**CSE 537 - Artificial Intelligence**

**Report: Project 2**

**(Multi-Agent Pac-Man)**

**Khan Mostafa**

109365509

**Abhijit Betigeri**

109229784

{khan.mostafa, abhijit.betigeri}@stonybrook.edu

Department of Computer Science

Stony Brook University

## Designing Agents namely Multi-Agent for classic Version of Pac-Man including Ghosts

## Q1. Reflex Agent – Improvements in game considering food & ghost locations

**Methodology Used:** <add – here>

**Execution Details**

The basic set1 with testClassic Layout could be cleared successfully with our defined evaluation function

Set 1:

python pacman.py -p ReflexAgent –l testClassic

*Pacman Game status:* Win

*Total Score:* 564

*Total Cost for the Path Found*: ?? (not sure how to put it – don’t see stats for it)

*Search Nodes Expanded*: ?? (not sure how to put it – don’t see stats for it)

Set 2: We use the mediumClassic layout with one ghost

python pacman.py -–frameTime 0.1 -p ReflexAgent –k 1

*Pacman Game status:* Win

*Total Score:* 1118

*Total Cost for the Path Found*: ?? (not sure how to put it – don’t see stats for it)

*Search Nodes Expanded*: ?? (not sure how to put it – don’t see stats for it)

Set 3: Usage of mediumClassic layout with two ghost

python pacman.py -–frameTime 0.1 -p ReflexAgent –k 2

1st Run:

*Pacman Game status:* Win

*Total Score:* 1182

*Total Cost for the Path Found*: ?? (not sure how to put it – don’t see stats for it)

*Search Nodes Expanded*: ?? (not sure how to put it – don’t see stats for it)

2nd Run:

*Pacman Game status:* Win

*Total Score:* 1303

*Total Cost for the Path Found*: ?? (not sure how to put it – don’t see stats for it)

*Search Nodes Expanded*: ?? (not sure how to put it – don’t see stats for it)

3rd Run:

*Pacman Game status:* Win

*Total Score:* 1200

*Total Cost for the Path Found*: ?? (not sure how to put it – don’t see stats for it)

*Search Nodes Expanded*: ?? (not sure how to put it – don’t see stats for it)

**Note: Repeated Run with 2 ghosts – Pac-Man wins signifying evaluation functions to quite good.**

Set 4: Usage of openClassic layout repeatedly ie ten times

python pacman.py -p ReflexAgent –l openClassic –n 10 -q

Here are scores of each run, average score and win rate.

Average Score: 1257.5

Scores: 1260.0, 1257.0, 1257.0, 1258.0, 1244.0, 1260.0, 1264.0, 1257.0, 1259.0, 1259.0

Win Rate: 10/10 (1.00)

Record: Win, Win, Win, Win, Win, Win, Win, Win, Win, Win

## Q2. Adversarial search Agent in the provided Minimax Agent Class

**Methodology Used:** <<add – here >>>

Execution Set 1:

python pacman.py –p MinimaxAgent –l minimaxClassic – a depth=4

*Pacman Game status:* Win

*Total Score:* 516

Note: Minimax agent won 655/1000 games in one of our runs.

Q3. Alpha-Beta pruning

Methodology Used: <<Add-here>>

Analysis

<<<<ADD- HERE>>>>>>>

# References

|  |  |  |
| --- | --- | --- |
|  |  |  |