Baseline Heuristic (Movement)

Search Method	Depth	Result
Minimax	3	Your agent won **50.0%** of matches
Minimax + Alpha/Beta	3	Your agent won **50.0%** of matches
Minimax + Alpha/Beta	4	Your agent won **53.8%** of matches
Minimax + Alpha/Beta	5	Your agent won **63.1%** of matches
Minimax + Alpha/Beta	6	Your agent won **65.3%** of matches
Minimax + Iterative Deepening	-	Your agent won **79.7%** of matches
Minimax + Iterative Deepening + Alpha/Beta	-	Your agent won **78.1%** of matches

Conclusion:
The baseline movement heuristic demonstrates steady improvement with deeper searches when using alpha-beta pruning. At depth 5 and beyond, the agent performs significantly better than at lower depths. The introduction of iterative deepening further boosts performance, indicating that being able to handle more depths within a given time limit is critical. However, the minimal difference between iterative deepening alone and the combination with alpha-beta suggests that pruning contributes slightly but not decisively beyond a certain depth.

Hybrid Heuristic with standard parameters

Search Method	Depth	Result
Minimax	3	Your agent won **50.0%** of matches

Search Method	Depth	Result
Minimax + Alpha/Beta	3	Your agent won **43.4%** of matches
Minimax + Alpha/Beta	4	Your agent won **55.3%** of matches
Minimax + Alpha/Beta	5	Your agent won **%** of matches (hangs)
Minimax + Alpha/Beta	6	Your agent won **%** of matches (hangs)
Minimax + Iterative Deepening	-	Your agent won **63.1%** of matches
Minimax + Iterative Deepening + Alpha/Beta	-	Your agent won **76.6%** of matches

Conclusion:
With standard parameters, the hybrid heuristic performs comparably to the baseline heuristic at depth 3 but slightly underperforms with alpha-beta pruning at the same depth. However, the results begin to converge and slightly surpass the baseline at depth 4. Hangs at deeper depths could indicate inefficiencies in the heuristic, such as slow evaluation or poor move ordering, possibly causing the search to exhaust its time limit. Iterative deepening significantly improves performance, showing that allowing the algorithm to progressively explore deeper levels is key to harnessing the hybrid heuristic's full potential.

Parameters for Hybrid Heuristic

Parameter	Current Value	Description
`mobility_weight`	1.0	Weight assigned to the immediate mobility (number of available moves).
`future_mobility_weight`	0.5	Weight assigned to future mobility (flexibility after making a move).
`proximity_weight`	-0.2	Penalty applied based on proximity to the opponent (negative value encourages distance).

Parameter	Current Value	Description
`trap_bonus`	-100	Large penalty applied when opponent has 2 or fewer moves remaining.

Hybrid Heuristic with different (badly tuned) parameters

Search Method	Depth	Result
Minimax	3	Your agent won **32.8%** of matches
Minimax + Alpha/Beta	3	Your agent won **39.2%** of matches
Minimax + Alpha/Beta	4	Your agent won **49.5%** of matches
Minimax + Alpha/Beta	5	**hangs**
Minimax + Alpha/Beta	6	**hangs**
Minimax + Iterative Deepening	-	Your agent won **48.8%** of matches
Minimax + Iterative Deepening + Alpha/Beta	-	Your agent won **72.5%** of matches

Conclusion:
These results highlight the sensitivity of the hybrid heuristic to its parameters. Badly tuned parameters lead to a substantial drop in performance compared to the standard setup. The agent's winning percentage dips to as low as 32.8% in some configurations. The hangs observed at deeper depths, especially with alpha-beta pruning, suggest that poor parameters might be forcing the agent into difficult-to-evaluate positions or creating evaluation bottlenecks. The relatively low performance with iterative deepening further emphasizes the importance of carefully selecting heuristic parameters.

Different Parameters for Hybrid Heuristic

Parameter	Current Value	Description
`mobility_weight`	0.5	Weight assigned to the immediate mobility (number of available moves).
`future_mobility_weight`	1.0	Weight assigned to future mobility (flexibility after making a move).
`proximity_weight`	-0.4	Penalty applied based on proximity to the opponent (negative value encourages distance).
`trap_bonus`	-50	Large penalty applied when opponent has 2 or fewer moves remaining.

Hybrid Heuristic with standard parameters and increasing time limit

Search Method	Time Limit /ms	Result
Minimax + Iterative Deepening + Alpha/Beta	150	Your agent won **76.0%** of matches
Minimax + Iterative Deepening + Alpha/Beta	300	Your agent won **76.5%** of matches
Minimax + Iterative Deepening + Alpha/Beta	450	Your agent won **80.5%** of matches
Minimax + Iterative Deepening + Alpha/Beta	600	Your agent won **81.0%** of matches

Conclusion:
Increasing the time limit yields slight improvements in performance, but the rate of improvement diminishes after a certain point. Between 150ms and 300ms, the agent's win rate remains largely unchanged, but beyond 300ms, there is a modest gain in performance. This suggests that the agent is able to explore additional depths and improve its decision-making when more time is available, but the hybrid heuristic's performance plateaus as the depth gains become marginal in the additional time.

Tuning Parameters for Hybrid Heuristic

Parameter	Current Value	Description
`mobility_weight`	1.5	Weight assigned to the immediate mobility (number of available moves).
`future_mobility_weight`	1.0	Weight assigned to future mobility (flexibility after making a move).
`proximity_weight`	-0.2	Penalty applied based on proximity to the opponent (negative value encourages distance).
`trap_bonus`	-50	Large penalty applied when opponent has 2 or fewer moves remaining.

Trap Bonus Condition: `trap_bonus = -50 if len(opp_liberties) <= 2 else 0`

Questions to answer for the report:

1. What features of the game does your heuristic incorporate, and why do you think those features matter in evaluating states during the search?
2. Analyze the search depth your agent achieves using your custom heuristic. Does search speed matter more or less than accuracy to the performance of your heuristic?

Answer to Question 1

My **hybrid_heuristic** builds on the movement heuristic but adds additional layers of evaluation to consider a broader range of factors. Here's a breakdown:

Hybrid Heuristic:

- **Movement Component**: Like the movement heuristic, it evaluates the difference between the player's available moves and the opponent's available moves, prioritizing mobility.

- **Future Mobility**: It estimates the future mobility for both the player and the opponent by simulating potential moves, adding another dimension of foresight to the evaluation.
- **Proximity to Opponent**: It includes a penalty based on the distance between the player and the opponent, encouraging the agent to maintain a strategic distance.
- **Trap Bonus**: A large penalty is applied when the opponent has very few moves (2 or fewer), potentially leading to a strategic trap.

Key Differentiation from Movement Heuristic:

- **Complexity**: The hybrid heuristic evaluates not only immediate mobility but also future mobility and the strategic importance of proximity and trapping the opponent.
- **Strategic Depth**: By considering these additional factors, the hybrid heuristic aims to account for longer-term planning, while the movement heuristic is purely focused on the immediate situation.

Answer to Question 2

The search depth my agent achieves using the custom hybrid_heuristic depends heavily on the time available for decision-making. While search speed allows the agent to explore deeper levels of the game tree, the performance of my custom heuristic shows that a balance between speed and accuracy is crucial. In my experiments, the heuristic's overall accuracy significantly impacts performance, as poorly tuned parameters lead to bottlenecks and diminished returns from additional search depth. Thus, accuracy of the heuristic evaluation often outweighs sheer depth, particularly when the heuristic is well-calibrated to the game's dynamics.

However, when both accuracy and speed are optimized—through techniques like alpha-beta pruning and iterative deepening—the agent performs best, showing that while depth matters, it's the quality of the evaluations at each depth that truly determines success.

Overall conlusion of using a more complex heuristic than the baseline (movement) heuristic

For Knight's Isolation, immediate mobility is a critical aspect of the game, and the movement heuristic captures this perfectly. While more complex heuristics like my hybrid_heuristic may introduce interesting strategic considerations, they might not yield significant improvements due to the relatively simple structure of the game. The movement heuristic may already be close to an optimal strategy because it aligns directly with the core objective of maintaining mobility and restricting the opponent's options.