

### Compilation Instructions:

I have included a `makefile` for you. Just calling `make` will compile with your machines default version of `g++`. `make all` will recompile everything. Calling `make c11` will compile everything using the `-std=c++11` flag. I included a threading option which I will explain later. It requires a `c++11` compliant compiler. I have tested these compilation options on my personal computer and on the unix server. (Note: the `c++11` and threading does not work on the unix server so do not try that on unix)

Compiler: `g++` (MacPorts `gcc48 4.8.0_0`) 4.8.0

Hardware: 2.4 GHz i5 2Core

### Input:

There is no command line parameters. You can change the board size in `main()` and the number of simulations in `hex.cpp` at the top of the file.

Each user turn will take 2 values, an  $(x, y)$  coordinate.

### Monte Carlo AI Description:

My implementation of the AI uses a Monte Carlo Method. For every un-played spot, make a copy and make a move on that board. Now I have a copy of the original board with one more move for every un-played spot on the original board. Now for each of those I randomly generate the rest of the board and test who wins. I do this random generation the amount of times specified in `hex.cpp` as `SIMSPERMOVE`. All the way, I am keeping track of the amount of wins for the board with the move.

The theory being, a certain move on a certain board will give the computer an innate advantage in the game. This advantage will show itself in the random generation.

### My implementation of threads

Essentially what I have implemented, is that each board on which we are doing simulations (one for each un-played space) will have its own thread.

As I noted earlier this requires some special compilation instructions. First of all you need a `c++11` compliant compiler. Personally I used `g++ 4.8` on my machine. Furthermore, I put in some preprocessor directives so that we can choose whether to implement threading or not. To enable threading, uncomment out `#define C11` at the top of `hex.cpp`. Then compile with the `-std=c++11` flag on `g++` by running `make c11`.