Lux AI

Lux AI Game mechanics

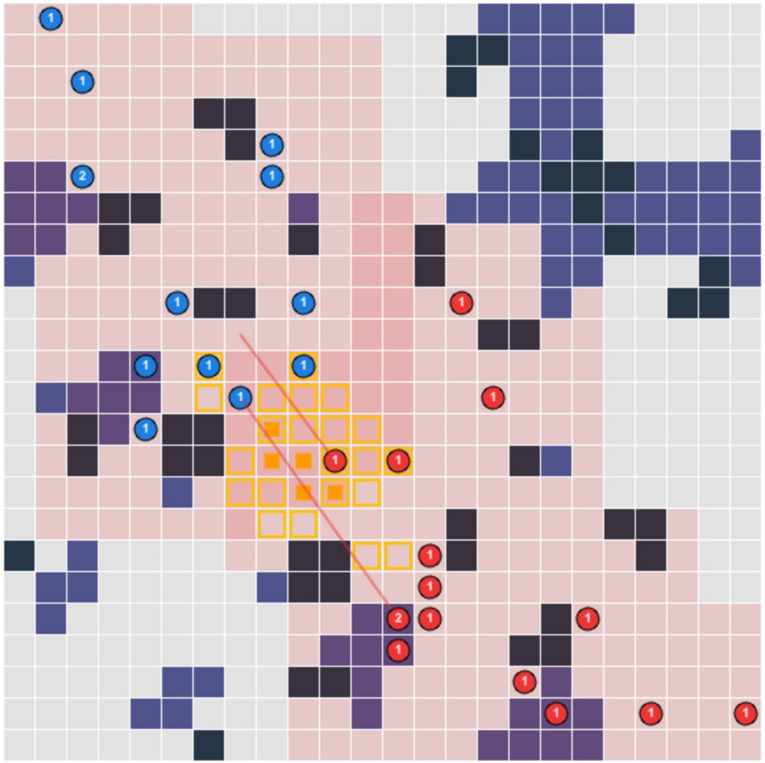
The goal is collect the most points from the hidden fragment nodes.

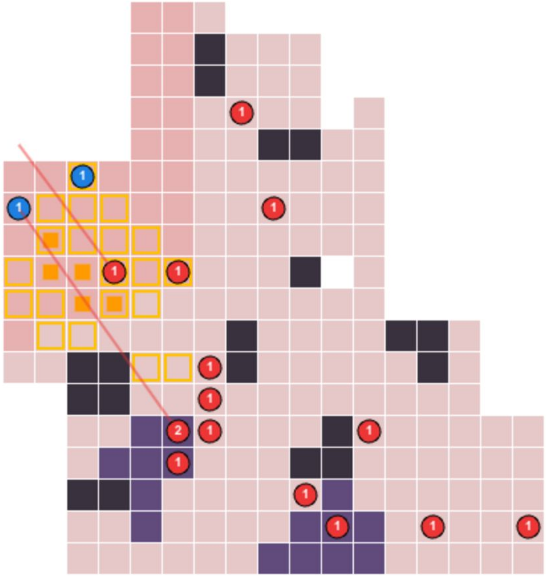
The fragments are always hidden.

The RL agents need to balance the combat and collecting fragment points.

The environment is partially observable with fragment locations randomly changing between episodes.

The parameters are the environments are also changing.





The solution - Multiagent RL

We want to challenge ourselves and research multiagent RL behaviors.

FYI, the past winning solutions were centralized models, issuing commands for all units.

Our approach is multiagent where each unit observes the environment and decide the action for itself.

We go with centralized training and decentralized execution paradigm.

The RL algorithm is Multiagent Proximal Policy Optimization.

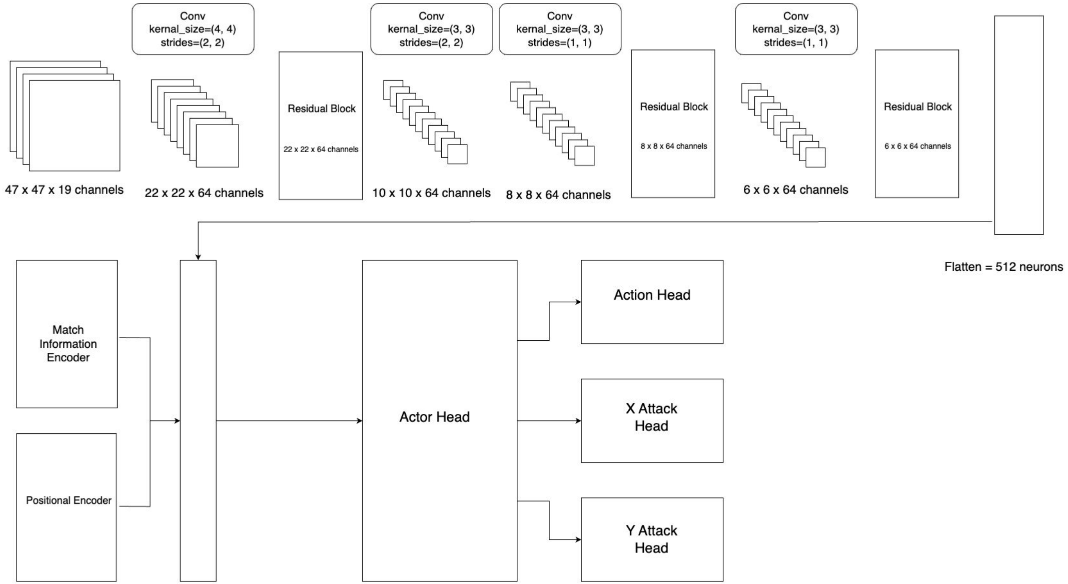
Actor, Egocentric 47 x 47 x 19 channels

1. Combined_asteroid_nebula, (nebula values are scaled according to nebula energy deduction)
2. Energy_map, (observable energy tiles)
3. team_energy_maps / 300,
4. opponent_energy_maps / 300,
5. team_unit_maps / 2, (units count / 2)
6. opponent_unit_maps / 2,
7. sensor_maps,
8. relic_node_maps,
9. Updated_points_map, (rewards cells, 1 for confirmed positive, -1 for confirmed negative, and probs values)
10. Updated_search_map, (unit travel history, useful for finding relic nodes in match 1, 2, 3)
11. Team_sapped_unit_maps, (map for unit who sapped previous turn

8 channels for 2 previous timesteps

1. Combined_asteroid_nebula, (nebula values are scaled according to nebula energy deduction)
2. Energy_map, (observable energy tiles)
3. team_energy_maps / 300,
4. opponent_energy_maps / 300,

Actor Model

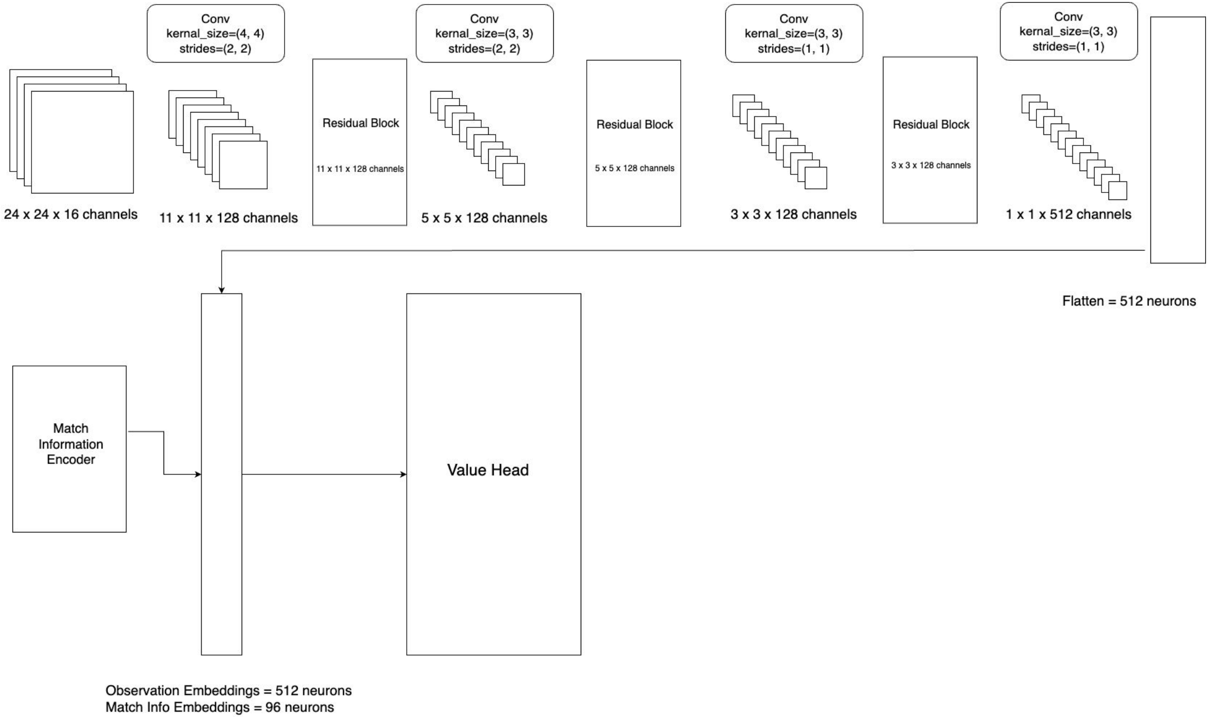


Observation Embeddings = 512 neurons
Match Info Embeddings = 386 neurons
Positional Embeddings = 64 neurons

Critic, Env State 24 x 24 x 16 channels

1. team_energy_maps / 300 (t-1 step)
2. opponent_energy_maps / 300, (t-1 step)
3. team_energy_maps / 300,
4. opponent_energy_maps / 300,
5. team_unit_maps / 2, (units count / 2)
6. opponent_unit_maps / 2,
7. Energy_map, (observable energy tiles)
8. Combined_asteroid_nebula, (nebula values are scaled according to nebula energy deduction)
9. Team_relic_node_maps,
10. opponent_relic_node_maps,
11. Team_points_map, (rewards cells, 1 for confirmed positive, -1 for confirmed negative, and probs values)
12. Opponent_points_map,
13. Team_search_map, (unit travel history, useful for finding relic nodes in match 1, 2, 3)
14. Opponent_search_map
15. Team_sapped_unit_maps, (map for unit who sapped previous turn)
16. Opponent_sapped_unit_maps

Critic Model



Action masking

- Mask invalid movements, (e.g, moving out of bounds, moving into asteroids)
- Collision avoidance i.e., don’t collide with enemy if energy is lower
- Sap only if we see opponents, Center, Up, Down, Left, Right directions
- Sap into the fog if the position is confirmed to be relic fragment