In our project, we implemented a generic model-based algorithm to model a scenario in the StarCraft II as a MDP, where 2 marines, which are 2 units controlled by the player, are expected to collect as many mineral shards as possible in the map.

So the first question here is that if it should be modeled as a single-agent system or multi-agent system – the setting seems like a multi-agent system, where the 2 units should cooperate with each other, but this is not the case for this game setting. The key to clarify this is how we define an agent. In this game setting, the agent is actually the player who controls the units rather than the 2 units themselves. This is a process of simulating human player’s operation, instead of treating the units as two independent robots. Some actions provided by the APIs, such as … are also defined for a player, a unit cannot select himself. Besides, you can’t control the 2 units both simultaneously and independently. Either you select both units and let them move together, or you select one unit to control and leave the other one aside. So, based on this comprehension, we modeled the scenario as a single-agent MDP.

Next we directly head to the elaboration of the evaluation and the result.

# Evaluation Result

Xxx

# Deviation from Proposal

1. In the proposal we presented an off-line method which conducts value iteration to update the expected value given all available states and actions, and then update the policy accordingly. Obviously this is not capable for our scenario here. We cannot collect all the states and all possible actions. So we implemented an on-line method, updating the values, rewards, transition model and the policy while sampling. If the current state has been collected before, it will act according to the policy. If it hasn’t been collected yet, a random available action will be executed.

2. Another deviation Jingyi has talked about, which is the single game and random game. At the beginning we wanna make our method qualified for any random initial state. However, it seems that it needs a relatively long process to finally make it, and our hardware couldn’t handle it when the maintained policy, action values and the transition model became very huge. So we just headed to focus on a repeated single game, where in each episode the initial state remains unchanged.

# Improvement

1. We also plan to take the mini map into consideration. You know mini map is a low-resolution approximation of the main screen map, it’s much smaller, so there would also be a much smaller state space. The units will move along a rough direction towards the minerals. This way may slightly alleviate the problem caused by the huge state space.

2. We will forbid some useless actions to reduce the action space. In our project, we just let all available actions to be operated in each state, but some of them have no contribution at all. We will test if abandoning those actions and only using useful actions will make the convergence faster.