Billy Lin

Reece Peters

"Assignment 3 for CSE 415, Winter 2021, University of Washington"

Deterministic Simplified Backgammon Agent

1. We worked together for every part of this agent.
2. **StaticEval Explanation:**

   We have several factors in evaluating the state of the board. We implemented a racing strategy by checking how far away each checker is from its respective homebase. Getting closer is better. This is the WEIGHT_RACE below, and it's the lowest since it's somewhat of a default option.

   We implemented a bearoff factor which encourages bearing off as many checkers as possible. Since this is how we win the game, this has the highest weight.

   We implemented a hit factor which encouraged hitting (and discouraged being hit). This has a fairly high weight because it's necessary for the strategy we decided to try and implement (i.e. locking an opposing checker out of the game).

   We encourage making stacks of two but no more than two. To encourage forming stacks on the files that form primes, we double the weight stack benefit for those specific indices.

   We also encourage making primes that are close to our homebase. A prime is a consecutive line of stacks that cannot be hit. By counting the number of primes we have and giving it a weight, we prioritize making primes to block our opponents' checkers in our own homebase.

```
WEIGHT_RACE    = 1
WEIGHT_BEAROFF  = 100
WEIGHT_HIT     = 20
WEIGHT_STACK    = 5
WEIGHT_PRIME    = 60
```

3. Any special considerations for Alpha-Beta pruning, such as ordering of successors best-first:

Stochastic Simplified Backgammon Agent

1. Who did what for this agent (not required if working alone).

   Same as above.

2. Other comments on the implementation.

   We wanted to try to implement Zobrist hashing, but found our solution cost too much memory and didn't give a staunch enough improvement in runtimes.

   We did implement alpha-beta pruning in expectiminimax which helped us reduce runtimes by a significant amount however.

Partnership retrospective (required for the partnership bonus).

1. What issues you faced or didn't face related to the partnership.

Sometimes we had scheduling conflicts which we had to work around. This was exacerbated by the difference in time zone.

2. Lessons you learned as a result of working in this partnership -- Billy Lin

   Reece is really good at math and he really brings the concept of math in implementing our evaluation function. He also provides very valuable comments and ideas in working together in our collaborative programming.

3. Lessons you learned as a result of working in this partnership -- Reece Peters

   Billy is a great partner and really helped me simplify complicated problems down to their core. There were several times where I felt frustrated and he managed to analyze the issue and come up with the next steps towards a solution. In addition, Billy is a very fast programmer compared to my somewhat sluggish pace. It helped me learn that I could afford to be more efficient with my time.