Первая проблема, с которой я столкнулся, - названия полей.

Я бы хотел оставить их нормальными, как А1, Е6 и т.д.

Но с другой стороны нужно дать компьютеру возможность понимать эту доску. То есть:

1. Компьютер должен понимать, что на этом поле стоит определённая фигура, или же этой фигуры тут нет.

2. Компьютер должен уметь "сдвигаться" по доске. То есть перейти с условно А6 на В7.

Давайте подумаем, что произошло с токи зрения человека.

С моей точки зрения буквенное значение увеличилось на 1. Численное значение тоже увеличилось на 1.

Теперь подумаем над тем, как это приподнести компьютеру:

chr(65)

А

Функция chr(число) возвращает нам значение в Юникоде. То есть за 65 номером скрыт символ заглавной А

ord('A')

65

Функция ord(символ) возвращает нам значение символа Юникода в число. То есть за А закреплено число 65.

Вот пример того, как можно изменить значение А на В

>>>ord('A')

65

>>>n = ord('A')+1

>>>chr(n)

'B'

Таким образом мы добились того, чтобы компьютер увеличивал/уменьшал значения букв.

С цифрами всё понятно. Но теперь надо сделать так, чтобы за каждой клеткой было закреплено положение фигуры на ней.

То есть условно говоря, что на поле А1 есть фигура и она называется ладья.

И НАОБОРОТ! Нужно, чтобы компьютер понимал, что ладья стоит на поле А1.

Теперь осталось подумать, как задать доску. Неужели придётся каждой клетке писать по функции?

То есть

def CELL(NAME, PIECE):

Где name - имя клетки, а piece - это фигура, которая стоит на этой клетке.

Но теперь другая проблема. Как мы будем определять положение фигуры на доске.

То есть нам нужно понять: стоит ли пешка на второй горизонтали или нет.

К тому же, как отделять цифры от букв в названии клеток?

Допустим, пешка стоит на поле А2.

Что положено положено по функции CELL в эту клетку?

Поле А2 у нас закреплено…

Ладно, давайте напишем для начала эту функцию, а потом посмотрим.

Вот и проблема с функцией. Зачем нам знать, какая фигура?

//IMPORTANT: WHEN MOVING, DON’T FORGET TO EARASE PRIVIOUS PIECE POSITION ON BOARD IN ORDER TO SAVE REAL PIECE BALANCE AND NOT DUPLICATE PIECE!!!//

def CELL(LETTER, NUMBER):

if (LETTER == 'A' and NUMBER == '1'):

PIECE\_IN\_CELL\_A1 = WHITE\_ROOK

if (LETTER == 'A' and NUMBER == '2'):

PIECE\_IN\_CELL\_A2 = WHITE\_PAWN

if (LETTER == 'A' and NUMBER == '3'):

PIECE\_IN\_CELL\_A3 = NO\_PIECE

if (LETTER == 'A' and NUMBER == '4'):

PIECE\_IN\_CELL\_A4 = NO\_PIECE

if (LETTER == 'A' and NUMBER == '5'):

PIECE\_IN\_CELL\_A5 = NO\_PIECE

if (LETTER == 'A' and NUMBER == '6'):

PIECE\_IN\_CELL\_A6 = NO\_PIECE

if (LETTER == 'A' and NUMBER == '7'):

PIECE\_IN\_CELL\_A7 = BLACK\_PAWN

if (LETTER == 'A' and NUMBER == '8'):

PIECE\_IN\_CELL\_A8 = BLACK\_ROOK

Возьмём для примера эту функцию. Мы спросили у программы, какое есть число и какая буква. Мы узнали, что в каких-то клетках есть какие-то фигуры, но теперь надо узнать, какие фигуры стоят на каких клетках!

def CELL(LETTER, NUMBER):

if (LETTER == 'A' and NUMBER == '1'):

PIECE\_IN\_CELL\_A1 = WHITE\_ROOK

if (LETTER == 'A' and NUMBER == '2'):

PIECE\_IN\_CELL\_A2 = WHITE\_PAWN

if (LETTER == 'A' and NUMBER == '3'):

PIECE\_IN\_CELL\_A3 = NO\_PIECE

if (LETTER == 'A' and NUMBER == '4'):

PIECE\_IN\_CELL\_A4 = NO\_PIECE

if (LETTER == 'A' and NUMBER == '5'):

PIECE\_IN\_CELL\_A5 = NO\_PIECE

if (LETTER == 'A' and NUMBER == '6'):

PIECE\_IN\_CELL\_A6 = NO\_PIECE

if (LETTER == 'A' and NUMBER == '7'):

PIECE\_IN\_CELL\_A7 = BLACK\_PAWN

if (LETTER == 'A' and NUMBER == '8'):

PIECE\_IN\_CELL\_A8 = BLACK\_ROOK

Вот так выглядит программа. Но что она должна возвращать?

Правильно, она должна возвращать: какая фигура там стоит!

Теперь программа возвращает, какая фигура там стоит.

Следующая задача: научить компьютер ходить разными фигурами. Хотя бы просто объяснить ему, как ходят фигуры, а затем уже объяснить остальные правила.