

BLACK BOX

The Ultimate Game of Hide and Seek

CS154 - Abstractions and Paradigms for Programming

Group Members :

- Suseendran B (160050105)
 - Jagadeep Sai .N (160050104)
-

About the game :

Black Box is played on a two-dimensional grid. The object of the game is to discover the location of objects ("atoms") hidden within the grid, by the use of the minimum number of probes ("rays"). The scoring is done based on the probe results.

Link: [Rules of the Game](#)

The Objective :

Is to strategically acquire information in order to deduce the game boards hidden solution.

The player can be the **Hider** or the **Seeker**. If the player is the *Seeker* then he should find the solution by using probes for which the computer, the Hider, gives a result. When the player plays as the *Hider* the "AI" kicks in to 'seek' out the solution.

The problems aims at developing artificial intelligence systems that undertake approaches in order to determine which strategies and properties of the board yield more accurate and optimal solutions.

Idea of the Solution :

- When player is *Seeker* and computer is *Hider* :

The computer stores a grid with the hidden 'atoms'. When the player probes a ray through a particular numbered box, the computer follows an *algorithm* to find out the result of the probe. Finally evaluates the score based on the probed rays and the final guess of the hidden 'atoms'. It's pretty straightforward.

- When player is *Hider* and computer is *Seeker* :

We propose two algorithms for finding the solution :

- Probabilistic approach : (does not guarantee accuracy but is fast)

Each square must have a threshold value to qualify as a guess. It is initially assigned a value of 0. The square values are modified based on the results of the probes.

The values on the squares are actually a measure of their probability to be a solution. So once the value crosses a limit, it can be declared as a solution with high probability, but not with assurance. In this way, the first four squares to cross the limit are the guessed solution.

The moves are chosen randomly in order to normalize the scoring process for each square of the board.

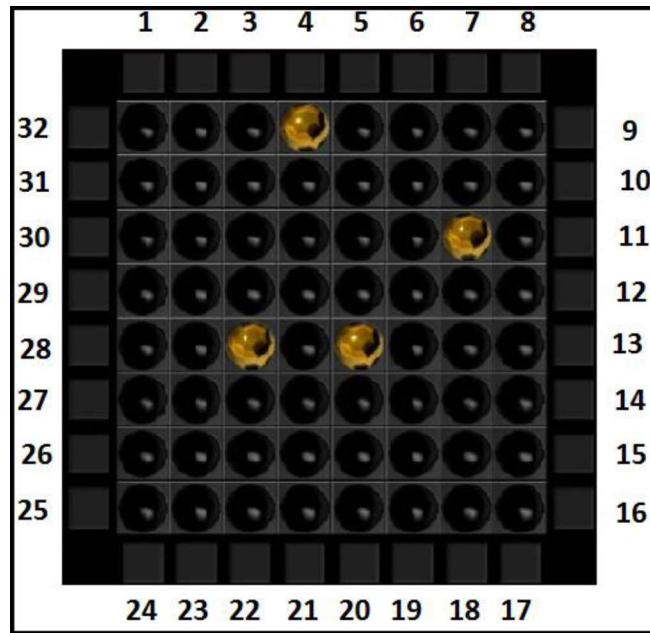
- Brute force with AI based move selection : (guarantees accuracy but is slow)

The computer has a list of all candidate solutions and it probes (plays its move), and records the result given by the player. The same probe is done on all the candidate solutions and contradictory candidates are removed. This is done until a final solution remains.

But playing random moves may consume more moves and may not give productive results. So the move selection is done based on an iterative filtering.

For each remaining move, we narrow/eliminate out candidate solutions as though some result were to have occurred and remember the number of remaining candidates. We do this for all result cases (a "hit", "reflection", or "detour"/"miss") and remember the greatest value for that particular move, it represents the worst case (where minimum number of candidates are eliminated) for that move. We then select the move which has the minimum of these values. This provides us the move which eliminates the most candidate solutions.

Sample Input and Output :



The results of all possible probes for the above board are given below

1	Deflection to 24	17	Deflection to 12
2	Deflection to 29	18	Hit
3	Reflection	19	Deflection to 14
4	Hit	20	Hit
5	Reflection	21	Reflection
6	Hit	22	Hit
7	Hit	23	Deflection to 27
8	Deflection to 10	24	Deflection to 1
9	Hit	25	Deflection to 16
10	Deflection to 8	26	Deflection to 15
11	Hit	27	Deflection to 23
12	Deflection to 17	28	Hit
13	Hit	29	Deflection to 2
14	Deflection to 19	30	Hit
15	Deflection to 26	31	Hit
16	Deflection to 25	32	Hit

- If the computer is the Hider, then the player will get the above results for the respective probes. He can deduce the solution with these results.
- If the computer is the Seeker ,then the player will be giving the above results according to the computer's probe. Then it uses the above mentioned algorithms to deduce the solution.