

Forum:[Hack Forums](#)

Thread: [Trying to simulate basic tournament](#)

Andy Dwyer - October 22, 2019, 10:56 am

Hi guys

I'm having a bit of a meltdown with some Uni work. We have been studying python for only a month or so, so my knowledge is still pretty shit, but we have now an assignment that is going to be very difficult for me.

I do not want someone to do this for me, I really would just appreciate some guidance on how I am going to write this myself. Right now I am trying to write some basic functions to begin but I keep getting stuck and cannot wrap my head around this because it's a big task.

The assignment:

Spoiler ([Click to View](#)) The purpose of this assignment is to simulate a chess tournament. In order to do that you

have first to simulate a game of chess. A realistic actual game of chess (e.g. horse at position G5) is not to be implemented – just the processes described below.

Read the LawsOfChess.pdf from FIDE (international chess federation) for the bigger picture.

You will have to simulate a simplified version of a chess tournament.

A game of chess (imaginary version) between two players includes the following.

- a) Each player is randomly assigned a colour, white or black. The white moves first.
- b) Each player has a set of possible legal and illegal moves. Legal moves are: move a piece, capture a piece and checkmate. Illegal moves are: touching a piece and then moving a different one, moving your pieces into check, moving a piece of the opponent.
- c) Legal moves are more likely than illegal moves.
- d) Legal moves happen with different probabilities. Moving a piece is more likely than capturing an opponent's piece. Checkmate is the least likely and the player that does it, must have done at least 10 moves.
- e) Each player has a clock that counts time passed while the player is thinking. When one player's clock stops the opponent's, clock starts. Each move can last between 10 seconds to 5 minutes.
- f) If a player surpasses 5 minutes (cumulative time in a game) they automatically lose.
- g) If a player does 2 illegal moves in a game they automatically lose.
- h) If the game has more than 25 moves (per player) without checkmate it is a draw.
- i) The player that does checkmate wins.
- j) A win counts as 2, a draw as 1 and a loss as 0.
- k) All players' moves are recorded.

You will have to simulate a two stages tournament of 5 players.

First simulate round-robin (play-all) tournament. Each contestant competes (plays a game of chess) against all opponents. For example, Player 1 will play against Players 2-5, Player 2 will play against Player 1, Players 3-5 and so on.

Then the two players with the highest score, from the robin round will play against each other 3 games. If there are draws, the player with the least playing time will progress.

Whoever wins 2 games wins the tournament. If a player wins 2 games in a row they will not play a third.

All games and final scores are recorded.

Right now I have a function "stage_game()" that contains variables that keep track of the moves taken, if they are illegal, etc. Within this function I want to run a "while scores < 25, and times < 300s" (pseudo I know), iterate over a chess move and log this to txt. My problem isn't with the syntax of the logging or the iteration, it's just where to get started. This is my first proper Uni assignment and I'm feeling super overwhelmed.

Again, I want to do this myself, I do not want any code written for me. I just really need some advice on where to begin. Right now I keep getting started then writing code that is too confusing then starting again. Literally any suggestions would be very welcome right now.

Danke Leute
thanks guys in advance

Goodies - October 22, 2019, 1:50 pm

Hey man, sorry you're feeling overwhelmed. It's a good idea to write or even draw out a high-level overview of what actually needs to be done. Python is easy enough to write in an extremely modular fashion.

A) Should be incredibly simple, just use some variant of returning `random.choice(["white", "black"])`

B) This will be where the most amount of logic will be for the program of course. Your best bet is to list out all possible valid moves for each piece and have some global constraints. For example, I would create a class for "ChessPiece" and then subclass it for each type of piece: pawn, rook, knight, bishop, queen, king. I would then have a function called `move_is_valid` (or similar) to determine if a move is valid. The parent class should check only GLOBAL constraints such as whether or not the piece is being moved off the board. Each child class should invoke the parent class implementation and then perform its own validation specific to its character.

C) Wait are the moves being randomly generated?

D) Seems weird, scholars mate can happen in 2 moves lol but rules are rules...

E) I'm honestly not sure how they would advise you do this with only a month into programming... if it were me, I would have another thread running in the background with a timer that sets a `threading.Event()` object which indicates whether the player can/can't move. But thread synchronization and primitives sound difficult.

F) That's a tough game lol I never liked speed chess. There would be either one or two threads.

G) Player-specific counter. Easy enough.

H) Global counter. Easy enough.

I) Duh

J) See G.

K) I would use a deque (double-ended queue) from the collections module which allows it to easily be recanted in order, but a list would work fine as well.

I know this isn't super helpful but hopefully it points you in the right direction.

datguy.dev - October 24, 2019, 10:35 am

It seems pretty straight forward besides the board creation, legal and illegal moves etc. I would focus there first as that will be the most time consuming part. I imagine this would also be hard if you aren't used to using classes or creating your own libraries for organization. This project is surely going to be a long winded one.

Check out: <https://github.com/niklasf/python-chess> Do a bit of a reverse lookup on the functions and find out how it generates valid moves and stuff? Nothing wrong with using a reference.

Andy Dwyer - October 26, 2019, 2:37 pm

Cheers for the replies guys!

So far I've got each player as an instance of the Player class. Stage_game() takes two players and calls make_move() for the players sequentially while all of the conditions are satisfied (either player doesn't have too many illegal moves, etc.). The problem I have now is that the logs seem a bit messed up.

The stage_game function basically just calls make_move(player1) then make_move(player2) while the conditions are true, yet in the logs sometimes player 2 takes 3 moves in a row, or player 1 takes 2 moves then the game finishes. I'm a bit confused with this because surely the functions are being called one after the other yet the logs seem sporadic. Any tips?

Thanks again.

edit: Fuck me, I'm retarded. Testing a dice roll between 1 & 10, I meant to write (if roll >= 0 and roll <= 6) and I wrote (if roll <= 0 and roll <= 6), so the roll on that array was never less than zero. Jesus Christ I can't believe literally one character has fucked up my whole day haha.

datguy.dev - October 26, 2019, 7:21 pm

(Oct 26, 2019 02:37 PM)Andy Dwyer Wrote: [\[CITE\]](#) Cheers for the replies guys!

So far I've got each player as an instance of the Player class. Stage_game() takes two players and calls make_move() for the players sequentially while all of the conditions are satisfied (either player doesn't have too many illegal moves, etc.). The problem I have now is that the logs seem a bit messed up.

The stage_game function basically just calls make_move(player1) then make_move(player2) while the conditions are true, yet in the logs sometimes player 2 takes 3 moves in a row, or player 1 takes 2 moves then the game finishes. I'm a bit confused with this because surely the functions are being called one after the other yet the logs seem sporadic. Any tips?

Thanks again.

edit: fugg me, I'm retarded. Testing a dice roll between 1 & 10, I meant to write (if roll >= 0 and roll <= 6) and I wrote (if roll <= 0 and roll <= 6), so the roll on that array was never less than zero. Jesus Christ I can't believe literally one character has fugged up my whole day haha.

I get that way after too many hours coding. Less time if I get frustrated. Spending days creating a program to find out there is an API available already... ? Make sure you take breaks if you aren't. Also, it's not really possible to help you without the code for future reference.