人工智能导论第一次作业实验报告

软件学院 刘博非 2021010612

问题一

井字棋中，状态空间较小，使用minimax实现的AI能够正常运行。

经过测试发现，无论先后手如何，AI都能找到一组不输策略。

问题二

在4×3的棋盘上:

minimax搜索在第一次落子时，由于状态空间较大，耗时较长，超过10min;

alpha-beta搜索第一次落子只需要2.45s，可见通过alpha-beta剪枝，速度提升较为明显。

经测试，发现在此情况下，AI为先手必胜。

问题三

评估函数采用非对称式设计，从当前(即将落子的)玩家的角度考虑局面的优劣:

- 若己方有活四或冲四，则必胜，直接返回0.99(不返回1是为了避免在能一步胜利的情况下，走到有活四或冲四的情况)
- 若对方有活四，则必输，直接返回-1
- 棋子距中心位置，对局面影响较小，权重取1
- 己方有活三，则一步即可为活四，赢面较大，权重为15
- 对方有冲四或活三，则必须封堵，其中冲四的封堵优先级更高，权重为16，活三则为14
- 冲三和活二对局面影响不太大，权重取值较小，且保证活二大于冲三(冲三极易被堵)即可
- 最后将算的权重处以100，以保证返回值在$[-1,1]$之间

具体代码如下

```
def detailed_evaluation_func(state):
    # TODO
    player = state.get_current_player() # the score is calculated from the
perspective of the current player
```

```
    info = state.get_info()
    score = 0.0 # score is [-1, 1]
    will_win = False
    will_loss = False
    for p, info_p in info.items():
        if p == player: # the probability to win
            if info_p["live_four"] > 0 or info_p["four"] > 0:
                will_win = True # return 0.99 # guarantee to win
                # 不应当在此直接返回，会和对手的活四产生顺序问题
            score -= info_p["max_distance"]
            score += info_p["live_three"] * 15
            score += info_p["three"] * 2
            score += info_p["live_two"] * 3
        else: # the risk to lose
            # 从当前玩家的角度，对手有活四，那么当前玩家必输
            if info_p["live_four"] > 0:
                will_loss = True # return -1
            score += info_p["max_distance"]
            # 对手有冲四或活三，都必须封堵
            score -= info_p["four"] * 16
            score -= info_p["live_three"] * 14 # very dangerous
            score -= info_p["three"] * 3
            score -= info_p["live_two"] * 5
    if will_win: # 己方优先级更高
        score = 99
    elif will_loss:
        score = -100
    # print("score",score)
    score = score / 100 # normalize to [-1, 1]
    return score
```

需要注意的是，我一开始以为一个depth是一次落子，即评估函数需评估AI落子后的局面，后来经助教讲解认识到应该是两次落子，即"one depth = two plies"，所以评估的还是AI即将落子的局面。

问题四

MCTS与alpha-beta搜索进行对战时，alpha-beta函数表现明显更好。

由于MCTS在落子时，通过随机模拟对局面进行判断，往往不能很好地估计局面，一方面无法对对手的进攻进行有效的防守，另一方面也即使在必胜局面也不能保证胜利，因此效果不佳。

问题五

在MCTS与AlphaZero进行对战时，无论先后手，AlphaZero均取得胜利。

可见，AlphaZero行棋更为合理。在蒙德卡洛搜索的过程中，通过人为定义的评估函数来对局面进行评估，能够在对战中较为合理地行棋。