Pacman Capture the Flag

Professor Jiwon Seo
TA: Taewon Kang, Hee-seung Moon
Special Credit: Hyunjun Kim

Based on UC Berkeley CS188 Class Material

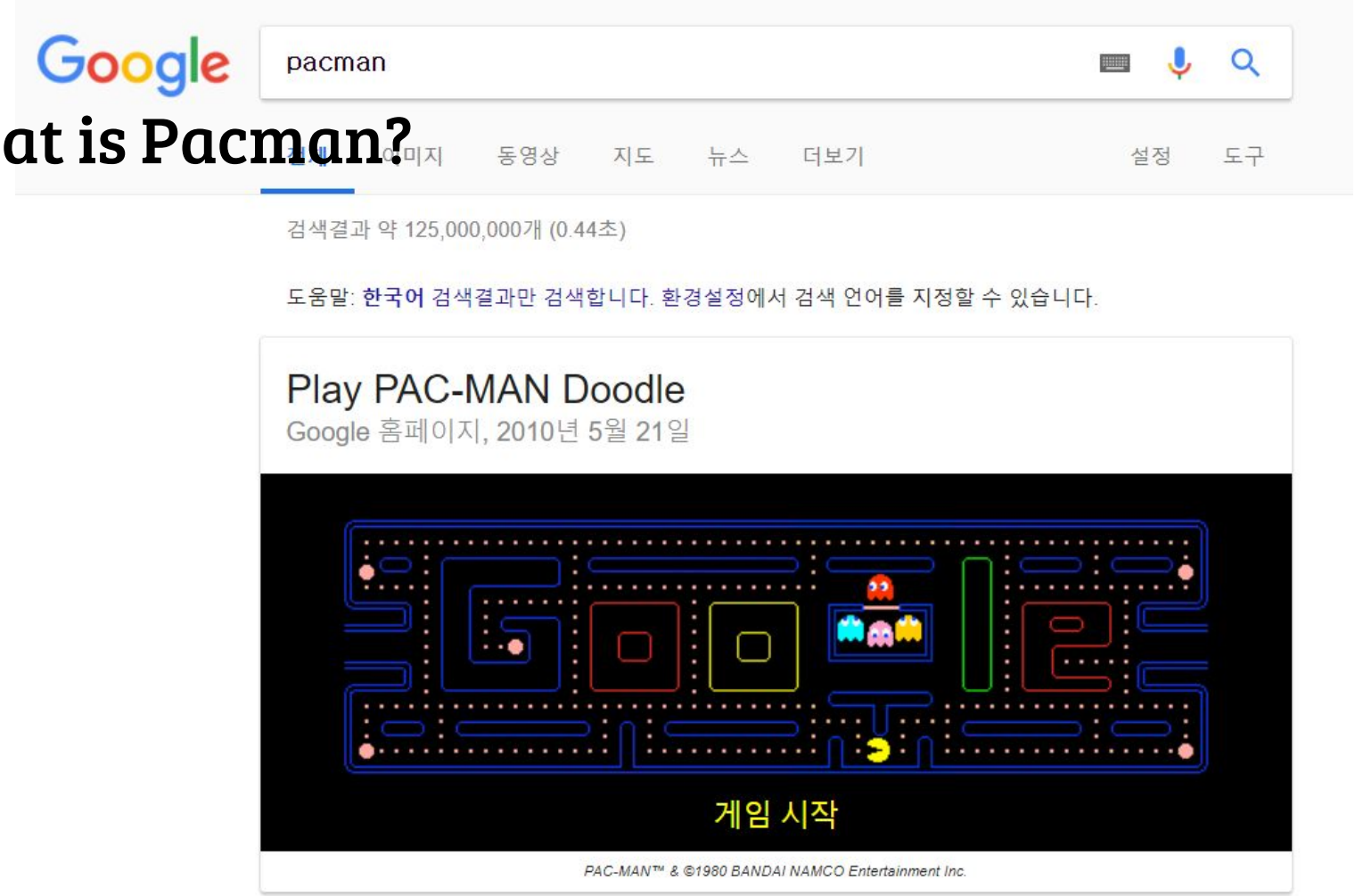
What are we doing?

Game = Compete to win

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

We will build a ‘Game AI’

What is Pacman?



사용자 의견

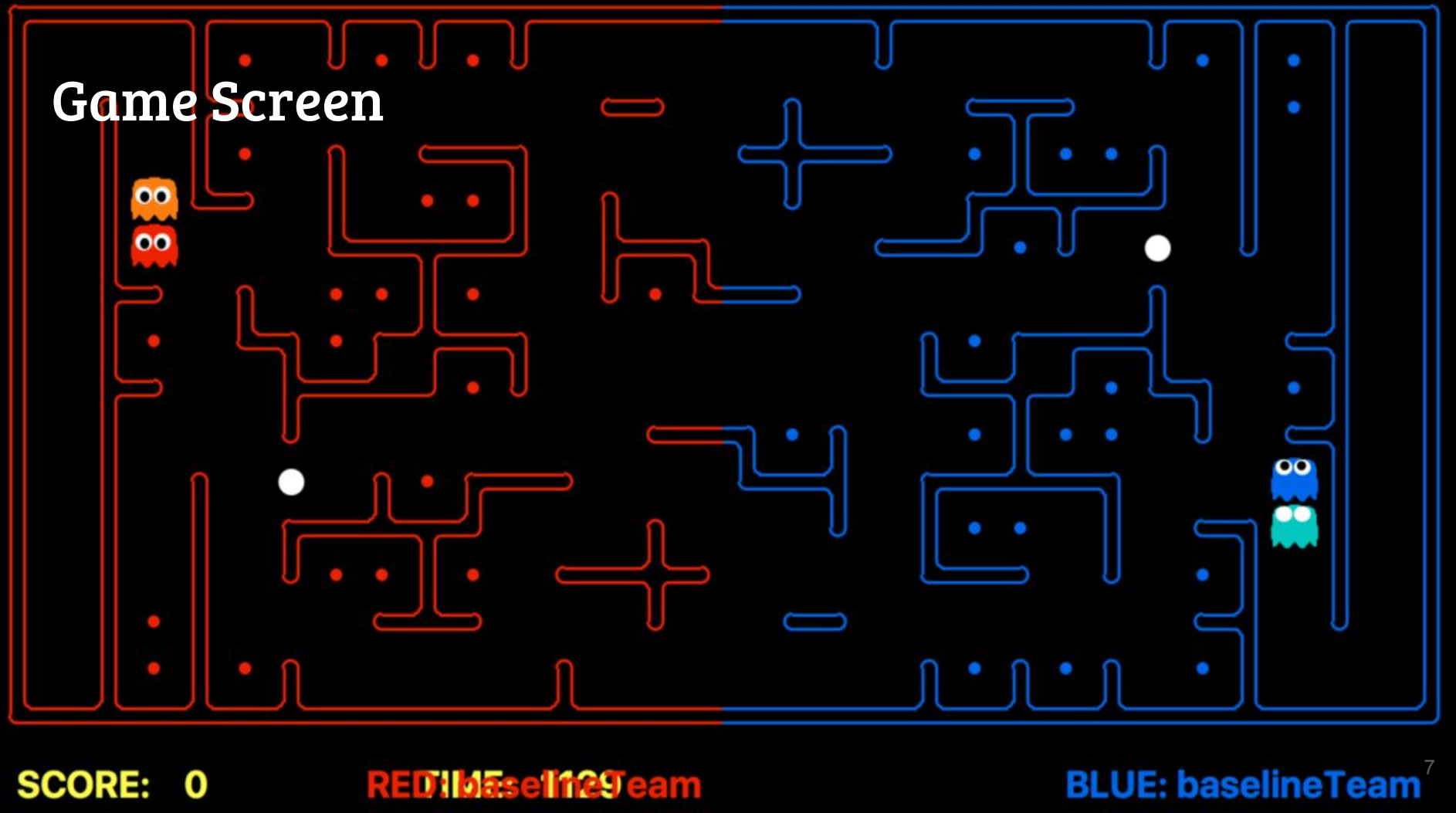
What is Capture the Flag?

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



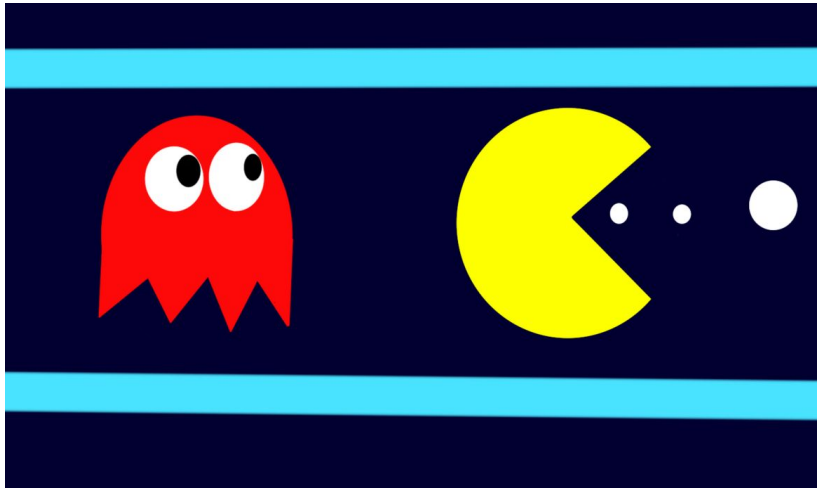
Game Rules

- 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**)
- Score = (Red team score) - (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.

Game Rules

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

- 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 ghosts become ‘scared’ for 40 moves
- ‘Scared’ ghosts can be killed by pacman
- Killed ghosts respawn at a starting point

About Programming

About Programming Language

- We will use '**Python 3**' for programming
- We will not lecture about grammars of python 3
- Here are some links that'll help:
  - 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; OOP)

Installing Python 3

https://www.python.org/

Windows:
https://zetawiki.com/wiki/%EC%9C%88%EB%8F%84%EC%9A%B0_Python_3_%EC%84%A4%EC%B9%98

MacOS:
https://zetawiki.com/wiki/%EB%A7%A5OS_python3_%EC%84%A4%EC%B9%98

Linux: Already installed

About Code Editor

Text Editor

VIM

Sublime Text

VS code

PyCharm

Easy / Less func. ⟷ Hard/ Various func.

Download Project Files

1. Download “pacman-capture-the-flag-master” folder:

   https://goo.gl/3WYwJZ

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

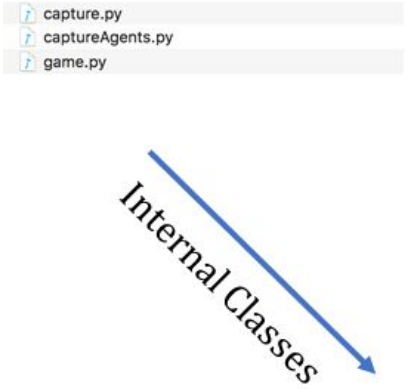
Actual Files Run

- 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



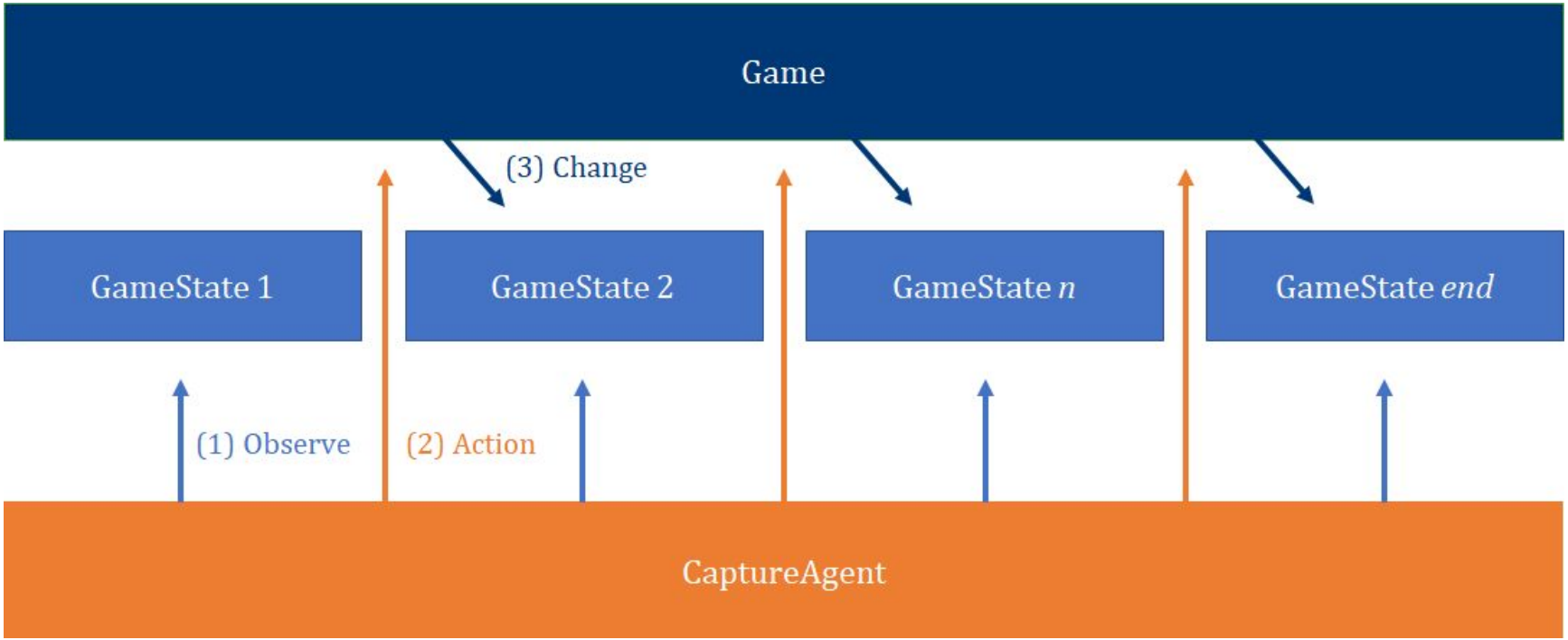
- **GameState (capture.py):** Involves all states in the game
- GameStateData (game.py)
- **Game (game.py):** Reflects **Agent**'s actions above **GameState**
- Configuration (game.py)
- **CaptureRules (capture.py):** Make progress to the game w.r.t. game rules
- AgentRules (capture.py)
- Agent (game.py)
- **CaptureAgent (captureAgents.py):** Define **Agent**'s state and action
- AgentState (game.py)
- Directions (game.py)
- **Action (game.py):** Define actions what **Agent** can do
- Grid (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 legel 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
getDistanceProb(trueDistance,noisyDistance):
실제 위치를 안다고 할 때, 실제 거리가 측정된 거리와 일치할 가능성

class CaptureAgent

1. Core functions

registerInitialState(gameState):
#처음에 초기화를 위해 한 번 불려지는 함수
chooseAction(gameState):
#주어진 gameState를 바탕으로 취할 Action을 결정. Action은 아래의 다섯 가지 중 하나를 선택할 수 있음

```
{Directions.NORTH: (0, 1),
 Directions.SOUTH: (0, -1),
 Directions.EAST:  (1, 0),
 Directions.WEST:  (-1, 0),
 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():
displayDistributionsOverPositions(distributions):

Creating myTeam.py

1. Make new <Team*n*.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
```

Creating myTeam.py

```python
from captureAgents import CaptureAgent
import random
import game


def createTeam(firstIndex, secondIndex, isRed):
  return [StupidAgent(firstIndex), StupidAgent(secondIndex)]

class StupidAgent(CaptureAgent):

  def registerInitialState(self, gameState):
    CaptureAgent.registerInitialState(self, gameState)

  def chooseAction(self, gameState):
    actions = gameState.getLegalActions(self.index)
    return random.choice(actions)
```

Creating myTeam.py

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

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

Running Project Files

- 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
```

Further Advices

- **baselineTeam.py** is a basic AI program given;

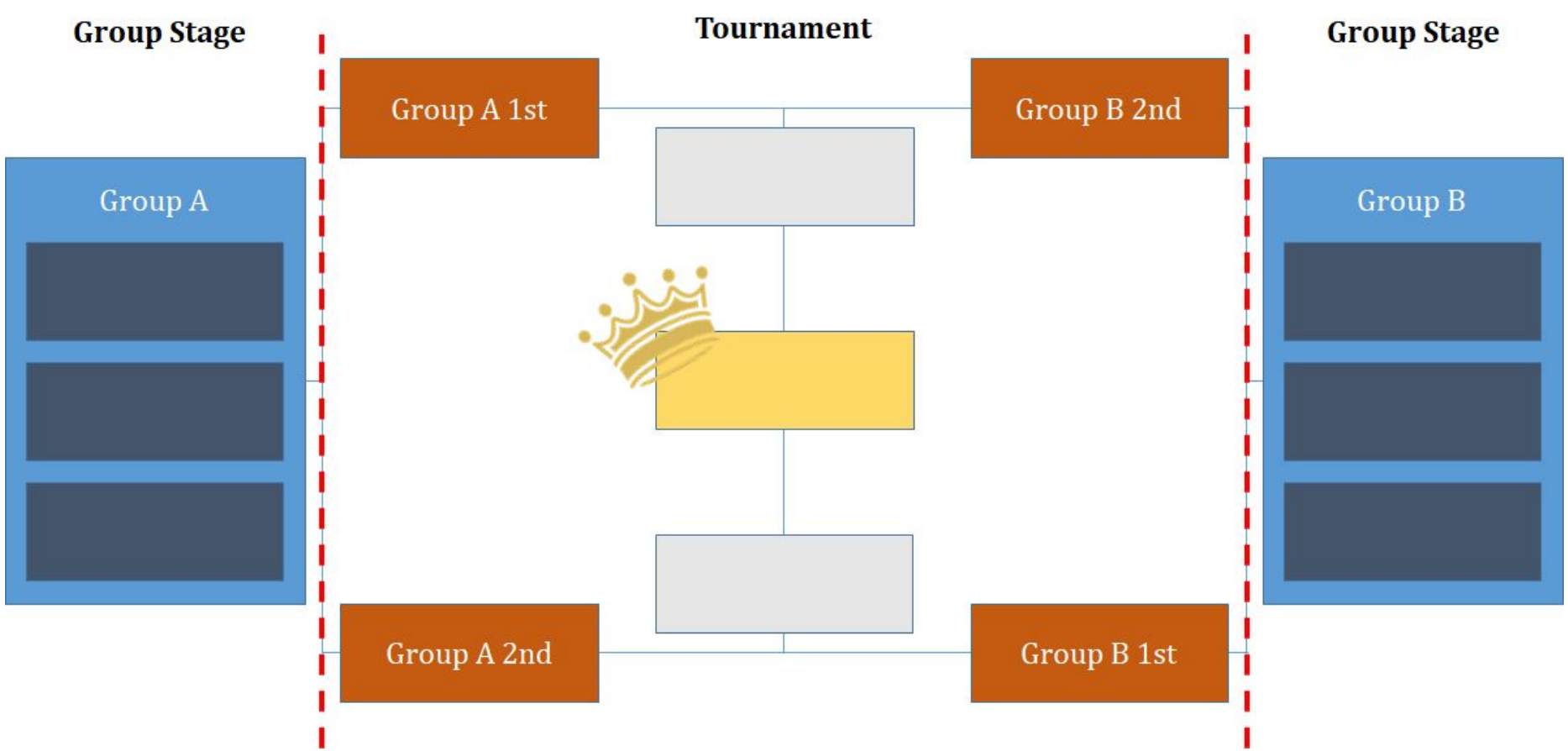
  Use it for understanding & forming a basic frame for your code

- You cannot use external libraries

- Discuss with your teammate & classmate (**Showing codes is strictly prohibited**) for various strategies

About Computation Time

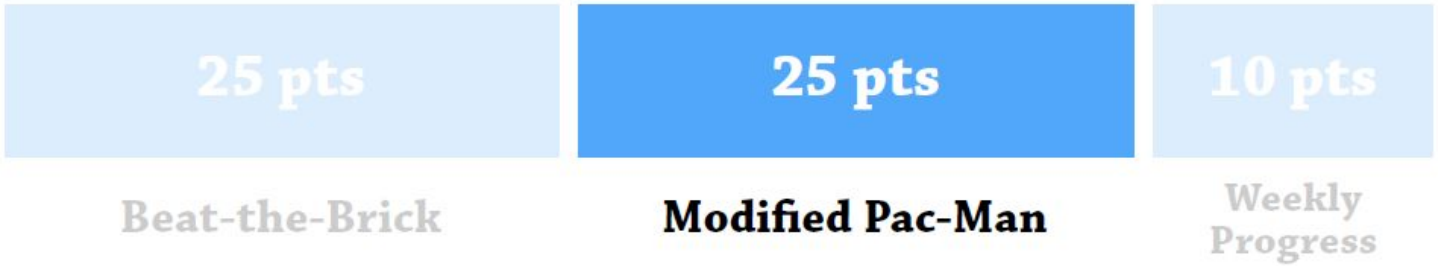
- **Reminder**: Each agent has **0.5 second to return each action.**
- To unify computation time measurement device, we will provide each team a SSH account which can connect to server computer and verify your computation time.
- You can copy your <Team*n*.py> file in the project folder and check your code whether it violates the computation time rule.
- The following account ID and password will be given next week (May 8th).


Group Stage
Tournament
Group Stage
Group A 1st
Group B 2nd
Group A
Group B
Group A 2nd
Group B 1st

Competition Rules

- Group Stage
  - 3 teams compete against each other 1 time (3 matches for one group)
  - 3 points for winning, 1 point for tie, 0 point for losing
  - Top 2 teams progress to tournament
  - If the records are tied, team with more score difference wins
  - If the records & score difference are tied, team with faster winning time wins
- Tournament
  - Two teams compete with the **Best of Three**
  - The record rules goes the same with the group stage

Grading Criteria



Team report: 10 points

Competition Result:

1st: 15 points, 2nd: 13 points, 3rd & 4th: 10 points, 5th & 6th: 7 points

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

- We will use plagiarism detector for your submitted codes

Project Schedule

- Competition date: June 12th (Tue)
- **Code submission date** will be

  "1 week before the competition (**June 5th 23:00**)"

  - You only need to submit <Team*n*.py> you made.
- **Report submission date** will be **June 11th 23:00**
- We will make confrontation table in June 5th

Submission: mail **taewon.kang@yonsei.ac.kr**