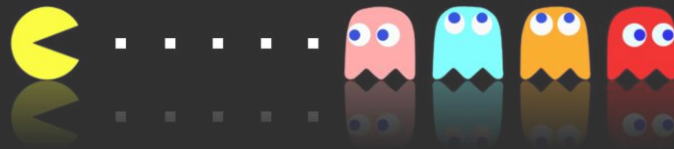


# PACMAN CAPTURE THE FLAG!



**Professor Jiwon Seo**  
**TA : Halim Lee, Taewon Kang**  
**Slide Credit : Taewon Kang**  
**Special Credit : Hyunjun Kim**

Based on UC Berkeley CS188 Class Material



# TIMELINE

WEEK	DATE					
1	3/7		프로젝트 소개, 팀빌딩. 아두이노 설치			강의1
2	3/14		아두이노 튜토리얼 (스위치, 초음파 센서, IR리시버)			2
3	3/21		2WD 터틀봇			3
4	3/28		실습1			강의 없음 (실습만 진행)
5	4/4		실습2			4
6	4/11		실습3			강의 없음 (실습만 진행)
7	4/18		Beat the Brick			5
8	4/25	중간고사 기간 (중간고사 없음)	ICT포럼 2019 참석으로 대체			
9	5/2		AI 프로젝트 설명	변형 팩맨	빗더브릭2	6
10	5/9		실습1		3	7
11	5/16		실습2		4	8
12	5/23		실습3		5	9 (총 9회 강의)
13	5/30		실습4		6	강의 없음 (실습만 진행)
14	6/5		코드 제출마감 (23시)			
14	6/6	현충일				
15	6/12		보고서 제출마감 (23시)			
15	6/13	자율학습 기간	팩맨 competition (17:30 시작)			
16	6/20	기말고사 (기말고사 있음)				



# **Video**

(Last year's final)

# What are we doing?

*Game = Compete to win*

- Analyze current state
- Expect opponents' action
- and choose your best action



## We will build a 'Game AI'





# What is Pacman?



# What is Capture the Flag?

You may have played this in your youth...

- Two teams / same flags each
- Each team should capture other team's flag and defend own team's flag
- The team which has more flags wins



# **About Pacman Capture the Flag**



# Game Screen



**SCORE: 0**

**RED: baselineTeam**

© 2019 IIT4312 Mechatronics Project

**BLUE: baselineTeam**

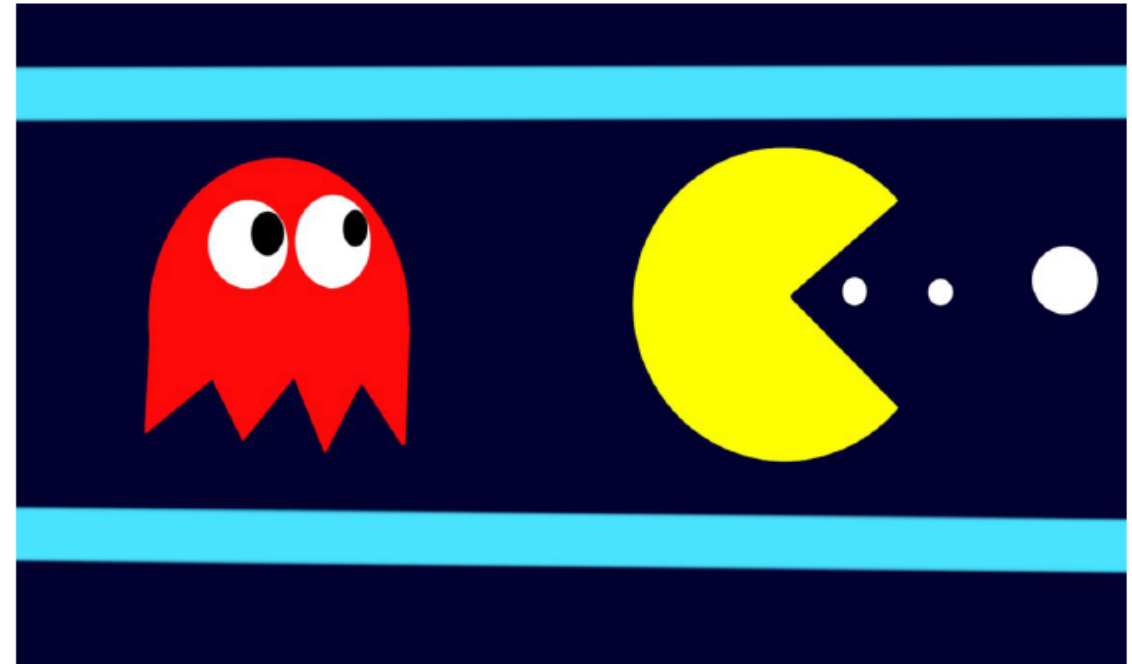
# Game Rules

- Red team vs. Blue team (two agents each)
- Each team gets a point when agents eat enemy camp's pellet and return your camp (**Doesn't score when you just eat pellets**)
- $\text{Score} = (\text{Red team score}) - (\text{Blue team score})$

# Game Rules

My agent becomes...

- Pacman (in the enemy camp)
  - Can eat pellet
  - Earns score when you return your side after eating pellets
  - **One pacman can carry 5 pellets at once**
- Ghost (in the our camp)
  - Cannot eat pellet
  - Can kill opposite pacman by chasing and eating it



# Game Rules

- Each agent has **0.5 second to return each action.**
- Each move which does not return within 0.5 second will incur a warning. After three warnings, or any single move taking more than 1.5 seconds, the game is forfeit.
- A game ends when one team returns all but two of the opponents' dots.
- Games are also limited to 1200 agent moves (300 moves per each of the four agents)
- If this move limit is reached, whichever team has returned the most food wins.
- If the score is zero (i.e., tied) this is recorded as a tie game.

# Q&A

## Q1. What happens when my Pacman is killed by a ghost?

- Pacman returns to the starting point.
- All eaten pellets by the Pacman are scattered in adjacent points

## Q2. What are those 'big pellets'?

- Power pellets
- When Pacman eats a power pellet, opponent team's ghost become 'scared' for 40 moves
- 'Scared' ghosts can be killed by Pacman
- Killed ghosts respawn at a starting point
- If 'scared' ghost goes to the opposing camp, the 'scared' state is released.

# **About Programming**



# About Programming Language

- We will use 'Python 3' for programming
- Useful links:
  - <https://wikidocs.net/43> (for beginners / python 2 based)
  - <https://tech.ssut.me/2015/07/24/python-3-is-the-future/> (for python 2 expertees)
  - <https://learnxinyminutes.com/docs/python3/> (quick prep)
  - <https://code.tutsplus.com/articles/python-from-scratch-object-oriented-programming--net-21476> (about Object Oriented Programming)

# Download Project Files

\* If you are using Windows, please set the Python as a environment variable[PATH]!

1. Download "IIT4312\_Pacman" folder:

[bit.ly/2H0RjT3](https://bit.ly/2H0RjT3)

2. To see if the project file is correctly downloaded:

```
> cd <프로젝트 폴더>/src  
> python3 capture.py
```

# Project Files in a Nutshell

## All Files

- autograder.py
- baselineTeam.py
- capture.py
- captureAgents.py
- captureGraphicsDisplay.py
- distanceCalculator.py
- game.py
- generateTournamentLayouts.py
- ghostAgents.py
- graphicsDisplay.py
- graphicsUtils.py
- keyboardAgents.py
- layout.py
- mazeGenerator.py
- myTeam.py
- pacman.py
- pacmanAgents.py
- score
- testClasses.py
- testParser.py
- textDisplay.py
- unpack.py
- util.py



## Actual Files Run

- baselineTeam.py
- capture.py
- captureAgents.py
- distanceCalculator.py
- game.py
- graphicsDisplay.py
- keyboardAgents.py
- layout.py
- myTeam.py
- util.py



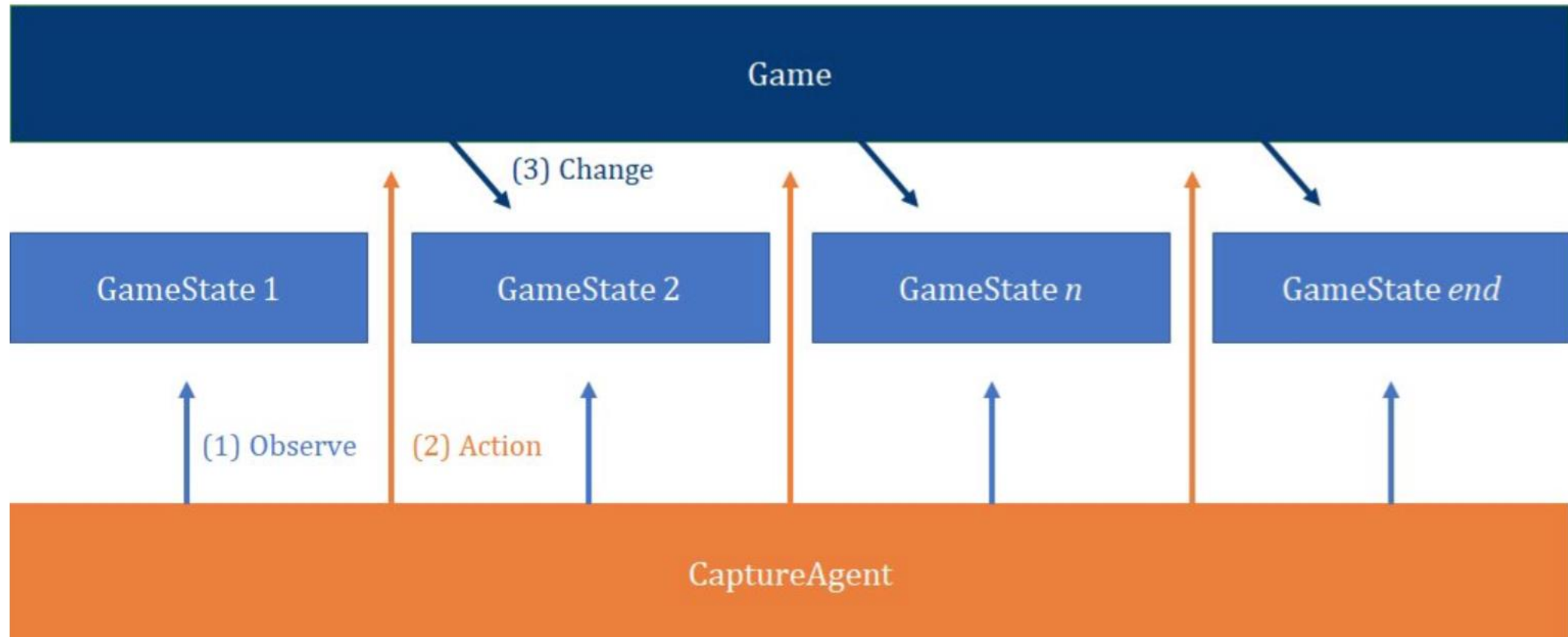
## Core Files

- capture.py
- captureAgents.py
- game.py

# Project Files in a Nutshell

- Only two classes you need to read & understand!
  - class **CaptureAgent** (captureAgents.py:46-304)
  - class **GameState** (capture.py:83-233)
- You'll have to
  - Derive class receives **CaptureAgent**
  - Observe **GameState** and program appropriate actions

# Project Files in a Nutshell



# class GameState

## 1. Typical functions

```
getLegalActions(agentIndex): # All legal actions that agent can do
generateSuccessor(agentIndex, action): # State after agent did certain action
getScore():
getRedFood():
getBlueFood():
getRedCapsules():
getBlueCapsules():
getAgentDistances():
hasWall(x, y):
hasFood(x, y):
```



# class GameState

## 2. Functions that give you ambiguous informations

`getAgentPosition(index):`

# 특정 Agent의 위치로, 적팀이라면 내 Agent와 5칸 이내 거리에 있어야 값을 읽어올 수 있다.

`getAgentDistances():`

# 모든 Agent간의 서로의 거리 리스트

`getAgentPosition(index)` : Exact but covers short range!

`getAgentDistances(index)` : Noisy but covers all range!

(Noise)  $\sim N(0, 3^2)$

# class CaptureAgent

## 1. Core functions

registerInitialState(gameState):

#처음에 초기화를 위해 한 번 불러지는 함수

chooseAction(gameState):

#주어진 gameState를 바탕으로 취할 Action을 결정. Action은 아래의 다섯 가지 중 하나를 선택할 수 있음

```
1  {Directions.NORTH: (0, 1),  
2    Directions.SOUTH: (0, -1),  
3    Directions.EAST: (1, 0),  
4    Directions.WEST: (-1, 0),  
5    Directions.STOP: (0, 0)}
```

# class CaptureAgent

## 2. Optional functions

getFood(gameState):

getFoodYouAreDefending(gameState):

getCapsules(gameState):

getCapsulesYouAreDefending(gameState):

getOpponents(gameState):

getTeam(gameState):

getScore(gameState):

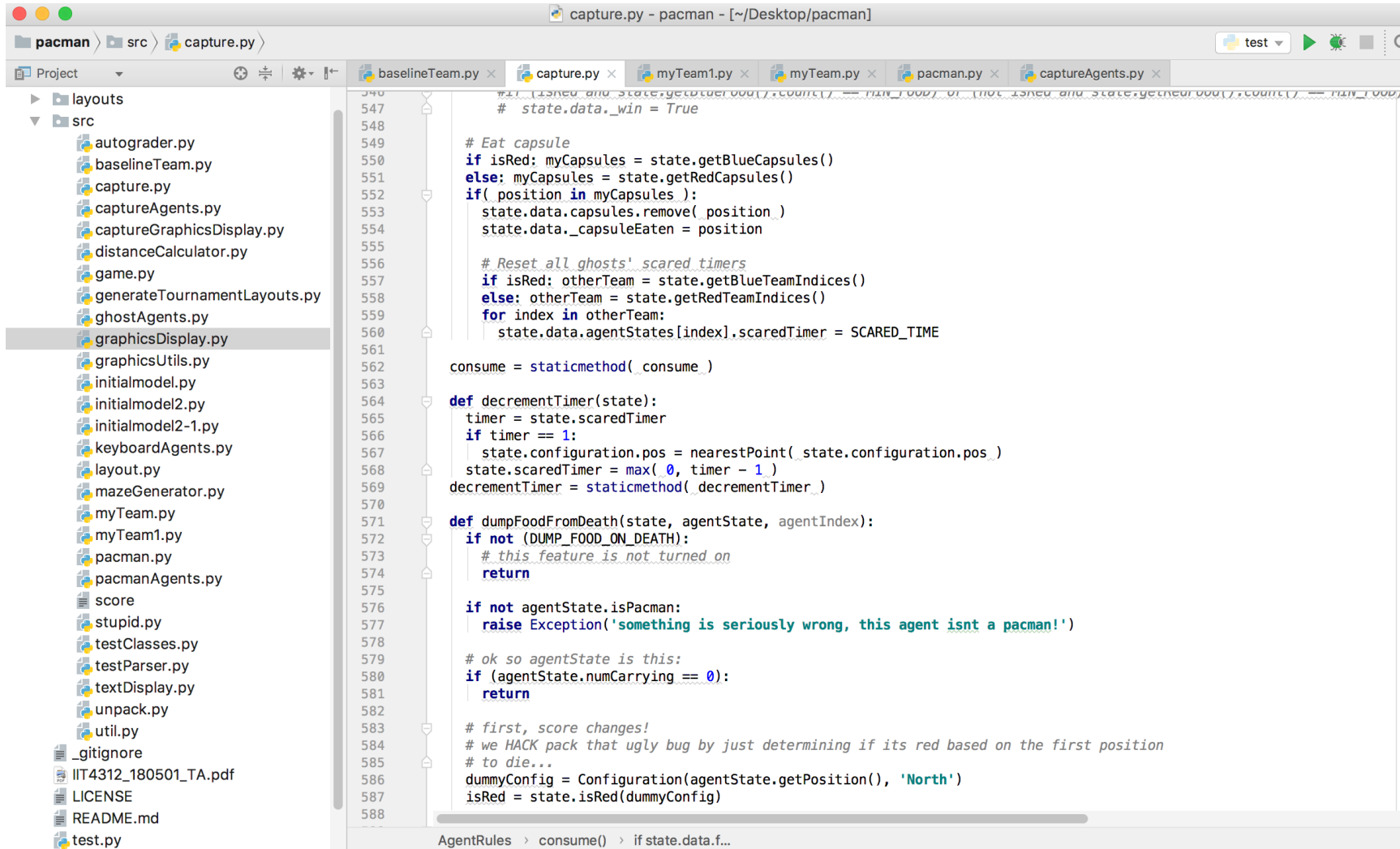
getMazeDistance(pos1, pos2):

getPreviousObservation():

getCurrentObservation():

debugDraw(cells, color, clear=False):

# Let's read the code together!



The screenshot shows a code editor window titled "capture.py - pacman - [~/Desktop/pacman]". The left sidebar displays a project tree with the following files and folders:

- layouts
- src
  - autograder.py
  - baselineTeam.py
  - capture.py
  - captureAgents.py
  - captureGraphicsDisplay.py
  - distanceCalculator.py
  - game.py
  - generateTournamentLayouts.py
  - ghostAgents.py
  - graphicsDisplay.py
  - graphicsUtils.py
  - initialModel.py
  - initialModel2.py
  - initialModel2-1.py
  - keyboardAgents.py
  - layout.py
  - mazeGenerator.py
  - myTeam.py
  - myTeam1.py
  - pacman.py
  - pacmanAgents.py
  - score
  - stupid.py
  - testClasses.py
  - testParser.py
  - textDisplay.py
  - unpack.py
  - util.py
- .\_gitignore
- IIT4312\_180501\_TA.pdf
- LICENSE
- README.md
- test.py

The main editor area shows the code for the `capture.py` file, with line numbers 540 to 588 visible. The code includes comments and Python code for handling capsules and team indices.

```
540 # if isRed and state.getBlueFood().count() == 0 or (not isRed and state.getRedFood().count() == 0):
541 # state.data._win = True
542
543 # Eat capsule
544 if isRed: myCapsules = state.getBlueCapsules()
545 else: myCapsules = state.getRedCapsules()
546 if (position in myCapsules):
547     state.data.capsules.remove(position)
548     state.data._capsuleEaten = position
549
550 # Reset all ghosts' scared timers
551 if isRed: otherTeam = state.getBlueTeamIndices()
552 else: otherTeam = state.getRedTeamIndices()
553 for index in otherTeam:
554     state.data.agentStates[index].scaredTimer = SCARED_TIME
555
556 consume = staticmethod(_consume_)
557
558 def decrementTimer(state):
559     timer = state.scaredTimer
560     if timer == 1:
561         state.configuration.pos = nearestPoint(state.configuration.pos)
562         state.scaredTimer = max(0, timer - 1)
563     decrementTimer = staticmethod(_decrementTimer_)
564
565 def dumpFoodFromDeath(state, agentState, agentIndex):
566     if not (DUMP_FOOD_ON_DEATH):
567         # this feature is not turned on
568         return
569
570     if not agentState.isPacman:
571         raise Exception('something is seriously wrong, this agent isnt a pacman!')
572
573 # ok so agentState is this:
574 if (agentState.numCarrying == 0):
575     return
576
577 # first, score changes!
578 # we HACK pack that ugly bug by just determining if its red based on the first position
579 # to die...
580 dummyConfig = Configuration(agentState.getPosition(), 'North')
581 isRed = state.isRed(dummyConfig)
582
```

The bottom status bar shows the current cursor position: "AgentRules > consume() > if state.data.f..."

# Let's code it together!

## Creating myTeam.py

1. Make new <Teamn.py>
2. createTeam function gets 3 parameters (firstIndex, secondIndex, isRed)
3. createTeam function returns each agent which receives CaptureAgent

```
1 def createTeam(firstIndex, secondIndex, isRed):  
2     return [AgentClass1(firstIndex), AgentClass2(secondIndex)]  
3  
4  
5
```

# Creating myTeam.py

Meet StupidAgent:

He/she gets legal actions he/she can and do it at random

```
1
2 class StupidAgent(CaptureAgent):
3
4     def registerInitialState(self, gameState):
5         CaptureAgent.registerInitialState(self, gameState)
6
7     def chooseAction(self, gameState):
8         actions = gameState.getLegalActions(self.index)
9         return random.choice(actions)
10
```



```
1  from captureAgents import CaptureAgent
2  import random
3  import game
4
5
6  def createTeam(firstIndex, secondIndex, isRed):
7      return [StupidAgent(firstIndex), StupidAgent(secondIndex)]
8
9  class StupidAgent(CaptureAgent):
10
11      def registerInitialState(self, gameState):
12          CaptureAgent.registerInitialState(self, gameState)
13
14      def chooseAction(self, gameState):
15          actions = gameState.getLegalActions(self.index)
16          return random.choice(actions)
17
```

- Test your AI by typing these in your command prompt:

```
> cd <프로젝트 폴더>/src  
> python3 capture.py -r <방금 만든 파일.py>
```

# Options for capture.py

## - Text `python capture.py --help` in your command window/terminal

- Run

```
python capture.py <옵션>
```

- Choosing side

```
-r redTeam.py -b blueTeam.py # RED팀 (redTeam.py), BLUE팀 (blueTeam.py) 설정 후 플레이  
# 만약 -r이나 -b를 설정하지 않으면, 기본 AI 팀 (baselineTeam.py) 이 설정됨
```

- Several options

```
-q # show your game result only in text  
--record # save your gameplay  
--replay # replay your saved gameplay
```

\* We will use `-l RANDOM` option for final contest

# Further Advices

- **baselineTeam.py** is a basic AI program given;  
Use it for understanding & forming a basic frame for your code
- You cannot use external libraries
- Discuss with your teammate & classmate for various strategies  
**But, Showing codes is strictly prohibited**

# About Computation Time

- **Reminder** : Each agent has **0.5 second to return each action**.
- To unify computation time measurement device, we will provide each team a SSH account which can connect to server computer and verify your computation time.
- You can copy your `<Teamn.py>` file in the project folder and check your code whether it violates the computation time rule.
- The following account ID and password will be given next week (May 9th).

# **About Final Contest**

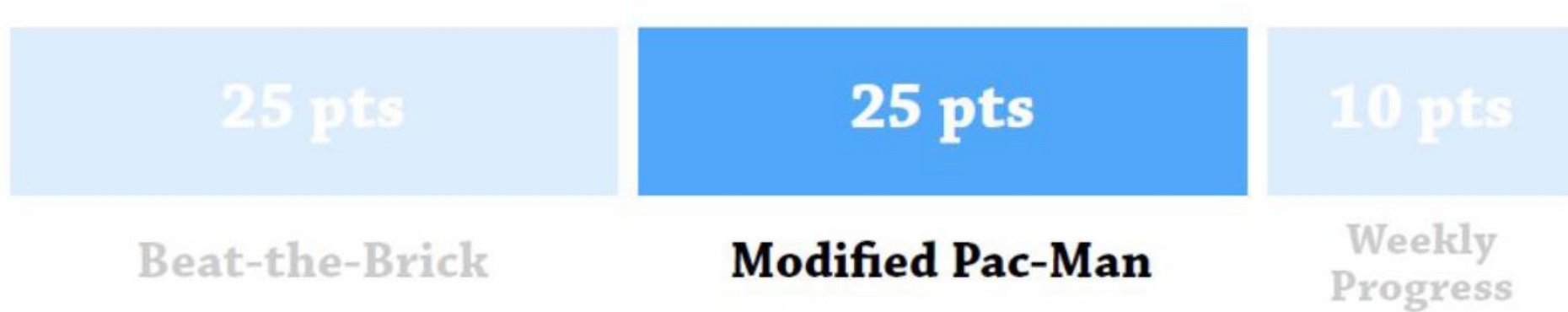


# Competition Rules

- **Preliminary : full league**
- **Semi-final, Final : tournament**
  - The 1<sup>st</sup> place team vs the 4<sup>th</sup> place team
  - The 2<sup>nd</sup> place team vs the 3<sup>rd</sup> place team

Please check handout and <Rules.pdf> file for detailed rules.

# Grading Criteria



**Team report :** 10 points

**Competition Result :**

1<sup>st</sup> : 15 pts, 2<sup>nd</sup> : 13 pts, 3<sup>rd</sup> & 4<sup>th</sup> : 10 pts, 5<sup>th</sup> & 6<sup>th</sup> : 7 pts, 7<sup>th</sup> : 5 pts

**Warning : Severe penalty if you cheat! (e.g. copying other's code)**

- We will use plagiarism detector for your submitted codes

# Project Schedule

- Competition date : June 13<sup>th</sup>
- Code submission date will be 8 days before the competition (June 5<sup>th</sup> 23:00)
  - You only need to submit <Teamn.py> you made.
  - **Late submission will not accepted.**
- Report submission date will be June 12<sup>th</sup> 23:00

**<Submission : mail [halim.lee@yonsei.ac.kr](mailto:halim.lee@yonsei.ac.kr)>**