Chess architecture:

* Every figure has own class
* Main attributes: color, weight, name, short\_name(displays on desk)
* Possible moves(прописать в каждом классе )

Desk содержит координаты всех фигур в отдельных переменных

Done:

Создается пустая доска 8х8

Bugs:

Neuron architecture:

Входной слой(подготовка данных): 16 нейронов(фигуры)

* Вход: текущие координаты [x,y]
* выход возможные будущие[[x1,y1],[x2,y2],[xn,yn]]

1 скрытый слой: оценка 1 хода каждой фигуры в баллах

* вход: [[x1,y1],[x2,y2],[xn,yn]]
* выход : {[x1,y1] : кол-во баллов,[x2,y2]: кол-во баллов,[xn,yn]: кол-во балло}

2 скрытый слой: 8 нейронов

* вход: {[x1,y1] : кол-во баллов,[x2,y2]: кол-во баллов,[xn,yn]: кол-во балло}
* выход: половина

3 скрытый слой: 4 Нейрона

* вход: {[x1,y1] : кол-во баллов,[x2,y2]: кол-во баллов,[xn,yn]: кол-во балло}
* выход: половина

4 скрытый слой: 2 Нейрона

* вход: {[x1,y1] : кол-во баллов,[x2,y2]: кол-во баллов,[xn,yn]: кол-во балло}
* выход: оптимальный ход + кол-во баллов {[x1,y1]:кол-во баллов}

В случае равного кол-во очков выбор рандомом(мб файл?)

далее просчет 4 ходов имея и выбор оптимального пути

далее запустить просчет ходов для обеих команд, для выбора оптимального исхода из 5 ходов.

При желании можно запустить просчет до мата, но время расчета будет от 3 мин.

Bugs:

What to -do: