

Reconnaissance Blind Chess Assignment

Team Members:

- Jason Wille (1352200)
- Kaylyn Karuppen (2465081)
- Reece Lazarus (2345362)

Approach used for Improved Agent:

The approach used in order to improve the performance of the **RandomSensing** agent was implemented in the class **ImprovedAgent**. This class is very similar to the **RandomSensing** agent but improves the way that the agent implements the function **choose_sense**.

The approach had 4 different cases (the first case to catch returned a value):

- 1) If the opponent attacked one of our squares. Sense that square. This part of the approach was inspired by the TroutBot.
- 2) The second part of the approach would create a piece map which stored an array for each square on the chess board. The arrays consisted of what we thought the possible pieces on that block would be. We would then create sets from these arrays. This would represent the number of unique pieces we believe could be at that square. We then ordered the squares by the corresponding set length and returned the square with the highest set length provided the set length wasn't 1 and that the length of the array didn't equal the total number of possible states (this would be when we are certain about what is at that block). We also made sure to move squares away from edges to not lose out on any learning.
- 3) If the above two cases didn't return a sense choice. We would then check what move we were going to make and then just sense at that block.
- 4) If we were going to make a none move, then we randomly chose a sense block that wasn't on the edges of the board.

Results for Round Robin Tournaments:

| RandomSensing vs Baseline Agents | | | |
|----------------------------------|-----------------|---------------|--------------|
| Opponent | Opponent Colour | Winner Colour | Win Reason |
| TroutBot | White | Black | King Capture |
| TroutBot | White | White | King Capture |
| TroutBot | Black | White | King Capture |
| TroutBot | Black | White | King Capture |
| RandomBot | White | Black | King Capture |
| RandomBot | White | Black | King Capture |

| | | | |
|-----------|-------|-------|--------------|
| RandomBot | Black | White | King Capture |
| RandomBot | Black | White | King Capture |

The table above shows results for a round robin tournament where **RandomSensing** is used to play against the two baseline agents **TroutBot** and **RandomBot**. The **RandomSensing** bot played each agent four times, in which two of the games it played as white and two of the games it played as black. The results for this tournament can be found in the directory [round_robin_2_random_sensing](#).

| RandomSensing vs ImprovedAgent | | | |
|--------------------------------|---------------|---------------|--------------|
| RandomSensing | ImprovedAgent | Winner Colour | Win Reason |
| White | Black | Black | King Capture |
| White | Black | Black | King Capture |
| White | Black | Black | King Capture |
| White | Black | White | King Capture |
| White | Black | White | King Capture |
| Black | White | Black | King Capture |
| Black | White | Black | King Capture |
| Black | White | White | King Capture |
| Black | White | White | King Capture |
| Black | White | White | King Capture |

The table above shows results for a round robin tournament where **RandomSensing** is used to play against the **ImprovedAgent**. The bots played each other 10 times, 5 as white and 5 as black. The **ImprovedAgent** won 60% of the time playing as both black and white. This is not the boost in performance that we would have liked to have seen but at least it is beating the **RandomSensing** agent more than 50% of the time. We would need to run a larger round robin tournament to test if these results hold. The results for these games can be found in the directory [round_robin_3_random_sensing_improved_agent](#).

| ImprovedAgent Round Robin | | | |
|---------------------------|-----------------|---------------|--------------|
| Opponent | Opponent Colour | Winner Colour | Win Reason |
| RandomSensing | White | White | King Capture |
| RandomSensing | White | White | King Capture |

| | | | |
|---------------|-------|-------|--------------|
| RandomSensing | Black | White | King Capture |
| RandomSensing | Black | Black | King Capture |
| TroutBot | White | Black | King Capture |
| TroutBot | White | Black | King Capture |
| TroutBot | Black | White | King Capture |
| TroutBot | Black | Black | King Capture |
| RandomBot | White | Black | King Capture |
| RandomBot | White | Black | King Capture |
| RandomBot | Black | White | King Capture |
| RandomBot | Black | White | King Capture |

The table above shows results for a round robin tournament where **ImprovedAgent** played 4 games against each **TroutBot**, **RandomBot** and **RandomSensing**. **ImprovedAgent** played half its games as white against each bot and the other half as black. The results for these games can be found in the directory **round_robin_4_improved_agent_all**.

Conclusion:

The results that we observed showed that our **RandomSensing** and **ImprovedAgent** both comfortably beat the **TroutBot** and the **RandomBot**. The improvement that was made on the **RandomSensing** agent to create the **ImprovedAgent** was not that substantial. We also could have analysed the performance in more detail by running larger round robin tournaments. There are many more advanced techniques that can be used to **choose_sense** that could be explored in more detail in order to improve our results for the **ImprovedAgent**.