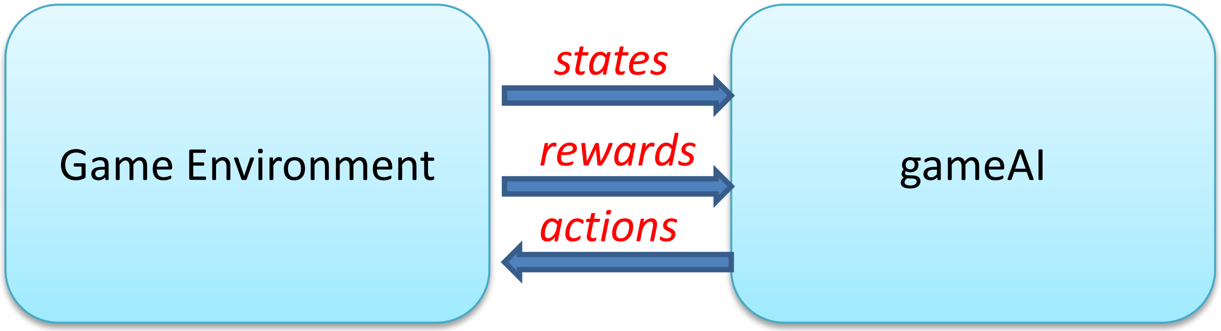
Create gameAI with Reinforcement Learning Algorithm

Why do we use RL algorithm to create gameAI?

The RL approach has strong links to the cognitive processes that occur in the human brain and proved effective in many kinds of games.

The referenced work which use TD learning(one form of RL) to play Hearts(a 4 player card game) can beat one of the best-known programs in the world in the imperfect information version of Hearts.

Our Gravyboat game is also a multi-player imperfect information game which has at most 6 rounds and 6 turns in each round. Each player can get *state* and *reward* from game environment, and decide which card to play in current state(take an *action*).



Create gameAI with Reinforcement Learning Algorithm

SARSA Learning (a type of Model free RL)

SARSA is similar with TD learning, the main difference is that TD learning evaluate the state-value but SARSA learning evaluate the *action-value* at each state for each action.

$$\begin{tikzcd} s_1 \arrow[r, "r_1", "a_1"'] & s_2 \arrow[r, "r_2", "a_2"'] & s_3 \arrow[r, "r_3", "a_3"'] & \cdots\cdots \arrow[r, "r_n", "a_n"'] & \boxed{\text{Terminal}} \end{tikzcd}$$

$$Q(s_i, a_i) \leftarrow (1-\alpha)\times Q(s_i, a_i) + \alpha\left[r_i + \lambda Q(s_{i+1}, a_{i+1})\right]$$

$Q(s_i, a_i)$	is the value of taking action a_i from state s_i
α	is the learning rate
r_i	is the immediate reward for taking action a_i from state s_i
λ	is the discount factor

Create gameAI with Reinforcement Learning Algorithm

The procedure of SARSA learning

```
Initialize Q(s,a) arbitrarily
Repeat (for each episode):
    Initialize s
    Choose a from s using policy derived from Q( e.g., ϵ-greedy)
    Repeat (for each state of episode):
        Take action a, observe r and s'(next state)
        Choose a' from s' using policy derived from Q(e.g., ϵ-greedy)
        update Q(s,a)
        s ← s'; a ← a';
```

It’s proved that the SARSA learning will converge to optimal policy at sufficient runs.

Create gameAI with Reinforcement Learning Algorithm

The possible difficulties for using SARSA learning in GravyBoat

The state space in GravyBoat is huge, for a 4 player game, without considering the derelicts ships, there are 54!/50! = 7590024 states. It will cost much time and memory to train such huge state space, so, we need to find out some efficient method to create the state approximator.

Considering the influence of derelicts ships

Considering the action of using Emergency Stop! Card

Considering the case when the opponents playing Repulsor cards or Tractor Beam cards.

Since the GravyBoat is a totally imperfect information game, the player do not know which card will be playing by the opponents before revealing. So, we will use Independent Learning at first(do not consider which action will be taken by the opponents).