// Version 2-29-20 12:13 AM - Sami Melhem/Evan Wolfe

// Added a loop to restart the title screen when navigating to and from the rules page and 2 loops in the Rules to restart the

import java.util.Scanner;

import static java.lang.System.\*;

public class ChessRunner

{

private static Scanner keyboard = new Scanner(System.in);

public static void main(String[] args)

{

int i = 0;

while(i!=1)

{

out.print('\u000c');

out.println("###################################");

out.println("# WELCOME TO CHESS: JAVA EDITION! #");

out.println("###################################");

out.println("\n\nRules (1)\nStart (2)");

int f = 0;

while(f!=1)

{

String rules = keyboard.nextLine();

switch(rules)

{

case "1": printRules();f++;break;

case "2": i++;f++;break;

default: out.println("\nInvalid input. Please input either \"1\" for the rules of Chess: Java Edition, or \"2\" to start.\n");break;

}

}

}

out.print('\u000c');

out.println("###################################");

out.println("# WELCOME TO CHESS: JAVA EDITION! #");

out.println("###################################");

out.println("\n\nPlaying by yourself? (1)\nOr playing with a friend? (2)");

String players = keyboard.nextLine();

i--;

while(i!=1)

switch(players)

{

case "1": out.println("\nThat requires a coded AI, and that has not been written yet. Please try again.");players = keyboard.nextLine();break;

case "2": Multiplayer.start();i++;break;

default: out.println("\nInvalid number of players, please enter either 1 or 2");players = keyboard.nextLine();break;

}

}

public static void printRules() //Contains a table of contents, navigable via inputting an integer

{

String pageNumber = "";

int i = 0;

while(i!=1)

{

int j = 0;

out.print('\u000c');

out.println("###################################");

out.println("# WELCOME TO CHESS: JAVA EDITION! #");

out.println("###################################");

out.println("\n\nWhat page of rules would you like to view?\nBasic Rules(1)\nSpecial Rules(2)\nHow to Win(3)\nBack to Main Menu(4)");

String page = "";

int f = 0;

while(f!=1)

{

pageNumber = keyboard.nextLine();

switch(pageNumber)

{

case "1": page = "Basic Rules";f++; break;

case "2": page = "Special Rules";f++; break;

case "3": page = "Winning Rules";f++; break;

case "4": i++;j++;f++;break;

default: out.println("\nInvalid input. Please input either \"1\" for the basic rules, or \"2\" for the special rules, \"3\" for How to Win, or \"4\" for the Main Menu");break;

}

}

while(j!=1)

{

if(page.equals("Basic Rules"))

{

out.print('\u000c');

out.println("###################################");

out.println("# WELCOME TO CHESS: JAVA EDITION! #");

out.println("###################################");

String print ="\n\nKing(k or K) can only move one spot in any direction\n";

print += "Queen(q or Q) can move diagonally, vertically, & horizontally, can move as many spots if applicable, and can not move like a knight\n";

print += "Bishop(b or B) can move diagonally and can move as many spots if applicable\n";

print += "Knight(n or N) can move like an L: forward two & left or right, and can jump over pieces\n";

print += "Rook(r or R) can move horizontally or vertically and can move as many spots if applicable\n";

print += "Pawn(p or P) can only move forward and can move forward twice if pawn has never moved, can take out opponents one space diagonally\n\n";

print += "What page would you like to visit next?\nTable of Contents(1)\nSpecial Rules(2)\nHow to Win(3)";

out.println(print);

f--;

while(f!=1)

{

String answer = keyboard.nextLine();

switch(answer)

{

case "1": page = "Table of Contents";j++;f++; break;

case "2": page = "Special Rules";f++; break;

case "3": page = "Winning Rules";f++;break;

default: out.println("\nInvalid input. Please input either \"1\" for the Table of Contents,\"2\" for the Special Rules, or \"3\" for How to Win");break;

}

}

}

else if(page.equals("Special Rules"))

{

out.print('\u000c');

out.println("###################################");

out.println("# WELCOME TO CHESS: JAVA EDITION! #");

out.println("###################################");

String printSP = "\n\nCastling: This involves the King and the Rook, the king moves two spaces to the right or left and always overlaps the Rook, the Rook moves two spaces on the king's side or three spaces on the queenside, the King and Rook must've never moved before castling, there are no pieces in between the King and the Rook, the King must not be in check, and the king can't pass through a square that makes the King under attack\n";

printSP += "Promotion: When you can promote a pawn to a Queen, Knight, Rook, or Bishop\n";

printSP += "En Passant: When a pawn(1) can take out another pawn(2) after that pawn(2) moved forward two spaces, so pawn(1) may move diagonally to take out the pawn(2)\n";

out.println(printSP);

out.println("What page would you like to visit next?\nTitle Screen(1)\nBasic Rules(2)\nHow to Win(3)");

f--;

while(f!=1)

{

String answer = keyboard.nextLine();

switch(answer)

{

case "1": page = "Table of Contents";j++;f++; break;

case "2": page = "Basic Rules";f++; break;

case "3": page = "Winning Rules";f++;break;

default: out.println("\nInvalid input. Please input either \"1\" for the Title Screen,\"2\" for the Basic Rules, or \"3\" for How to Win");break;

}

}

}

else if(page.equals("Winning Rules"))

{

out.print('\u000c');

out.println("###################################");

out.println("# WELCOME TO CHESS: JAVA EDITION! #");

out.println("###################################");

String printW = "\n\nCheck: When a piece is in the path to take out the King. To get out of check you move away from the king, block the path with another piece, or capture the queen. \n";

printW += "Checkmate: When you win the game, which means you or the opponent can't move the king away, block the path, or capture the piece to prevent losing the game\n";

printW += "Draws: When you and the opponent have tied. This could happen due to the king not being in check but can't move anywhere or block a path, choosing to draw, not enough pieces on the board to force a checkmate, a player declares a draw if the same exact position is repeated three times, or five consecutive moves have been played where neither player has moved a pawn or captured a piece \n";

out.println(printW);

out.println("What page would you like to visit next?\nTitle of Contents(1)\nBasic Rules(2)\nSpecial Rules(3)");

f--;

while(f!=1)

{

String answer = keyboard.nextLine();

switch(answer)

{

case "1": page = "Table of Contents";j++;f++; break;

case "2": page = "Basic Rules";f++; break;

case "3": page = "Special Rules";f++;break;

default: out.println("\nInvalid input. Please input either \"1\" for the Table of Contents,\"2\" for the Basic Rules, or \"3\" for the Special Rules");break;

}

}

}

}

}

}

} // slight changes to printing the rules so it prints, and formatting(added Knight jumping) - Alex