Testing Instructions

Gomoku is the Japanese name for the common game - five in a row. Running the four commands below will showcase the functionalities.
Command 1 or other mis-specified commands will bring up the correct format for specifying the command.
Command 2 will start a game between two human players on a 9x9 board. I would suggest playing a friendly game to test user input, board representation, rule implementation and winning.
Command 3 will start a game between a human and a Bot. The Bot uses simple blocking and extension strategies specified in the program. There are situations where it does not make the best moves; however, it will still use what it's been taught.
Command 4 will start a game between two Bots. Just sit back and watch.

```
1. python Gomoku.py
2. python Gomoku.py human Player1 human Player2 9
3. python Gomoku.py human Human computer Bot 19
4. python Gomoku.py computer Bot1 computer Bot2 9
```

Project Status

In my mind, the Gomoku project is complete with the following features: 1) simple and robust user interface; 2) built-in rules and logic; 3) a decent computer player.
Computer strategies can clearly be further improved through more effort and more lines of code. However, what I am most excited about is to train a neural network that can learn by playing against itself and create an AlphaGomoku grandmaster.

Challenges

As I built the program, one challenge I faced was that many parts of the program require pattern matching. Initially, I needed to evaluate if either player had won the game, and so I wrote code to check rows, columns and the two diagonal directions and see if there are five in a row. Later, I realized that computer players also needed to check patterns - 2/3/4 in a row. It became repetitive and difficult to maintain, so I created a Pattern class that checks n in a row. In the original project design document, I didn't think about the Pattern class, but this led me to create a Pattern class to centralize and reuse pattern matching code.

Another challenge I faced was checking patterns on the diagonal lines. I had a hard time working out the algebra within the for loop, and in the end Stackoverflow came to my resort. I also discussed it with a friend and found that it can be thought of as axis rotation.