Vishal Samal(2017209) Divyam Anshumaan(2017147) M Badri(2017164)

# Cooperation and Competition in Multi-Agent Reinforcement Learning (via Pac-Man)

## Motivation

- Multi-Agent systems can be used to effectively address problems in domains such as robotics, distributed control, telecommunications and economics.
- The complexity of these tasks makes it difficult to solve with preprogrammed agent behaviours. The agents must instead discover a solution via learning.
- Many tasks can be dealt with in a cooperative manner. It is important that we are able to train agents to act cooperatively by sharing instantaneous information, episodic experience and learned knowledge.
- We attempt to simulate state-of-the-art algorithms in MARL with close attention to cooperative learning algorithms and Self Play algorithms for training.

## Problem Formulation

- We will be using the Pac-Man API from CS 188 of University of California, Berkely.
- We put two opposing teams in a maze-like grid (as typically seen in Pac-Man games).
- For the simplest version of this problem:
    - The team consists of two Pac-Mans.
    - The grid is symmetrically divided into areas for each team.
    - The objective of each team is to maximize its score before terminal conditions are met and the game ends.
        - This can be done by eating the coins present in opponent areas.
        - This can be done by eating opponent Pac-Mans in your territory.
        - There is also a negative living reward to motivate the agents.
    - Terminal conditions:
        - The clock time runs out
        - There is only one Pac-Man team that has not been eaten
    - Training is done in a Self Play manner, where a team plays against itself to iteratively improve its policy.
- We can arbitrarily increase the complexity of the game to enforce greater cooperation between the agents and see interesting behaviors:
    - Information sharing can only be done when Pac-Mans are in adjacent spaces
    - We limit their vision upto a certain number of spaces.
        - A team can share what each member is currently seeing.
    - We increase the number of teams and agents per team.
        - We can add a benign extra team, that agents can learn to cooperate with or take advantage of.

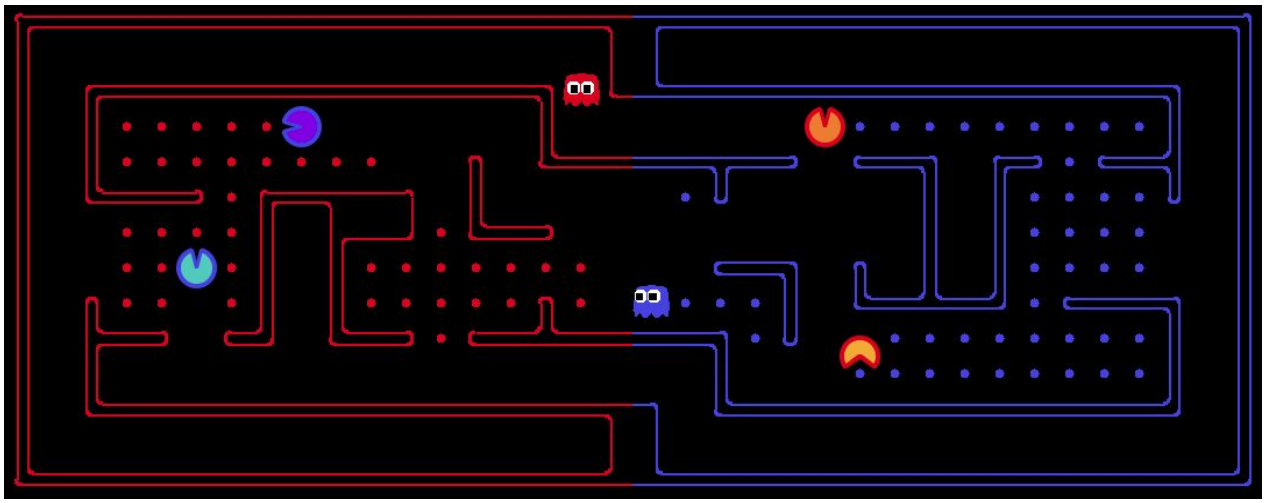

Fig. Screenshot from Contest-2 of CS 188 Fall 2018 Link

## Proposed Learning Algorithm

- State-Of-The-Art
  - PPO (Link)
  - Multi-Agent MDPs - Multiple Actors, Centralized Critics
  - LOLA (Link)

## Existing Work

- **Iterative improvement using adversarial agents via Self-Play (Link):** AlphaGo Zero has learned by playing against itself. According to its creators, it surpassed the performance of AlphaGo which was trained by playing against professionals.

- **Emergent Behaviors in Multi-Agent Interaction (Link):** Here they have explored the Cooperative and Adversarial training of agents in a setup of Hiding and Seek game where the two teams with different objectives(one team has to hide and the other has to seek) discover newer and unexpected strategies.

- **Cooperative Agents (Link)**: In the GitHub link for Books and Papers we have a Review paper with multiple works on Cooperative Multi-Agent Learning.

Vishal Samal(2017209) Divyam Anshumaan(2017147) M Badri(2017164)

## Baseline Algorithms

- SARSA, Q-Learning (Classical/Deep Learning)
- MC, TD(n)
- Policy Gradient
- Actor-Critic Methods

## References:


- Cooperative Agents: (Link 1 , Link 2 , Link 3)
- Competition, Sharing and Self Play: (Link)