Checkers Documentation

class piece:

def \_\_init\_\_(self, color, king, life, canmove, movelist, canjump, jumplist, square):

Initializes the attributes of each checker piece

def draw\_board():

Input: None

Output: None

Changes the background and draws the checkerboard

def draw\_pieces():

Input: None

Output: None

Draws the starting checker pieces and assigns attributes to each piece

def draw\_rules():

Input: None

Output: None

Changes the background to the image of rules

def menu():

Input: None

Output: None

Changes to the menu background where the player can clic whether to play, read the rules or quit the game

def coordstosquare(des\_sqr\_coords:list, blacksquares: dict):

Input: Coordinates list [x,y] and blacksquares dictionary

Output: Square name (e.g. ‘a7’)

Takes the coordinate values and searches the dictionary for the corresponding square name

def movability(currentpiece, sameteampiecelist, oppteampiecelist):

Input: a checker piece, the list of pieces from the same team, and the list of pieces from the opposite team

Output: “force” or “dontforce” (signifying whether jumps need to be forced in a given turn)

Adds all a checker’s possible moves to its movelist and all its possible jumps to its jumpdict

def squarecursorlocation():

Input: None

Output: Square the cursor is on

def draw\_king(square\_name):

Input: Square name

Output: None

Adds king piece indicator (crown image)

def highlight(square\_name):

Input: Square name

Output: None

Highlights possible moves yellow

def highlight\_blue(square\_name):

Input: Square name

Output: None

Highlights possible move cursor is hovering over, blue

def redraw():

Input: None

Output: None

Redraws the checkerboard and checker pieces after a player makes a move

def gameplay():

Input: None

Output: None

Runs the game: sequence of moves, whether a team has won