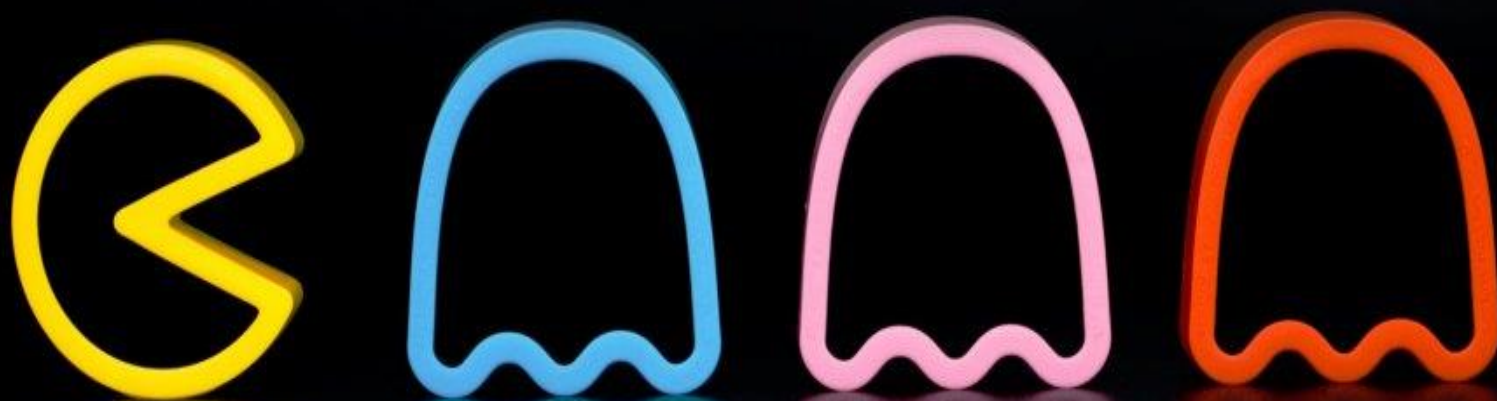


Pacman Project Milestone II

Multi-Agent Pacman



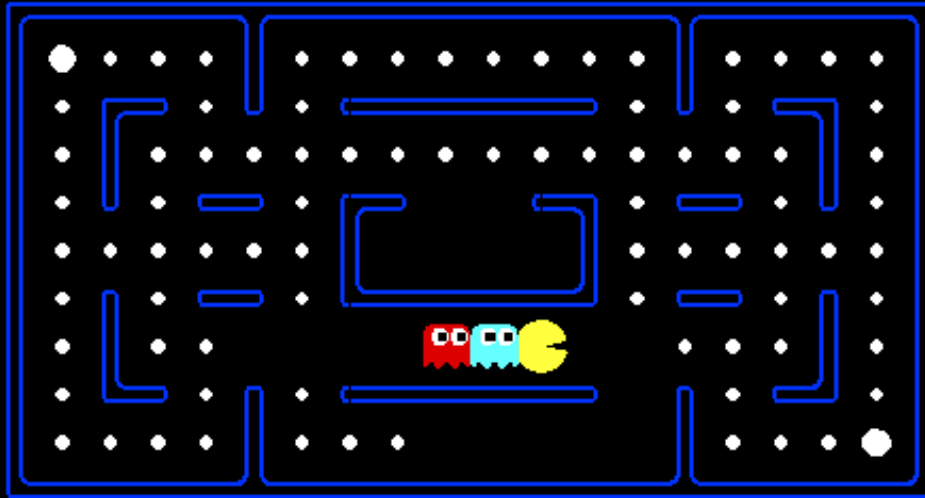
【人工智慧概論】

授課教師／孫春在

助教／呂冠霆 吳晨晨 柯維然 黃慧瑜 黃奕齊 劉長諺 林怡秀 鍾幸芸

日期／2017.05.09

Multi-Agent Pacman



Pacman, now with ghosts.
Minimax, Evaluation

Objectives

P3-1 Reflex Agent (40%)

P3-2 Minimax (25%)

P3-3 Alpha-Beta Pruning (25%)

P3-4 Better Evaluation (**Bonus**, 30%)

-
1. Adversarial Search - Minimax
 2. Alpha-Beta Pruning
 3. Evaluation Function

Adversarial Search

Minimax (P3-2)

Pacman

(max)

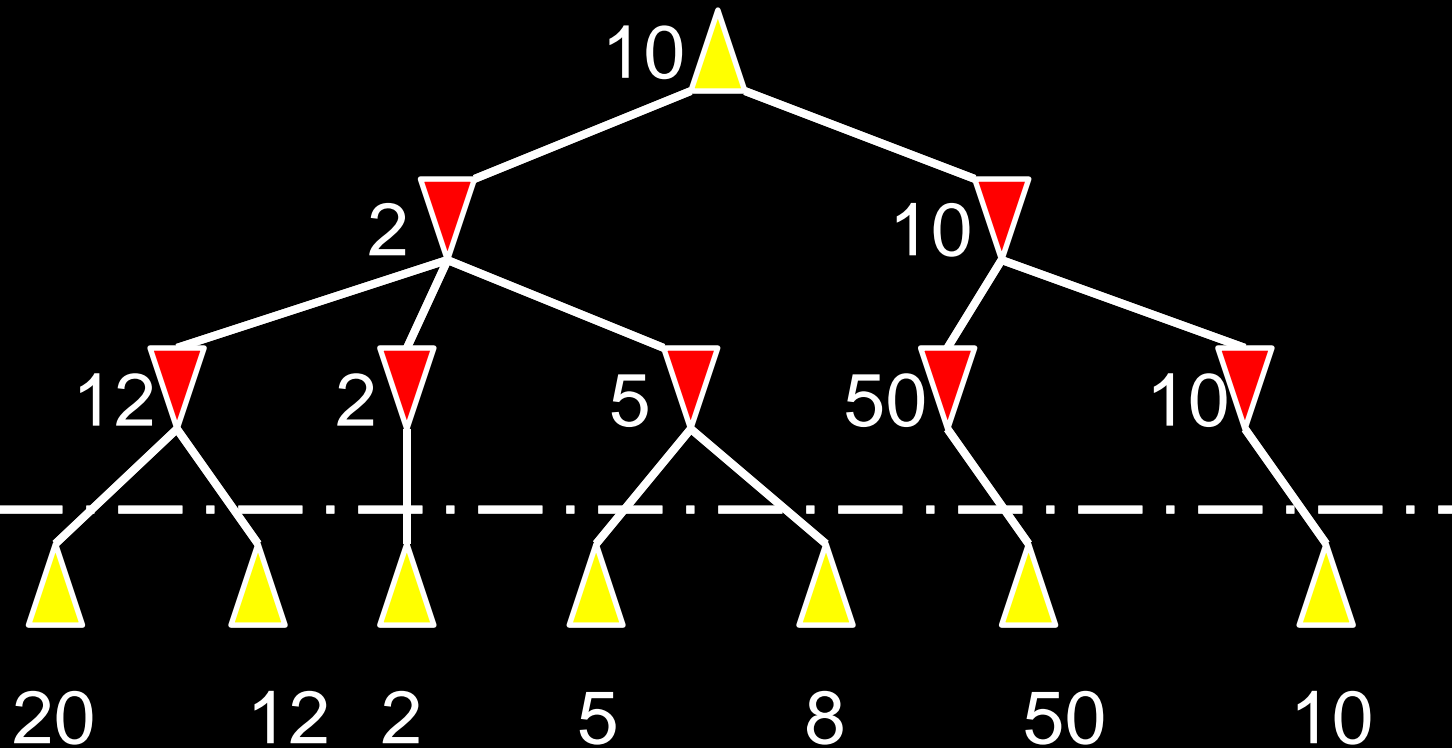
Ghost 1

(min)

Ghost 2

(min)

Pacman



Minimax Implementation

def value(state):

if the state is a terminal state: return the state's utility

if the next agent is MAX: return max-value(state)

if the next agent is MIN: return min-value(state)

def max-value(state):

initialize $v = -\infty$

for each successor of state:

$v = \max(v, \text{value}(\text{successor}))$

return v

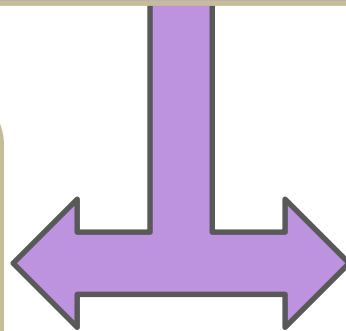
def min-value(state):

initialize $v = +\infty$

for each successor of state:

$v = \min(v, \text{value}(\text{successor}))$

return v

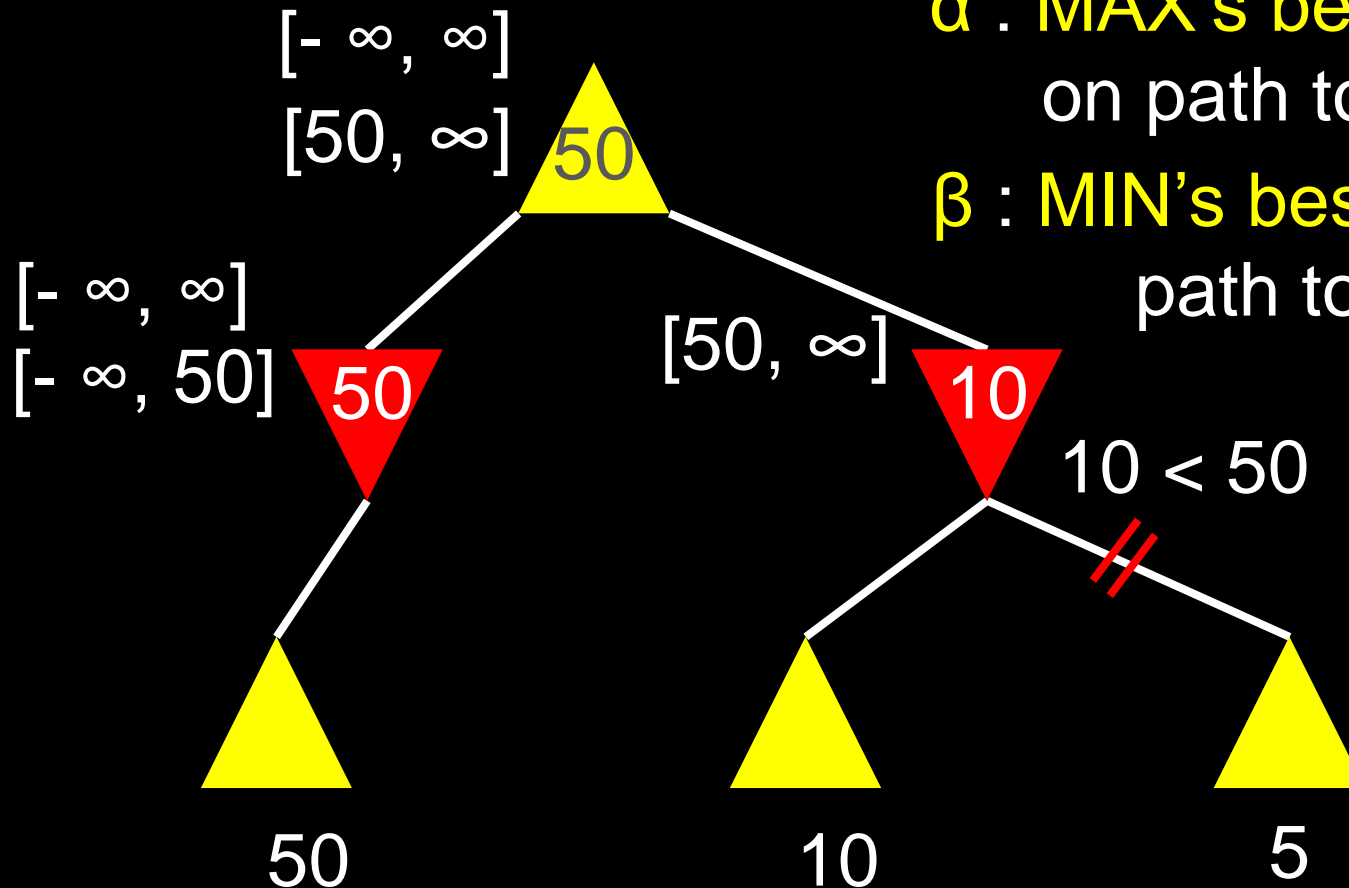


Alpha-Beta Pruning (P3-3)

$[\alpha, \beta]$

α : MAX's best option
on path to root

β : MIN's best option on
path to root



Alpha-Beta Implementation

α : MAX's best option on path to root
 β : MIN's best option on path to root

```
def max-value(state,  $\alpha$ ,  $\beta$ ):  
    initialize  $v = -\infty$   
    for each successor of state:  
         $v = \max(v, \text{value}(\text{successor}, \alpha, \beta))$   
        if  $v \geq \beta$  return  $v$   
         $\alpha = \max(\alpha, v)$   
    return  $v$ 
```

```
def min-value(state,  $\alpha$ ,  $\beta$ ):  
    initialize  $v = +\infty$   
    for each successor of state:  
         $v = \min(v, \text{value}(\text{successor}, \alpha, \beta))$   
        if  $v \leq \alpha$  return  $v$   
         $\beta = \min(\beta, v)$   
    return  $v$ 
```


Alpha-Beta Pruning

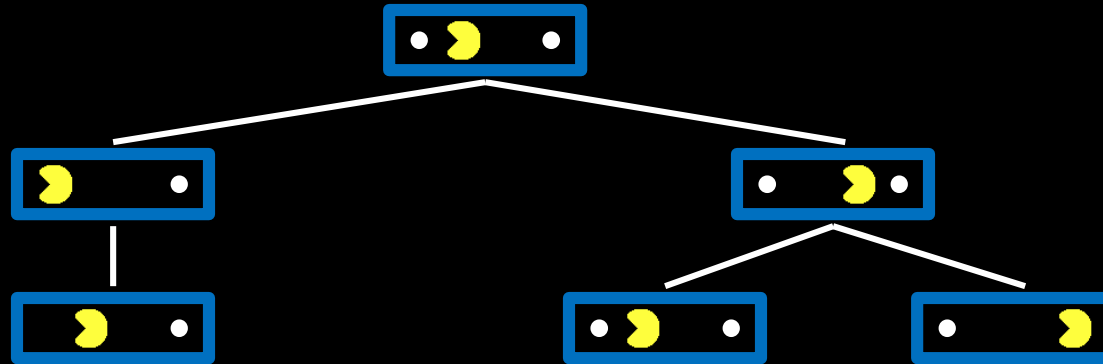
- Step by Step: Alpha Beta Pruning
<https://www.youtube.com/watch?v=xBXHtz4Gbdo>
- Coursera AI course week 5 (5-1~5-3)
<https://www.coursera.org/learn/rengong-zhineng/home/week/5>

Evaluation Function (P3-1,P3-4)

Weighted linear sum of features

$$\begin{aligned} \text{eval}(\text{state}) = & \text{Weight}_1 * \text{Feature}_1 \\ & + \text{Weight}_2 * \text{Feature}_2 \\ & + \text{Weight}_3 * \text{Feature}_3 + \dots \end{aligned}$$

Why Pacman Starves



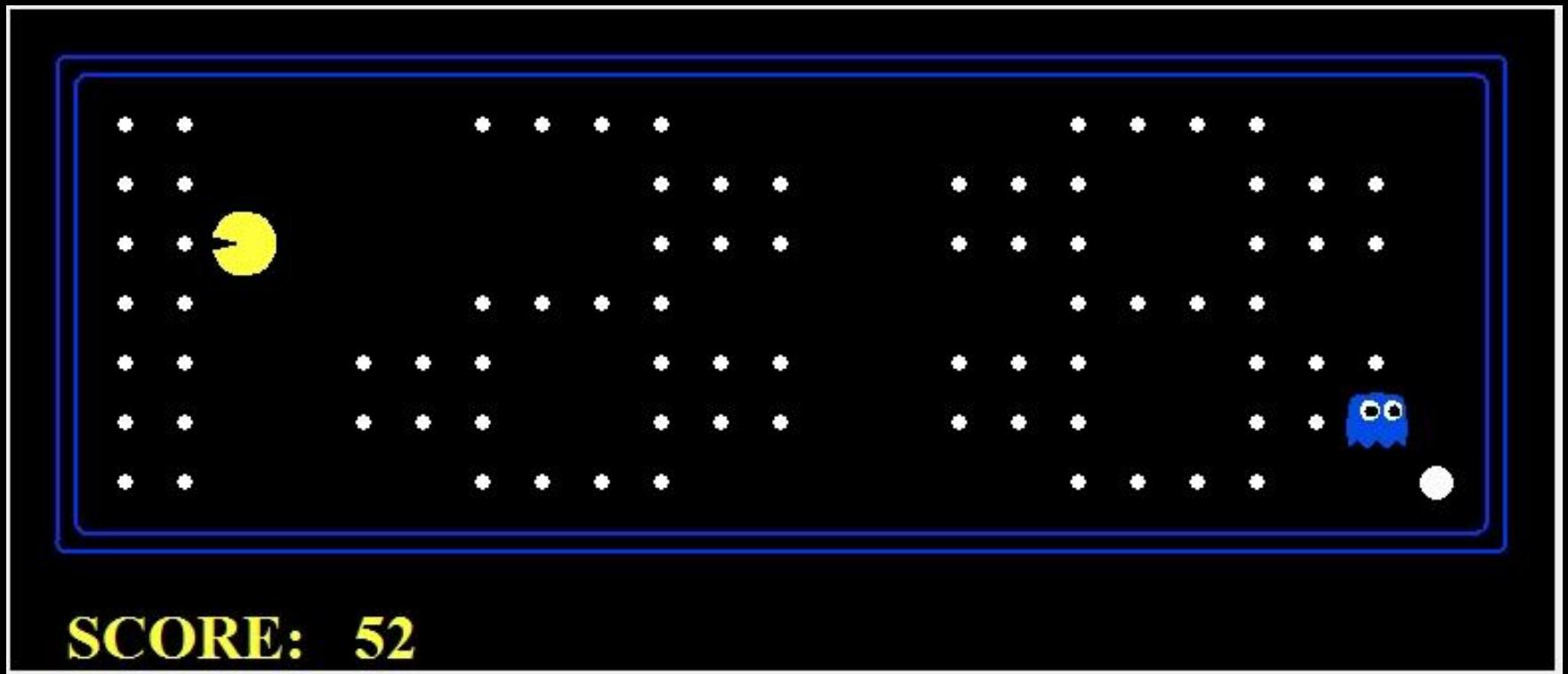
- **A danger of replanning agents!**

- He knows his score will go up by eating the dot now (west, east)
- He knows his score will go up just as much by eating the dot later (east, west)
- There are no point-scoring opportunities after eating the dot (within the horizon, two here)
- Therefore, waiting seems just as good as eating: he may go east, then back west in the next round of replanning!

Objectives (1/4)

Reflex Agent

- `pacman.py -p ReflexAgent -l openClassic`



Objectives (1/4)

Reflex Agent

- Simple evaluation

- $\text{eval}(\text{state}, \text{action}) = w_1 f_1 + w_2 f_2 + \dots$

- Grading (40%)

- `autograder.py -q q1 --no-graphics`
 - openClassic, 10 times
 - 5/10: +30
 - 10/10: +10

Objectives (2/4)

Minimax

- `pacman.py -p MinimaxAgent`
`-l minimaxClassic -a depth=4`



Objectives (2/4)

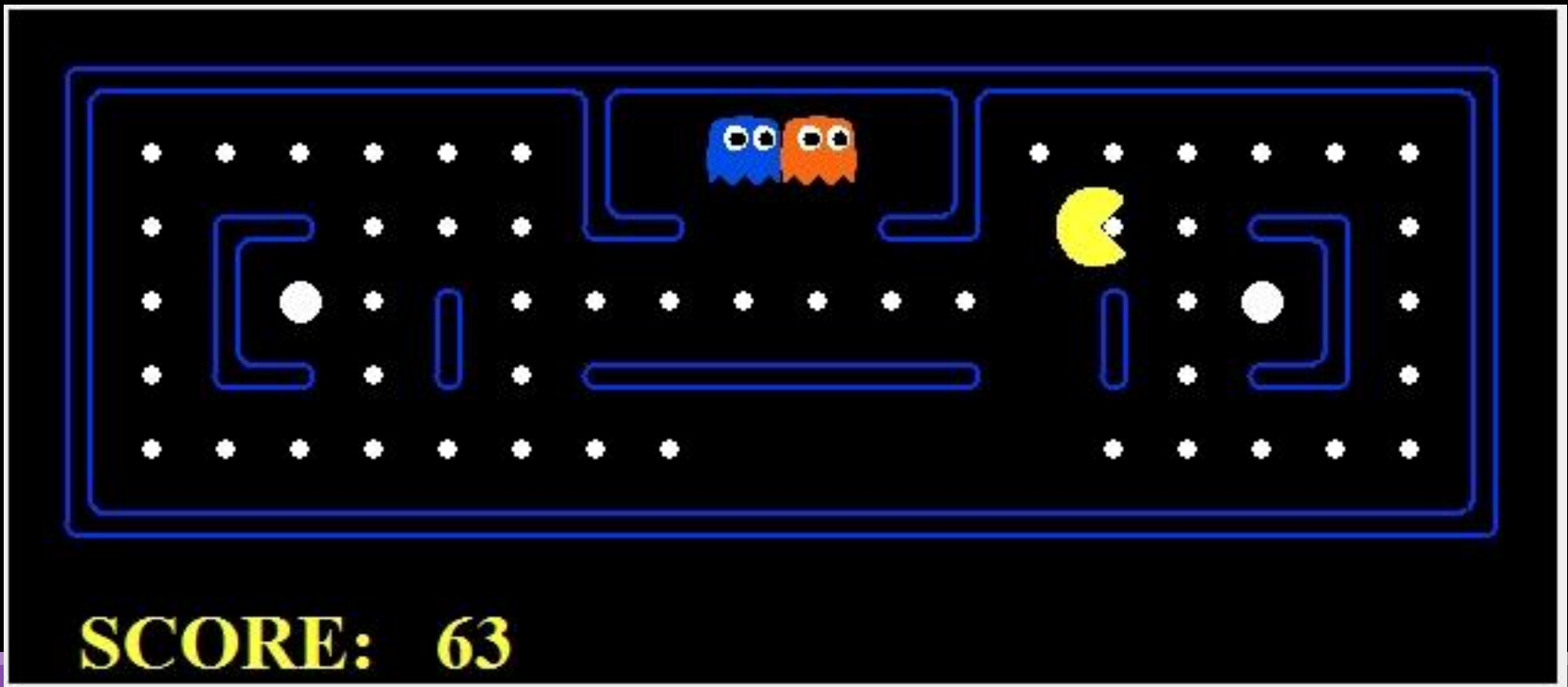
Minimax

- **Must use**
 - `self.depth()`
 - `self.evaluationFunction()`
 - default: `scoreEvaluationFunction()`
- **Grading (25%)**
 - `autograder.py -q q2 --no-graphics`

Objectives (3/4)

Alpha-Beta Pruning

- pacman.py -p AlphaBetaAgent
-l smallClassic -a depth=3



Objectives (3/4)

Alpha-Beta Pruning

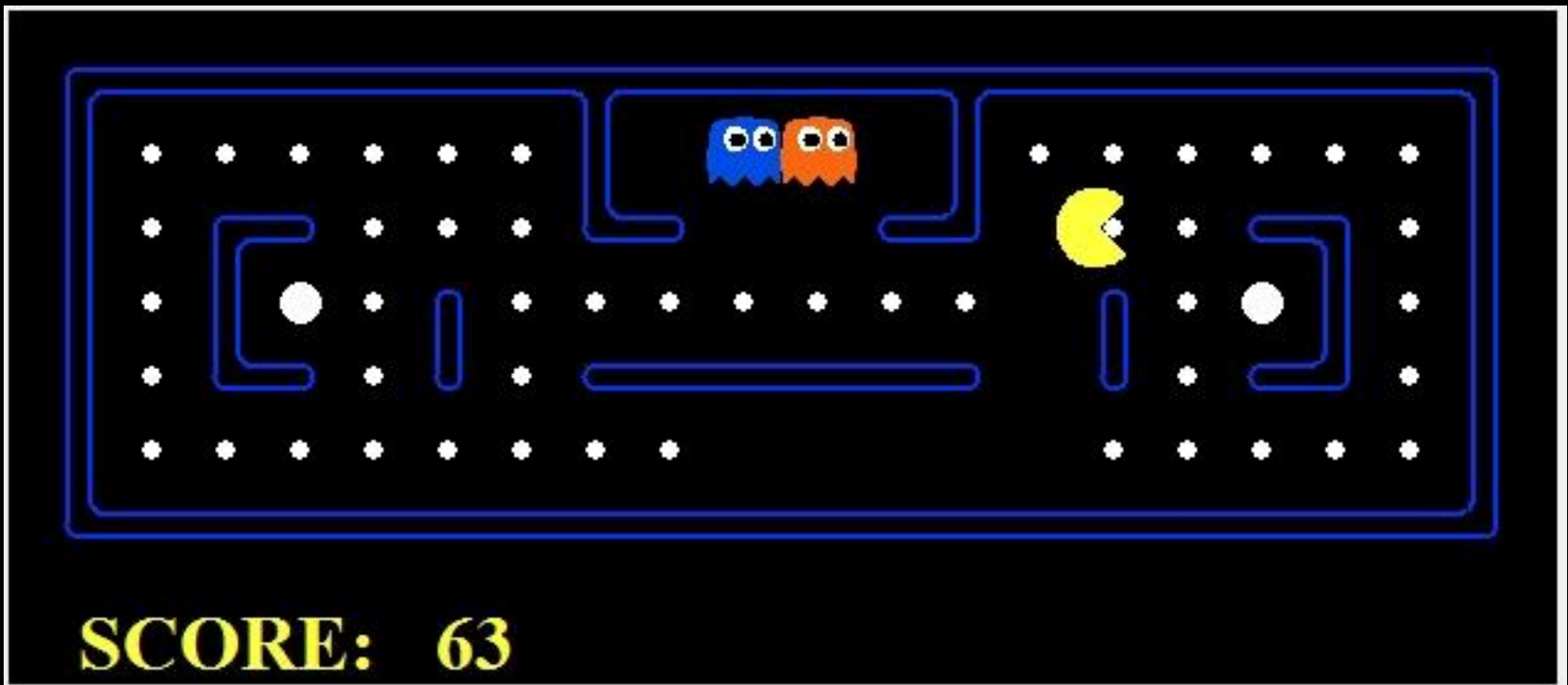
- Grading (25%)

- `autograder.py -q q3 --no-graphics`

Objectives (4/4)

Better Evaluation

- `pacman.py -p AlphaBetaAgent -l smallClassic -a depth=3,evalFn=better`



Objectives (4/4) Bonus

Better Evaluation

- Better evaluation
 - $\text{eval}(\text{state}) = w_1 f_1 + w_2 f_2 + \dots$
 - Describe your features in the comments.
- Grading (30%)
 - `autograder.py -q q5 --no-graphics`
 - smallClassic, Alpha-Beta, depth=3, 10 times
 - 5/10: +10
 - 10/10: +5
 - avg > 500: +10
 - avg > 1000: +5

Options

-z 0.5	0.5x window size
-n #	Play # times
-q	Quiet mode, no graphics
-g DirectionalGhost	Using directional ghost
-k #	Number of ghosts = #
-f	Fixed random seed; line 533, pacman.py
--frameTime 0	No frame time

Submit

- Edit and upload **multiAgents.py** to e3
- Search for “[Project 3] YOUR CODE HERE”
- Deadline: **May/25** 23:59 (about 3 weeks)
- Late policy: 80%
- **No plagiarism**