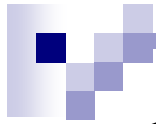




Programming - Practice 11

- **Tic-Tac-Toe simple AI**



Game AI

■ Rule-based approach

- 여러 단계의 수를 계산하기는 어려움

- 예: Dumb AI

1. 내가 이길 수 있는 곳에 둔다 (나 연속 2개)
2. 상대방이 두면 내가 지는 곳에 둔다 (상대방 연속 2개)
3. 5가 비어 있으면 둔다
4. 2, 4, 6, 8 순으로 비어 있으면 둔다
5. 1, 3, 7, 9 순으로 비어 있으면 둔다

1	2	3
4	5	6
7	8	9

- if-else 문으로 구현

```
if (position = gotTwo(X))
    board[position] = X;
else if (position = gotTwo(O))
    board[position] = X;
else if (isEmpty(5))
    board[5] = X;
.....
```

Game AI

■ Heuristics

□ 각 위치마다 점수를 저장하여 빈 곳 중 점수가 가장 큰 곳을 선택

■ heuristic[10] or heuristic[3][3] ...

■ 연속된 3개가 가능한 곳은 가로 3줄, 세로 3줄, 대각선 2줄

■ 8개 줄 각각에 대해 점수 계산하여 각 셀에 더해준다

■ 점수가 같은 곳이 있으면 임의로 선택

□ 예) Heuristic 1 (내가 X일 때):

내 말의 개수만큼 +1

상대 말의 개수만큼 -1

X	O	
	X	
		O

가로계산

0	0	0
1	1	1
-1	-1	-1

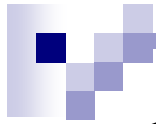
세로계산

1	0	-1
2	1	0
0	-1	-2

대각선

1	0	0
2	2	0
1	-1	-2

최대값(2)을 가진
빈 곳 4 선택



Game AI

■ Rule-based와 Heuristic 결합

□ 예시

1. 내가 이길 수 있는 곳에 둔다 (나 2개)
2. 상대방이 두면 내가 지는 곳에 둔다 (상대방 2개)
3. 내가 쌍 2를 만들 수 있는 곳에 둔다 (1개 있는 줄이 2개 이상)
4. 상대가 쌍 2를 만들 수 있는 곳에 둔다 (상대방 1개 있는 줄이 2개 이상)
5. Heuristic이 가장 높은 곳에 둔다

Game AI

■ Heuristics 개선

□ 공격과 방어를 고려

■ 필승 조건과 필패 조건을 고려한다

□ 예) Heuristic 2 (내가 X일 때):

나만 2개 있으면 +1000

상대만 2개 있으면 +100

나만 1개 있으면 +15

상대만 1개 있으면 +10

비어 있으면 +1

X	O	
	X	
		O

가로계산

0	0	0
15	15	15
10	10	10

세로계산

15	0	10
30	15	25
25	10	20

대각선

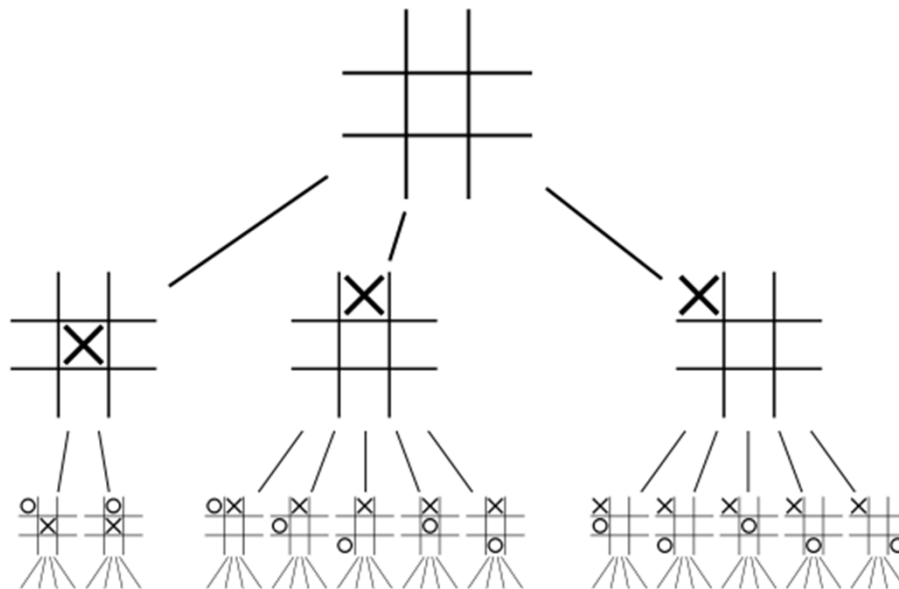
15	0	25
30	30	25
40	10	20

최대값(40)을 가진
7 선택

Game AI

■ Minimax search

- Tree 구조에서 최적의 경우 선택
- Heuristic 사용하여 Min(나), Max(상대)를 번갈아 선택
- 최악의 경우(Max) 가정하여 최선의 경우(Min) 선택



나만 2개 있으면 +1000
상대만 2개 있으면 +100
나만 1개 있으면 +15
상대만 1개 있으면 +10
비어 있으면 +1



▶ Tic-Tac-Toe simple AI

- 컴퓨터가 똑똑하게 두도록 만들어 보자
 - 모두가 최선의 선택을 하면 항상 비김
 - 상대방의 수를 고려해야 함 – Minimax 등의 기법
 - 고급 기법이 필요하므로 이 강좌에서는 구현하지 않는다
 - 현 상태만 고려하여 최선의 선택을 하도록 만들어 본다
 - Rule-based, Heuristic, 또는 둘을 혼합하여 구현
 - Computer가 두는 곳을 AI를 통해 선택한다
 - hack을 통하여 좀 더 똑똑히 만들 수 있다 (rule 추가 또는 heuristic 설계)
 - 이 강의자료에 나온 예시 중에는 Heuristic 2가 가장 우수함
 - 더 좋은 방법을 고안하여도 좋음
 - 제출
 - README.txt 파일을 만들어서 사용한 AI방법 설명
 - 소스코드와 README.txt파일을 .zip으로 압축하여 제출



▶ Challenge problem

- 단계적 AI Level을 만들어 보자 (20% 가산점!)
 - Smart, Dumb, Dumber의 세 단계로 AI level을 만들어 본다
 - Smart: 똑똑한 선택을 하는 단계 (이번 시간 목표)
 - Dumb: 지난 시간의 Challenge problem (p.2 rule-based 예시)
 - Dumber: 비어 있는 곳에 임의로 둔다 (순차적)
 - 메뉴에서 AI level을 바꿀 수 있게 한다
 - 컴퓨터 vs 컴퓨터 모드를 추가한다. 여기서 첫 번째 컴퓨터는 4번에서 선택한 AI level, 두 번째 컴퓨터 AI level은 랜덤하게 정한다.
 - 메뉴 예시:

```
### Tic-Tac-Toe ###
1 : Play first (X)
2 : Play second (O)
3 : Computer vs Computer
4 : Select AI (current: dumb)
0 : Exit
Enter command:
```
 - 첨부한 tictactoe.exe 참조