

Pacman Project Milestone

一心雙身Pacman



【人工智慧概論】

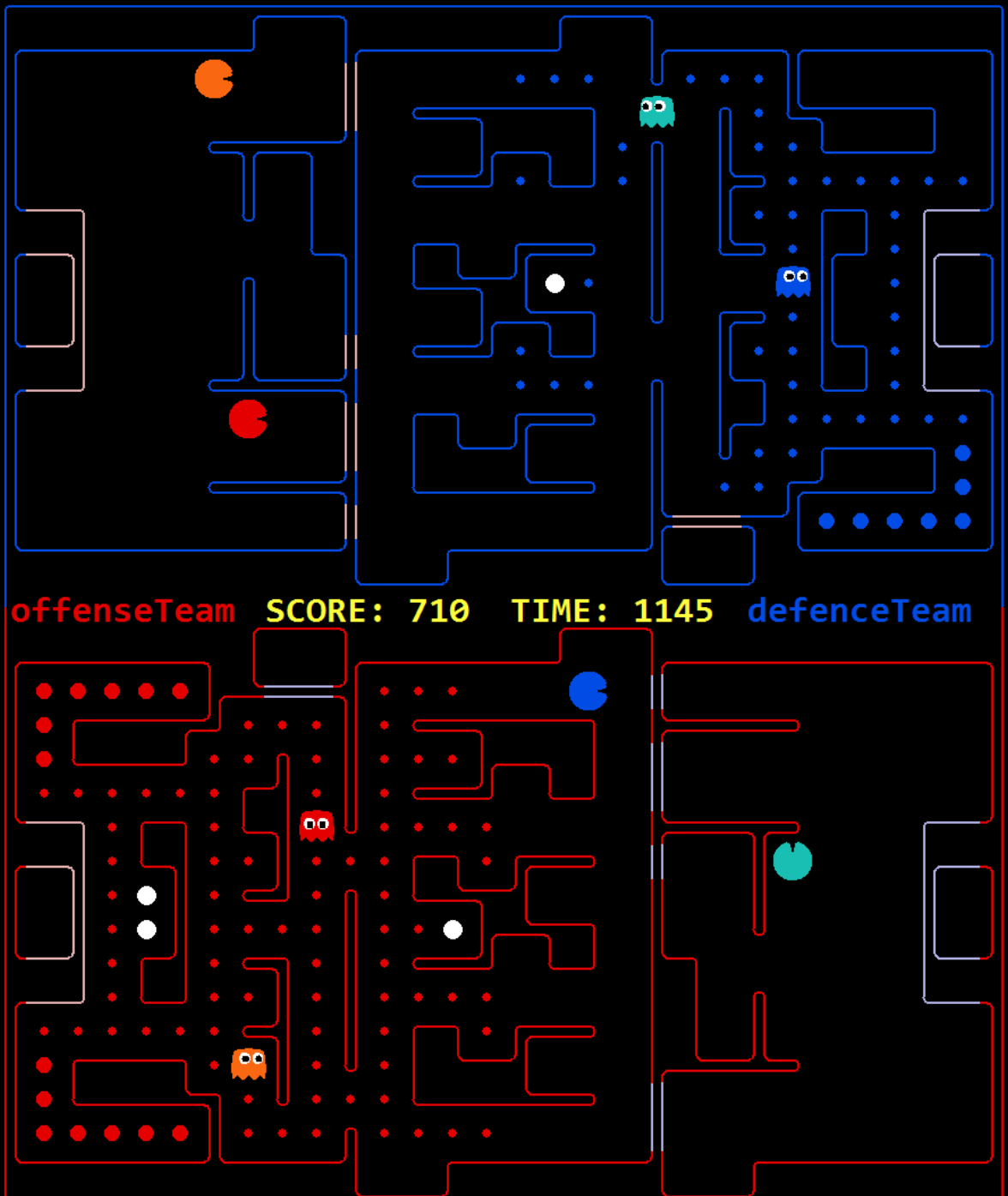
授課教師／孫春在

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／2017.05.11

一心雙身

Pacman

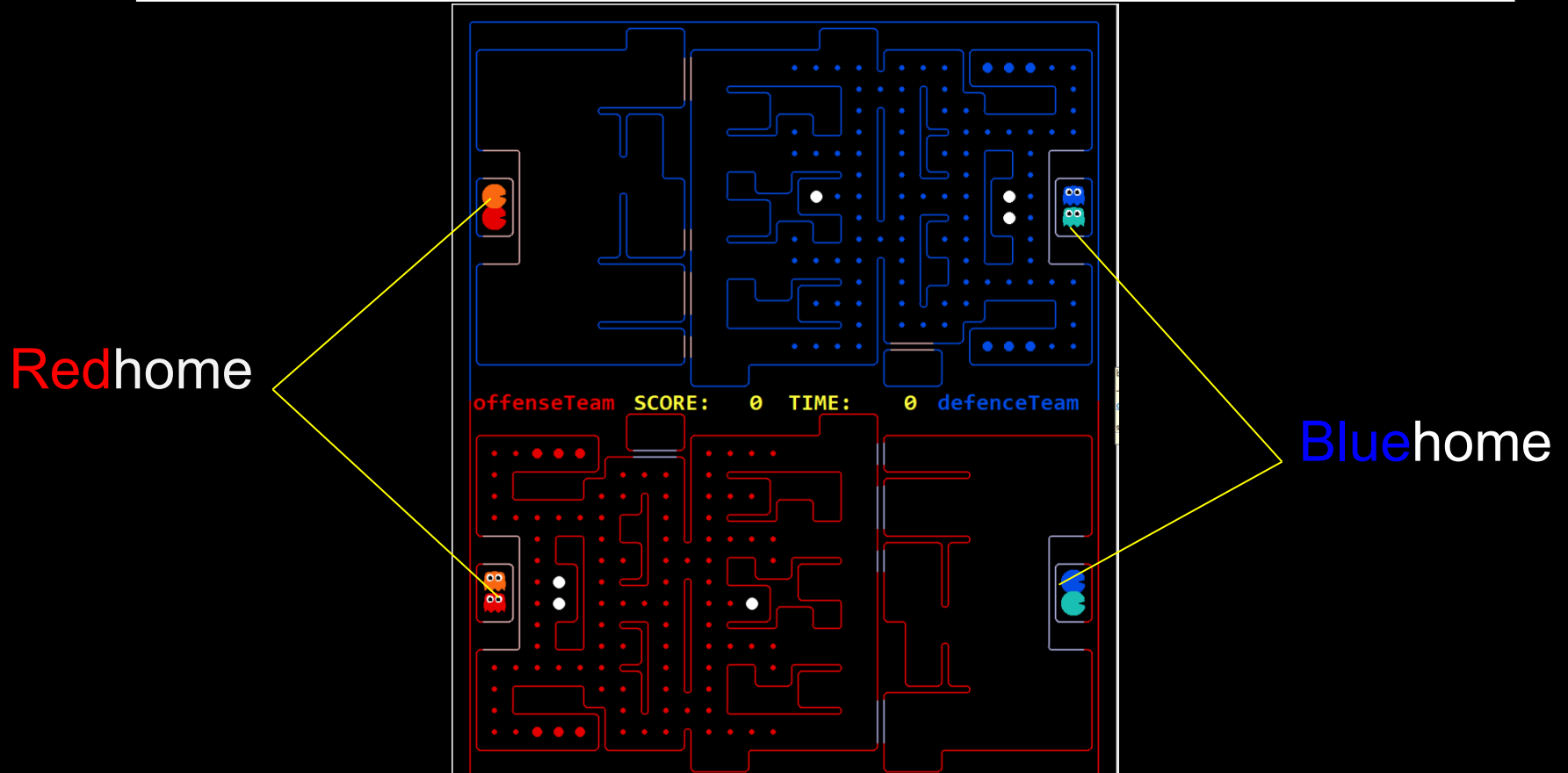


-
- Game Rules
 - Preparation
 - Tournament

-
- Game Rules
 - Preparation
 - Tournament

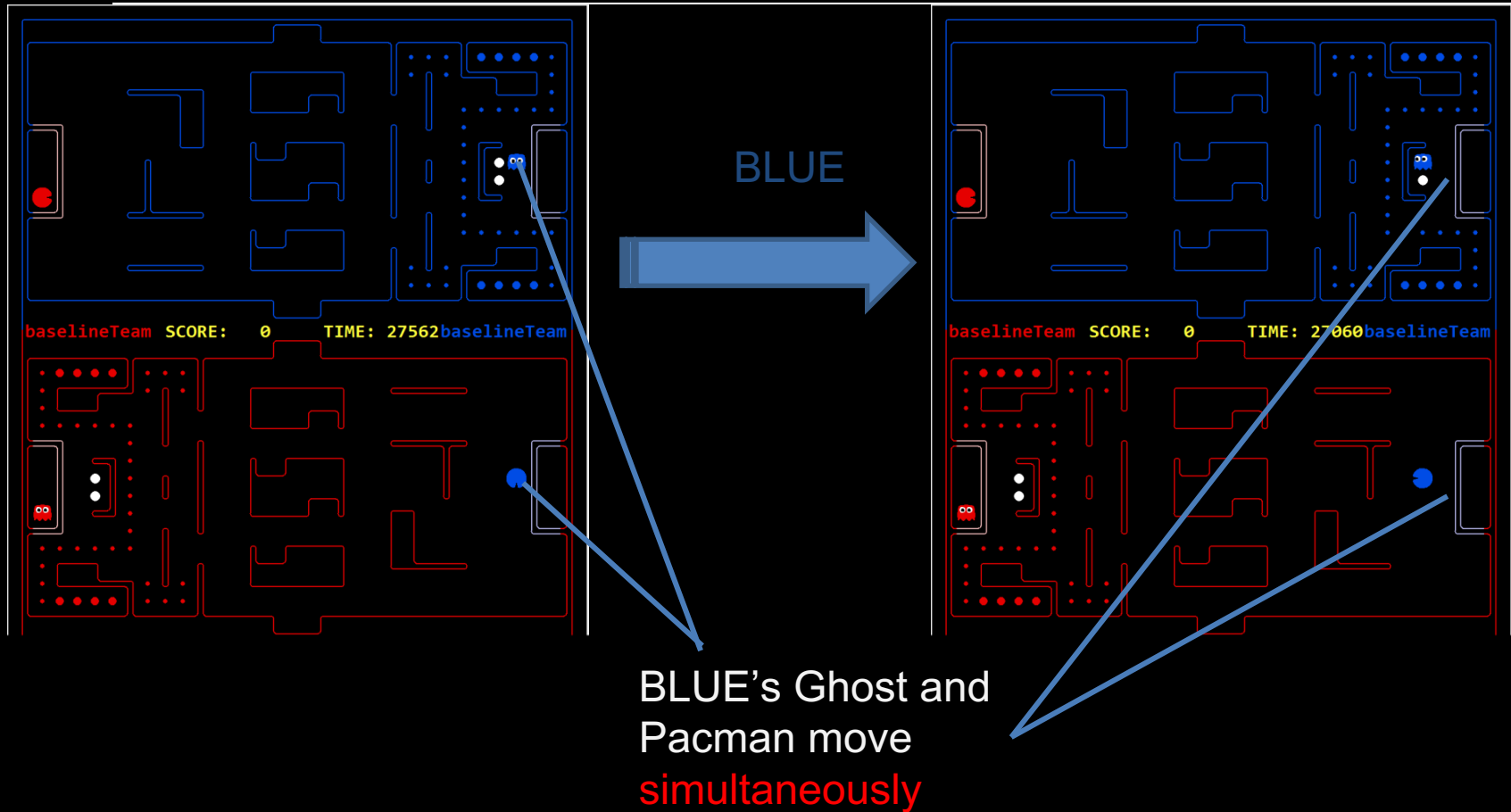
Game Rules(1/6)

Layout



Game Rules(2/6)

Movement



Game Rules(3/6)

Victory Condition

Time Limit: 1800 actions

As time's up, team with more score win.

Score: **Red** wins (+), **Blue** wins (-), Tie (0)

Game Rules(4/6)

Scoring

- Dot: 10
 - Big Dot: 50
 - Pacman: 150
 - Ghost(after eating capsule): 150
- capsule effect time: 100 units

Game Rules(5/6)

Respawn

- Dot: 250 units
- Big Dot: 625 units
- Capsule: 625 units

Game Rules(6/6)

Computation Time

- Limit for initialization
 - >15 sec: **Lose**
- Limit for actions
 - >1 sec: **Warning**
 - >3 sec: **Lose**
 - 3 **Warnings**: **Lose**
- Environment
 - Intel i7-6700 @ 3.2GHz
 - 16G RAM

-
- Game Rules
 - Preparation
 - Tournament

Preparation(1/3)

Implementation

- Restrictions
 - Rename the `00_myTeam.py` to `##_TeamName.py`
 - ◆ e.g. `50_TaipeiAssassins`
 - Your agent must inherit `CaptureAgent`
 - ◆ `registerInitialState()`
 - ◆ `chooseAction()`
 - Assign `2` agents to `createTeam()`
 - You can design any amount of agents
 - No multi-threading
 - Turn off all outputs
- Refer to `baselineTeam.py` for more information.

Preparation(1/3)

Implementation

There're 3 default teams:

1, defense team:

Both 2 pairs of agents focus on defense only

2, offence team:

Both 2 pairs of agents focus on offence only

3, baseline team:

1 pair of agents focus on defense

The other focus on offence

Preparation(2/3)

Index

- each agent has two indices, store in a tuple.

	agent 0	agent 1	agent 2	agent 3
pacman (index[0])	0	1	2	3
ghost (index[1])	4	5	6	7

Preparation(2/3)

APIs

- Class
 - GameState(capture.py)
 - CaptureAgent(captureAgents.py)

Preparation(2/3)

APIs - GameState

- Class GameState
 - getLegalActions(index)
 - generateSuccessor(index, action)
 - getAgentState(index)
 - getAgentPosition(index)
 - getInitialAgentPosition(index)
 - getRedTeamIndices()
 - getBlueTeamIndices()
 - isOnRedTeam(index)

Preparation(2/3)

APIs - GameState

- Class GameState
 - `getScore()`
 - `getRedFood()`
 - `getBlueFood()`
 - `getRedCapsules()`
 - `getBlueCapsules()`
 - `getCapsules()`
 - `getTransport()`
 - `getWalls()`
 - `hasFood(x, y)`
 - `hasWall(x, y)`

Preparation(2/3)

APIs - CaptureAgent

- Class CaptureAgent
 - chooseAction()//override this to make your agent.
 - getFood(gameState)
 - getFoodYouAreDefending(gameState)
 - getCapsules(gameState)
 - getCapsulesYouAreDefending(gameState)
 - getTransport(gameState)

Preparation(2/3)

APIs - CaptureAgent

- Class CaptureAgent
 - getOpponents(gameState)
 - getTeam(gameState)//your teammates
 - getTransport
 - getScore(gameState)
 - getMazeDistance(pos1, pos2)
 - getPreviousObservation()
 - getCurrentObservation

Preparation(3/3)

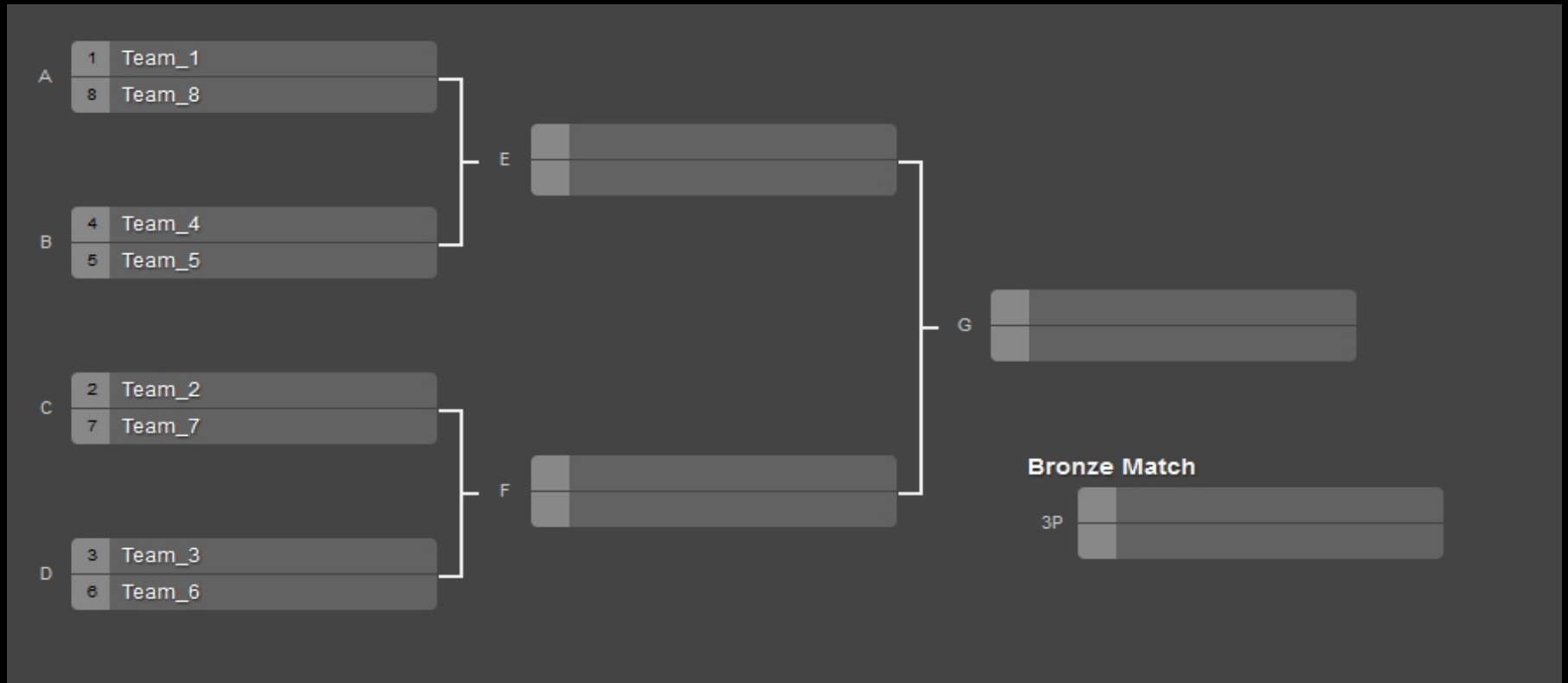
Options

- python capture.py
 - r [##_TeamName]: Load the **red** team
 - b [##_TeamName]: Load the **blue** team
 - l [Layout]: Load another layout
 - c: Catch exceptions and enforce time limits
 - n #: Play # games
 - q: quiet mode, no graphics
 - z: layout size
 - i: execution time(default 1800)
 - keys0: control the first agent of **red** with keyboard
 - keys1: control the first agent of **blue** with keyboard
 - First: WASD, Second: IJKL

-
- Game Rules
 - Preparation
 - Tournament

Tournament Bracket

- Round-Robin Qualifiers (offline)
- Top 8: Single Elimination (in class 6/8)



Schedule

- Upload `##_TeamName.py` to e3
- Submit Deadline: 6/2 23:59 (3 weeks +)
- Late Policy: 0%

- Round-Robin Qualifiers (6/2 - 6/7, offline)
- Top 8: Single Elimination (6/8, in class)

Grading

- According to your final results:
 - Crashed: 0
 - Runnable: 70
 - Beat Any Dufault Team: 5 for each team
 - Top 8: 90
 - 4th: 95
 - 3rd: 100
 - 2nd: 110
 - 1st: 120