import java.util.Scanner;

public class BattleShips{

public static boolean turn(Board board1){

boolean myTurn=true;

boolean fired=false;

boolean alreadyNotFired=false;

Canon canon1 = new Canon(true);

Scanner scColFire = new Scanner (System.in);

Scanner scRowFire = new Scanner (System.in);

board1.showFog();

System.out.println("Enter column number");

int colNumFire=scColFire.nextInt();

System.out.println("Enter row number");

int rowNumFire=scRowFire.nextInt();

//alreadyNotFired= board1.checkFireBefore();

fired=canon1.fire(rowNumFire,colNumFire,board1);

if (fired)//&&(alreadyNotFired))

myTurn=true;

else

myTurn=false;

return myTurn;

}

private static Board initialize (){

boolean myTurn=true;

Board board1 = new Board(10);

Scanner sc = new Scanner(System.in);

System.out.println(" PRESS s if you want to show BOARD");

System.out.println(" PRESS p if you want to place boats");

System.out.println(" PRESS c if you want to cancel boats placement");

Scanner sc1 = new Scanner (System.in);

int colNum;

int rowNum;

Scanner scCol = new Scanner (System.in);

Scanner scRow = new Scanner (System.in);

int colNumFire;

int rowNumFire;

Scanner scHVPos = new Scanner (System.in);

char inpHVPos;

Scanner scColFire = new Scanner (System.in);

Scanner scRowFire = new Scanner (System.in);

while(myTurn){

char input=sc1.next().charAt(0);

switch (input) {

case 's': board1.show();

break;

case 'p':

{

try

{

System.out.println(" Enter column number");

colNum=scCol.nextInt();

System.out.println(" Enter row number");

rowNum=scRow.nextInt();

System.out.println(" Enter Boat Position (for horizental press 'h' and for vertical press 'v')");

inpHVPos=scHVPos.next().charAt(0);

//if((x < 0 || x >= numRows) || (y < 0 || y >= numCols))

if((colNum<0||colNum>10)||(rowNum<0||rowNum>10))

System.out.println("You cannot breach the grid limit i.e. from " + colNum+ " \* "+rowNum);

board1.placeBoat(rowNum,colNum,inpHVPos); //Rows \* Columns

}

catch (ArrayIndexOutOfBoundsException e) {

System.out.println("You should not enter a number greater then Board limit");

System.out.println("Place the ships again");

}

break;

}

case 'c': board1.clear(board1.size,board1.size);

break;

}

System.out.println(" PRESS n ONLY if you want to play more AND any other character to continue operations");

char out=sc1.next().charAt(0);

if(out=='n')

myTurn=false;

}

return board1;

}

private static boolean checkWinner (Board board){

int counter=0;

boolean flag=false;

for(int i=0;i<board.size;i++){

for(int j=0;j<board.size;j++){

if (board.get(i,j)=='x')

counter++;

if (counter==3)

flag=true;

else

flag=false;

}

}

return flag;

}

/\*

public void showFog (){

//Show board WITHOUT revealing boats (Kanske OPPONENT's game)

}

private boolean checkNear (int i, int j, Board board){

//check box AROUND and return TRUE if there is SHIP around.

System.out.println(" but it was close one");

}

\*/

/\*

{

fire(i,j, board);

if((board[i,j]=="x")||(board[i,j]=="."))

{

System.out.println("Shoot again");

}

else

break;

}

\*/

public static void main(String [] args){

//System.out.print("\033[H\033[2J");

//System.out.flush();

System.out.println("Wellcome to BattleShips Game between Amer and the Opponent Enemy");

System.out.println(" ");

Board board1 = new Board(10);

Board board2 = new Board(10);

Viking viking1= new Viking("Amer");

Viking viking2= new Viking("Enemy");

System.out.println("Now "+ viking1.getName()+ " will place the BOATS");

System.out.println(" ");

board1= initialize();

System.out.println(" ");

System.out.println("Now "+ viking2.getName()+ " will place the BOATS");

System.out.println(" ");

board2= initialize();

boolean destroyed=false;

boolean fired=true;

System.out.println(" Let the tournament begins ");

//for(int i=0;i<3;i++){

while(!destroyed){

System.out.println(" ");

System.out.println(" Now "+ viking1.getName()+ " will take the turn");

fired=true;

System.out.println("The "+ viking2.getName() + " s position against the opponent is BEFORE SHELLING ");

while(fired){

//board2.show();

fired = turn(board2);

destroyed=checkWinner(board2);

}

System.out.println("The "+ viking1.getName() + " s score ON THE OPPONENT's AREA is ");

board2.showFog();

System.out.println(" ");

if(destroyed)

fired=false;

else

fired=true;

System.out.println(" ");

System.out.println(" Now "+ viking2.getName()+ " will take the turn");

while(fired){

//board1.show();

fired = turn(board1);

destroyed=checkWinner(board1);

}

System.out.println("The "+ viking2.getName() + " s position against the opponent is ");

board1.showFog();

}

System.out.println("One of the players destroyed all the opponent's ships");

//}

}

}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%55

public class Canon{

boolean flag=false;

Canon(boolean flag)

{

this.flag=false;

}

public boolean fire(int i,int j,Board board){

if(board.get(i,j)=='o')

{

board.set(i,j,'x');

System.out.println("Kaboom");

flag=true;

}

else

{

board.set(i,j,'.');

System.out.println("Sploosh");

/\* if(board.get(i,j+1)='o')||(board.get(i,j-1)='o')||(board.get(i,j-1)='o')||

(board.get(i-1,j)='o')||(board.get(i+1,j)='o')||(board.get(i+1,j+1)='o')||

(board.get(i+1,j-1)='o')||(board.get(i+1,j-1)='o')||(board.get(i-1,j-1)='o'))

System.out.println("That was close");\*/

}

/\*

checkNear();

\*/

return flag;

}

}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

import java.util.ArrayList;

import java.util.Scanner;

public class Viking{

public String name;

public int score;

Viking(String name){

this.name=name;

}

public String getName(){

return name;

}

}

&&&&&&&&&&&&&&&&&&&&&&&&&6

public class Board{

public int size=10;

public char [][] Grid = new char[10][10];

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Vincents comments

Loop inside the constructor

and with SPACE as a char

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Board(int size){

for(int i=0;i<size;i++)

for(int j=0;j<size;j++)

set(i,j,' ');

this.size=size;

}

public void setSize(int size){

this.size=size;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Vincents comments

change int to Char

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public char get(int i, int j){

return Grid[i][j];

}

public void set(int i, int j, char x){

Grid[i][j]=x;

}

public void show(){

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Vincents comments

Not to mix String with char OUTPUT

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

for(int i=0;i<size;i++){

for(int j=0;j<size;j++){

System.out.print("[");

//System.out.print((char)27 + "[33m");

System.out.print(get(i,j));

System.out.print("]");

}

System.out.println(" ");

System.out.println(" ");

}

}

public void showFog(){

for(int i=0;i<size;i++){

for(int j=0;j<size;j++){

System.out.print("[ ");

if (((get(i,j))=='x')||((get(i,j))=='.'))

System.out.print(get(i,j));

System.out.print("]");

}

System.out.println(" ");

System.out.println(" ");

}

}

public boolean checkFireBefore(){

boolean alreadyFired=false;

int count=0;

for(int i=0;i<size;i++){

for(int j=0;j<size;j++){

System.out.print("[ ");

if (((get(i,j))=='x')||((get(i,j))=='.'))

{

System.out.println("take turn again as the place you fired is already fired");

alreadyFired=true;

count++;

//System.out.print(get(i,j));

}

System.out.print("]");

}

System.out.println(" ");

System.out.println(" ");

}

System.out.println("count is "+ count);

return alreadyFired;

}

public void clear(int i, int j){

for(i=0;i<size;i++)

for(j=0;j<size;j++)

set(i,j,' ');

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Vincents comments

Put space in char ' ' instead of X or a

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

for(i=0;i<size;i++){

for(j=0;j<size;j++){

System.out.print("[");

System.out.print(get(i,j));

System.out.print("]");

}

System.out.println("");

System.out.println("");

}

}

public void placeBoat(int i, int j, char orient){

//System.out.println("HH"+i+"jay "+j);

for(int x=0;x<3;x++){

if(orient=='h')

{

//System.out.print((char)27 + "[33m");

set(i,j++,'o');

}

else

set(i++,j,'o');

}

/\*

if(j>=6)

{

try {

System.out.println("You are crossing the horizental (playing area) limit");

break;

} catch(ArrayIndexOutOfBoundsException e) {

System.out.println("The column number you have entered is invalid");

System.out.println("Please enter a columan number between 1 and 8");

}

}

else {

set(i++,j,'o');

if(i>=6)

try {

System.out.println("You are crossing the vertical (playing area) limit");

break;

} catch(ArrayIndexOutOfBoundsException e) {

System.out.println("The row number you have entered is invalid");

System.out.println("Please enter a row number between 1 and 8");

}

}

\*/

}

}