Decoy Movement Algorithm for Mr. X

1. Verbal Explanation

The Decoy Movement algorithm allows Mr. X to perform risky moves in order to mislead detectives while maintaining a minimal level of safety.

Note: Safety and risk are unified into a single evaluation function called `field_score`. High values indicate more aggressive (riskier) moves, while low values indicate safer moves.

The algorithm works as follows:

1. **Fail-Safe Checks for Current Position:**
   - If any detective is within 2 fields of Mr. X, Decoy Movement is disabled.
   - If the next reveal turn is within 2 turns, Decoy Movement is disabled.
   - Compute `field_score` for the current position of Mr. X. If the score exceeds a threshold, execute a safe move (maximize distance from detectives) and skip Decoy Movement.
2. **Candidate Generation:** Generate all neighboring fields as candidate moves.
3. **Candidate Evaluation:**
   - For each candidate, compute `field_score` considering distance to detectives and escape routes.
   - Remove candidates that are too risky (score above a maximum allowed threshold).
4. **Move Selection:** Choose a candidate probabilistically using a roulette wheel based on `field_score`.
5. **Fallback:** If no candidates remain, select a defensive move maximizing distance from detectives.

2. Flowchart

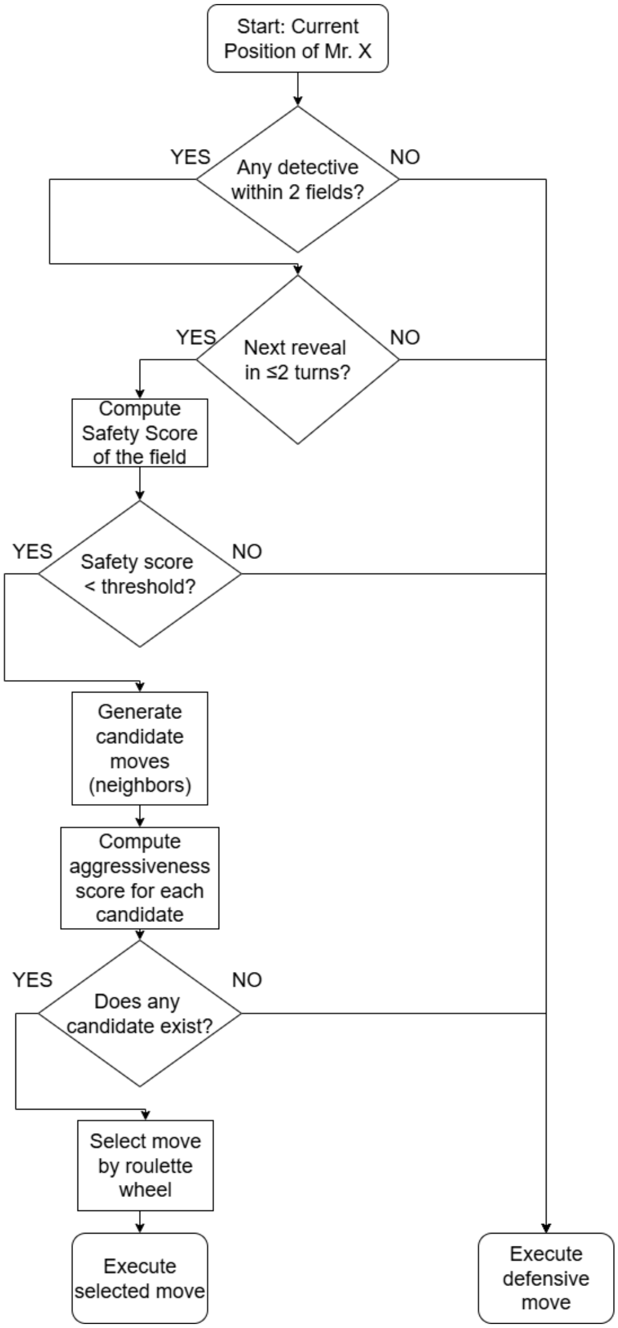
Start: Current Position of Mr. X
Any detective within 2 fields?
YES
NO
Next reveal in ≤2 turns?
YES
NO
Compute Safety Score of the field
Safety score < threshold?
YES
NO
Generate candidate moves (neighbors)
Compute aggressiveness score for each candidate
Does any candidate exist?
YES
NO
Select move by roulette wheel
Execute selected move
Execute defensive move

Figure 1: Decoy Movement Flowchart

3. Pseudocode

```
Algorithm 1: Decoy Movement Algorithm for Mr. X (with distance and reveal checks)

Input: posX, posDetectives, graph, turn, reveal_schedule, parameters candidate_limit, weights $w_r, w_e, w_o$,
       thresholds current_score_max, candidate_score_max
Output: Selected move for Mr. X
if any detective is within 2 fields of posX or next reveal in 2 turns then
    Execute defensive move (maximize distance from detectives);
    return
end
current_score ← compute_field_score(posX);
if current_score > current_score_max then
    Execute defensive move (maximize distance from detectives);
    return
end
candidates ← neighbors of posX;
foreach candidate p in candidates do
    $field\_score_p$ ← compute_field_score(p);
    if $field\_score_p$ > candidate_score_max then
        remove p from candidates;
    end
end
probabilities ← normalize(field_score values of remaining candidates);
if candidates not empty then
    Selected move ← roulette wheel selection using probabilities;
end
else
    Selected move ← defensive move (maximize distance);
end
return Selected move
```