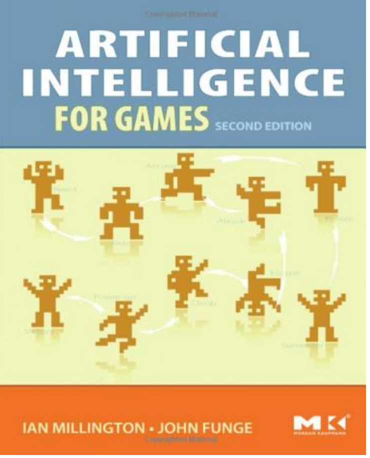
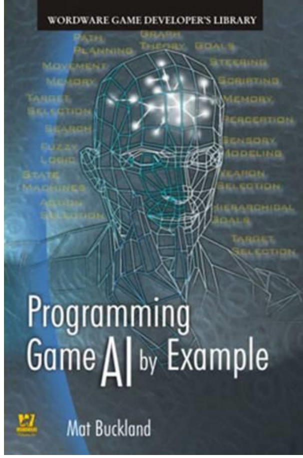
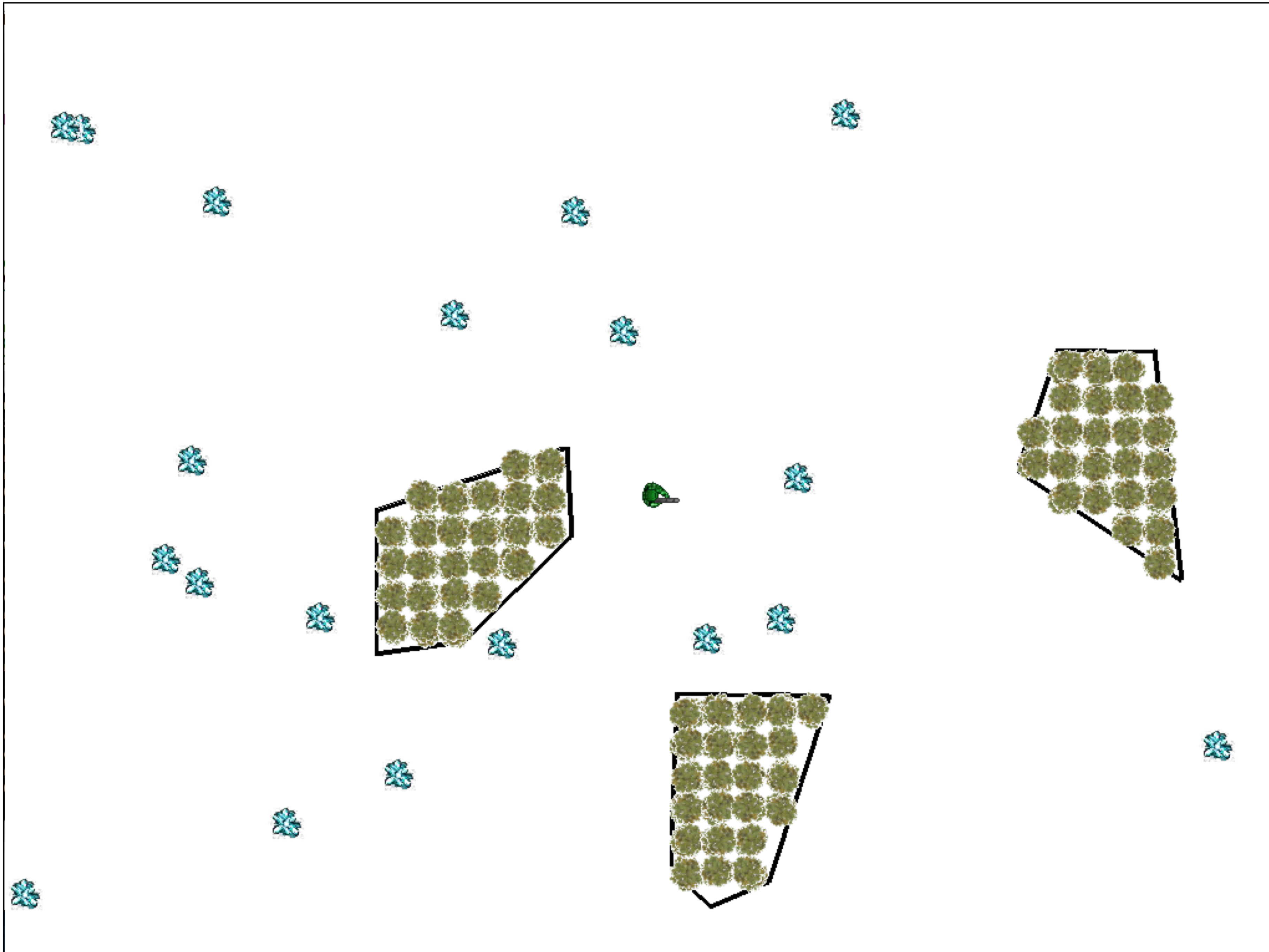


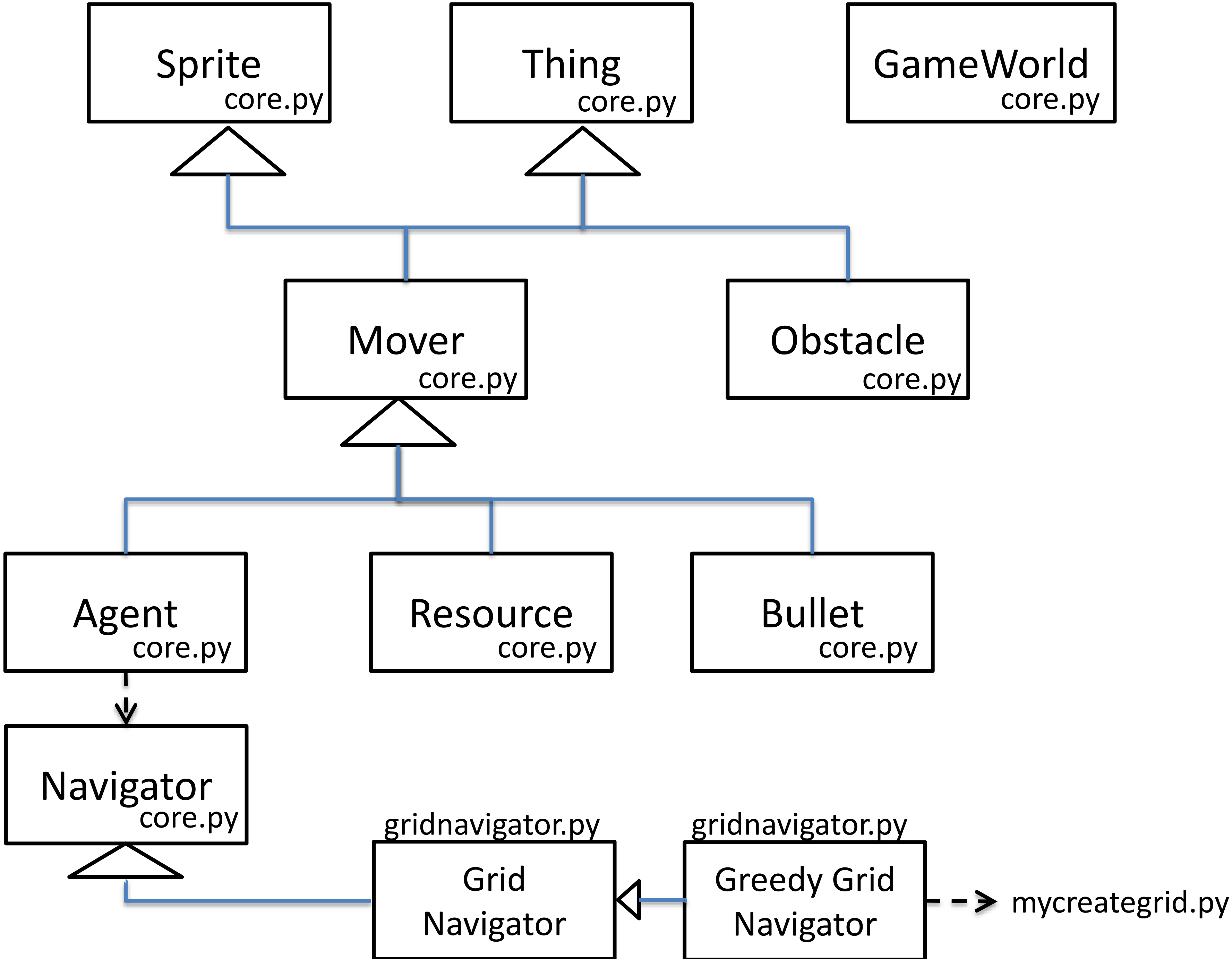


“the question of whether computers can think is like the question of whether submarines can swim” -- Dijkstra

Game AI: The set of algorithms, representations, tools, and tricks that support the creation and management of real-time digital experiences



http://game-ai.gatech.edu/sites/default/files/documents/documentation/object-hierarchy.png



PREVIOUSLY ON…

Class N-1

1. How would you describe AI (generally), to not us?
2. Game AI is really about
   - The I____ of I_____. Which is what?
   - Supporting the P_____ E______ which is all about… making the game more enjoyable
   - Doing all the things that a(nother) player or designer…
3. What are ways Game AI differs from Academic AI?
4. (academic) AI *in* games vs. AI *for* games. What's that?
5. What is the complexity fallacy?
6. The essence of a game is a g__l and set of r_?
7. What are three big components of game AI in-game?
8. What is a way game AI is used out-of-game?

Common (game) “AI” Tricks?

- Move before firing – no cheap shots
- Be visible
- Have horrible aim (being Rambo is fun)
- Miss the first time
- Warn the player
- Attack “kung fu” style (Fist of Fury; BL vs School)
- Tell the player what you are doing (especially companions)
- React to own mistakes
- Pull back at the last minute
- Intentional vulnerabilities or predictable patterns

Liden, “Artificial Stupidity: The Art of Intentional Mistakes”. *AI Game Programming Wisdom*.

Common Game AI techniques?

- Path planning, obstacle avoidance
- Decision making
  - Finite state machines
  - Trigger systems
  - Behavior trees
  - Robotics architectures
- Scripting, trigger systems
- Command hierarchies—strategic, tactical, individual combat
- Emergent behavior—flocking, crowds
- Formations
- Smart environments
- Terrain analysis—finding resource, ambush points
- Dynamic difficulty adjustment
- Drama management
- Procedural Content Generation

Intelligent vs. random

Graphs, Search, & Path Planning

2016-05-19

Graphs

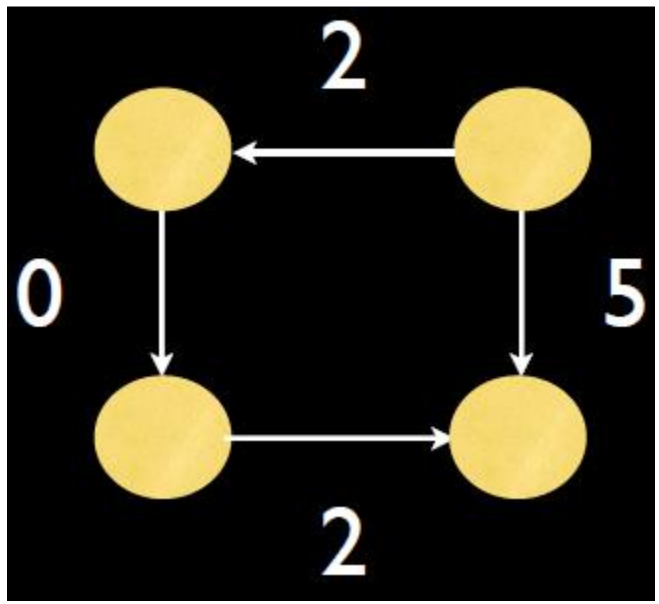
- What is a graph?
- What defines a graph?
- How can we represent them?
- How does representation effect search?

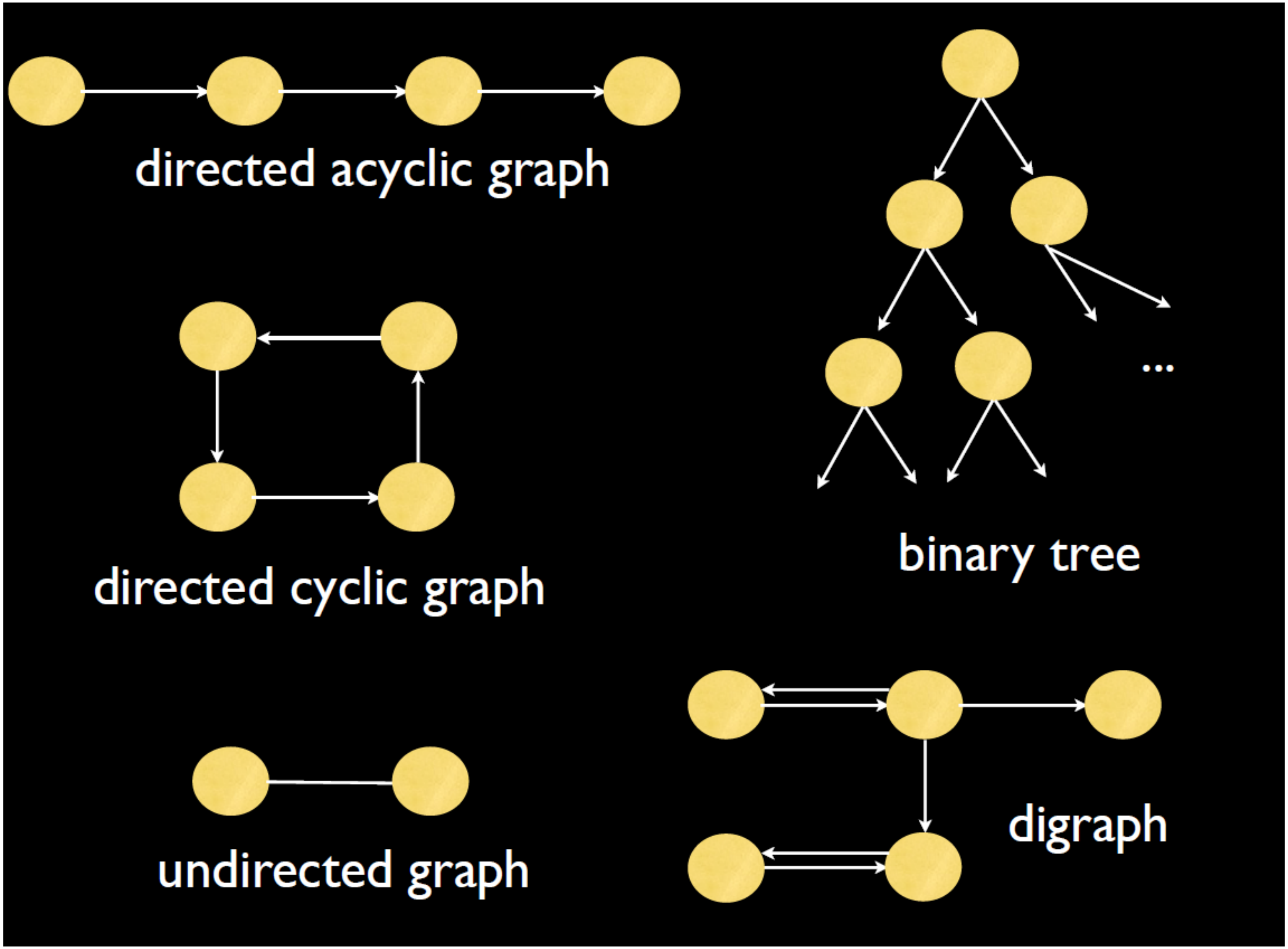
- **Applications to GAI?**

- See Buckland CH 5 for a refresher

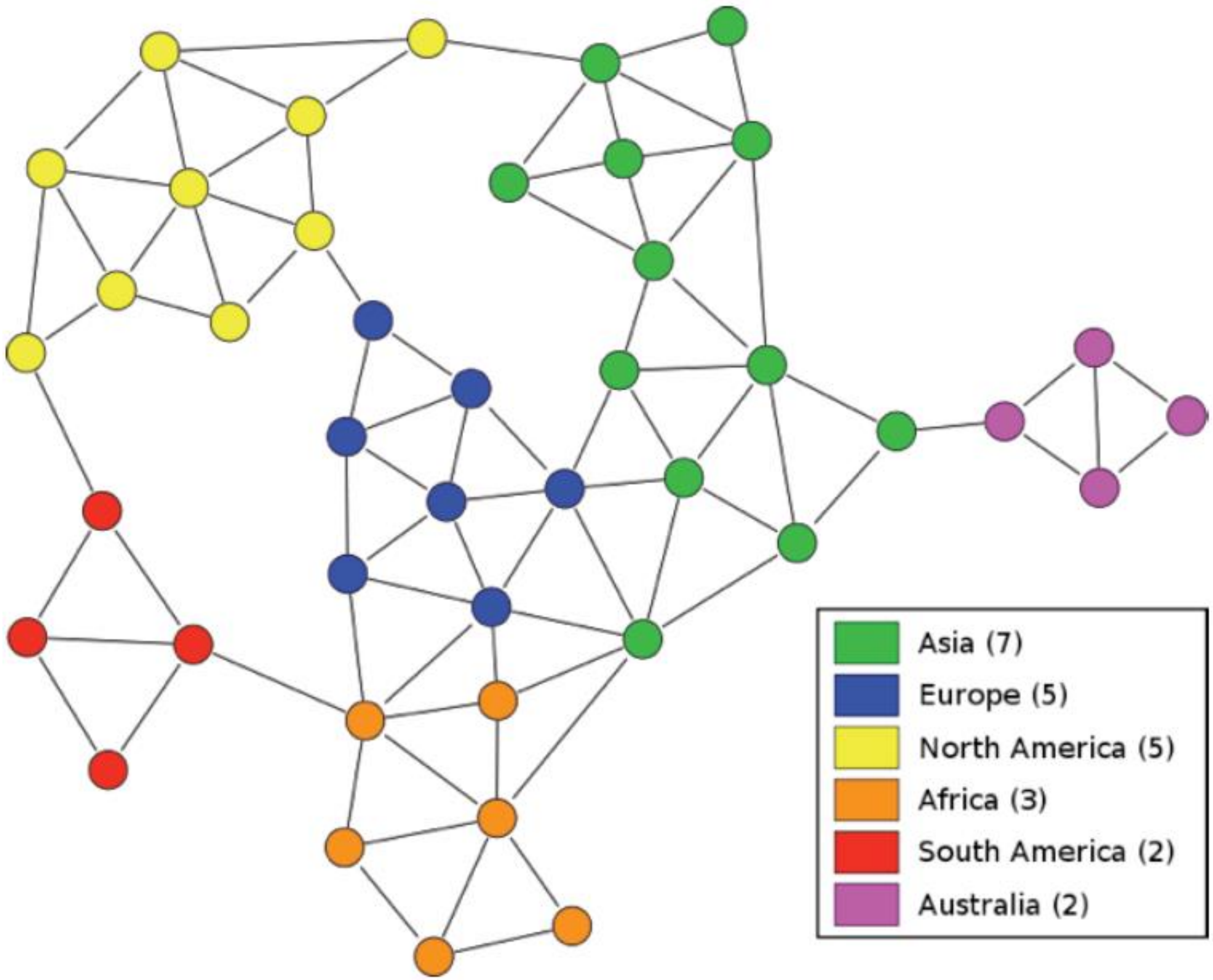
Graphs (2)

- G = {N,E}, N: Nodes, E: Edges (with cost)



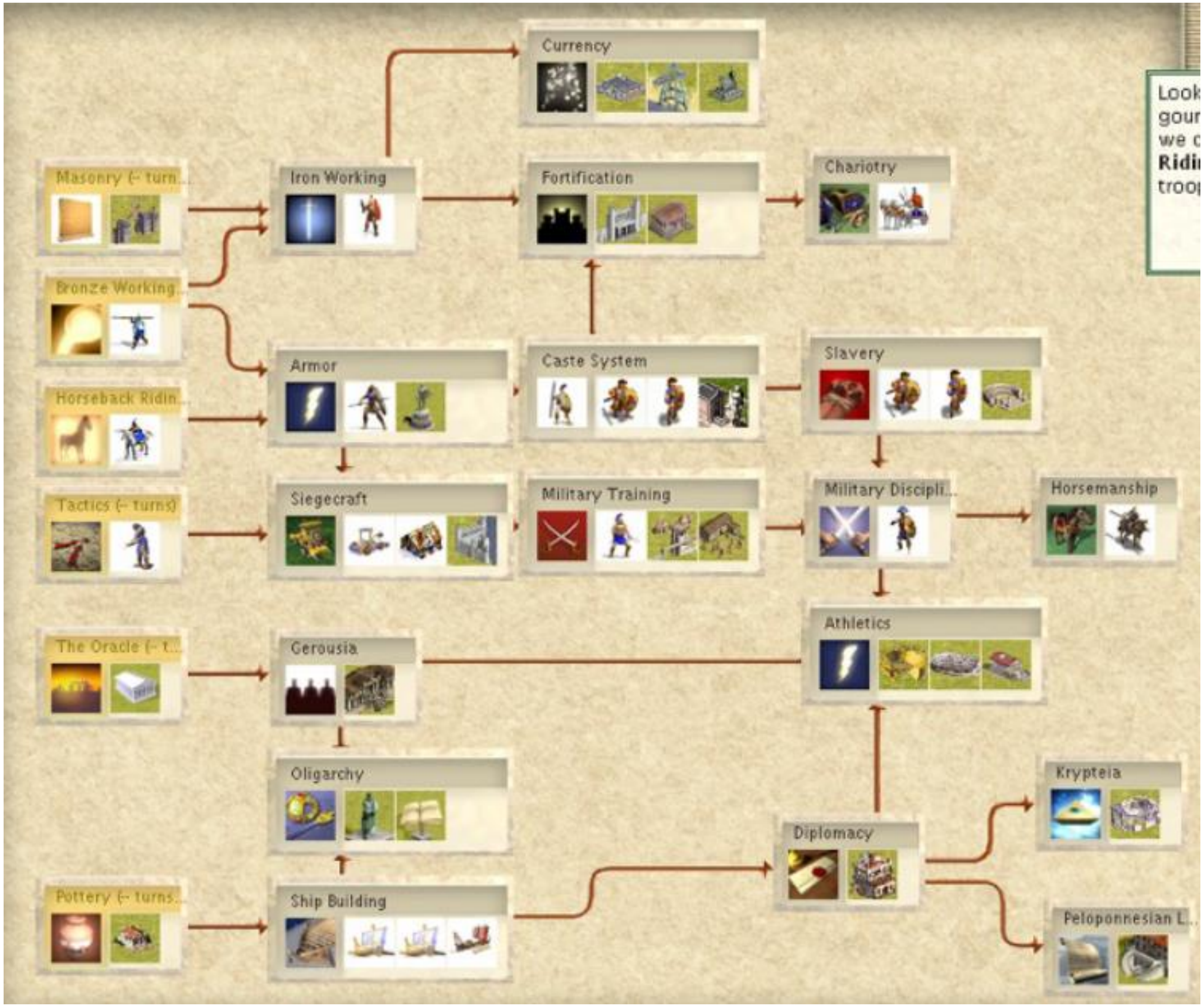

directed acyclic graph
binary tree
...
directed cyclic graph
digraph
undirected graph

Risk



Asia (7)
Europe (5)
North America (5)
Africa (3)
South America (2)
Australia (2)

RTS Dependency Tree



Graphs Killer App in GAI

- Navigation / Pathfinding
- Navgraph: abstraction of all locations and their connections
- Cost / weight can represent terrain features (water, mud, hill), stealth (sound to traverse), etc
- What to do when …
  - Map features move
  - Map is continuous, or 100K+ nodes?
  - 3D spaces?

Graph Search

- Uninformed (all nodes are same)
  - DFS (stack – lifo), BFS (queue – fifo)
  - Iterative-deepening (Depth-limited)
- Informed (pick order of node expansion)
  - Dijkstra – guarantee shortest path ($E\log_2 N$)
  - A* (IDA*)…. Dijkstra + heuristic
  - D*

http://en.wikipedia.org/wiki/A*_search_algorithm

Heuristics

- [dictionary]*"A rule of thumb, simplification, or educated guess that reduces or limits the search for solutions in domains that are difficult and poorly understood."*
- *h(n)* = estimated cost of cheapest path from n to goal (with goal == 0)

Path finding problem solved, right?

- Hall of shame:
  - Compilation
    - http://www.youtube.com/watch?v=lw9G-8gL5o0
  - Sim City (1, 2 … 5)
    - https://www.youtube.com/watch?v=MXMnZvBbaMM
  - Half-Life 2
    - http://www.youtube.com/watch?v=WzYEZVI46Uw
  - Fable III
  - DOTA 1+2
  - WoW

World Representation

- Ghallab, Nau, Traverso example
  - (from *Automated Planning* textbook)
  - A={pickup, putdown, load, unload, move}
  - S: 5 locations, 3 piles per loc, 3 cranes, 100 crates

  - State transition system has 10^{277} states (10^{190} x particles in universe)

Path finding models

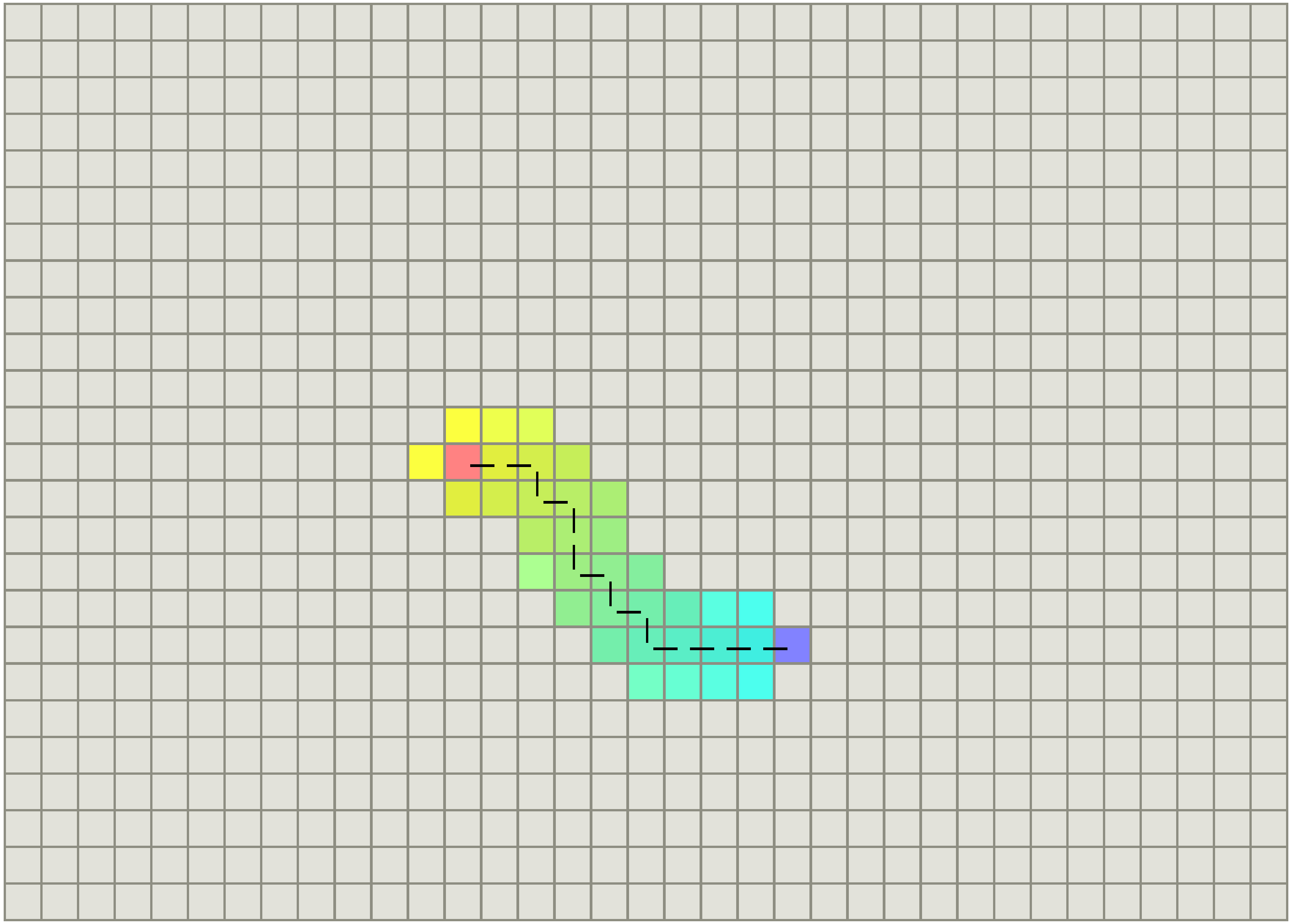
1. Tile-based graph – “grid navigation”
2. Path Networks / Points of Visibility NavGraph
3. Expanded Geometry
4. NavMesh

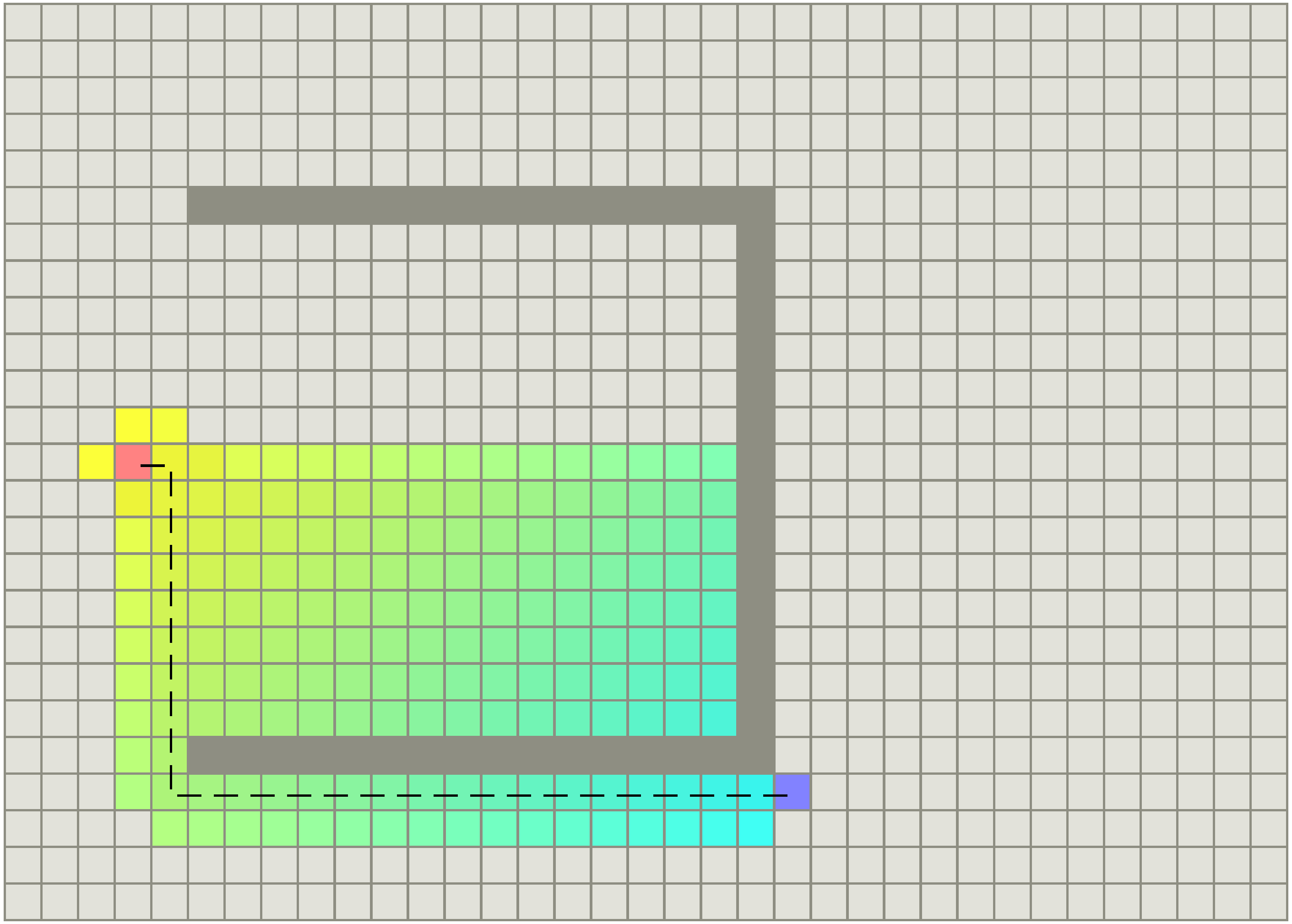
Model 1: Grid Navigation

- 2D tile representation mapped to floor/level
  - Squares, hex; 8 or 6 neighbors / connectivity
- Mainly RTS games
- One entity/unit per cell
- Each cell can be assigned terrain type
- Bit mask for non-traversable areas
- Navigation: A*, Dijkstra

Path Planner

- Initial state (cell), Goal state (cell)
- Each cell is a state agent can occupy
- Sort successors, try one at a time (backtrack)
- Heuristic: Manhattan or straight-line distance
- Each successor stores who generated it





Grid navigation: pros

- Discrete space is simple
- Can be generated algorithmically at runtime
- Good for large number of units
- A* works really well on grids (uniform action cost, not many tricky spots)

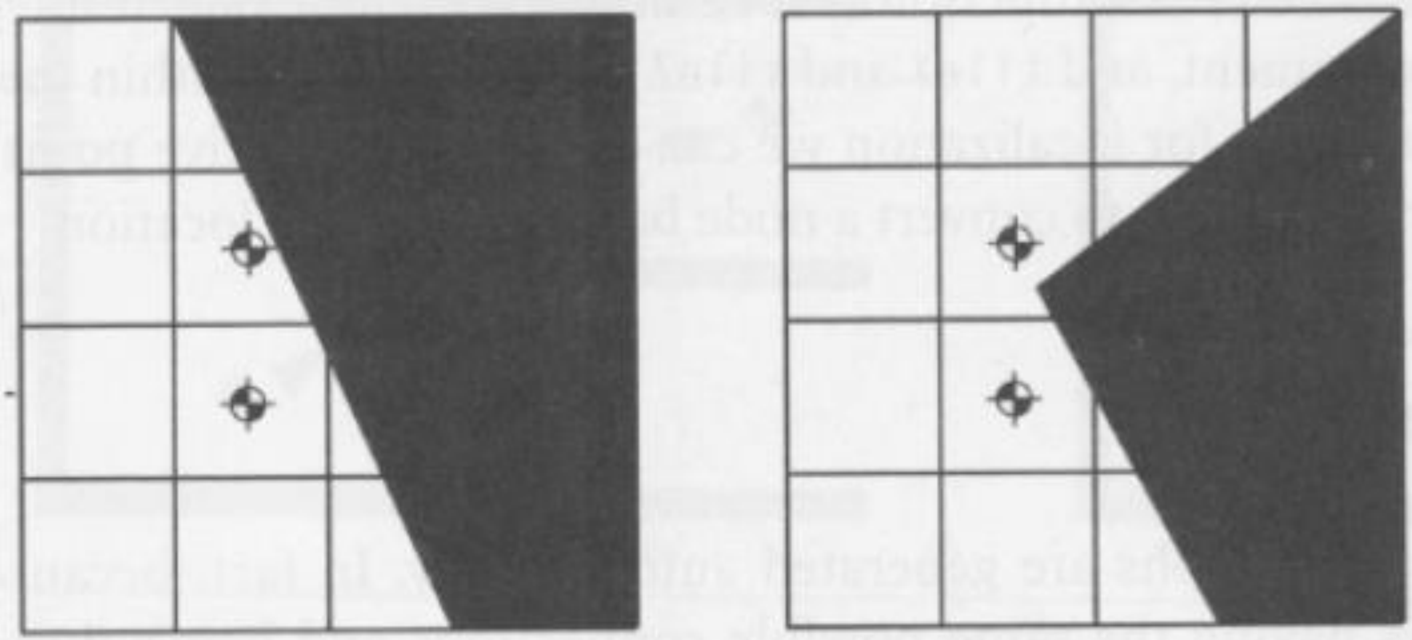
Grid navigation: cons

- Discretization “wastes” space
- Agent movement is jagged/awkward/blocky, though can be smoothed
- Some genres need continuous spaces
- Partial-blocking hurts validity
- Search must visit a lot of nodes (cells)
- Search spaces can quickly become huge
  - E.g. 100x10 map == 100k nodes and ~78k edges

New Problems

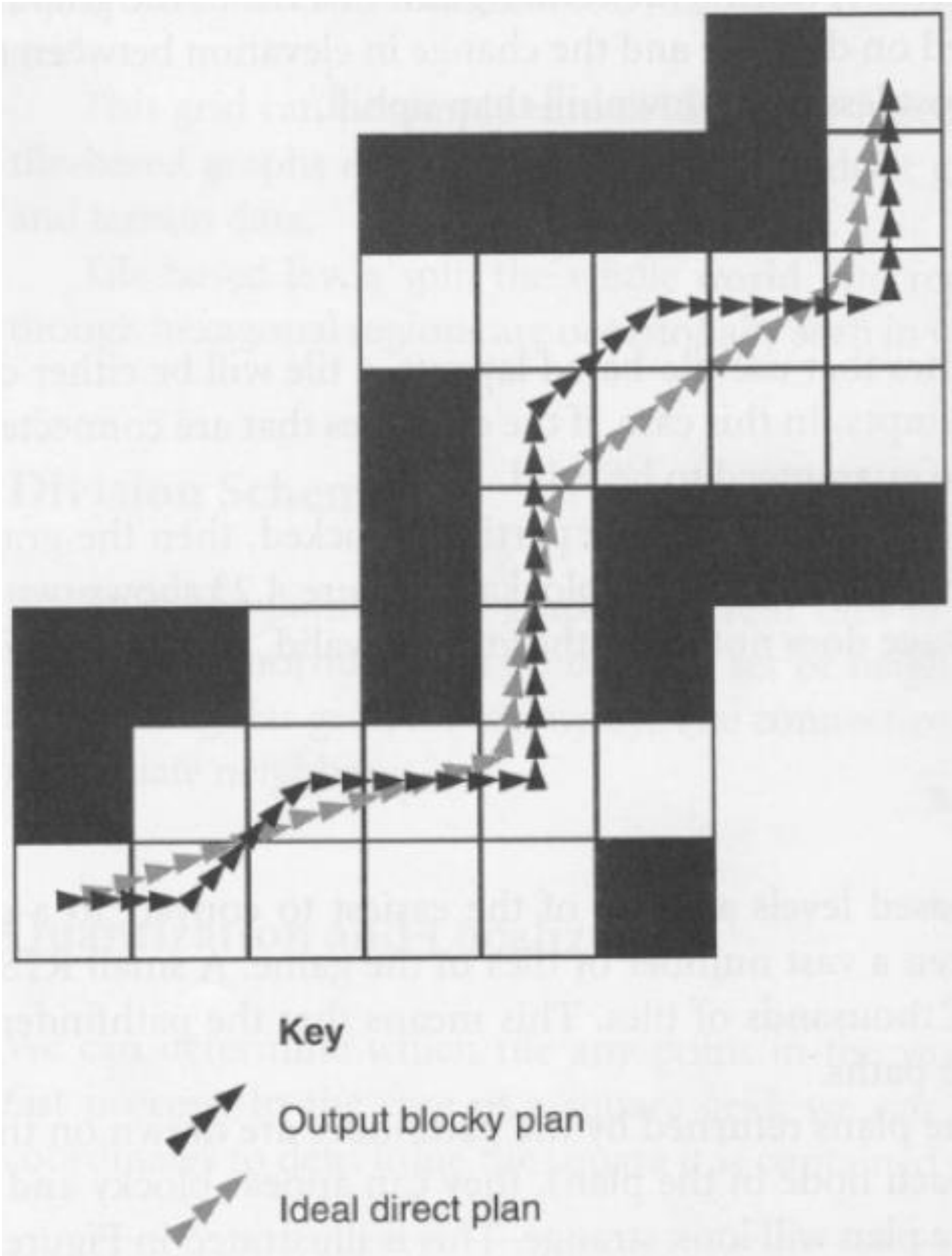
- Generation
- Validity
- Quantization
  - Converting an in-game position (for yourself or an object) into a graph node
- Localization
  - Convert nodes back into game world locations (for interaction and movement)

Validity



Valid partial blockage

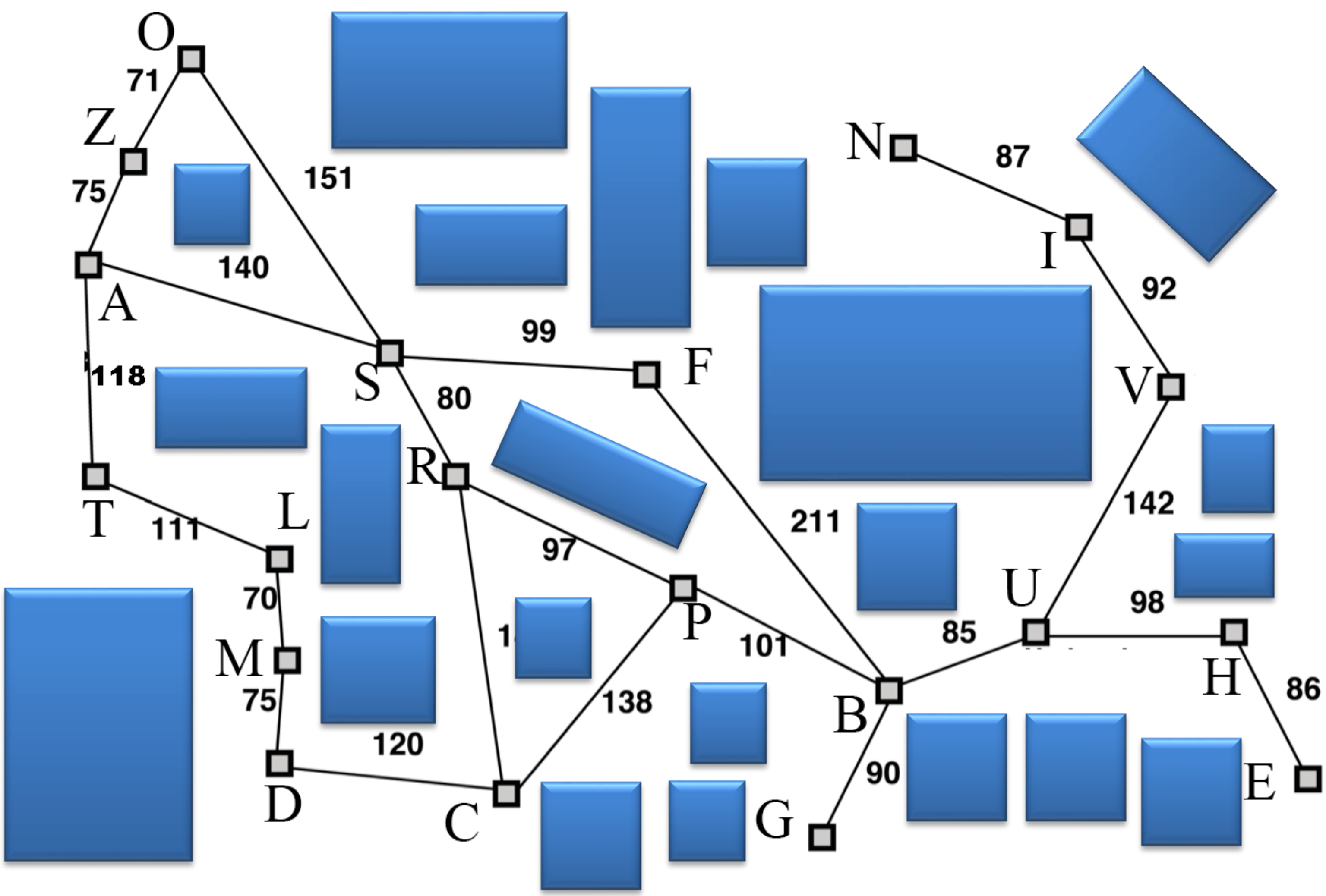
Invalid partial blockage



- String pulling
- Splines
- Hierarchical A*

M2: Path Networks

- POV: Points of visibility NavGraph (see B CH 8)
- Discretization of space into sparse network of nodes
- Two-tiered navigation system
  - Local, continuous
  - Remote
- Connects points *visible to each other* in all important areas of map
- Usually hand-tailored (can use flood-fill)



O
71
Z
75
151
A
140
118
S
99
F
80
R
T
111
L
97
P
70
M
75
120
D
C
138
101
B
90
G
211
85
U
98
H
86
E
N
87
I
92
V
142

Waypoints