

Limit Hold 'em Poker

by Kevin McCusker

The purpose of this project is to create a Limit Hold 'em Poker game, to be used by anyone who wants to play poker against computer players, whether it is for fun, to get better at poker, or any other reason. The rules of the game are as follows: first, all players receive two cards face down, the first two players to the left of the deal each place an initial bet, the small blind and big blind, and then there is a round of betting. There are then 3 cards placed face up (called the flop), and another round of betting ensues. Another card is flipped up (called the turn), and then another round of betting follows. The final card is then turned over (the river), accompanied by the final round of betting. The stakes of this game is 100-200, which means the small blind is 50 chips, the big blind is 100 chips, bets before the flop and on the flop are 100 chips, and bets on the turn and river are 200 chips. The maximum amount of bets on any round is four, or 400 chips on the first two rounds and 800 on the last two rounds. An asterisk denotes which player is currently next to act.

To play the game, users simply start the program, read the instructions, and then begin to play. The computer players act one at a time, with a one second delay between turns so the player can see the action as it progresses. For the UML of this program, see separate diagram.

The computer AI makes decisions based on extensive checking of the hand they currently have, as well as the position the game is currently in. Because predictability is something you want to avoid in poker, decisions are based on a probabilistic determination of the hand. For example, if a computer player starts with the hand Ace-Ten he will raise 80% of the time, and call the other 20% of the time, whereas if a computer player starts with the hand Two-Three, he will fold 85% of the time, call 10% of the time, and raise 5% of the time. This ensures that the computer is making reasonable moves without being too predictable. Similar assessments are made as the ones above in every decision the computer makes.

The error handling in the program is to check the preconditions in every function, such as that the index for accessing a card in a player's hand (must be 0-1). The error is thrown, and caught in the main function.

In comparing data structures, I compared the heap implementation from "Ford and Topp" vs. a heap I created, used for checking the value of a flush hand, or a hand that only has high card. I also compared a custom vector vs. the STL vector, and the STL vector vs. an array for holding and accessing the cards in the deck. I thought that the "Ford and Topp" implementation would be faster, because it was not abstracted into a class, but rather used functions to create a heap. I thought that an array would be faster than both my vector and the STL vector, because it can access the elements directly without the overhead of a class. I also thought that my vector would be faster

than the STL, because it would only be resizing once, then accessing, and my class would contain less overhead than the STL.

The results of testing the data structures confirmed my hypotheses. I tried selecting every card in the deck 10,000 times with the array, my vector, and the STL vector, and this took .801 seconds, 3.283 seconds, and 4.196 seconds respectively.

For testing the heap, I tested how long it took to create the “high card” hand with the different data structures 1,000 times each. My heap took .323 seconds to do this, and the heap from “Ford and Topp” only took .231 seconds. I think the reason I got these results for both of the comparisons is the same as my hypotheses above.

This project did not require many references. The only references I used was the heap implementation in “Ford and Topp,” as well as using the book as a source for implementation for error handling.

Overall, I think the poker game turned out well. The computer AI makes appropriate decisions, and, according to testing, plays better than randomly. In testing half of the players playing randomly vs. half of the players using the AI, those using the AI were clearly coming out ahead in the long run. The game runs well, is fun, and can help improve the poker skills of users.