class GameState():

def \_\_init\_\_(self):

#the board is a 7x9 2d list

#the first character illustrates the colour of the piece, either 'r' or 'b'

#the "" empty string represents an empty space

self.board = [

["","rA","rG","rE","rG","rA",""],

["","","rG","rG","rG","",""],

["","","rG","rG","rG","",""],

["","","","","","",""],

["","throne","","throne","","throne",""],

["","","","","","",""],

["","","bG","bG","bG","",""],

["","","bG","bG","bG","",""],

["","bA","bG","bE","bG","bA",""]]

self.blueToMove = True

self.moveLog = []

import pygame

pygame.init()

yellow,brown,black = (245,237,128),(218,187,87),(0,0,0)

gameDisplay = pygame.display.set\_mode((800,600))

pygame.display.set\_caption("ChessBoard")

#Size of squares

size = 60

#board length, must be even

boardLength = 8

gameDisplay.fill(black)

cnt = 0

for i in range(1,boardLength+1):

for z in range(1,boardLength+1):

#check if current loop value is even

if cnt % 2 == 0:

pygame.draw.rect(gameDisplay, yellow,[size\*z,size\*i,size,size])

else:

pygame.draw.rect(gameDisplay, brown, [size\*z,size\*i,size,size])

cnt +=1

#since theres an even number of squares go back one value

cnt-=1

#Add a nice boarder

pygame.draw.rect(gameDisplay,brown,[size,size,boardLength\*size,boardLength\*size],1)

pygame.display.update()