Solution Summary

Overall Approach

For this assignment, we implemented an autonomous agent for FAUhalma that plays as player A on the AISysProj server. Instead of using deep adversarial search, we deliberately chose a greedy, heuristic-driven approach. This decision was mainly motivated by practicality: FAUhalma allows long multi-hop jump moves and involves up to three players, which quickly leads to a very large branching factor. A greedy agent with a well-designed heuristic can make consistent progress while staying computationally lightweight and reliable within the server's timing constraints.

The implementation is structured in a modular way. State representation, move generation, heuristics, and agent logic are clearly separated. This made it easier to ensure rule correctness first and then focus on improving the quality of decisions made by the agent.

Board Representation and State Modeling

The game state is represented directly using the axial coordinate system provided by the server. Each state consists only of the peg positions for players A, B, and C, stored as immutable tuples of coordinates. This mirrors the server's JSON representation closely and avoids unnecessary conversions or additional abstraction layers.

All board-specific constraints such as valid board cells for star and rhombus shapes, the removed center cell, and the definition of start and home regions are handled through constant definitions. Distances are computed using standard hex-grid techniques by converting axial coordinates to cube coordinates and applying the max-norm distance. This distance measure provides a consistent notion of "progress toward home" independent of the visual orientation of the board.

State transitions are handled functionally: applying a move produces a new state instead of mutating the existing one. This simplifies reasoning about correctness when evaluating multiple candidate moves.

Move Generation and Rule Handling

Correct move generation is the foundation of the agent. We explicitly generate all legal adjacent moves and all legal jump moves for a given state. Adjacent moves are straightforward one-step moves in any of the six axial directions, provided the destination cell is valid and not the removed center.

Jump moves require more careful handling. For each axial direction, we consider increasingly distant target cells and examine the intermediate cells between source and target. A jump is considered legal only if all intermediate cells lie on the board, the removed center cell is never crossed, at least one intermediate cell is occupied, and the pattern of occupied versus empty intermediate cells is symmetric. Importantly, symmetry is checked purely on occupancy, not on which player owns the pegs. This follows the formal rule description and avoids incorrectly rejecting legal jumps that pass over mixed opponent pieces.

The swap rule is also implemented during state transitions: if a move lands in the moving player's home region and the destination is occupied by an opponent, the pegs are swapped.

While this situation is relatively rare, modeling it ensures that the internal state always remains consistent with the official game rules.

Greedy Agent and Heuristic Design

The agent itself follows a greedy decision policy. On each turn, it enumerates all legal moves for player A, evaluates each move using a heuristic scoring function, and selects the move with the highest score. Designing this heuristic was the main "don't-know" decision of the assignment.

The core heuristic component measures how much a move improves A's total distance to its home region. This directly reflects the primary objective of FAUhalma: getting all pegs into the opposite corner as quickly as possible. On top of this, longer jumps are rewarded, since they often represent a significant positional gain in a single move. A small directional bias encourages movement toward A's target corner, helping to avoid sideways or backward moves that do not contribute to long-term progress.

In addition, the heuristic includes a relative-progress component that compares A's distance-to-home with the leading opponent. This helps the agent prefer moves that not only improve its own position but also maintain or increase its advantage in a multi-player setting. Finally, moves that place a peg directly into the home region receive an explicit bonus, reflecting their strategic importance near the endgame.

Practical Behavior Across Environments

In practice, this greedy strategy performs consistently in the targeted three-player star environments. In particular, the agent shows stable and competitive behavior in the ws2526.1.2.7 configuration, where its heuristic aligns well with the board dynamics and typical opponent behavior. In more demanding environments such as ws2526.1.2.8, the same greedy approach still produces valid and reasonable play, although the increased complexity and stronger opposition make it harder to achieve the same level of dominance. This highlights a natural limitation of greedy decision-making without deeper lookahead, rather than a correctness issue.

Conclusion

Our solution focuses on accurate modeling of FAUhalma's geometry and movement rules combined with a carefully designed greedy heuristic. The key insights are treating the board as an axial hex grid with cube-distance evaluation, generating jump moves by enforcing symmetric occupied/empty patterns across valid intermediate cells, and scoring moves based on progress toward home, jump efficiency, and relative advantage. While simple in structure, this approach yields a rule-correct agent that behaves reliably on the server and achieves solid results in practice.