# Chess AI Final Project

## Advice on getting started

1. Fire up a Python 3 environment, and clone the repo:  
     
   git clone <https://github.com/marcusbuffett/command-line-chess.git>
2. Read this **carefully**: http://blog.mbuffett.com/creating-a-basic-chess-ai-using-python/
3. To get a feel for the interface, here is what I did:  
     
   Typed **ipython**  
   Then typed **%run main.py**I then used the ? to figure out the different options and noted that typing l allows me to see all the legal moves (something we will definitely need).

### Task 1 (10 points)

Create a new file called **play\_game.py** (start by copying main.py) that plays a fully automated game (i.e., no input from a user needed). Do NOT create a smart AI to battle against the built-in AI. Instead, use the getRandomMove function that is already implemented for you. In other words, create a random agent to compete against the built-in minimax AI.

### Task 2 (10 points)

Create a new file called **alpha\_beta\_game.py** (start by copying main.py) that plays a fully automated game improving on the built-in minimax AI using the alpha-beta pruning algorithm you know and love. Your AI should be able to beat the minimax solution implemented by the original authors.

### Task 3 (10 points)

Create an improved AI algorithm using one of the other methods we implemented and used in Pacman (e.g., classification, reinforcement learning, etc). Call this **best\_game.py** (start by copying main.py). This should beat your other AIs obviously. One thing I would love to see people attempt is move ordering (<https://chessprogramming.wikispaces.com/Alpha-Beta>. Don’t get too caught up in that though. Go with what you are most comfortable in but do a little research before picking something. Chess AI has been studied a lot, so do some reading (for example: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.109.810&rep=rep1&type=pdf>).   
  
Task 4 (0 points but this is required if you want ANY points)

Create a file called **readme.txt** that describes how you implemented the previous three tasks. If you had problems with any of the above tasks, then indicate that in this readme. I expect extra care when describing what you did for Task 3 as it is so open ended. I cannot stress this enough, if you do not include this **readme.txt** file, I will not grade your submission 😊