

Partner 1 name:

Ching Zheng Ing

Partner 2 name:

Tatsuhiko Araki

## Deterministic Simplified Backgammon Agent

Tatsuhiko started first, writing the static evaluation function to count pips. He tried writing the minimax but was unsuccessful initially.

After discussion and reading through the code documentation together, Zheng Ing wrote the minimax code and debugged static evaluation function for count pips. However, Zheng Ing wasn't sure how to adapt the code for Alpha Beta Pruning, especially regarding where to store alpha and beta variables. Tatsuhiko researched online and found multiple online resources to help and showed them to Zheng Ing. Together, they were then able to modify the minimax function for alpha beta pruning. However, no ordering for alpha beta was used. After completing algorithm Tatsuhiko tried to fix static evaluation function which use not only count pips but also bear off numbers and home checker numbers and hit checker number.

### Static evaluation function

Static evaluation function was initially a simple pip count (for early bird special), as we wanted to write the minimax function properly. Afterwards, in addition to pip, we gave an incentive for players to hit, penalizing the opponents greatly if they have checkers in the bar. Furthermore, we gave an incentive to bear off. We counted numbers of checkers which were out of the board. We added numbers of checkers which were in home side. After that, we calculated the difference between each of these four elements. And then, we calculated sum of these four numbers. However, since each of these four values has a different degree of importance, it was necessary to add a weight to each value. We adjusted a weight by watching description of assignments<sup>3</sup>.

## Stochastic Simplified Backgammon Agent

As the deterministic simplified backgammon closely followed the minimax wrote by Zheng Ing, he was able to write the expectiminimax code relatively easily.

## Partnership Retrospective

We did not have much issues working together. Sometimes there's a little language barrier between us but nothing too major. It was a little challenging to find time where both of us were free to discuss.

### Lessons you learned as a result of working in this partnership – Ching Zheng Ing

Zheng Ing took a great amount of time initially reading the course documentation. Hence at the start, he seemed to not have contributed much to the project. However, after thoroughly reading through the code, he was able to implement the code examples into the minimax. He followed the skeleton agent implementation closely, and successfully implemented the minimax which I had been stuck on for awhile. He was also able to determine where the different python functions rely on one another, hence being able to test our codes.

### Lessons you learned as a result of working in this partnership – Tatsuhiko Araki

Tatsuhiko takes the assignment one step at a time, implementing the static function and then moving on. In my case (Zheng Ing), I tend to get overwhelmed by the overall problem and spend many hours thinking of how to solve everything at once, which is difficult when the problem is large. Him taking these steps (such as starting on the static function) initially helped give me confidence in solving the overall issue. I could focus on minimax and applied wishful thinking to solve this issue. I assumed the static function would work beautifully, then could focus on minimax, Tatsuhiko also showed resourcefulness in sourcing Alpha Beta pruning coding examples that closely fit to my current minimax code.

### Additional comments

When it comes to static evaluation function, I (Tatsuhiko Araki) set weight of static evaluation function to pass the test cases by thinking about specific case. However, when considering how to use this program practically, we may need to change this value by asking experts to be able to handle more core cases.