

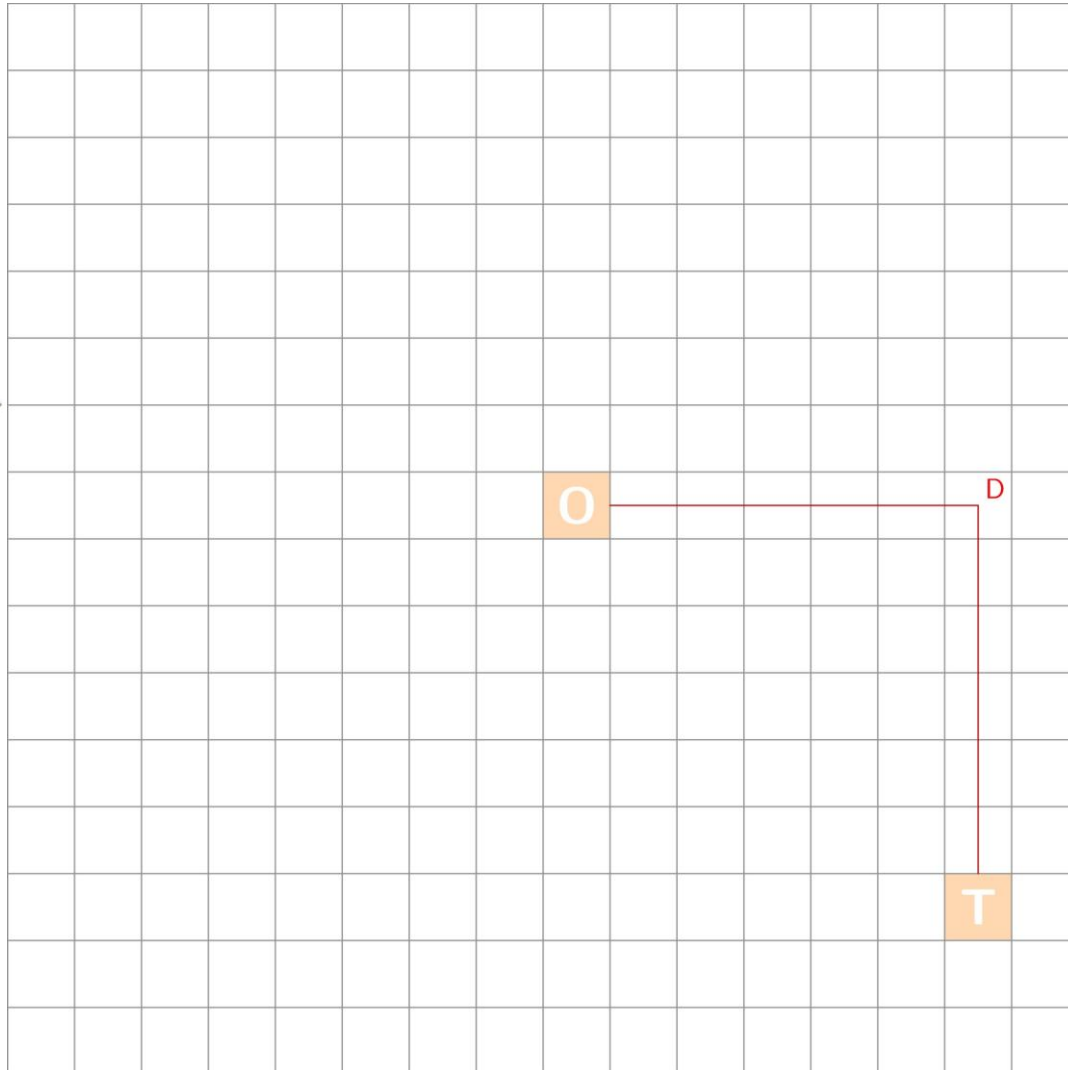
Fault-Tolerant ANTS



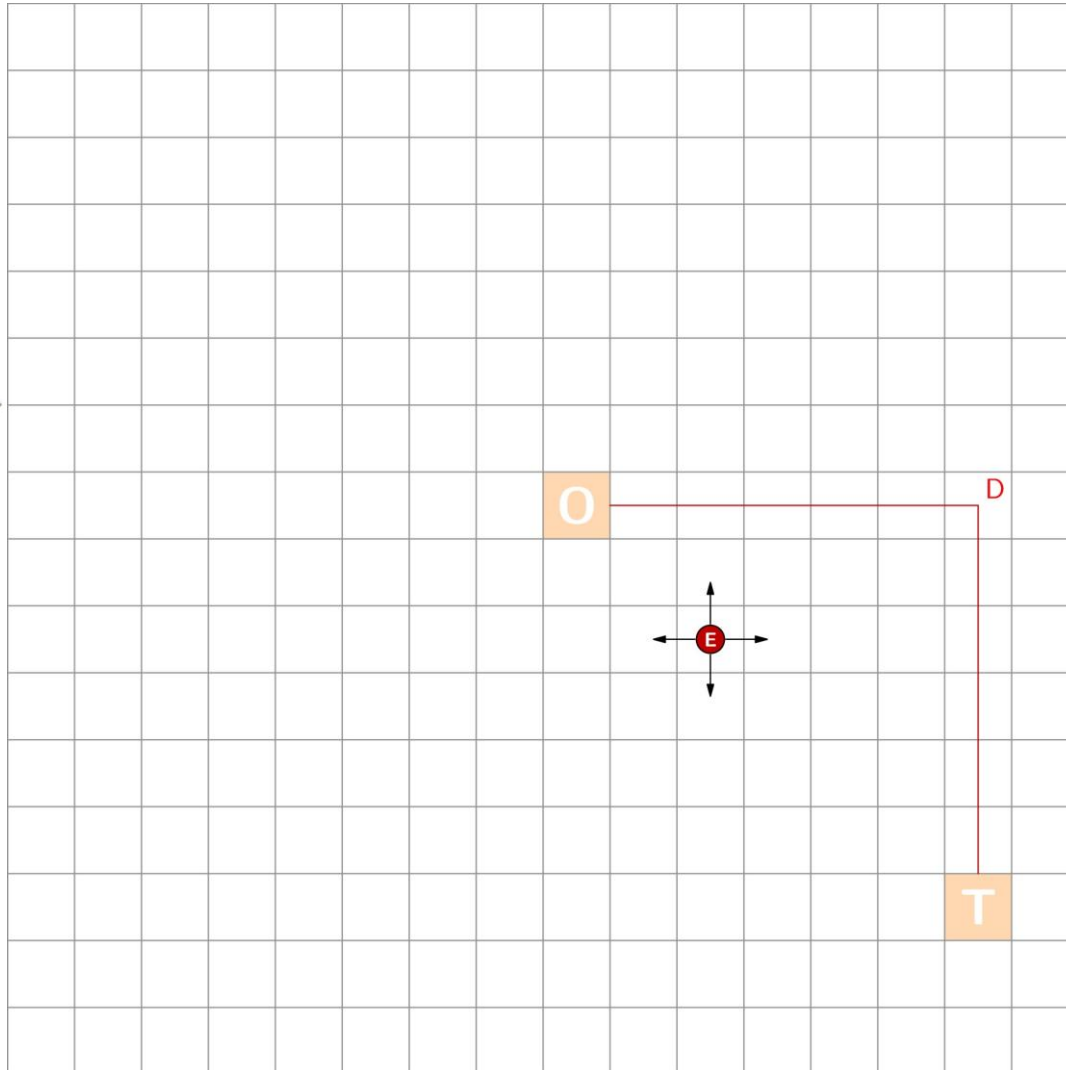
Jara Uitto



Model



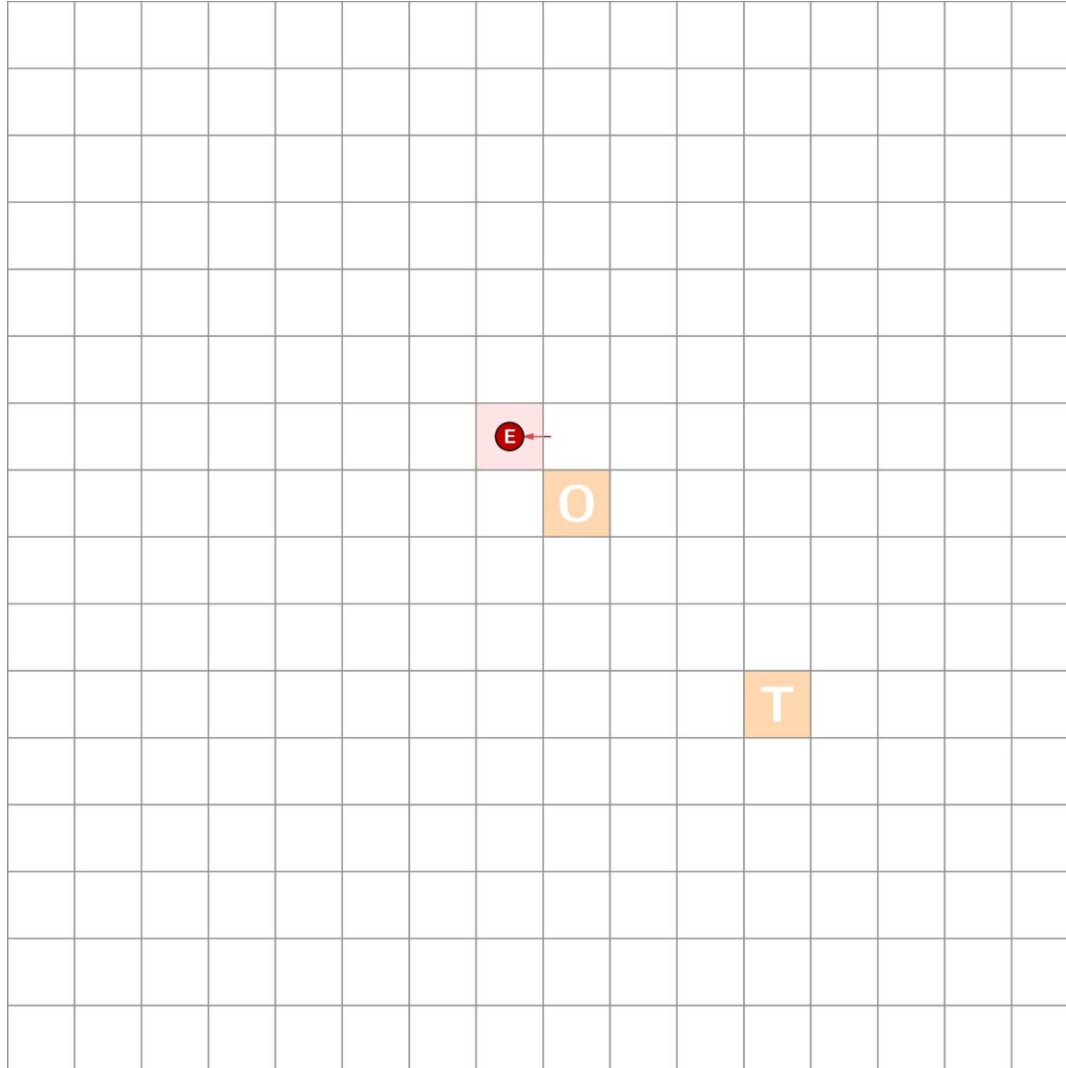
Model



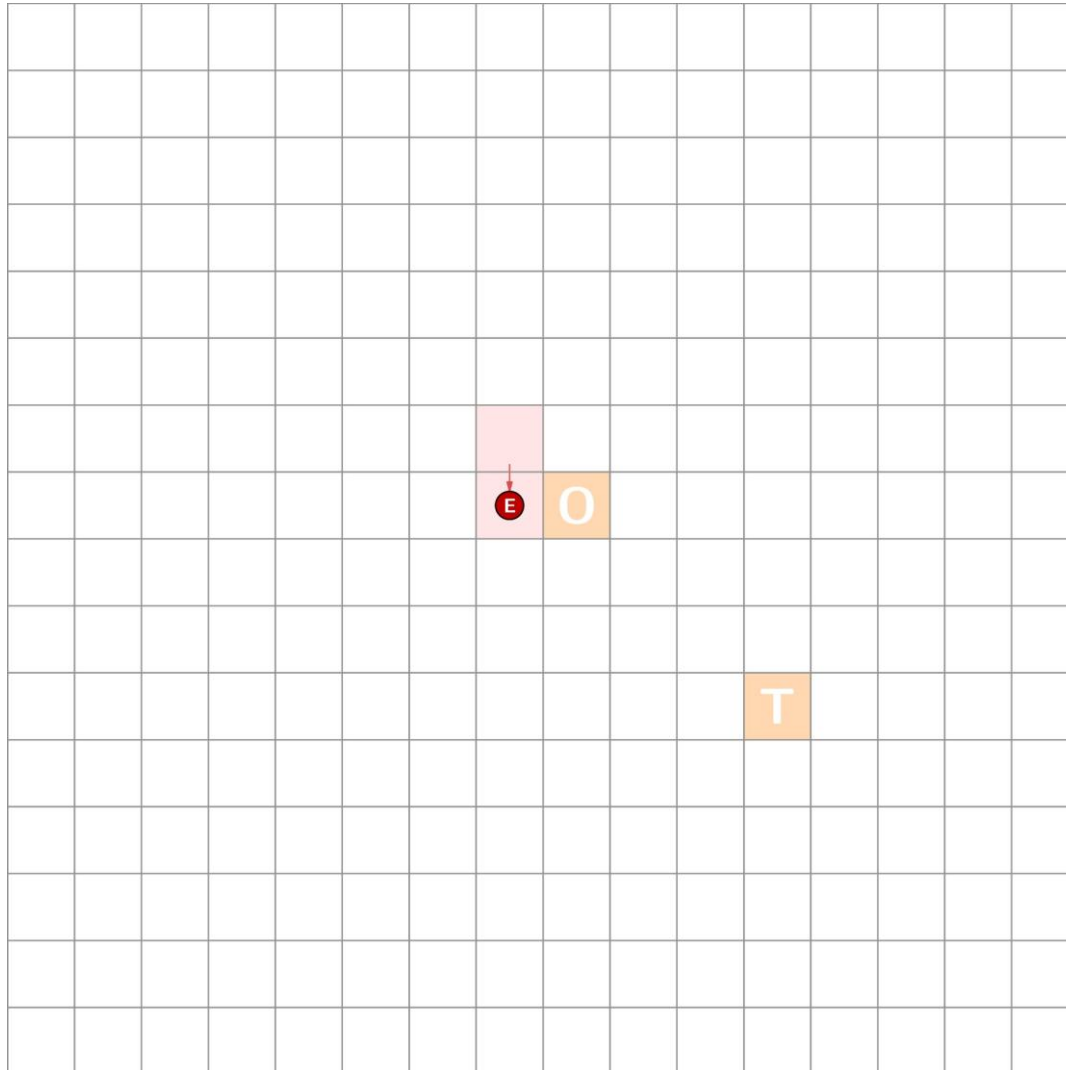
Original Model

- ANTS (Ants Nearby Treasure Search) problem by Feinerman, Korman, Lotker, and Sereni [PODC 2012]
- n ants, controlled by Turing machines.
- Communication restricted to origin.

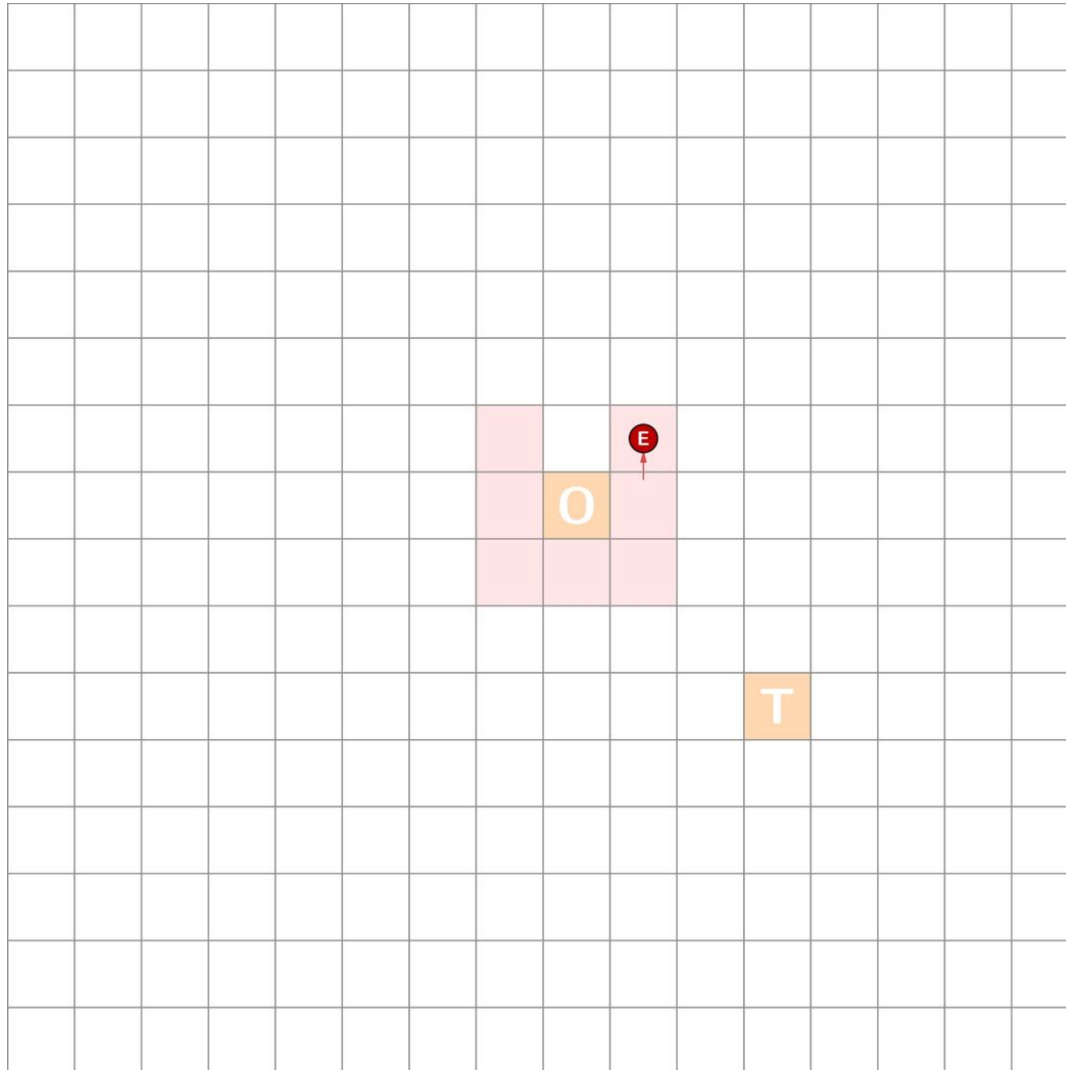
Spiral Search



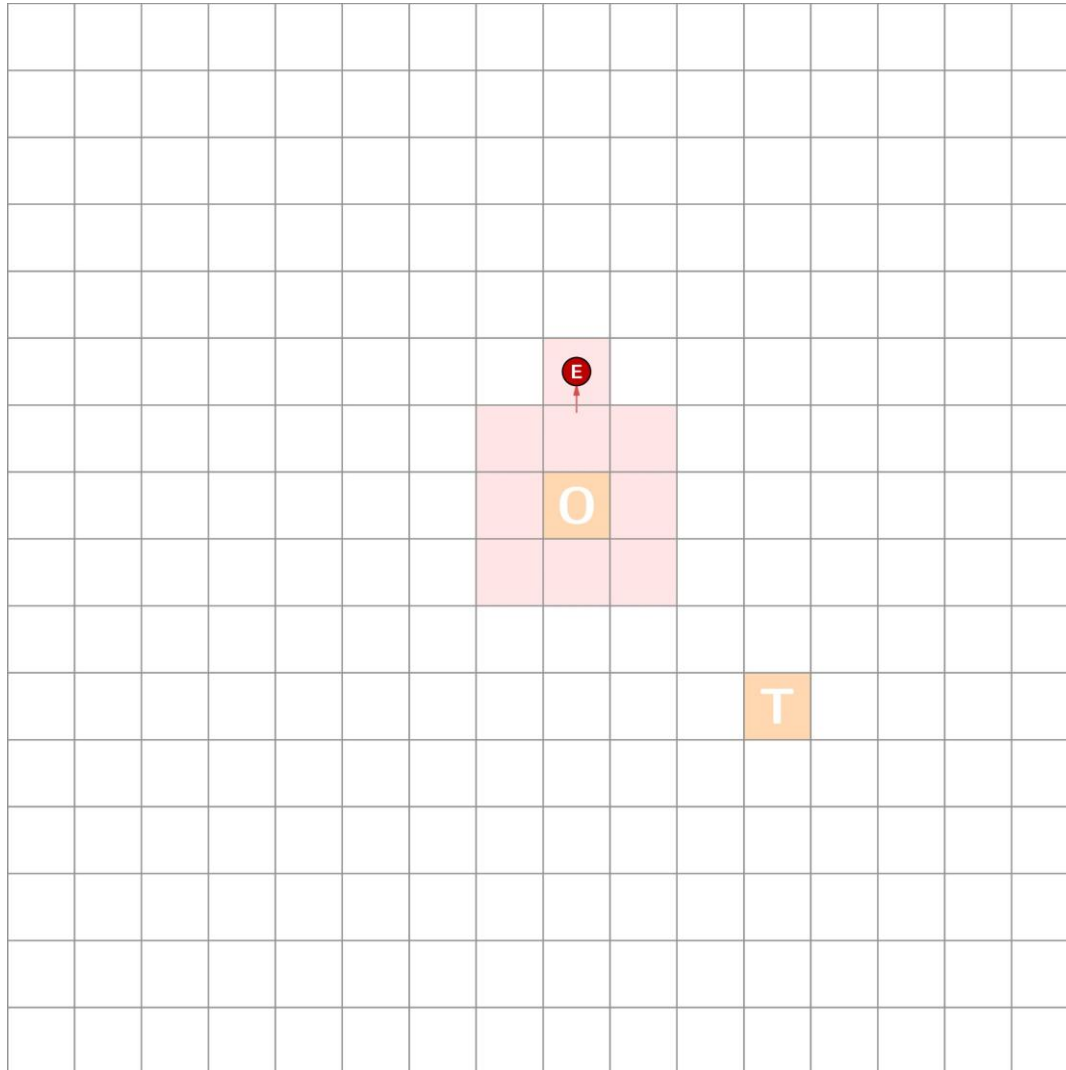
Spiral Search



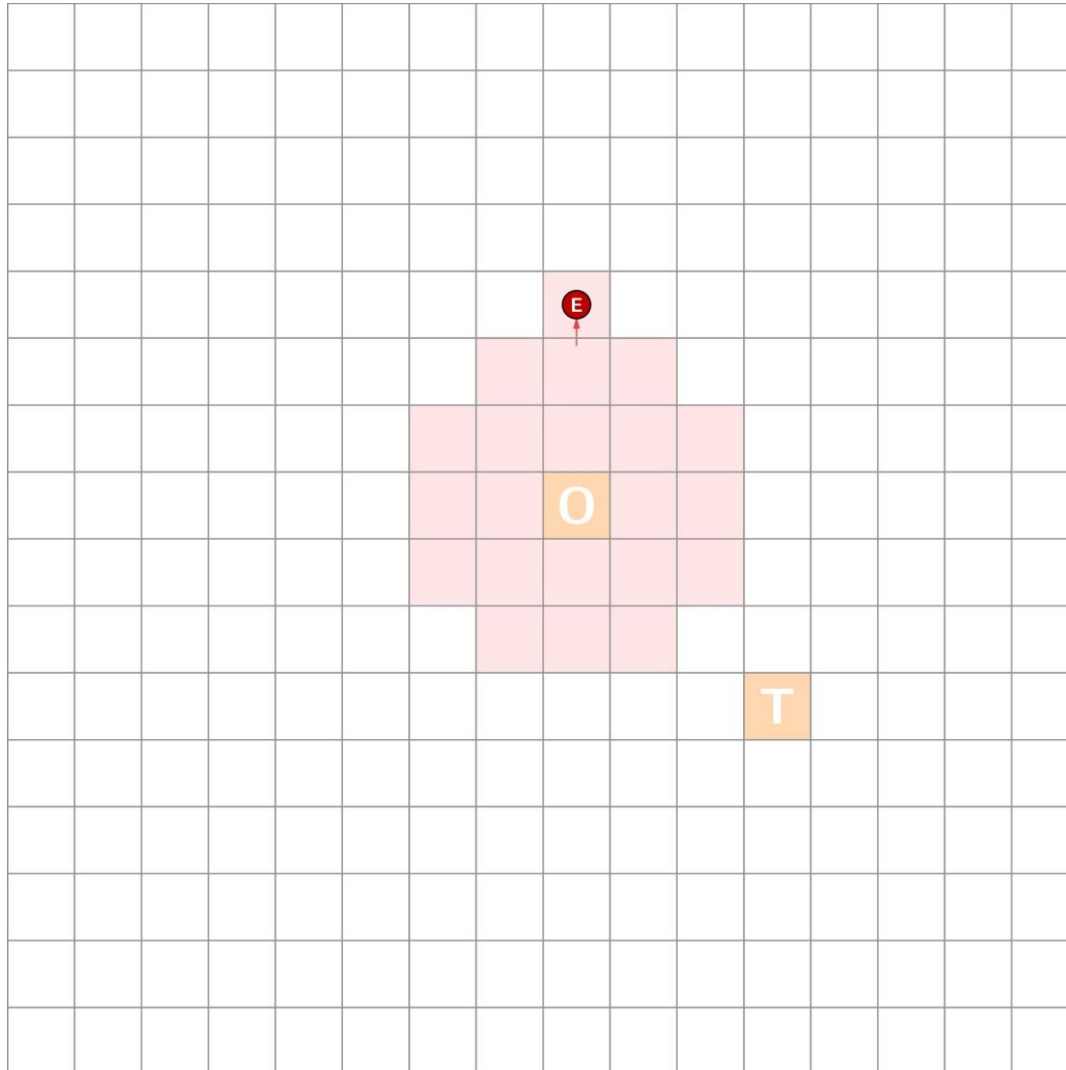
Spiral Search



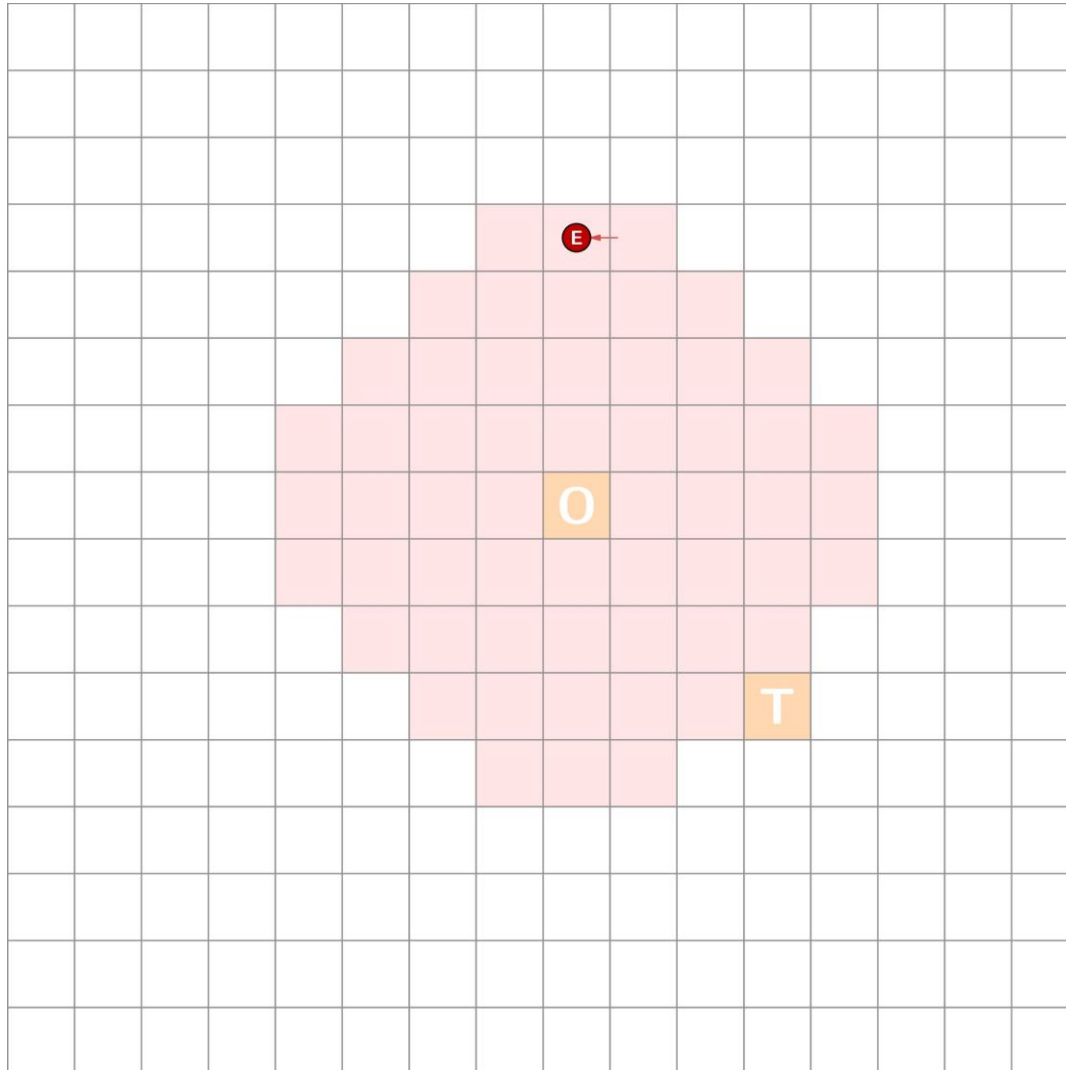
Spiral Search



Spiral Search



Spiral Search



Spiral Search

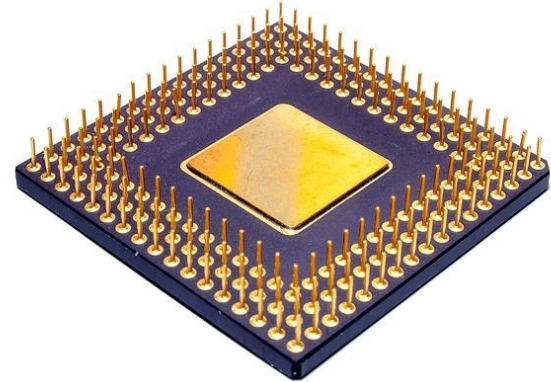
- Parallelized version achieves a running time of $O\left(D + \frac{D^2}{n}\right)$.
- The bound is tight, but requires a constant approximation of n .

Our Model



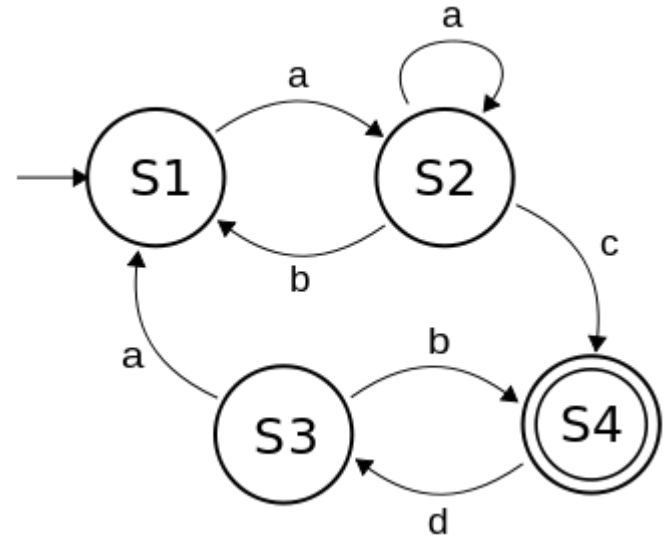
?

||

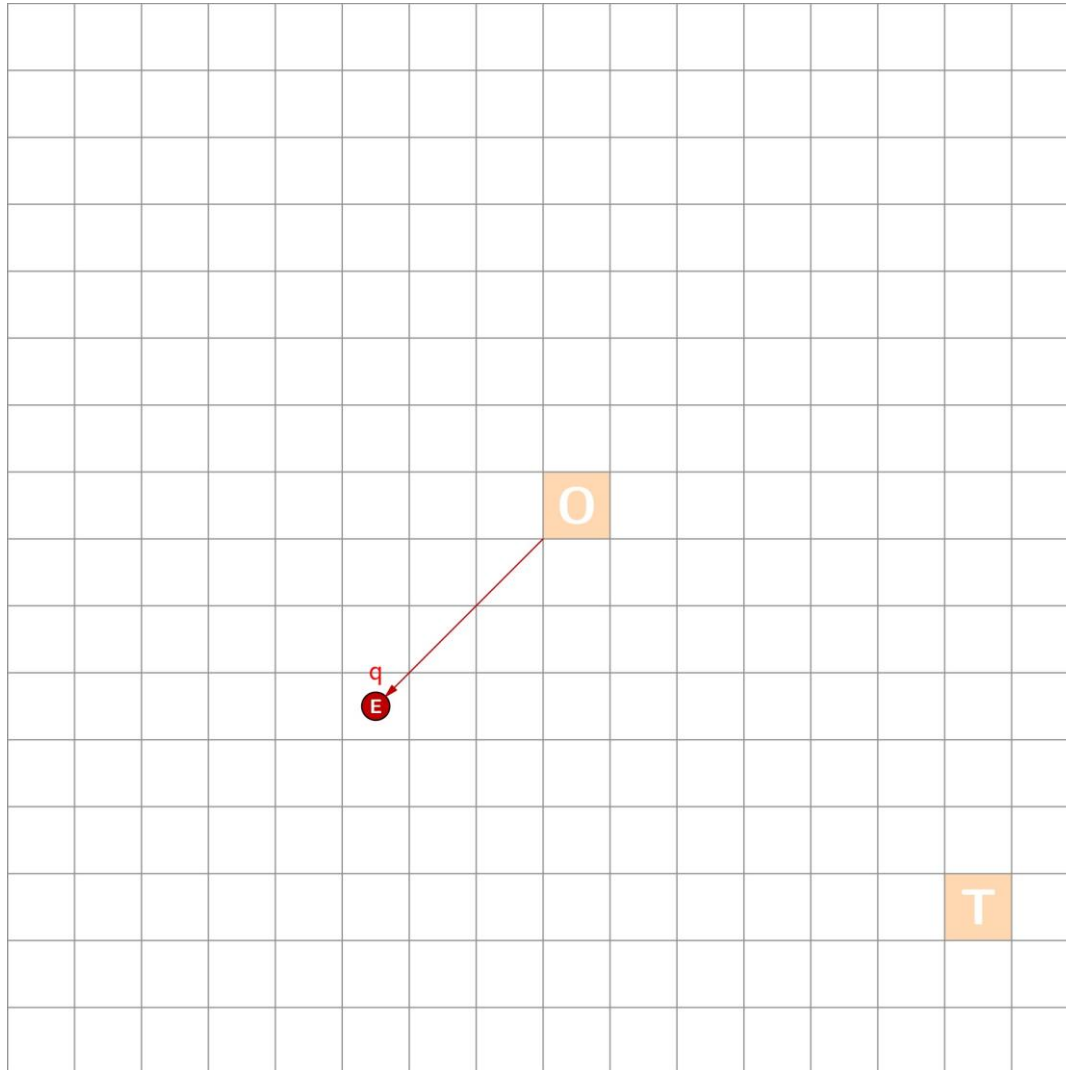


Our Model

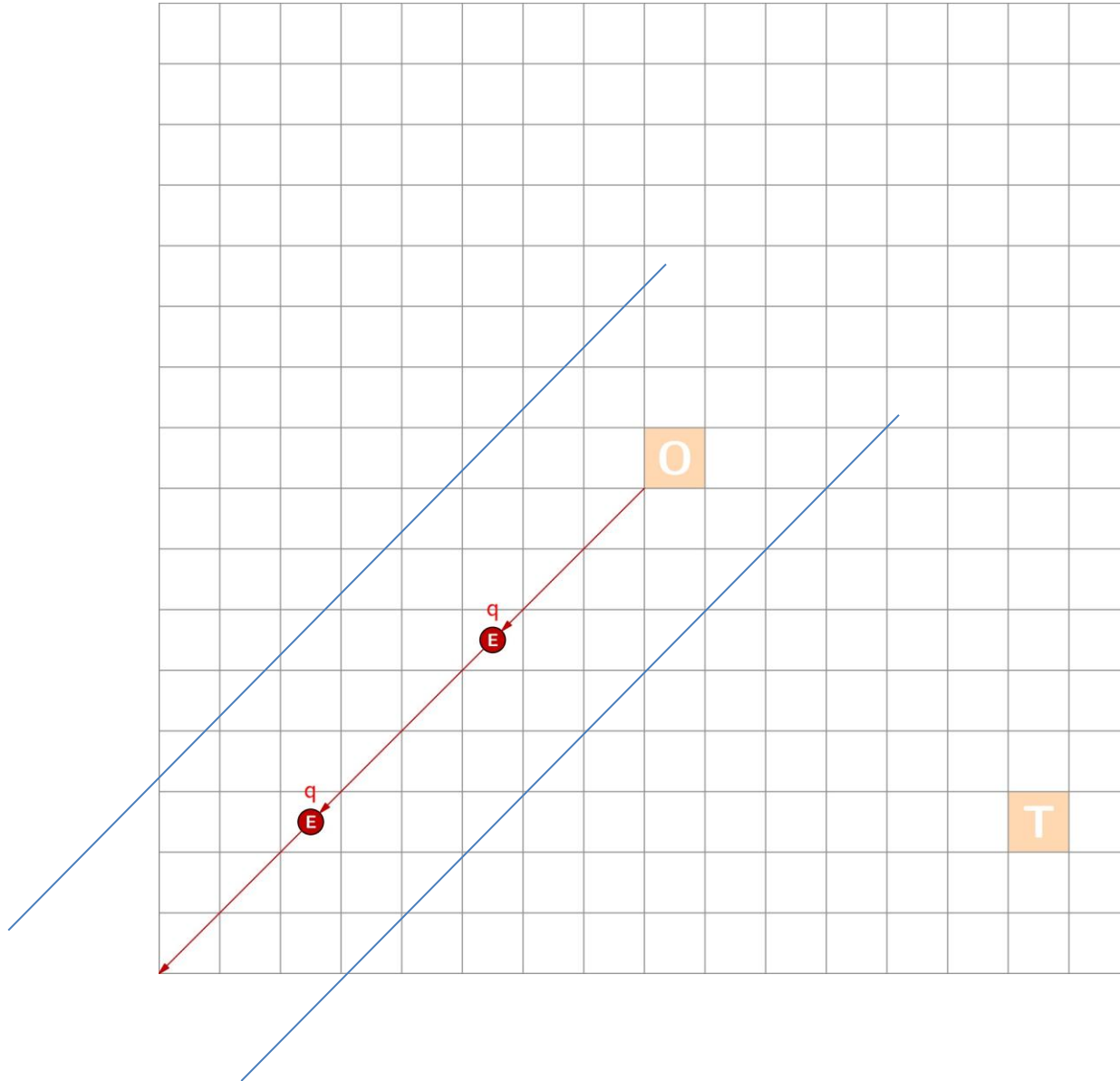
- Ants controlled by finite state machines [ICALP2014].



Our Model

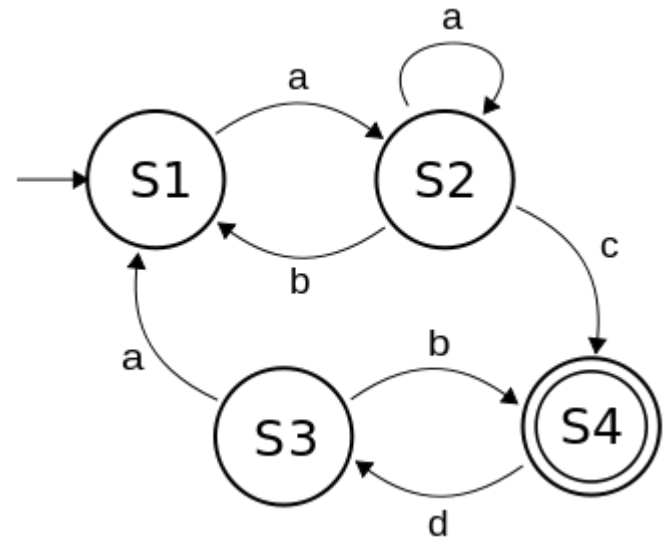


Our Model

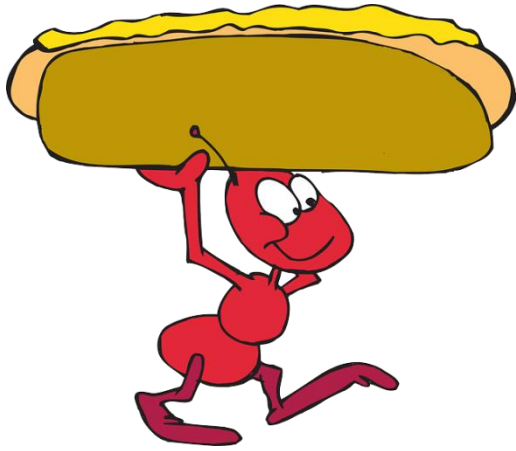


Model

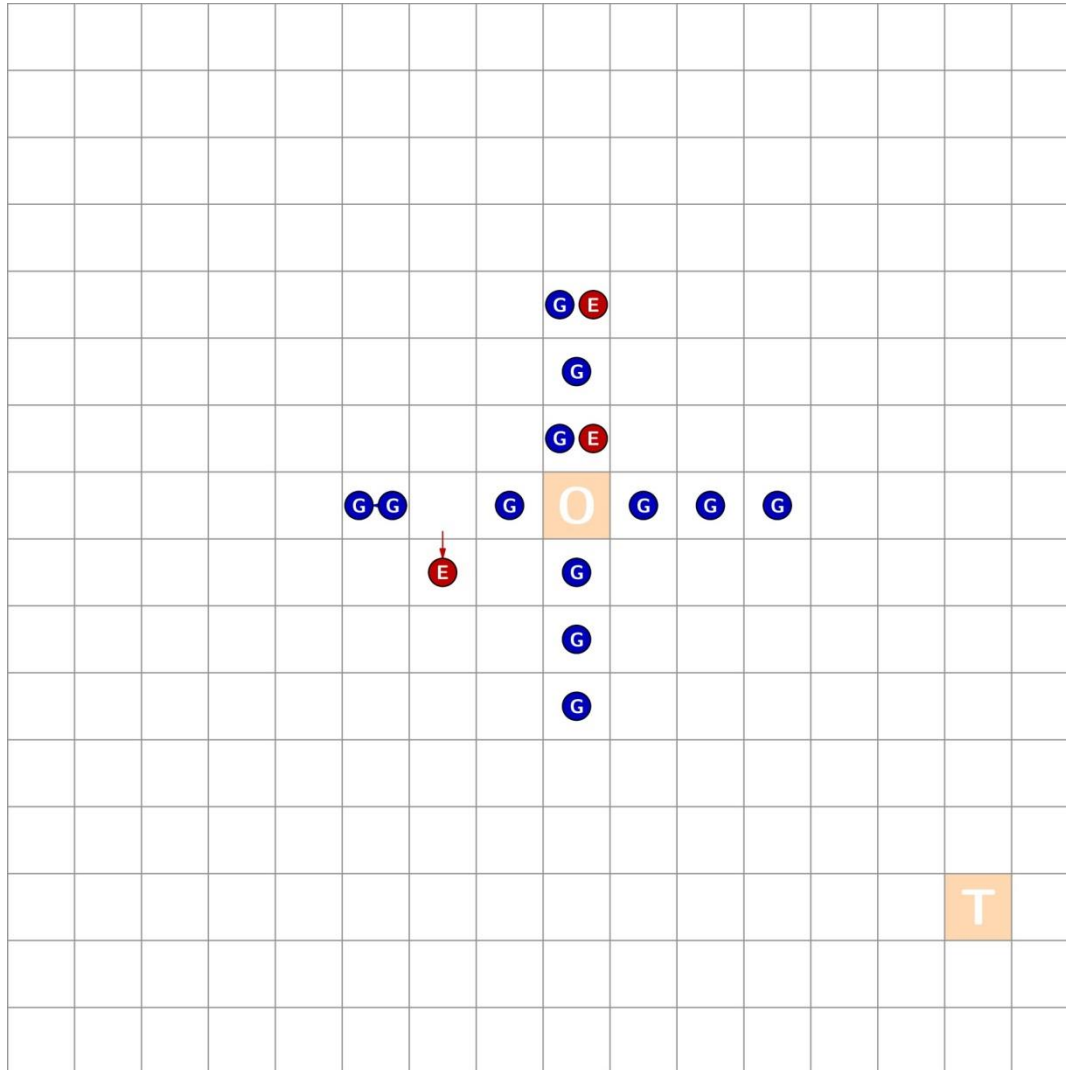
- Ants controlled by finite state machines.
- Communication allowed with nearby ants.



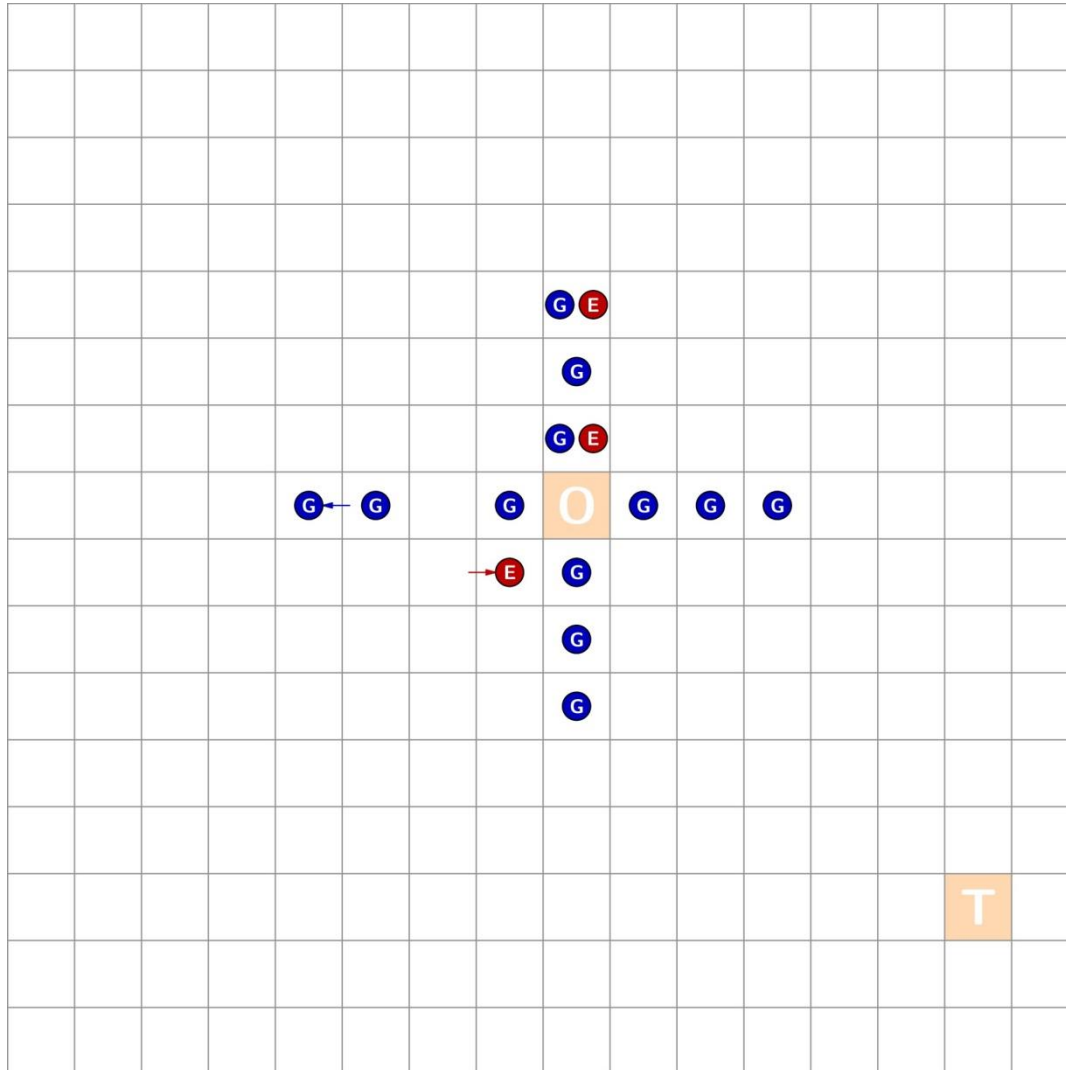
Diamond Search



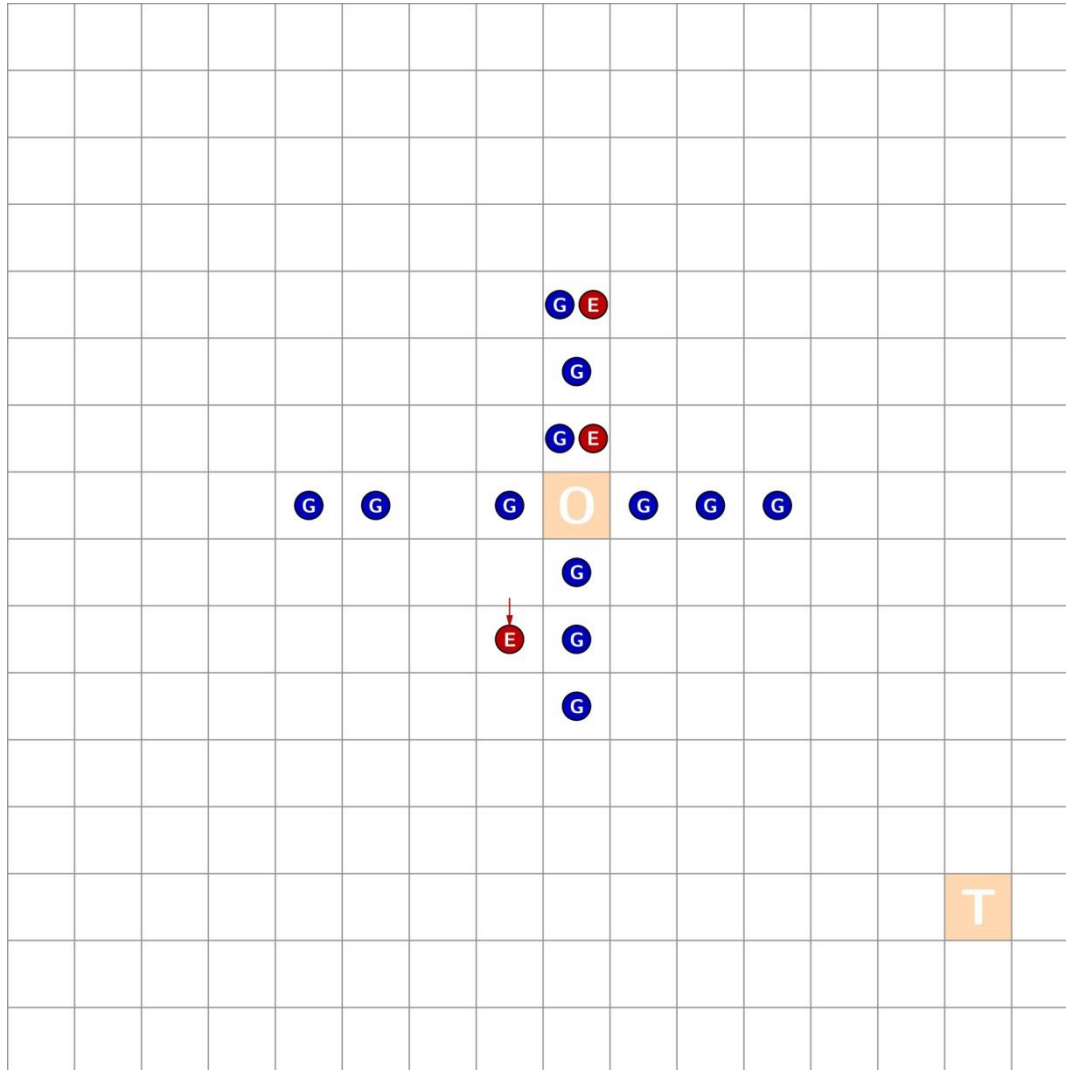
Diamond Search



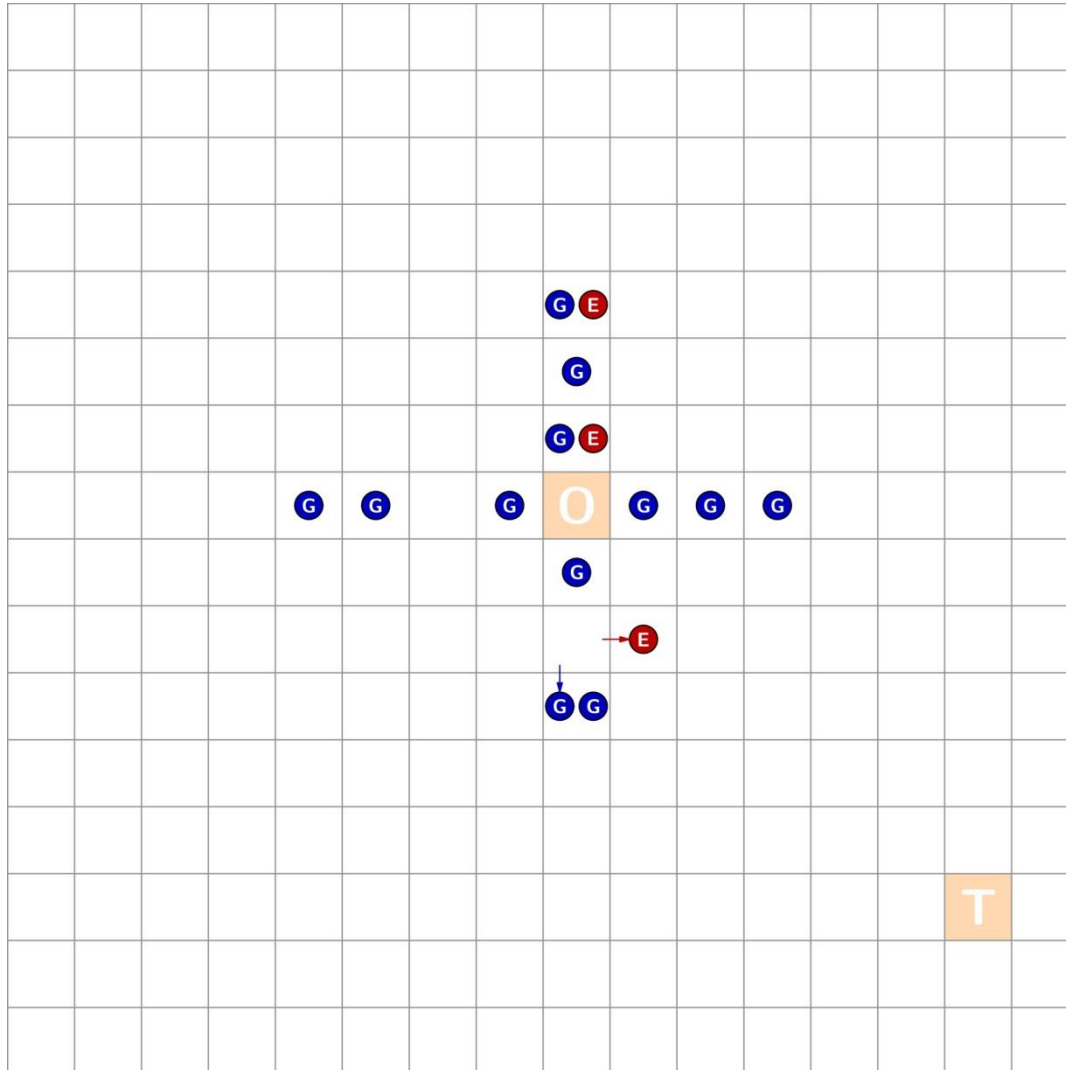
Diamond Search



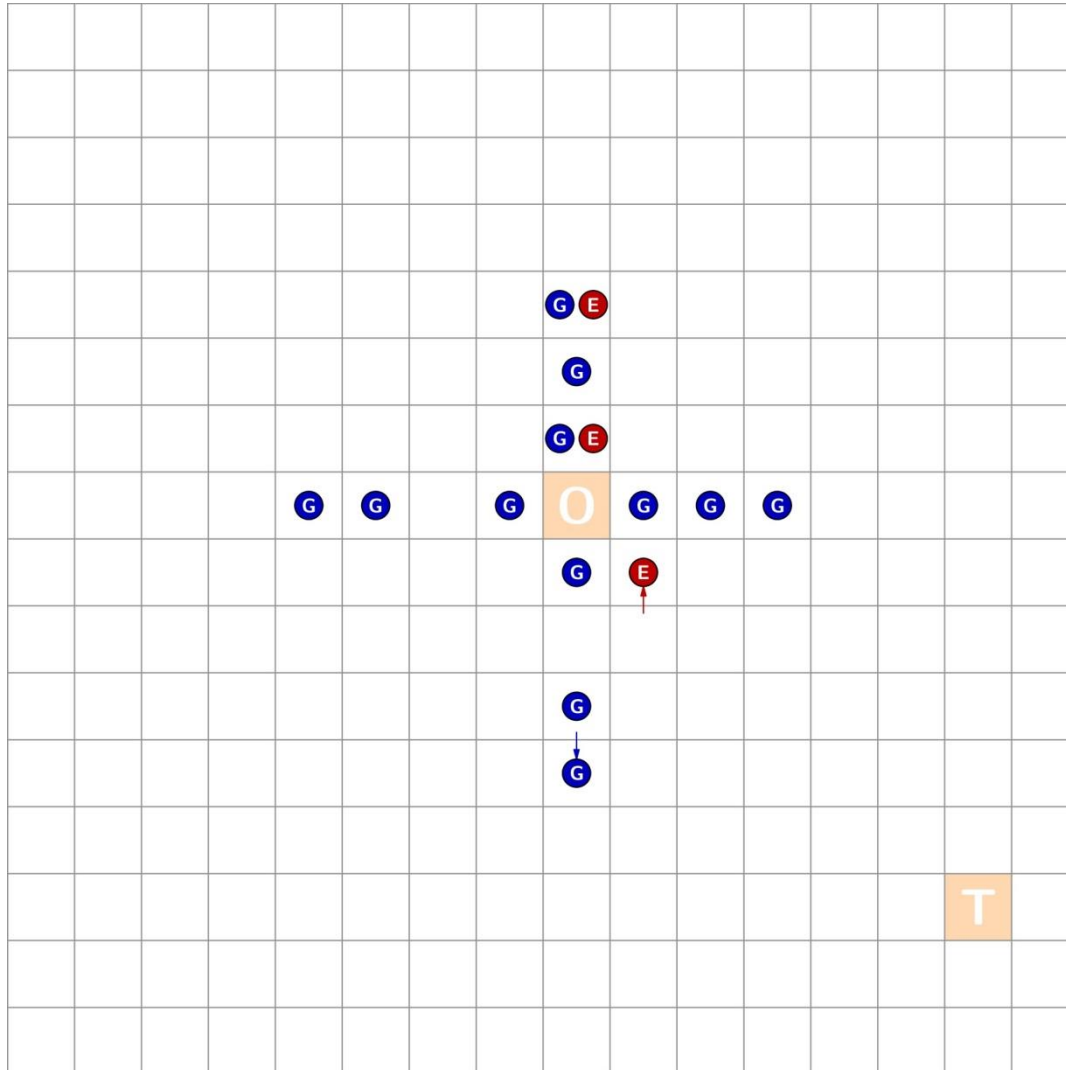
Diamond Search



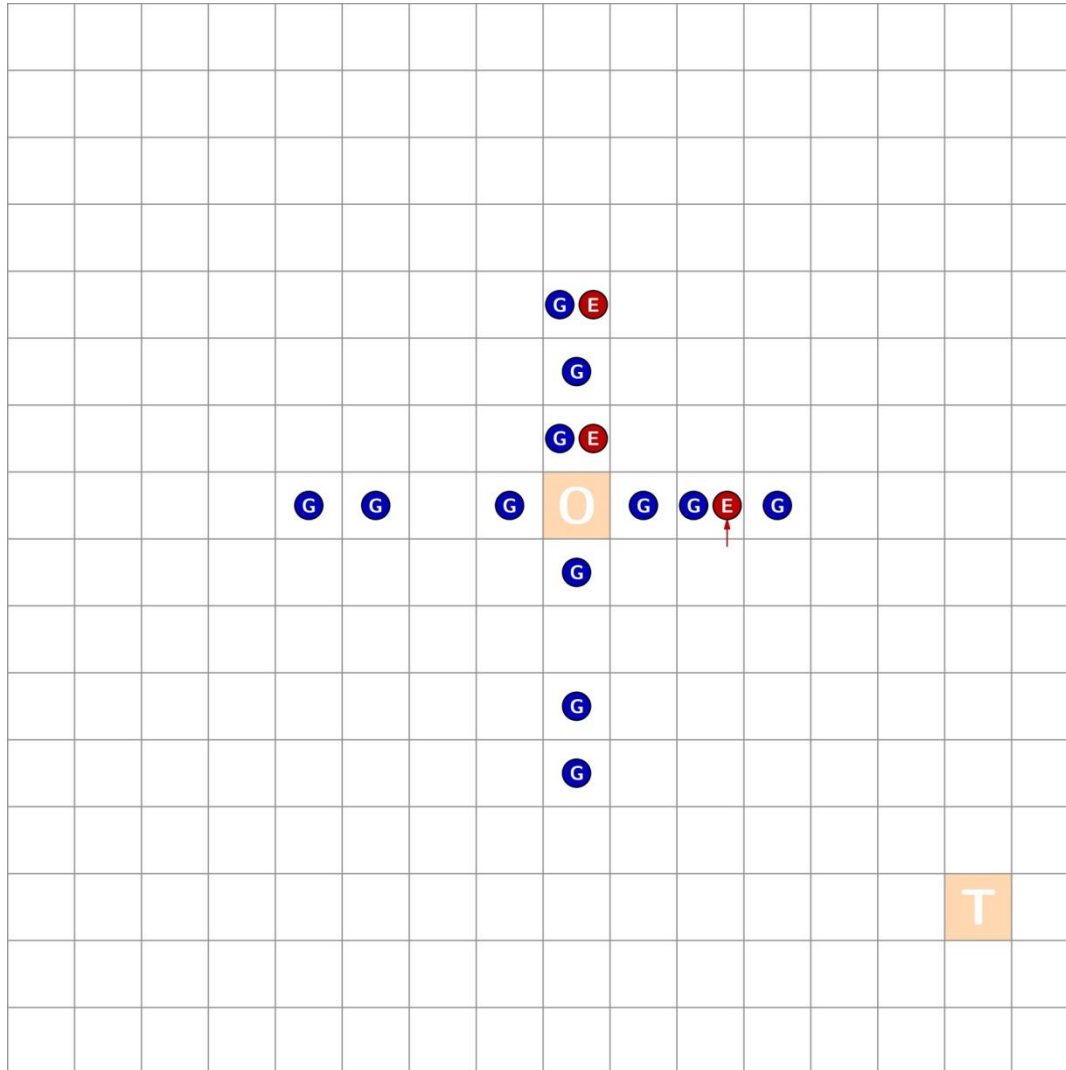
Diamond Search



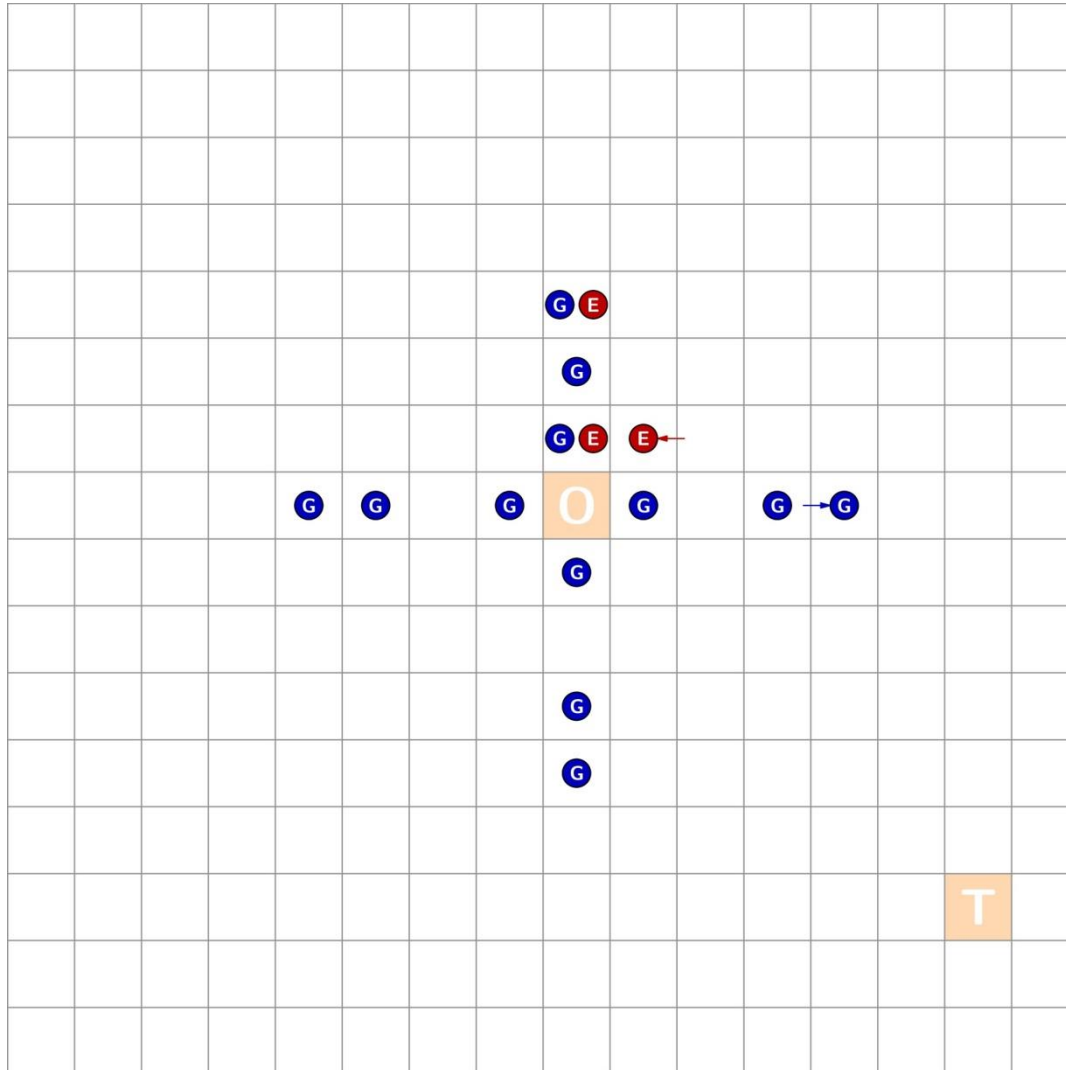
Diamond Search



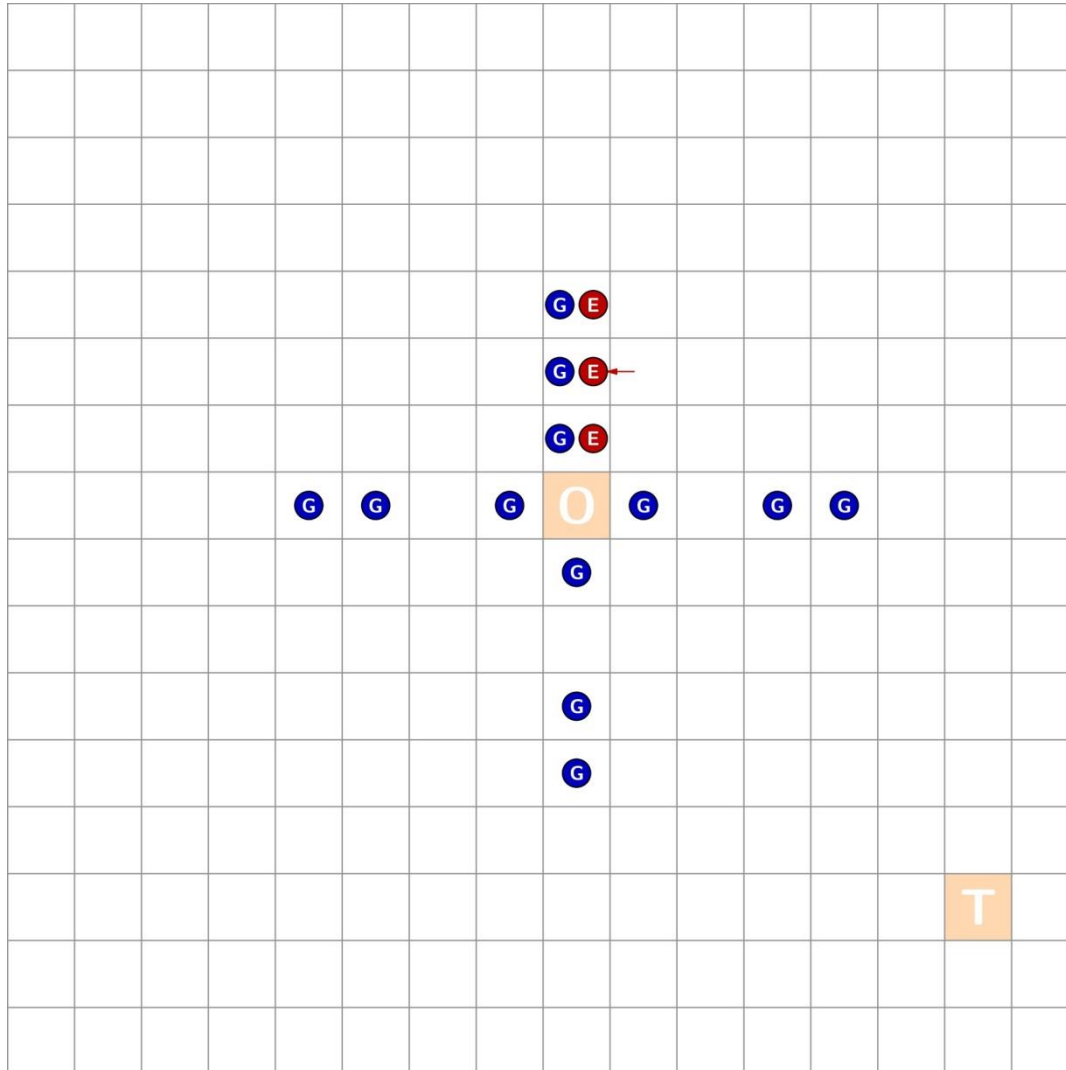
Diamond Search



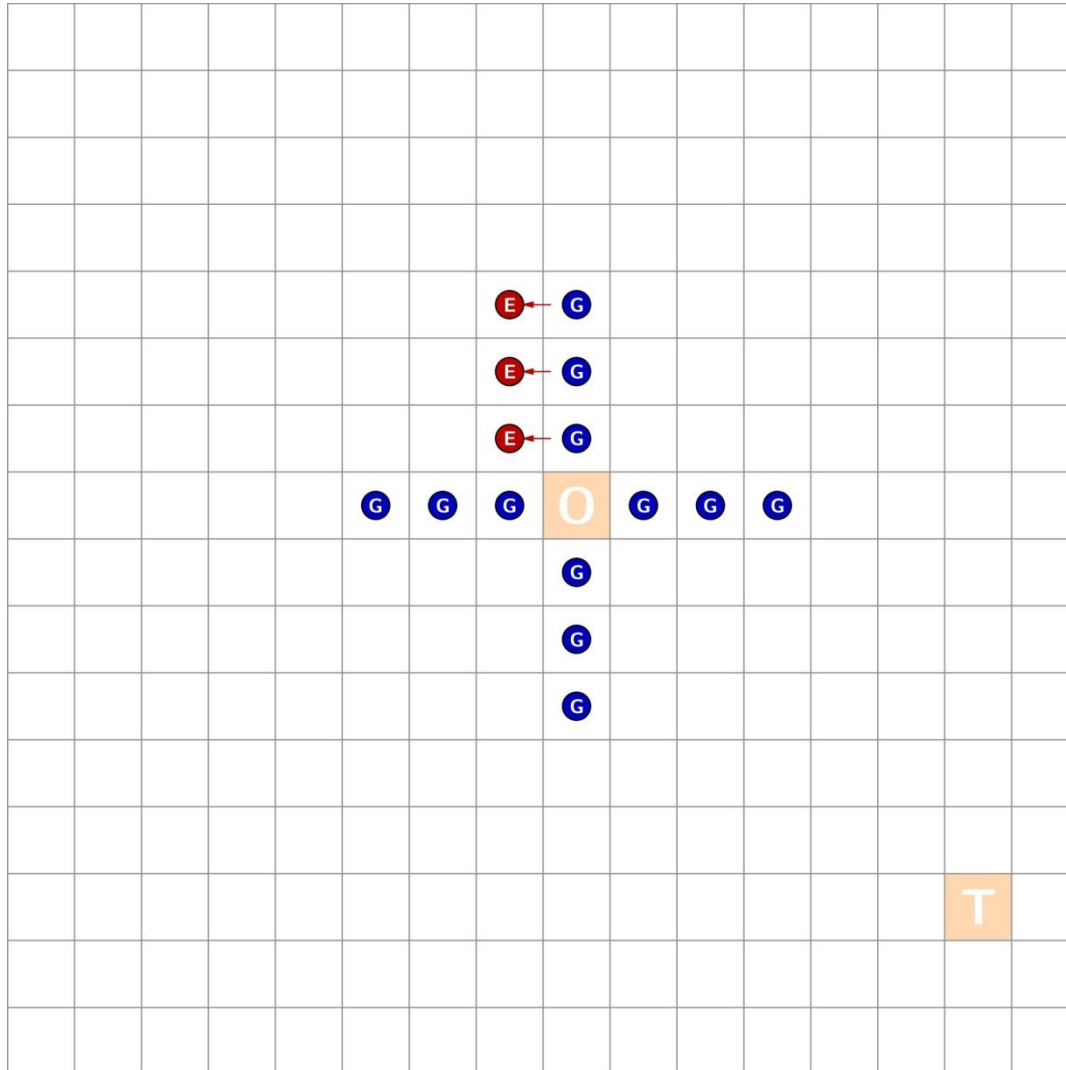
Diamond Search



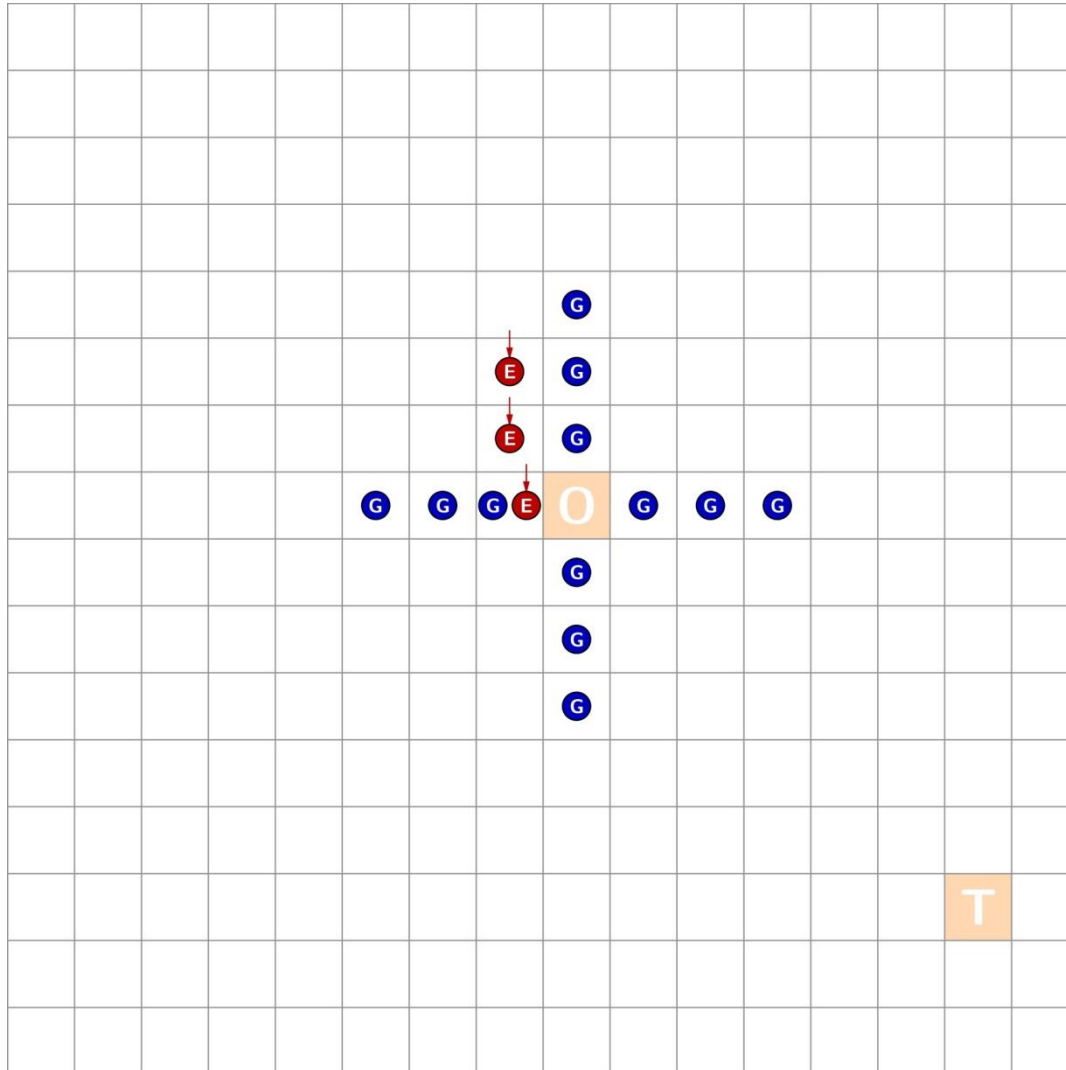
Diamond Search



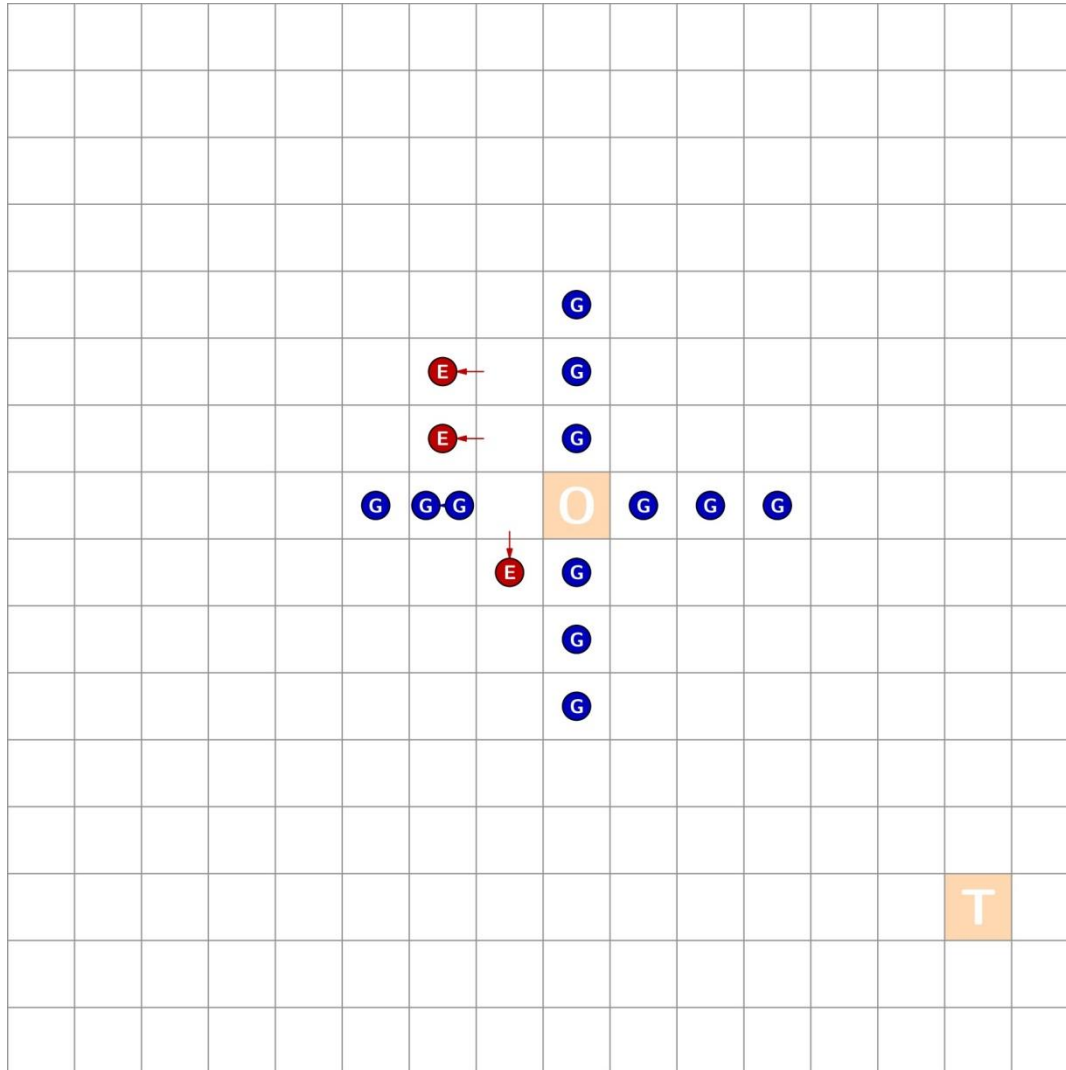
Diamond Search



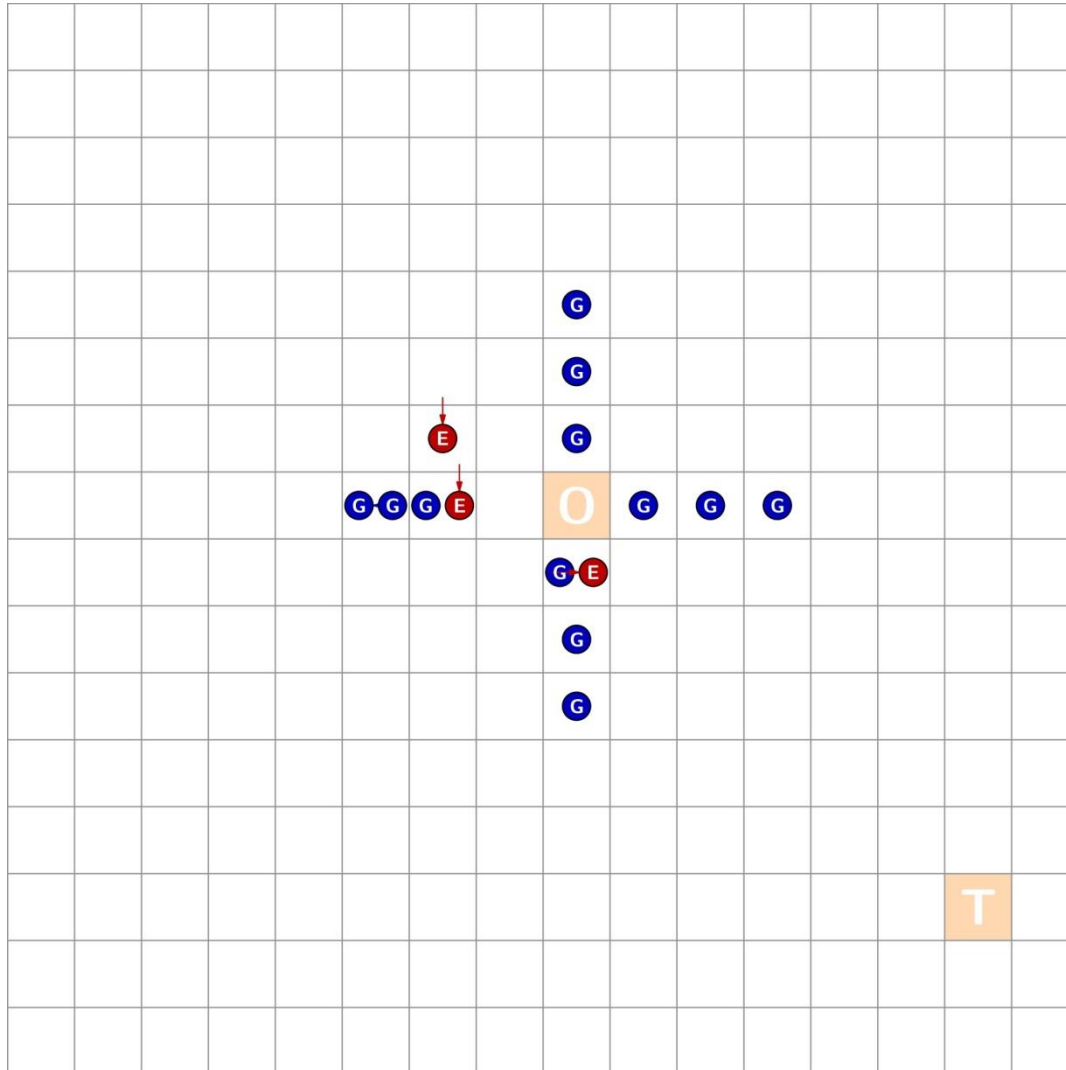
Diamond Search



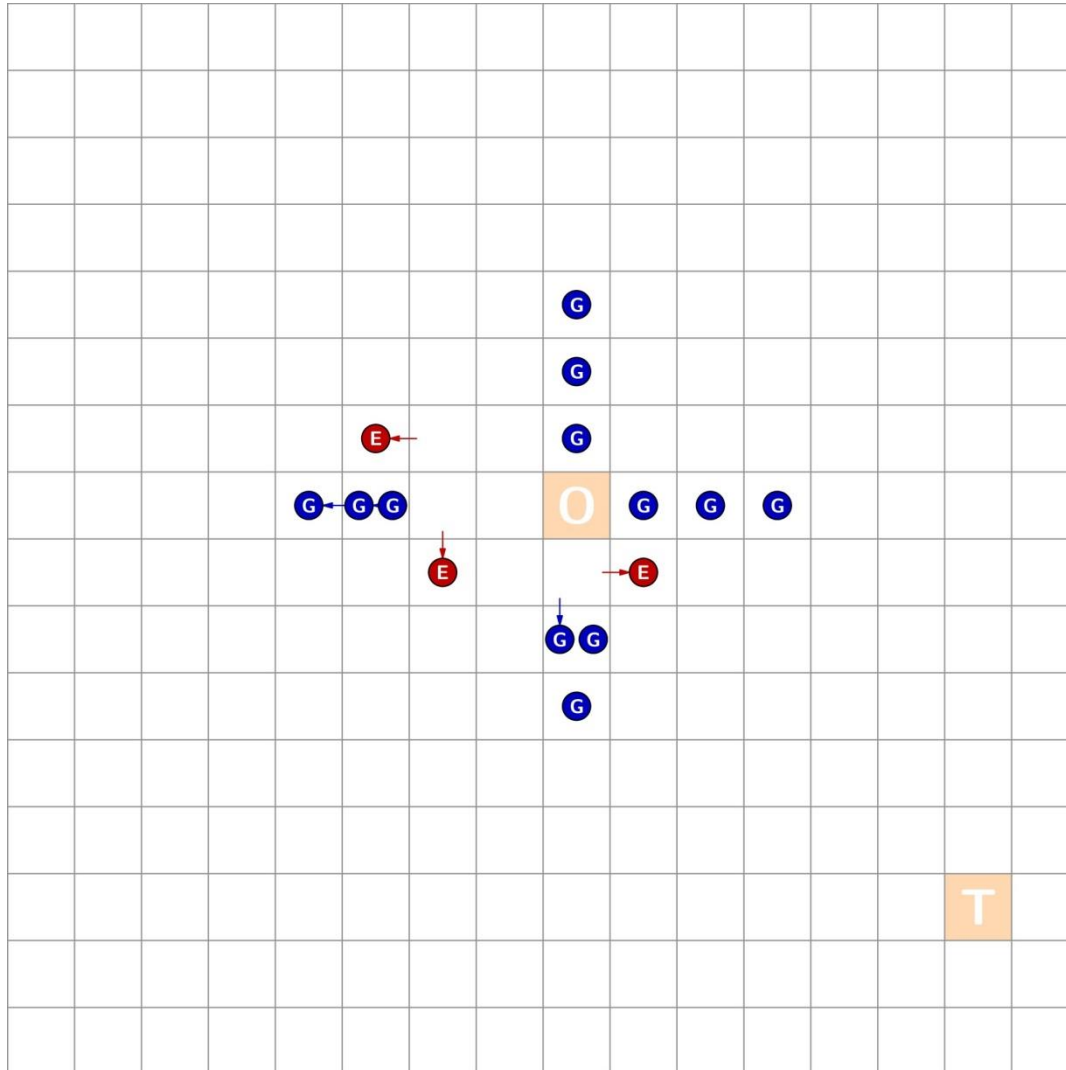
Diamond Search



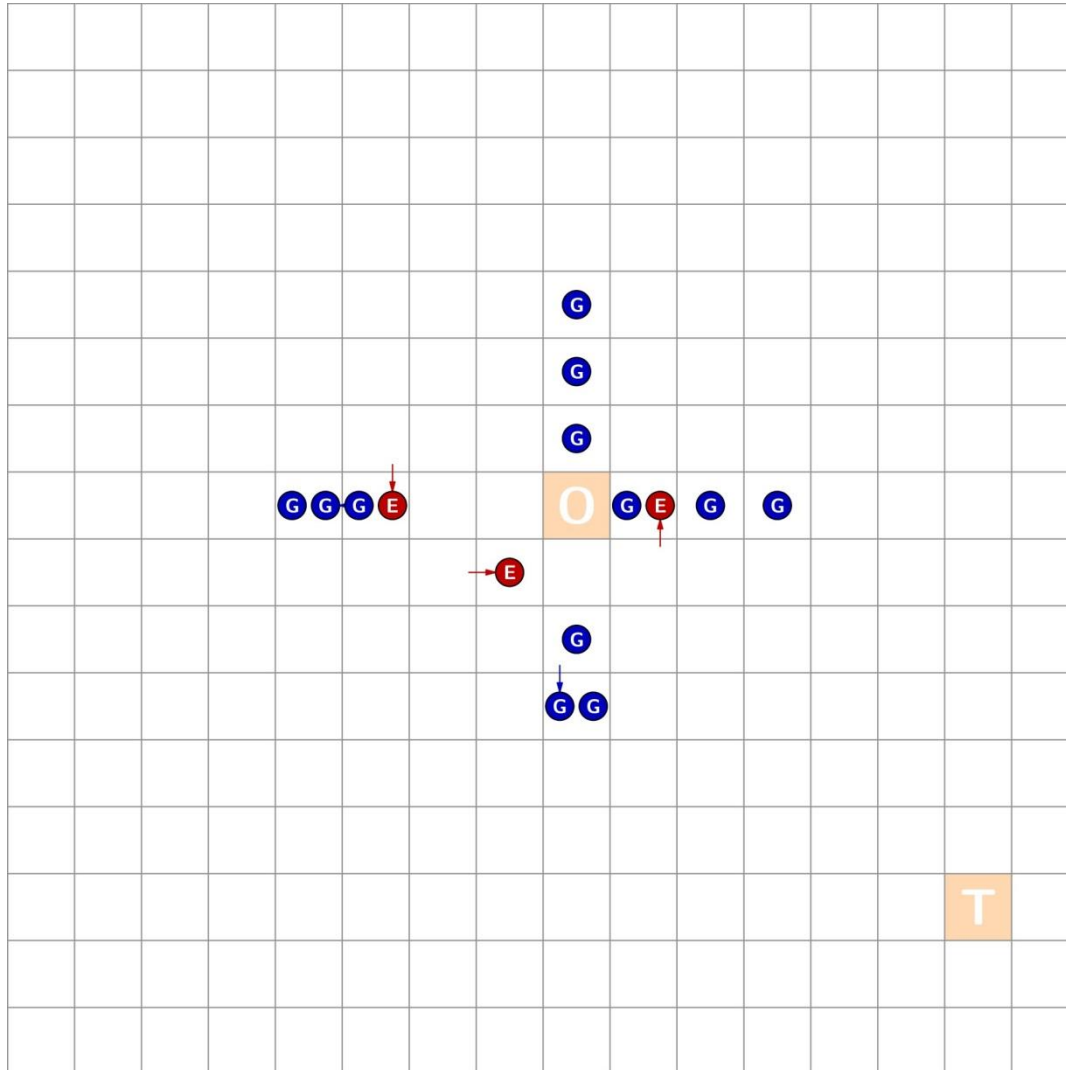
Diamond Search



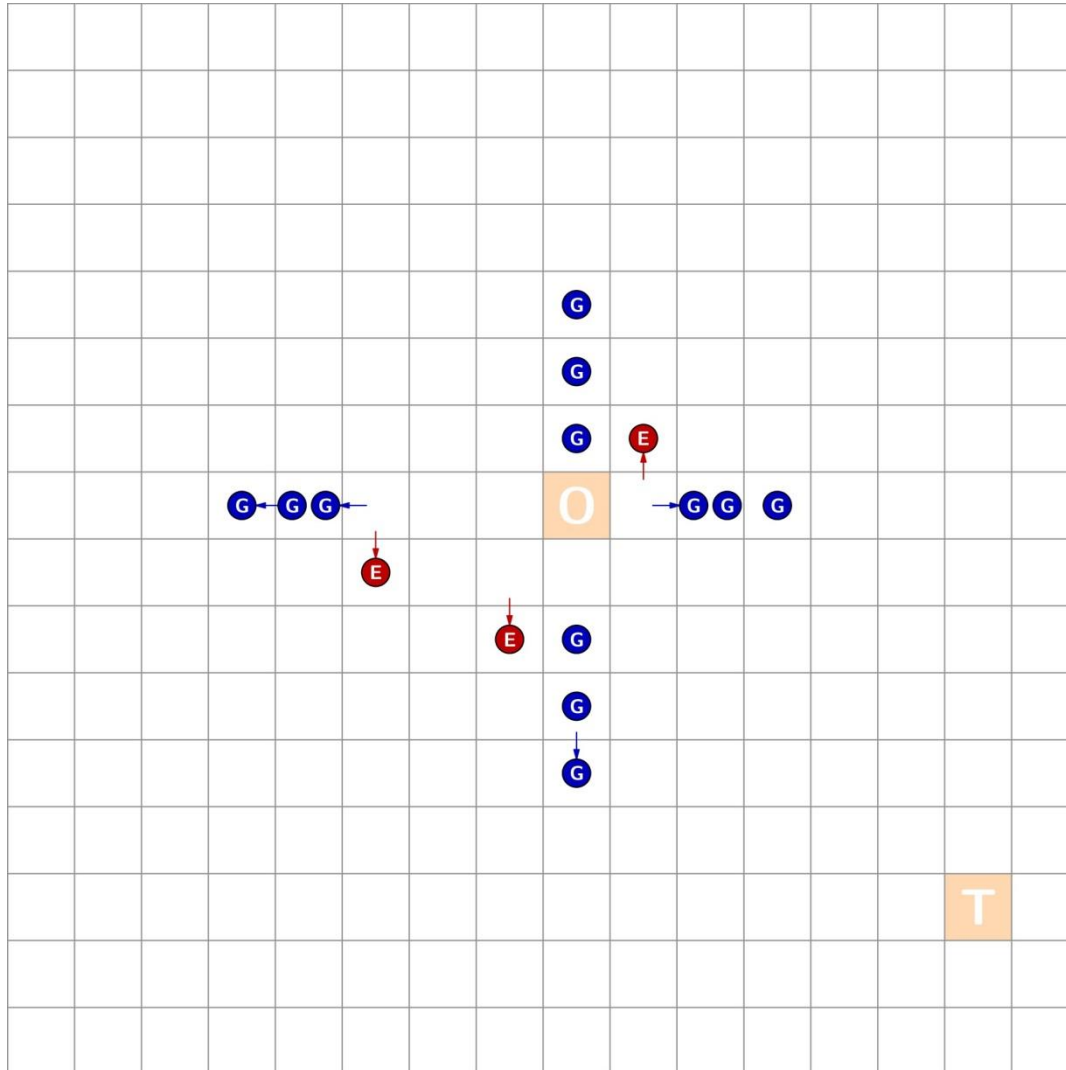
Diamond Search



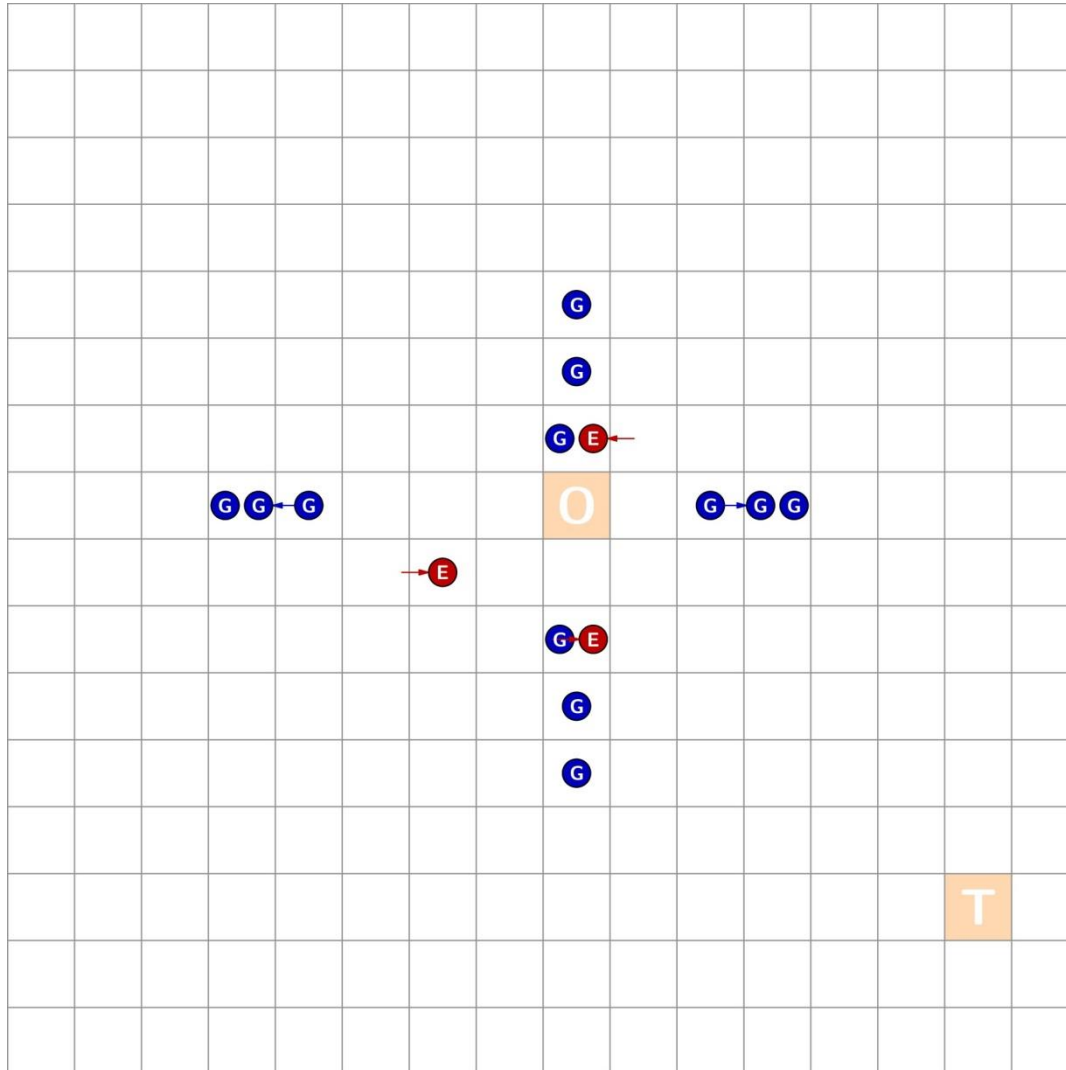
Diamond Search



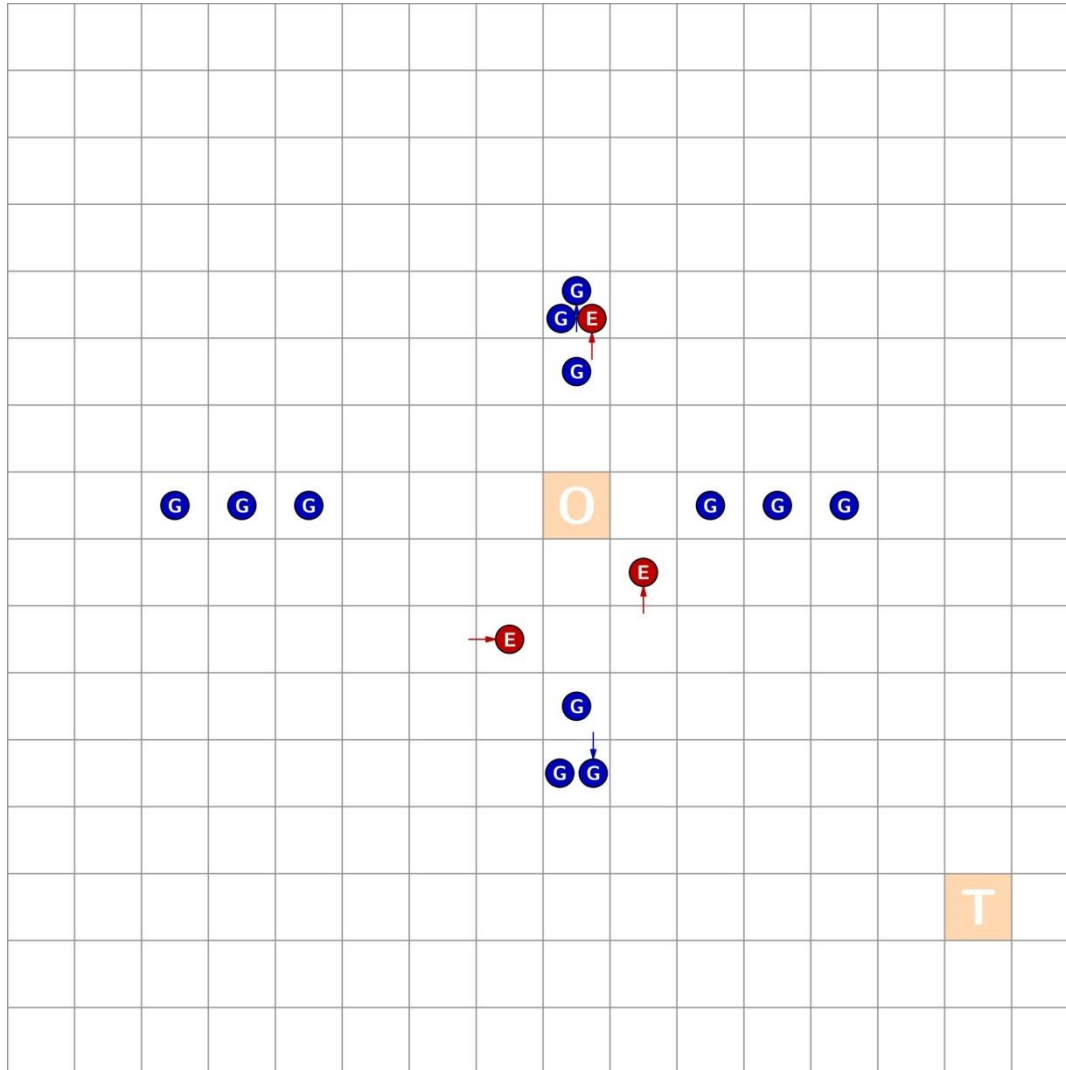
Diamond Search



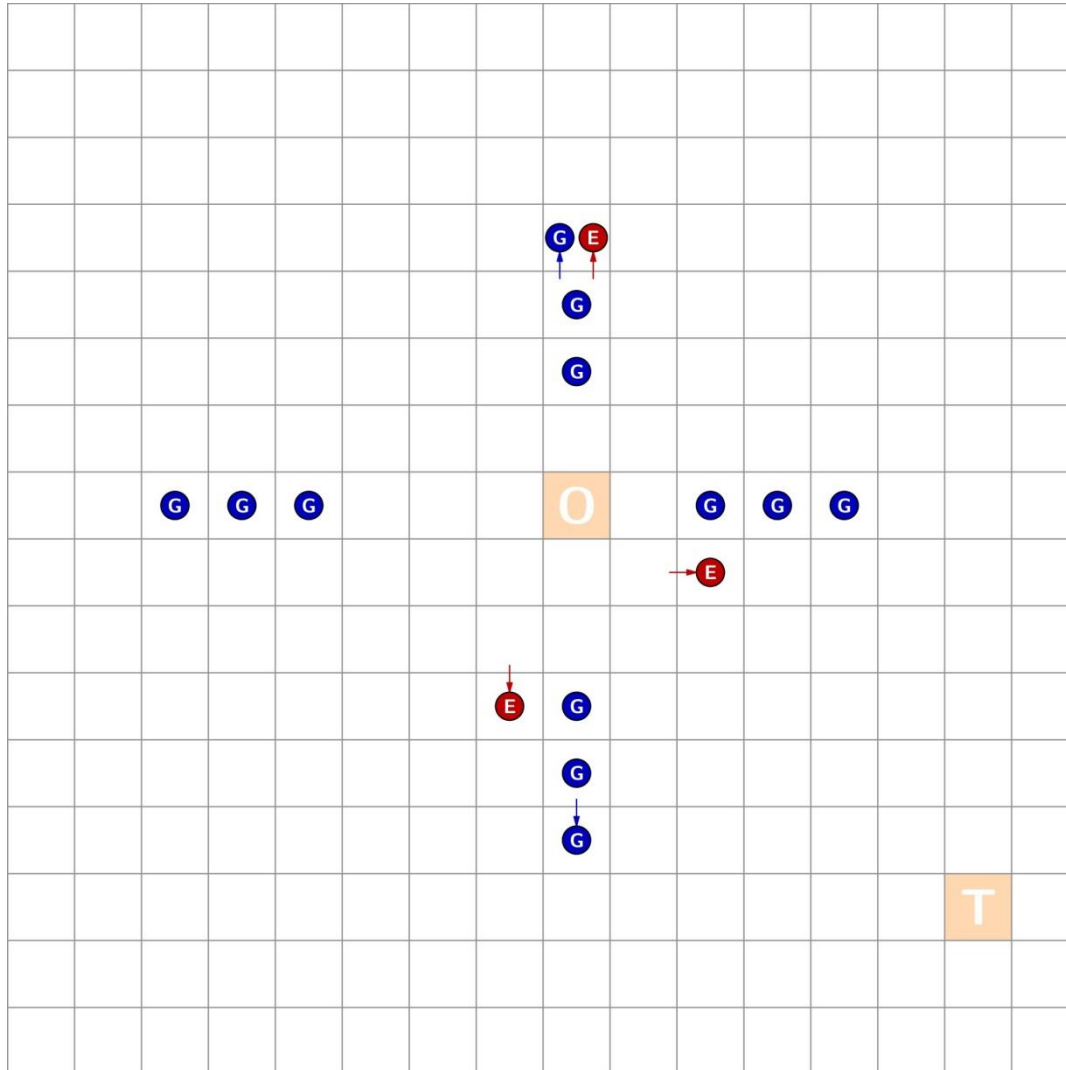
Diamond Search



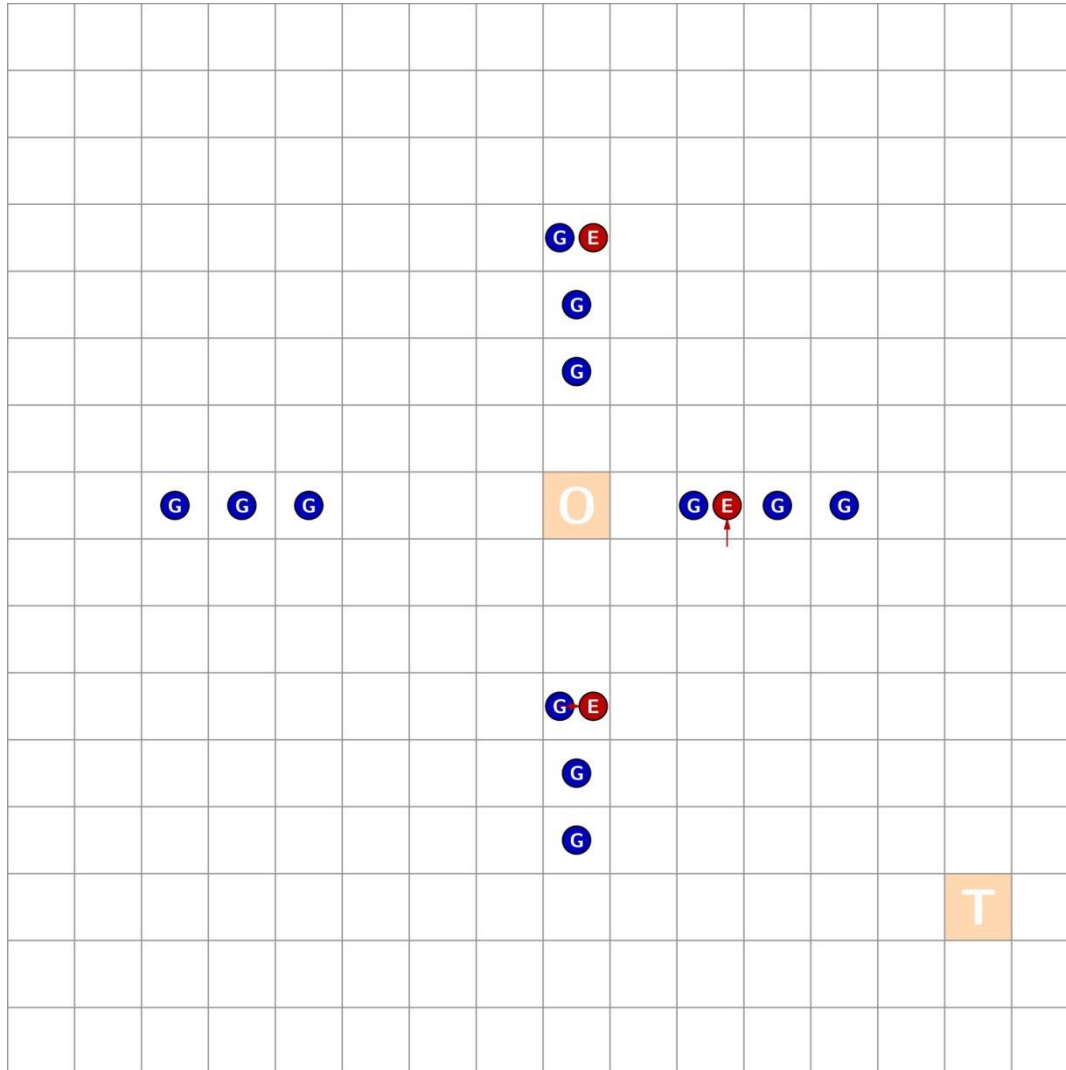
Diamond Search



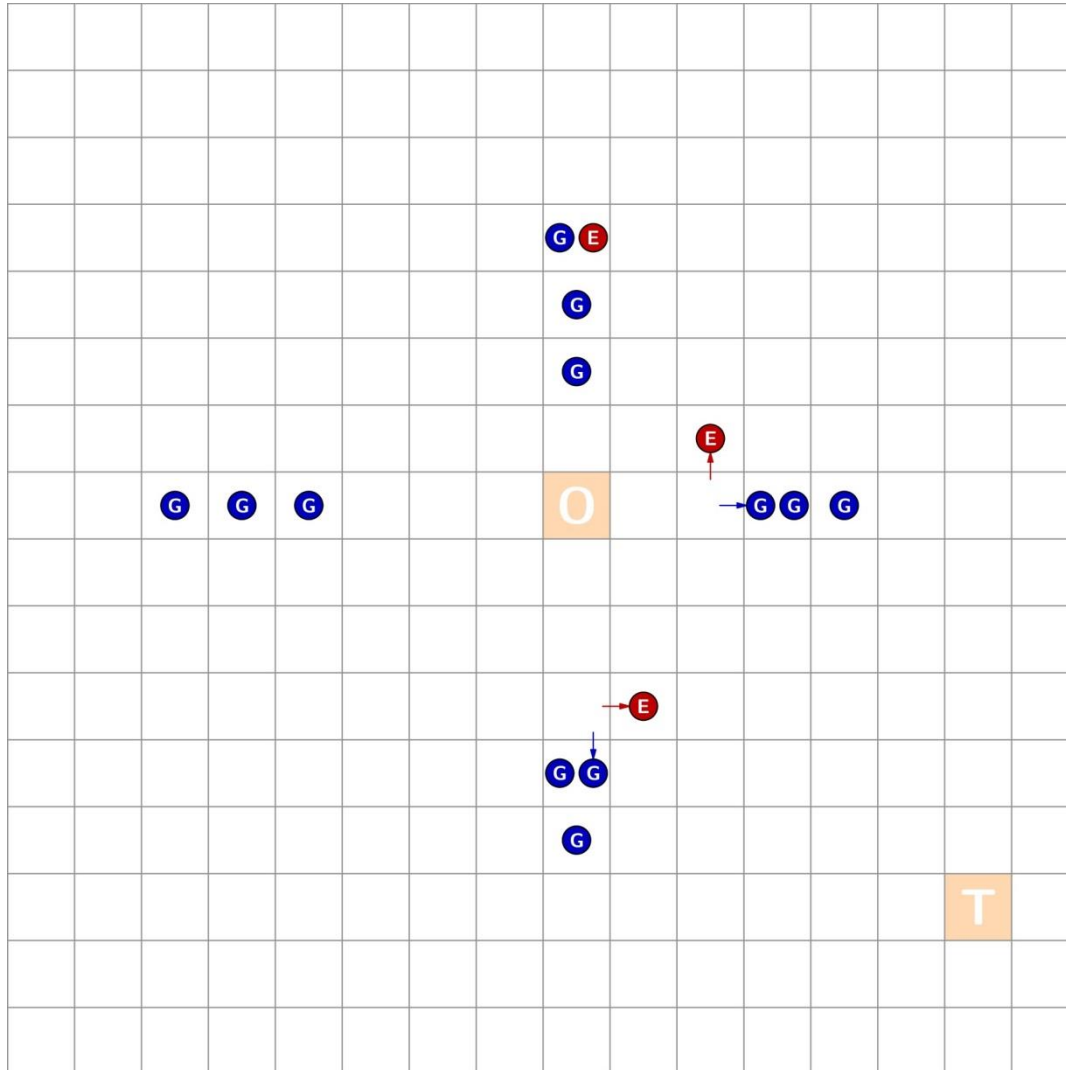
Diamond Search



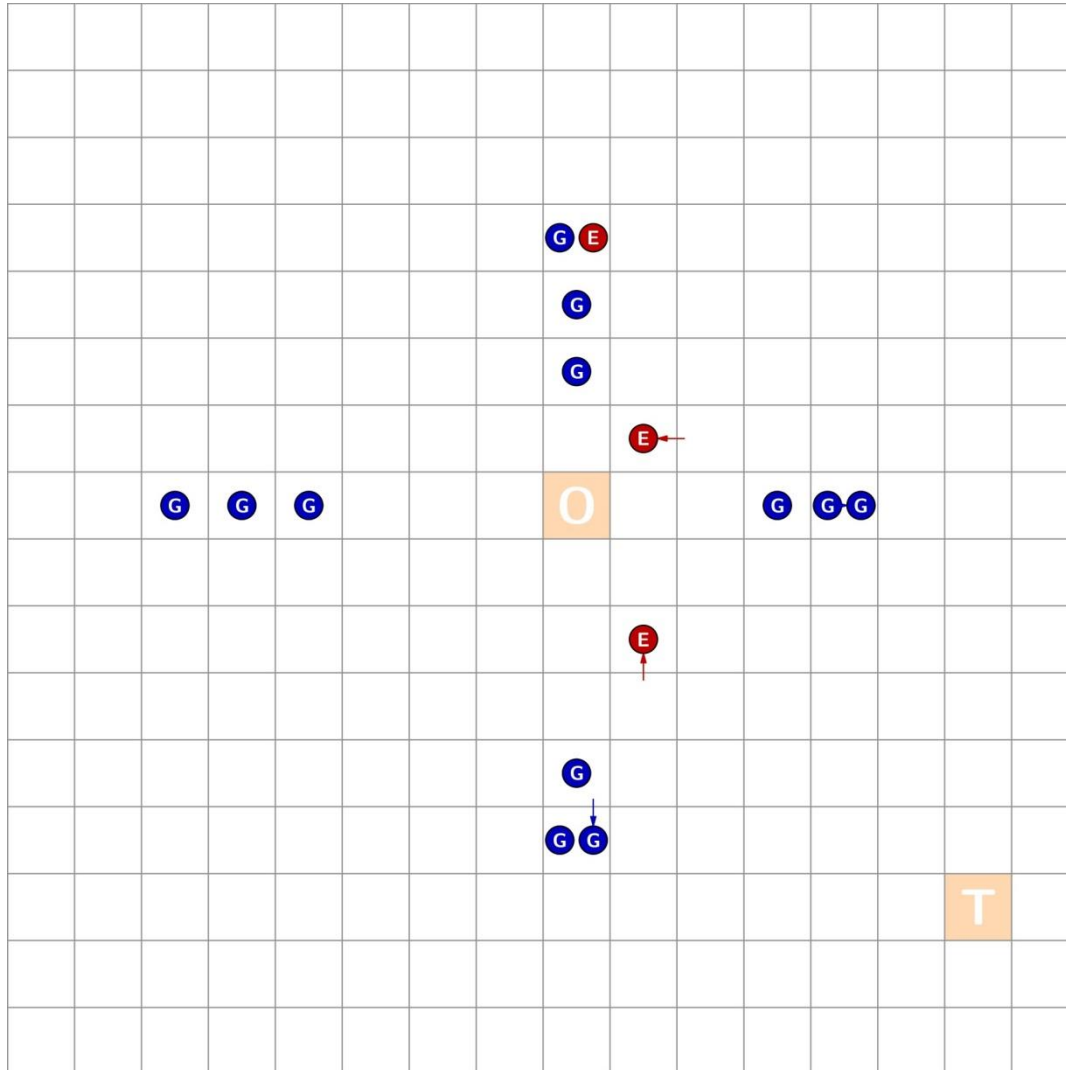
Diamond Search



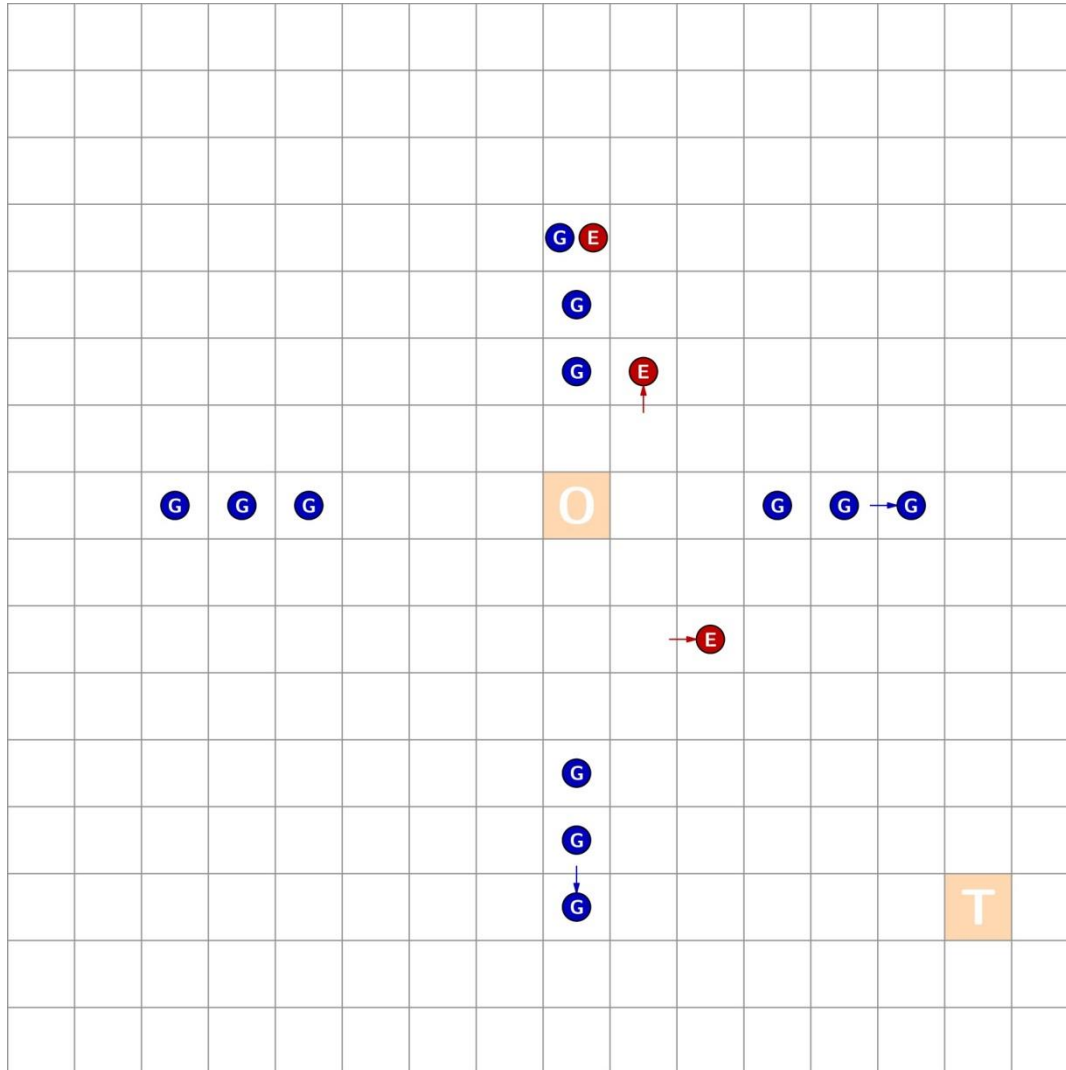
Diamond Search



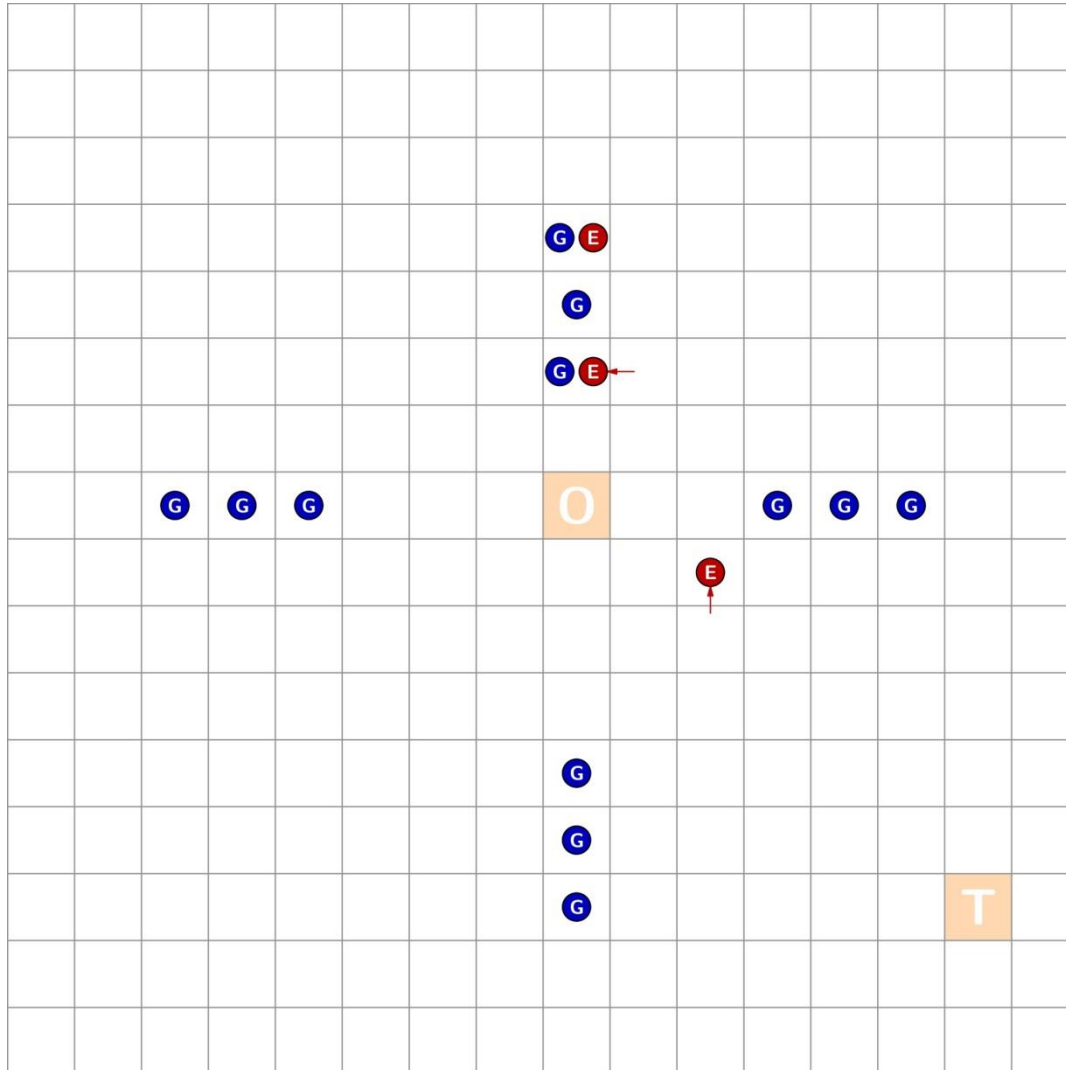
Diamond Search



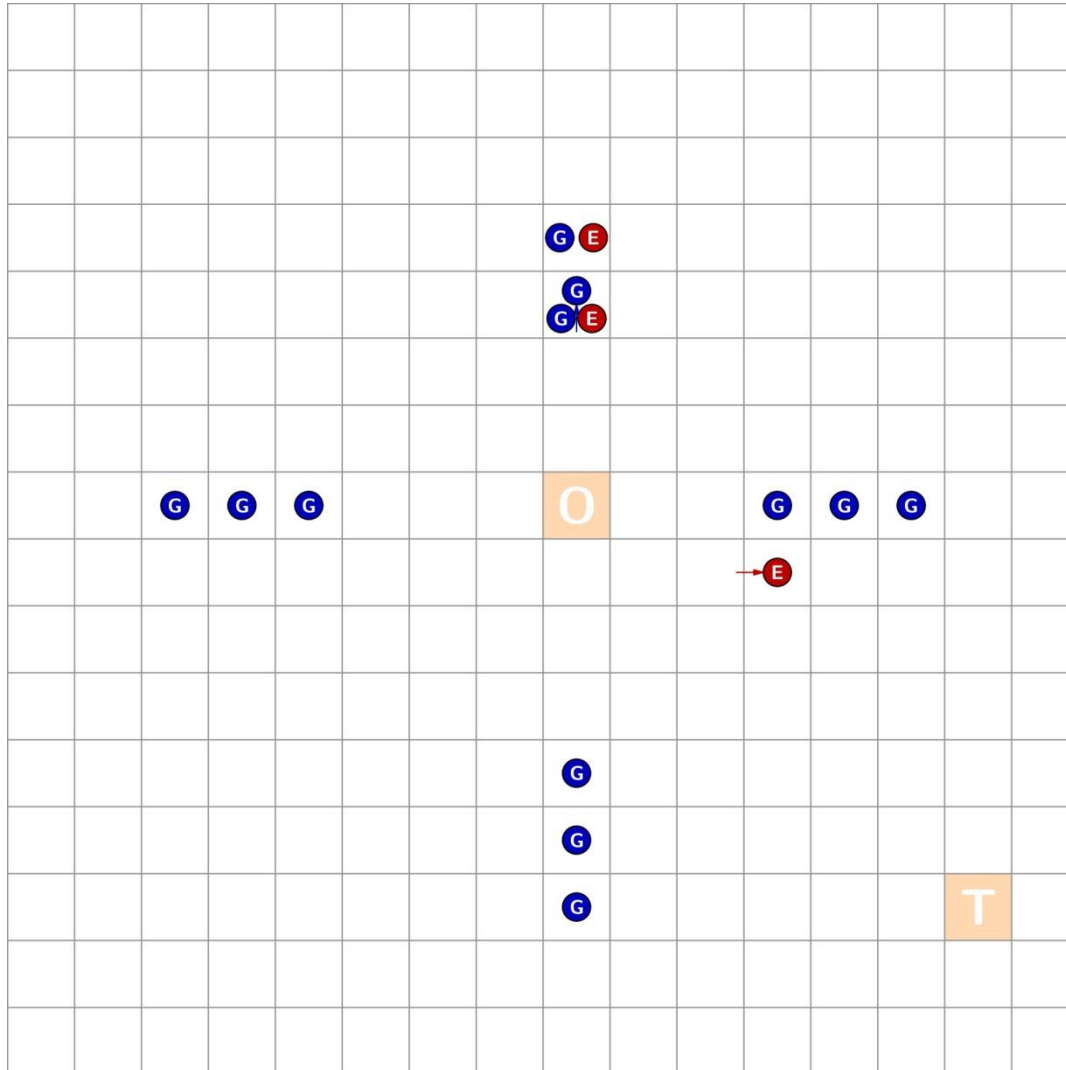
Diamond Search



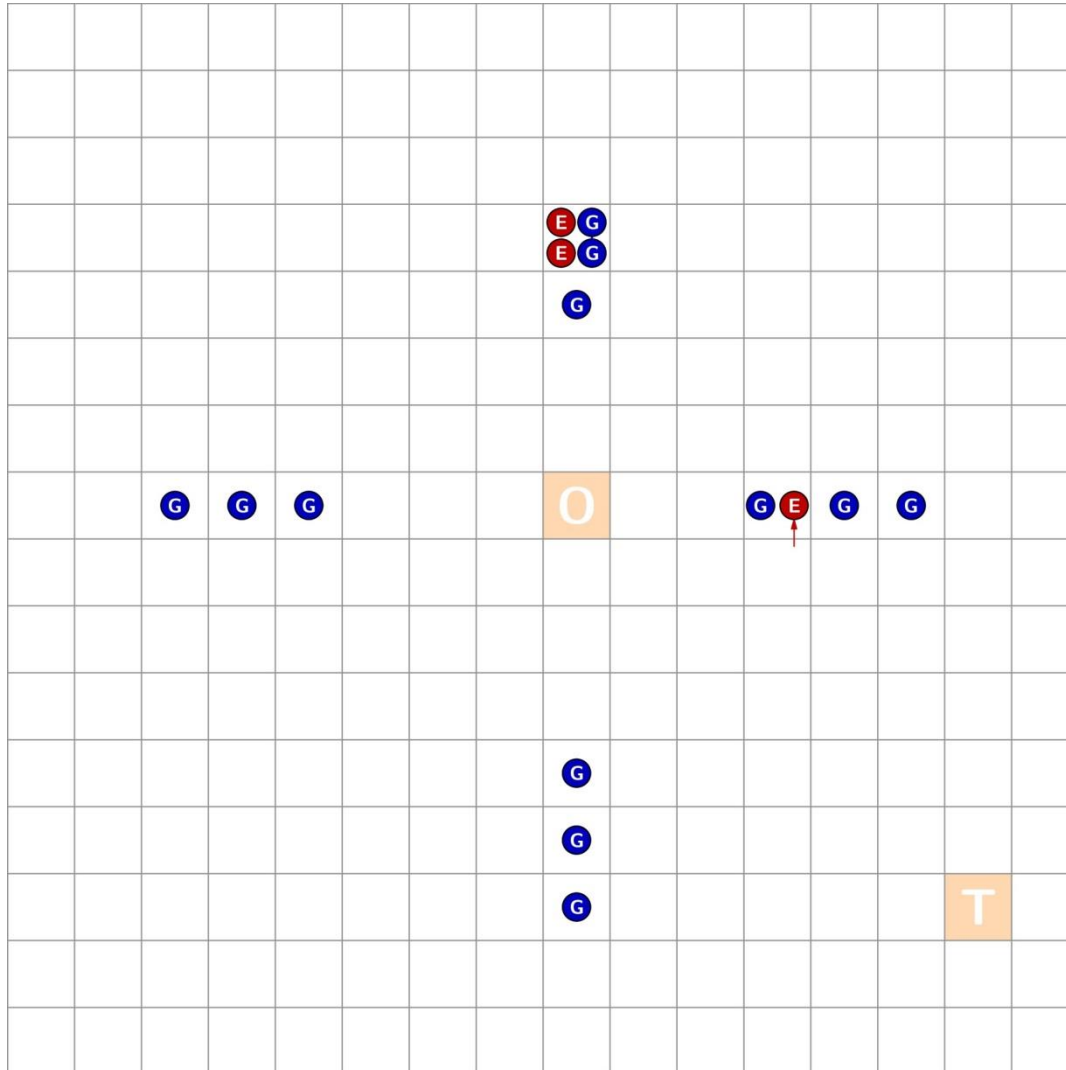
Diamond Search



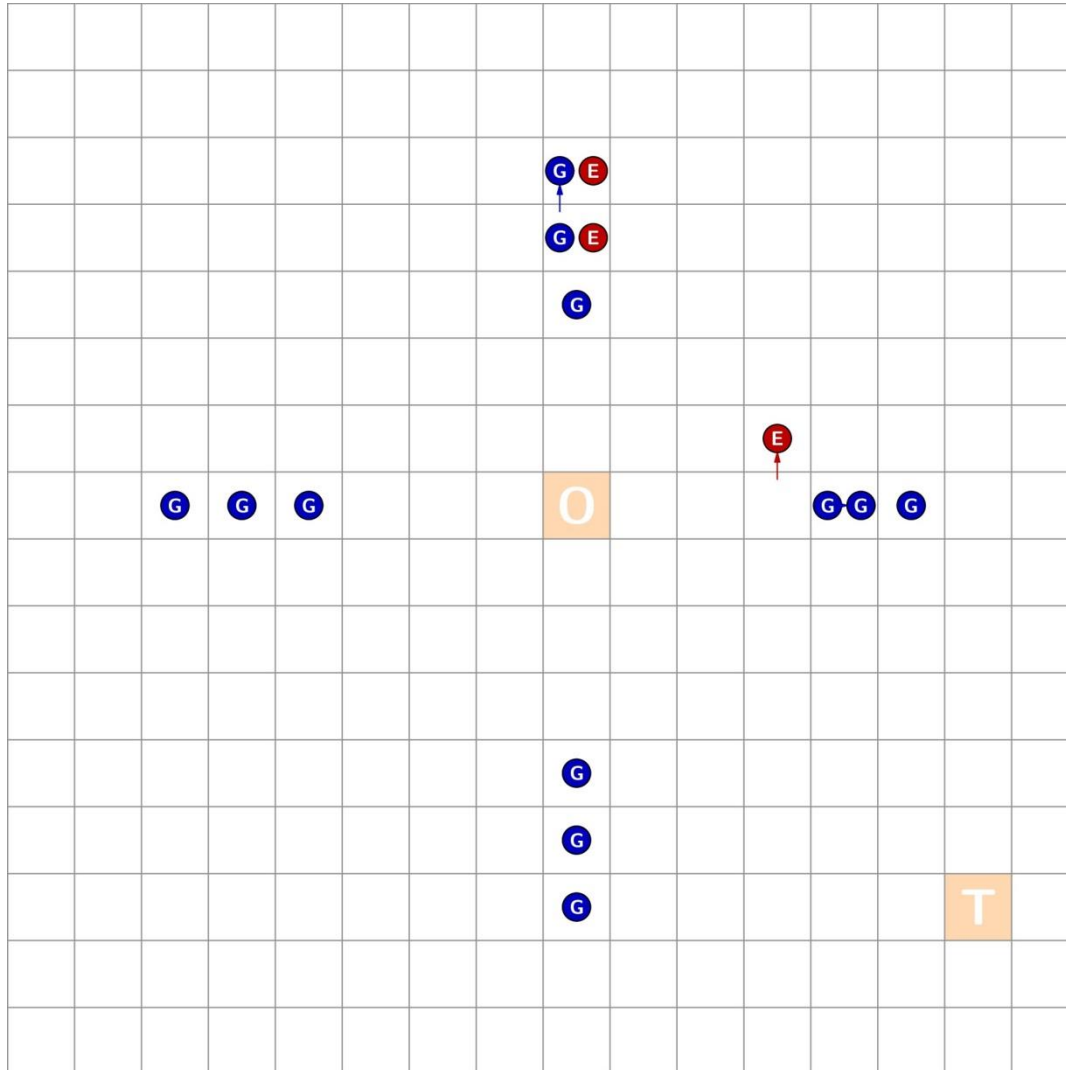
Diamond Search



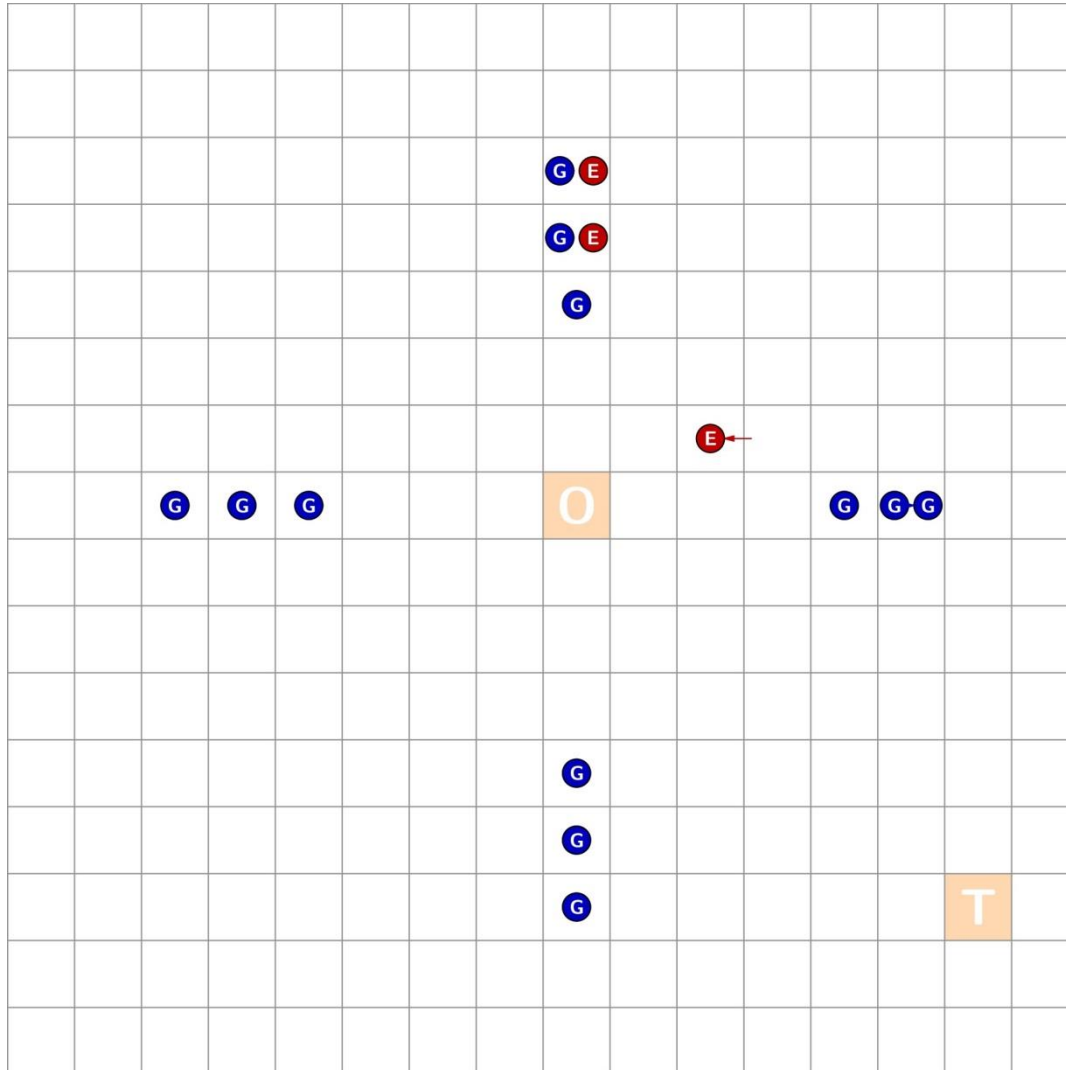
Diamond Search



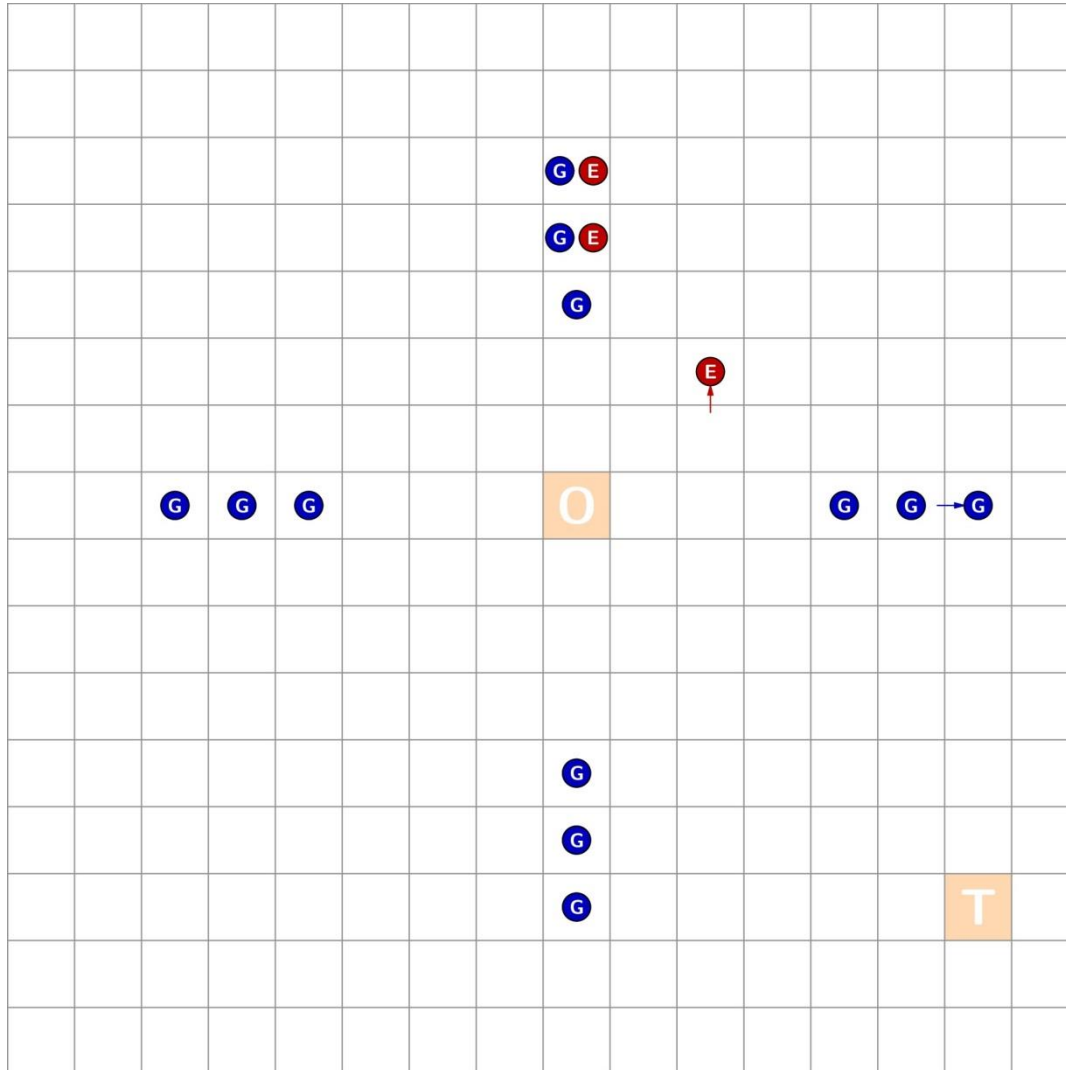
Diamond Search



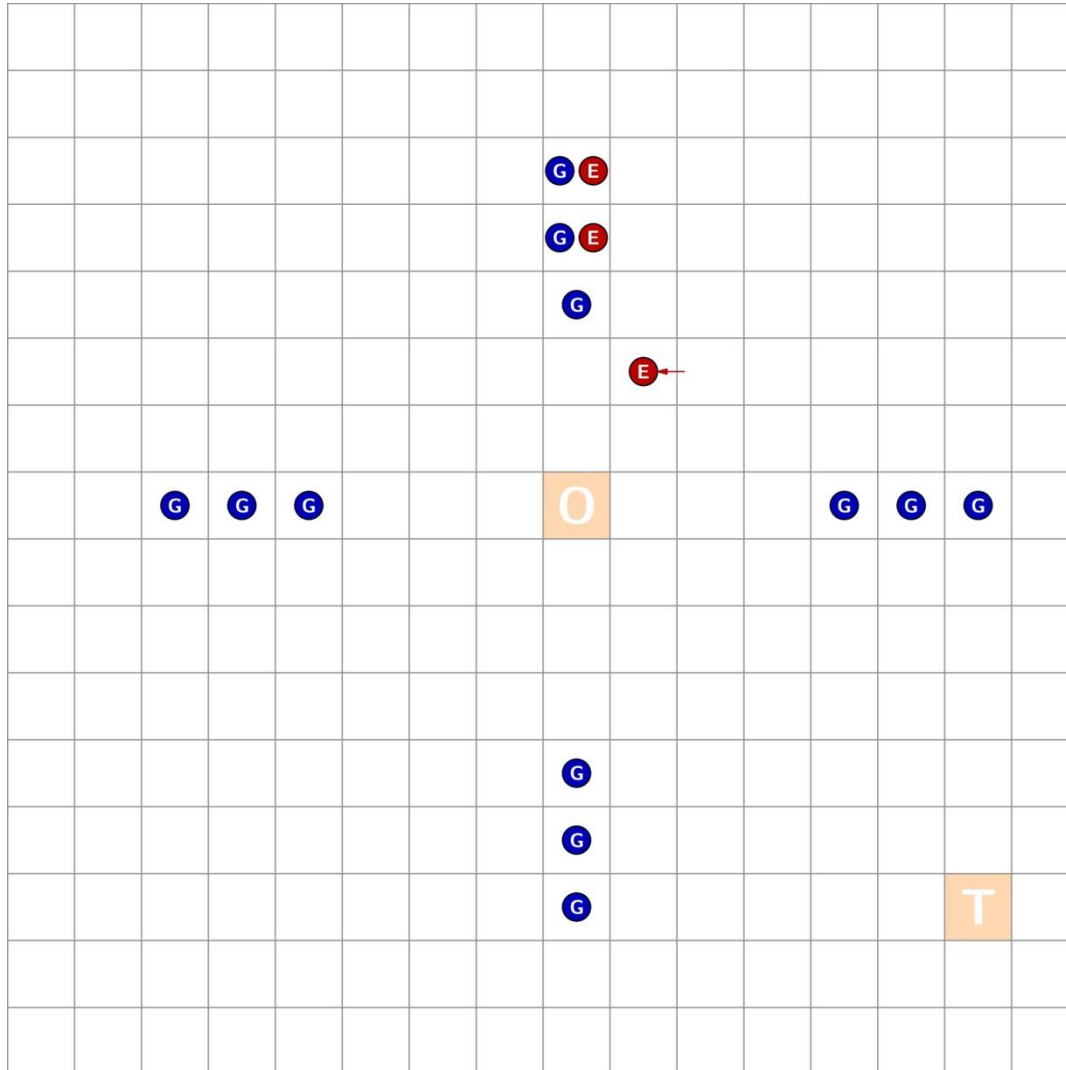
Diamond Search



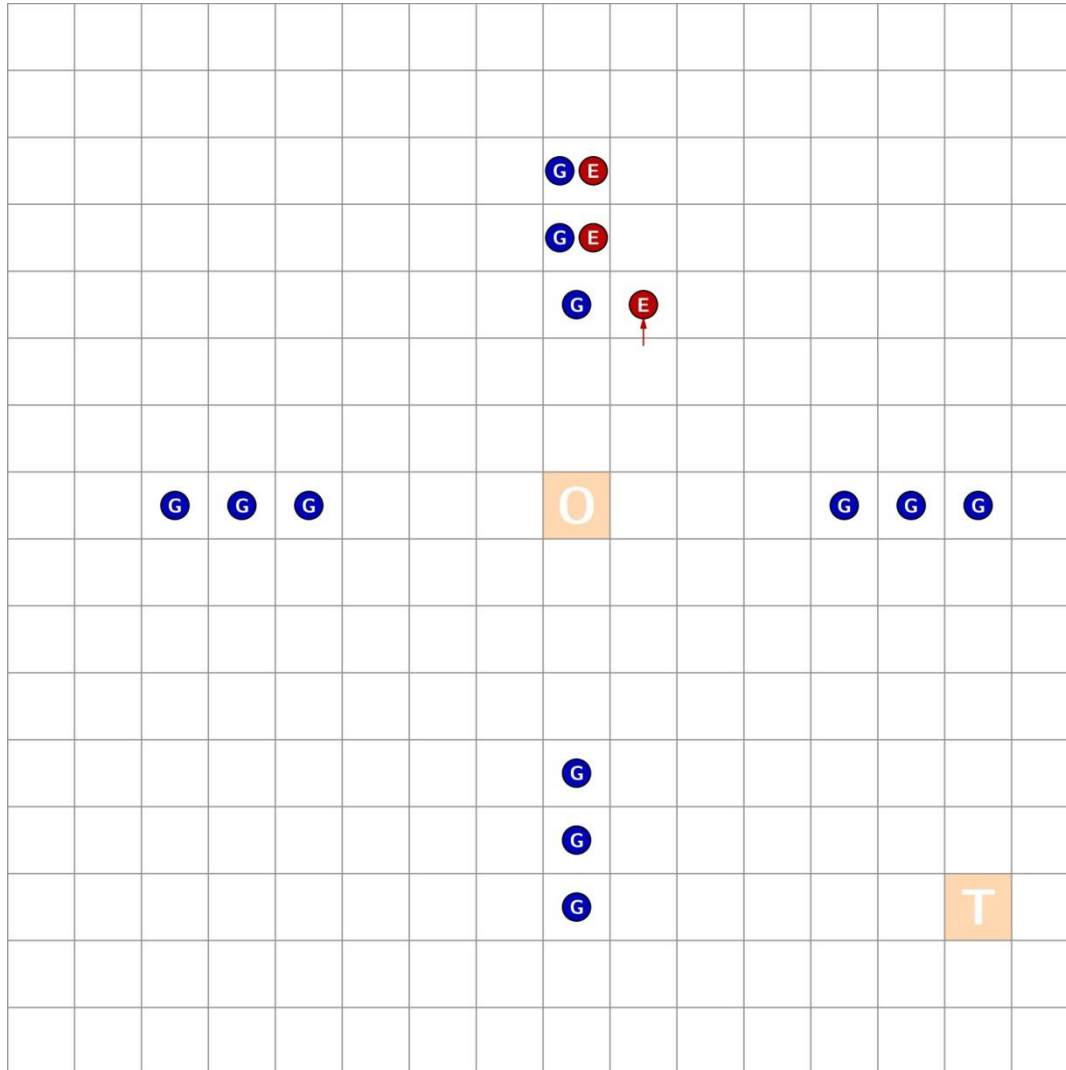
Diamond Search



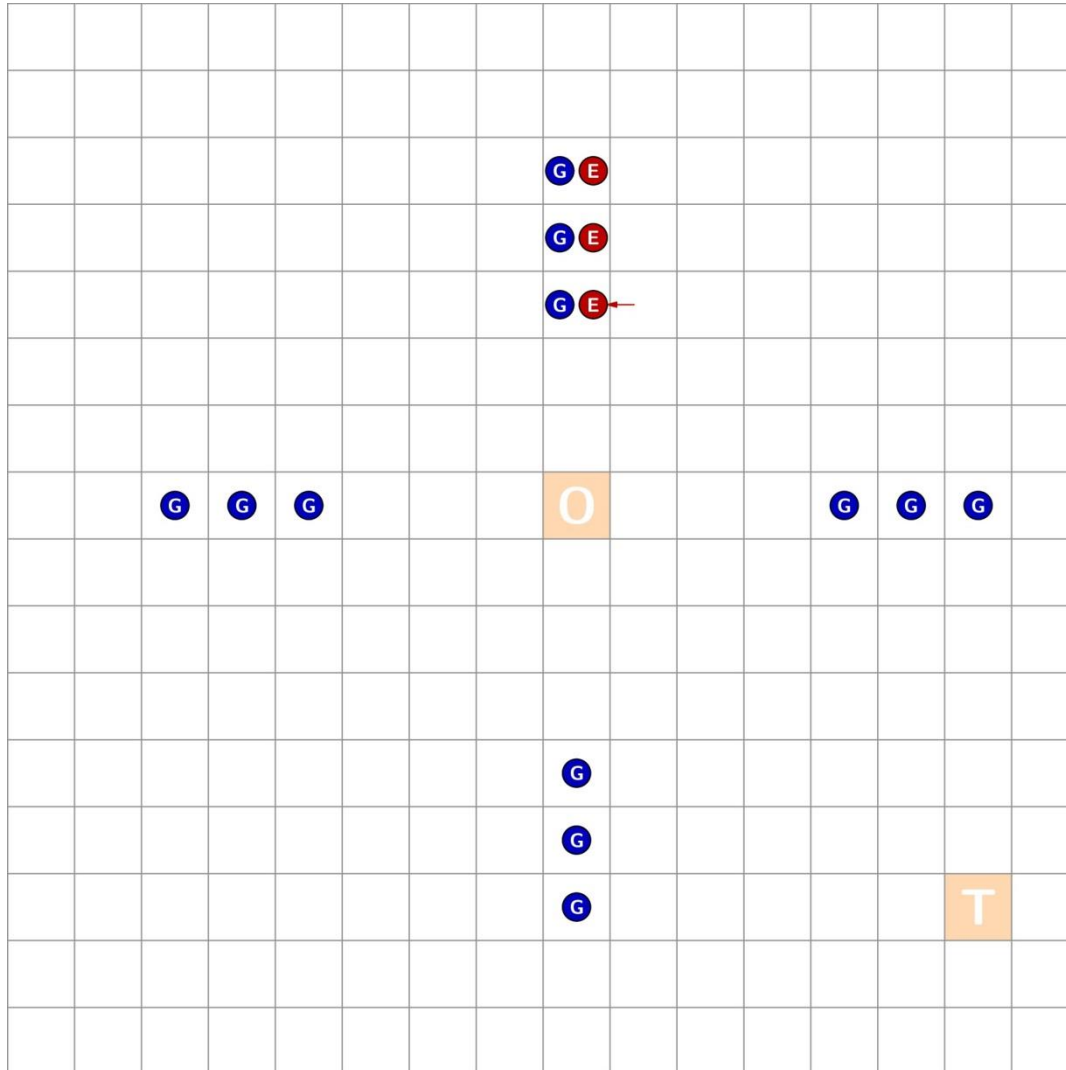
Diamond Search



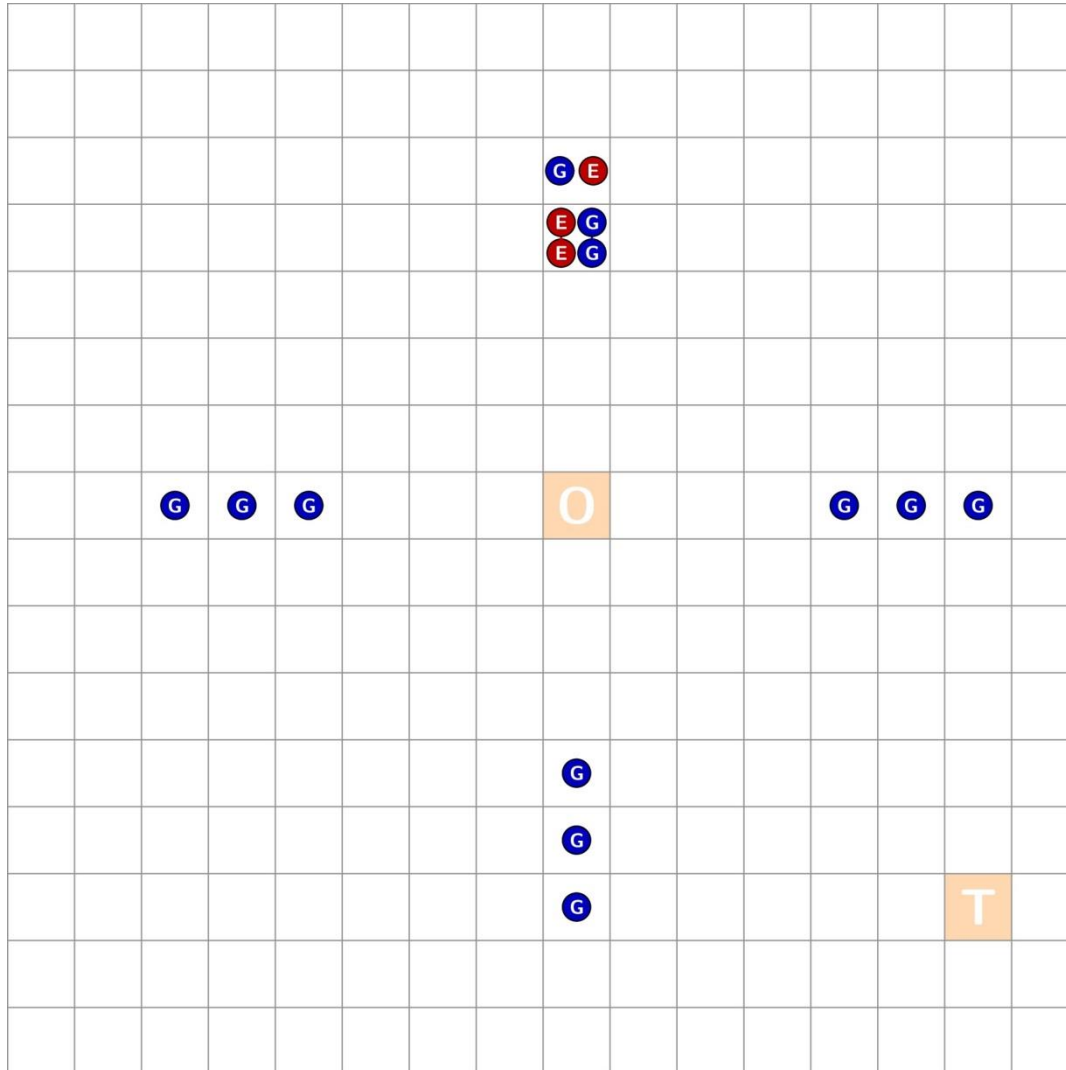
Diamond Search



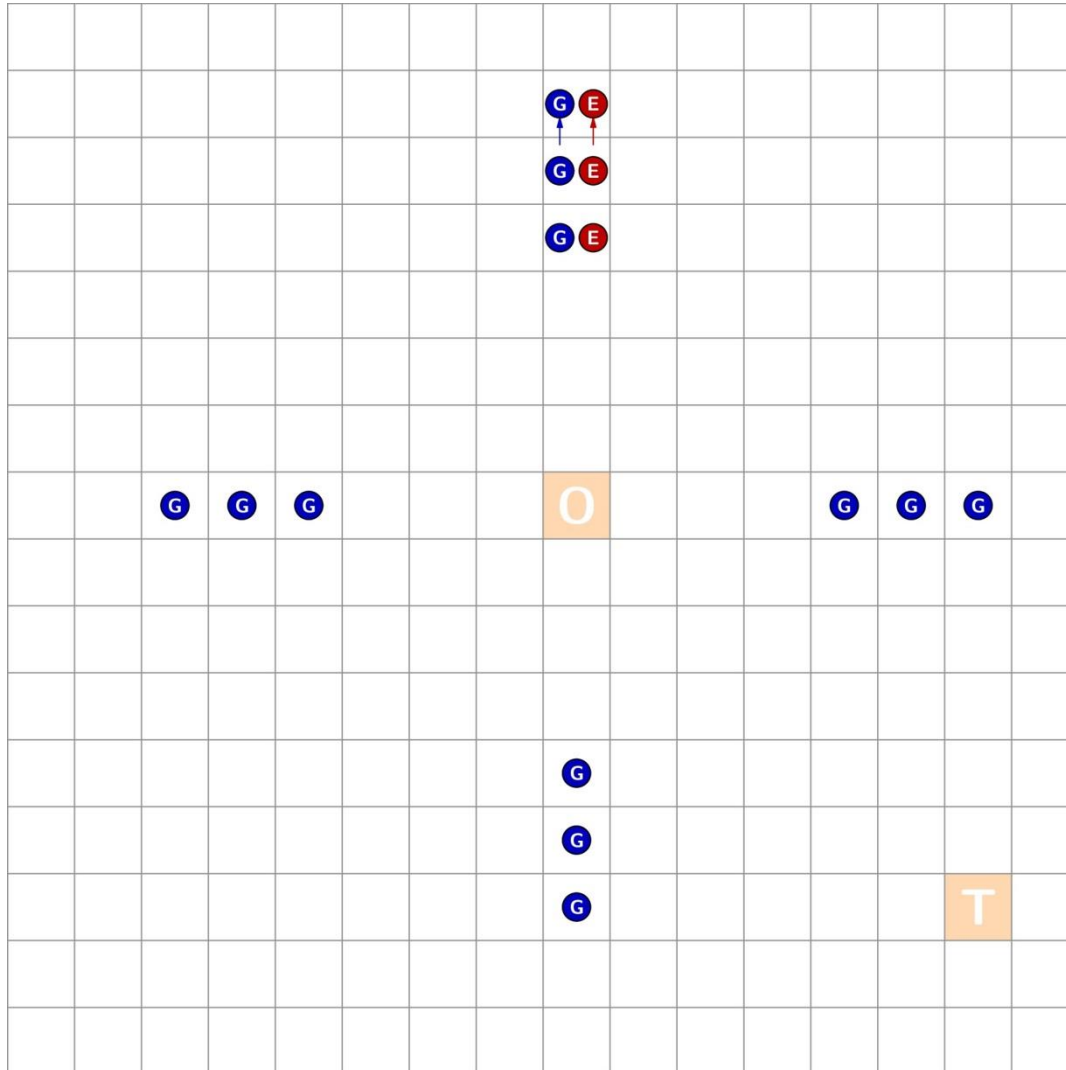
Diamond Search



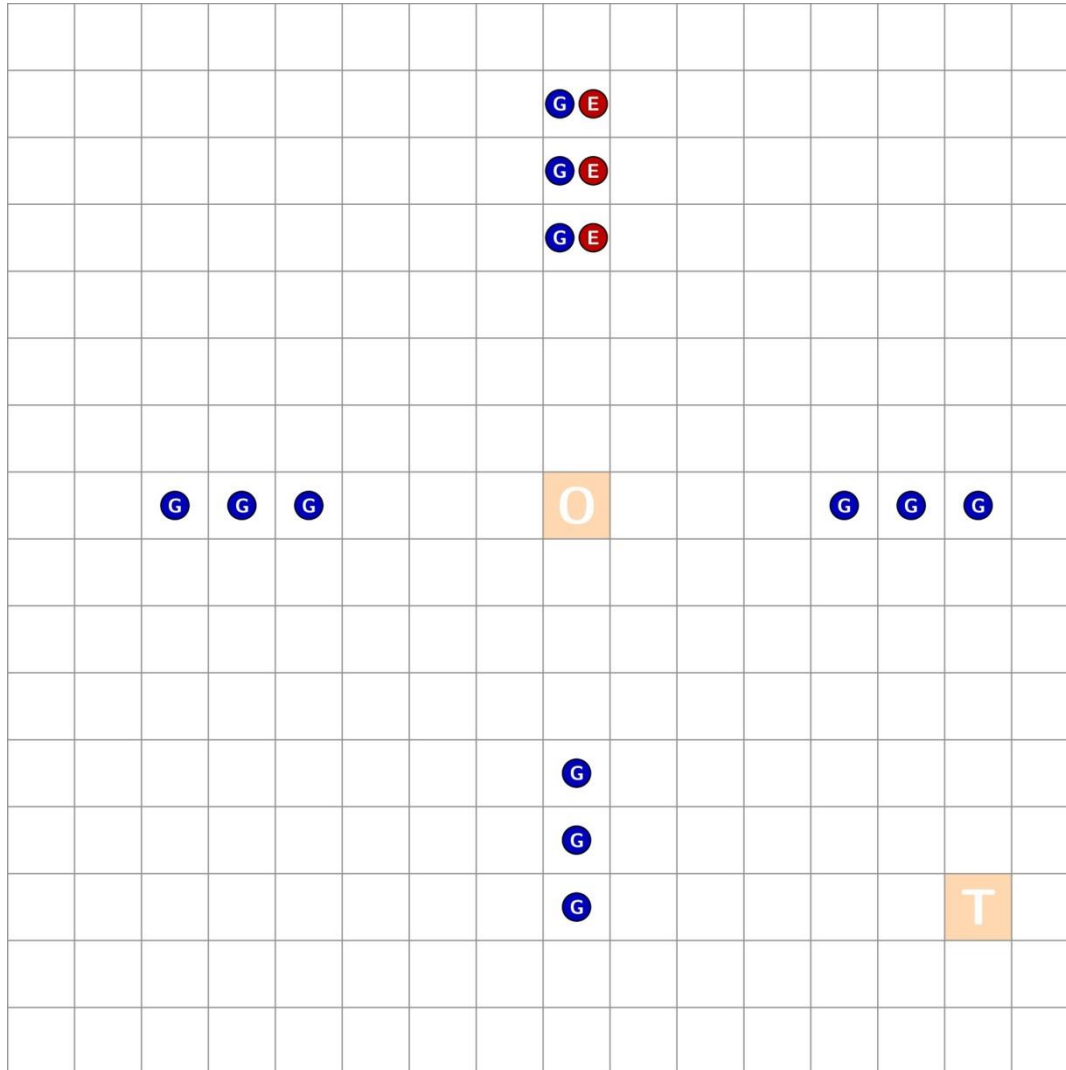
Diamond Search



Diamond Search



Diamond Search

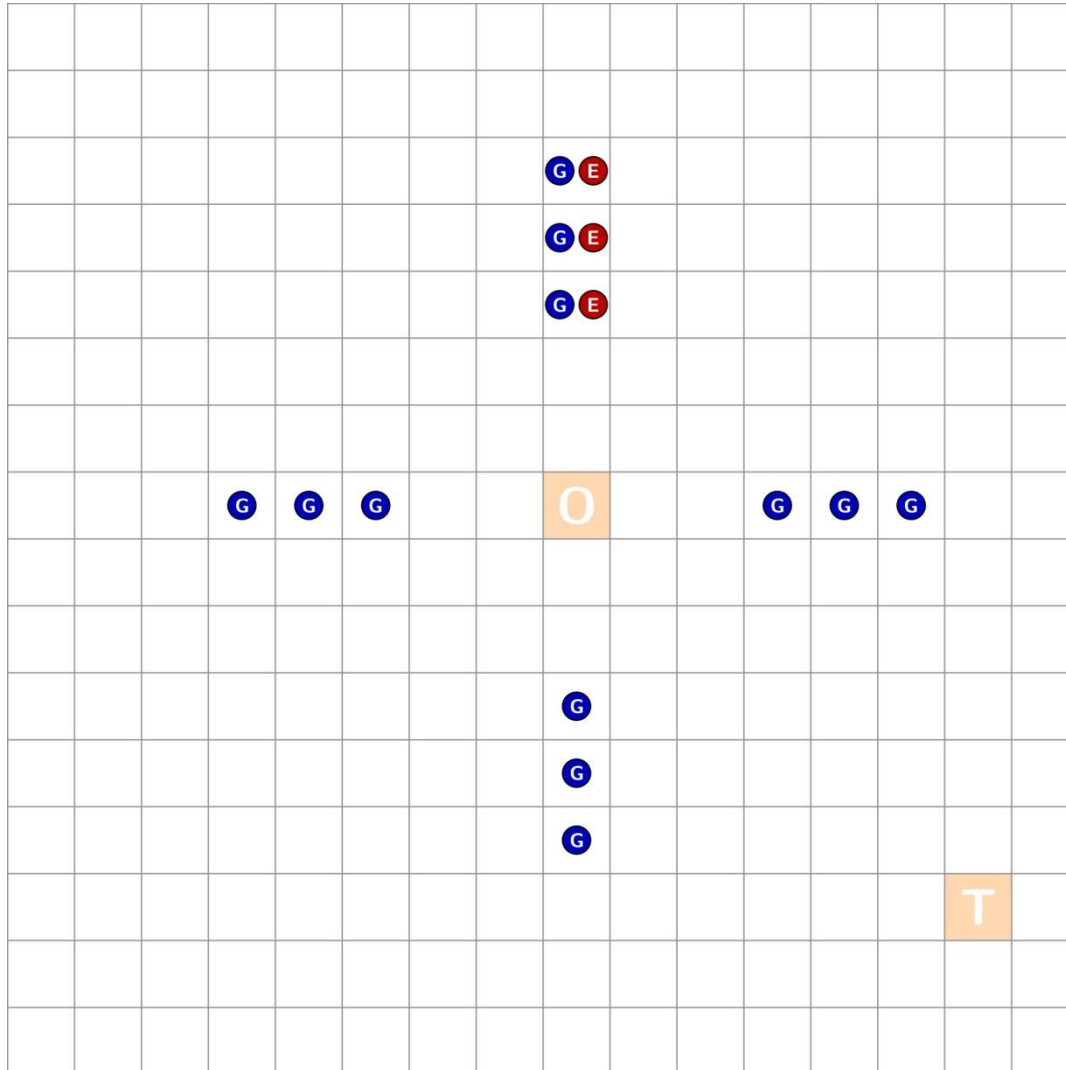


Diamond Search

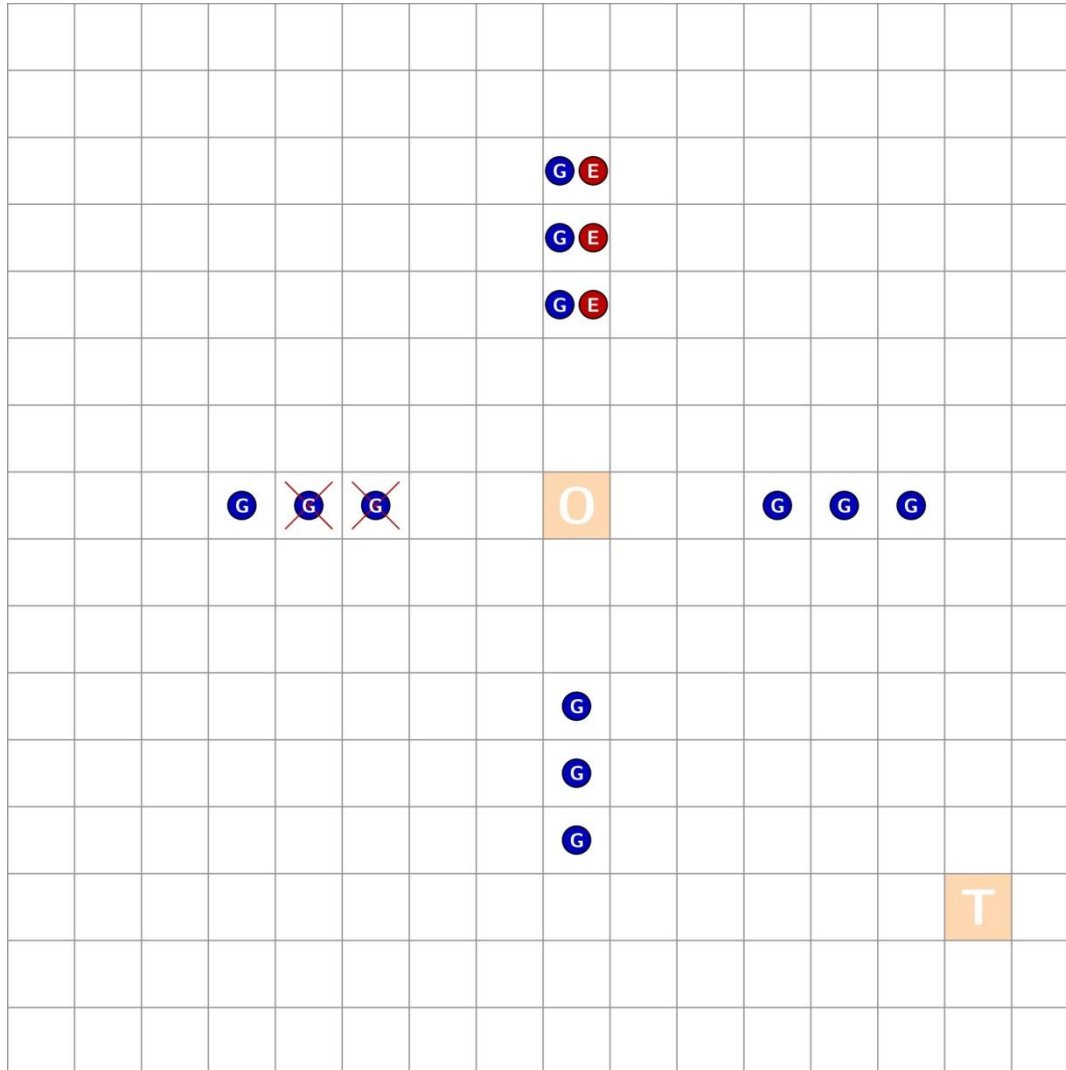
- Only works if ants are robust.
- What if the ants may fail?



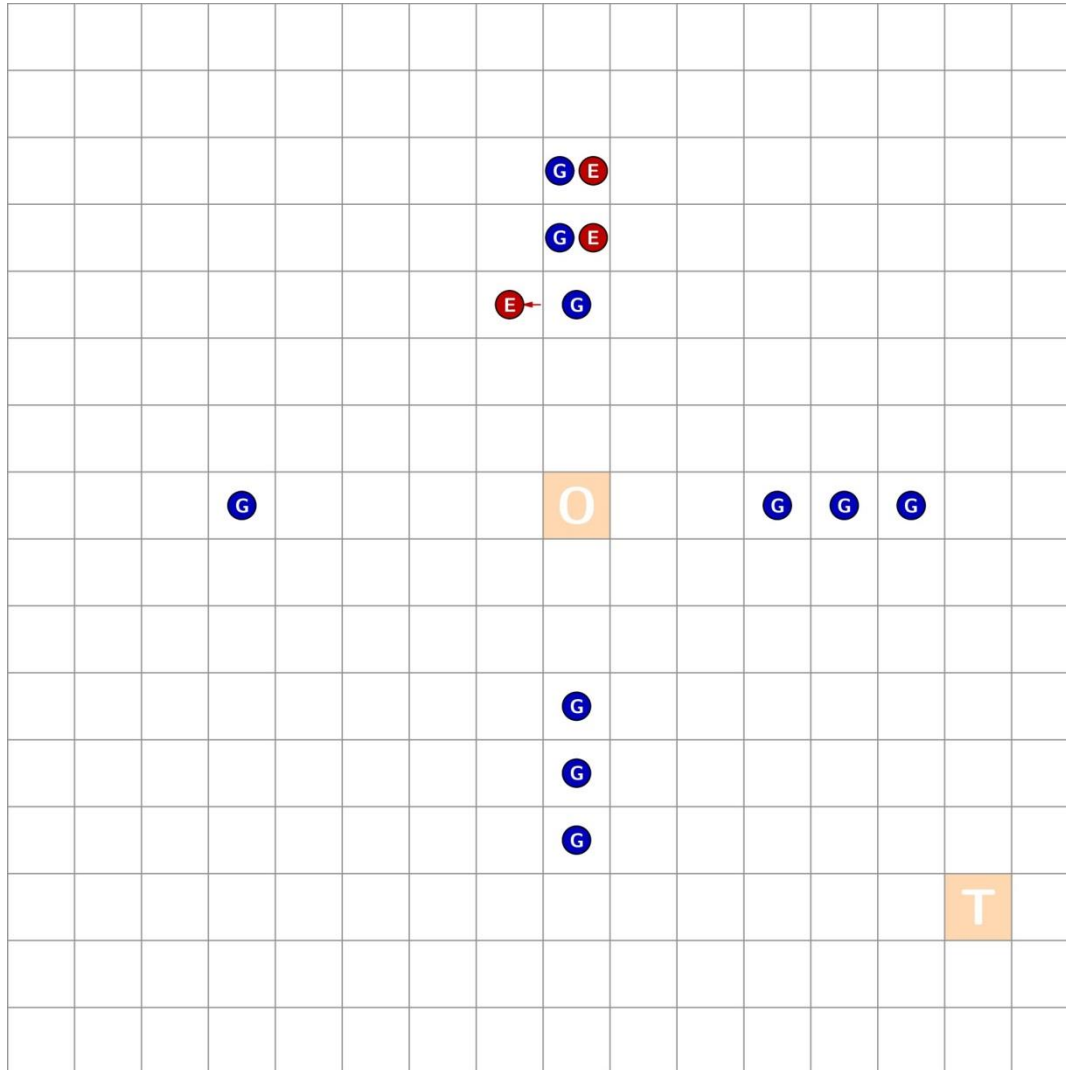
Diamond Search - Issues



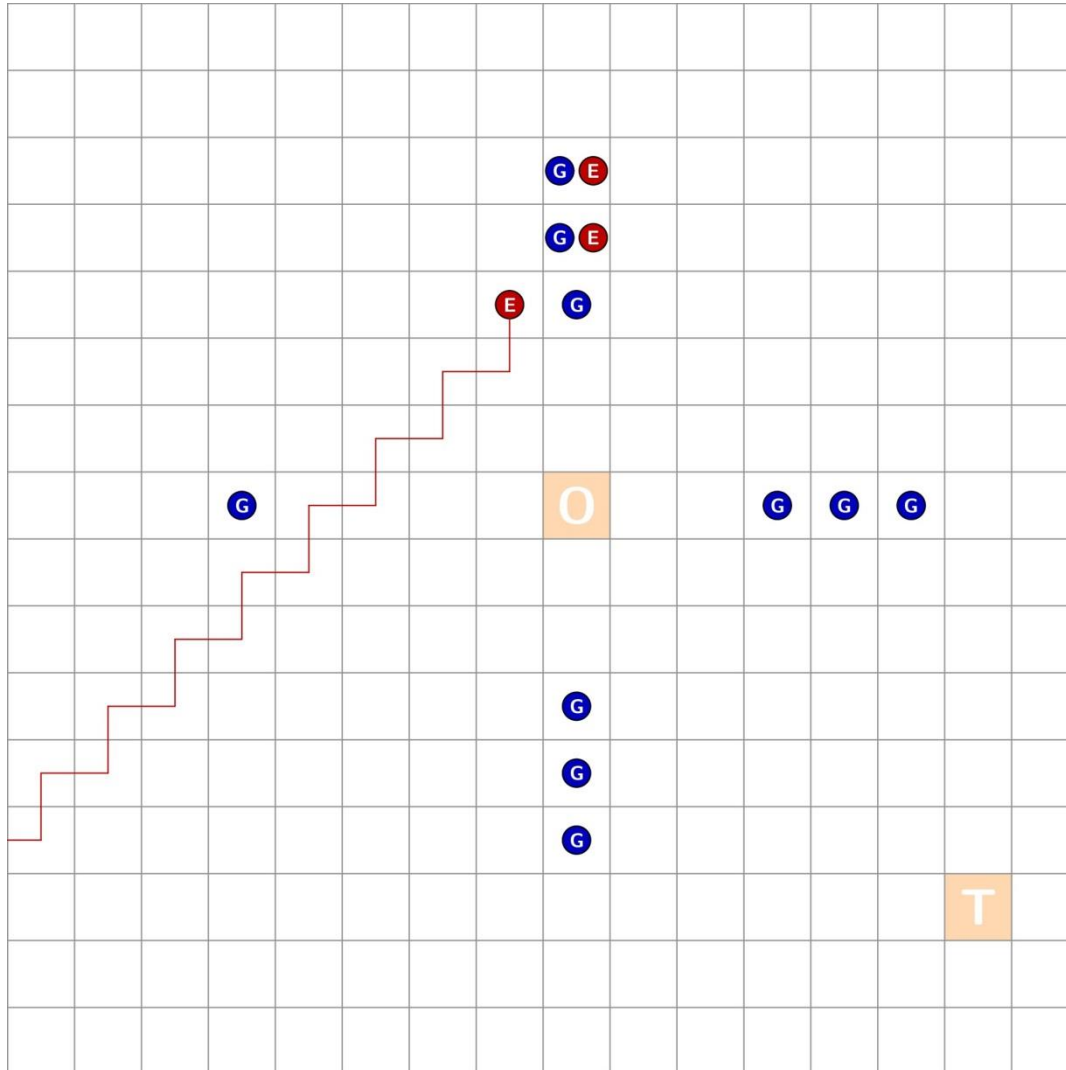
Diamond Search - Issues



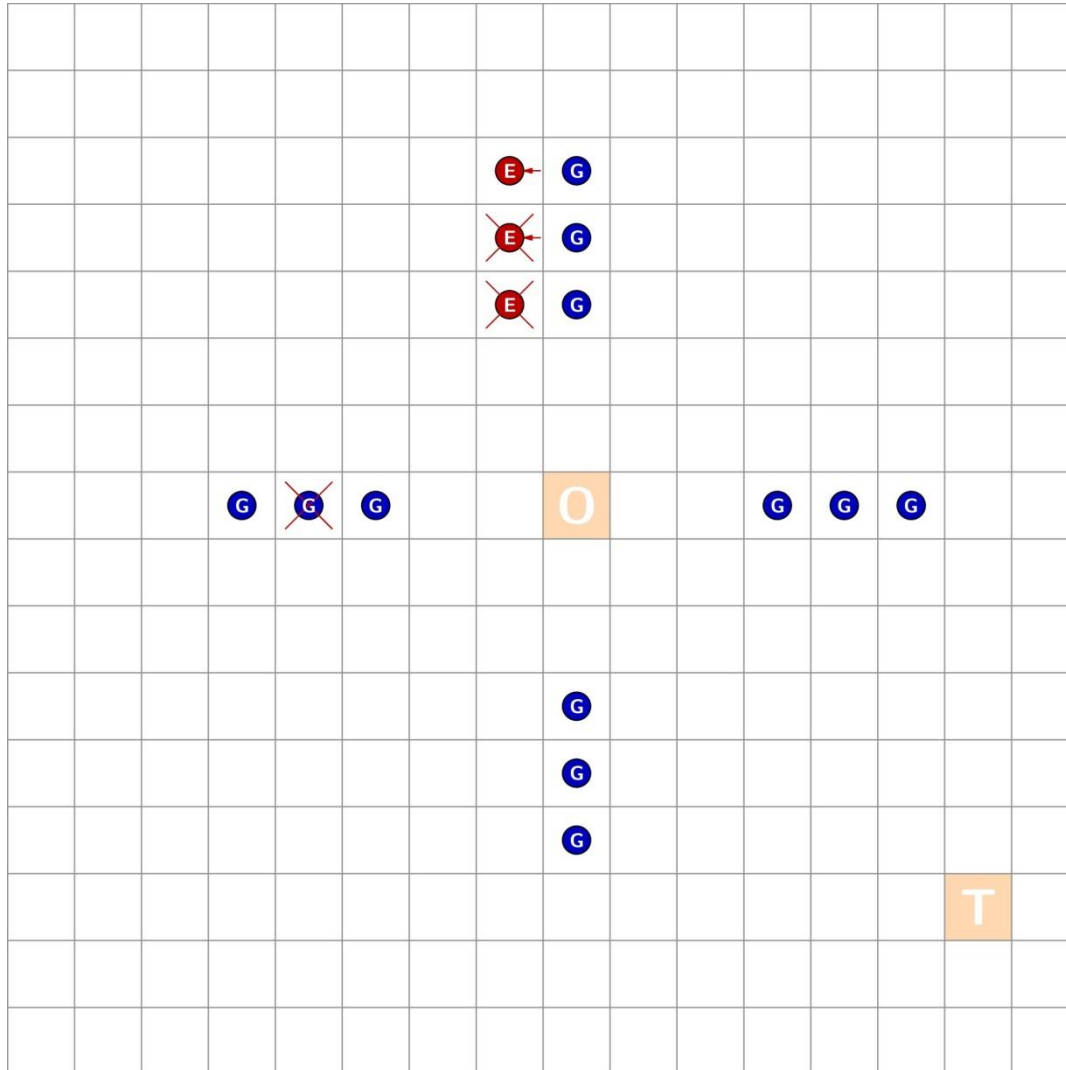
Diamond Search - Issues



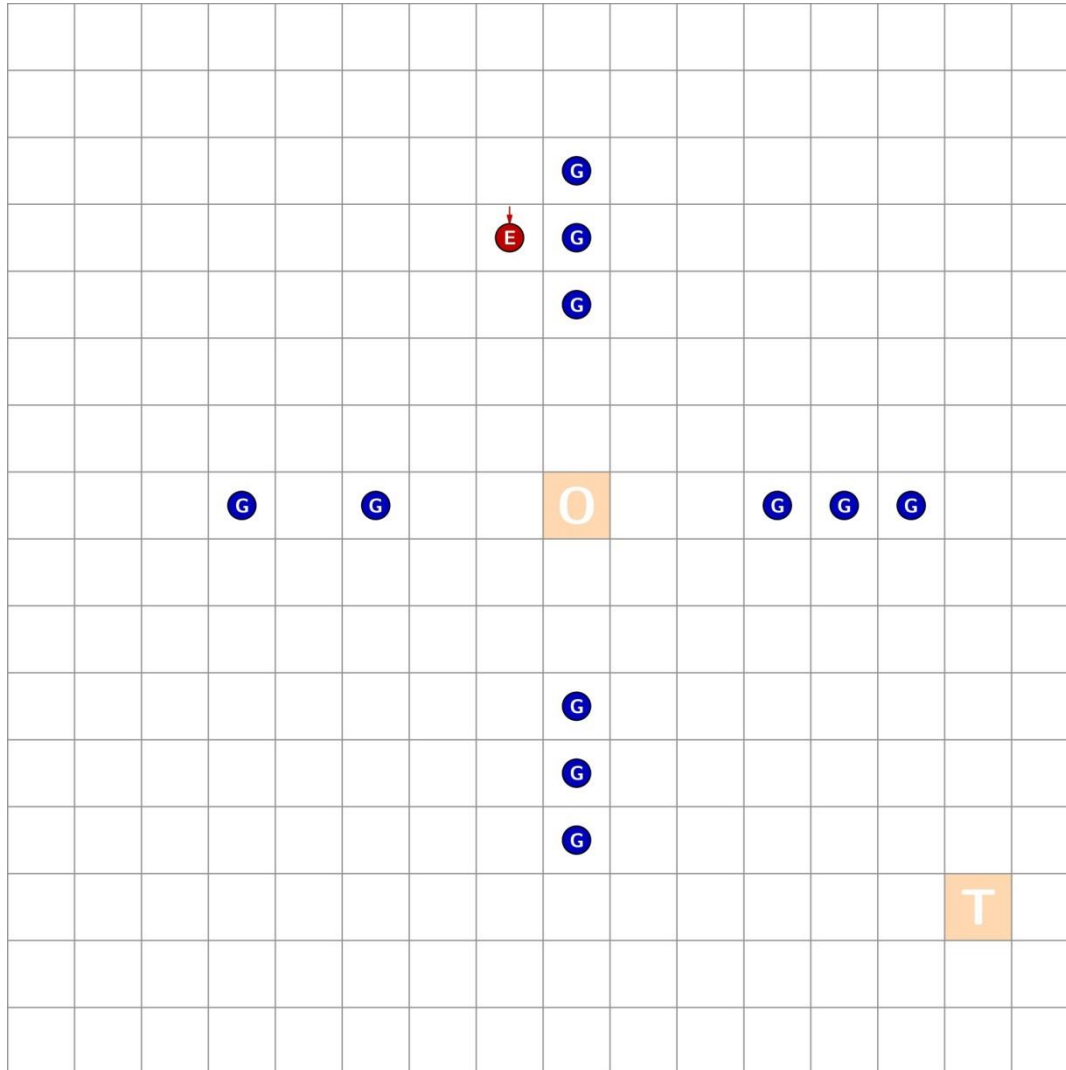
Diamond Search - Issues



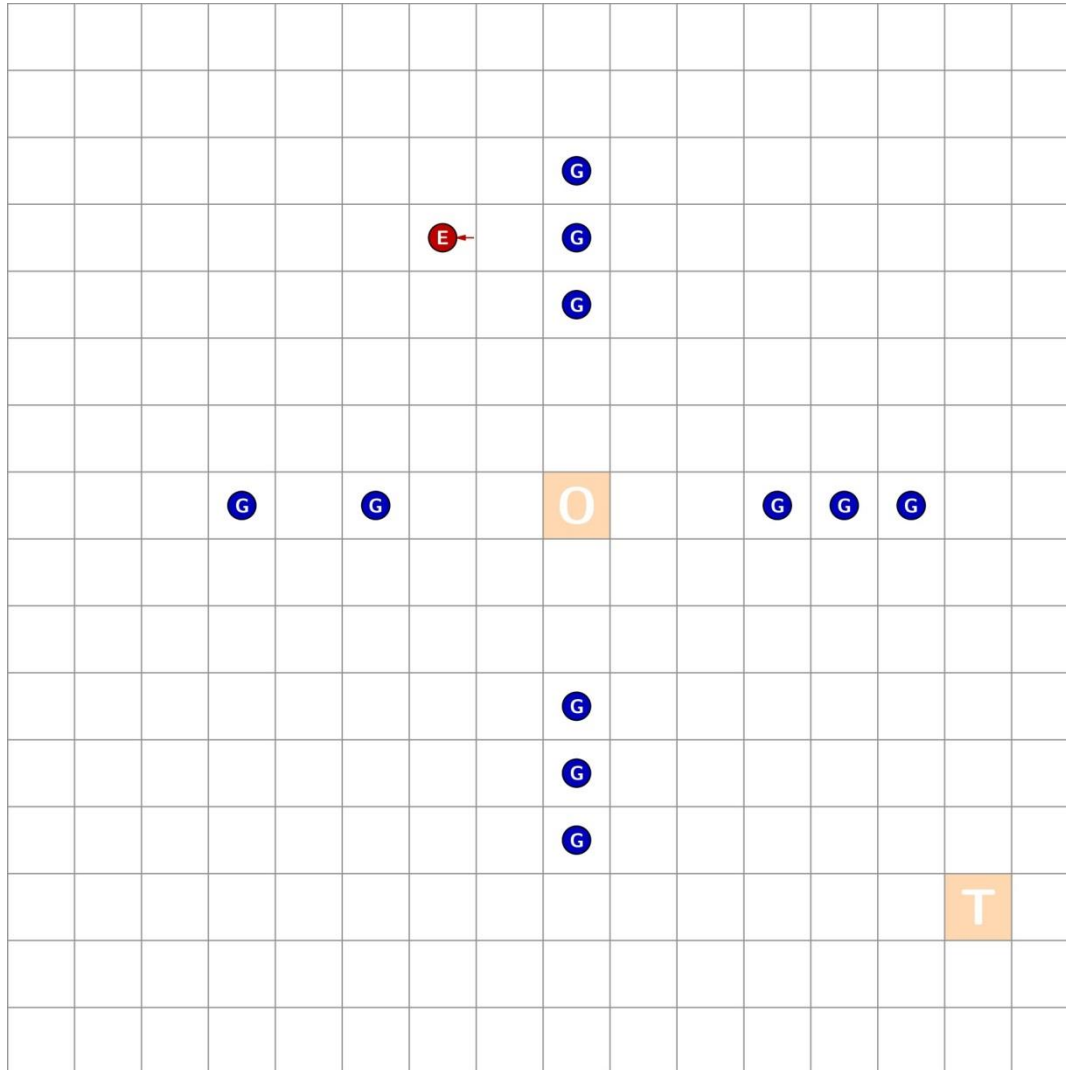
Diamond Search - Issues



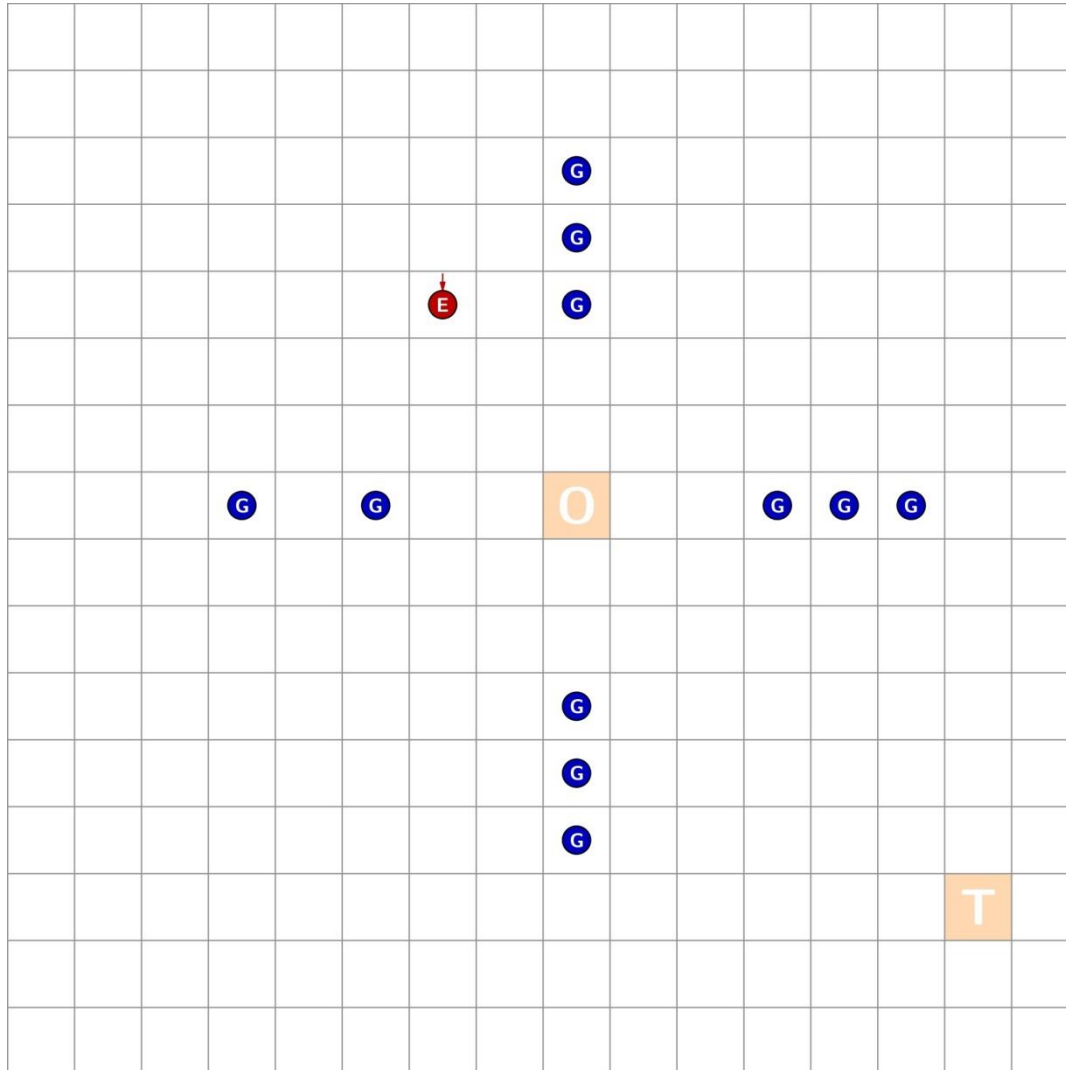
Diamond Search - Issues



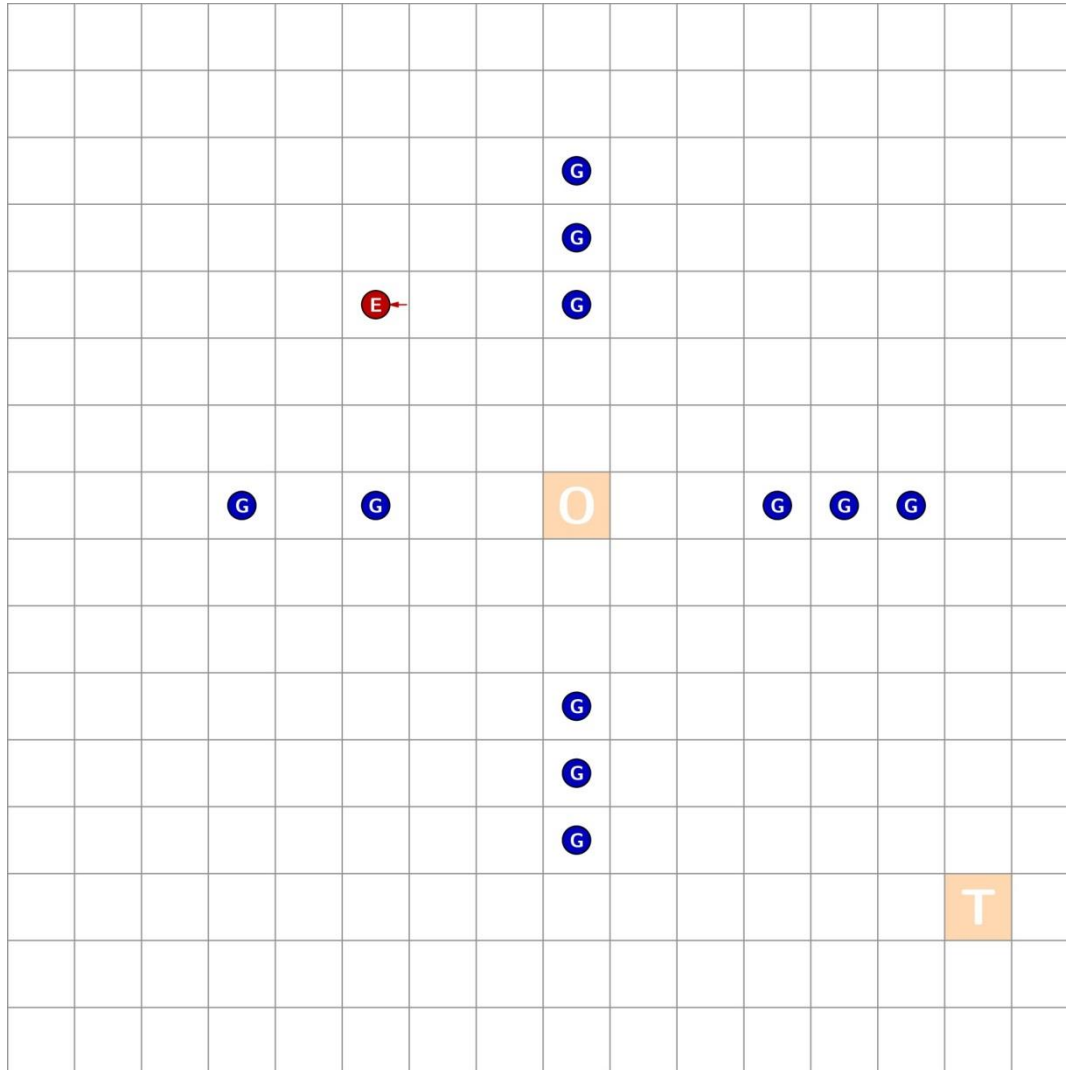
Diamond Search - Issues



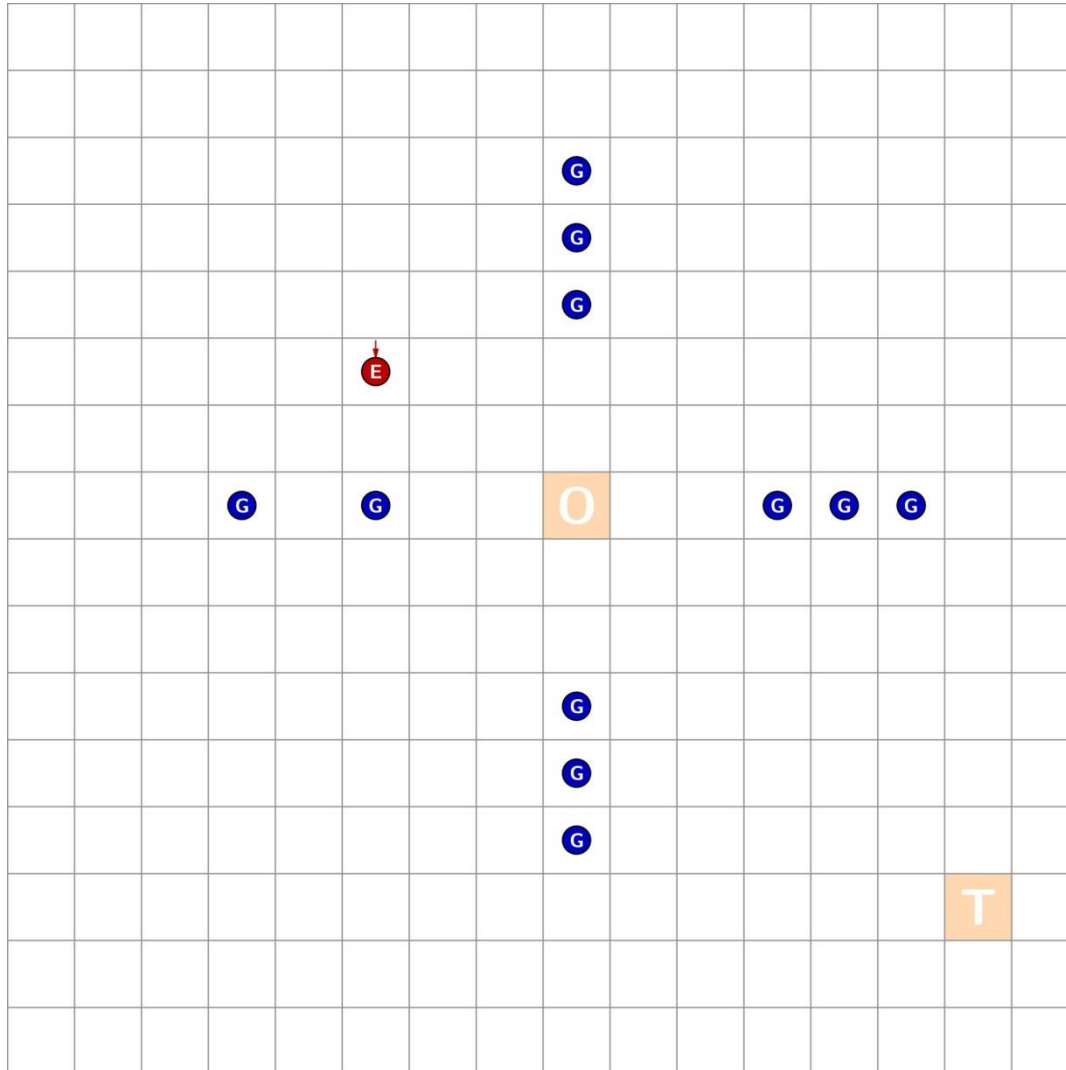
Diamond Search - Issues



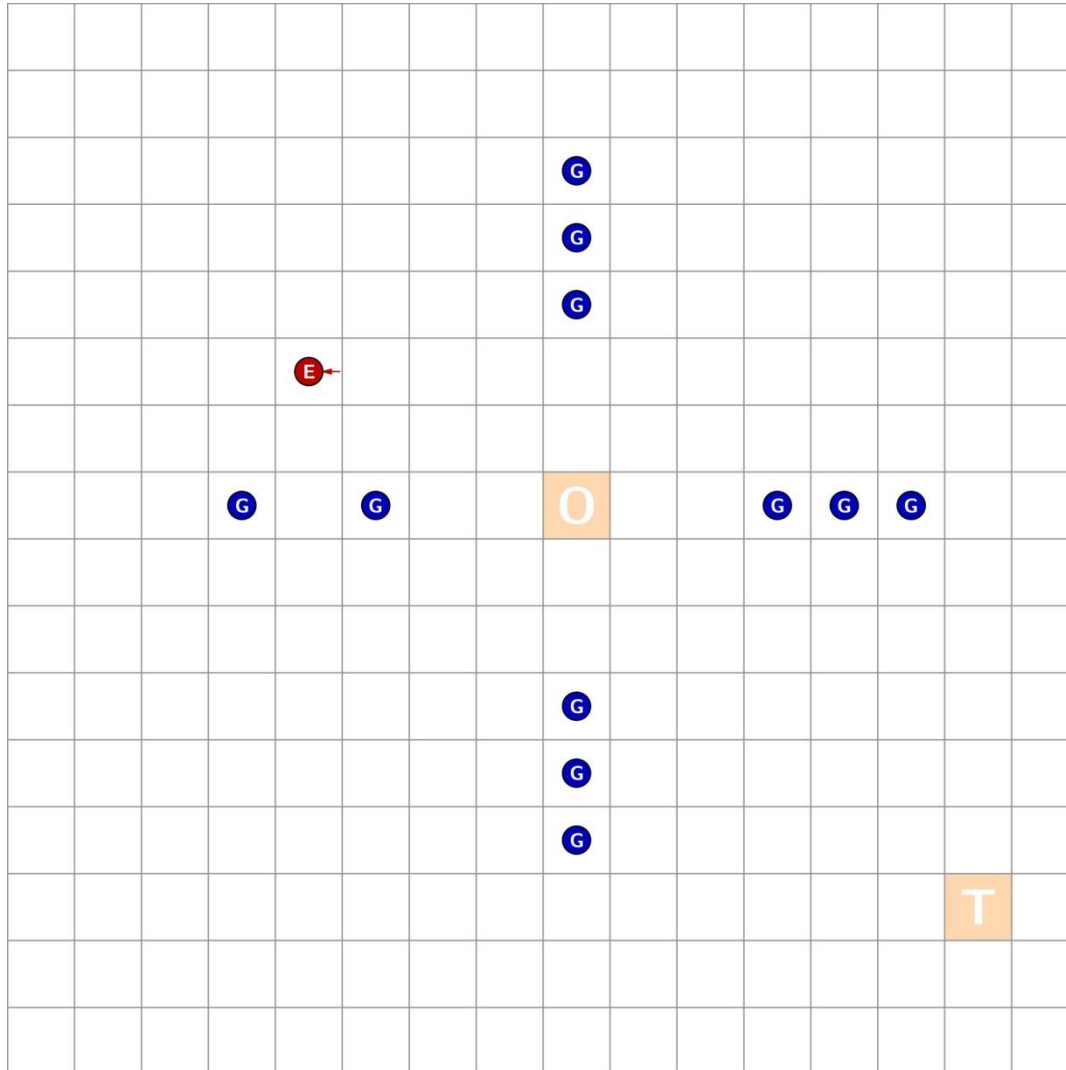
Diamond Search - Issues



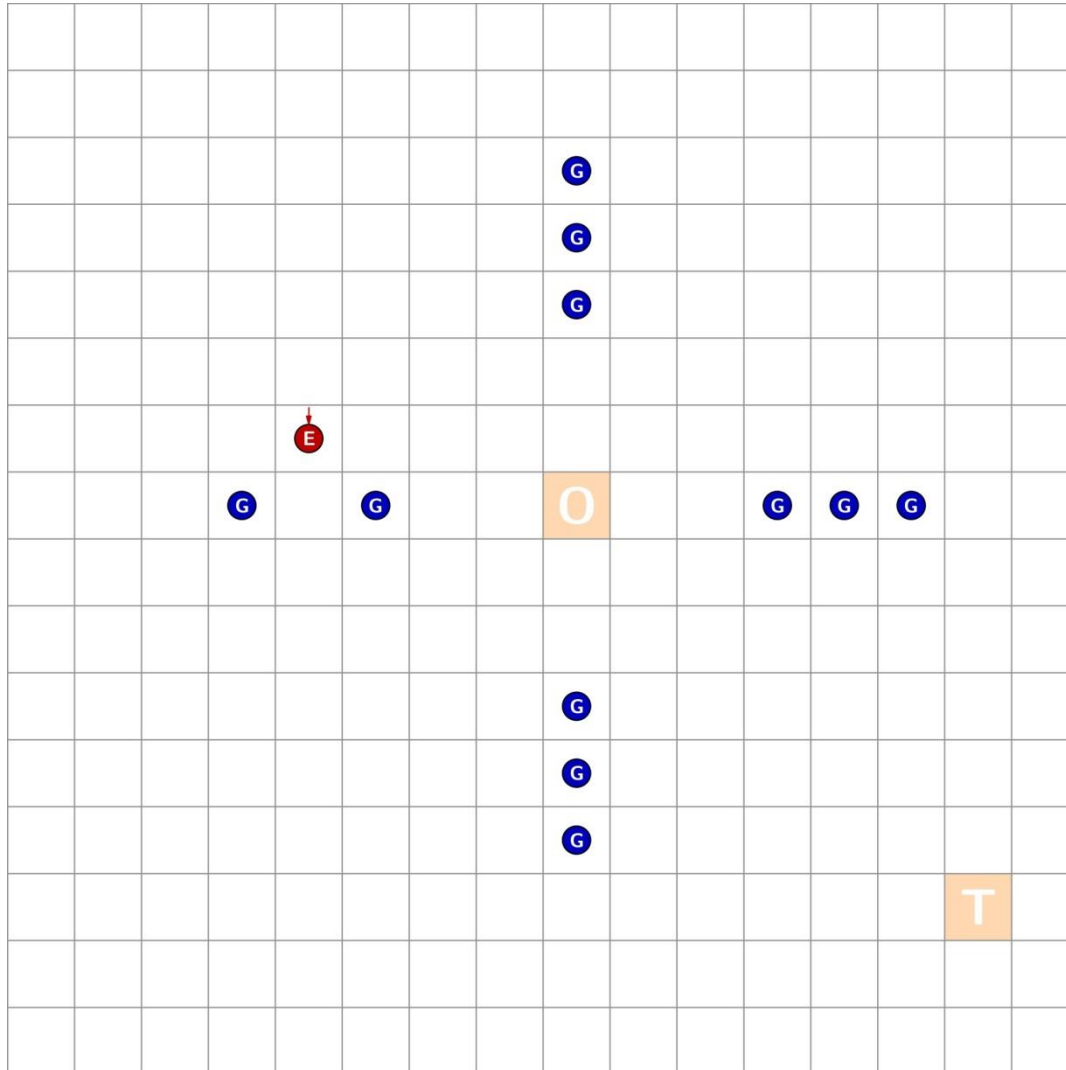
Diamond Search - Issues



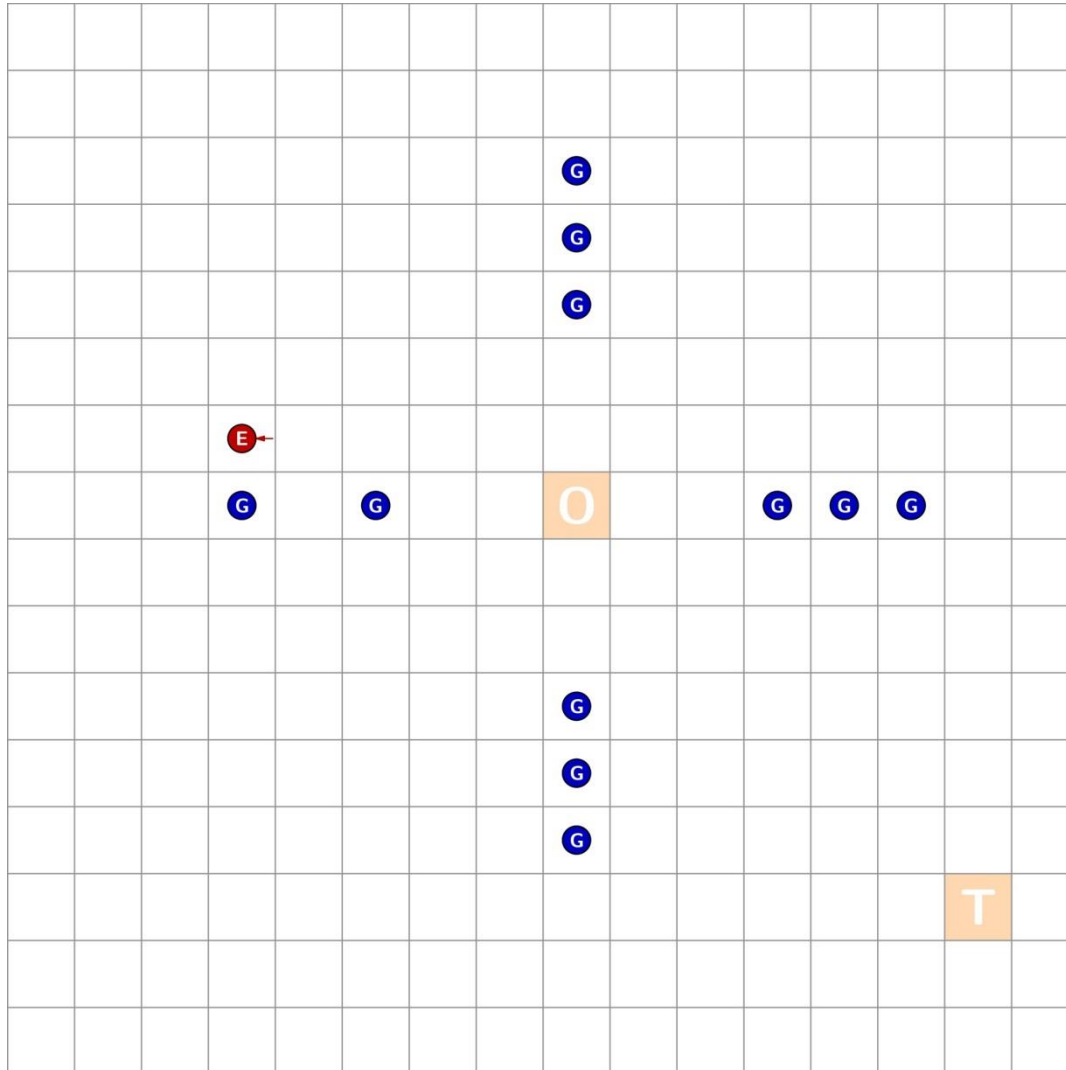
Diamond Search - Issues



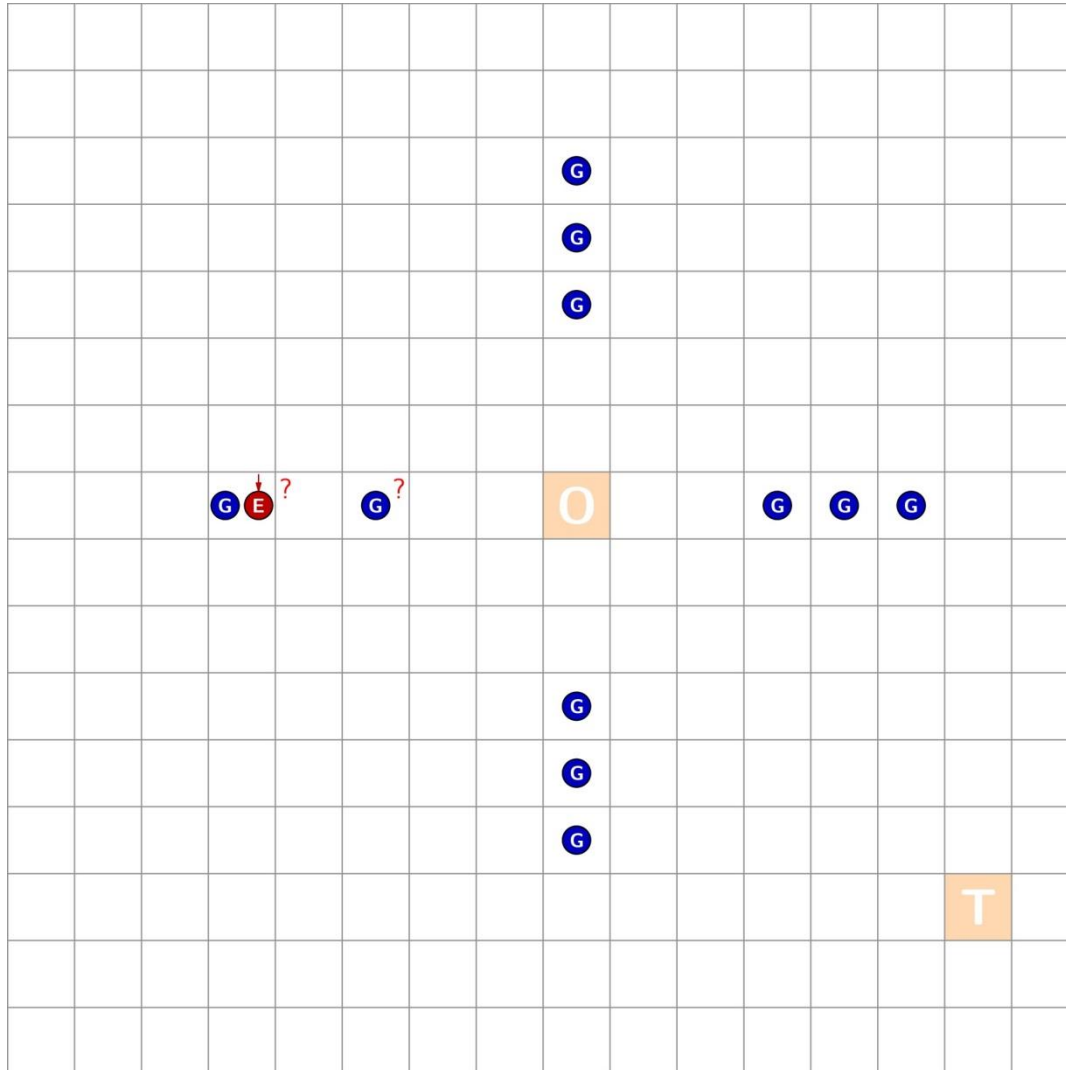
Diamond Search - Issues



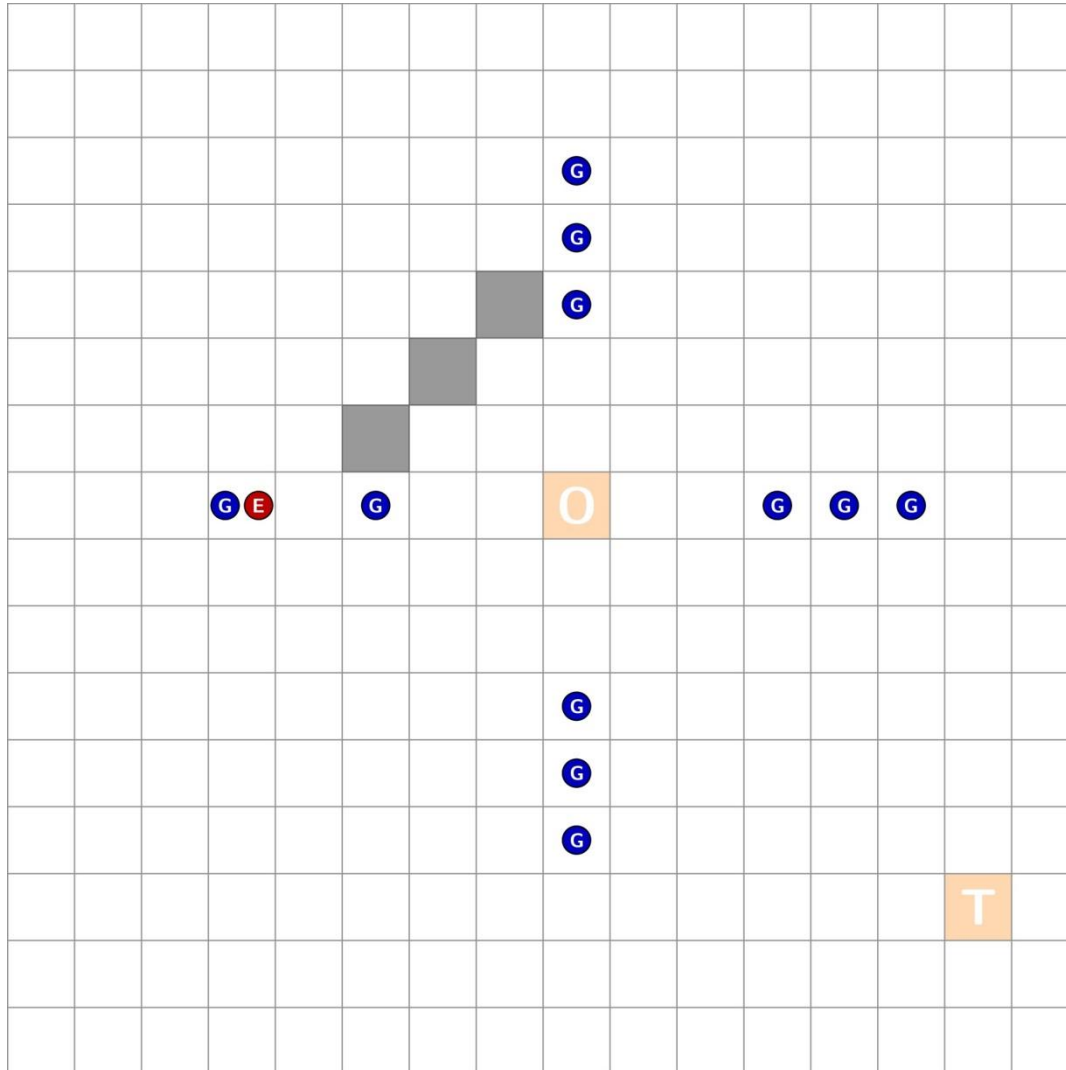
Diamond Search - Issues



Diamond Search - Issues



Diamond Search - Issues

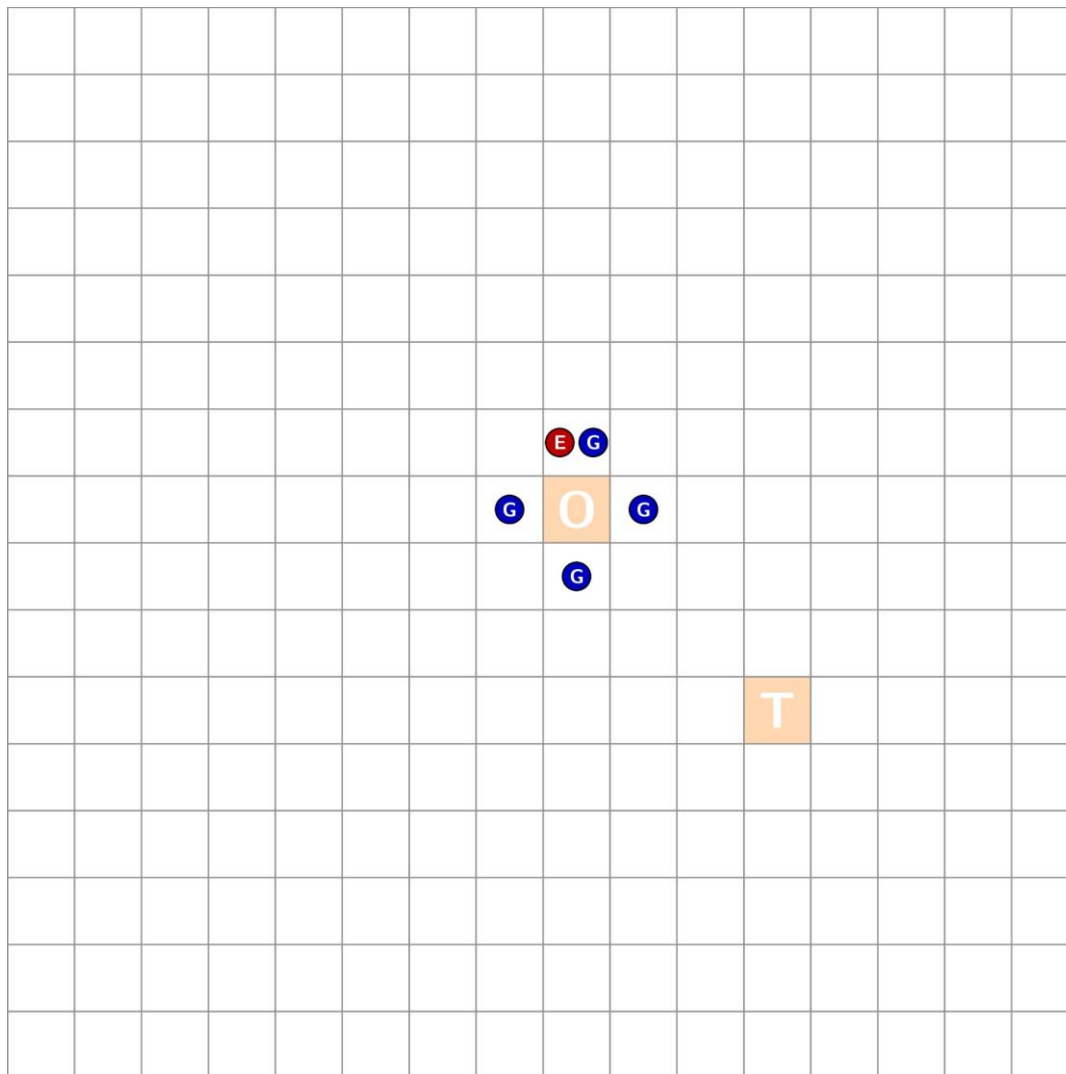


Robust Search

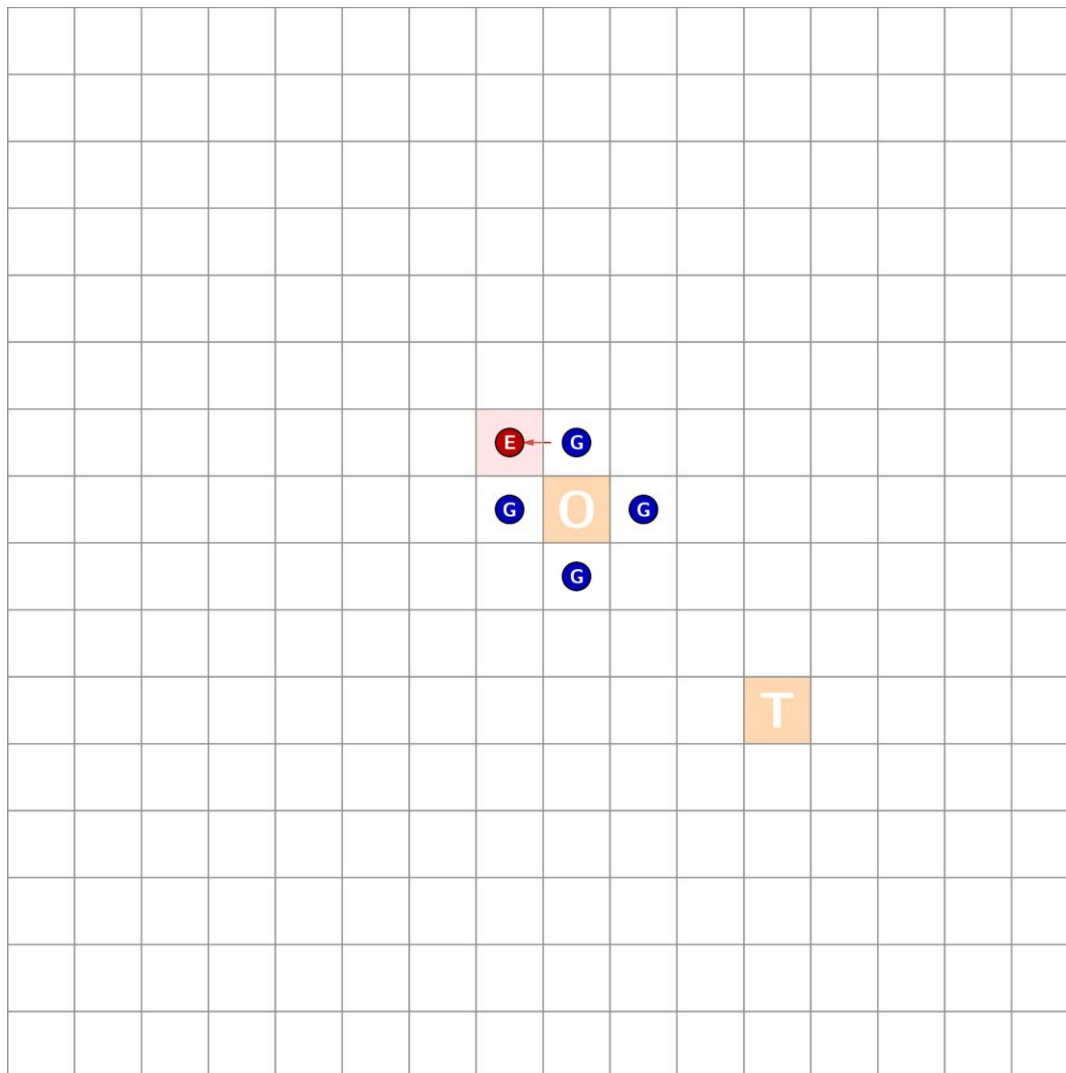
- Assume that at most n/c ants fail.
- Cluster up ants as a robust super-ant.



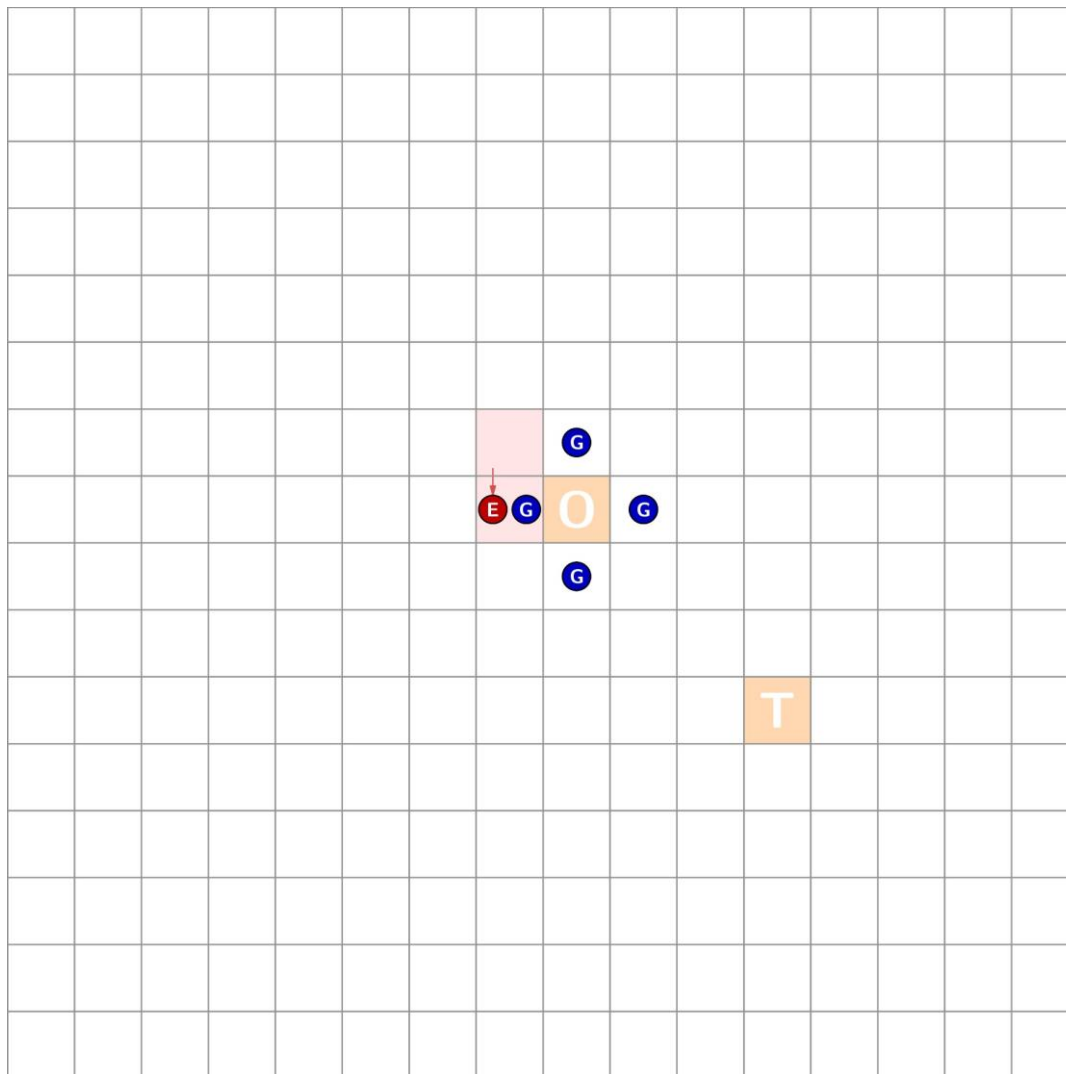
Robust Search



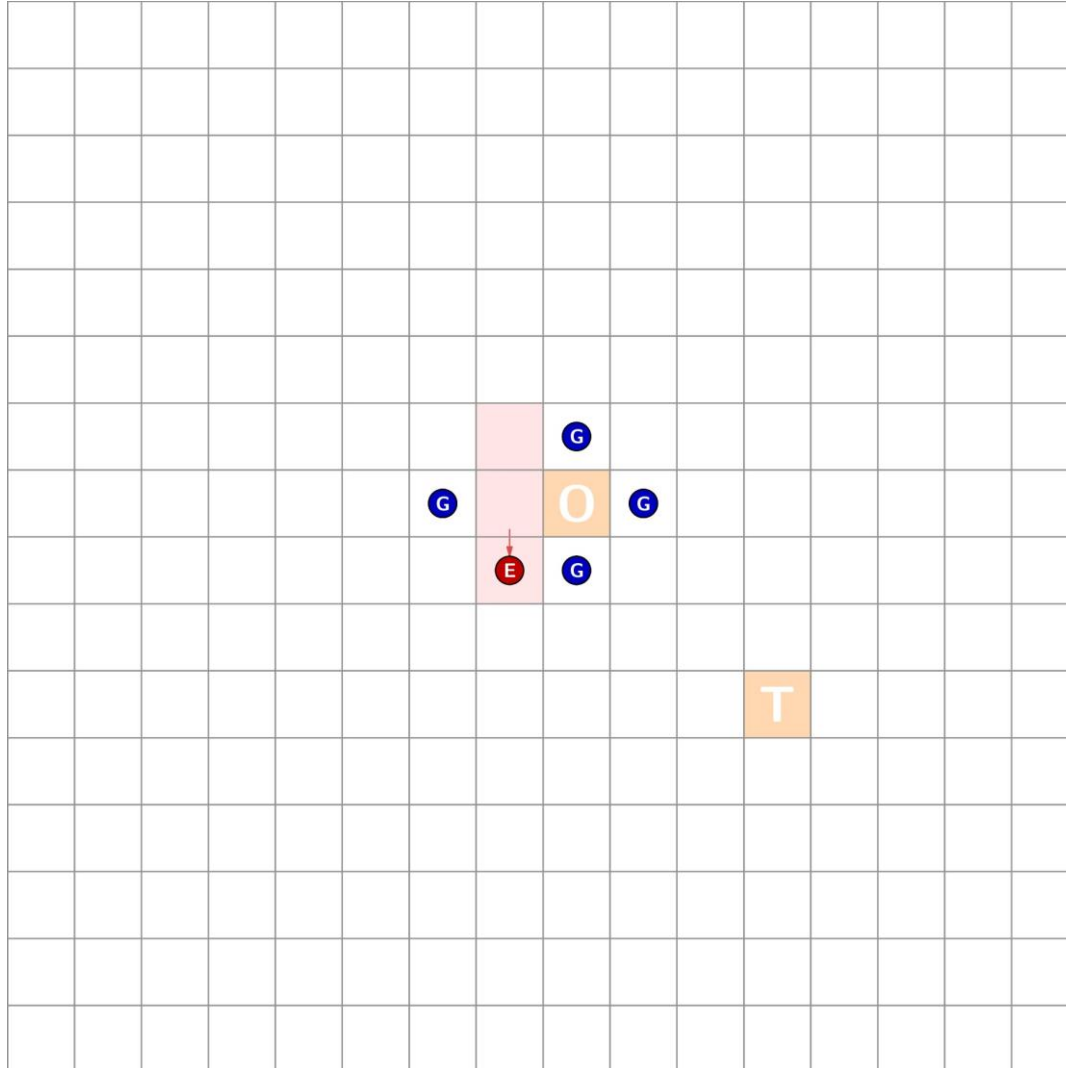
Robust Search



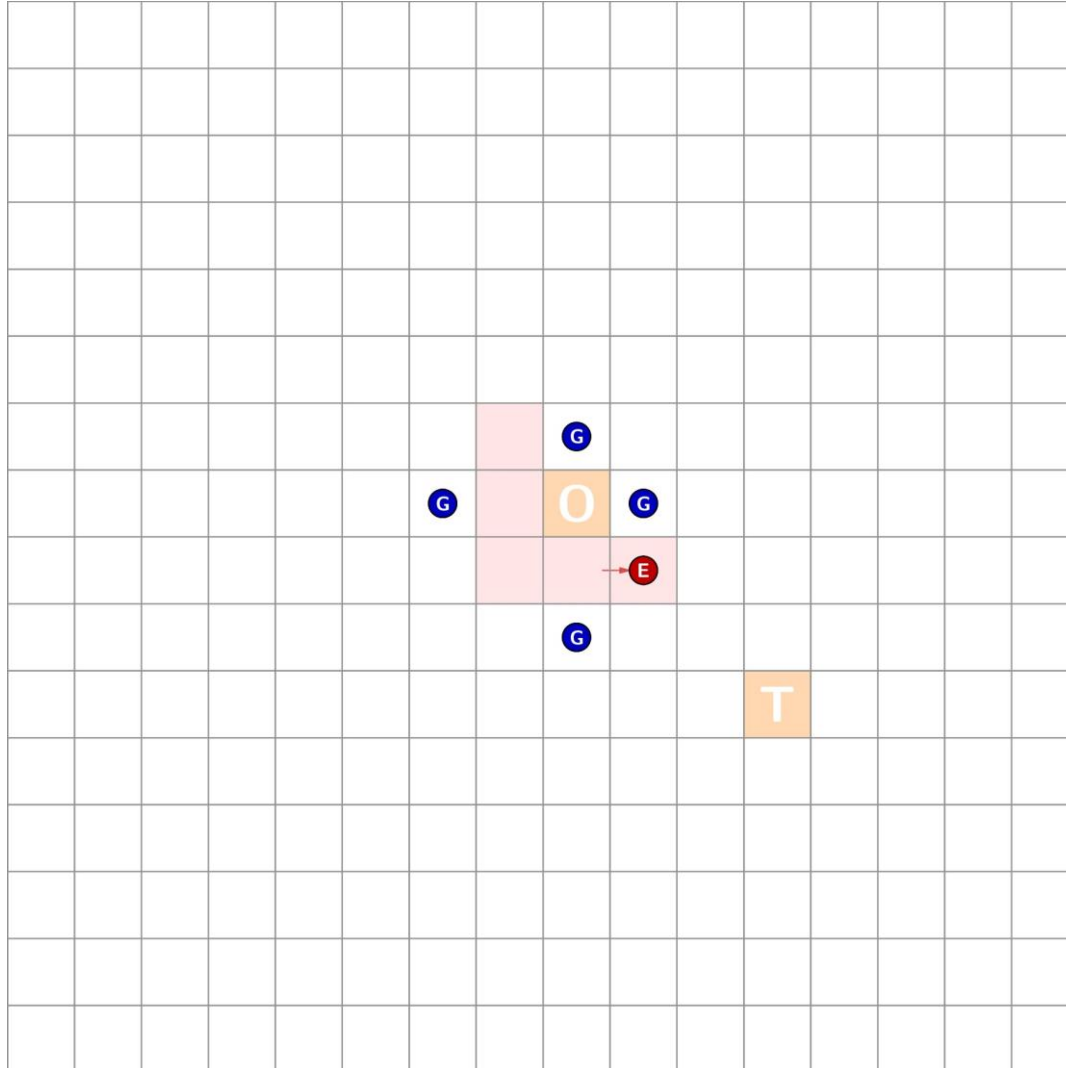
Robust Search



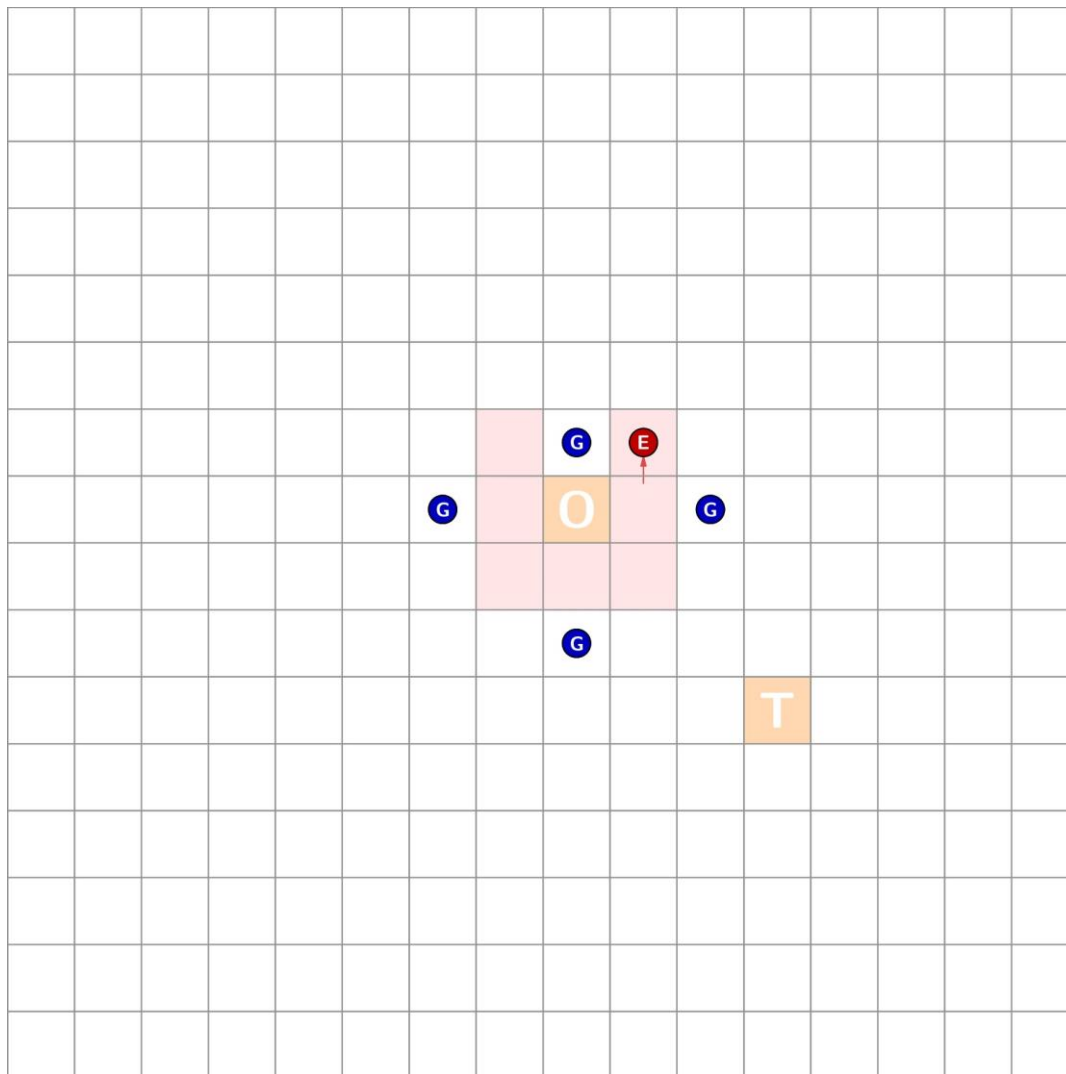
Robust Search



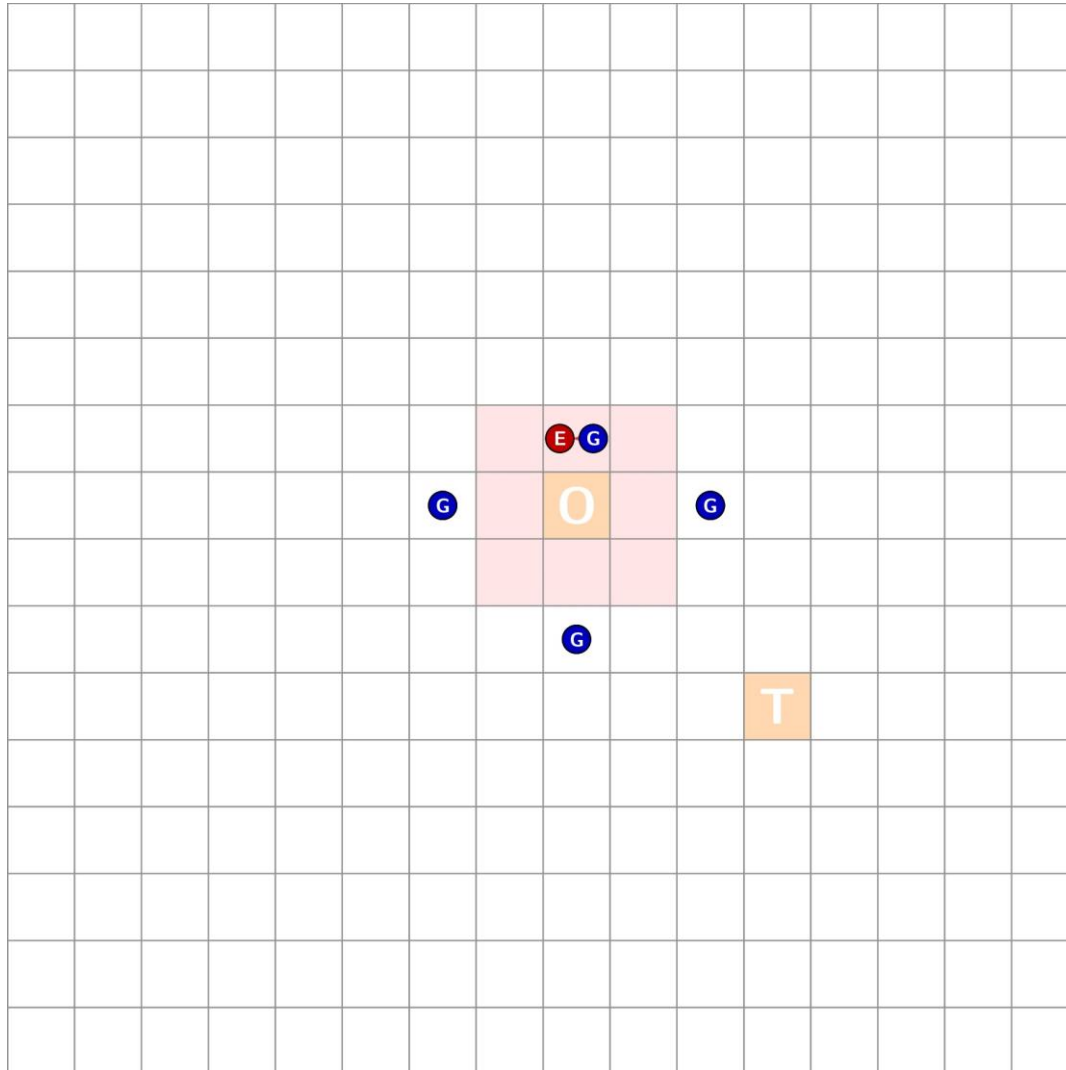
Robust Search



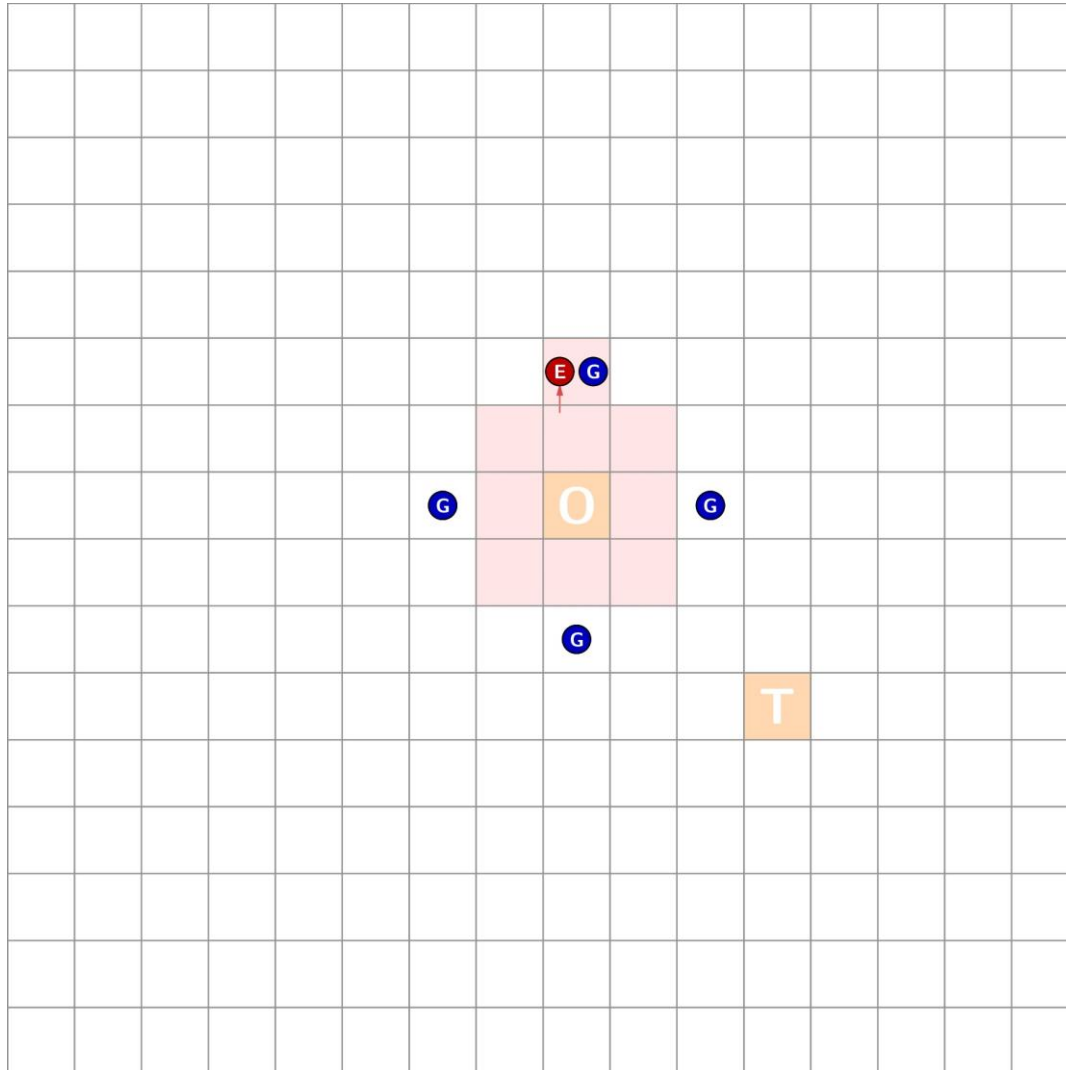
Robust Search



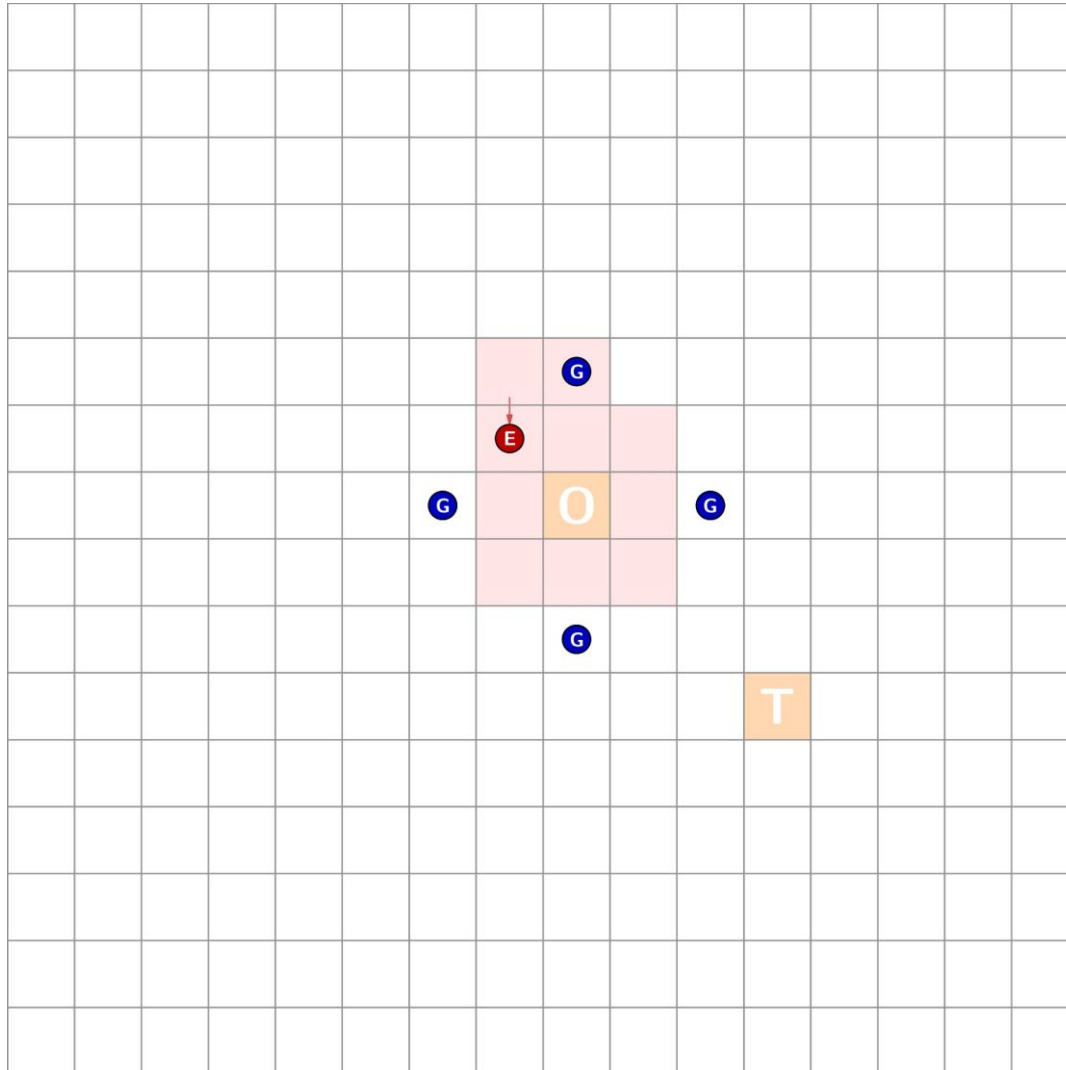
Robust Search



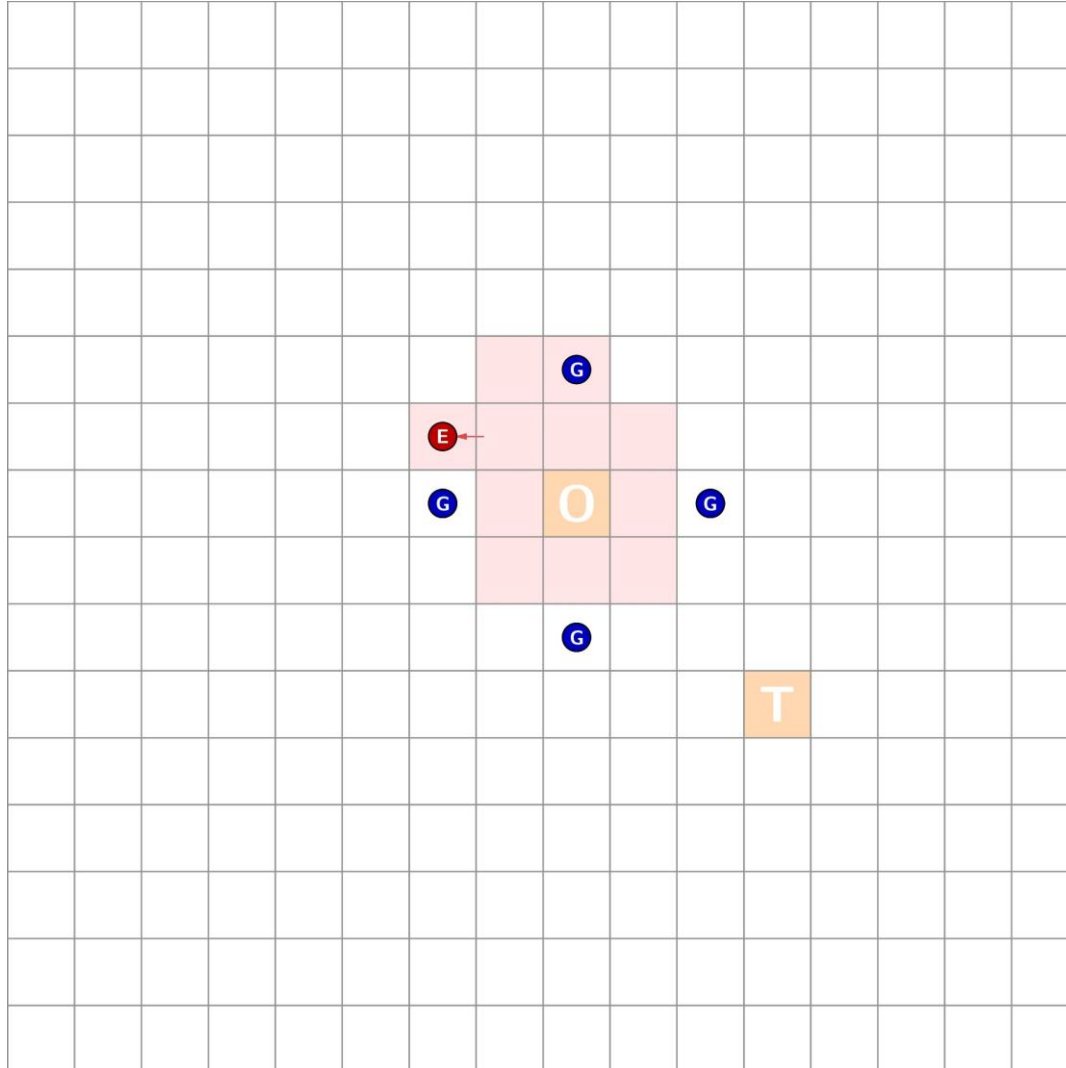
Robust Search



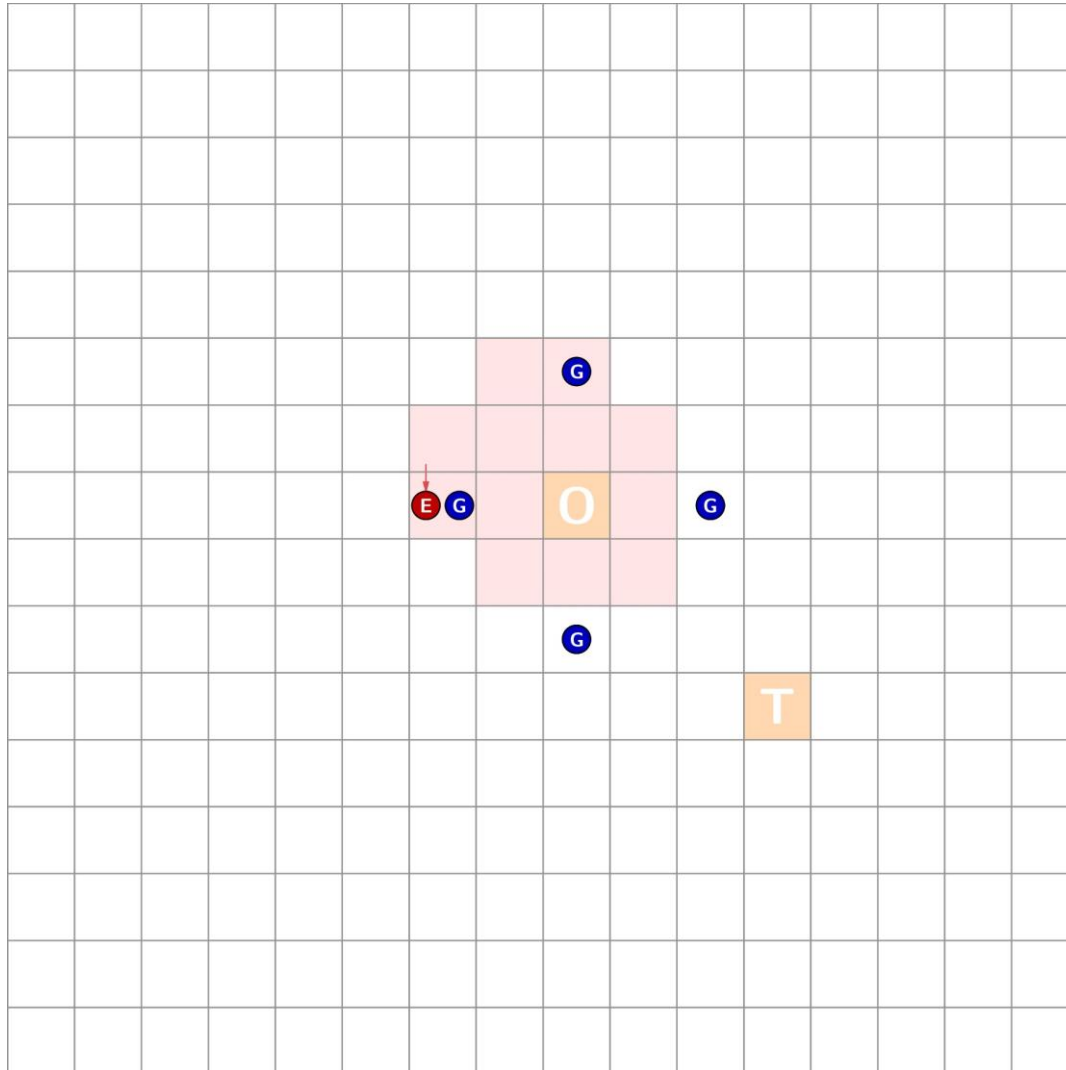
Robust Search



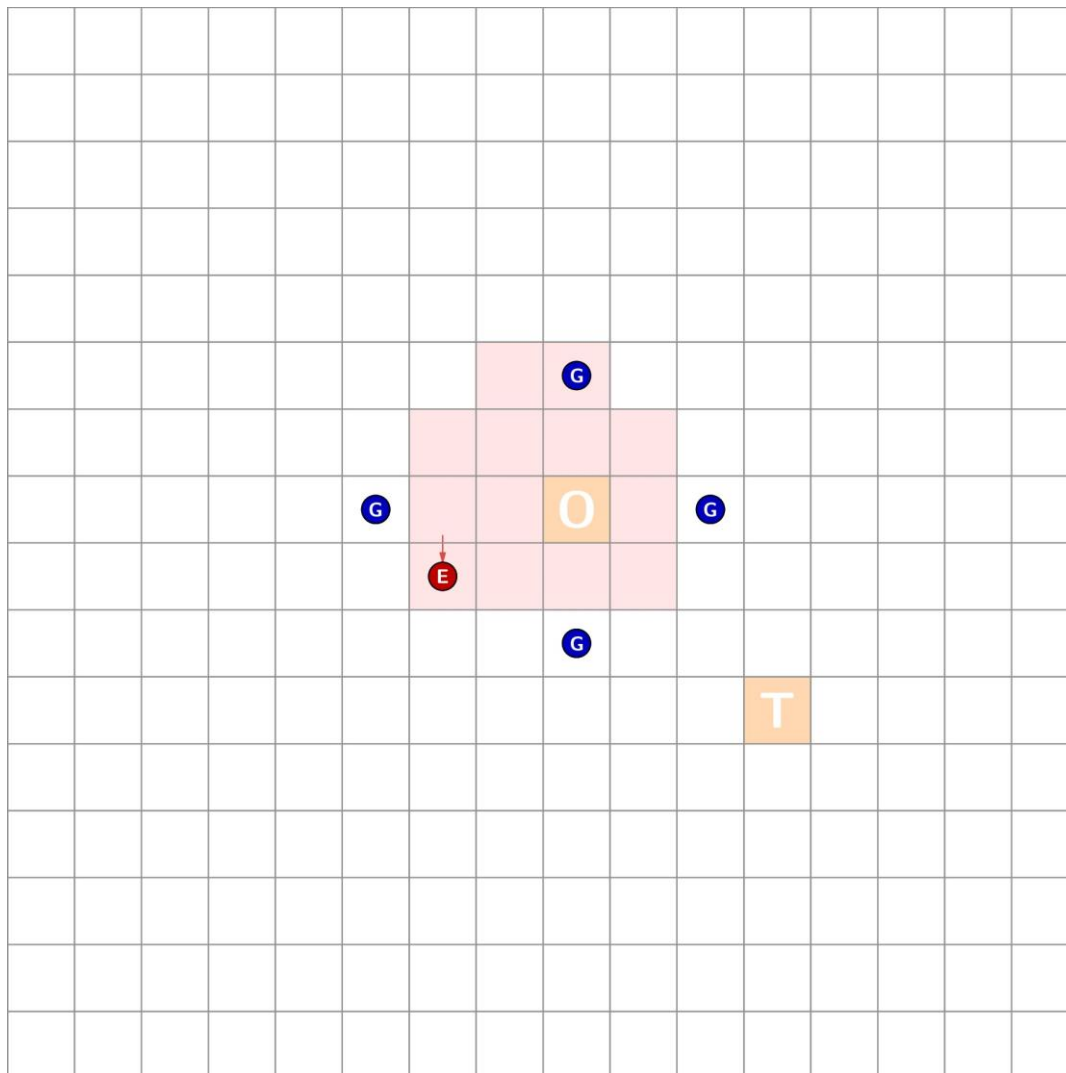
Robust Search



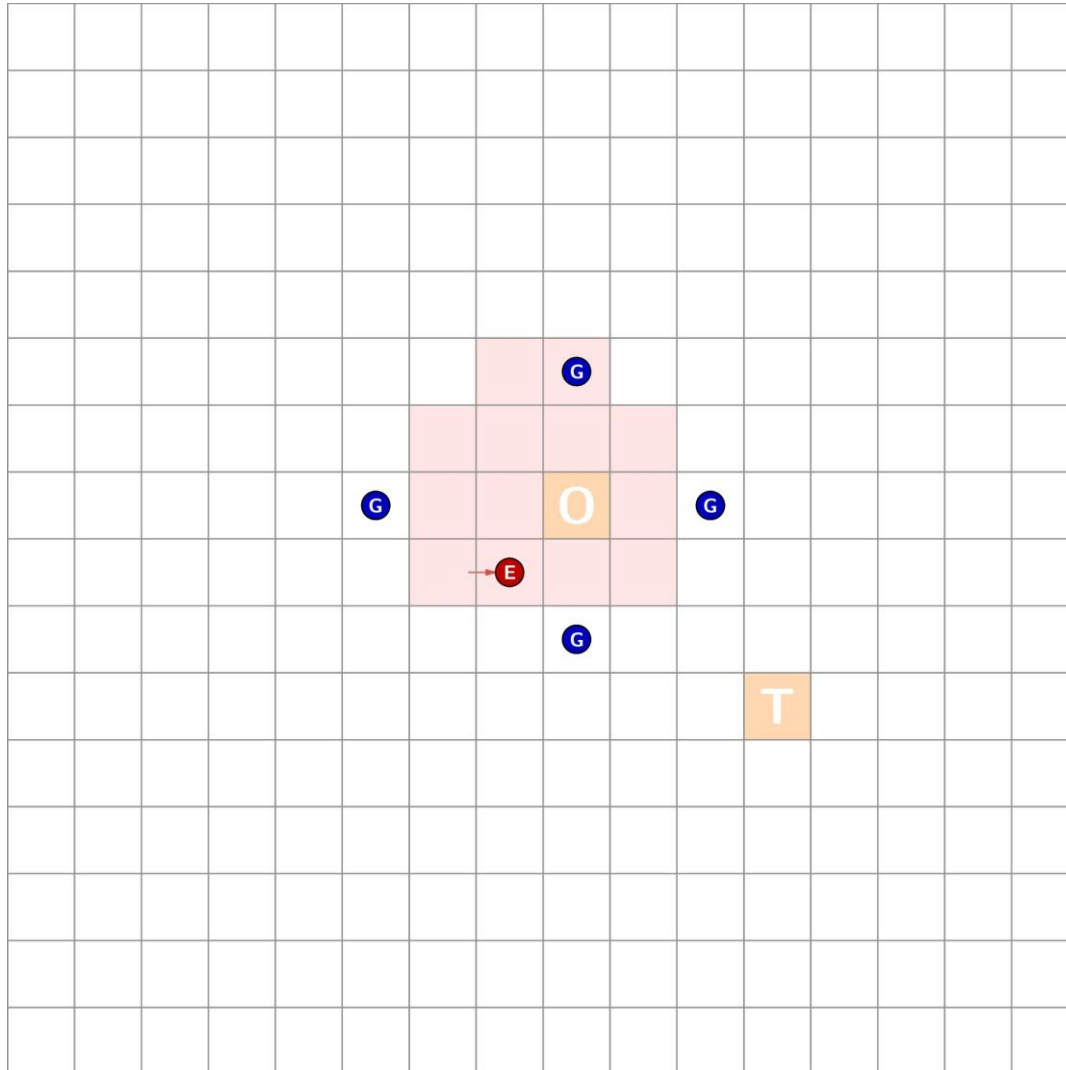
Robust Search



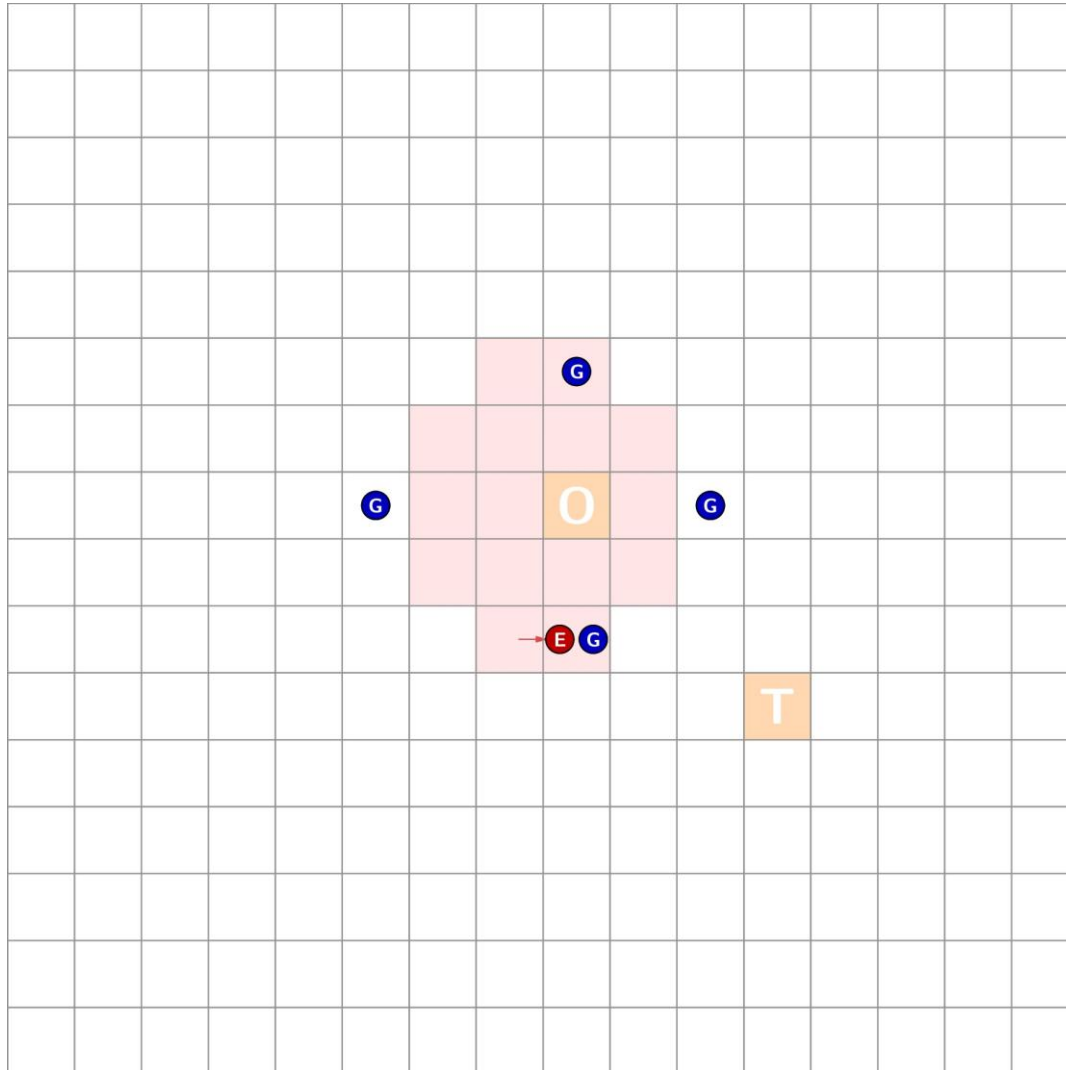
Robust Search



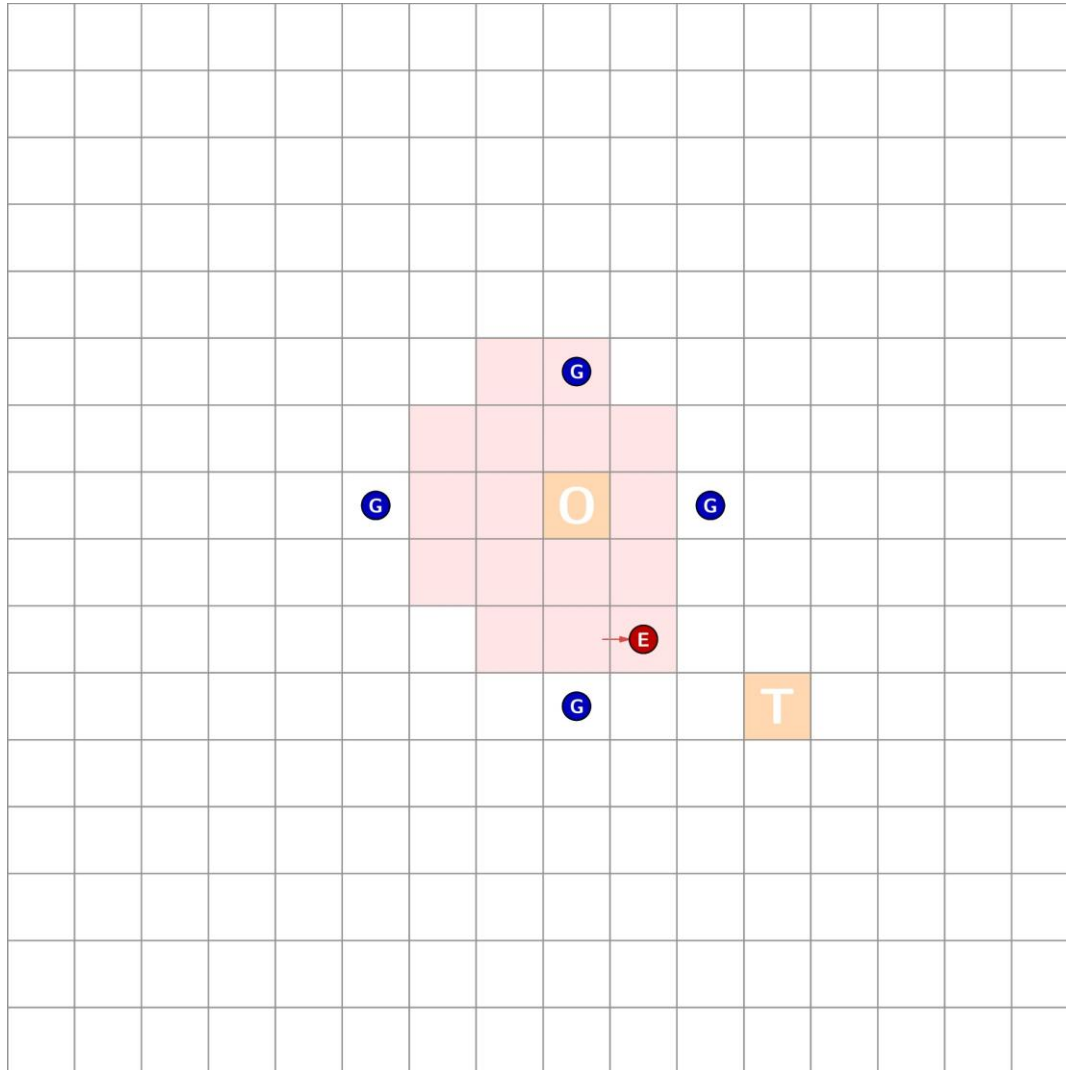
Robust Search



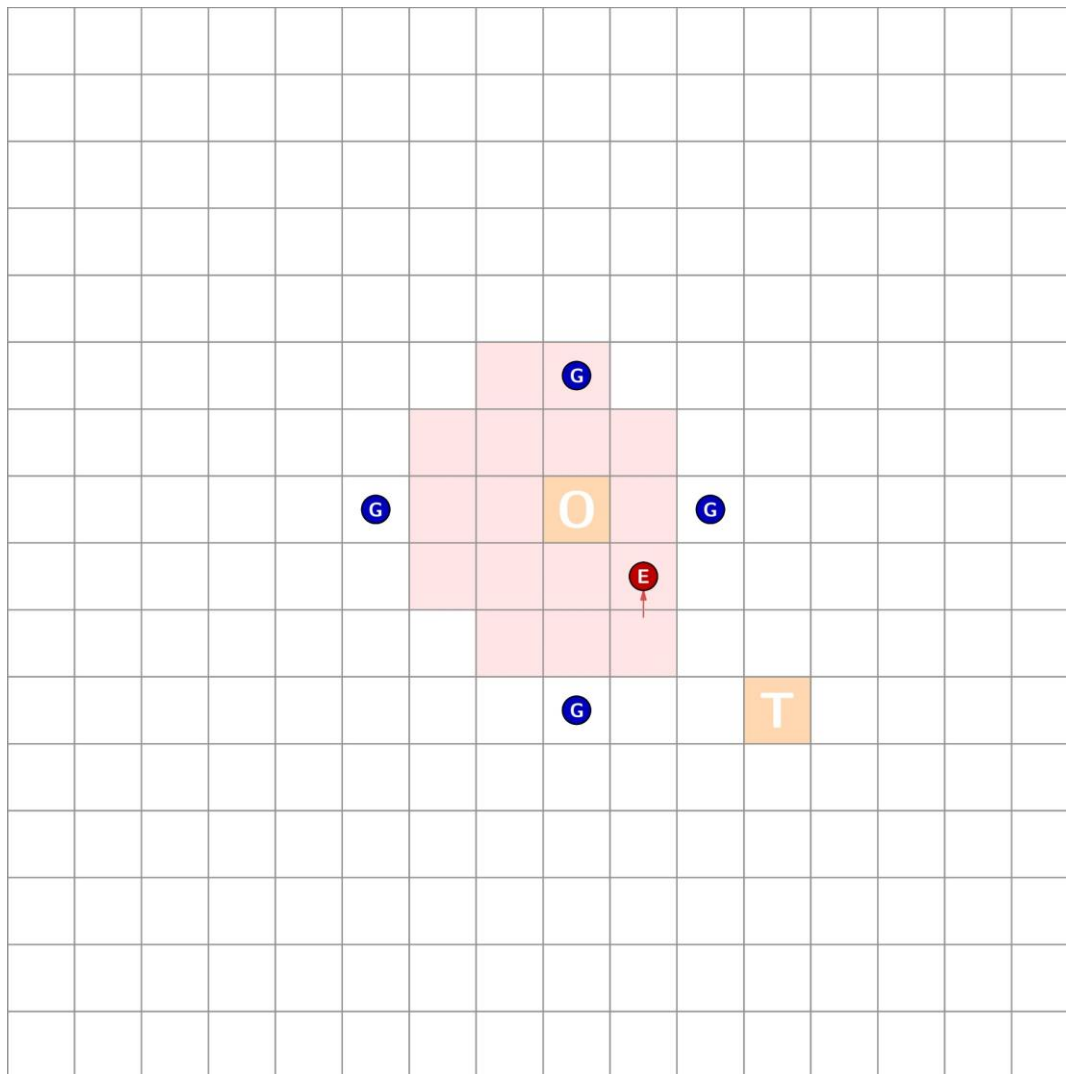
Robust Search



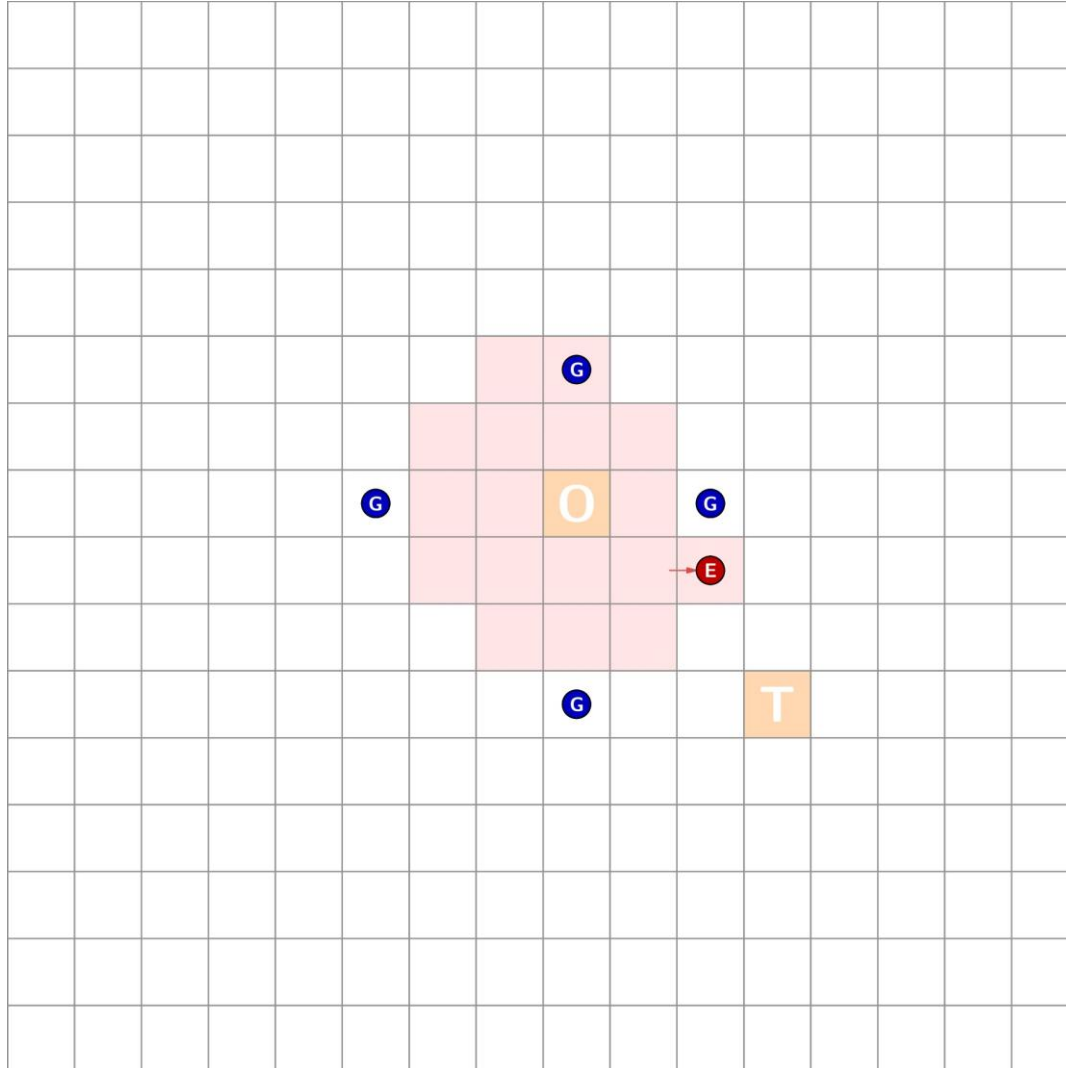
Robust Search



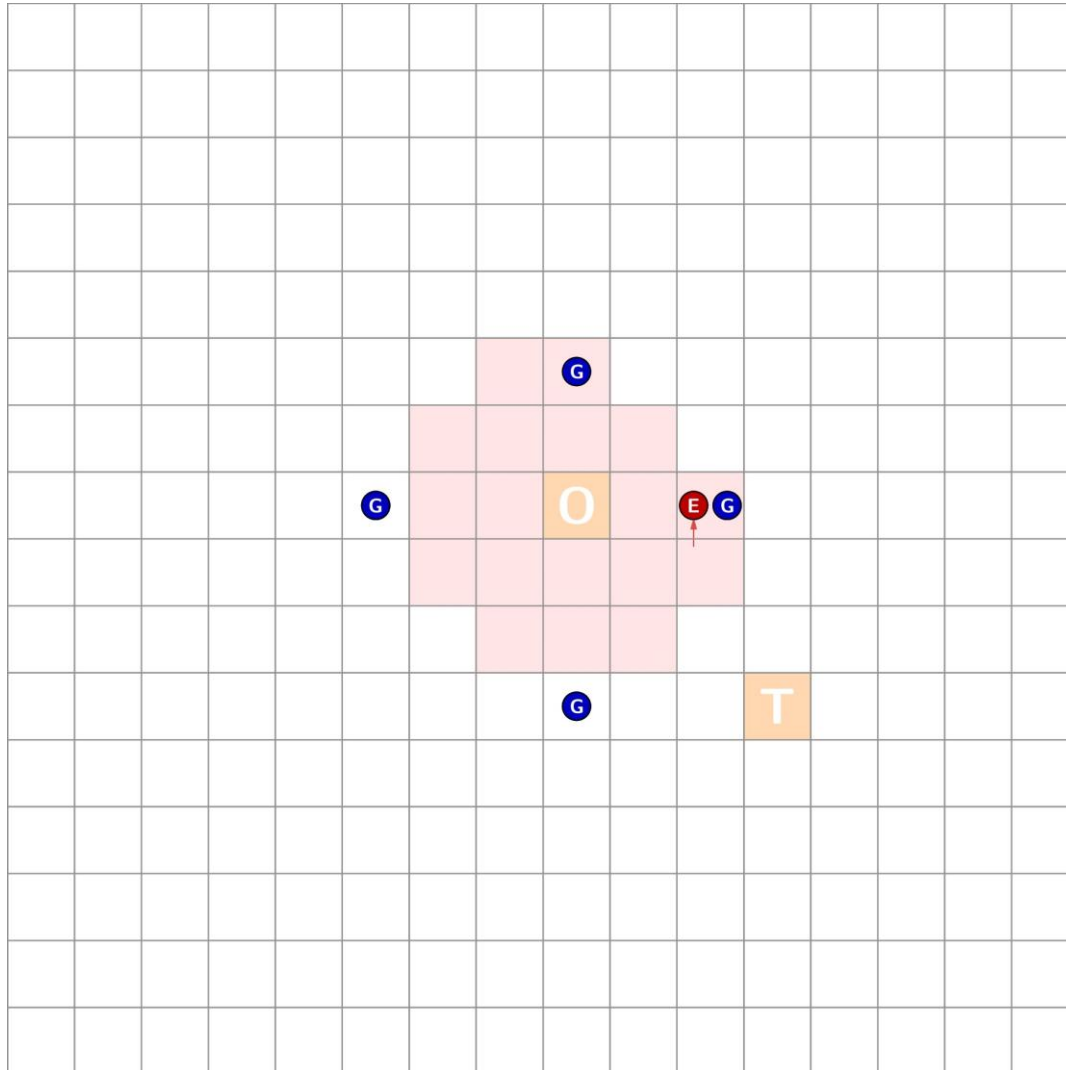
Robust Search



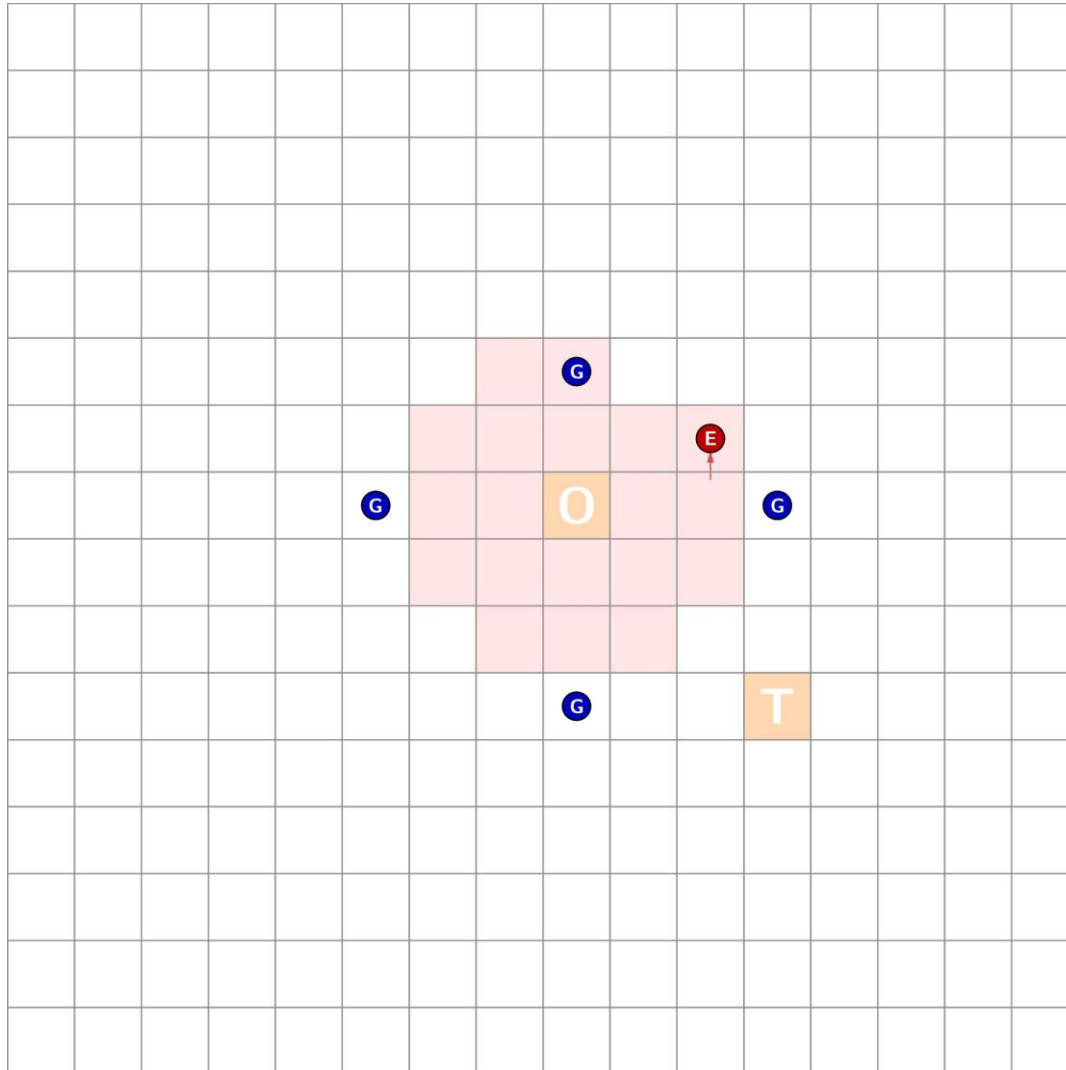
Robust Search



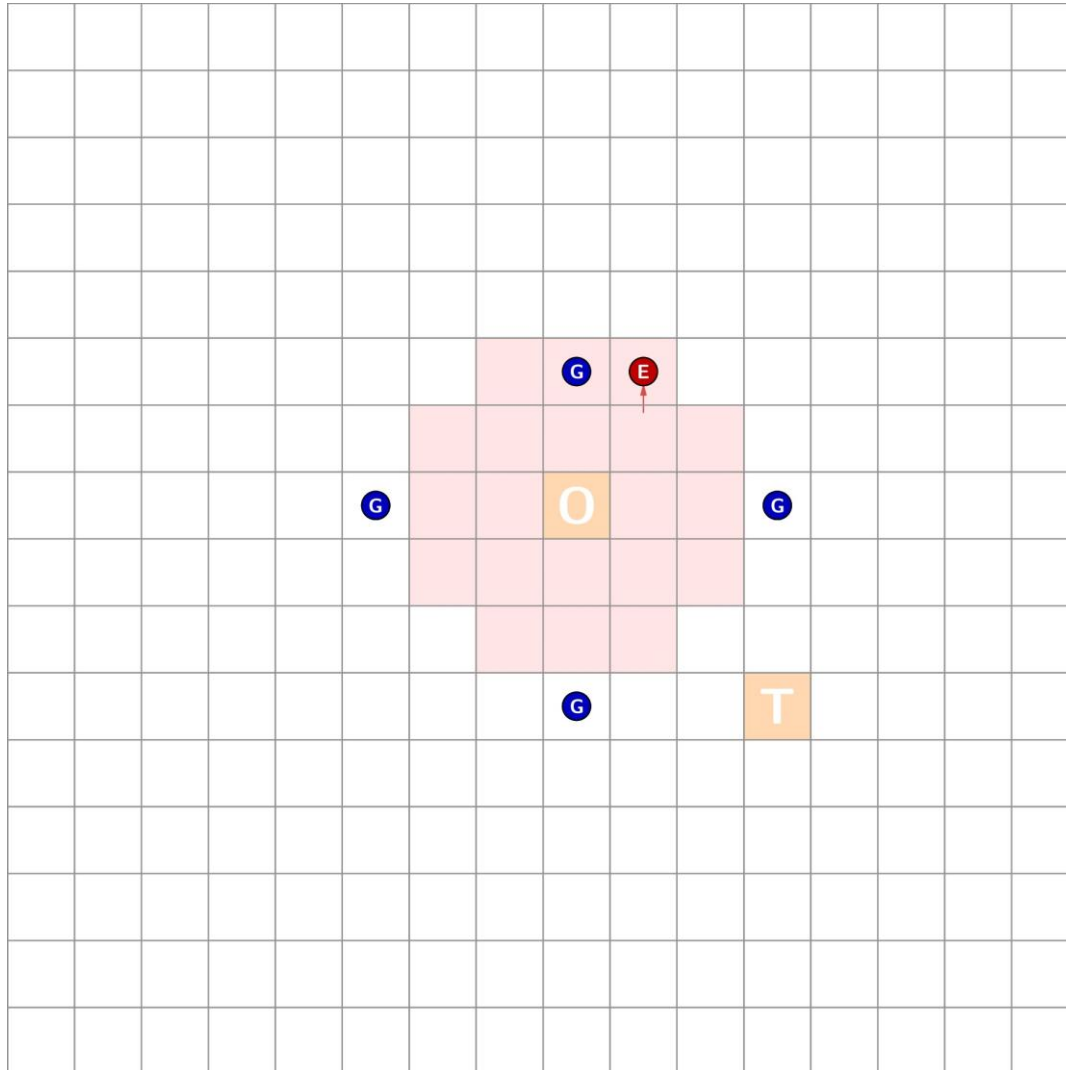
Robust Search



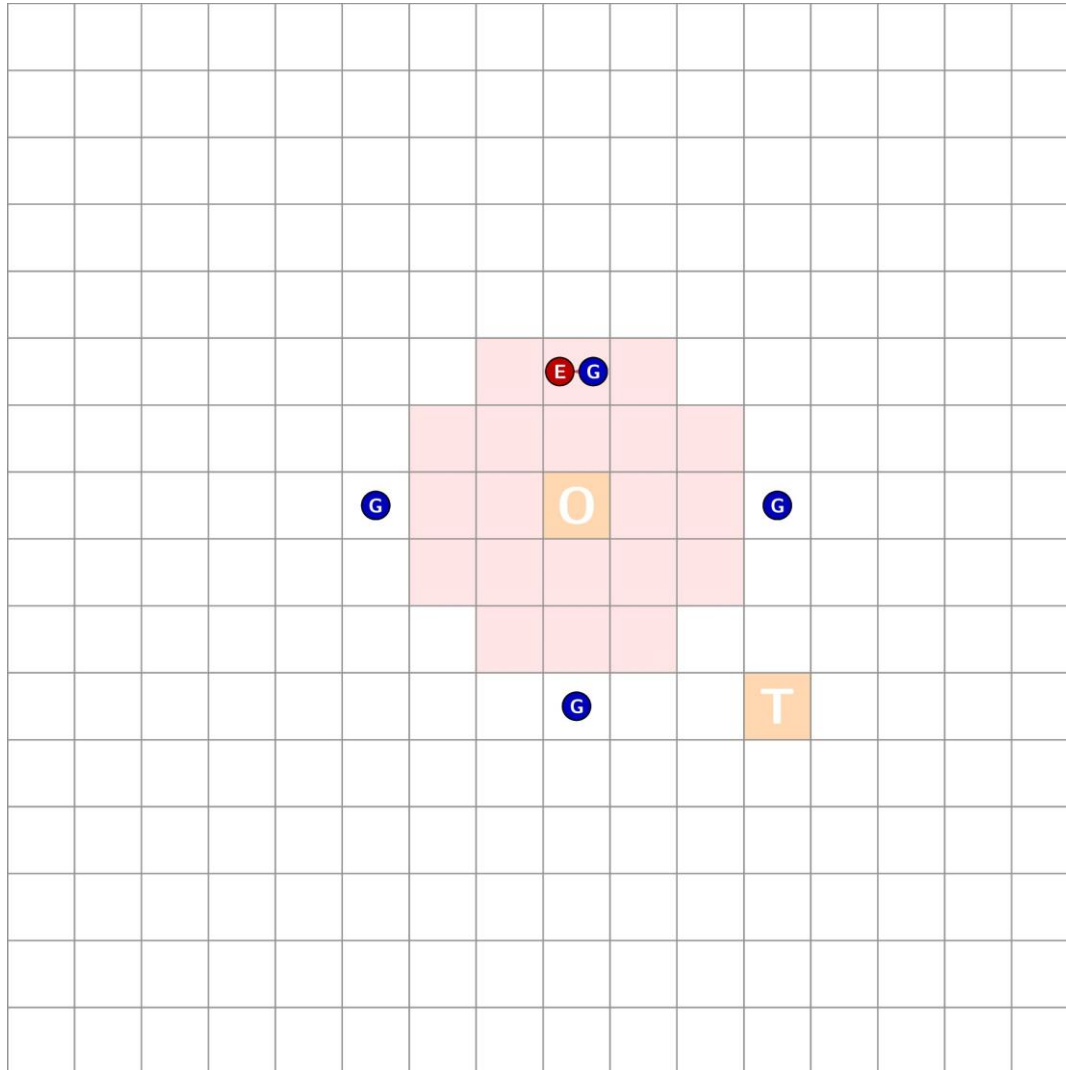
Robust Search



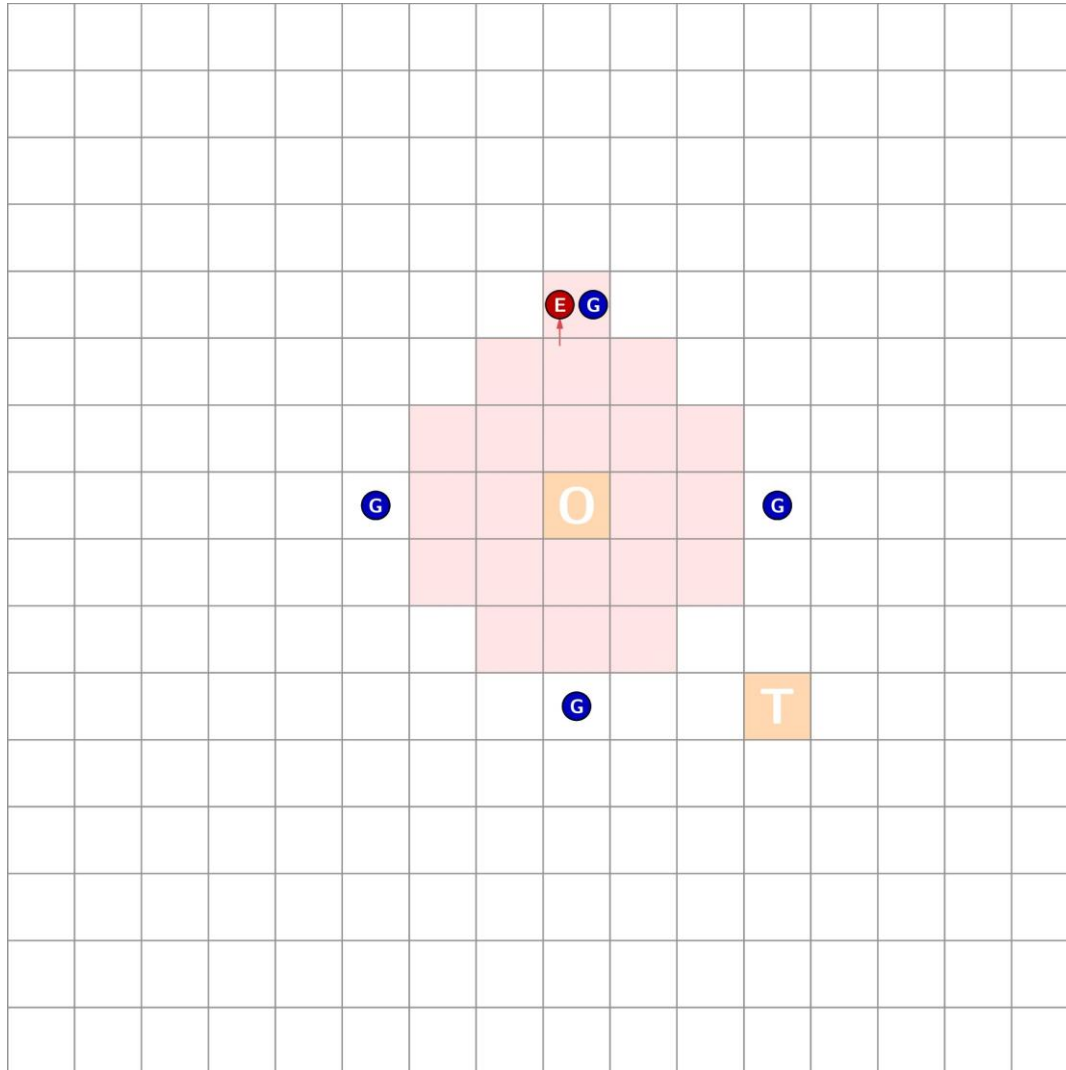
Robust Search



Robust Search



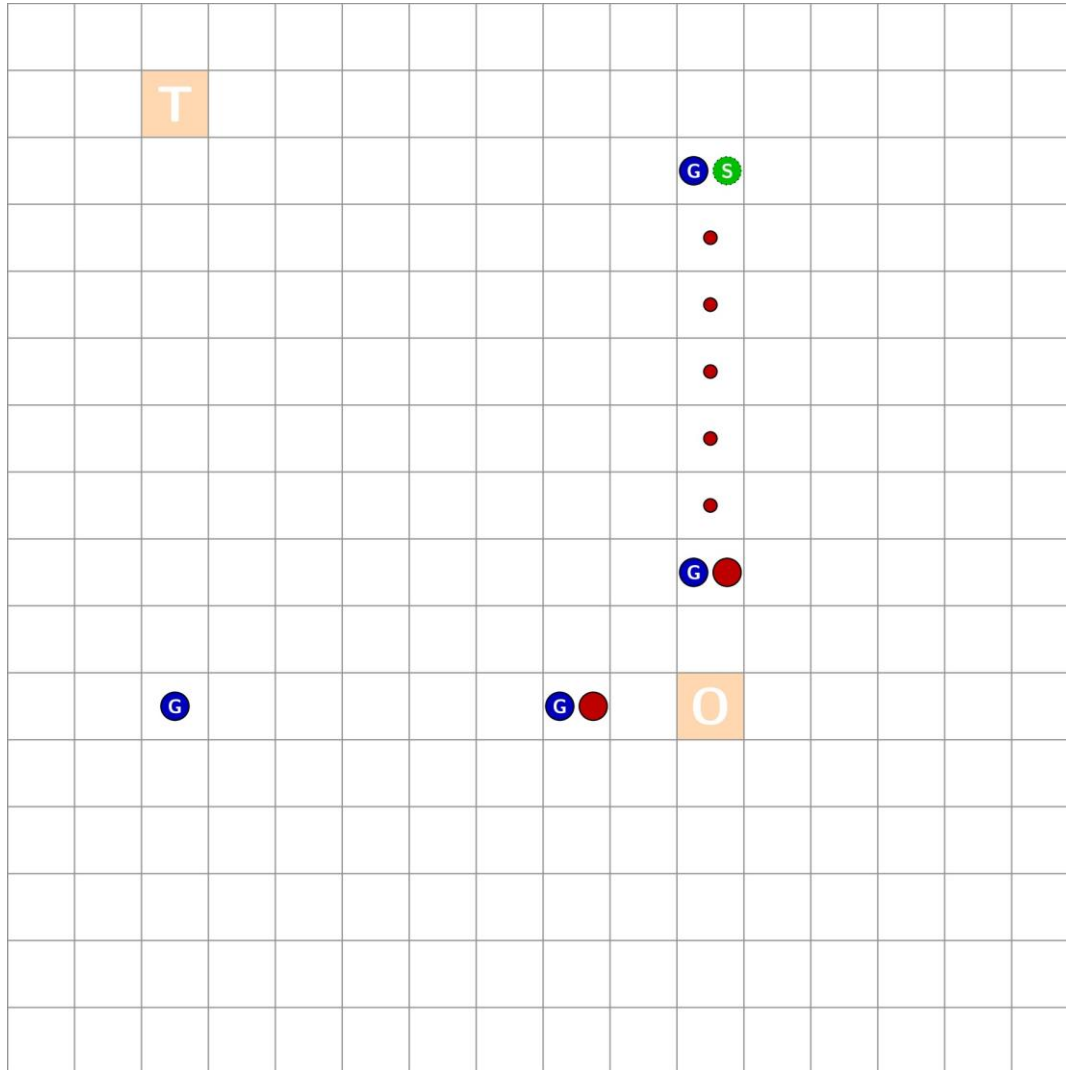
Robust Search



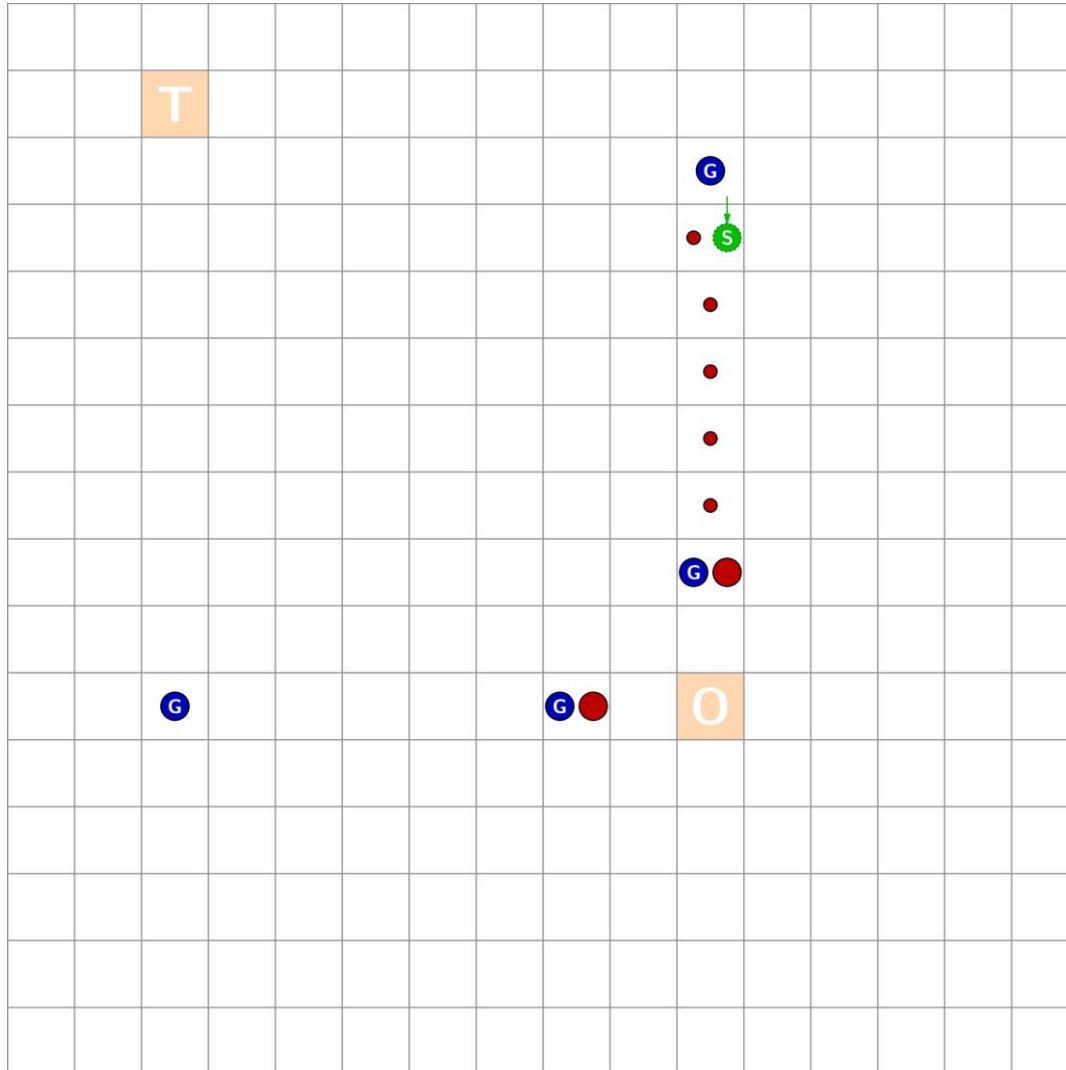
Robust Search

- Can create at most c super-ants.
- Using only super-ants is slow... $\Omega(D^2)$

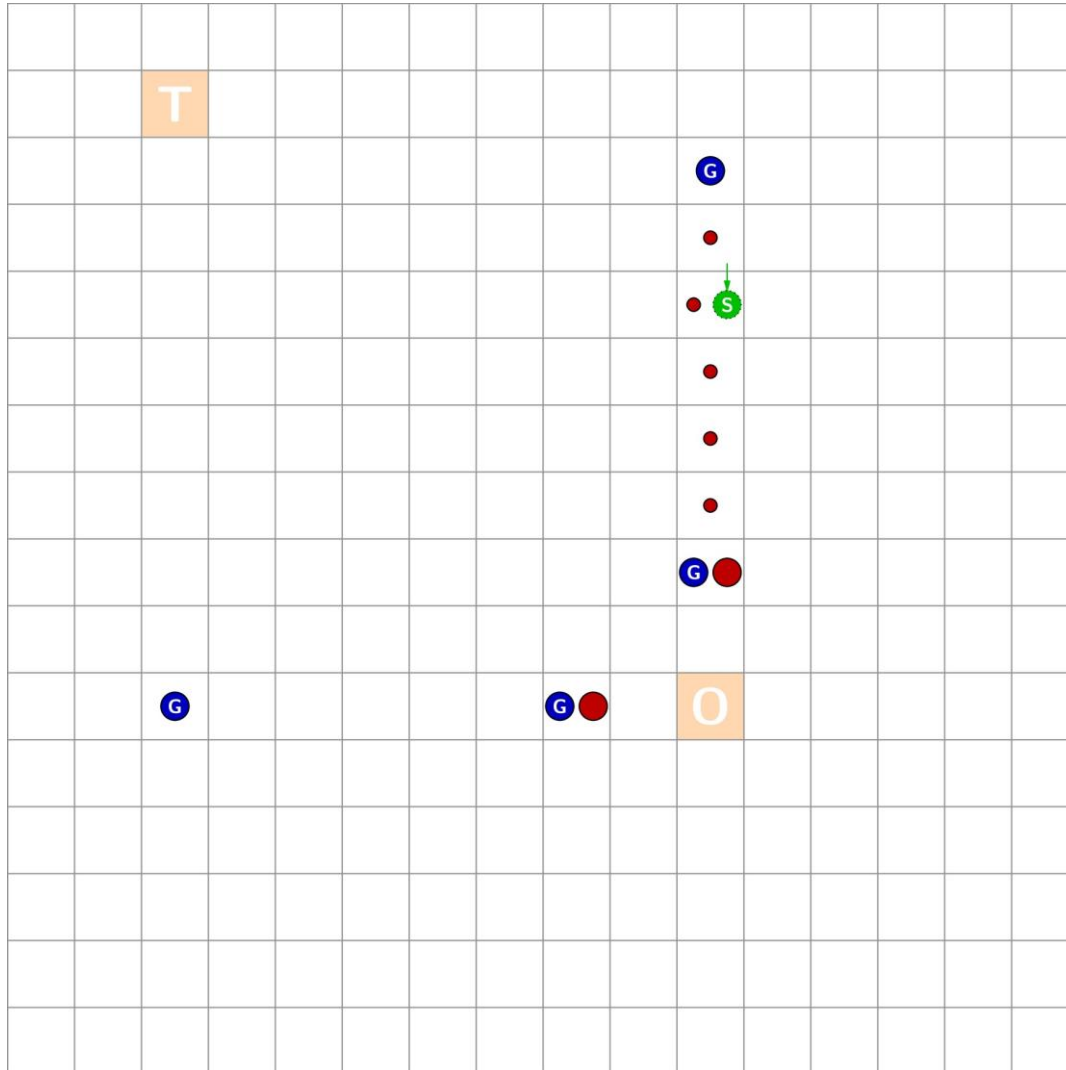
Robust Search



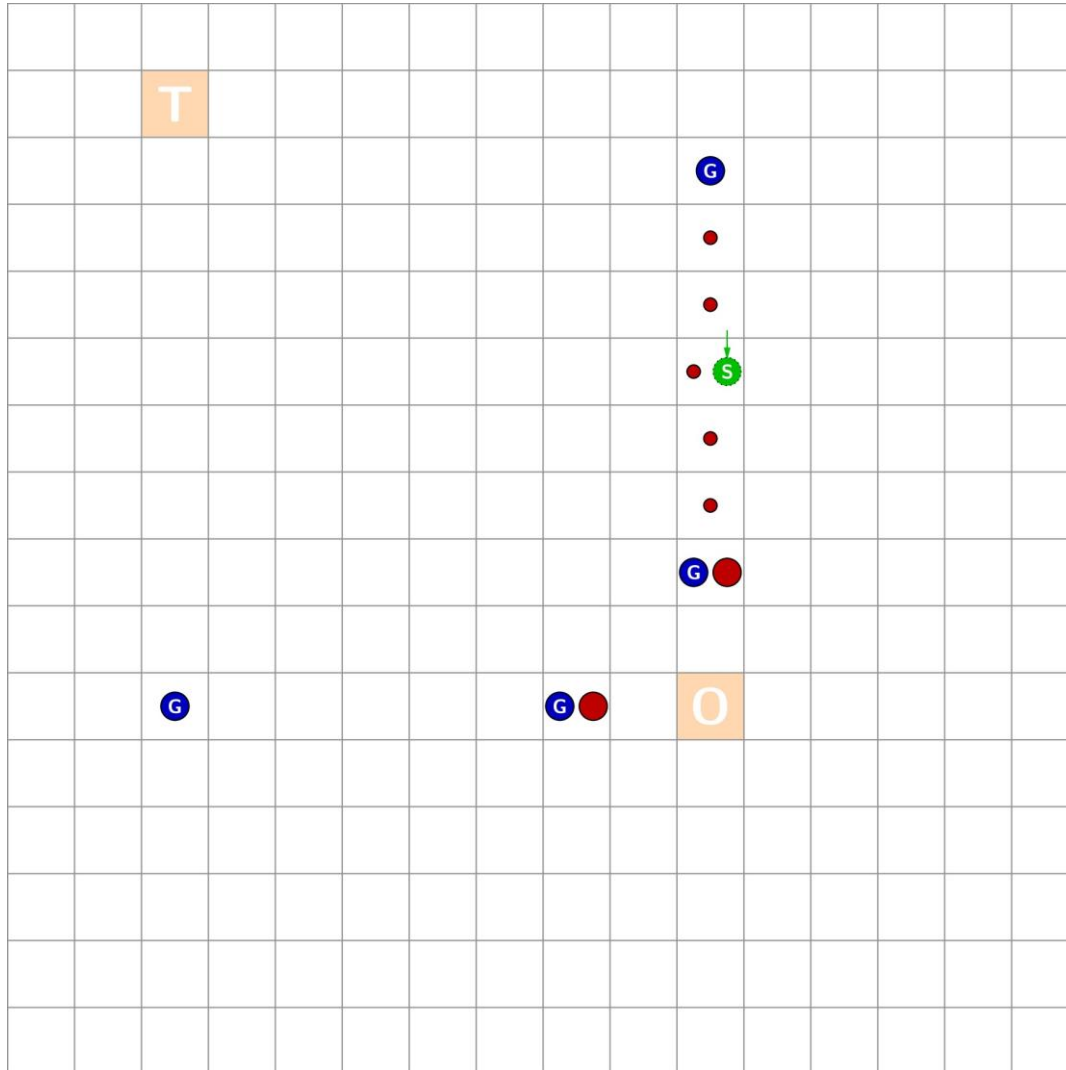
Robust Search



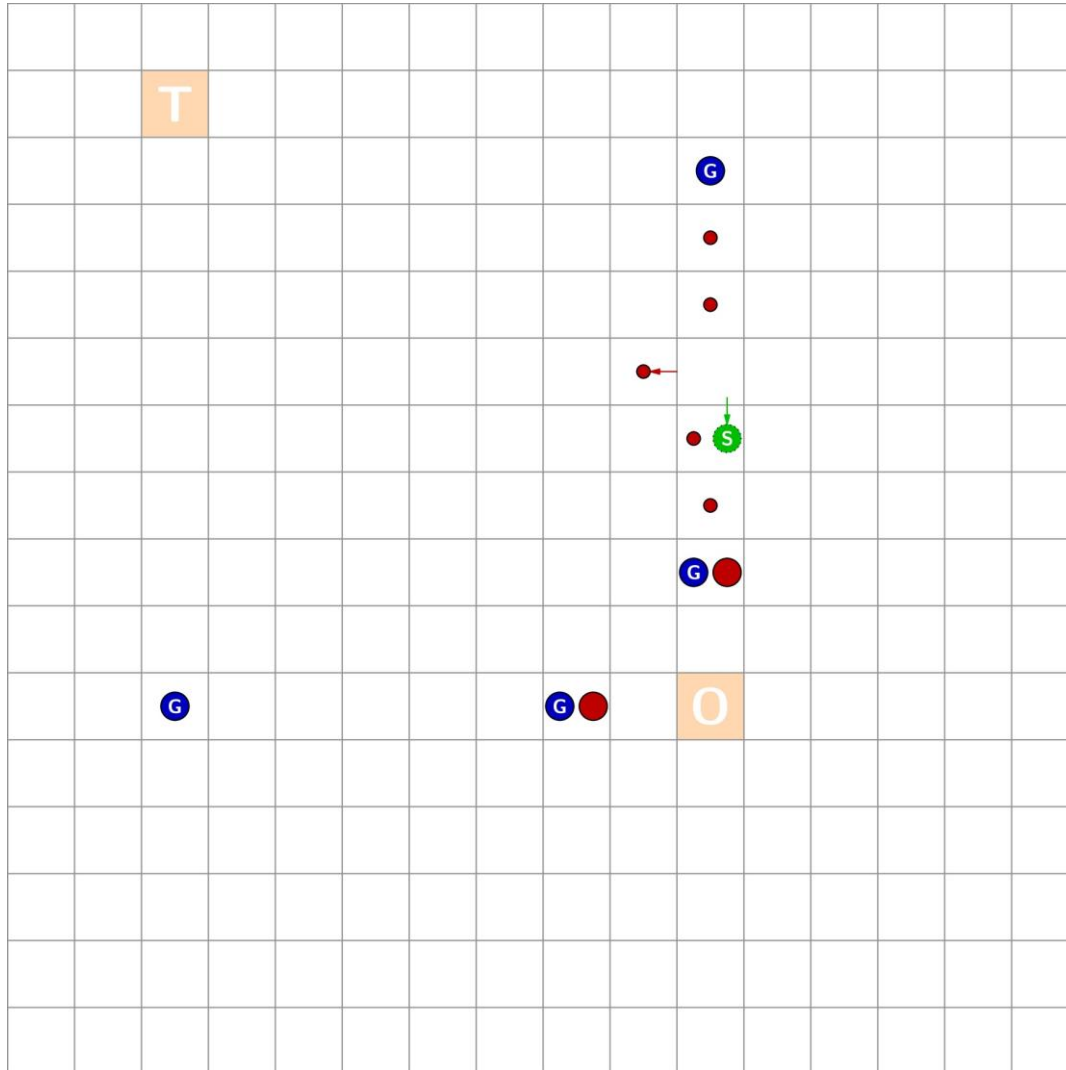
Robust Search



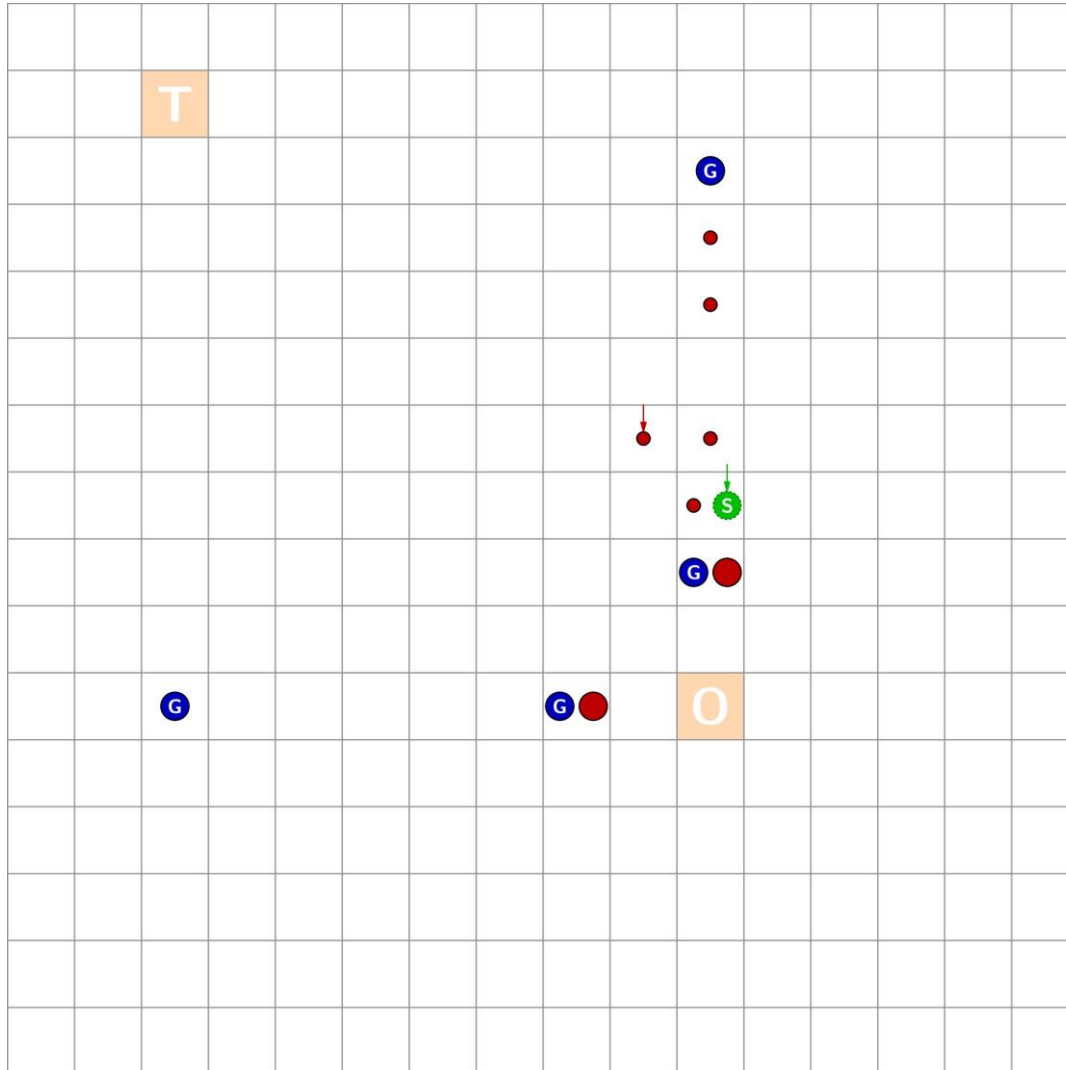
Robust Search



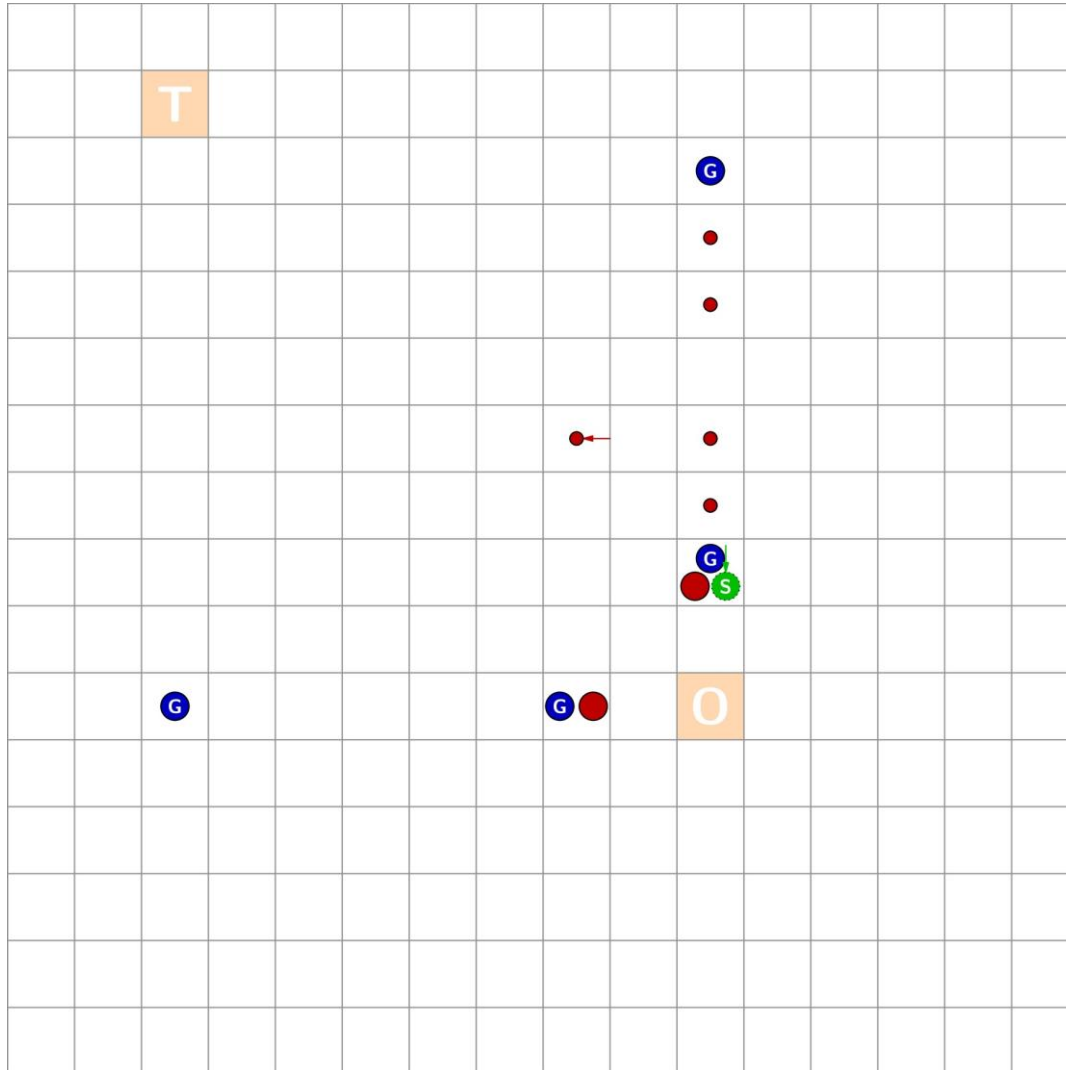
Robust Search



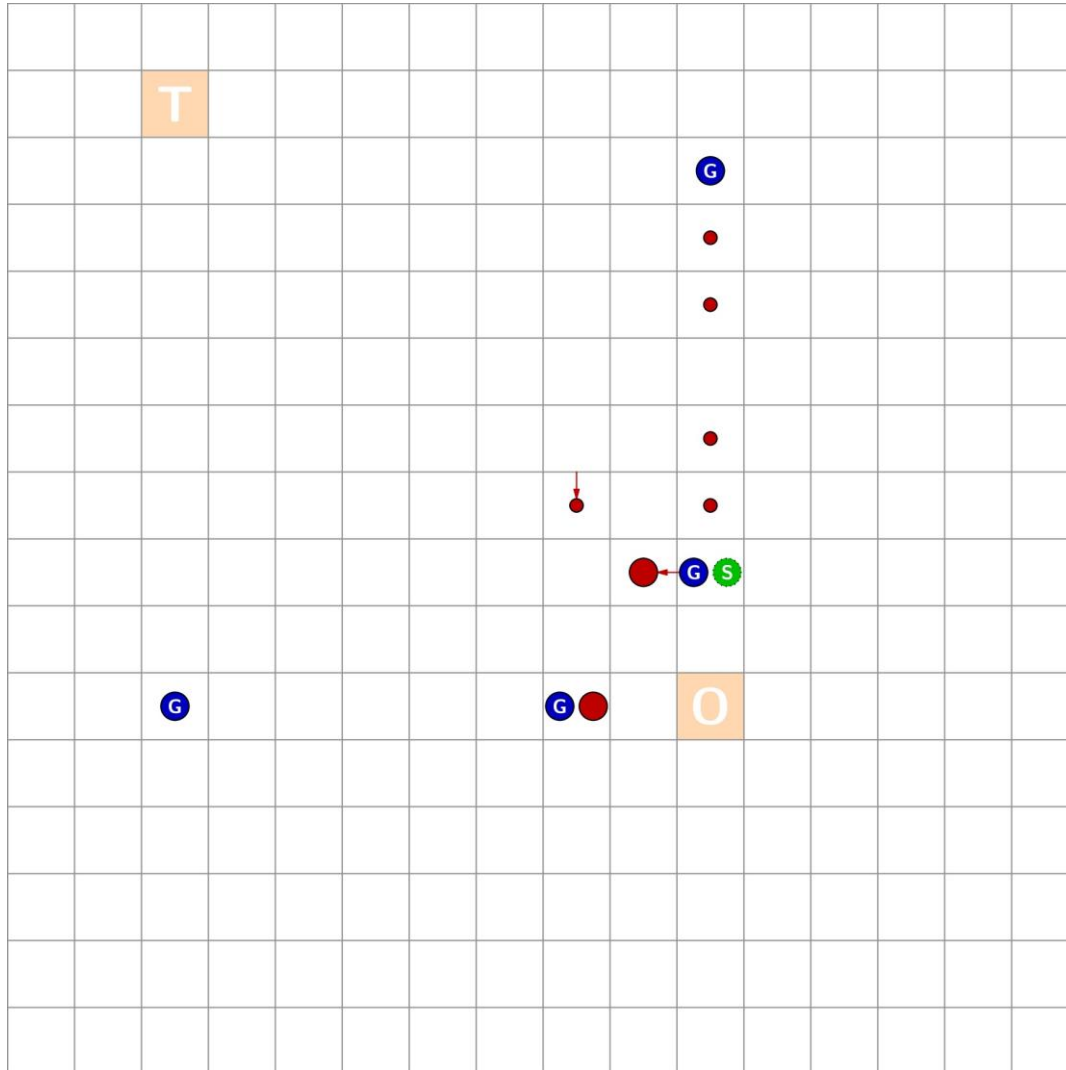
Robust Search



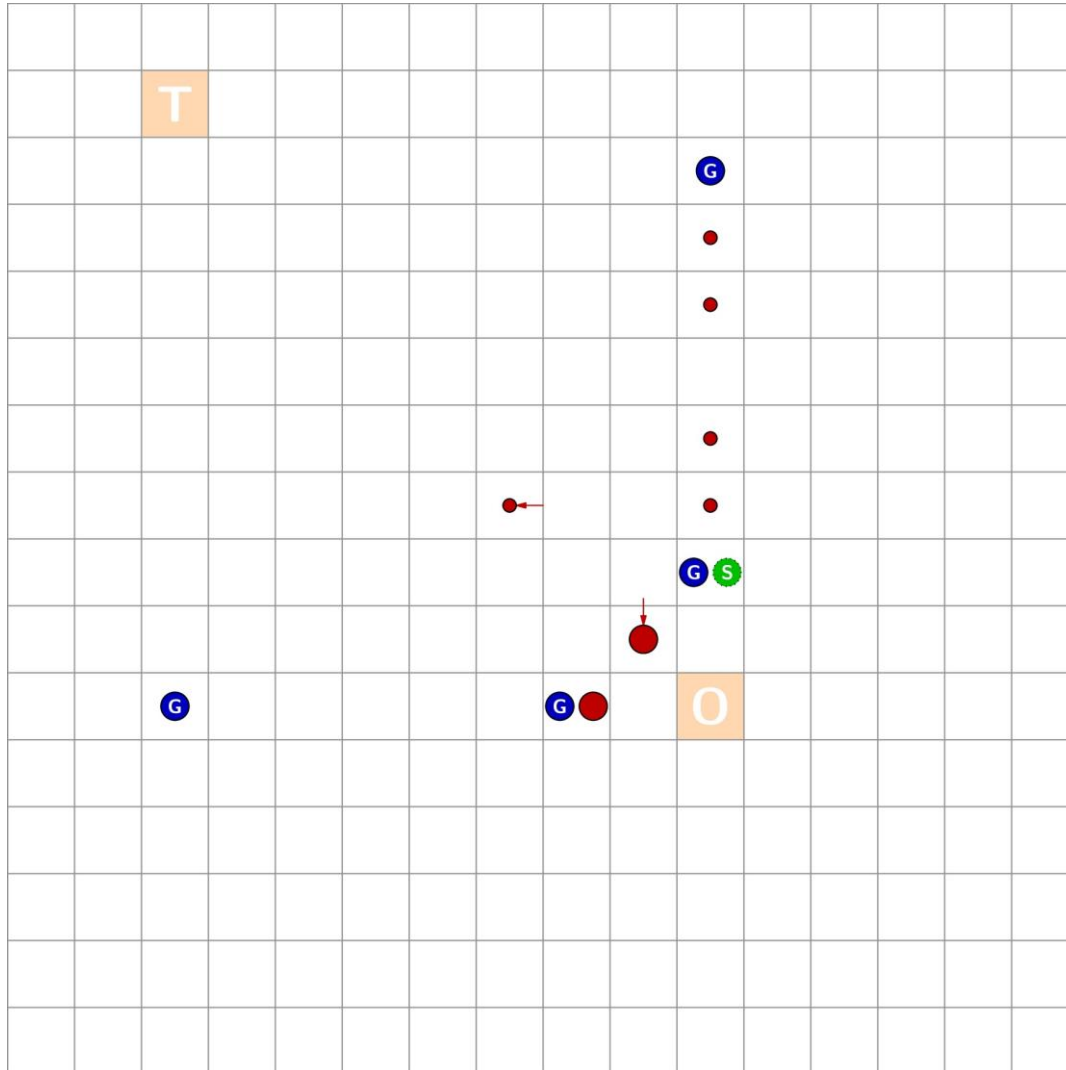
Robust Search



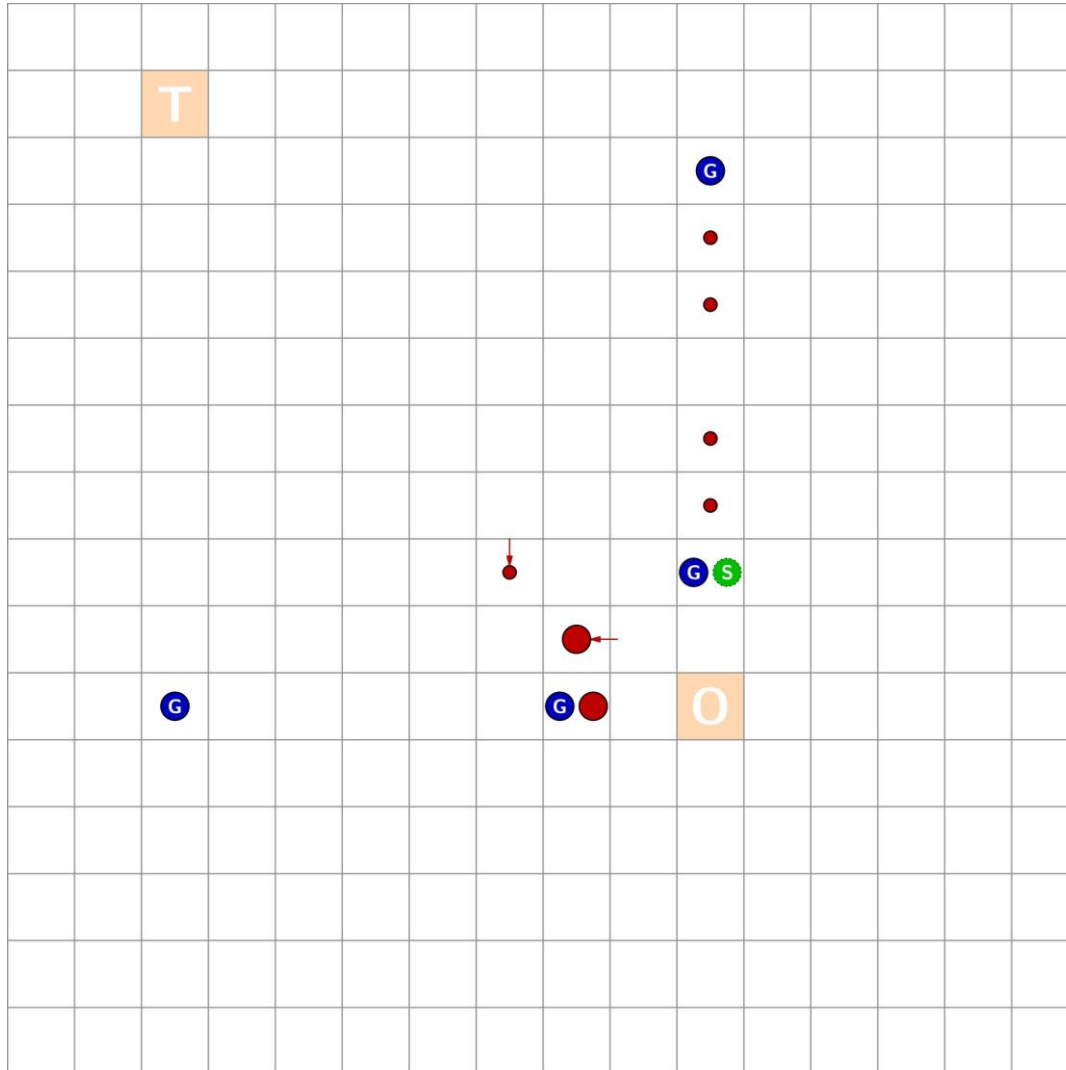
Robust Search



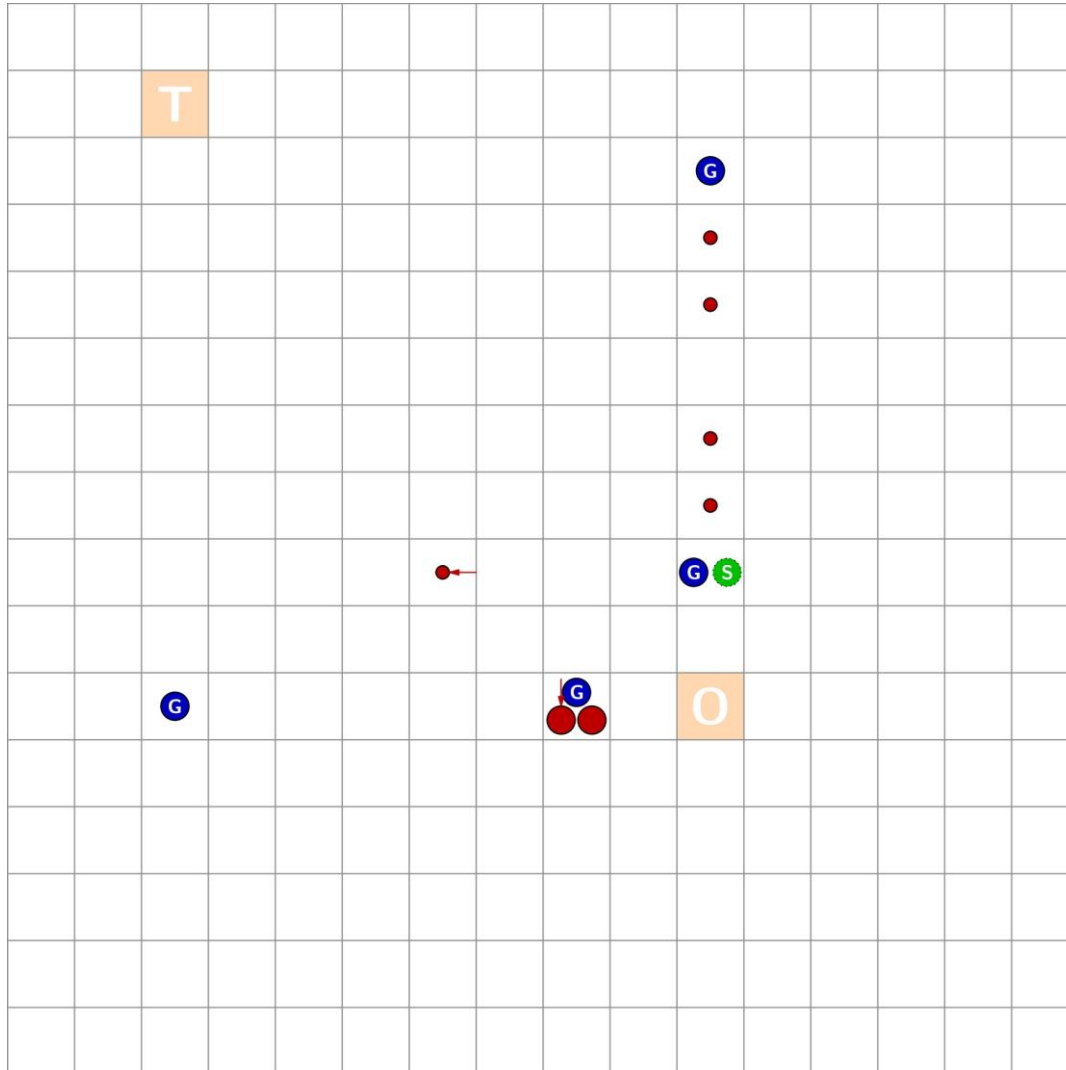
Robust Search



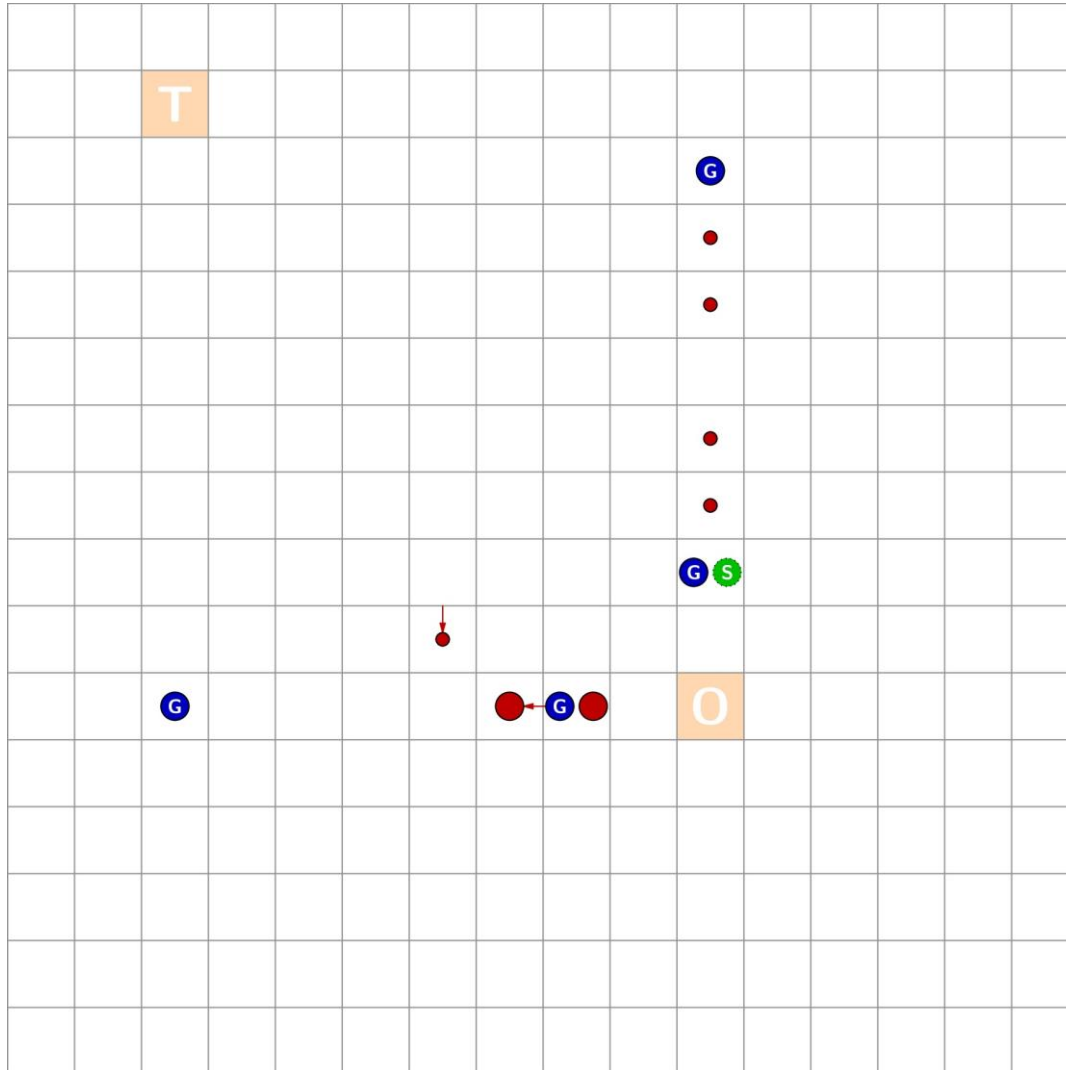
Robust Search



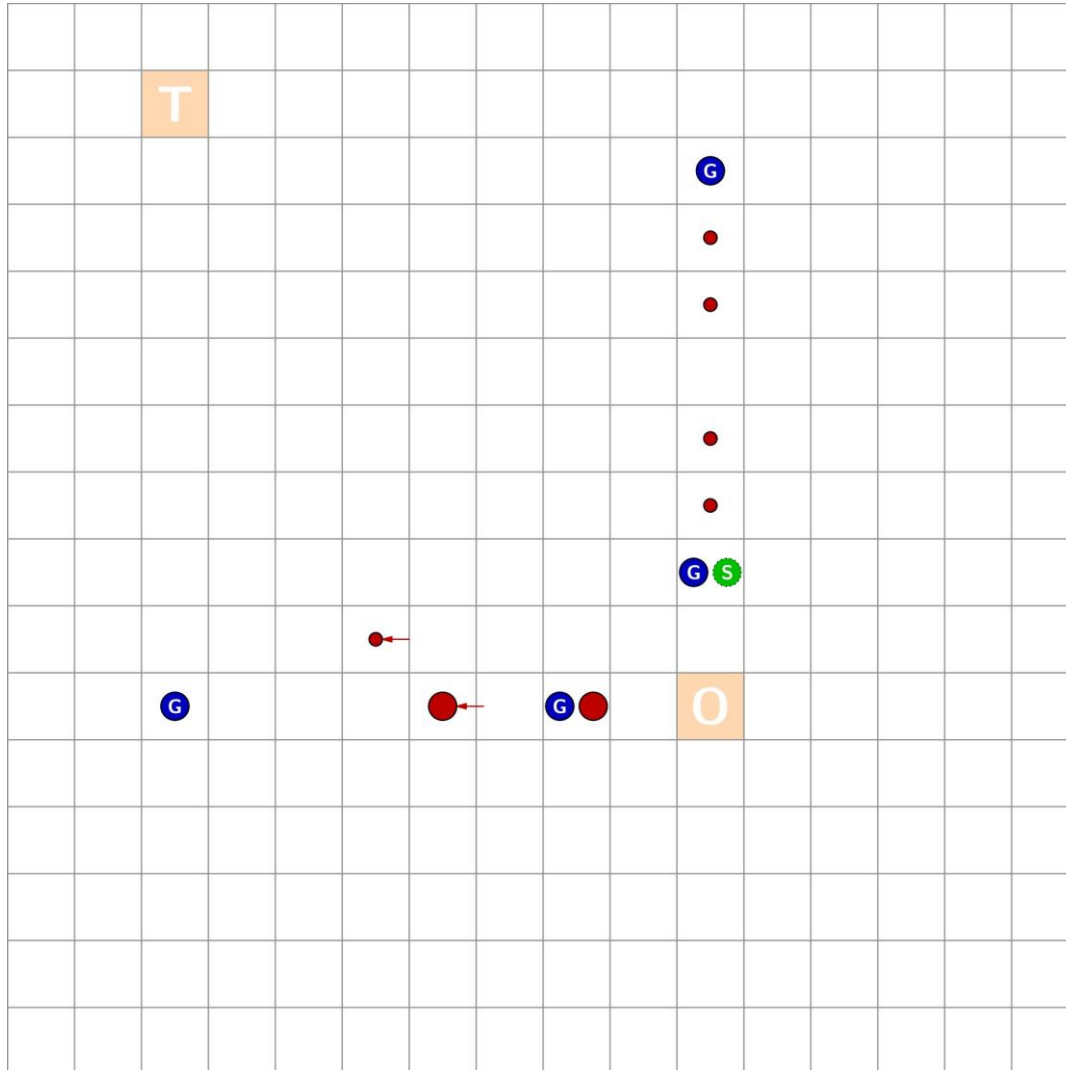
Robust Search



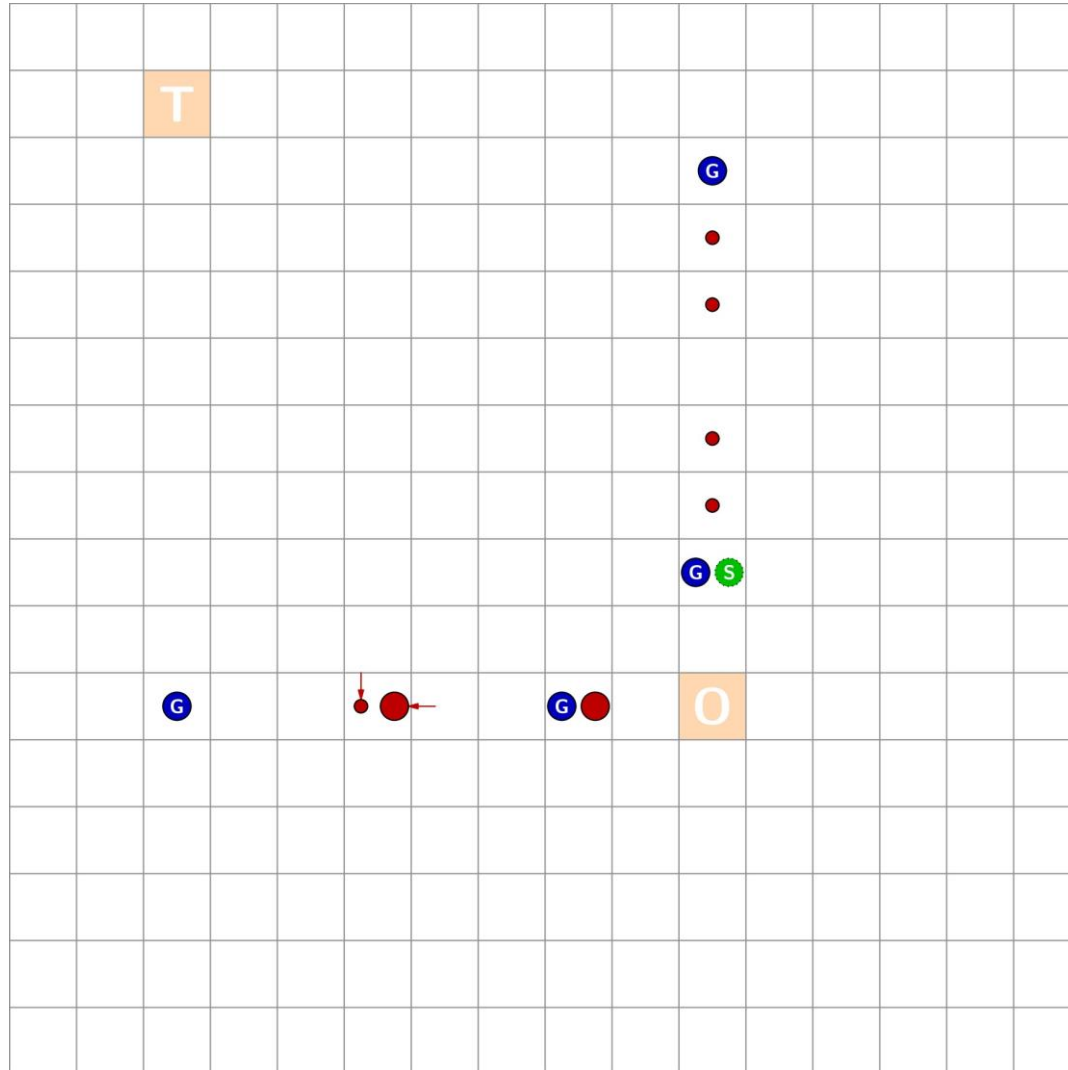
Robust Search



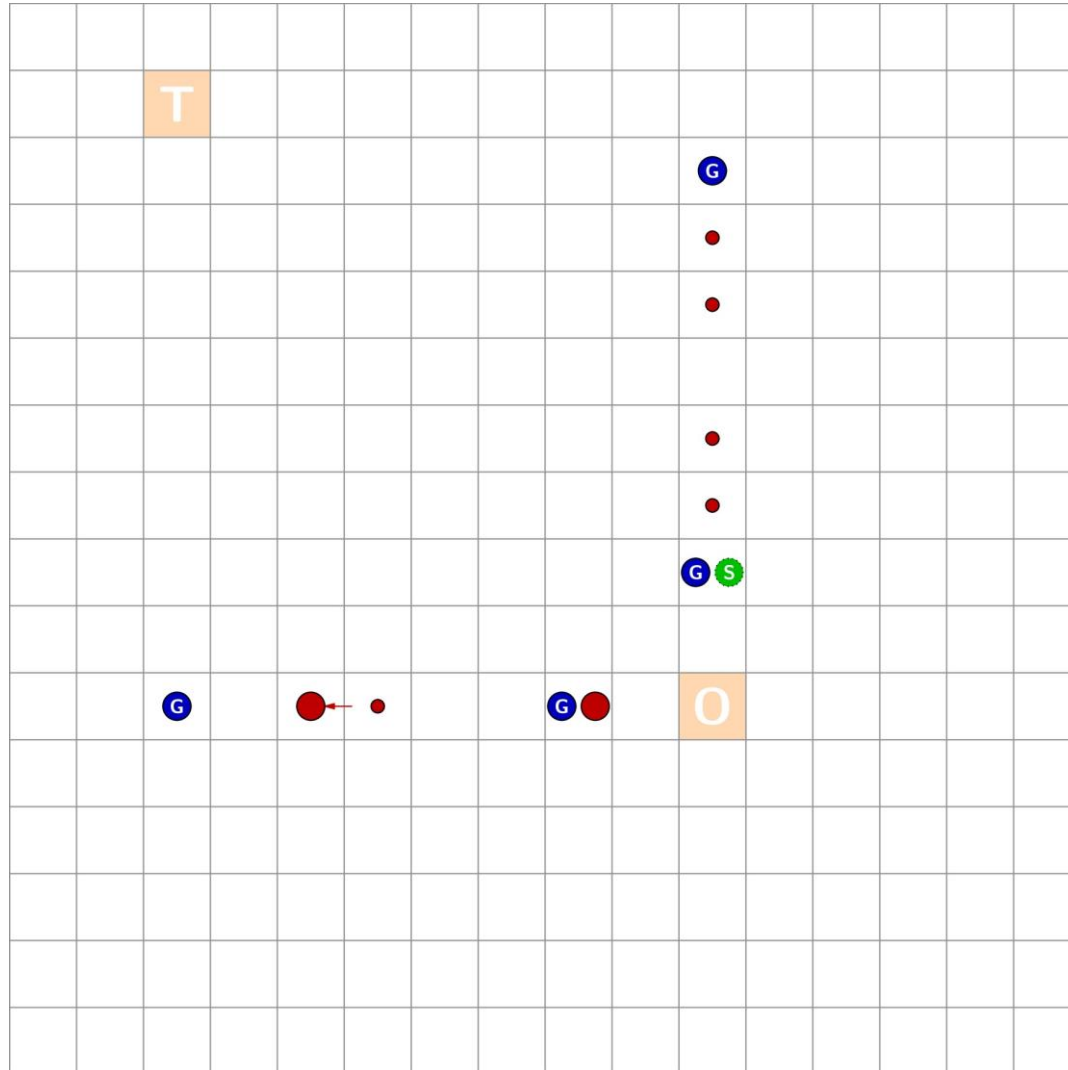
Robust Search



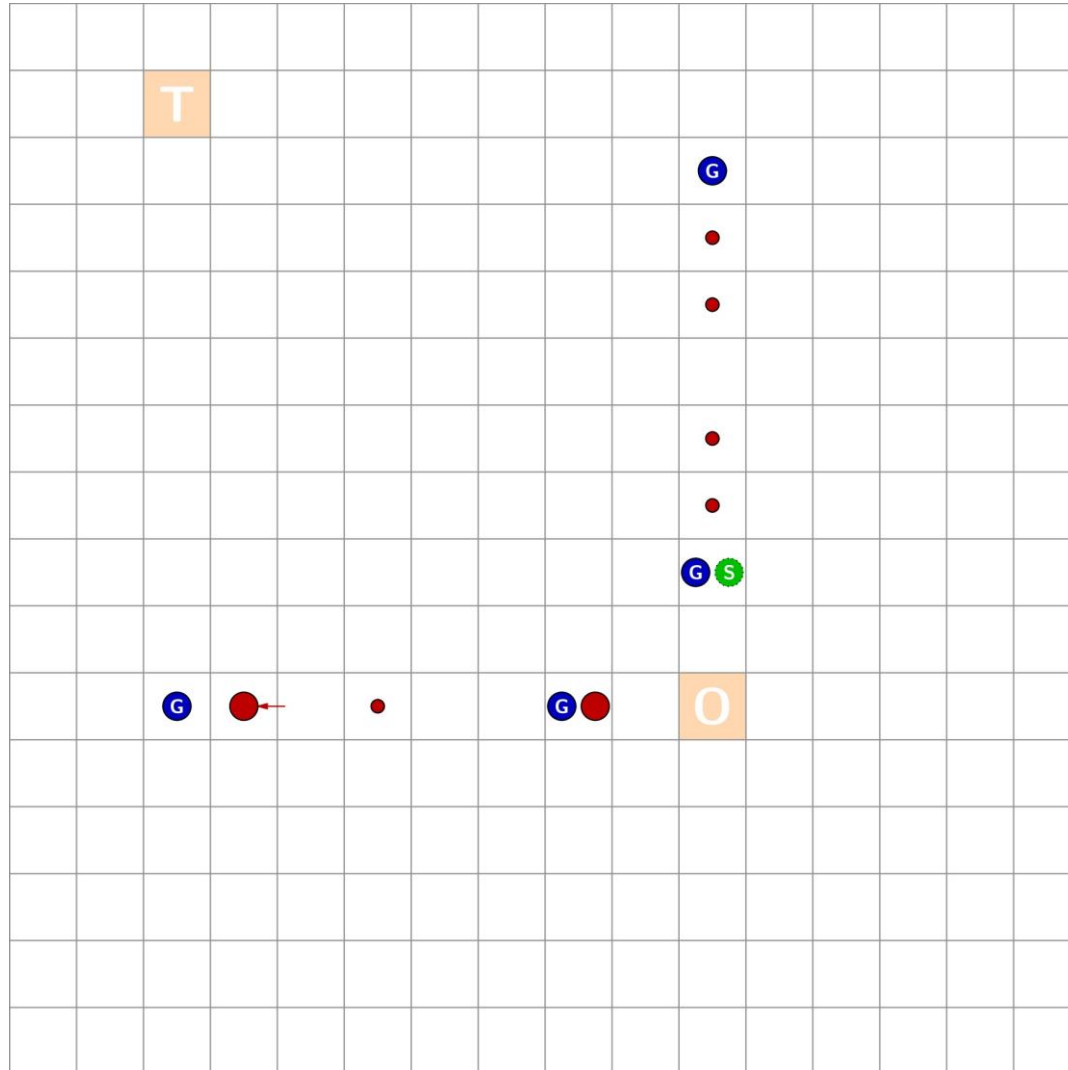
Robust Search



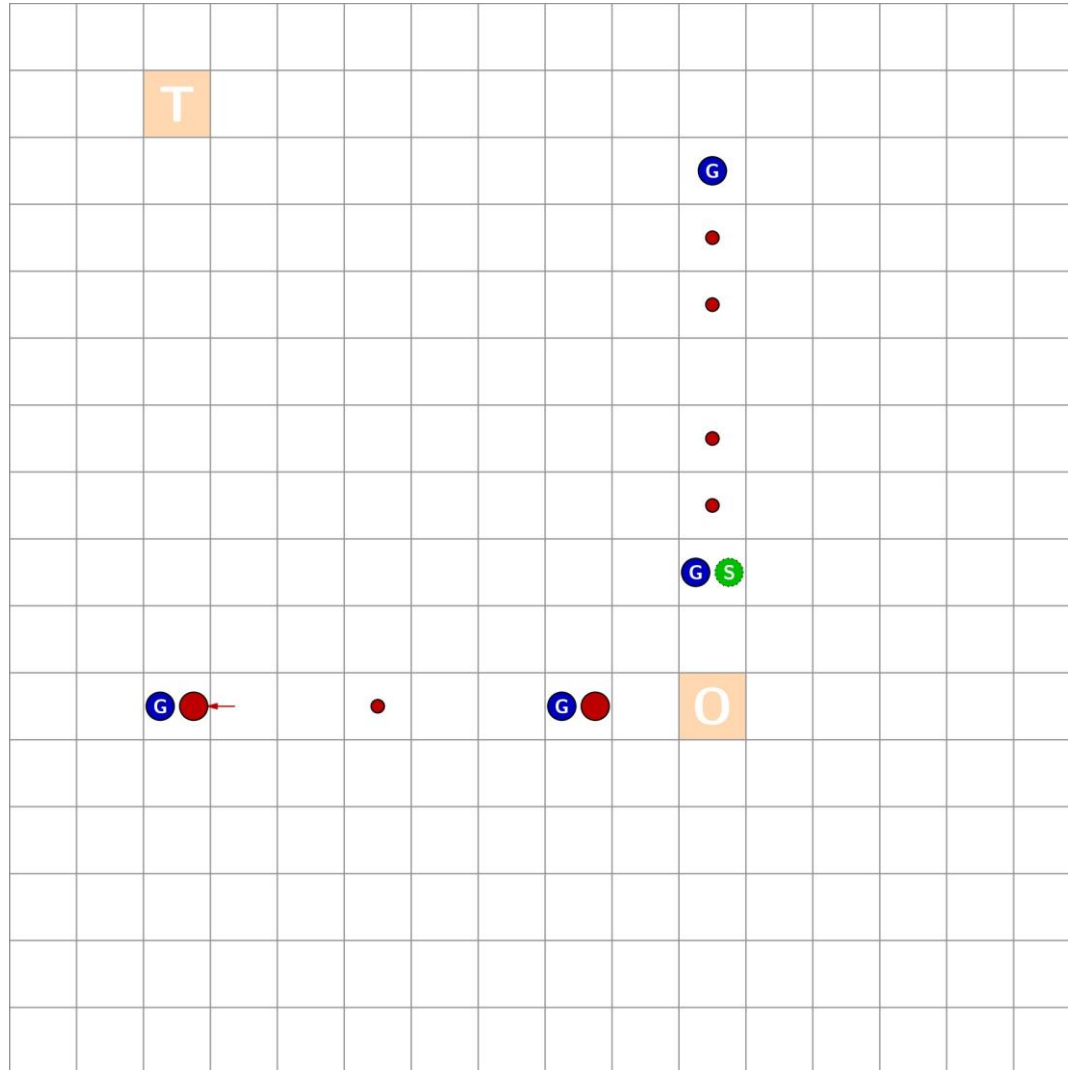
Robust Search



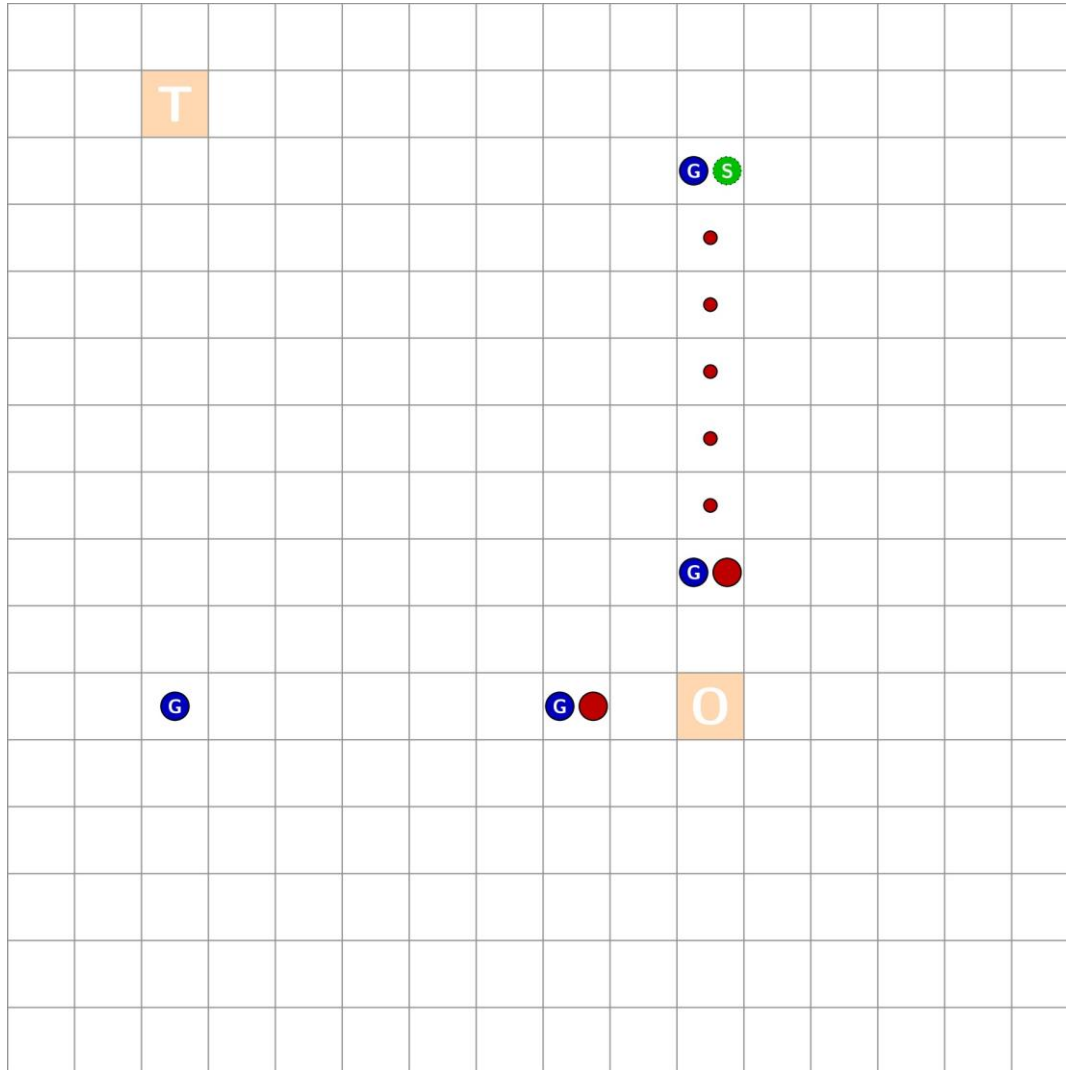
Robust Search



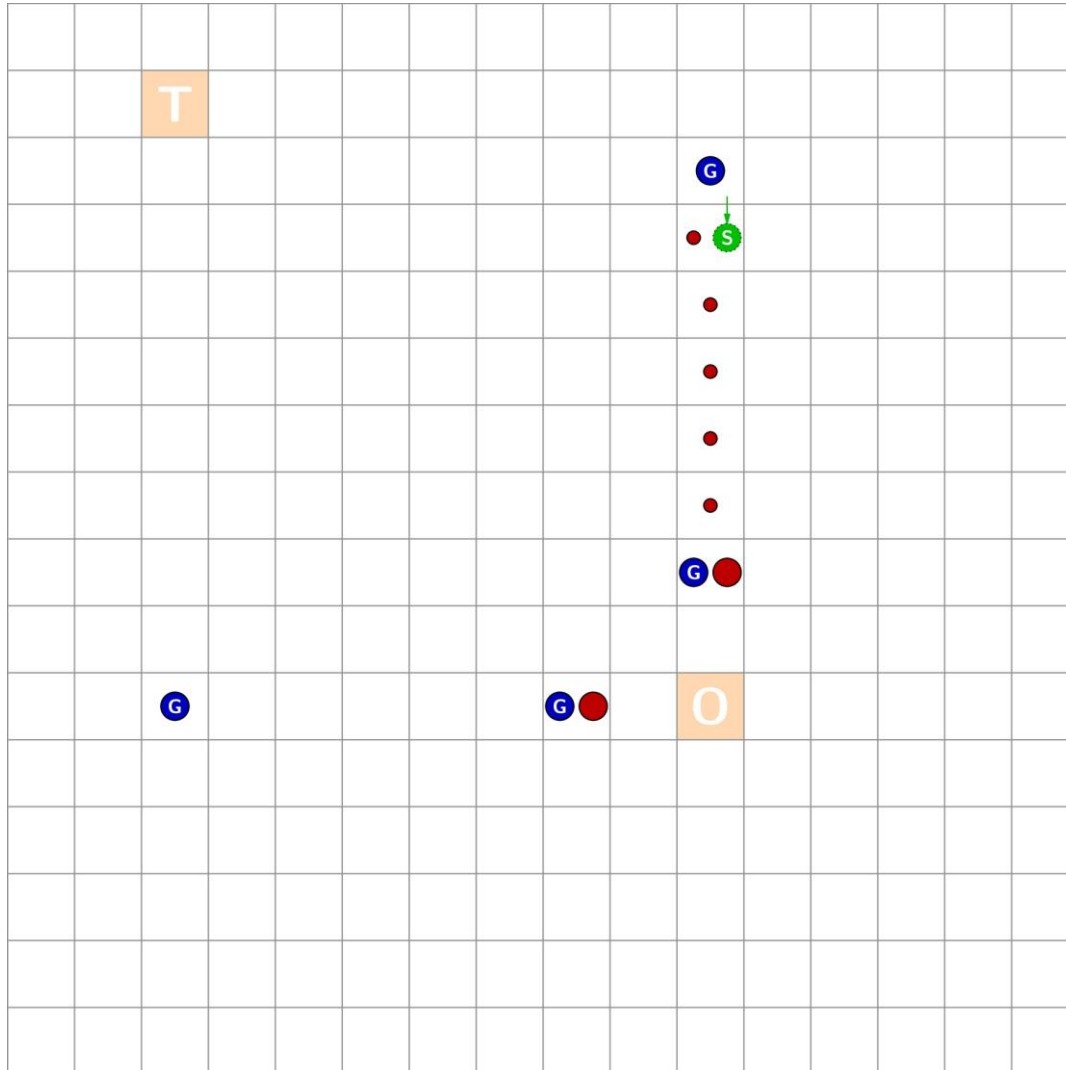
Robust Search



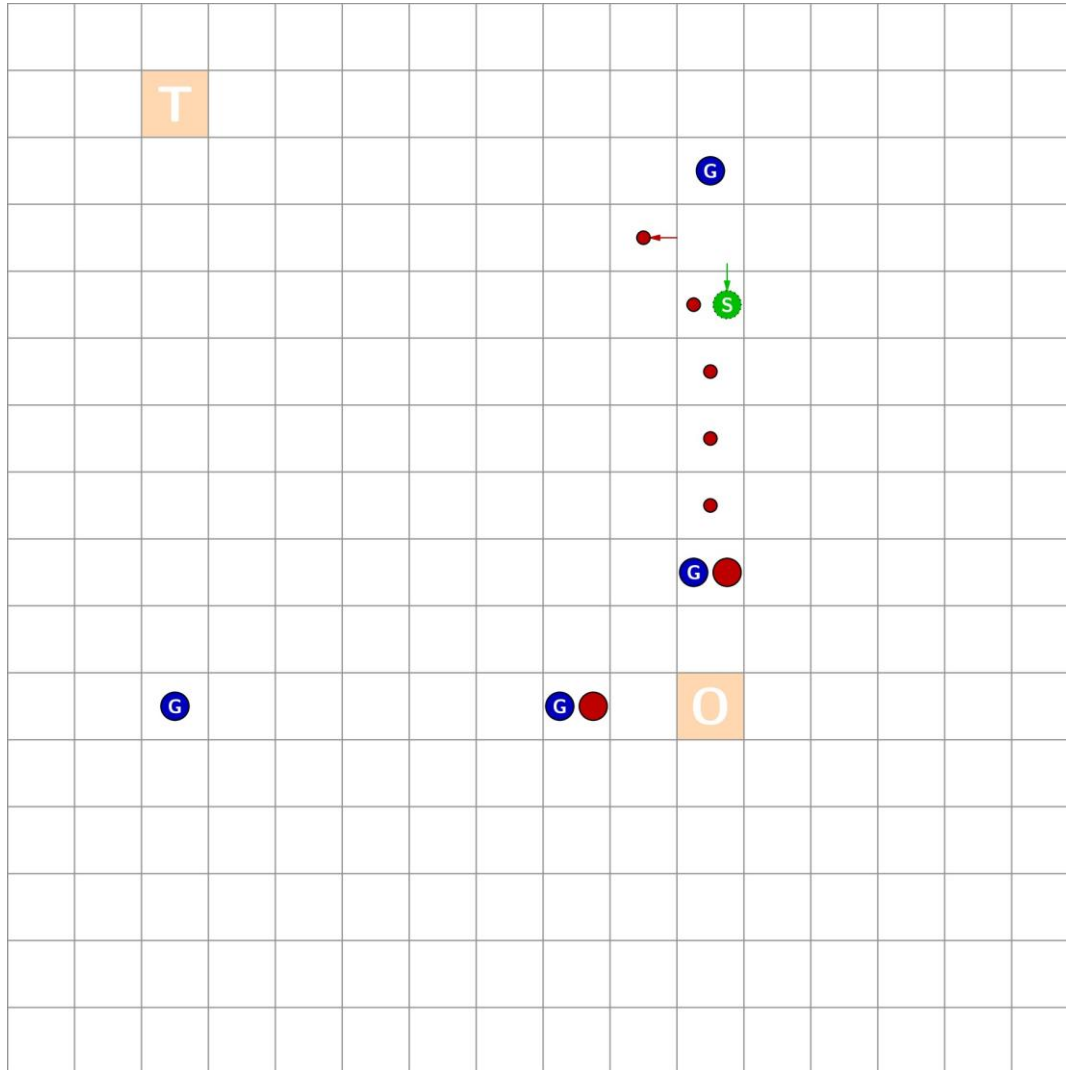
Robust Search



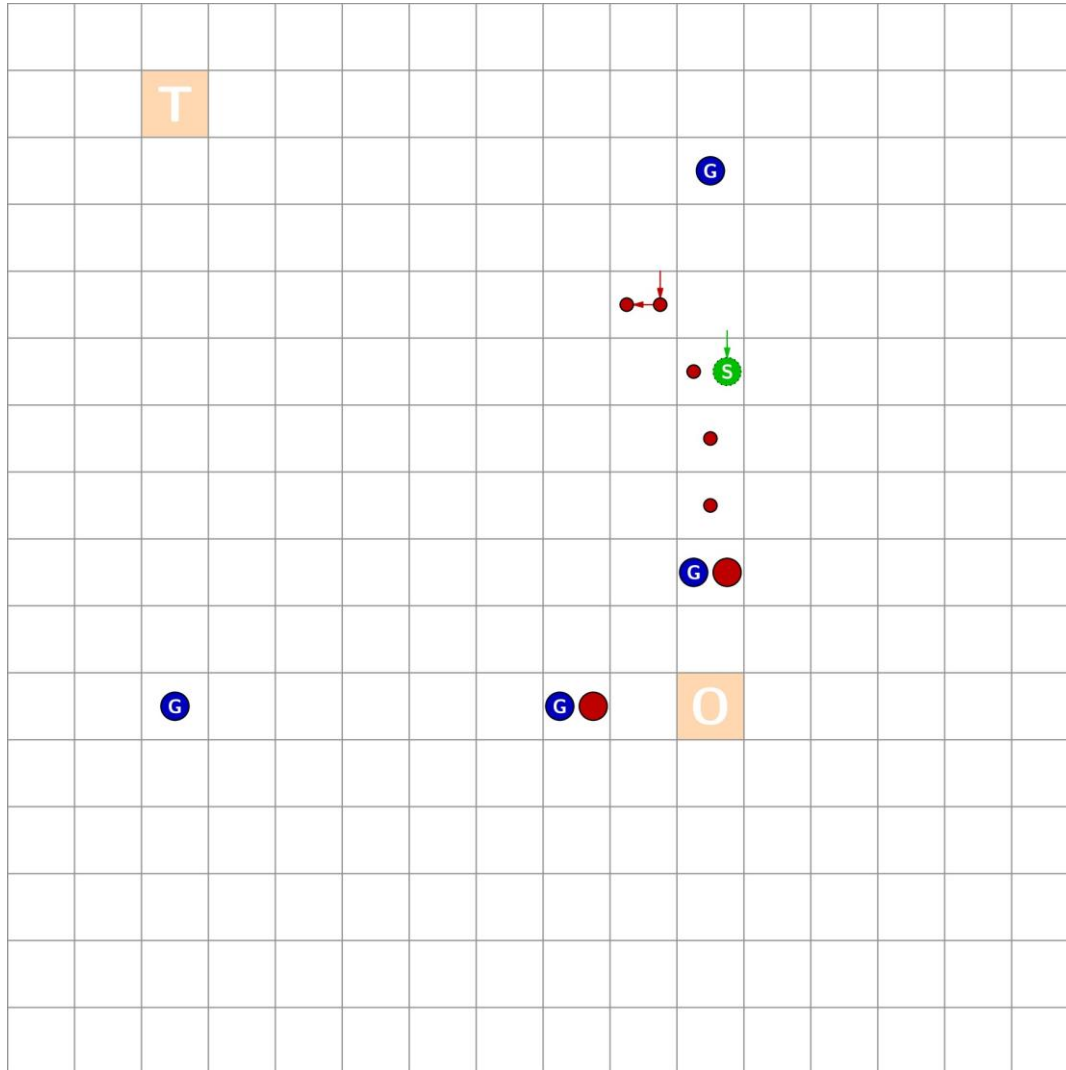
Robust Search



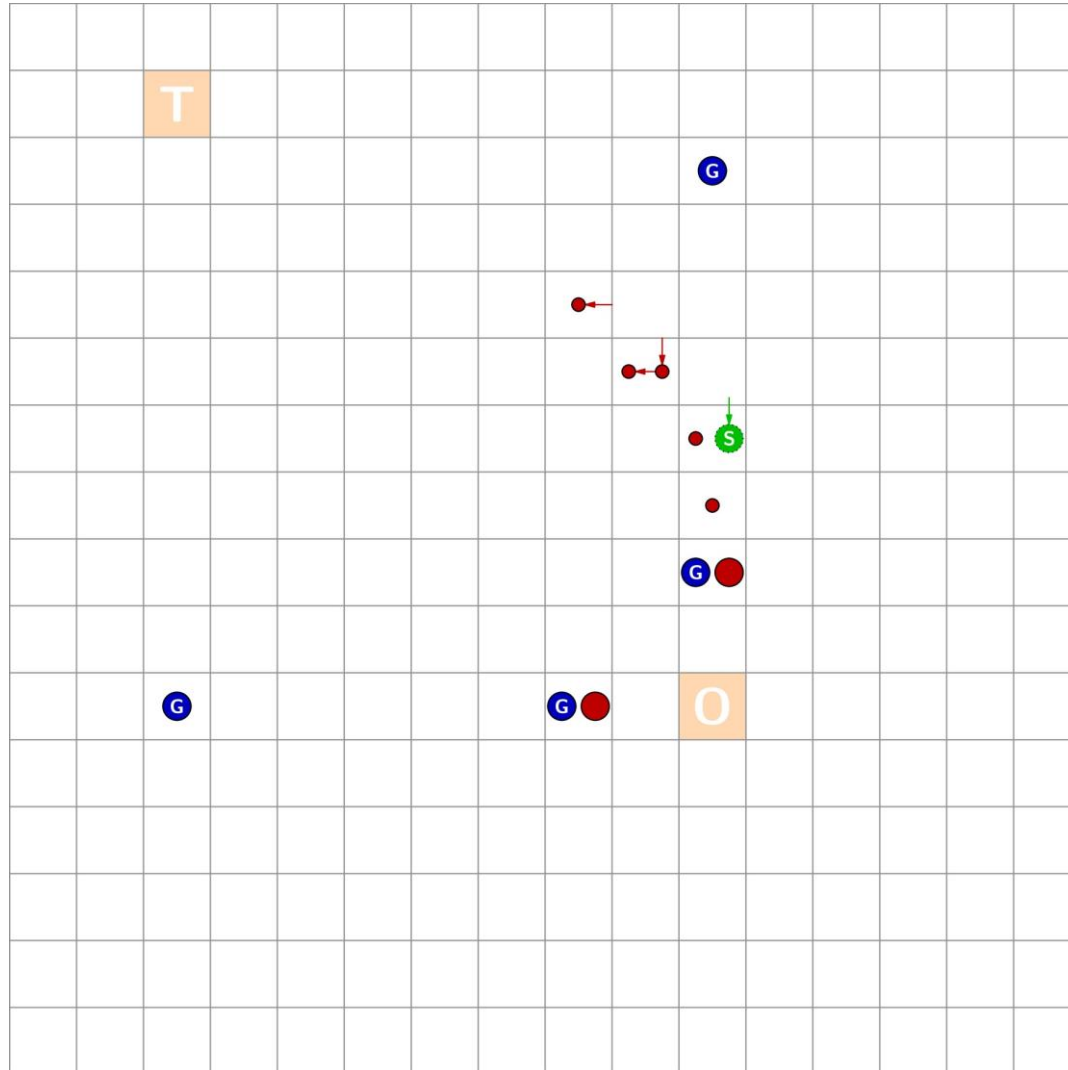
Robust Search



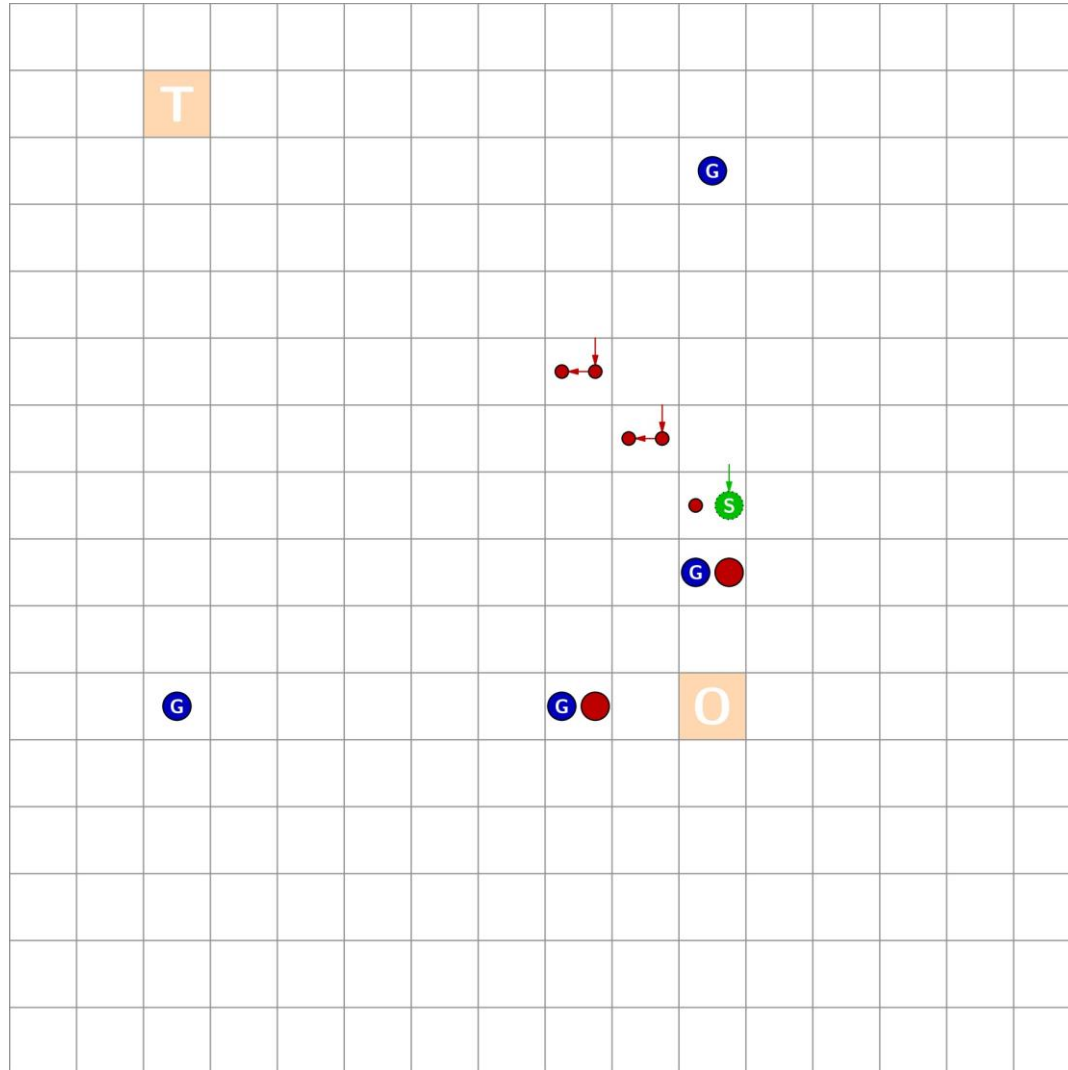
Robust Search



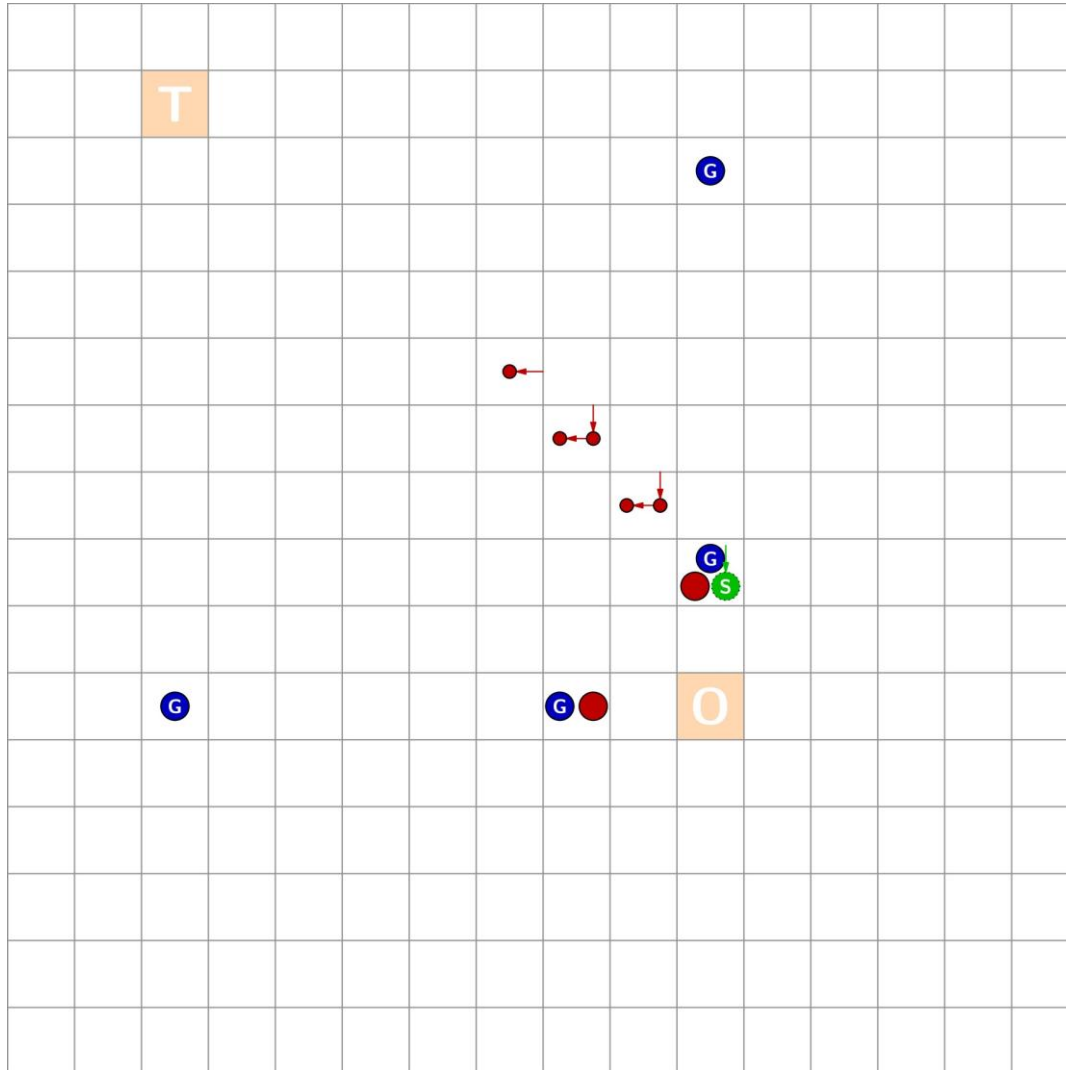
Robust Search



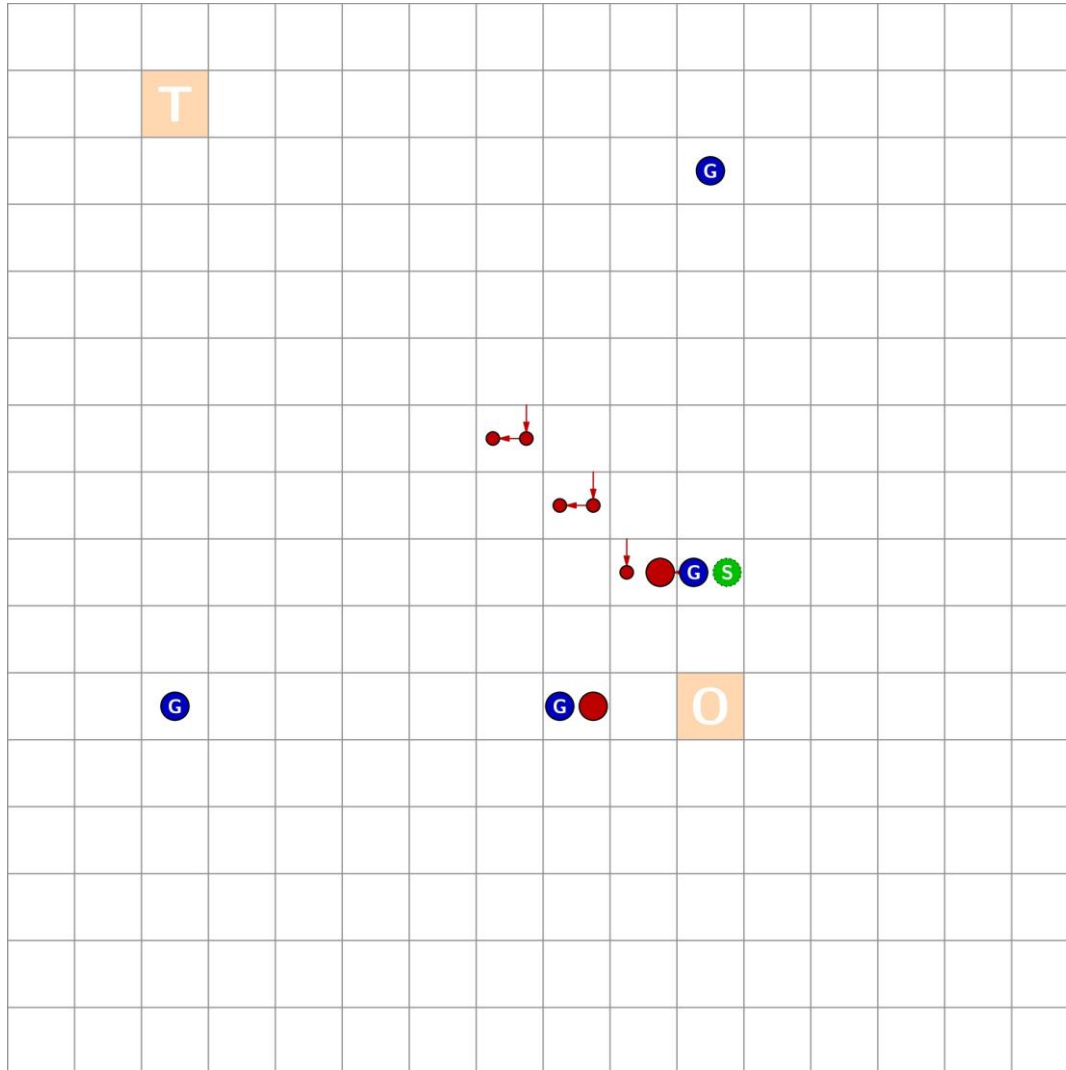
Robust Search



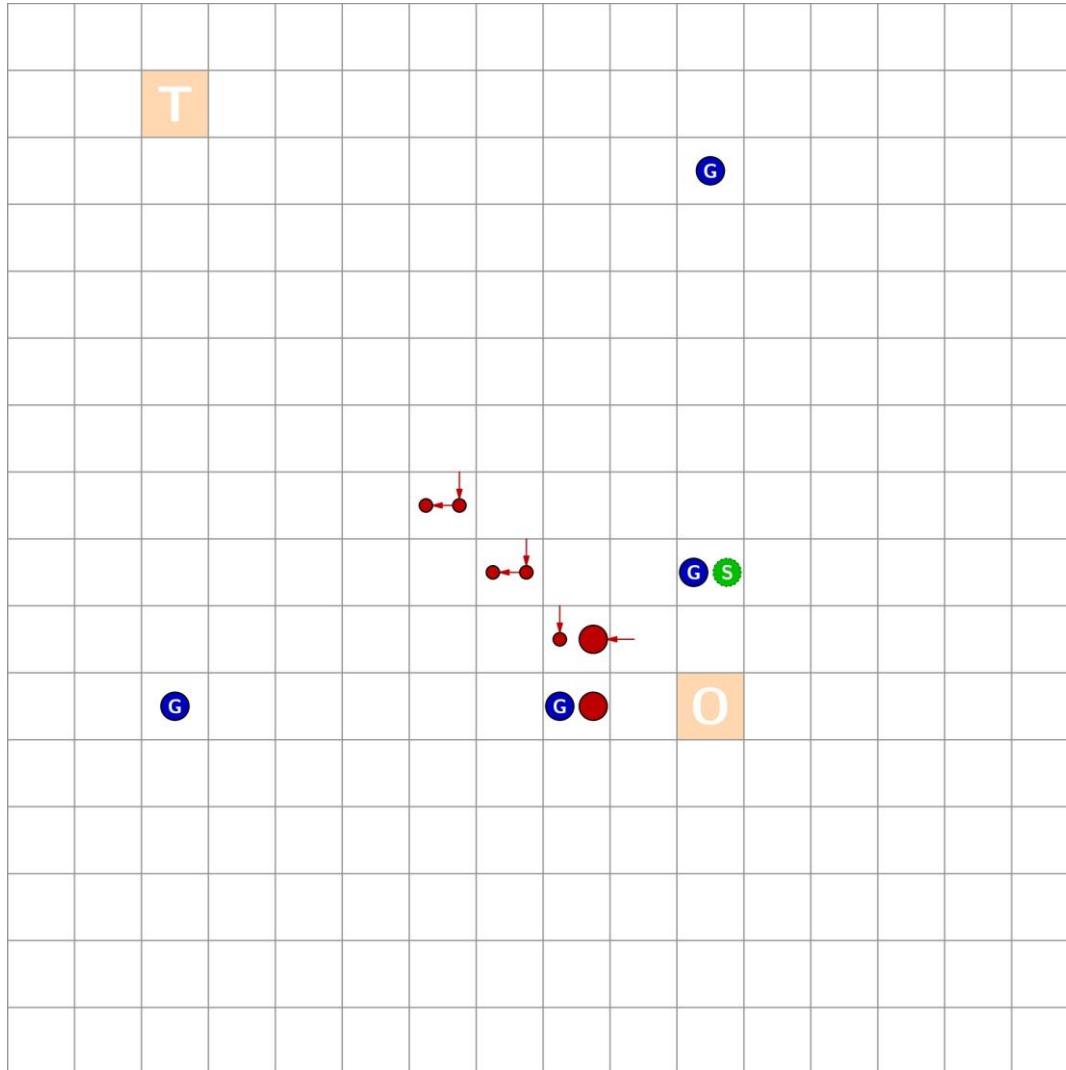
Robust Search



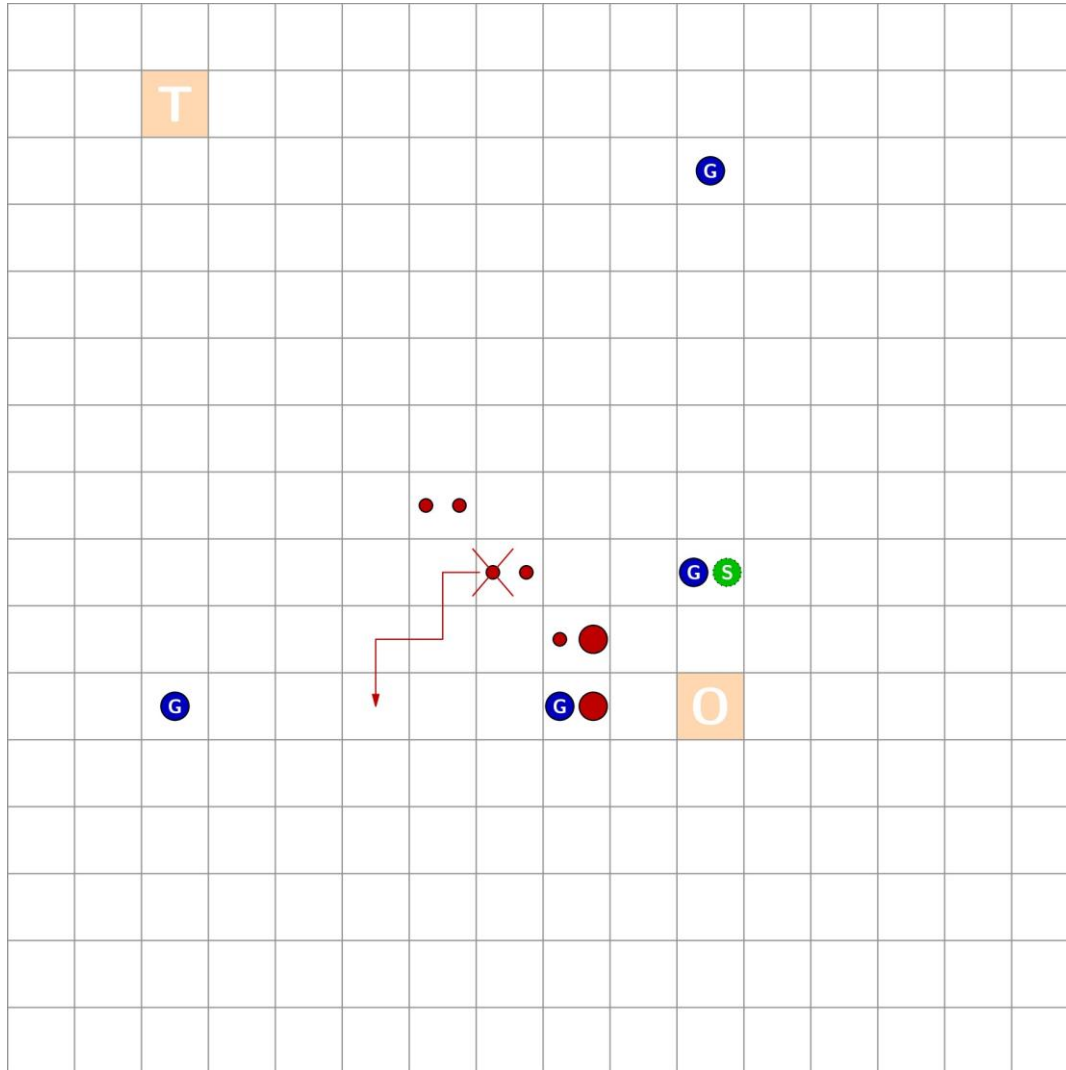
Robust Search



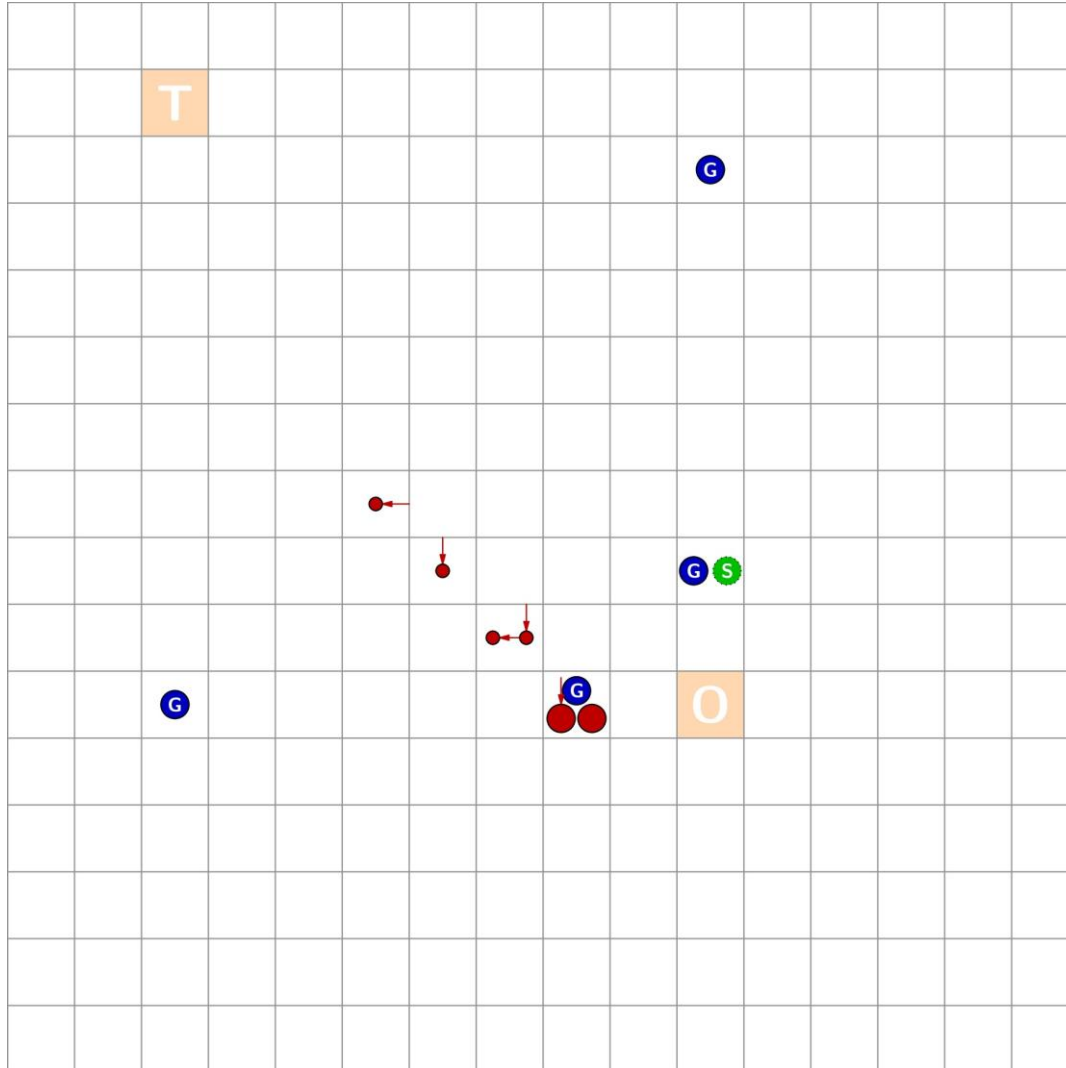
Robust Search



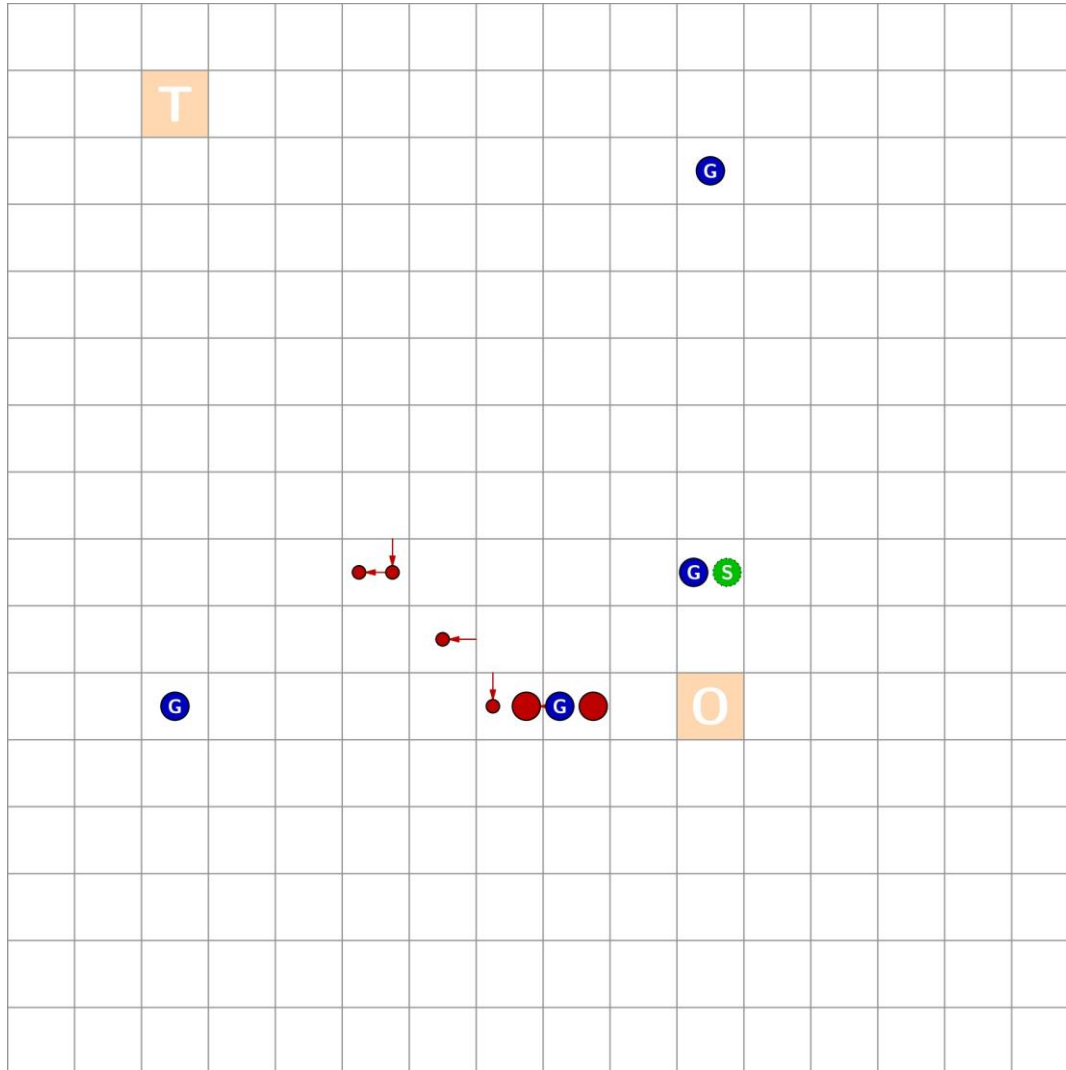
Robust Search



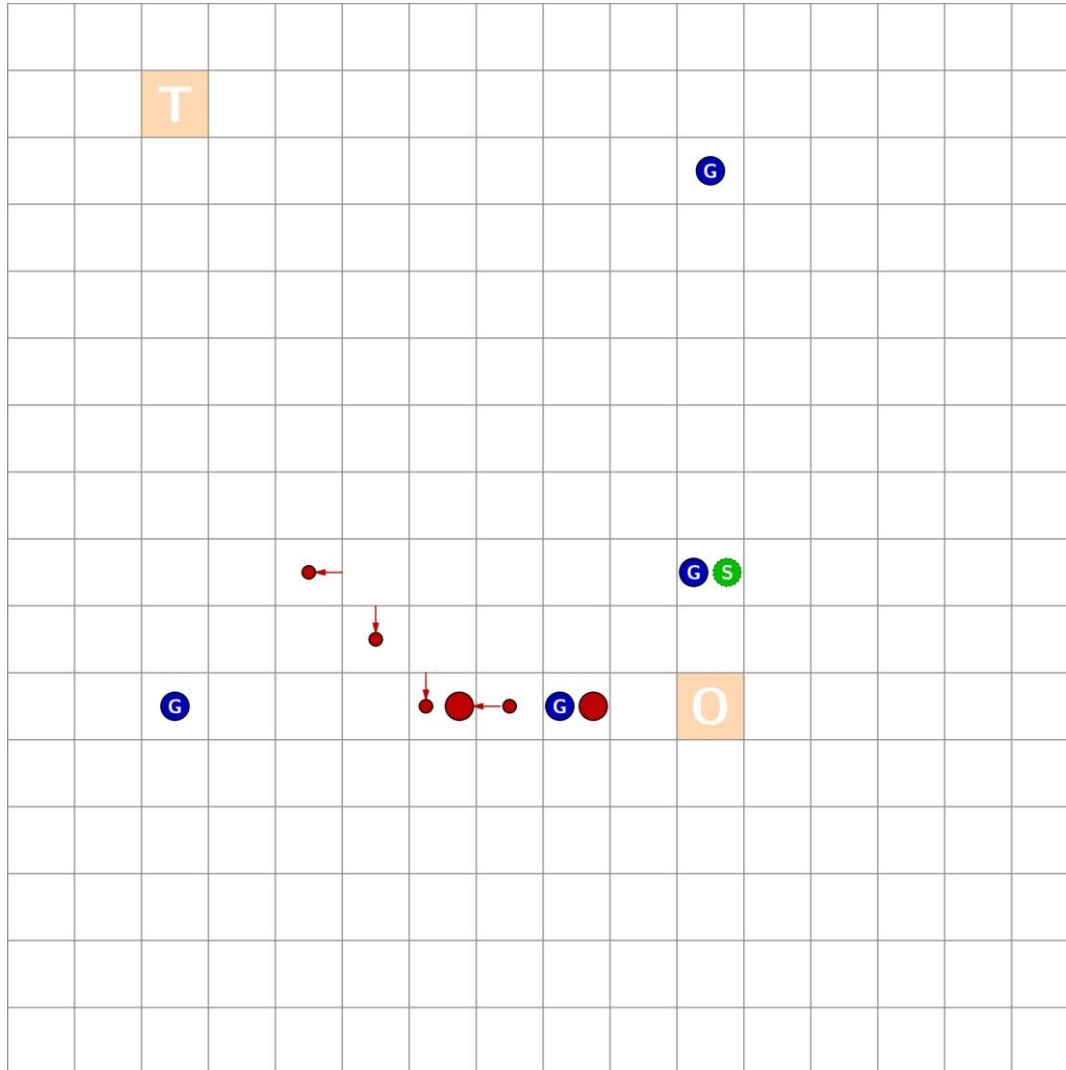
Robust Search



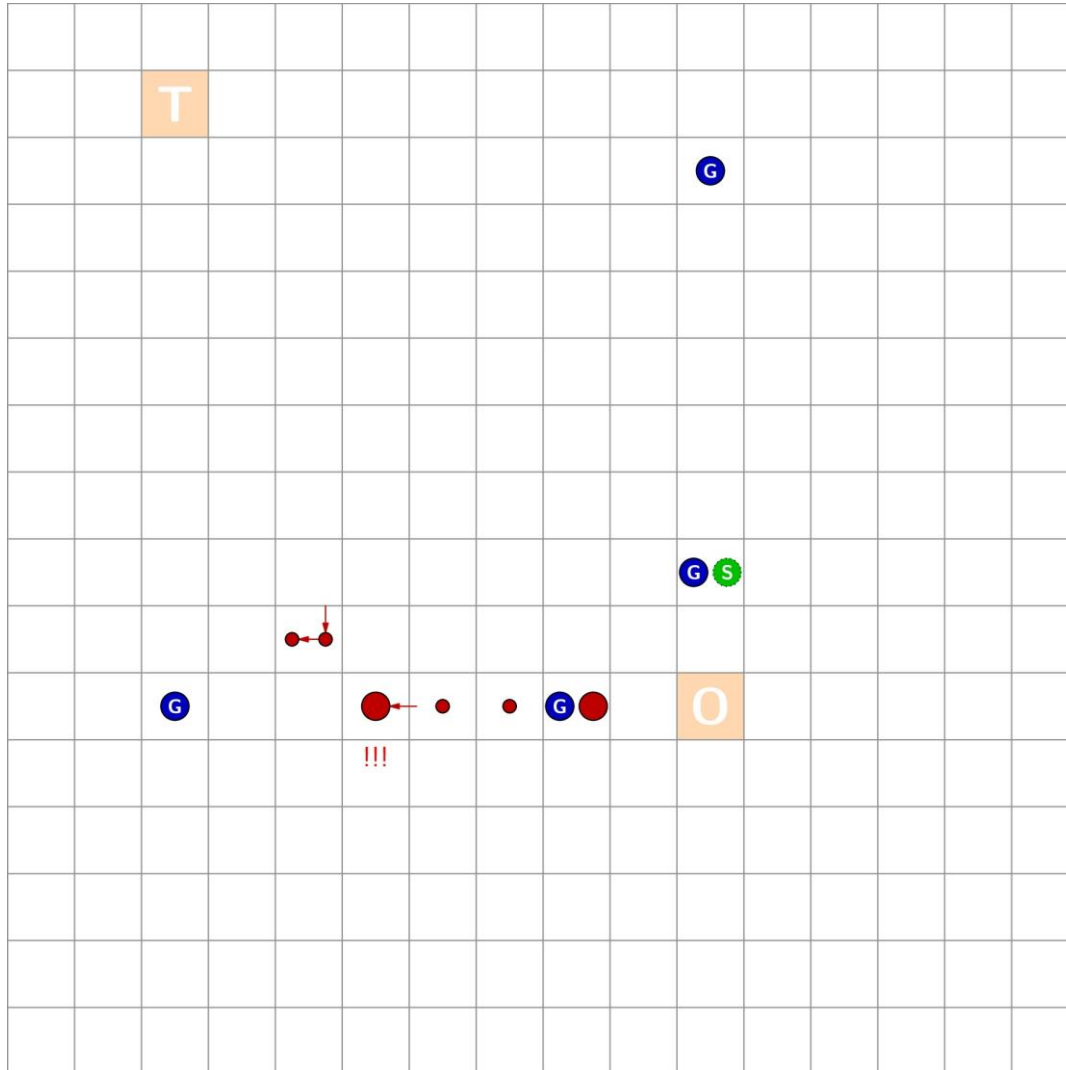
Robust Search



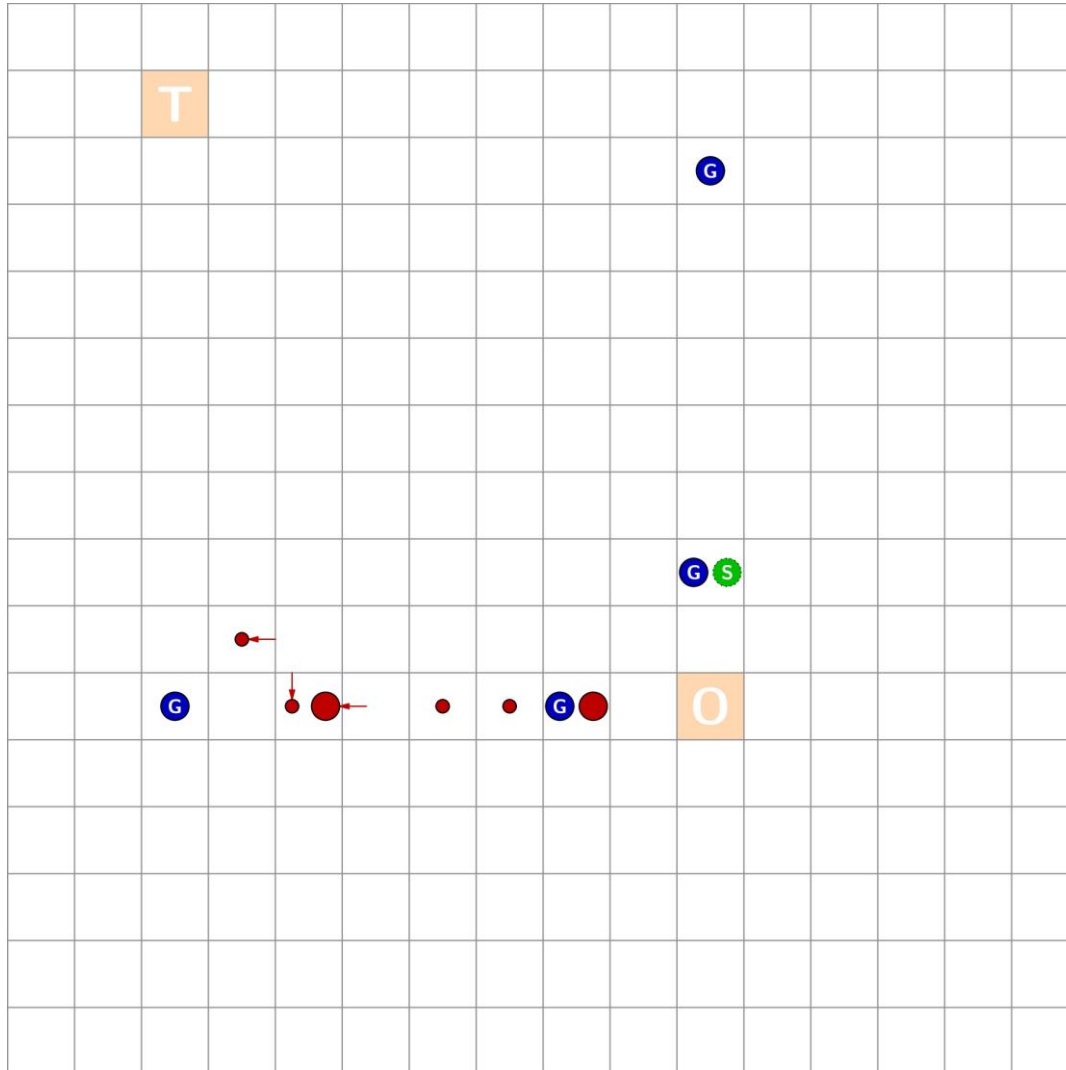
Robust Search



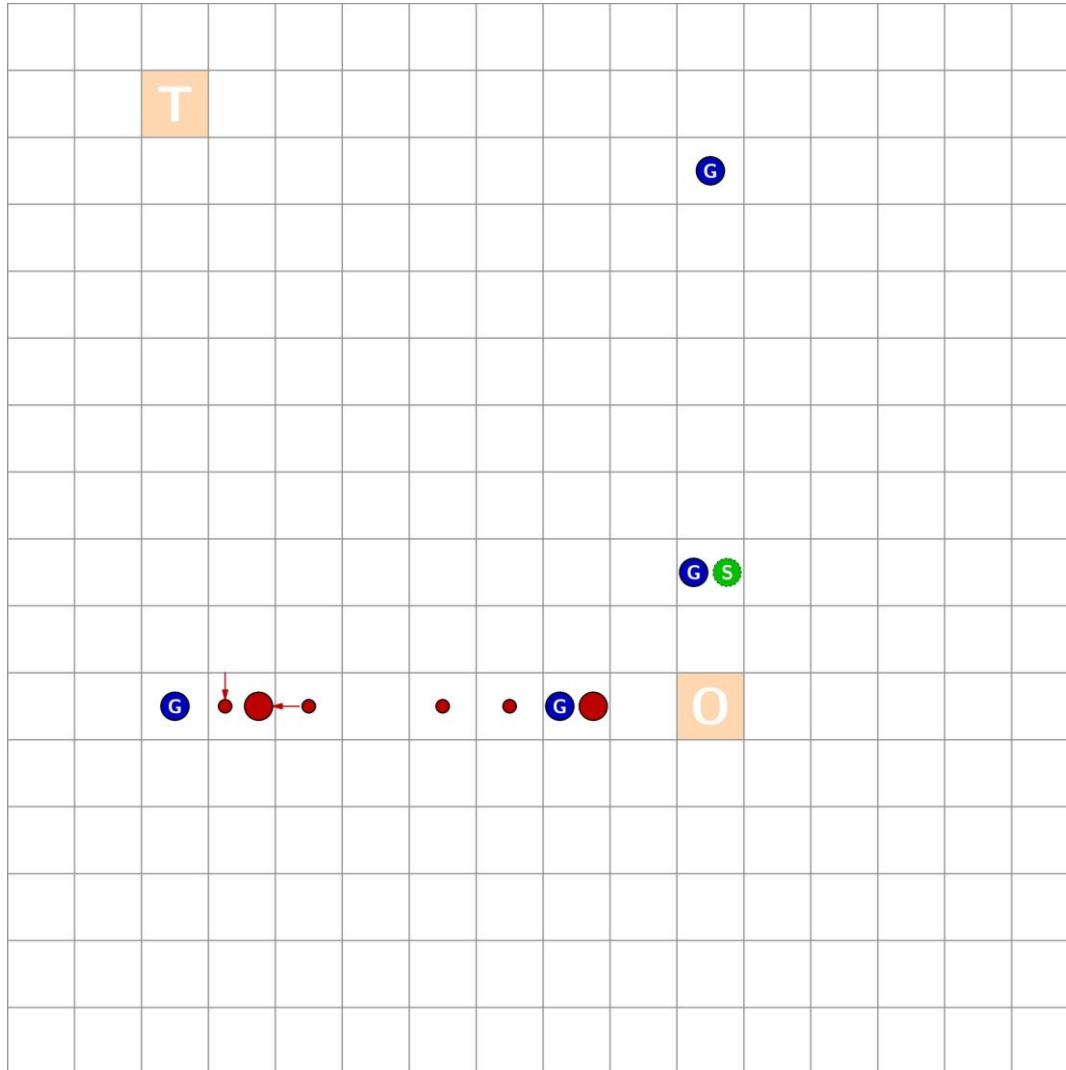
Robust Search



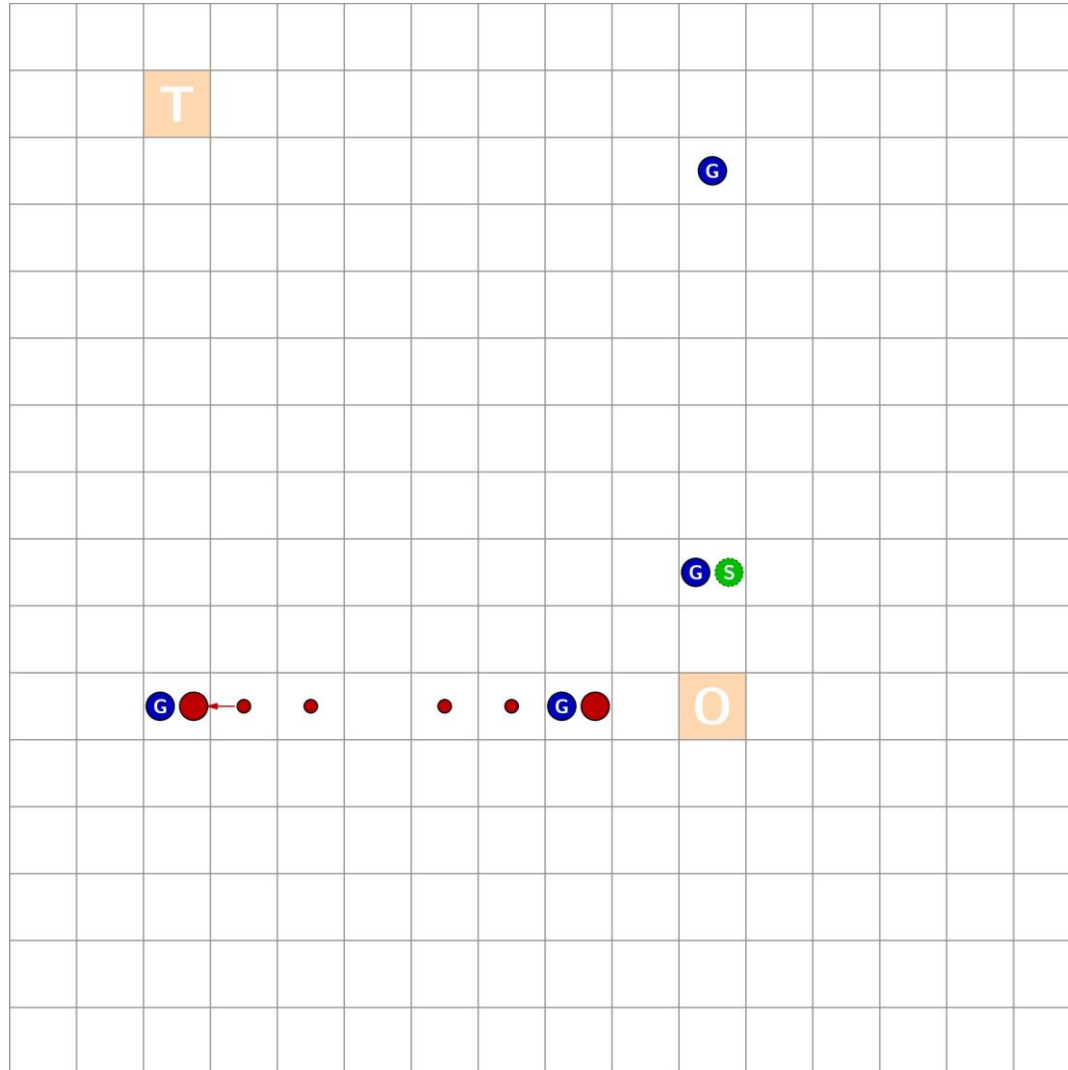
Robust Search



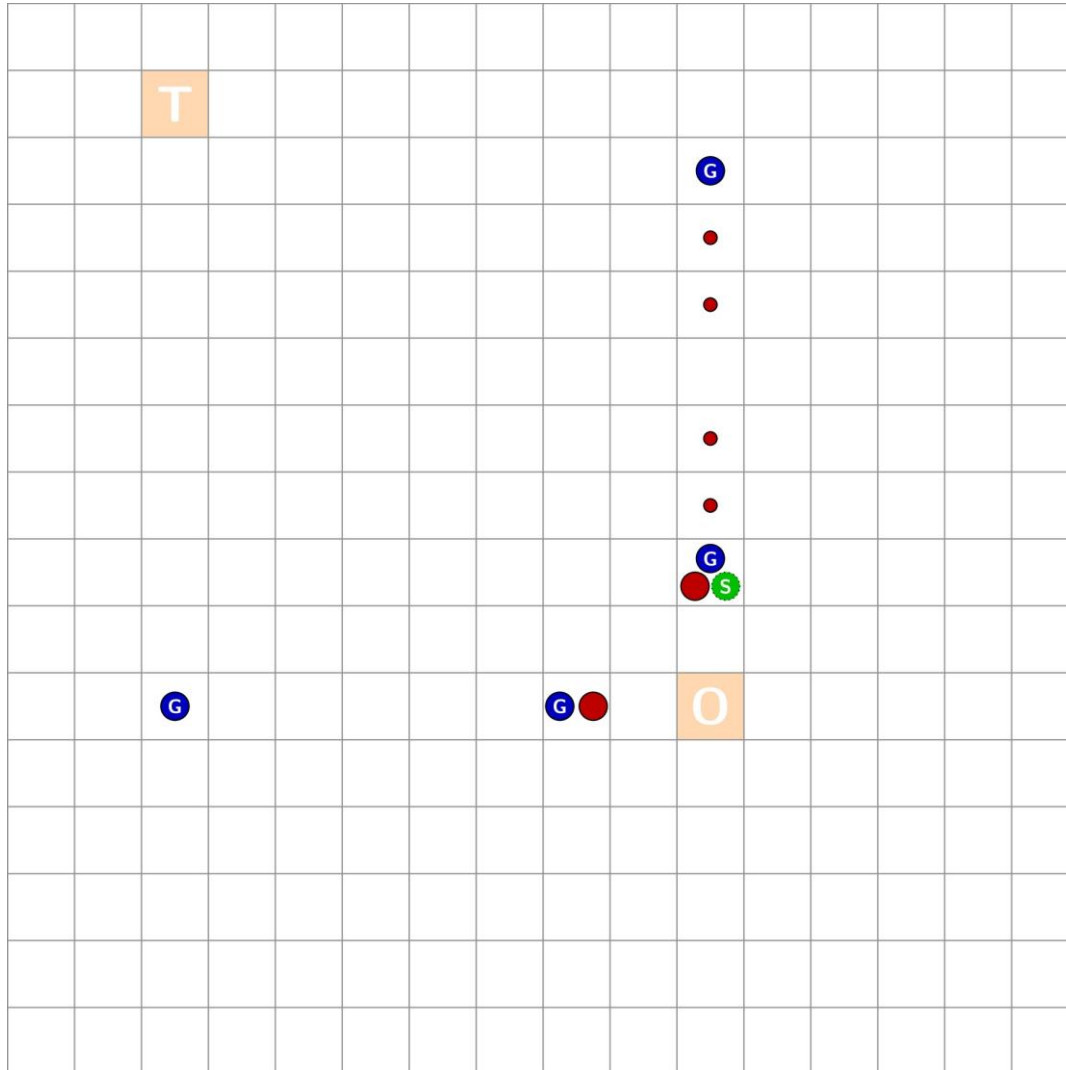
Robust Search



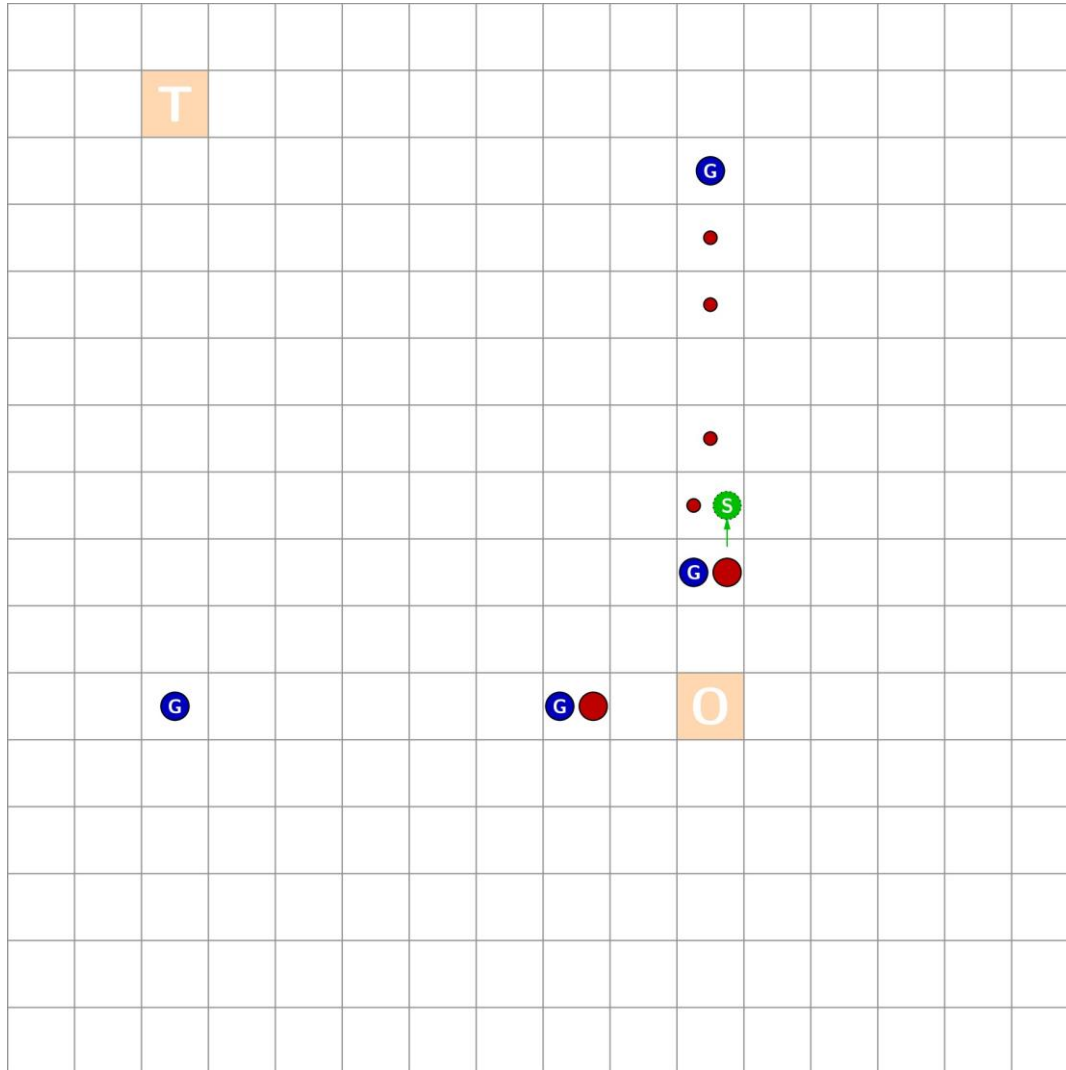
Robust Search



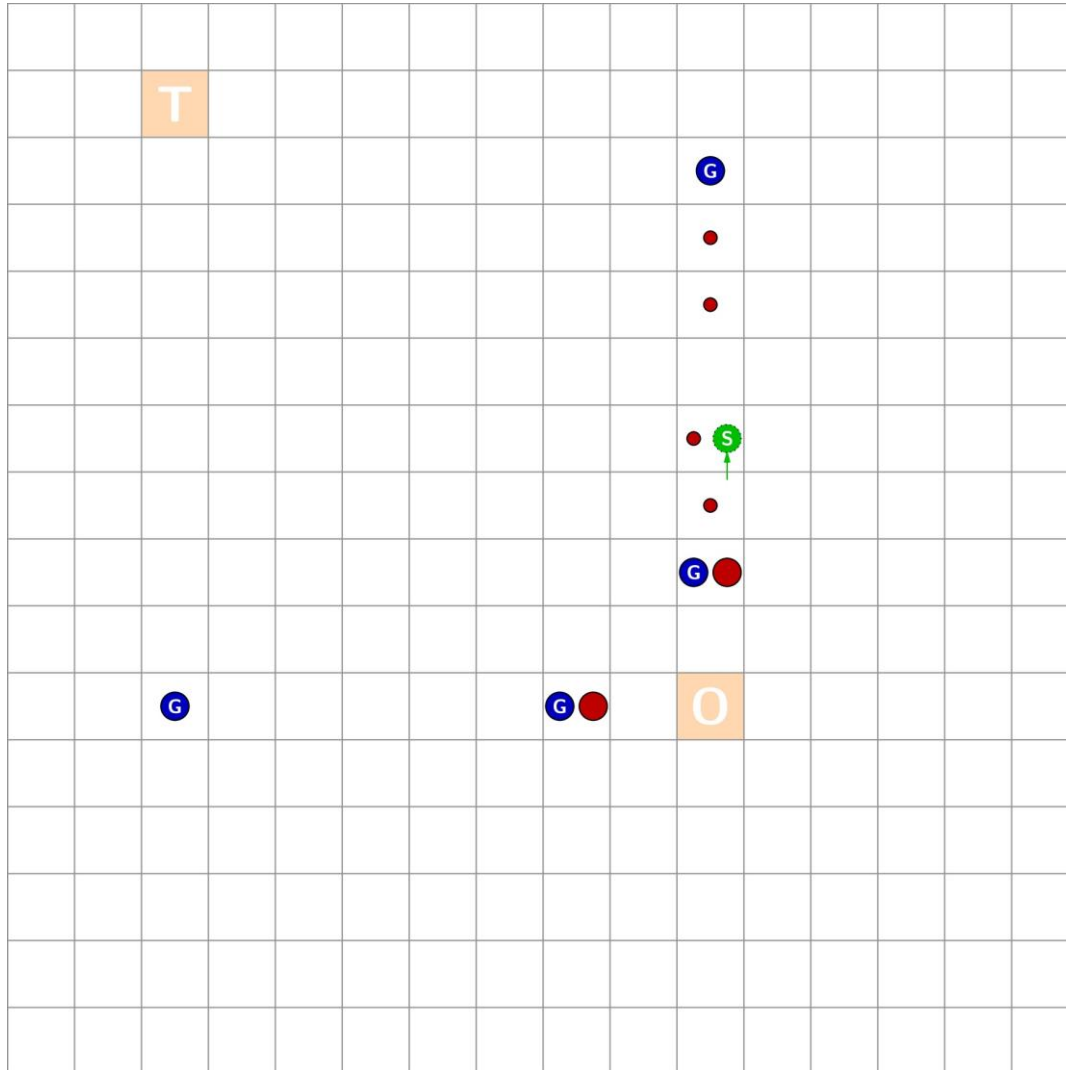
Robust Search



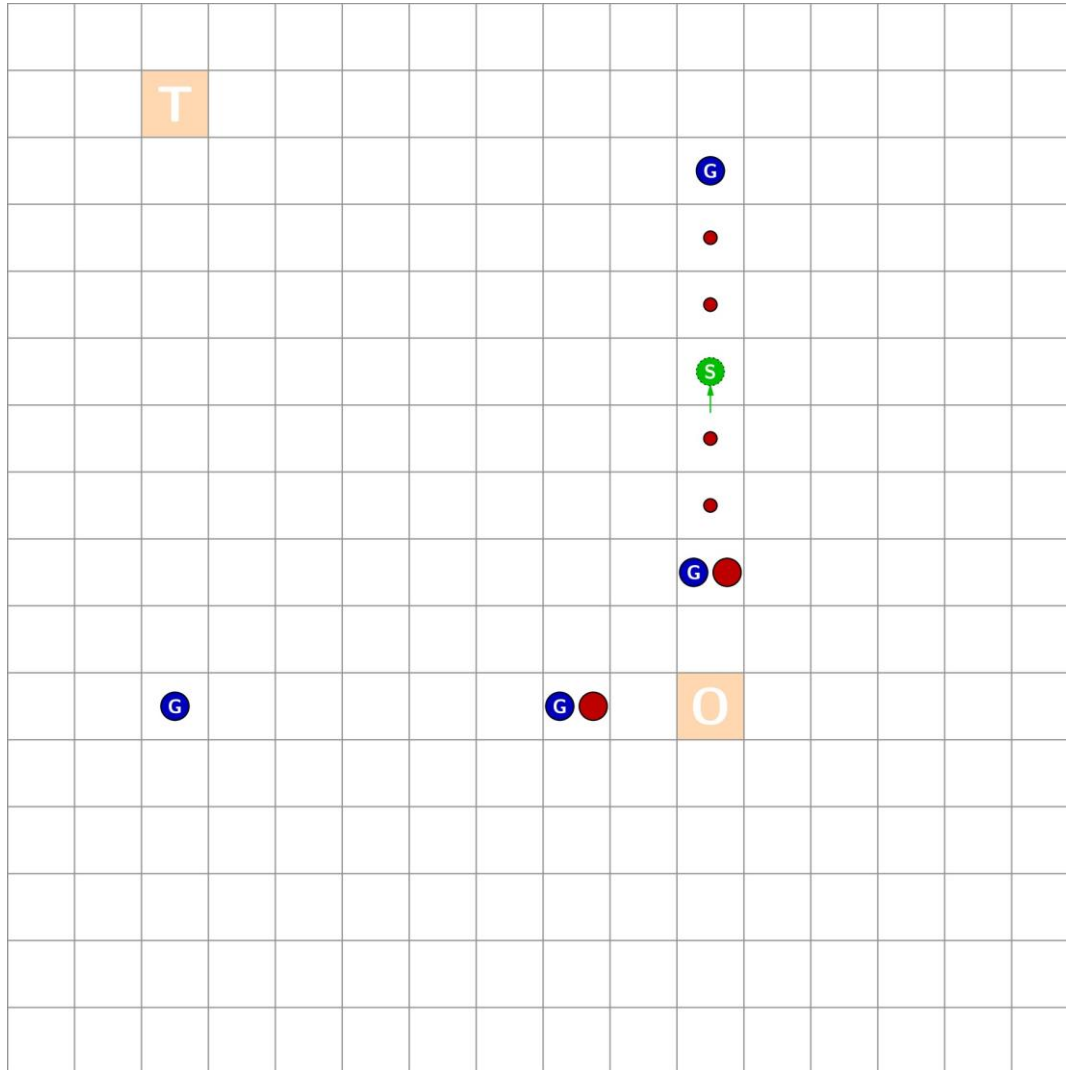
Robust Search



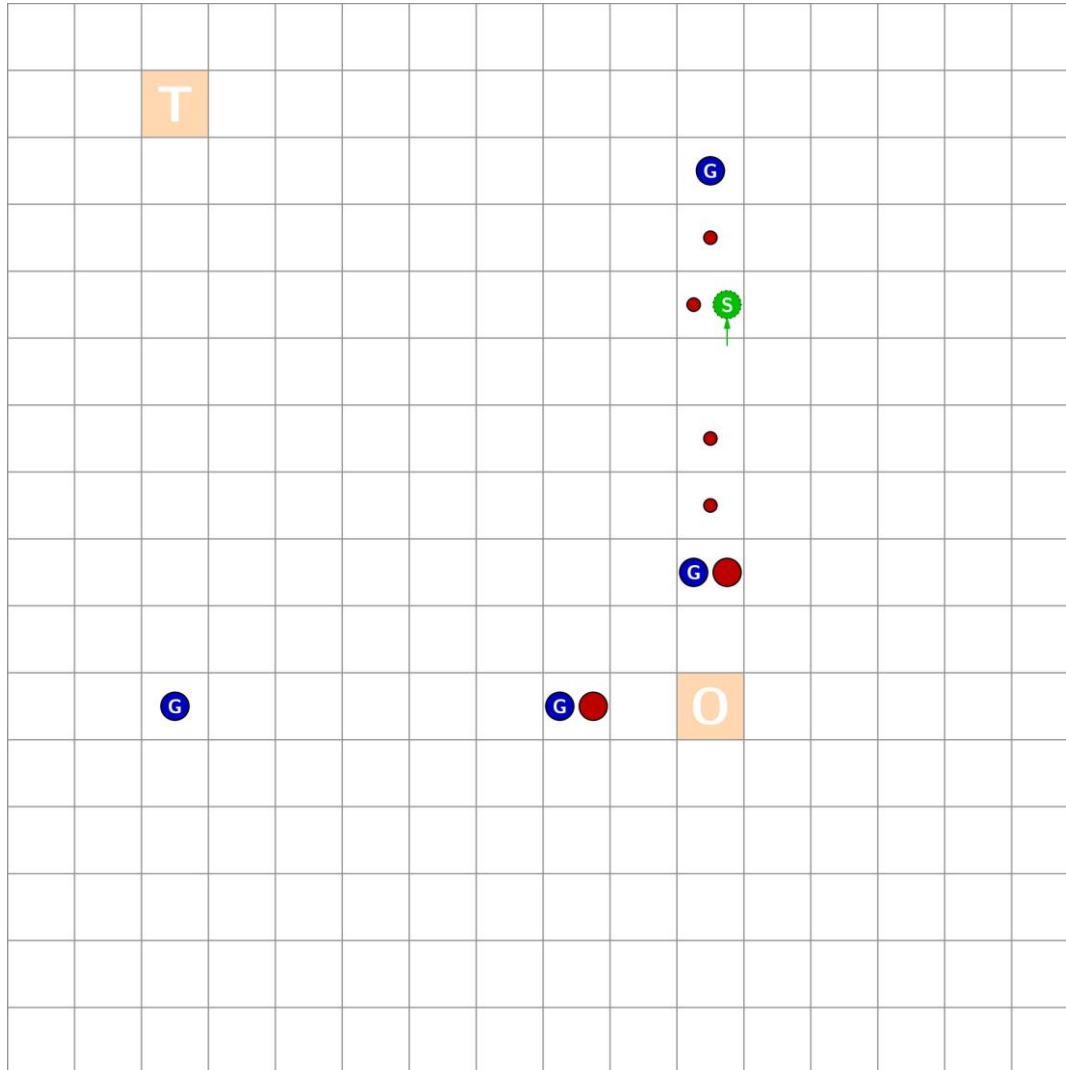
Robust Search



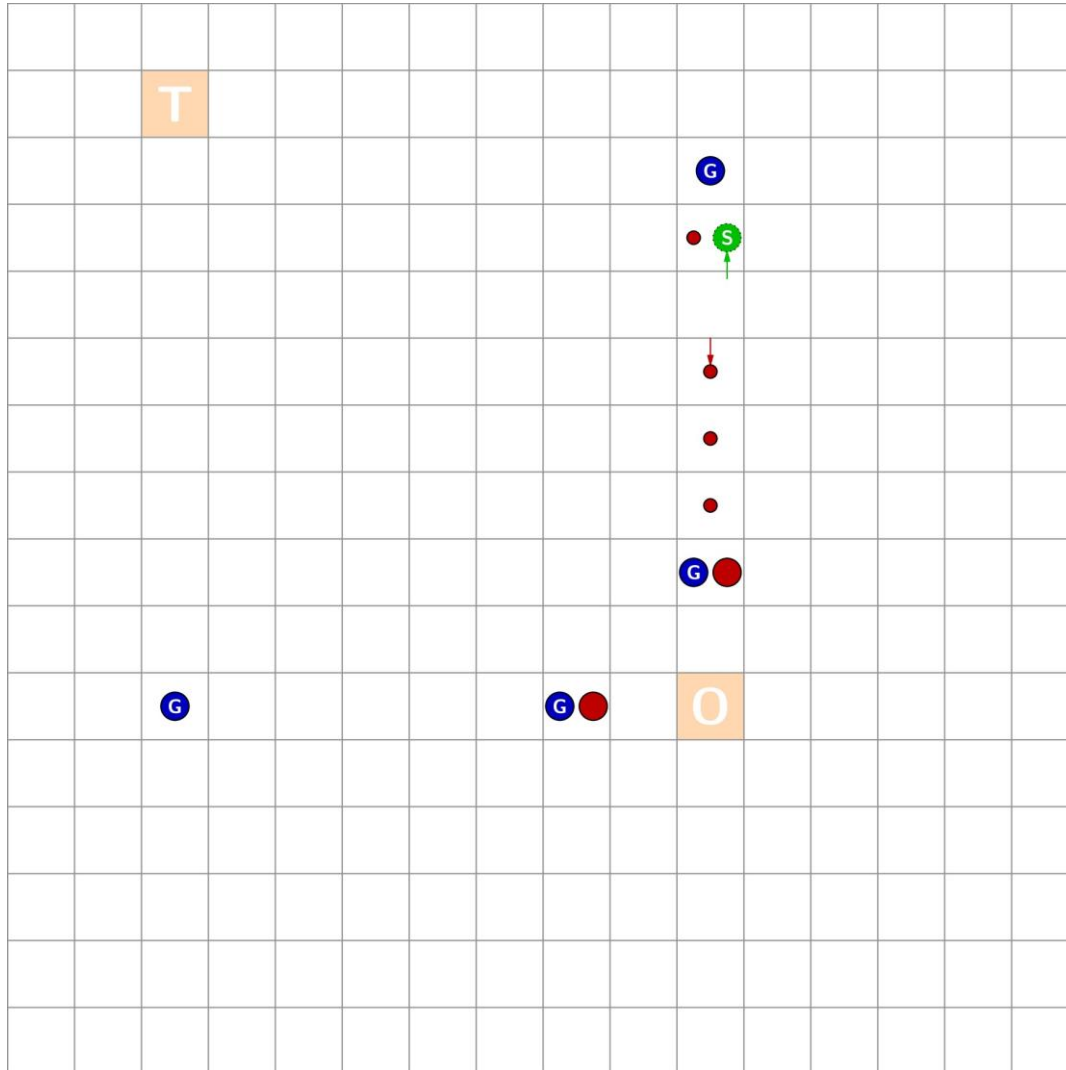
Robust Search



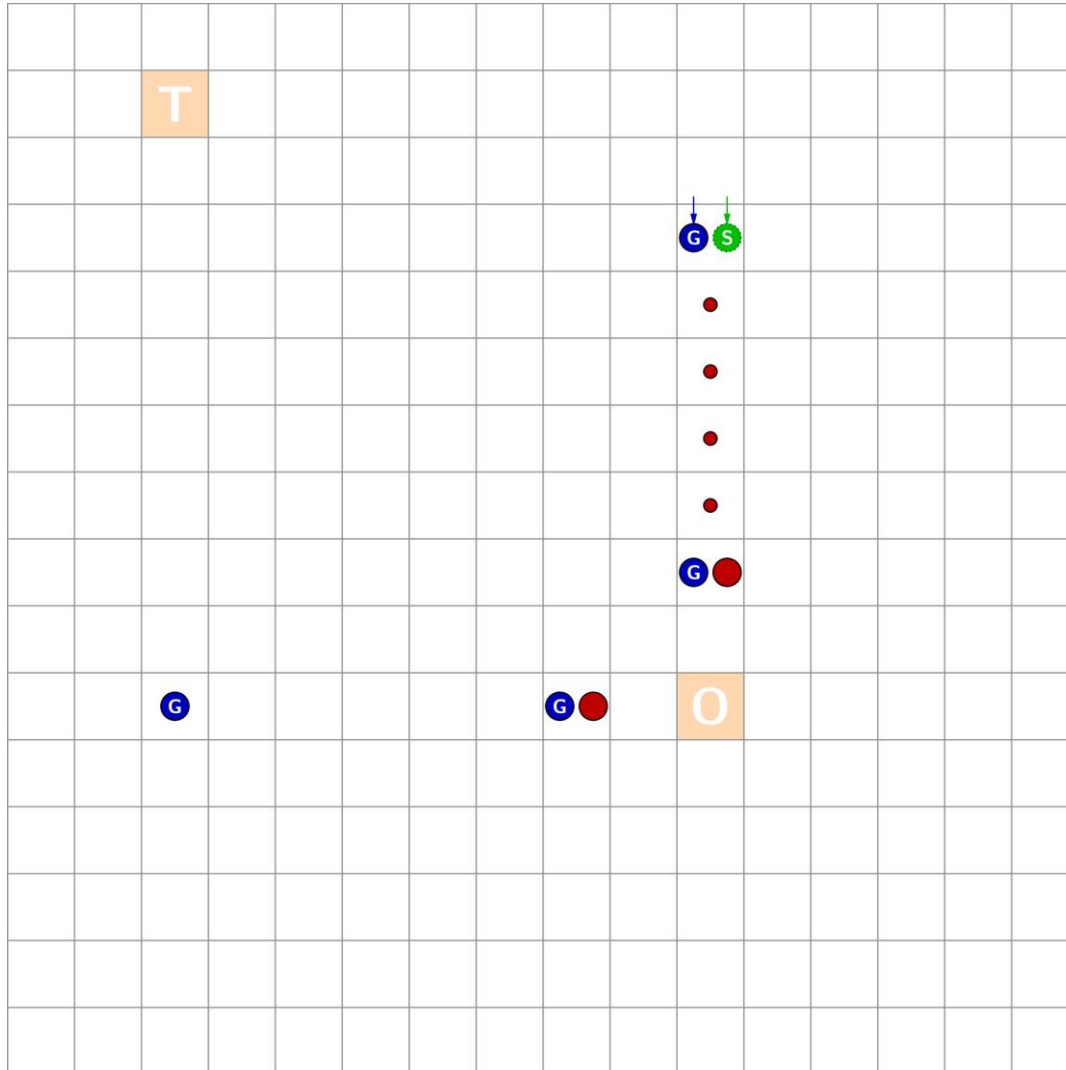
Robust Search



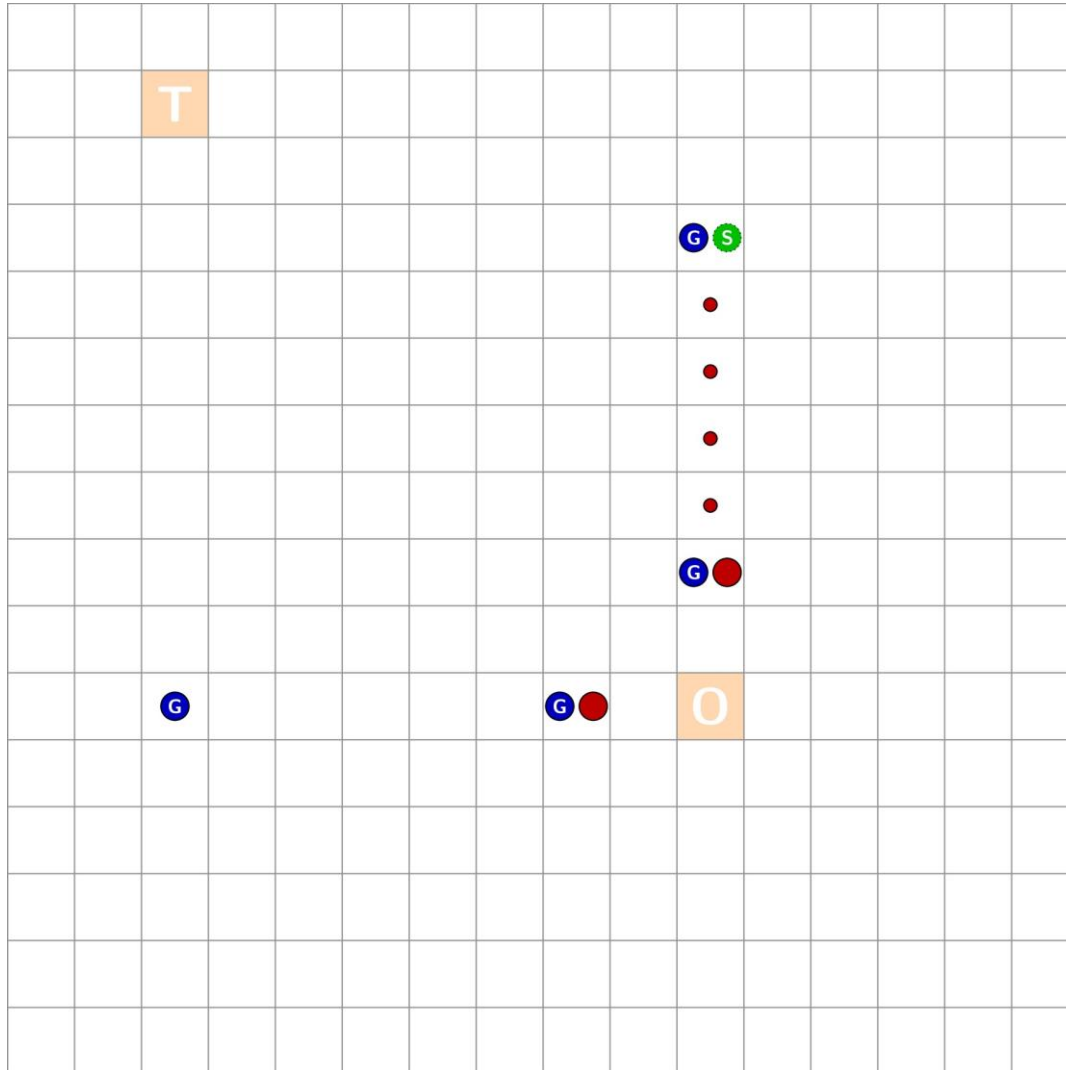
Robust Search



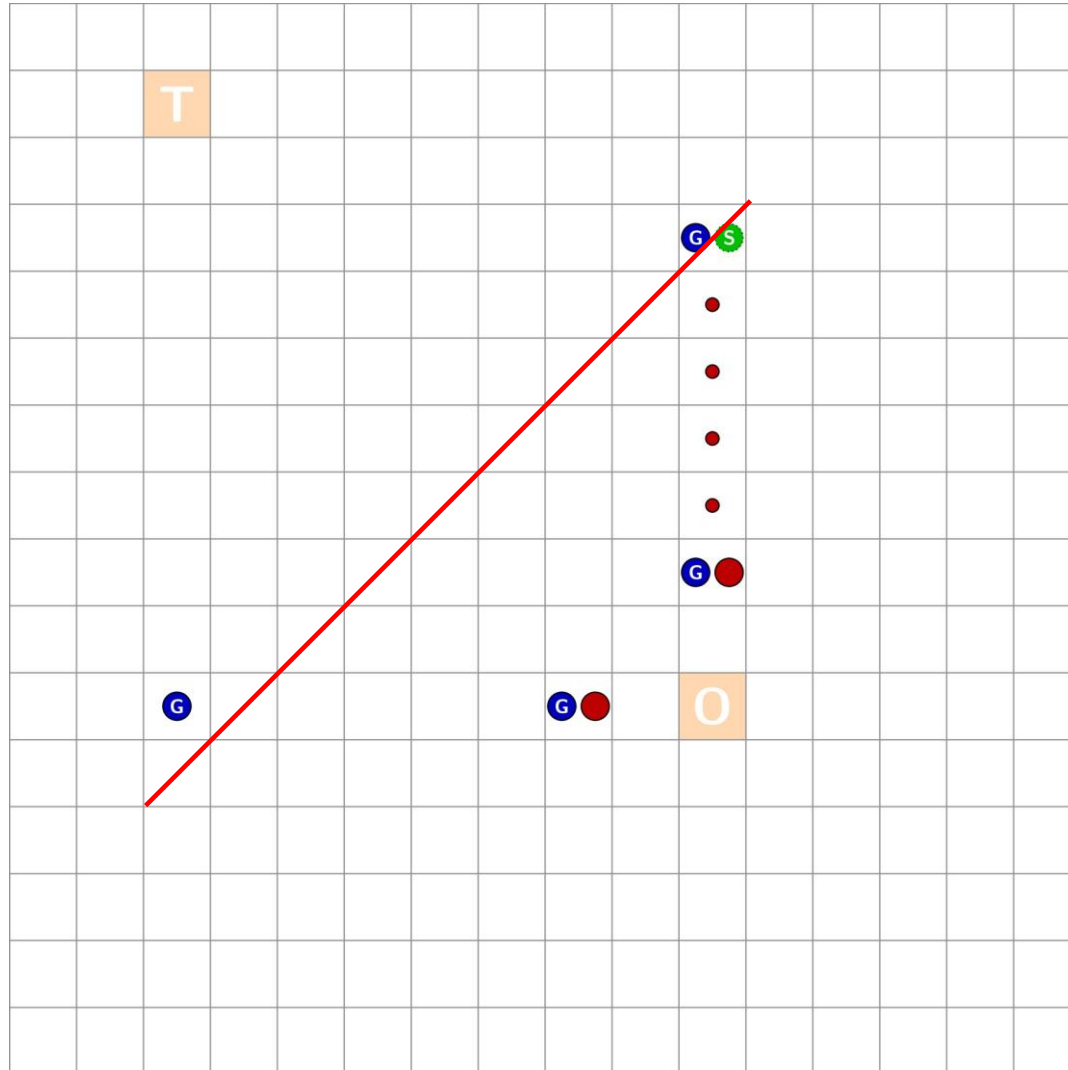
Robust Search



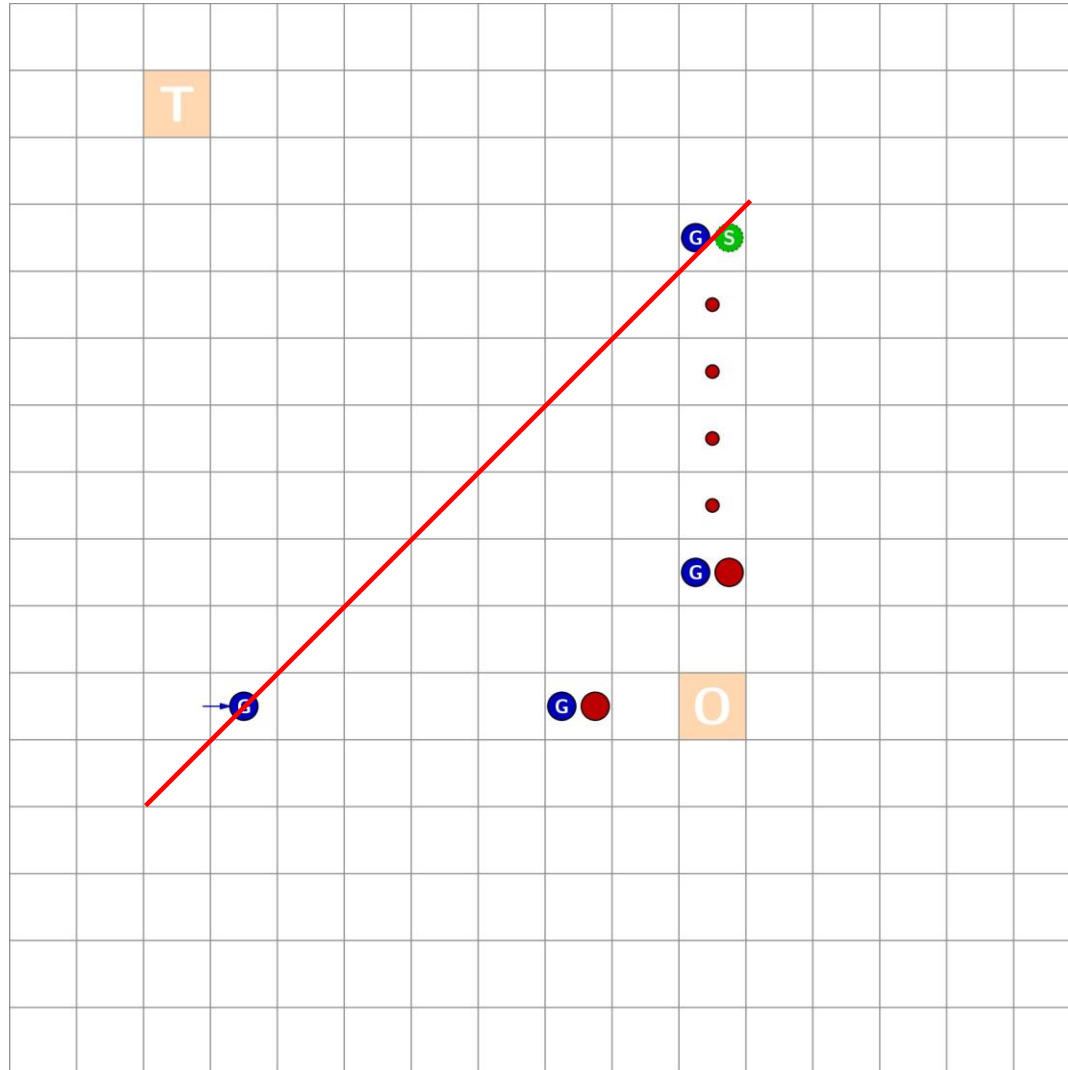
Robust Search



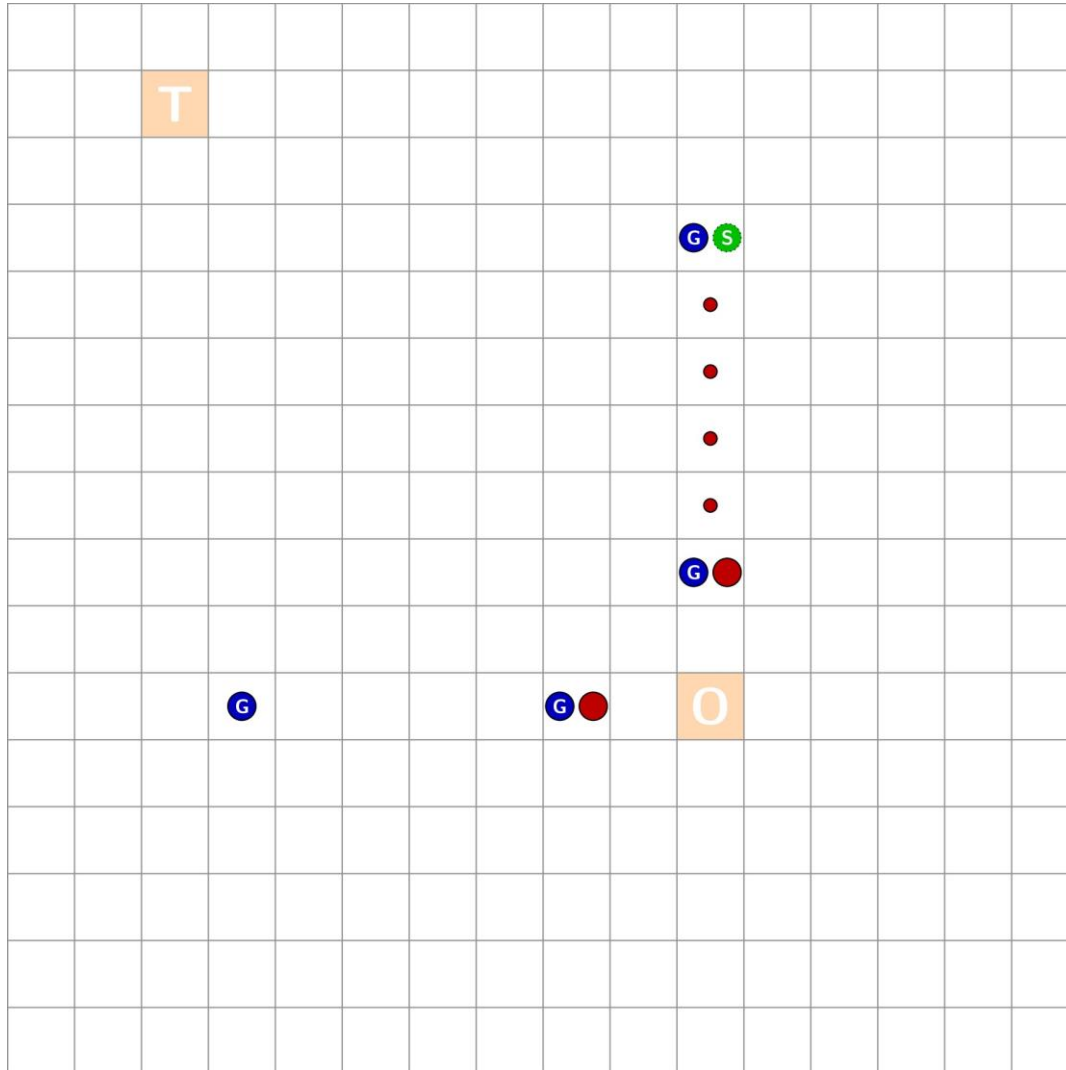
Robust Search



Robust Search



Robust Search



Robust Search

- No (additional) agents are lost during the execution.
- No loss in speed in the case of no failures.

Results

- Every failure can force a repetition of one sweep.
- Find the treasure in time $O(D + \frac{D^2}{n} + Df)$ time, where f is the number of failures.

Questions?

*Thanks to my co-authors Yuval Emek, Tobias Langner,
David Stolz, and Roger Wattenhofer*

