Unrailed! 2

Reinforcement Learning Agent

Donny Nguyen
Hanlei Gao

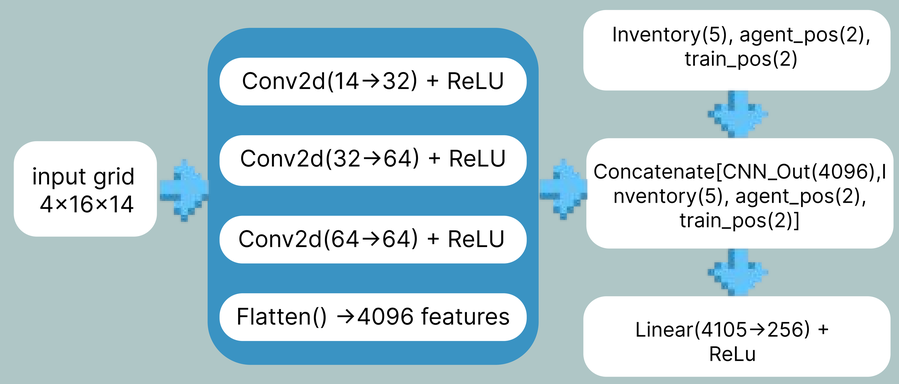
Introduction

Goal:
Train a reinforcement learning (RL) agent to autonomously play Unrailed 2, a game requiring navigation, resource gathering, crafting, and real-time planning.

Why It's Important:

- Reflects real-world challenges in robotics and autonomous systems
- Agents must coordinate many actions under time pressure and uncertainty
- Evaluates whether RL agents can plan many steps ahead instead of reacting short-term

Methodology



Feature Extractor

Curriculum Learning + PPO
We trained the agent using Proximal Policy Optimization (PPO) for stable learning and applied curriculum learning with four stages: movement, resource gathering, crafting, and track placement.

Scheduler
To reduce random exploring, we would guide the agent in the correct direction based on the inventory of the agent.

Reward Shaping
reward = step_penalty + Σ(event_rewards) + progress_scale × BFS_delta

Observation Space
14-Channel Spatial Grid (4×16×14):
Each channel is one entity type (trees, trains, etc)
Preserves spatial structure for CNN features

Action Space
6 Discrete Actions: STAY, UP, DOWN, LEFT, RIGHT, INTERACT

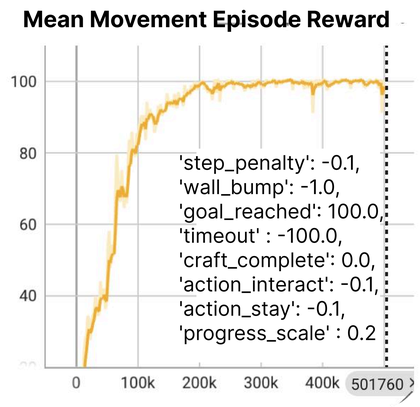
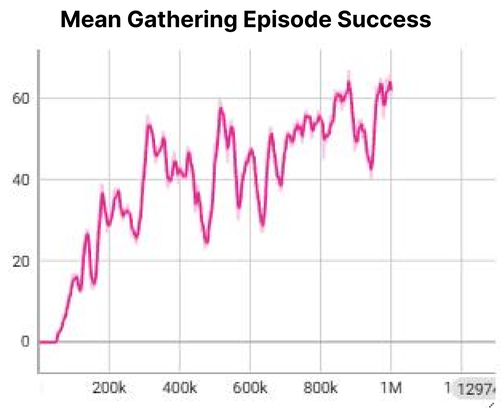
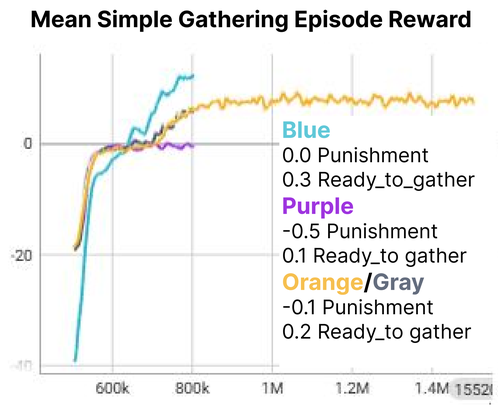
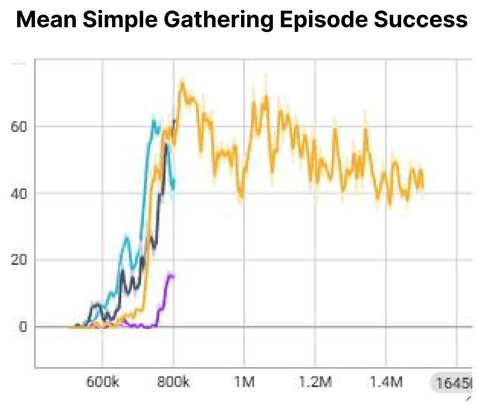
PPO Hyperparameters

"MultiInputPolicy",
learning_rate=01e-4,
ent_coef=0.05,
n_steps=2048,
batch_size=64,
gamma=0.99,
gae_lambda=0.95,
n_epochs=10,

Discussion

- RL agents are very sensitive to incentives
- Environment design is just as important as model design
- Step count tuning was critical to balance exploration and task completion

Future

- Scaling the agent to larger, full-sized maps
- Enabling multi-agent cooperation
- Integrating computer vision and a keyboard control bridge to play directly in the real game

Results