

# Report on Chain Reaction Game

Zahid Al Hasan, 2105087

## Introduction

I was told to build an instance of the game "**Chain Reaction**", where I used **python** and its famous game library **pygame**. I have kept 3 playing modes:

- **Human vs Human**
- **Human vs AI**
- **AI vs AI**

## Experimental Setup

The first mode is very casual, it does not need any AI involvement. So no heuristics or minimax search algorithm were needed. For the next two modes, I kept option for the following **5** heuristic evaluation functions:

1. **Orb difference heuristic** : Measures the number of red orbs and the number of blue orbs, and their difference is the heuristic value. Count of red orbs - count of blue orbs is the heuristic value for the red player, and the negative is the heuristic value for the blue player.
2. **Territory heuristic** : Measures the current color of all the cells, where cell color means the color of the orbs the cell is holding. If no orb is held by a cell, I used the term "null" as the cell color; and such cells had no impact in this heuristic evaluation. Actual heuristic value is found in the same way as before.
3. **Mobility heuristic** : This is even smarter. Finds the possible cells a particular colored orb can be kept in. Similar way was followed for finding the heuristic value.
4. **Critical mass proximity heuristic** : A cell that had only one orb less than its critical mass was supposed to be a critical cell; and was a threat to the other colored cell as it was about to burst. A good heuristic in action.
5. **A combined heuristic** : This is basically a weighted sum of the previous 4 heuristics.

And ultimately, a **Minimax Search Algorithm** that uses one of the heuristics to find the best possible next move for a player.

For testing purpose, I set the **time limit to 6 minutes**, set the **starting player as the red player**, made changes to the depth limits and the heuristics each time; and noted down the results. The results are shown in a table on the next section.

## Results

Below are the experimental data found from playing the game with different AI heuristics and different search depths:

Table 1: **Human vs AI**

Heuristic	Search depth	Elapsed time	Winner	Moves	Remarks
Orb_diff	4	315s	AI	62	sneaky, clever
Mobility	3	360s	none	74	sneaky
Territory	3	360s	none	82	sneaky
Combined	2	360s	none	82	little aggressive
Critical	3	274s	AI	62	unpredictable playing
Territory	2	360s	none	61	dind't play that well
Mobility	2	360s	none	76	started well but game not impressive
Orb_diff	3	166s	AI	62	quite aggressive and clever playing
Critical	4	234s	AI	42	unimpressive start, good playing overall
Combined	3	360s	none	94	sneaky

Table 2: **AI vs AI**

Heuristic		Search depth		Elapsed time	Winner	Moves	Remarks
Red AI	Blue AI	Red AI	Blue AI				
Orb_diff	Critical	3	3	303s	Red AI	113	red too aggressive, blue clever
Combined	Mobility	2	3	360s	none		both followed patterns, red dominated
Critical	Territory	3	4	360s	none	94	red dominated awesomely
Mobility	Combined	3	3	360s	none	102	red dominated powerfully, about to win
Territory	Combined	3	4	360s	none	89	red dominated
Combined	Critical	2	3	-	-	-	red dominated
Orb_diff	Territory	3	2	309s	Red AI	121	red dominated all over the game
Mobility	Orb_diff	2	2	-	-	-	blue dominated first, red made comeback

## Discussion

First of all, I did not try all the possible permutations for testing and so, cannot have a fully concrete comment on the performances of different heuristics. What I can only have is some idea about how they can possibly act.

An important observation from the Table-2 is that the starting player puts the red orb at the very beginning; and in all the cases, red dominates. I tried swapping the starting player and it was the blue player that dominated then. The reason is - the first player gives the best possible move and opponent can't surpass that move in order to reverse-dominate it with the help of any heuristic.

However, Based on the tables, some important characteristics are observed like - following patterns, dominating, playing aggressive etc. Following a particular pattern can be assumed to be a drawback for an AI agent in this case, because this implies that the agent is not willing to tackle any new scenario; rather it keeps doing something monotonously. Playing aggressively means that each time an AI agent fills a cell as much as it can and does nullify opponent neighbor cells - resulting in an almost-full board with only one-colored cell of the aggressive agent. And, unexpected performance implies that the AI places orbs at cells humans can barely guess, and one such single move can turn the game around at a certain state unexpectedly.

Finally, in a nutshell, we notice from the experiment that **AI agents performed well when the orb difference heuristic and the critical mass proximity heuristic was integrated**; while playing human vs AI, in all these cases the AI won amazingly. Similar observation goes for AI vs AI, but then the first move played even more vital role. The **territory heuristic couldn't show impressive performance, whereas the rest two heuristics were somewhat average**.