

# COSE361 2024S Assignment4

## 1. Introduction

과제4의 과제는 각 진영의 agent가 서로 경쟁하는 adversarial search와 관련된 과제이다. 지금까지의 과제에서는 자신의 최선의 행동만 고려해도 되었다면, 이번 과제에서는 상대방의 행동까지 고려해야 한다는 특징을 가진다.

과제4의 각 offensive, defensive agent는 기본적으로 reflex agent이다. 각 agent 행동에 따라 가중치를 주는 evaluation function을 통해 각 agent는 점수가 높은 행동을 최선의 행동으로 간주하도록 코드를 작성하였다.

## 2. Method

다음은 yourbaseline 코드에 대한 설명이다. 기본적인 틀은 baseline.py의 틀을 유지하며 getWeights, getFeatures, evaluate 등의 함수는 기존 baseline.py의 함수들을 사용하였다. OffensiveReflexAgent와 DefensiveReflexAgent 클래스의 수정을 통해 성능을 개선하고자하였다.

### 2.1. 2.1 baseline1

baseline1은 과제2에서 제출하였던 reflex agent와 유사하게 enemy와 food까지의 거리를 기반으로 코드를 작성하였다.

기존 baseline코드와 유사하게, ReflexAgent라는 클래스를 정의하여, offensive 및 Defensive agent의 공통적 행동 특성을 정의하였다. 이후 Offensive와 reflexive agent에 따라 evaluation function을 정의하여 코드를 작성하였다.

- Offensive agent에서는 ghost와의 거리에 양수값을, food와의 거리에 음수값의 가중치를 두어 agent가 enemy와는 멀어지도록, food와는 가까워지도록 하였다. 이때 거리별로 함수값을 다르게 두어, 가까운 food와 enemy에 대한 반응도를 높였다.
- Defensive agent 역시 enemy와의 거리에 음수값을 두어 agent가 enemy와 가까워지도록, 즉 침입자를 잡을 수 있도록 하였다. 또한 enemy와의 거리에 따라 함수값을 다르게 두어, 가까운 enemy에 대한 반응도를 높였다.

### 2.2. baseline2

baseline2는 baseline1에서 defensive agent의 동작을 개선하였다. food와의 거리에 대한 evaluation function을 추가하여, defensive agent가 food와 가까운 거리, 즉 팀의 경계선에서 효율적으로 방어할 수 있도록 개선하였다.

### 2.3. baseline3(your best)

baseline3는 baseline2에서 offensive agent의 evaluation function 요소를 추가하였다. enemy와 food뿐만 아니라 부가적으로 home 및 capsule에 대한 요소를 추가하여 개선하였다. 그리고 function값을 가까운 거리에 대해 더욱 민감하게 반응할 수 있도록 조정하였다.

- powercapsule에 대한 가중치를 추가하였다. 하지만 capsule에 대한 가중치가 food를 넘지 않도록 설정하여 pacman이 capsule보다 food에 집중할 수 있도록 하였다.
- 기존 baseline 코드에서는 가지고 있는 food 개수가 많더라도 home으로 돌아가지 않았다. baseline3는 home까지의 거리에 대한 가중치를 두어 food를 먹었다면 집으로 돌아가는 방향으로 유도되도록 변경하였다.

output

	your_best(red)
	<Average Winning Rate>
your_base1	1.0
your_base2	0.9
your_base3	0.0
baseline	1.0
Num_Win	3.0
Avg_Winning_Rate	0.725
	<Average Scores>
your_base1	2.4
your_base2	7.5
your_base3	0.0
baesline	3.3
Avg_Score	3.3

### 3. Results

다음 결과를 확인해보면, 기존의 baseline보다 agent의 성능이 개선된 것을 확인할 수 있다.

### 4. Conclusion & Free discussion

#### 4.1. Conclusion

baseline1과 baseline2의 차이를 보면 알 수 있듯이, offensive agent보다 defensive agent의 역할이 중요함을 알 수 있다. 그리고 각 evaluation function의 가중치 조절을 통한 각 행동의 우선순위 역시 점수 획득에 큰 영향을 줄 수 있음을 확인할 수 있었다.

#### 4.2. free discussion

- Q. yourbest 코드에서 알고리즘의 성능을 더 개선할 수 있는 방법은 무엇이 있을까?
- A. 성능이 비슷한 알고리즘일수록 경계에서 대처하는 경우가 존재한다. 그리고 점수획득이 아니라 승패의 관점에서 보면 1점을 먹더라도 상대방이 아예 점수를 획득하지 못하도록 막는 것이 더욱 효율적이다. 따라서 2 agent 모두가 1점을 획득한 이후에는 defensive agent로 작동하는 알고리즘을 작성하고 싶었으나, 이에 대해서는 더욱 고민이 필요할 것 같다.