Chess J.

Version log:

* Chess A3-4 - doesn’t allow passage through pieces
* Chess A3-3 - adding piece movement options in class
* Chess A3-2 - making all pieces move able
* Chess A3-1 - copying in some of the stuff from A2-1.
* Chess A3 - 07.06 - starting to move pieces

To do:

* Sort bugs!!
* Add check
* Add check mate
* Add kings can’t touch
* Add pawns can ‘jump’ from start
* Add that French thing that allows pawns to take pawns if the ‘jump’
* Add queening - when a pawn gets to the opposite side
* Add castling
* Then start creating opponent.
* Convert to 3 levels - game, shapes, rules.

Log:

23.07.20

So now need to check whether any pieces on that square.

Well first check if in check

Then check if move put into check!

Done, says check if currently in check

* Chess A3-4-2 - says check if currently in check

Should easily be able to adapt for moving into check. Didn’t manage to sort. Started to get buggy, recommend starting with Chess A3-4-2 and turns parts into functions to simplify the code then go back to add check

Back, will try and clean up first…

* Chess A3-4-2-3 - cleaned up but buggy

22.07.20

So today I will try and add a win condition, king can’t move out of check, will have to add stalemate to. First need to add check rule, so home piece can’t move a piece so it’s piece is in check.

At this point I’m think maybe it would be better to transform this game into a different dimension. As in at the moment it works on a kind of rules based on piece position. But maybe it could be more sophisticated than that and instead work in vectors, like have 3 a level operation. Actual chess on top, which converts to shapes, which converts to rules.

That would probs be more efficient. For now will stick to rules.

So firstly I can’t move my piece if it’s pinned. So I need to check it is pinned, i.e. check that there is anyone piece attacking my king.

So first I need to find the kings position, I then then need to check all the squares around it? prehaps this is where I need to introduce shapes…

For this to work I need to check all pieces and a lot of options. Well I need to check for any pieces on the diagonals, the vert/hoz and knights

* Chess A3-4-1 - highlights all possible squared for king to be attacked from

Now need to check if there is relevant threat.

10.07.20

Don’t let pieces move over each other

So find all the squares in-between and check for either side pieces

* Chess A3-4-0 - doesn’t allow passage through pieces

09.07.20

Adding these rules so pieces can’t go over pieces is harder than I thought. Because not only is it the piece it lands on but it’s also all the pieces it goes through and so far nothing is considering those squares. So will need to calculate them.

Spent 40 minutes trying and failing to build a class-function that I had already made. Complete forgot how to use objects in terms of doing class-functions with them. Not sure how I’ve managed to forget everything. Didn’t even realise how much I’d forgotten till I realised how dumb I was being, comparing what I was trying to do which what I had done before. Feels like trying to simulate a dynamic system in my head. Need to just plan it out.

Because pawns act differently I am getting very confused. Pawns normal movement is like the rest. Pawns movement for attack is in the attack function whilst the rest is in the move function and then if there is a piece there it is removed. But because the remove function isn’t in the move function (because pawns are different) pieces are getting remove regardless of legality. So what I really need to do is separate pawns as their movement at taking mechanics are different. But even writing a simple if statement for pawns confuses me. so the current piece is saved in the for. How to I test is variables can I just write the if straight into the main loop or do I need to write a clad-function and sort it there and return yes or no. surely I can access the objects variable in the main loop or what would be the point.

Would help if when you looked for help in python you were looking at python manuals not java!

Done a bit

* Chess A3-3-6 - stopped self collision end (not move through)
* Chess A3-3-7 - sorted pieces move and don’t collide with same colour

now to add don’t collide with other colour really only for pawns so will add that to that

Next will sort bug (back row don’t take)

* Chess A3-3-8 - king movement added and bug removed - all pieces move and take

Can’t seem to stop pawns standing on top of other pieces. It should be simple to sort but I’ve sat here for 20 minutes and failed????

40 mins to realise my if function was imbedded in my for loop. I wanted all the for loop to check for a certain condition and then do something else, there’s a different operator for that probably but I’ve done it this way now, where for loop assigns a variable true if condition met at any instance and if function reads that variable. Done that twice not very smooth.

* Chess A3-3-8 - sorted basic movement and rules (forgot to re-save)

01.07.20

First will stop pieces colliding

Add movement options instead, still need to add collisions

Chess A3-3-4 - all movement options added just need to sort destruction for pawns/other pieces

(Either pawns is a special case or need to make special case for all)

30.06.20

Leggo

* Chess A3-3-2 - Sorted out flickering thing.

Now there is lots to think about. I suppose the order isn’t important.

First I will add turn, white goes first or “light”

* Chess A3-3-3 - turn sorted, I’m so good at this lol

Next? Add the ability to destroy each other pieces i.e function take destroy

Almost sorted, for some reason pawns now move horizontally.

Sorted that, now just need to allow to take both horizontals.

Sorted

* Chess A3-3-4 - pawns can eat.

29.06.20

Retuning to this after weeks.

* Chess A3-3-1 - testing

So pawns have been kind of done. I think first the best thing to do is add same colours from moving on top of each other

I’ve tried to clean up but made it flashy and error prone. Might need to look up why and how to fix/better ways to do it.

07.06.20

From yesterday - So first click choses a square and check piece there, second click either moves, does nothing, or unselects Original Square. Repeated taps on Illegal squares does nothing.

First click checks to see any pieces on square if piece is there highlights then wait for next click. Ignoring legal/illegal moves at this point. Then add the distance of the new click to piece. Then can add unclicked, then add limitations and removing other pieces.

3:37pm - okay can move piece around. First will add multi-pieces then add rules for single piece

4:07pm - successfully copied all parts in board is set up but only extra piece can move now need to make all pieces move able.

4:39pm - worked out how to break a loop

5:43pm - can move all the pieces + select/deselect works. next? Add the movement rules can’t move to square occupied by same colour, maybe first add the movement rules.

So if I create a new function of the pieces class called options and it uses the class of piece to determine the option a piece has and then only those options turn into the areas which can be click on. Then I can add other rules into options such as piece blocking and eventually check and eventually AI for ‘enemy’.

This options thing is a nice idea but I need to work out where I am gonna put it.

So have hardcoded rule for pawns would like to find a better way or create a library - tomorrow!

06.06.20

LS - 0, 2, 360, 540 = 0, 240, 450, 630

LS - 90, 240, 450, 630 = 90,

Yea so good starting idea, first create the board and the pieces..

3pm - not using a clear strategy trying to work out inside my head.

So what I’m trying to do is - populate the pawns. Create 8 pawn pieces, with their independent positions stored.

4:40 - So I have an option for the back row, either add all the pieces using one line of dictionary.

But each piece will need its own class anyway to follow its rules of movement etc right?

Tbh I’m not sure how I’m going to implement the rules for each piece. I’m imagining the knight. I reckon at some point I’m gonna need to change it to non-continuous positons.

5:12pm - learning how to put in buttons.

Either create a transparent button or put the button

So I know how to create a button but I want all the buttons to do the same thing. Also I need like an index to connect buttons, pieces and the squares they are on. Current pieces are put at a positons saved in a list. The list is ?

7:44pm - well maybe I can do it with collisions using clicks as collisions. I need to map it all out in some way.

Well now I’ve got options, do I want the button to be transparent until clicked on?

A2-3-3 - basic on/off switch for button working. Now need to scale for all buttons, then need to connect to piece. (Working on smaller 2x2 grid)

So do I need to check that every button has been pressed every time? - smashed by using the positon of click, however no remove it just repaints over.

Now to connect piece and square. Every square will be an object noting its piece and every piece an object noting its square.

So now need to create a grid to make the positons easier, lucky chess already has a system..

11pm, so got the letters and numbers… took an hour though.

Next to create objects of positions? Unless I can feed the click position into the list of pieces and add the nessacery units of movement directly, cause I can for pieces in pieces if x = if if y = then add next click. So need to able to toggle the first click and only allow clicks on legal moves. So first click choses a square and check piece there, second click either moves, does nothing, or unselects Original Square. Repeated taps on Illegal squares does nothing. Pigeon hole or tunnel vision.

02.06.20

So first thing is to create the chess board which will be quiet easy. User interaction is next - which will depend on what types of inputs you can use in pygame, hopefully there is a clock option.

Plan - a1:

Create pieces class for all pieces

Then class for specific pieces - 6 types, which will hold their rules somehow

First generate board and pieces and go from there

16.03.20

First create the board and pieces

Add interactive element (inputting moves)

Then add: rules for pieces movement, win/lose, turn

Then add random moves for computer

Then add strategy for computer. Initial plan - defensive strategy predicting check mate, attacking strategy random.

Then come up with attacking strategy theory.

Then combine the two. Or total control theory.

Then read and compare.

15.03.20

Chess is a game of options.

If a piece is taken it has 0 options.

The aim of chess is reduces the kings options to 0.

For example all pawns have 3 options to start with. Along with the knights.

All other pieces have 1 option. That is nothing.

Maybe this is something to do in python, first make a logic one. Then using tensor flow. Make a machine learning one.