I. Millington & J. Funge

Basics

- Characters in games usually move
  - Movement calculation often needs to interact with the “Physics” engine
    - Avoid characters walking through each other or through obstacles

  Traditional:
    - Characters simply move (often at fixed speed) without regard to how physical objects accelerate or brake

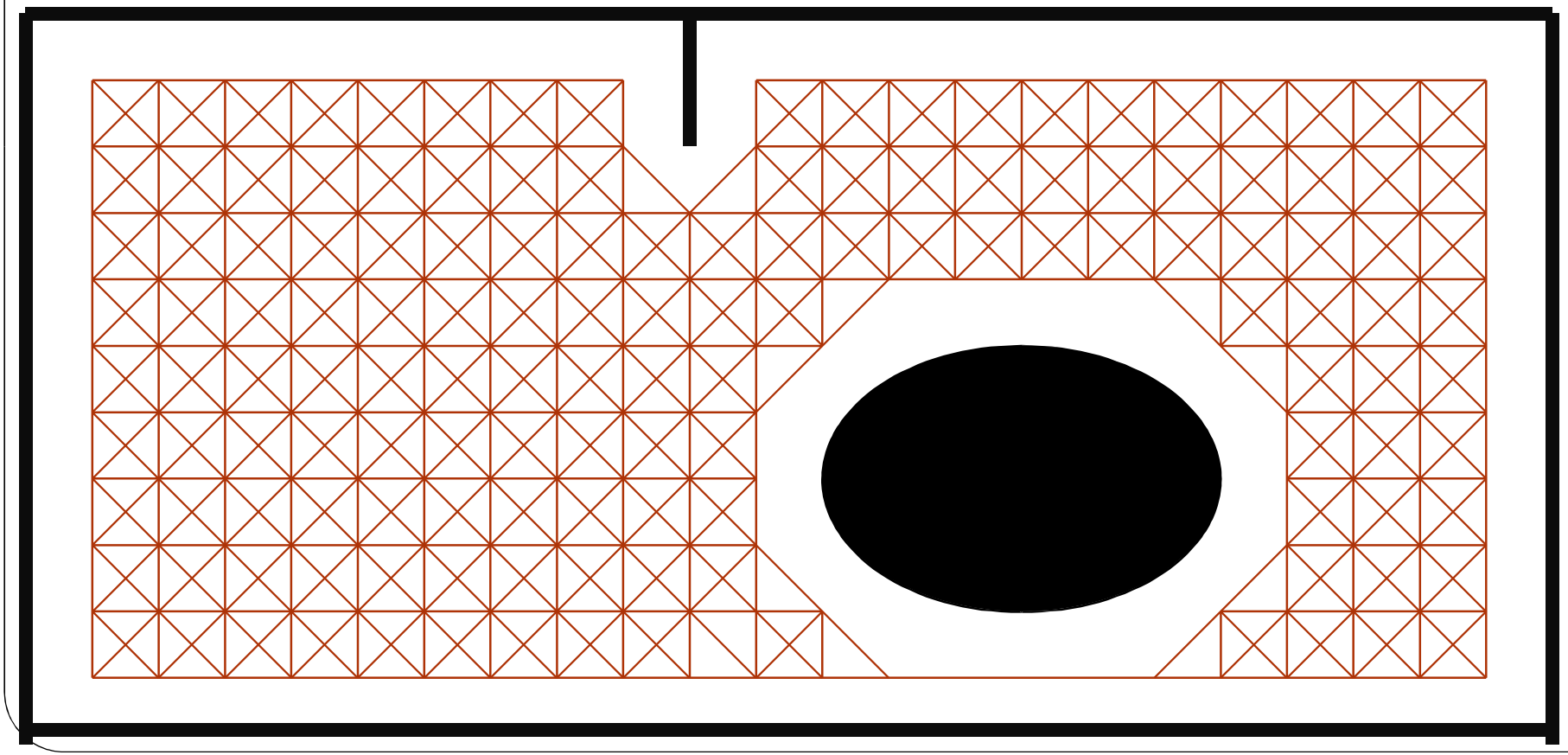
  Newer approach:
    - Characters accelerate and turn based on physics

Relationship with Pathfinding

설명 없음

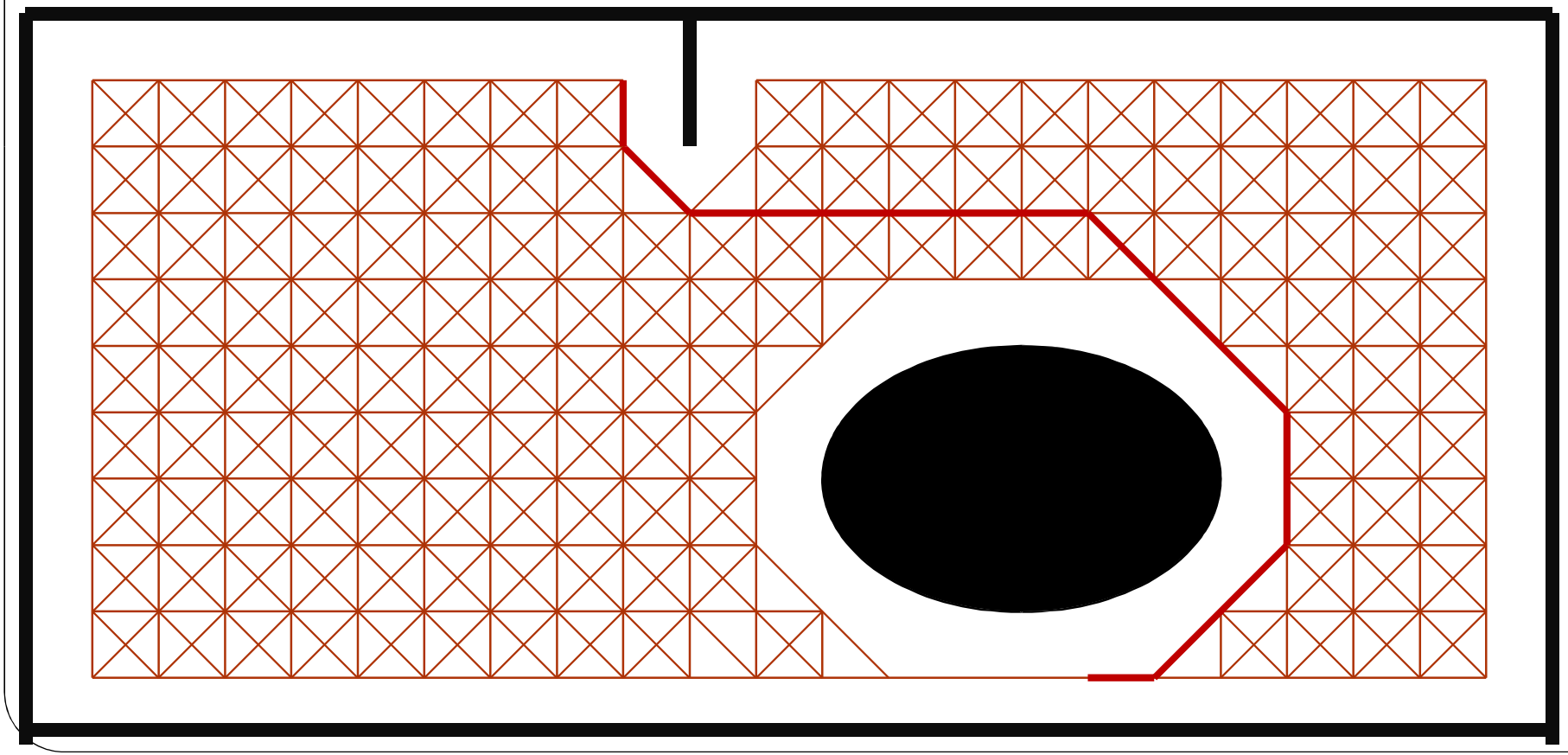
Relationship with Path Finding

- Classical path finding moves the character along a path in a graph

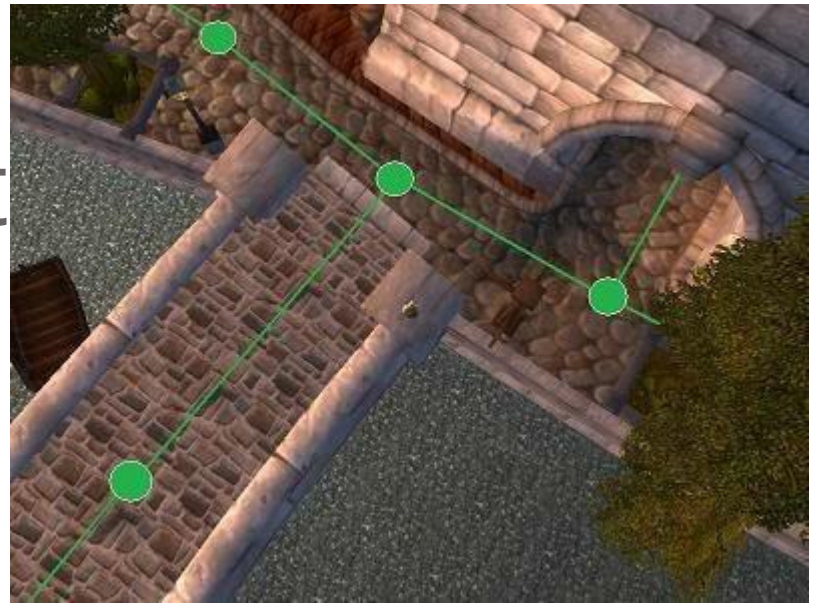


Relationship with Path Finding

- Classical path finding moves the character along a path in a graph (grid graph)



Stormwind



Stormwind
Way point graph

Convex polygons

Way point graph

- Waypoint graphs are relatively easy to implement.
- Waypoint graphs are easy to modify if the changes are known ahead of time. For instance, if a door in the world closes and is locked, it is easy for the developer to mark the edges in the graph that cross the opening of the door and block them when the door is shut.
- Waypoint graphs represent only a small fraction of the points found in a grid. This sparse representation of walkable space is both cheap to store and leads to inexpensive path planning requests.

- Path quality can suffer if there are not enough walkable edges in the graph, but too many walkable edges will impact storage and planning complexity.
- Waypoint graphs may require manual placement of nodes to get good path quality.
- Localization on waypoint graphs requires mapping between game space and the graph. If a character is knocked off of the graph, it may be unclear where the character should actually be within the waypoint graph.
- Because there is no explicit representation of the underlying state space, smoothing off the waypoint graph can result in characters getting stuck on physics or other objects.
- Dynamic changes are difficult when they aren't known ahead of time. If a character can create an unexpected hole in a wall, new connections on the waypoint graph are needed.

Navigation Mesh

- With the accurate representation of a polygon it is easier to correctly perform smoothing both before and during movement. This accuracy can also be used for tighter animation constraints.
- Path planning on navigation meshes is usually fast, as the representation of the world is fairly coarse. But, this does not impact path quality, as characters are free to walk at any angle.
- Navigation meshes are not as memory-intensive as grids as they can represent large spaces with just a few polygons.

- The time required to implement a navigation mesh is significant, although good open-source implementations are available [Mononen 11].
- Changes to navigation meshes can be difficult or expensive to implement, especially when contrasted with changes to grid worlds.

Way points

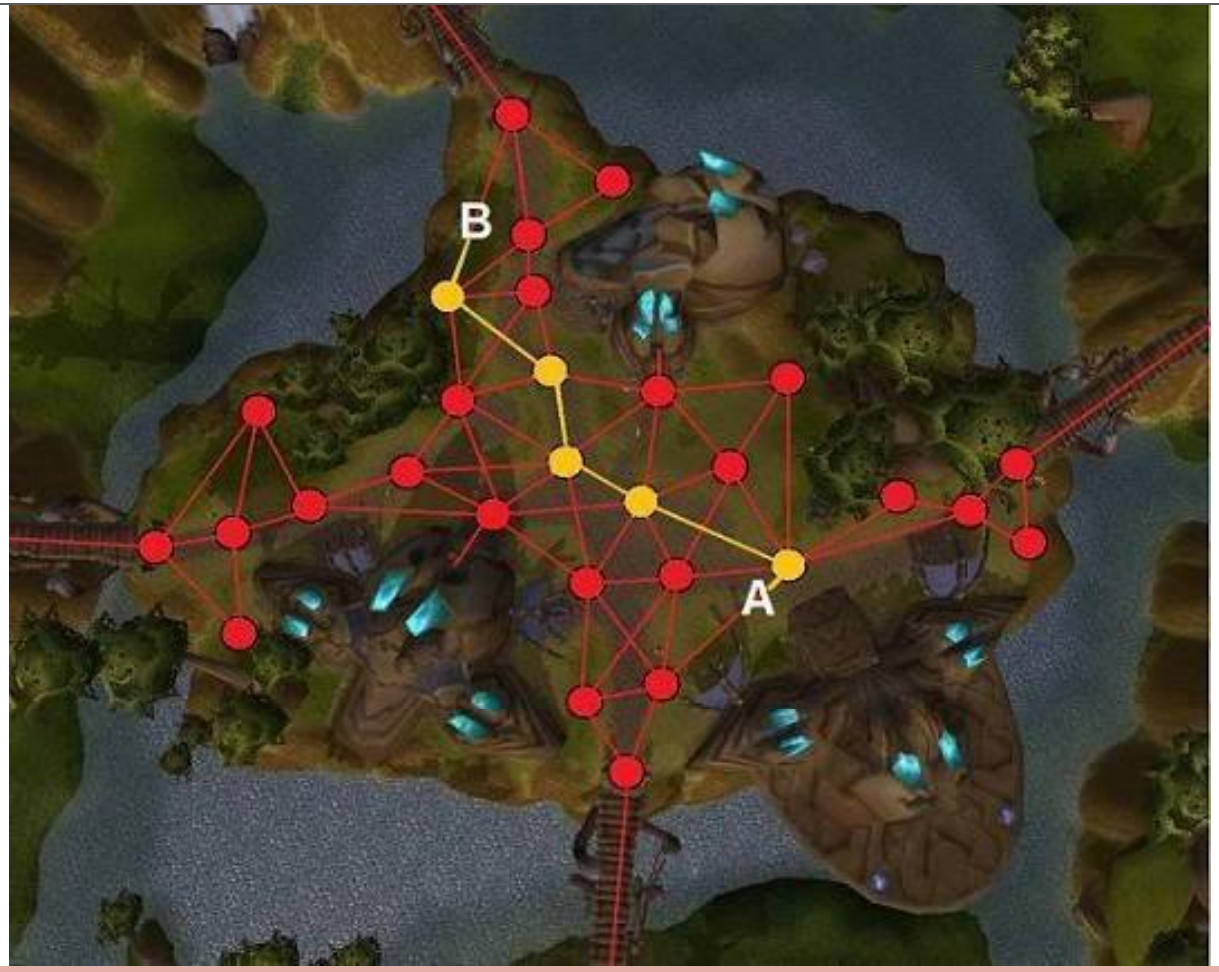
WoW: Halaa

Halaa (WoW)

Way points

Halaa

Convex polygons



Graph Smoothing

Please note that information is available on the graph


B
P3
P2
P!
A

free movement

종류 Seek , flee , wander , persue, jump

타겟을 쫓아감.

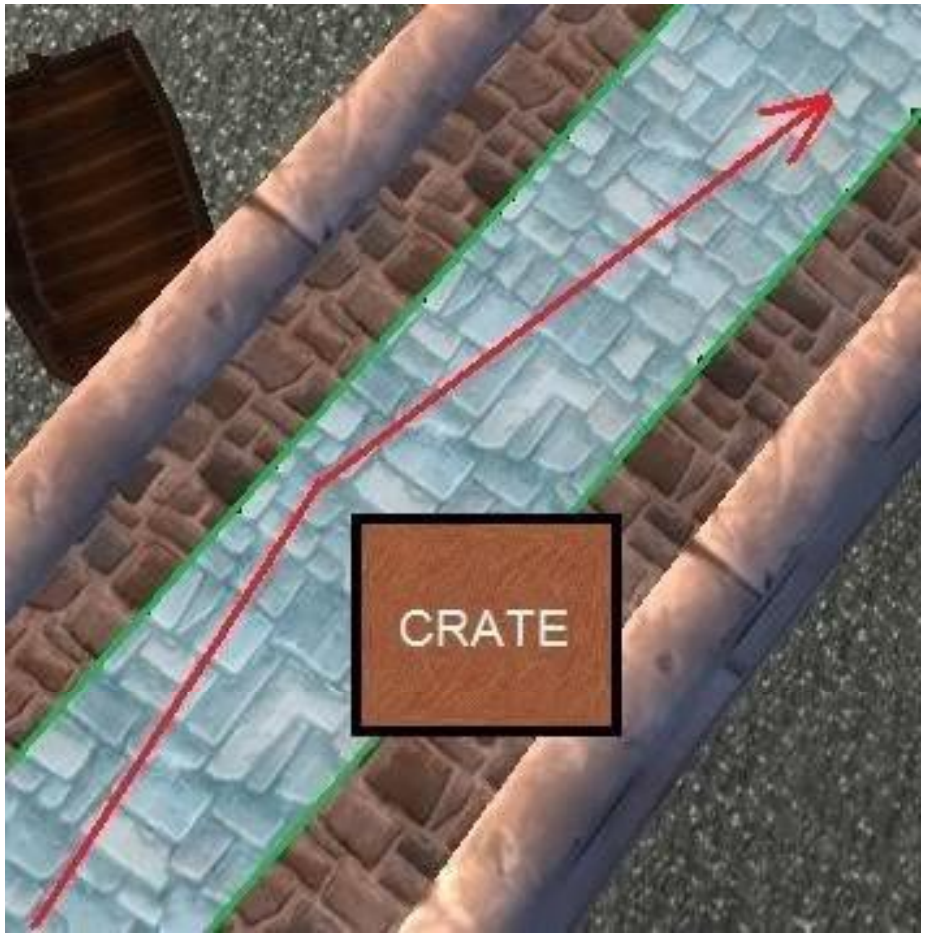
타겟에 움직였을 때 그 앞을 향해서 감

Single , crowd movement (flock movement)

formation (모바 군대처럼 진영을 갖춘 formation)

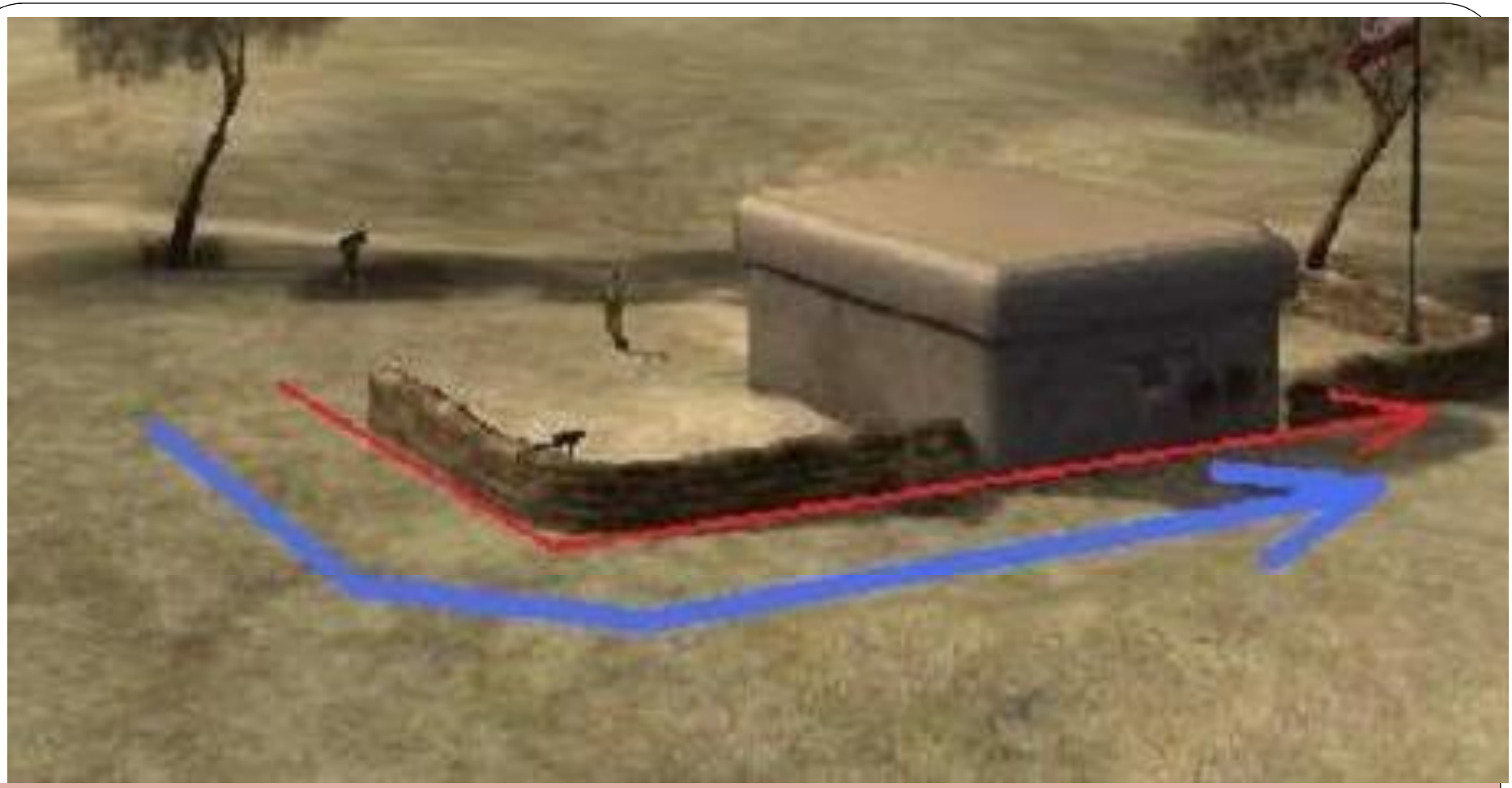
새떼처럼 같이 떼로 움직이는 것들

Way points, with obstacle

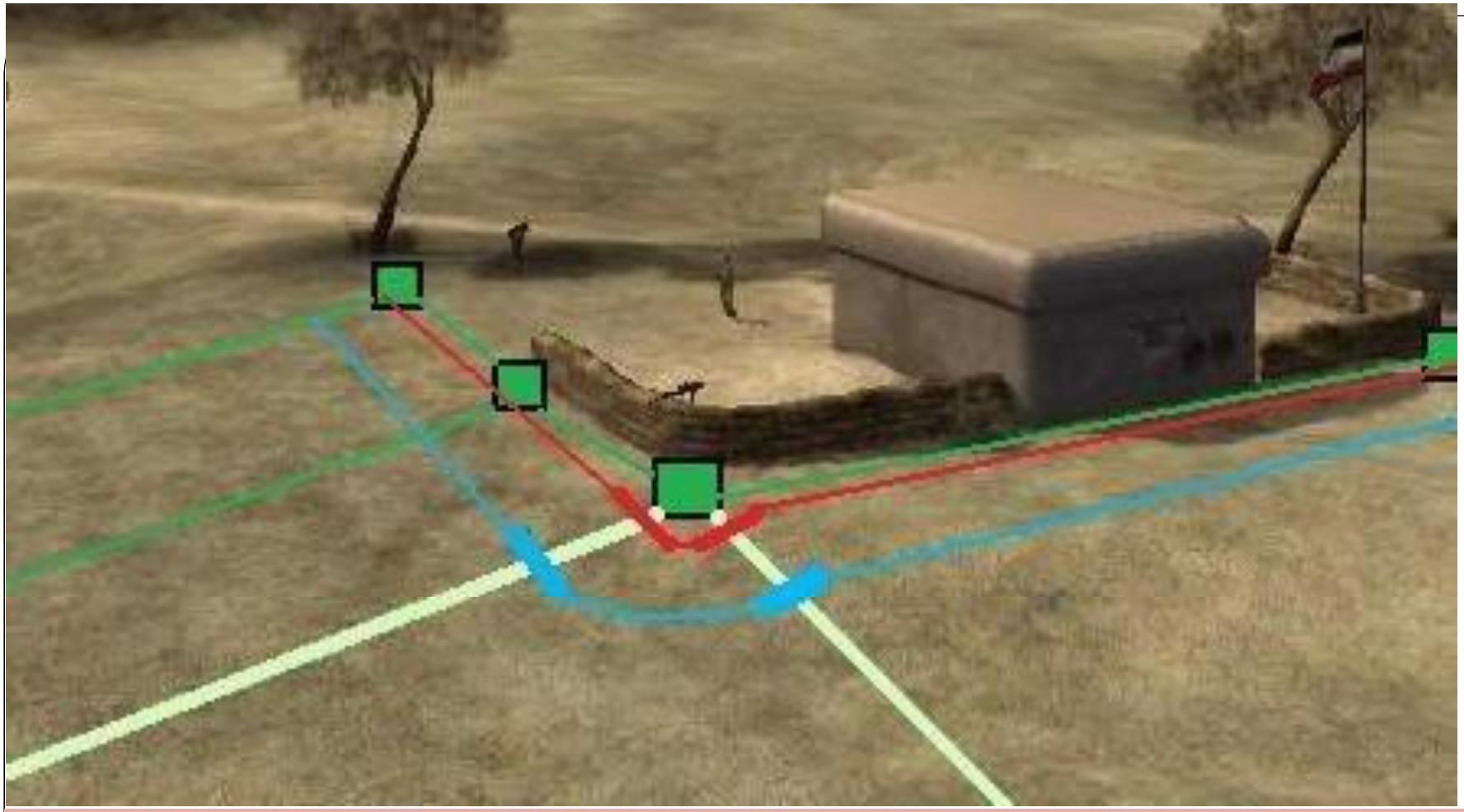


With convex polygons, one can put obstacles

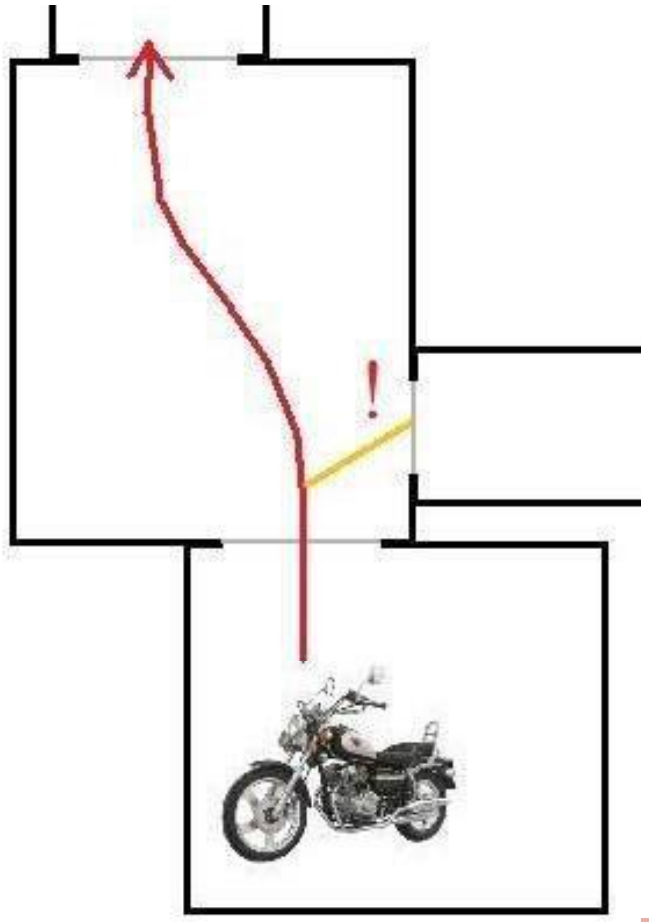
Desert bunker



A soldier can move close to the sandbags
A tank needs to take wider turns



Navigational mesh combined with movement



Movement radius is impossible to model with waypoint graph

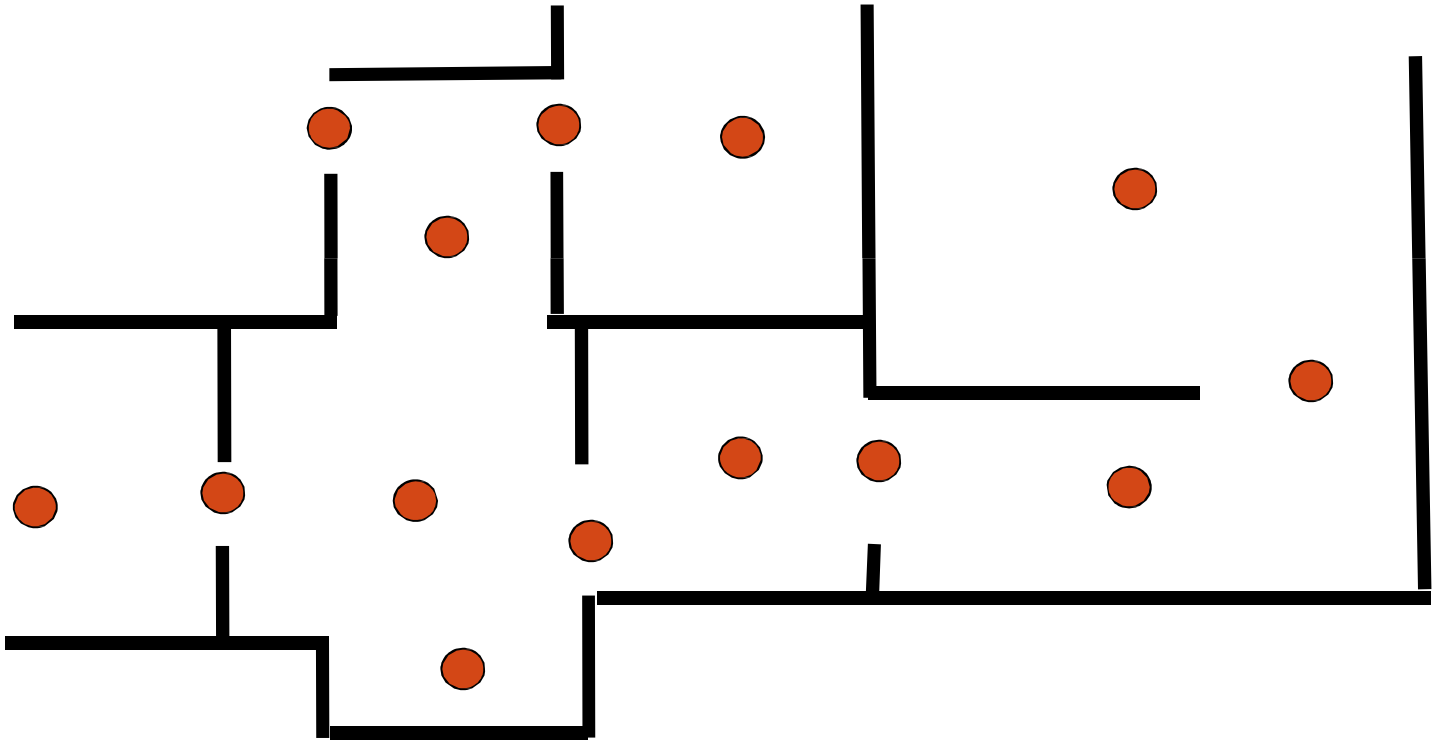
You can mark places

Use the navigational mesh for autonomous movement.

Use the points for scripted behavior

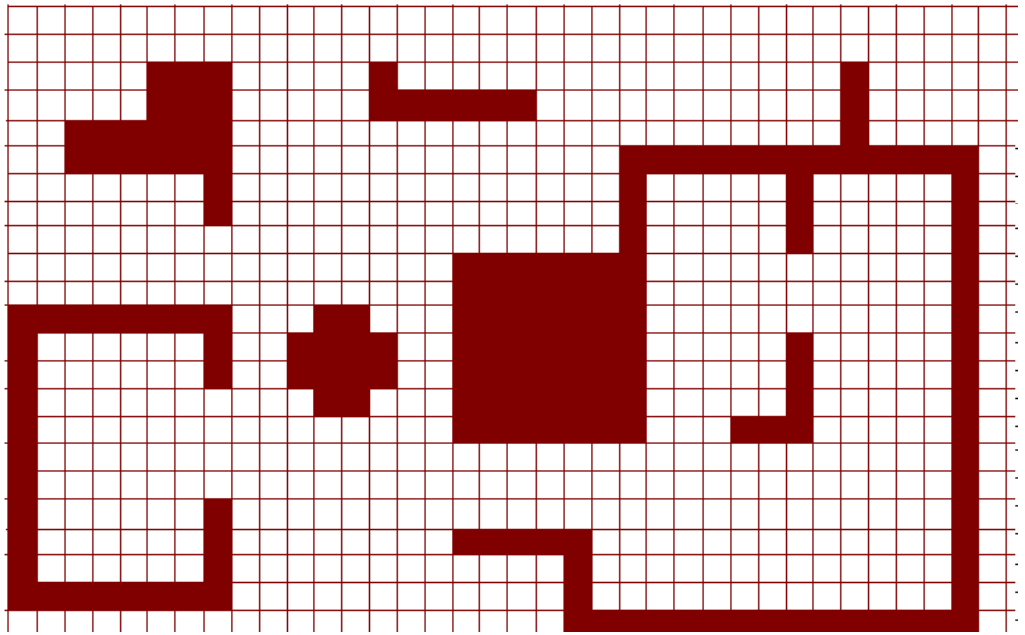
Waypoint graph generation

- Wide approximation



Waypoint graph generation

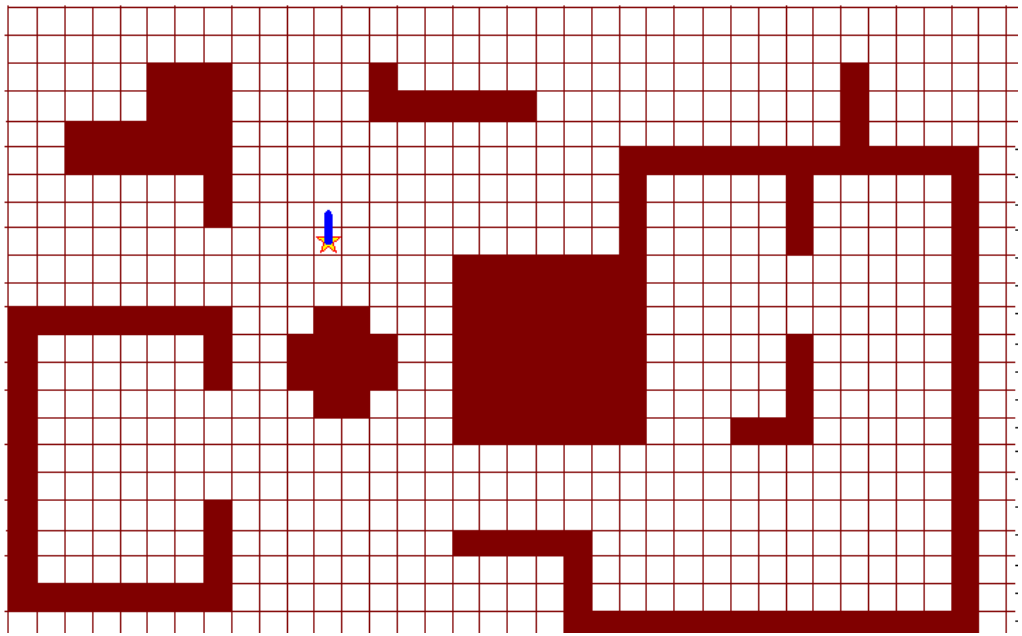
- Automatic generation



unfilled rectangle(cell)을 way point로 정의.
인접한 waypoint들 사이를 edge로 연결

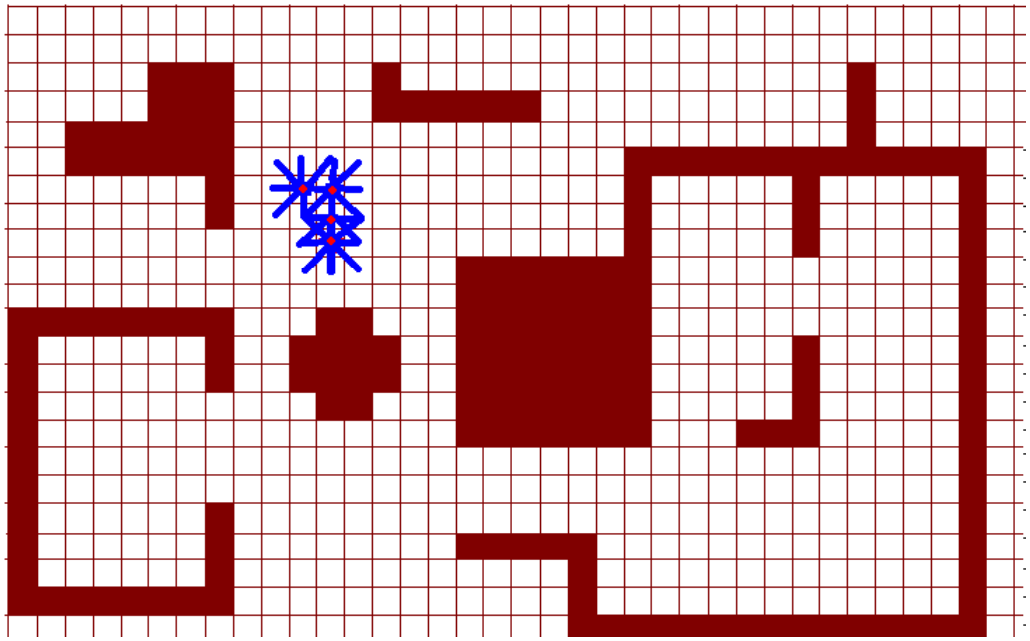
Waypoint graph generation

- Dense graph generation
  - Start at a point (unfilled rectangle)
  - Generate points in 4 (or eight directions) at unit distance



Waypoint graph generation

- Automatic generation

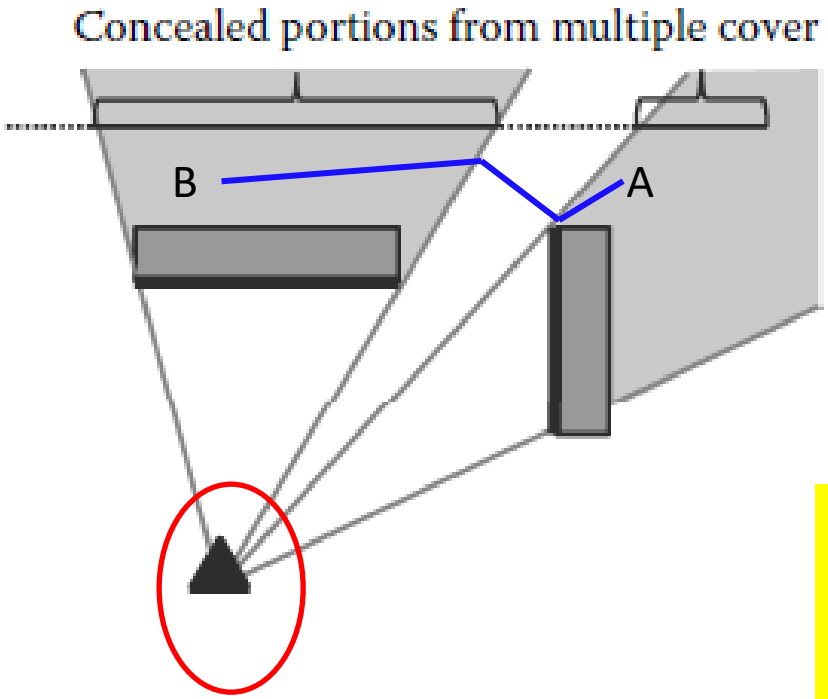


Challenges

- Obstacles that can be overcome
  - Fords
  - Jumps

- Tactical Pathfinding (ex. safe paths)

Tactical Pathfinding for a safe path

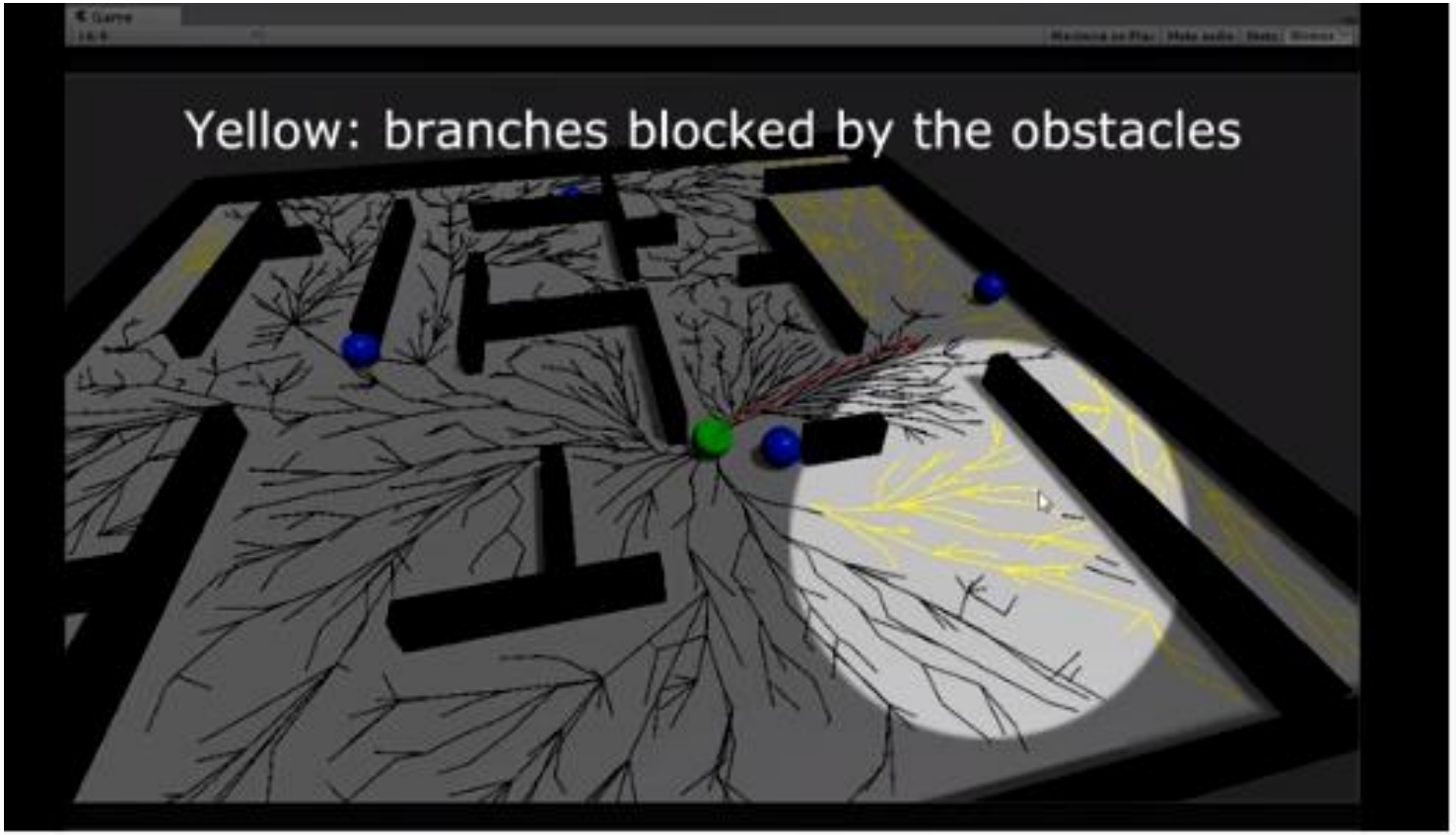


Enemy

총을 쏠 수 있음

회색 영역은 Enemy의 총격을 피할 수 있는 영역임.

- https://www.youtube.com/watch?v=QLNSkFnBYuM
  From a paper titled " **RT-RRT*: a real-time path planning algorithm based on RRT*** "



강의종료

다음 강의 주제는 ‘Free movement'

Free movement