Let's Get Together
Group 6 - "I'm gonna catch you!"
Group Members: John Cieslewicz, Miguel Maldonaldo, and Sowmya Viswanath

1 Introduction

The goal of this project was cooperation among autonomous entities, or *Friends*, to discover and exchange information. The friends were designed to cooperate in two environments: environments where all friends were of the same type and environments where the friends from different groups had to cooperate. After first creating a single player friend that relied on specific behavior from the other friends in a game, our group changed out implementation to a friend that searches for other friends based on the trails left on the board. This strategy offers good versatility and cooperation in the face of other friends' unknown behaviors. This paper will discuss the evolution of our player, detailing both the single player and chaser versions. We will also present the main strategies discussed in class and explore the advantages and disadvantages of our player with respect to these other strategies. We will conclude with an analysis of the tournament results and a brief description of each group member's contributions to the project.

2 Game Setup

Each friend begins the game with a distinct piece of information and the game terminates when on friend possesses all of the information. Sharing of information occurs when two friends occupy the same location. Sharing is symmetric--two friends who share will each have the same information following the share. The environment of this game is a two dimensional board. No friend knows its absolute position on the board, thought he dimensions of the board are know. The board also wraps horizontally and vertically. On each turn every friend may decide to move in nine different ways: horizontally, vertically, diagonally, or stay put. As friends move around the board they leave behind trails in each cell they visit, these trails represent the total information that has passed through a given location.

3 The First Generation- Horizontal and Vertical

Over the period of the first two discussion classes for this project, many groups, including ours, independently developed a single player strategy that took advantage of the board geometry. By moving only horizontally or vertically and behaving in a deterministic way, friends could efficiently locate each other and exchange information, even on a large board. These players were referred to throughout the class as "Vertical and Horizontal" players. This section will describe our player's implementation and the shortcomings of this player that caused us to abandon the idea and move on to the *chaser* strategy. We also briefly present refinements to this strategy that we did not pursue.

3.1 Group Six's Vertical and Horizontal Player

Our vertical and horizontal players moved only horizontally or vertically. Vertically moving players continuously moved north. Horizontal players moved horizontally until they encountered a cell whose trail contents demonstrated an information gain due to a vertical player. We define information gain the cardinality of the set difference between a player's known information and the information contained in a cell's trace information. To determine if the information gain in a cell is due to a vertical player, the horizontal player would also have to find evidence of information gain due to the same pieces of information in the cells above and below its position. Once a cell with information gain is encountered the player does not move until the information gain is eliminated, that is, the vertical player returns and shares its information with the horizontal player and thus there is no more information to be gained at this location. The horizontal player then continues in its original direction looking for another cell with information gain. In this manner, all information will be shared among the players as long as one important condition is met: there must be at least one vertical and at least one horizontal player. At creation we assigned each player to be horizontal or vertical with probability .5. Therefore, the chance that all of our players moved in the same direction in a game with *n* players is given by:

$$P(\textit{same direction}) = (1/2)^n$$

Unfortunately, this probability is not small for games with few players and is one of the limitations of the horizontal and vertical strategy, as implemented by our group.

Analyzing this shortcoming made our group realize other failures of this strategy in terms of a multi-player game. One obvious solution to this problem is for each player to independently detect that all the players are moving in the same direction and then take action to fix the situation. The amount of time required to realize this condition is not trivial.

For instance, in a two-player game on a square board, the threshold would be $h+w$ turns, where h and w are the height and width of the board, respectively. This is derived from the worst-case scenario in which the horizontal player begins in the vertical player's column, directly below the vertical player. It will thus take one full rotation to encounter the vertical player's path, just missing the vertical player, which then takes another revolution to reach the halted horizontal player. Expand this scenario on a square board to w+1 players in which only one player moves horizontally and each vertical player occupies one of the columns and the threshold becomes h*w. This is because, in the worst case scenario, the vertical players are arranged such that the horizontal player must wait a full vertical revolution at every column it visits. Note that the threshold is this long for all players since the final vertical player has no knowledge of a horizontal player until the number of rounds equal the threshold, at which point the game is won.

Once the threshold situation is identified, the player faces another challenge: what to do? In a single player game, one solution may be to randomly choose a horizontal or vertical orientation again, but this strategy suffers from the fact that board already has trails on it, complicating decision-making. More importantly, for a player designed to also function in a multiplayer environment, reaching the threshold may represent a failure in a single player environment or the fact that the environment is actually a multiplayer one in which this strategy will not work. In a multiplayer environment, deciding to retry the single player strategy could be detrimental to cooperating with other players. Because the detection threshold is so high, a vertical or horizontal player may not become aware of the multiplayer environment for many rounds, adversely affecting other types of players' ability to interact with it. For this reason, we decided that the horizontal and vertical player was ill suited for the multiplayer configurations discussed in class. Instead we decided to implement a more flexible player that could adapt and work well with other strategies as well as succeed in single player environments. Section 4 presents this new player.

3.2 Other Ideas

There were a number of strategies discussed by other groups in class that we found worthy and fascinating, which we did not pursue. This section presents a brief overview of those ideas. One idea that had particular merit in a single player game was an enhanced horizontal and vertical player. The idea was to have every player make one horizontal and one vertical rotation. The board would now contain a grid of trails that can be used to find the exact location of any player. Coordinating the movement of the players to ensure that they exchange information remained a challenge for the group implementing this player through the last discussion class. Further, this idea faces the same problem in detecting a multiplayer environment that other vertical and horizontal player faced. Another important idea brought up in class was the idea of "bees and flowers." The idea is that some players (the bees) seek out other players, while some players remain in a local area (the flowers) hoping to be found. The reasoning behind this idea is that if all of the players are moving around the board they may miss each other, whereas a flower does not move much so it can be found by a "bee" and act as an information collector, sharing information with each bee that visits it. Our final chaser player is definitely a "bee" is this classification scheme. The bees and flowers strategy had some problems in special situations where all of the organisms were bees or flowers in a single player game.

We experimented briefly with a player that scanned each row in succession, either moving right for one board width and then descending one row or moving left for one board width and then ascending one row. This player always succeeded in solving the single player game when there was at least one player moving in each direction. The time to solve the game , however, was higher than other vertical and horizontal players in a single player environment. In a multiplayer environment the player did not perform well since it ignored players not directly in front of its path and covered the board with its trail, which made it difficult for chasing players to find it.

In the end we stuck with our chaser strategy because it seemed to offer a robust solution in both single and multiplayer environments without having to detect which environment was present or dramatically change its strategy.

3.3 Standards and Deadlocks

One of the interesting discussions during class was the creation of class “standards”. Since the players would have to interact, it was argued that there should be a standard to control that interaction. The discussion first centered on ways to standardize the horizontal and vertical player idea so that different groups’ versions could interact successfully. The class quickly abandoned this idea since a mistake by just one group in implementing such a standard could adversely affect all of the other players in a multiplayer environment.

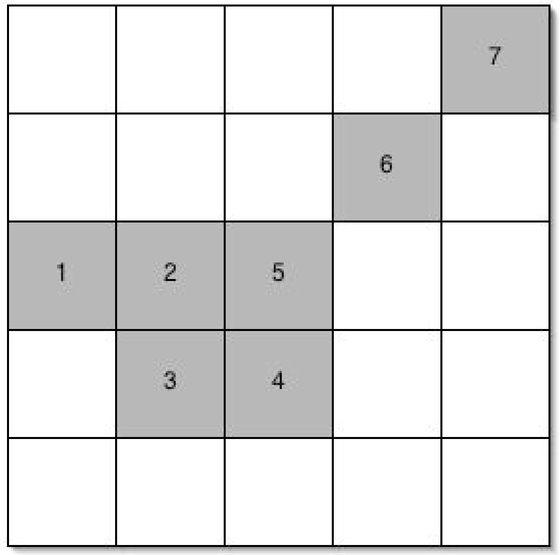
The standards discussion continued with respect to negotiating communication between two players. Because all players move simultaneous, two neighboring players hoping to exchange information that move to their neighbor’s square will miss each other and fail to exchange information. Therefore, it is important to have a scheme for deciding which player should move as well as a way to guarantee that a player does not deadlock when the other player does not behave as expected. One of the best solutions for deciding when to move was suggested by James who divided the possible moves into two L-shaped regions. Depending on which region the neighbor was in the player either moved or remained still. This worked well since if Player B is in the first L-shaped region from Player A’s perspective, then Player A is in the second L-shaped perspective from the perspective of Player B; therefore, the two players choose different behaviors and successfully share information. A problem still exists in that not all players used this scheme or implemented it in the same way. Some groups discussed using probabilistic decisions to decide whether to move to a friend or stay put. In the end it was evident to all of the groups that it is important to make sure players were deadlock free so progress would always be made during a game, since standards could not be relied on in a multiplayer game.

4 Anatomy of a Chaser

The idea was a simple one, to use the other player trail to attempt to follow their path and ultimately catch them. Our first player implemented this idea. It would randomly walk the board looking for the trail of an unknown friend and start following this path until an unknown friend was encountered. The decision process for our player started by evaluating all surrounding cells and if any cell was occupied by an unknown player we meet using the “classroom convention”. The convention was defined to move the player whose meeting direction was less than the direction from the other player to him.

If no new information gains on the neighboring cells the player would analyze the trail of all the neighboring cells and choose a path and player to follow. The player follows this friend until he meets with this player or any other unknown player. If our player is unable to find a path to follow, he would move with 2 in 3 probabilities in a random direction. We some time stopped moving in case another player was following us we would meet up with him.

Our initial test run showed us that our player will sometimes work but most likely it ended up getting stuck cycling on the other player path.



This scenario lends it self for our player to cycle between cells 2, 3, 4, 5. The only way or player can successfully solve this puzzle is by choosing in cell 5 to follow towards cell 6. But there are only 1 in 4 changes that our player would select the correct cell, since the board lacks historical information.

4.1 A Better Chaser

To improve our player we decided to bind the number of rounds a player would chase another one. For our initials test we hard coded this value at 10. While this did not prevent the cycling on a path it did limit it to a maximum number of rounds, forcing our player to break out of the cycle. We limited our player to not chase the same player in consecutive chases, but we can meet players again after 10 rounds, this value should be the same as the number of rounds to chase a player. These changes significantly improved our player in small board scenarios, but it did not scale well to larger boards.

The two main problems for scaling our model was the hard coded value for number of rounds to chase our player was too small to consistently find player in large spaces. Also the random movement usually confined our player to a limited board area and did not attempt to maximize board coverage nor information gain when moving.

4.*2 A Smarter, Better Chaser*

To address these two main limitations we decided to make our player smarter as well a better. To improve its chasing we make the number of rounds to chase depend on the board size, the number of player, and the total number of rounds the chase would last. To make our player smarter we created our own board history. Since we wanted to keep the player decision time reasonable, we choose to only store our own movement and not all available information. We created a grid representation of the board taking our starting position as cardinal (0, 0). Every move we would update the information related to the cell we where currently occupying. By storing the round number that we visited each cell we are able to leverage that information when trying to move or chase another player.

The information allowed our player to rapidly identify and move away from path cycles and to maximize board coverage by moving toward unknown areas of the board. Also to improve the changes of information gain in our movement, our player chooses only from the four diagonal directions when moving randomly. These changes made our player much more consistent in finding other players but it still lacked sufficient board coverage since the choosing of a direction was entirely random.

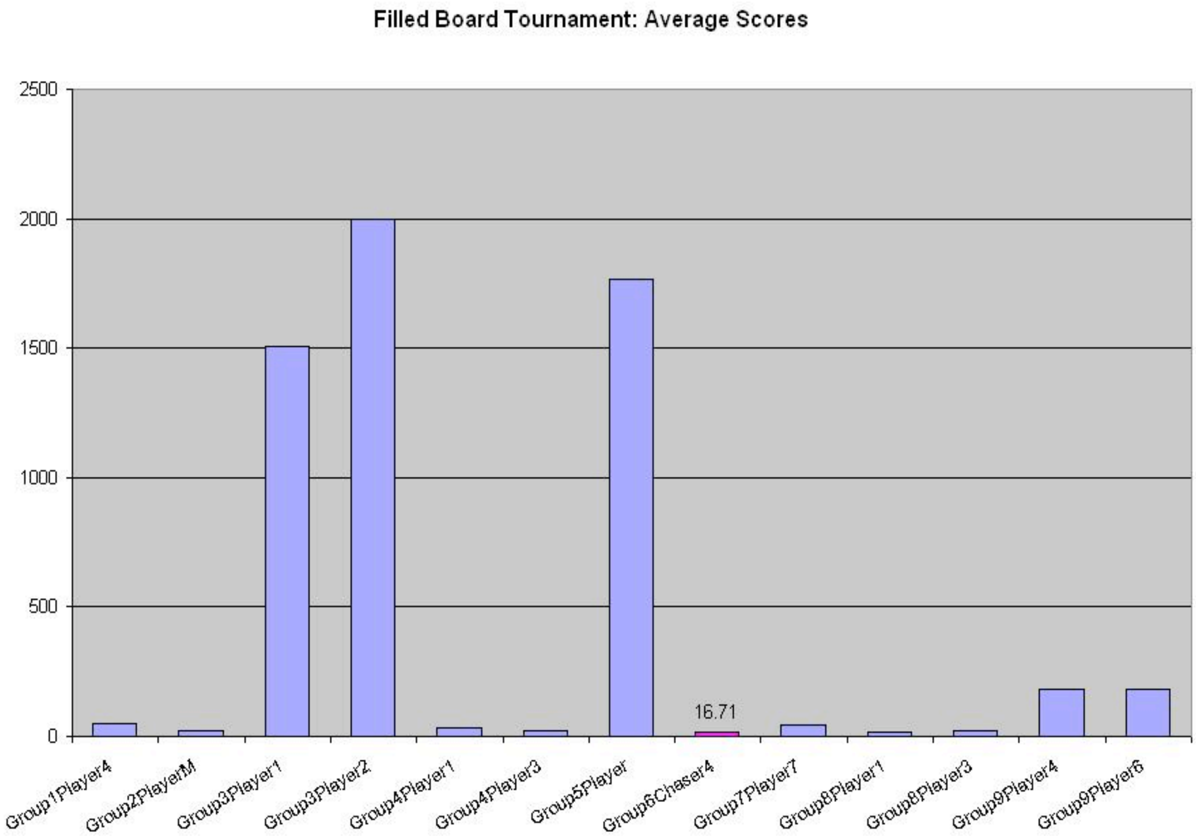
4.3 The Final Chaser

To improve our final player we focused on making improvements on its board coverage. To address this problem when our player decided to move instead of randomly choosing a diagonal direction every time, our player would pick a direction and move in this direction until a player or its trail is found. This was also bounded by the calculated bound for chasing. This final change allowed our player to converge on large and sparse scenarios. Moving in a consistent direction made it easier for our player to explore larger board areas while searching for unknown player trials.

5 Tournament Analysis

The results of the tournament varied with the size of the board as well as the mode it was being played in. The various cases are discussed below.

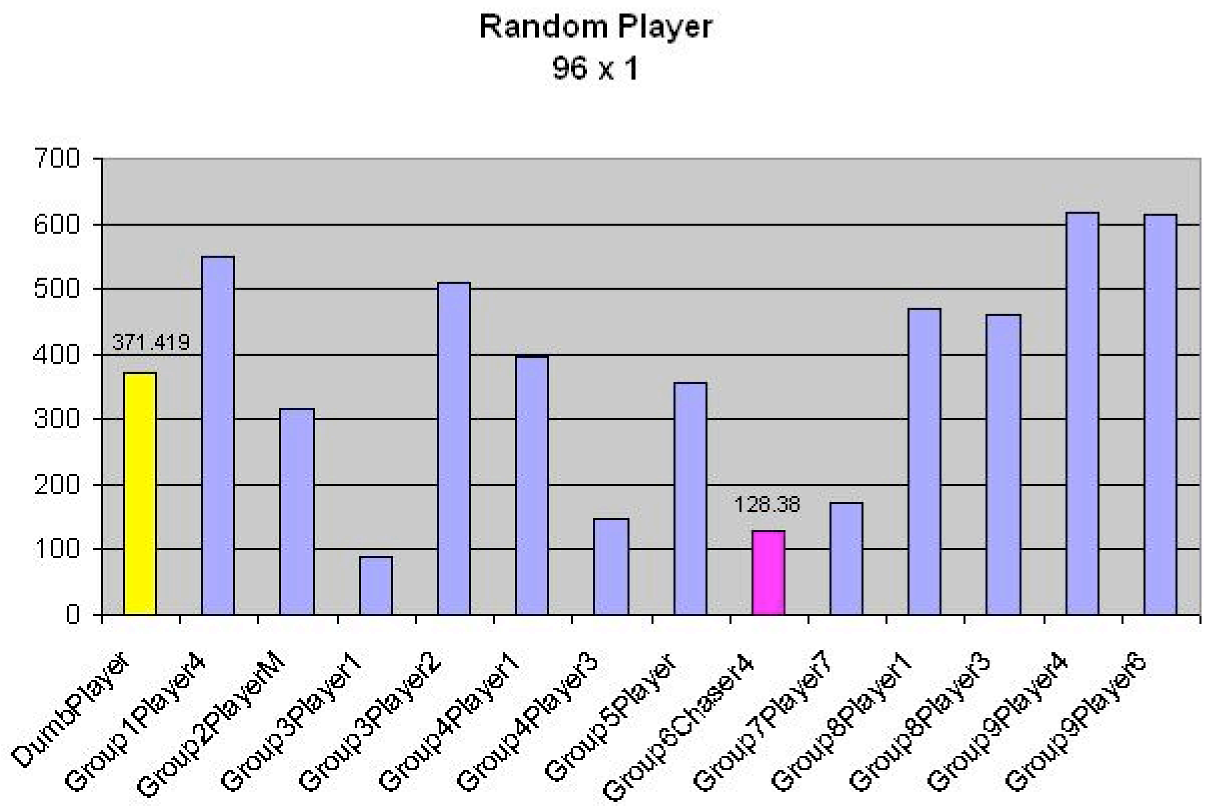
5.1 Filled Board



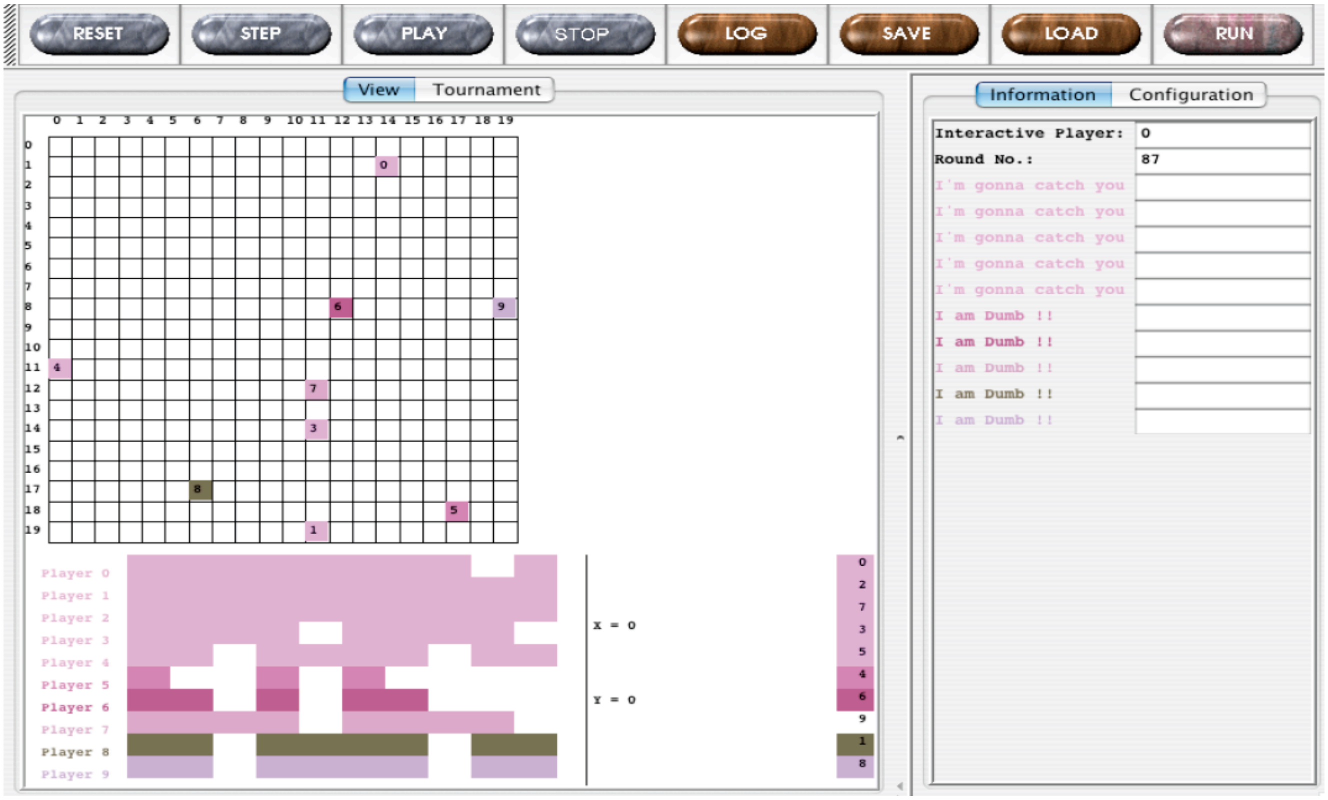
On a filled board (L15, W11, P165), our player **ranked first** based on average score of 16.71 after 2000 rounds. The strategy that we employed after the discussion in the last class was to implement a version of the dumb player that moved randomly but converged in very few rounds. We used the size of the board and the number of players to determine whether the player had to be a chaser or follow the random strategy. A coverage fraction of 40% was multiplied with the area of the board and compared with the number of players. The player moves into the random mode if the players cover more than 40% of the board. We chose this fraction after running some tests and observed that at this point, the number of rounds for convergence using the chaser strategy was very close to that of the DumbPlayer.

5.2 With the Random Player

Our player has the about the same or lesser average score as the DumbPlayer. However in certain conditions such as the (L2, W75) and the (L92, W1) board configurations, our player helped the Random Player to converge in a significantly lesser number of rounds. The (L92, W1) scenario is illustrated here. This leads us to conclude that while the player is a chaser and hence moves about a lot, it moves in an intelligent manner trying to collect as much information as possible which allows it to help converge with the DumbPlayer (which moves in an uneducated manner) faster.

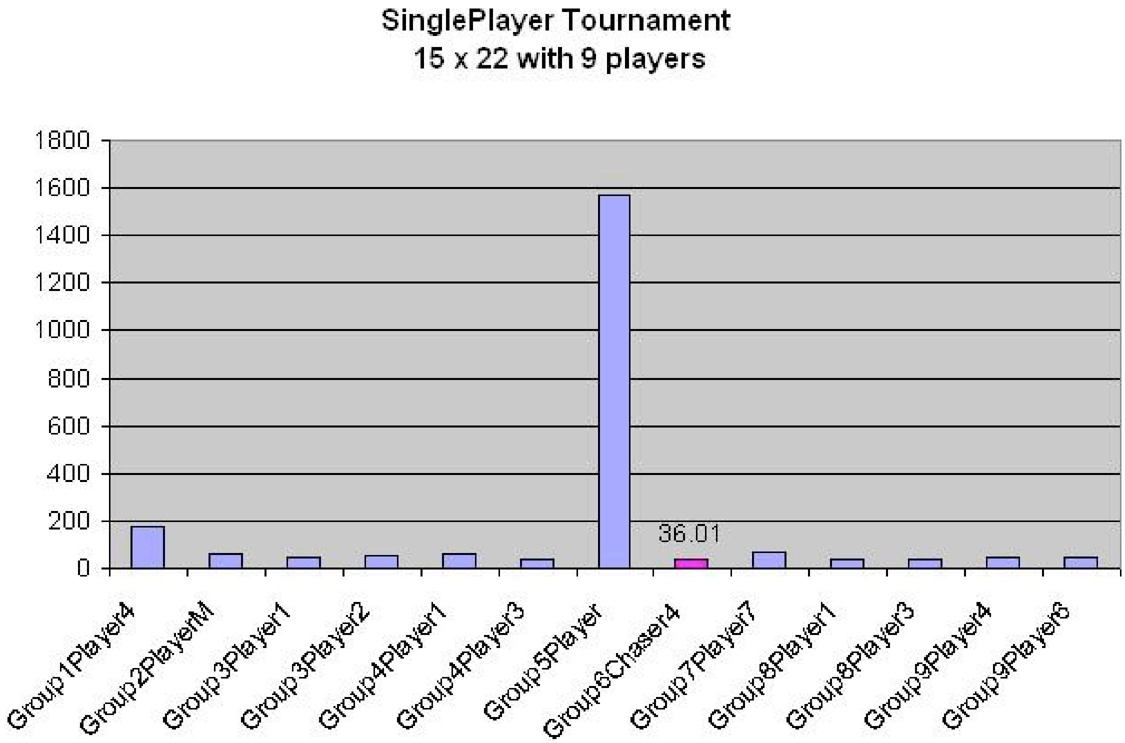


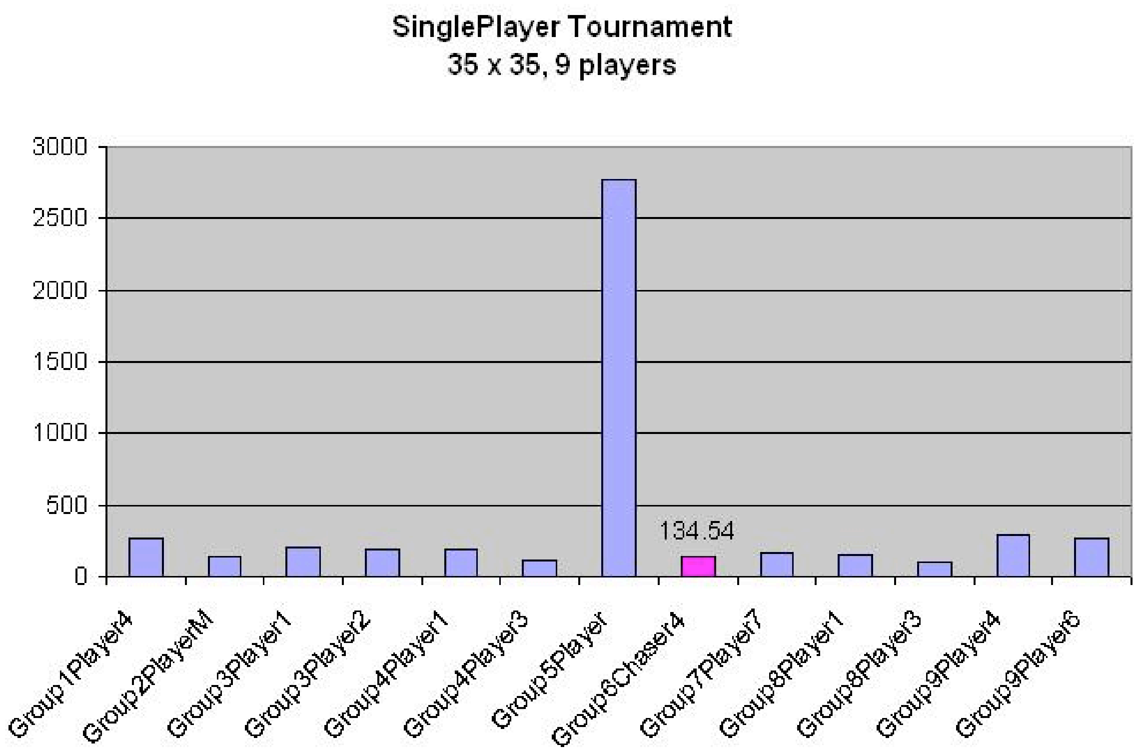
We see in that the number of rounds is reduced to almost a third because of the informed movement of our player. Another interesting feature that we noticed as a result of running tournaments with instances of our player and the DumbPlayer is the amount of information gained by our players versus the Dumb ones. The tournament consists of 10 players with 5 players of each type. The convergence occurs at round number 87, with our player ending up with all the information. What is interesting is that at the end of the game, most of our players have most of the information and are missing only 5 units of information in all. On the other hand, the DumbPlayers possess only a fraction of the entire information, and miss out on 18 units of information in all. Thus the chasers follow the trail of higher information gain and contribute to the early convergence of the game.

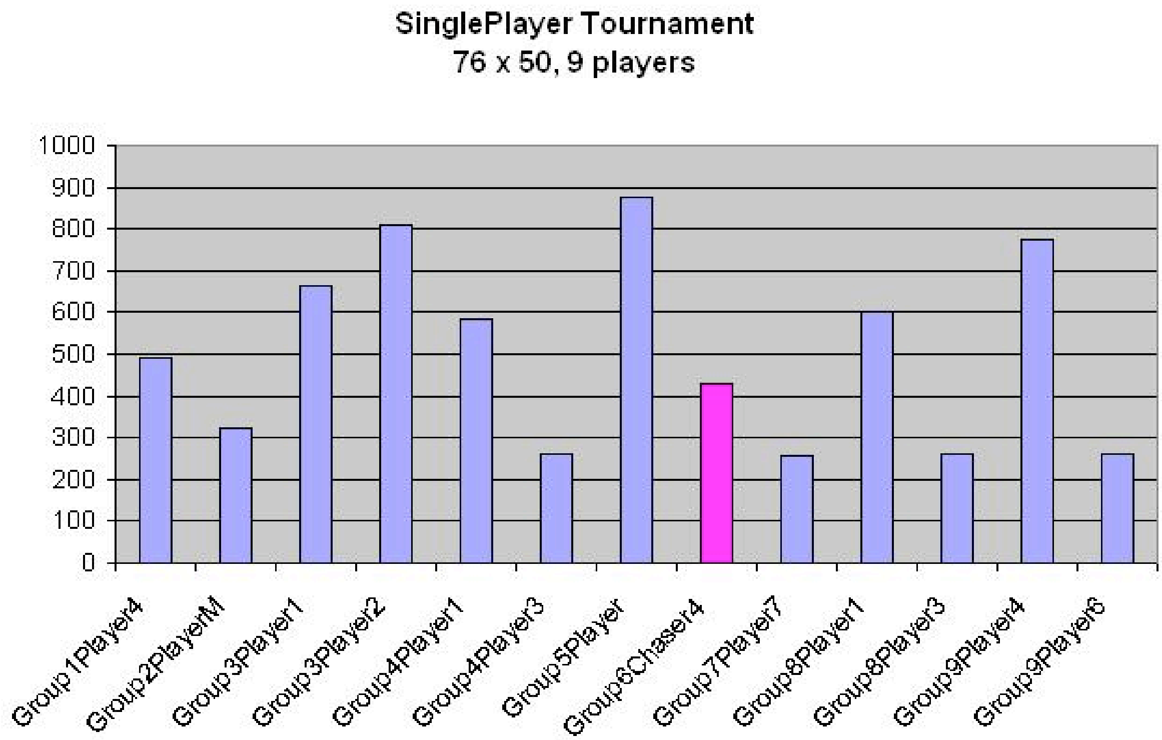


5.3 Single Player Tournaments

Our player performs very well in the single player mode. It excels in small board scenarios and we consistently **rank the first** in terms of average score. The following figures illustrate the performance of our player with the increase in board size. The player ranks first in the small board case, but the performance drops with increase in board size.


SinglePlayer Tournament
15 x 22 with 9 players
1800
1600
1400
1200
1000
800
600
400
200
0
36.01
Group1Player4
Group2PlayerM
Group3Player1
Group3Player2
Group4Player1
Group4Player3
Group5Player
Group6Chaser4
Group7Player7
Group8Player1
Group8Player3
Group9Player4
Group9Player6


SinglePlayer Tournament
35 x 35, 9 players
3000
2500
2000
1500
1000
500
0
134.54
Group1Player4
Group2PlayerM
Group3Player1
Group3Player2
Group4Player1
Group4Player3
Group5Player
Group6Chaser4
Group7Player7
Group8Player1
Group8Player3
Group9Player4
Group9Player6



The improvement in performance in the case of smaller boards can be attributed to the amount of area of the board that is 'dirty' at any given time. In the case of larger sizes, the board gets dirty after a while with the movement of all the players, and our player has to travel a lot more to get the same amount of information. On the other hand, when the area is smaller, the player chases the right trails and gets the information much faster. *Thus our player is optimized to perform best in the case of smaller boards.*

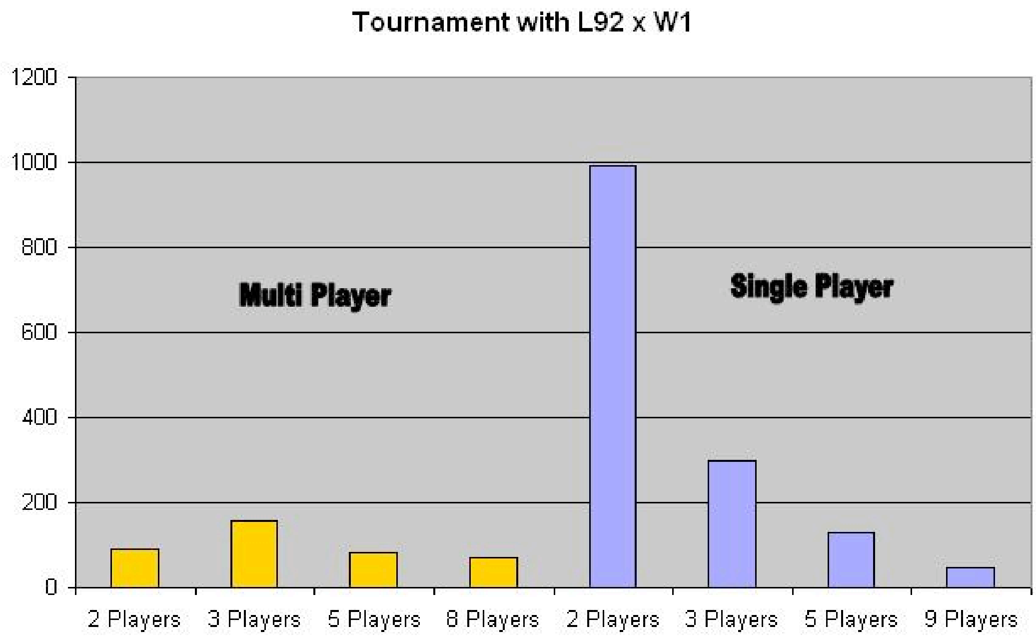
5.4 Multiplayer Tournaments

The performance of our player in the multiplayer mode does not match its performance in the single player tournaments. The tournament players included a lot of different strategies such as the flowers and bees, the chasers and the horizontal and vertical players. While the chaser strategy was implemented with the intention of gaining as much information as possible and then moving around to share this information with the other players, the strategies that combine the horizontal/vertical movement along with the flowers/bees converge faster as there is less random movement involved. The following figure illustrates our player's performance in the large board scenario with 8 players at a time. Our player ranks about midway in terms of average score.

5.5 Special Board Configurations

On a 2 x N board, where N was 75, our player performed very well when the number of players on the board was large. This was observed to be the case in both single and multiplayer tournaments as shown in the next figure. In particular, we **ranked second** in the single player mode with 8 players on the board. We see a larger convergence when there are just 2 players since they end up chasing each other for a long time before they finally exchange information.

On a N x 1 board, where N was 92, our player showed similar traits by converging faster when there were more number of players on the board. This was again observed for both the single and the multiplayer tournaments. However, an interesting feature is that we **ranked first** in the multiplayer mode when the number of players was just 2. This occurs because we are no longer chasing our own player, but someone else whose strategy complements ours.



5.6 Contribution of our friend to the rest of the players

The following method was used to determine the contribution of our player to the entire set of friends. In the multiplayer scenario (L35, W35, 8 players), the average of the all the average scores of the players in tournaments that we were not included in was computed to be 128. Similarly the average of the all the average scores of the players in tournaments that our friend participated in was calculated to be 122. Thus we see that our friend is a positive influence on the group and reduces the number of rounds to converge, though by a small number.

7 Individual Contributions

Sowmya was involved in the team discussions and generation of ideas for the different strategies and players developed. She was in charge of the tournament breakdown and analysis and contributed to the report and presentation. John built the AbstractFriend class that formed the basis of all of our players. He also developed the horizontal the vertical player and contributed the analysis of that player. Miguel is responsible for the creation and refinement of our final player, the *chaser*; he also contributed the description and analysis of that player.