🧠 vs. 🤖

My solution approach was to first understand how each substrate works by creating hardcoded (rule-based) policies, then switch to trained models after gaining enough understanding to shape the rewards. In the end, my hardcoded policies outperformed my best trained models by 146% on average so I used the hardcoded policies for the final evaluation.

After many iterations modifying hyperparameters, reward shapes, and testing different algorithms (PPO, DQN, MARWIL) using the Ray/RLLib framework, my best trained model used PPO with observation space of a flattened 11x11x3 RGB image with values normalized between -1 and 1. The rewards were customized; for allelopathic harvest, the agent was awarded 2 points for collecting any ripe berry, 1 point for changing the color of an unripe berry to red, 0.025 points for turning to face unripe plants that need to be changed to red, and 0.025 points for navigating toward the nearest unripe non-red plant. The action space included all possible actions except for planting green and blue to ensure the agent only plants the preferred color. The environment consisted of 16 focal players with the player_who_likes_red role. The changes to the default PPO config (using Ray 2.6.1) were use_lstm=True, _disable_preprocessor_api=True, and the following model config:

```
"conv_filters": [ [16, [3, 3], 1],
                  [32, [3, 3], 1] ]
```

Figure 1 shows the episode reward mean (using the custom rewards) during training for 3700 episodes. It took 81 hours to complete this training on my local machine.

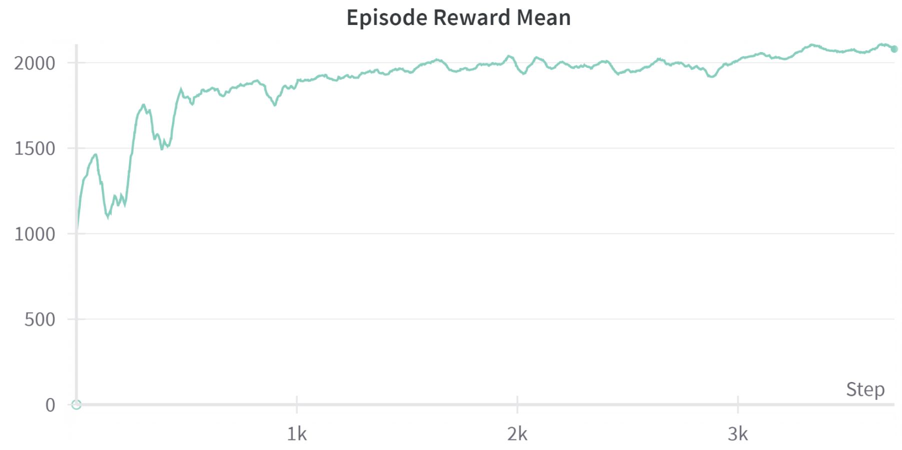

Figure 1. Episode reward mean training PPO on allelopathic harvest substrate

Figure 2 shows that the hardcoded policy outperformed the trained model by 146% on average. I believe the trained models could eventually outperform my hardcoded policies with more time to iterate on hyperparameters and reward shapes. Additionally, allowing more training time and exposing these models to a variety of different scenarios could further enhance their learning and adaptability. Due to the large amount of time it took to train on my local machine, I switched my focus toward improving the hardcoded policies for the final evaluation.

Scenario	PPO Average Score	Hardcoded Policy Average Score
allelopathic_harvest__open_0	1.05	4.05
allelopathic_harvest__open_1	0.59	1.37
allelopathic_harvest__open_2	1.73	2.09

Figure 2. Comparison of trained model (PPO) and hardcoded policy mean focal score (updated) average of 3 episodes on each scenario

Below I will cover the strategy used by the hardcoded policies.

Territory Rooms

The focal agents begin by running through a predefined sequence of actions (rotate and fire claiming beam). This allows the focal agents to detect which neighboring players are focal or background players; if the neighboring player does not perform the predefined action, it can be assumed to be a background player. Next, the focal agent detects the nearest reachable unclaimed or background-claimed wall and navigates to it; this continues until a background player enters or until timestep 70 when the focal agents begin to destroy walls to reach unclaimed or background-claimed walls in other areas.

The focal agents make use of breadth-first search to determine the shortest path to goals while avoiding obstacles like walls and other players. The areas that the background players can zap are considered as additional obstacles. Focal agents also detect when the background players have last zapped to take advantage of the delay before they can zap again.

When background players are reachable, the focal agents attempt to navigate to get the background player within zap range while avoiding obstacles. When the focal agent is injured or is in the delay period just after zapping, the focal agent will navigate to the farthest reachable area from all visible background players.

Focal agents can typically only see 4 out of the 8 other players at the start of the game, so I added an additional method for focal agents to signal to each other that they are focal as opposed to background players. When two players enter each other's view for the first time, they must fire claiming beam to signal that they are focal. Additionally, if any player is seen firing claiming beam more than 8 times, that player is known to be a background player because claiming beam is only used by focals to signal to newly seen players, and there are only 8 other players.

Clean Up

Focal agents initially navigate to the west wall and align side by side. At timestep 35, a role allocation occurs: the two focal agents closest to the water take on the role of dirt cleaners, while the others become apple harvesters. The dirt cleaners are programmed with specific conditions for task switching. Under one condition, they switch from cleaning dirt to harvesting apples; under another, they revert from apple harvesting back to dirt cleaning. In contrast, the focal agents designated as apple harvesters at the wall are assigned a single task; they exclusively harvest apples without any role change.

To switch the goal from cleaning dirt to harvesting apples, the dirt cleaners must either see no dirt for the past 20 timesteps while in water or see at least X other players in the water in the past 20 timesteps. Dirt cleaner 1 has X set to 2, and dirt cleaner 2 has X set to 3. Any focals who didn’t reach the wall by timestep 35 become additional dirt cleaners with X set to 2. To switch the goal from harvesting apples back to cleaning dirt, the focal must see no apples for the past 3 timesteps while in grass.

Focal agents avoid stepping into the zap ranges of background players.

Allelopathic Harvest

The highest priority for focal agents is to navigate to the nearest ripe berry. If there are no ripe berries, the focal agents will change the color of the nearest non-red berries to red. If a green player happens to be within zap range while navigating, the focal will zap them. Focal agents avoid the zap ranges of background players during navigation.

I submitted two versions of this policy: one where focal agents prefer red and one where focals prefer green berries.

Prisoners Dilemma in the Matrix

Focal agents begin by navigating to resources if they are visible, or away from corners if no resources are visible. Focal agents prefer red but will navigate to green resources if there are no red visible. Once focals have collected at least 2 red or 2 green resources, they will navigate and interact with the nearest interactable player. If there are no interactable players visible at this point, the focal will make a full rotation to assist with spotting interactable players. If there are still no interactable players visible, the focal will continue to collect resources (or rotate in place if no resources are visible) until an interactable player is found.