

Mondrian's Revenge: A No-Frills Multi-Stage Strategy for the Game of Rectangles

Group 1

[[Mark Ayzenshtat](#)]

[[Edan Harel](#)]

[[Adam Rosenzweig](#)]

Table of Contents

I. Introduction

- A. General Observations
- B. Chasers: What are They Good For?
- C. A Diamond in the Rough

II. Offensive Strategy

- A. What Would Mondrian Do?
- B. Filling in the Holes
 - 1. Spirals are Pretty
 - 2. Thinking Locally
- C. Refinements

III. Defensive Strategy

- A. Chaser Detection Strategies
 - 1. The Grudge-Holding Sociophobic
 - 2. The Proximity Matrix
 - 3. Minding One's Tail
- B. Chaser Neutralization Strategies

IV. Conclusion

I. Introduction

A. General Observations

The players submitted for last year's run of *Rectangles* were instructive in revealing several truths about the problem. Like every other group, we examined the behavior and performance of these existing players under various gameplay conditions, drawing conclusions that helped us identify areas with potential room for improvement and formulate an effective strategy.

First, we observed that there were usually two fairly distinct stages in the average game of Rectangles. In the *early game*, large swaths of territory were up for grabs by both players. During this stage, the object of the game appears to be simply to gobble up a large amount of real-estate as quickly as possible. In the *late game*, most of the board is claimed, and only small pockets of unfilled cells remain. During this stage of the game, a player's goal shifts to

identifying which unclaimed territory is most worth its time and claiming it.

With the exception of a game being played by multiple *Army*-type players, the transition between the early game and the late game seemed to occur rapidly. Thus, in the very beginning, our overall plan was to: 1) develop a winning early-game strategy and, separately, a winning late-game strategy; and, 2) identify when the transition between the early and late games occurred and switch from the first strategy to the second. A simple method for detecting the transition is to compare the amount of black squares on the board to a threshold value, based on the board size. Two-thirds or three-quarters seem to be appropriate bounds for the threshold, with the exact value being tunable. This comparison is most simply and efficiently done by getting the scores of all players and summing them up. It makes more sense to determine the change in game phase by the number of black squares than the number of non-white squares because squares that are painted a player's color can be overwritten and thus are still valid for scoring new rectangles. Except for brute force algorithms that manage late-game by sweeping the entire board, or ones that identify valid, "scoreable" rectangles to capture throughout the game, detecting the transition is critical for a strategy to perform well throughout the game. Of course, a good late-game strategy is also essential.

We also noticed that, in the early game, it was clearly more advantageous to draw certain kinds of rectangles than others. It is trivial to show that square-ish rectangles have a lower perimeter-to-area ratio than long, narrow rectangles and are therefore more optimal candidates for drawing. Additionally, we observed that there was an implicit risk/reward tradeoff for rectangle size. The larger an attempted rectangle is, the more points it is worth if it is completed, but the more likely it is to be disrupted by an opponent's robot. Similarly, smaller rectangles are safer but yield smaller score gains. It seemed as if in our player's interest to draw square-ish rectangles with areas that fell in a certain "sweet spot" range.

B. Chasers: What are They Good For?

"*Absolutely nothing!*" Well, that's clearly not accurate – chasers performed well in the tournament, but we argue that their success reveals not an insight into the rules of the game but a critical oversight in them. Last year's players incorporated some chasing into their overall strategy but it was always limited in scope; one or two robots would harass an opponent while the remaining robots would continue some form of construction.

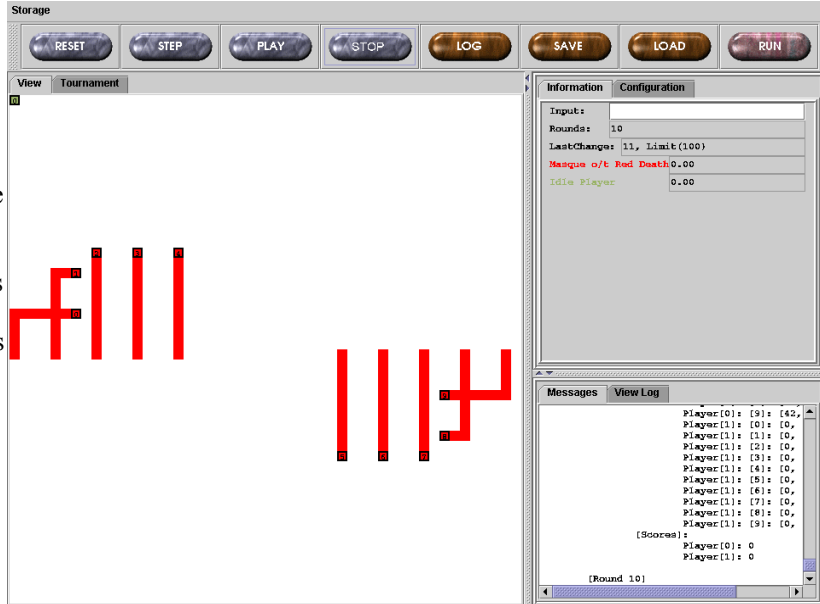
This year, at least two groups carried out *all-bot chasing* in two-player games. At some point in the game, each of their robots would snuggle up next to an enemy robot and mimic all of its moves. At the very least, this would prevent the opponent from making any further gains during the game; at best, it would allow the chasing player to parasitically claim its opponent's territory for itself. As witnessed by the tournament results, there are no effective counter-strategies to this behavior. Facing a minimally competent chaser in a two-player game, the best a player can do is to claim territory as rapidly as possible and hope that it is ahead in score by the time the chaser's robots catch up to its robots. This makes for very uninteresting games.

If a player following a simple strategy (e.g. at each turn, minimize the distance between one of your bots and an enemy bot) can defeat any opponent in most two-player games, it is certainly possible that the player's authors have surrounded the problem and created a novel, insightful approach. But it is vastly more likely that they are merely exploiting an oversight in the rules by the game's designers. Perhaps, the rules can be tweaked to only allow robots to "see" their opponents' robots within a certain radius of their own position instead of all of the robots on the game board. Or, alternately, chasers should be evaluated separately from other players in individual offensive and defensive tournaments. I do not mean to downplay the achievements of chaser groups so much as argue, again, that chasers' dominance makes for rather dull two-player games.

C. A Diamond in the Rough

Before we firmly settled on the Mondrian approach of our submitted player, we considered a few other early-game strategies. The discarded strategy that held the most promise was the *Diamond* builder.

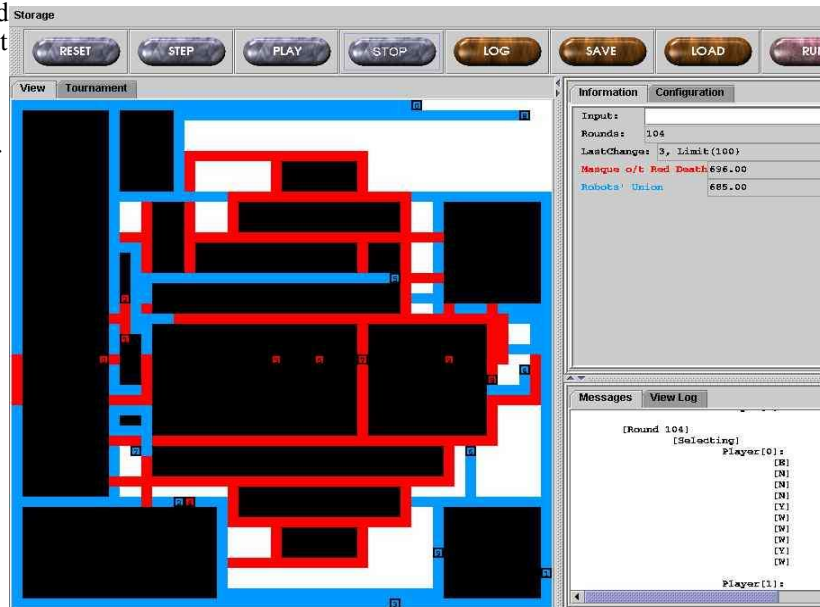
The Diamond opening move begins with all the robots situated horizontally along the middle of the board. The robots create rectangles as they move clockwise across the board, with the outermost robots making rectangles that are shorter in height and longer in width, and the center-most making



rectangles shorter in width and taller in height. The robots crisscross each other, working together in a manner similar to the *Army* robot or the *cavalry*, and allows them to create many smaller rectangles during their rotation across the board. This helps it to capture some points even if another player is trying to capture the same area.

The "diamond" shape of the pattern creates an implicit defense for the player as the jagged edge helps to prevent opponents from forming rectangles nearby. Given that it also tends to capture the center of the board - a key strategic location in most games - it does fairly well against grid

players that try to capture large sections of the board. However, the Diamond player was



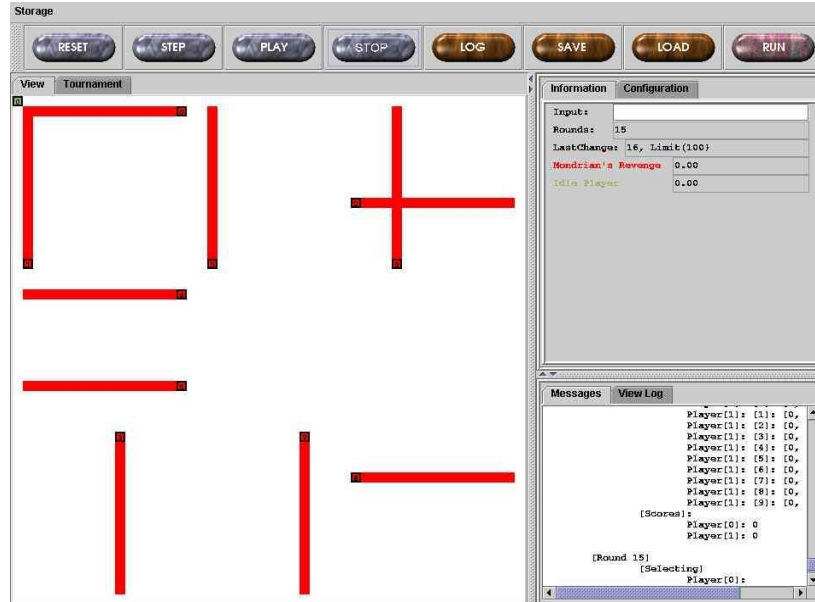
significantly weaker against players that concentrated on building out the corners of the board.

Overall, the performance of the Diamond player tended to be mediocre, with the player being ahead of the opponent during the beginning of the game a little more than half the time, but it lost ground by the time it finished the diamond and fell apart by the end-game. Some tweaking, and making the player more ambitious, as well as utilizing an effective end-game strategy would probably have made the player more competitive.

II. Offensive Strategy

A. What Would Mondrian Do?

Mondrian, the strategy that we developed for our submitted player, operates in several different stages. Initially, each of our robots is placed on the perimeter of the game board. Although this placement is relatively uniform, it maintains a certain degree of randomness that prevents it from falling prey to regional attacks. To elaborate, the robots are evenly



divided into two groups. Robots from one group will begin along either the top or bottom edge (equally likely to begin from one edge or the other) and cross the board vertically throughout the game. The other group will begin along either the left or right edge and cross the board horizontally throughout the game.

This placement has several positive properties that make *Mondrian* a highly effective early-game strategy. First, the strategy is clearly a simple one. Individual robots have no concept of rectangles, only of lines and direction. Since an equal number of robots move vertically as horizontally, rectangles tend to emerge over time. Second, on average, rectangles tend to form when the robots are halfway across the board. However, precisely when and where these rectangles form cannot be predicted due to the randomness of their initial placement. Thus, for yet another reason, the strategy is not susceptible to regional building. Finally, relatively large, square-shaped rectangles tend to emerge over the robots' first pass, regardless of board size.

After the robots have made one pass across the board, they will move along the edge to a new parallel, and then take off again across the board in the direction from which they originated. How they select along which parallel to cross the board is a very important part of the overall strategy. Initially, the robots would randomly select a parallel – clearly suboptimal behavior. An improvement that we implemented employed a technique that was a modification of an algorithm last year's *Men on a Mission* player used to select rectangle candidates. After the robots completed their initial traversal, they would randomly sample 6x6 regions of the game board, sorting them by number of unfilled squares. Then, a random random square in a high-scoring region would be selected and set as the target point. A robot would then move to a point parallel to that target point along the edge and set off for the opposite edge on a course that intersected the target point. The process would then repeat. This approach worked reasonably well at first and allowed us to keep our player “dumb,” lacking knowledge of rectangles or the presence of opponents' robots.

B. Filling in the Holes

Eventually, we agreed that the Mondrian strategy did not work well throughout the entire game. Although it was quite strong in the early game, it fell apart in the late game because the parallels that robots traversed would intersect mostly filled squares. Our competitive drive won out, and we traded algorithmic simplicity and elegance for the hybrid approach we initially thought we'd need to employ: separate strategies for the early and late games.

1. Spirals are Pretty

One late-game approach that we tried was to have each bot identify unfilled “holes” in the board using the familiar random sampling approach, move into the nearest hole, and begin to fill it in by drawing a spiral. This approach was simple in concept and implementation, but it was only effective roughly a third of the time. The reason for this was that bots expanded spirals evenly in all directions, while holes were not clearly equidimensional – some were tall and thin, others were short and fat.

2. Thinking Locally

Ultimately, we realized that other groups may have already solved the late-game problem, and it was in our interest (and theirs) to utilize their approach. We decided to use the *Localized Men on a Mission* strategy of Group 3. This strategy is a modification of last year's *Men on a Mission* strategy that selects potential rectangles adjacent to each robot's current position. Needless to say, this strategy was highly effective during the late game and, coupled with our original cross-hatch Mondrian strategy for the early game, made for a strong overall player.

C. Refinements

We made several refinements to our player before arriving at the current incarnation. For one, we played around with different spacing arrangements. Against certain opponents, the uniform placement of the robots around the perimeter of the board is too limiting and predictable. Whenever there is a large amount of overlap in the paths of our robots with those of opponents' robots, the strategy is not nearly as effective. To remedy this problem, we considered offsetting the starting location of each robot by ± 1 square in either relevant direction.

Also, we realized that if all of our player's robots traversed the board unopposed, the result would be a large filled area in the middle with individual colored spokes sticking out of it. The bots could potentially gain many additional points if they simply connected the spokes. Thus after each bot first arrives on the opposite side of the board, it takes N steps along the perimeter in either a clockwise or counterclockwise motion, where N is equal to the length of one side of the board. The aggregate result of this behavior is the player's bots connecting the spokes formed during their initial pass across the board, possibly yielding many additional points for the player.

Finally, since each bot would effectively circle a portion of the perimeter before continuing on to the late-game strategy, it became possible for us to position our bot one square in from the perimeter as shown in the above screenshot. This resulted in squares forming an average of two rounds earlier during the early game.

III. Defensive Strategy

A. Chaser Detection Strategies

Although we argued that all-bot chasing was a relatively uninteresting, though effective, strategy, our pride got the best of us and we devoted a significant amount of time trying to figure out how to effectively detect and counter such players.

1. The Grudge-Holding Sociophobic

Our first stab at detecting chasers was simple but surprisingly effective, and was appropriated and modified by at least two other groups. Basically, each bot would keep track of the enemy bot that came nearest to it during the last turn. If an enemy bot, E , was closest to the same friendly bot, F , for T turns, and E was currently within D squares of F , we would assume that E was chasing F and take appropriate corrective action (T and D are both tunable positive integers).

This approach worked reasonably well, but it had several negative consequences. For one thing, it did not take motion of enemy bots into account. Our player's bots would stop well short of the idle player's bots in square (0, 0), even though the idle player was not moving let alone chasing. Also, this strategy could be defeated by chasers who backed off temporarily once they realized that the "chase" was onto them.

2. The Proximity Matrix

Vlad from Group 3 created and I helped develop and refine an approach to chase detection that should have worked extremely well for two-player games if only we could fine-tune our metric. Essentially, we store a 2-dimensional "proximity matrix". Each row represents one of our bots, and each column represents one of the opponent's bots. We initialize each cell to a value of 1. Whenever a enemy bot moves closer to one of the player's bots, we increase (say, double) the value at their row/column intersection. Similarly, if they get farther apart, we decrease (say, halve) this value. The logic behind this is that, if the opponent is an all-bot chaser, one of the values in each row will be much higher than all others in that row. We can then take the determinant of this matrix or use another statistical approach to determine when this threshold is exceeded.

In practice, we were never able to get this approach to work. The metric we used to alter values in each cell in the matrix was never quite adequate and always yielded a player that was either too paranoid or not paranoid enough. If this problem is reassigned in a subsequent year, this would be an interesting idea for another group to develop.

3. Minding One's Tail

The final chase detection strategy that we settled on was also fairly simple but significantly more effective than the first strategy. Each bot would keep an ordered history of the cells that it has visited – a trail, if you will. Each round, it would measure the size of its trail going backwards, stopping when a square not of the player's color was encountered. If the size of the trail remained the same for N rounds, the bot assumed it was being chased. Bots employing this strategy rarely stop when near enemy bots that are, in fact, not chasing them, and, with little modification, are also not vulnerable to enemy bots that back off temporarily in order to thwart proximity-based detection approaches.

B. Chaser Neutralization Strategies

The simplest thing for a bot to do once it realizes that it's being chased is to stop moving altogether. Frequently, this is just as effective as any more involved strategy, but rarely is it effective enough. For reasons already discussed, there is little a player can do to thwart an all-bot chaser in a two-player game. The only thing to do is to build up a lead immediately and then recognize that you are being chased as early as possible.

If this defect in the game were corrected, other action besides stopping may prove to be effective. Depending on the chaser, it may be possible to redirect them (either to another player, or to direct multiple chasing bots to one chased bot) via intersecting robots' paths with each other. More complex strategies to acquire small rectangles even while being perfectly chased seem to be far too expensive to code and to spend time on for the benefit that would accrue from them.

IV. Conclusion

Throughout the tournament, our player regularly scored as first or second highest ranked non-chaser player, with which we are wholly satisfied. Our player makes much to most of its gains using the early game strategy, which is, of course, not revealed by a tournament that only considers final rank placement. Additionally, the *Mondrian* strategy is completely agnostic that it is playing a game with the object of drawing rectangles. Our strategy is relatively simple but demonstratively powerful, and can be made even more effective by considering alternate end-game strategies or tweaking the game rules to account for the dominance of all-bot chasers.