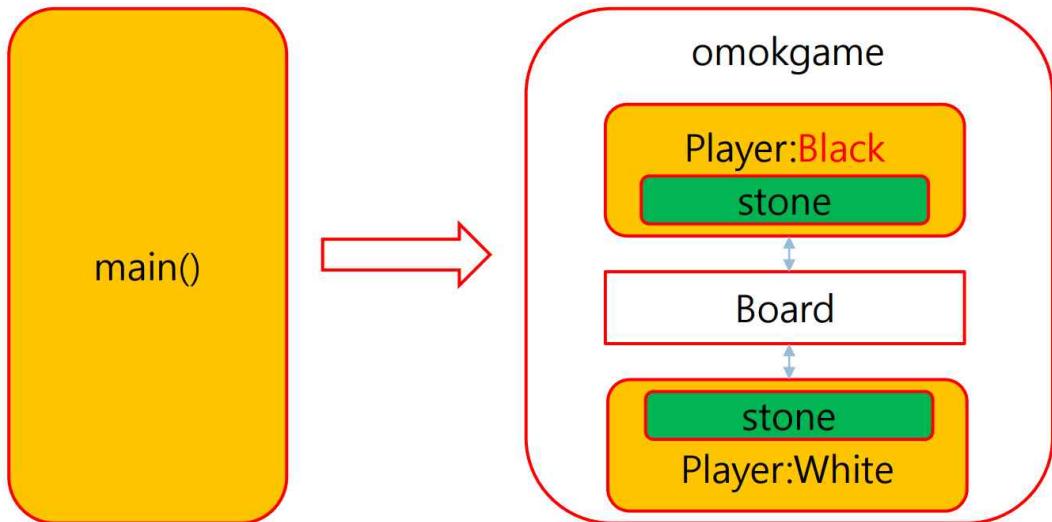


Battle용 인공지능 오목 프로그램 작성 및 평가 방법

## ■ Python용 오목 프로그램



### Python용 오목 프로그램

기본적으로 있어야할 코드 구성(나머진 마음대로)

main.py - Main function

Omokgame.py - Omokgame 클래스 - Game의 전반적인 진행을 관리

Board.py - Board 클래스 - 오목판의 상태를 관리

Stone.py - Stone 클래스, - 새로운 돌의 위치와 종료를 저장할 수 있으며, board와 , Play 간 정보 전달에 쓰임

Player.py - Player 클래스- 자기 차례가 왔을 때 다음에 둘 곳을 지정해주는 클래스이며, iot\_\*\*\* 클래스의 부모 클래스 - 인간에게서 키보드를 통해 black 돌의 위치를 받을 수 있는 클래스

Iot\_20203078\_KKR.py- Player 클래스를 상속한 AI 오목 (white ) player, - 난수를 이용해서 다음 수를 결정 : 학생들이 개선해야 할 클래스(흑돌용)

Iot\_20203078\_GIR.py- Player 클래스를 상속한 AI 오목 (white ) player - 난수를 이용해서 다음 수를 결정 : 학생들이 개선해야 할 클래스(백돌용)

오목 판 크기- 15 x 15 or 19 x 19

오목판의 각 칸- int 값을 가지며, 각각이 의미하는 것은 다음과 같다.

상태 = -1 의미 = 검은돌

상태 = 1 의미 = 흰돌

상태 = 0 의미 = 없음

Gomoku - Pro Rule- The black player puts the first move in the center of the board.- The white player will put the 2nd anywhere on the board.

- The black player will put the 3rd move outside 5x5 central square.

There are no forbidden moves. 6 or more in a row is not win.

게임 진행

각 수는 5초 이내에

두어야 함 (초과시 실격)- 잘못된 위치에 돌을 둘 경우도 실격

팀 별 리그전으로 모든 팀이 1대일 대결

3판 2승 (한번씩 흑, 백을 잡고 시합을 하며, 비길 경우 마지막)

3번째 판은 가위바위보를 하여, 흑백 결정 )- 2무일 경우 3번째 판이 존재- 1승 1무일 경우

1승자가 승리- 3번째 판이 비길 경우, 승패가 날때까지 2회 흑백을 바꾸어 가면 시합

( 승부가 나지 않을 경우 가위바위보로 승패 결정 )

등수 1 = 100

등수 2 = 95

등수 3 = 90

등수 4 = 85

등수 5 = 80

등수 6 = 75

등수 7 = 70