

MAvis Assignment 1

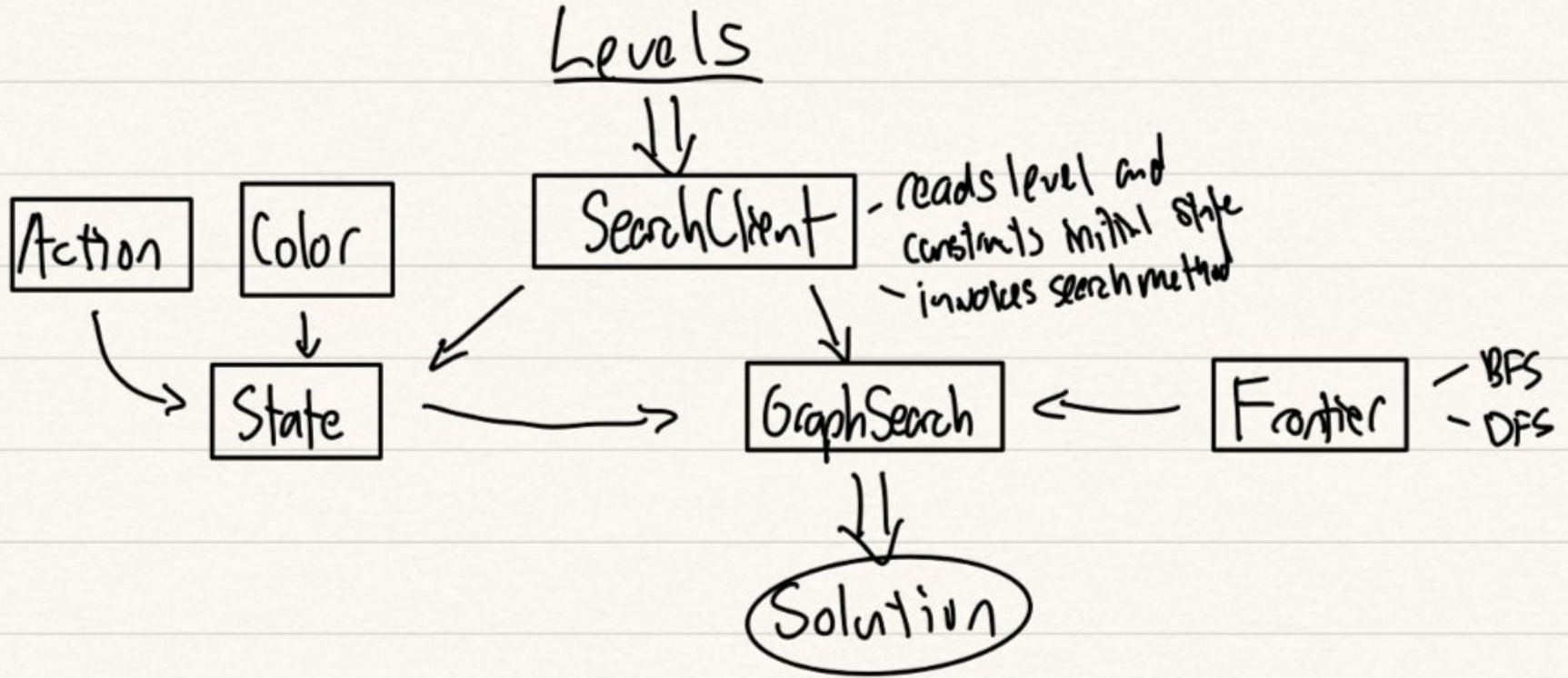
Group JKK

Group Declaration - JKK

- Kevin - Exercise 3 coding, Designing a BFS-Friendly level
- Kaiya - Exercise 2 coding, BFS analysis slides
- Justin - Exercise 2 coding, Exercise 1 slides, Recording benchmarks

We all discussed the exercises together as a group and contributed an equal amount to the final presentation.

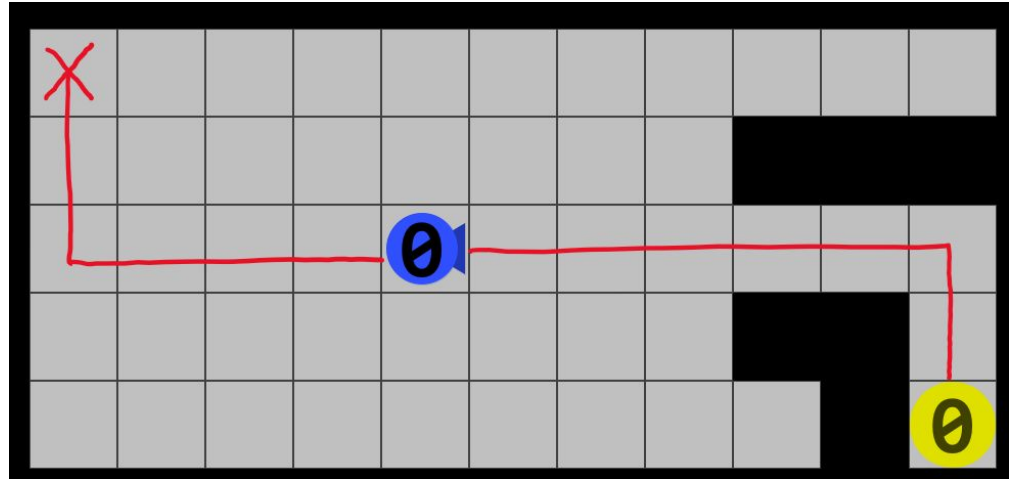
Code Structure Overview



Manual vs. GraphSearch Solution to MAPF00

Manual Solution:

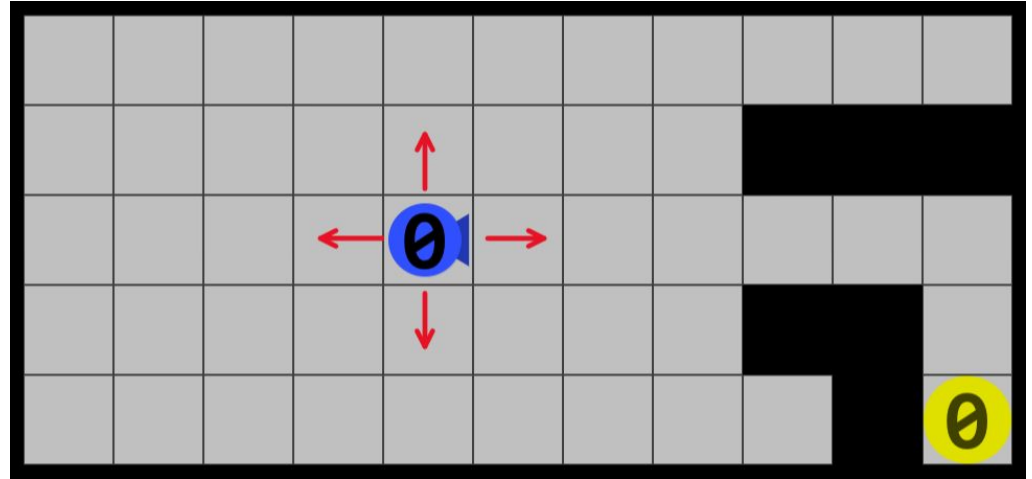
- 14 moves
- Hard to tell if we found the optimal solution in complex levels
- Humans visualize a path from start to goal



Manual vs. GraphSearch Solution to MAPF00

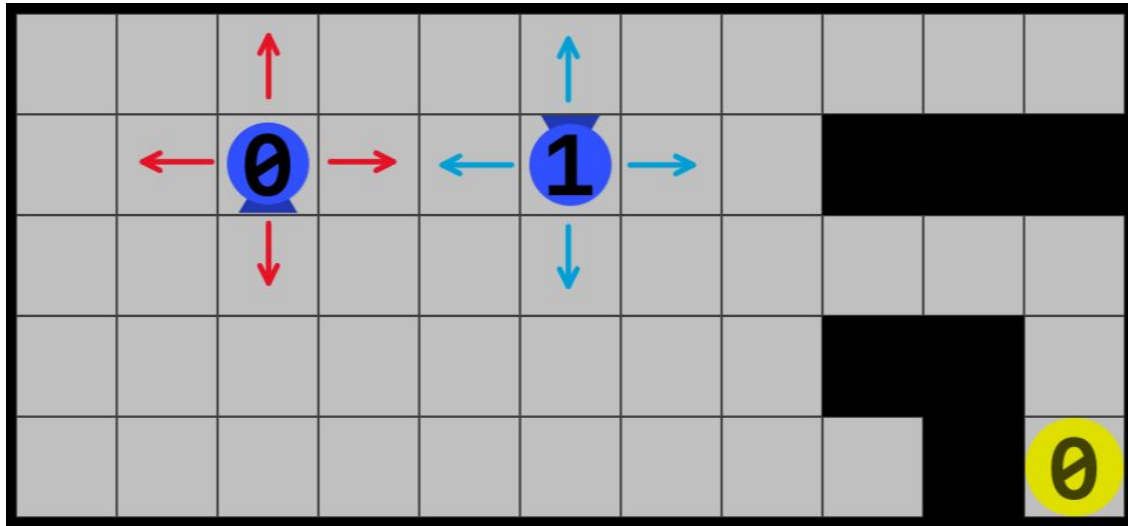
GraphSearch Solution (BFS):

- 14 moves
- Always finds optimal solution
- Algorithm searches through possible moves
 - Unaware of shape of the level



Effect of Adding More Agents

- State space increases (almost) exponentially
- Branching factor increases from 4 to 16 with second agent

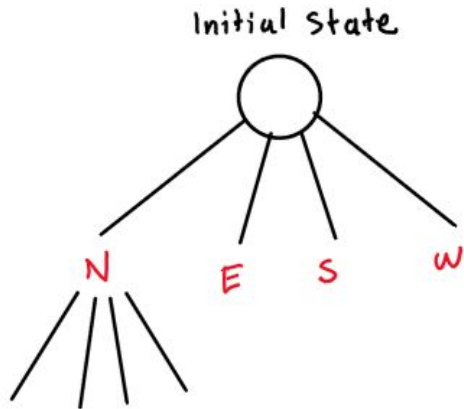


Effect on Branching Factor

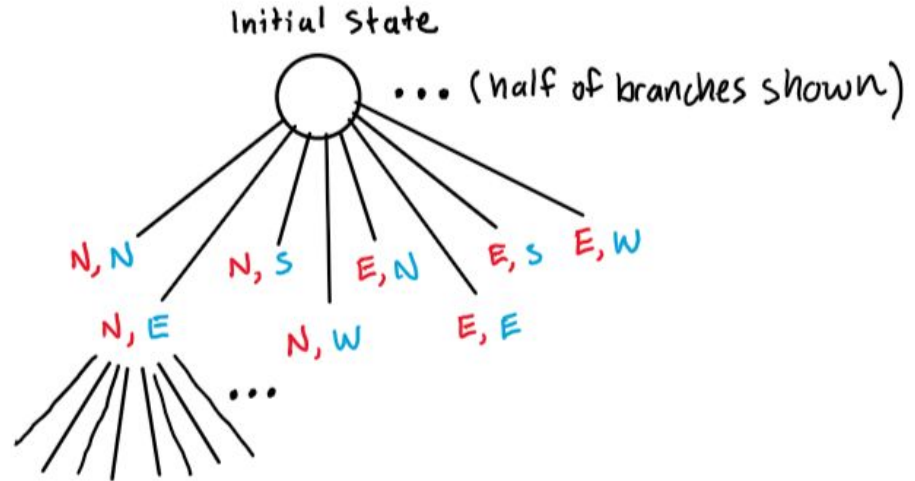
1 Agent: maximum of 4 possible states after the next move

2 Agents: maximum of 16 possible states

One Agent



Two Agents



Comparing Levels Solved with BFS

State Space Formula:

$$state\ space = n * (n - 1) * \dots * (n - (a - 1))$$

- n: number of cells
- a: number of agents

Example: 3 agents in MAPF02:

Theoretical state space:

$$49 * 48 * 47 = 110,544$$

Actual states generated: 110,445

Example: 5 agents in MAPFreorder2:

Theoretical state space:

$$15 * 14 * 13 * 12 * 11 * 10 = 3,603,600$$

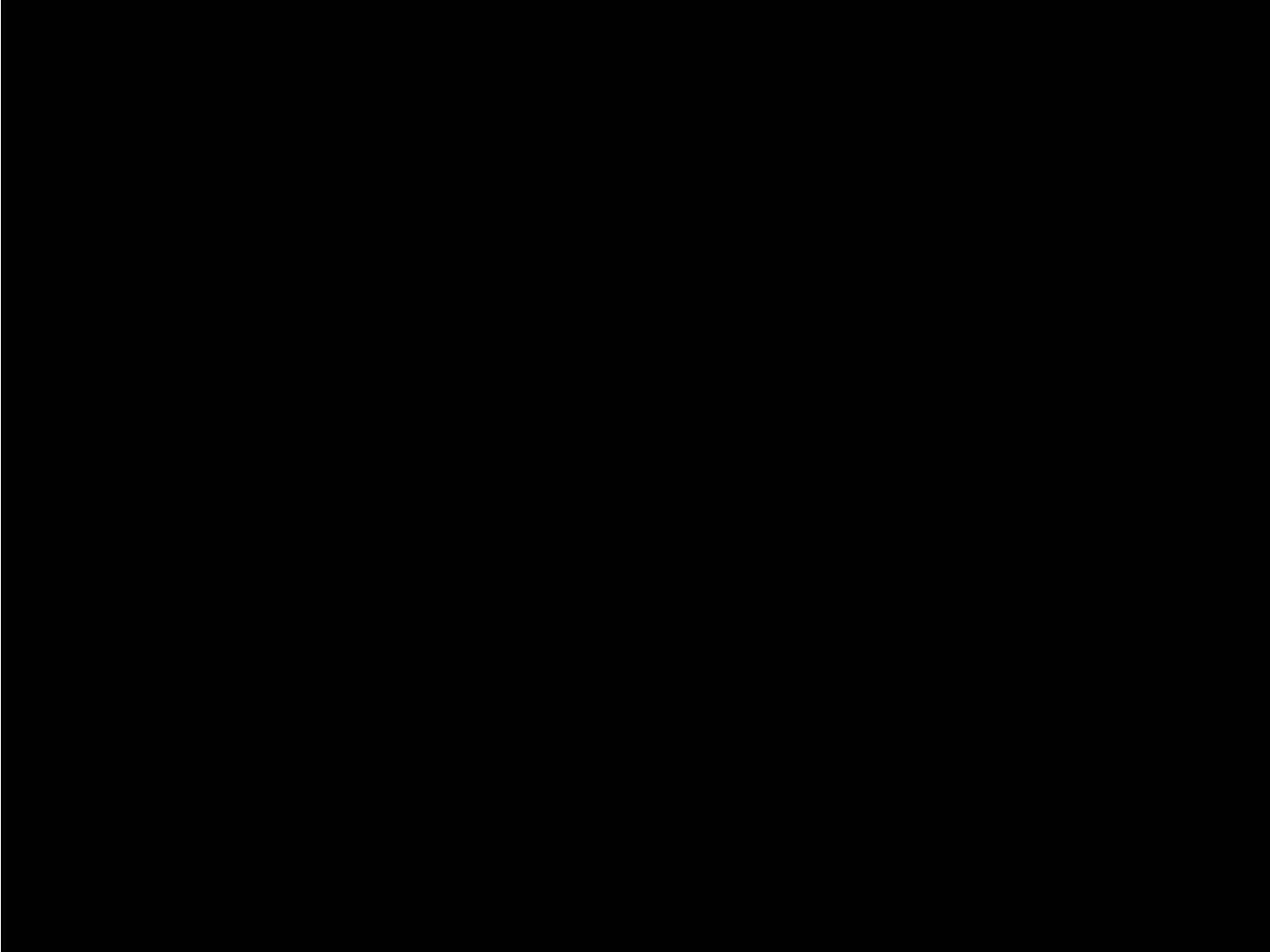
Actual states generated: 3,603,599

Comparing DFS and BFS

Implementation: Our DFS implementation uses a stack as the primary data structure while the BFS implementation uses a queue

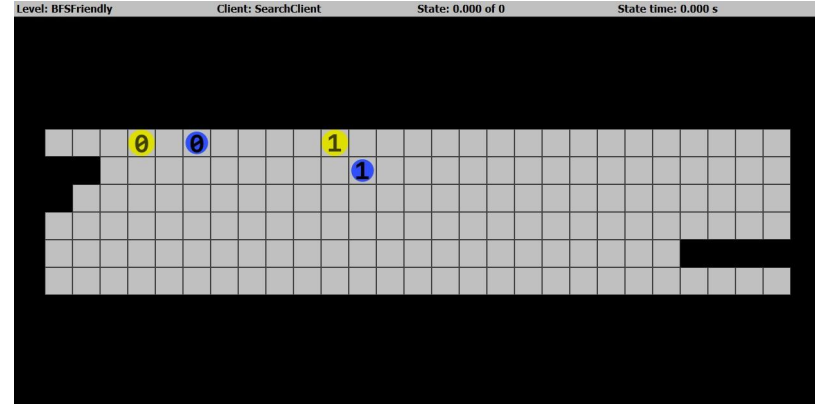
Notable levels: MAPF03, MAPFslidingpuzzle

| Level | Strategy | States Generated | Time/s | Solution length |
|-------------------|----------|------------------|----------|-----------------|
| MAPF00 | BFS | 48 | 0.031 | 14 |
| MAPF00 | DFS | 41 | 0.027 | 18 |
| MAPF01 | BFS | 2,350 | 0.146 | 14 |
| MAPF01 | DFS | 1,270 | 0.126 | 147 |
| MAPF02 | BFS | 110,445 | 5.71 | 14 |
| MAPF02 | DFS | 8,218 | 0.687 | 207 |
| MAPF02C | BFS | 110,540 | 5.7223 | 14 |
| MAPF02C | DFS | 86,870 | 165.612 | 3538 |
| MAPF03 | BFS | 5,063,873 | 2279.924 | 14 |
| MAPF03 | DFS | 128,511 | 277.022 | 608 |
| MAPF03C | BFS | 5,084,159 | 2204.779 | 14 |
| MAPF03C | DFS | | N/A | |
| MAPFslidingpuzzle | BFS | 181,289 | 1.5 | 28 |
| MAPFslidingpuzzle | DFS | 163,454 | 180.507 | 57558 |
| MAPFreorder2 | BFS | 3,603,599 | 172.078 | 38 |
| MAPFreorder2 | DFS | | N/A | |



Designing a BFS-Friendly Level

- Our core assumption in designing a BFS Friendly level was ensuring that the goal states for the agents were relatively close to the initial states.
- This is due to BFS agents finding solutions level-by-level (i.e. expanding their search “circle” by one cell on each pass)
- A DFS search strategy, by comparison, would struggle on this level due to it not being able to verify all shorter solutions before moving on to longer solutions



| | | States Generated | Time/s | Solution length |
|-------------|-----|------------------|--------|-----------------|
| BFSfriendly | BFS | 315 | 0.033 | 2 |
| BFSfriendly | DFS | 23,849 | 71.05 | 990 |