/\*\*

\* Version 0.1

\*

\*/

package main;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.Random;

import javax.swing.ImageIcon;

import javax.swing.Timer;

public class EnemyShip extends Ship{

private static ImageIcon enemyShip=new ImageIcon("image\\enemy ships\\2ndShipFirstPhase.png");

private int x,y;

private int imgX,imgY;

private int moveY, moveX;

private Random rand=new Random();

private boolean alive=true;

private Timer time;

private boolean FireMode;

private int ticker,limit;

public EnemyShip()

{

super(enemyShip);

imgX = enemyShip.getIconWidth();imgY=enemyShip.getIconHeight();

x=rand.nextInt(600-imgX);y=3;moveY=1;flightPattern(false);

time = new Timer(25,new TimeListener());

limit=400;

time.start();

}

public void setX(int x){this.x=x;}

public void setMoveY(int y){moveY=y;}

public void setMoveX(int x){moveX=x;}

public void setY(int y){this.y=y;}

public void setAlive(boolean flag){alive = flag;}

public void setRandomPosX(){x = rand.nextInt(600-imgX);}

public void setFireModeOn(){FireMode =true;}

public void setFireModeOff(){FireMode =false;}

public void setFireLimit(int i){limit-=i;}

public int getX(){return x;}

public int getY(){return y;}

public int getWitdh(){return imgX;}

public int getMoveY(){return moveY;}

public int getMoveX(){return moveX;}

public boolean getFireMode(){return FireMode;}

public boolean isAlive(){return alive;}

public int getFireLimit(){return limit;}

/\*\*

\* Checks if bullet has made collision with enemy ship

\* @param xB X coordinates of the bullet

\* @param yB Y coordinates of the bullet

\* @return true if there is a collision

\*/

public boolean isColision(int xB, int yB)

{

if((xB+11)>=(x)&&(xB+11)<=(x+imgX)&&(yB-9)<=(y+imgY))

return true;

return false;

}

/\*\*

\* Checks if the enemy spawn ship overlaps with the rest.

\* @param e Other enemy ship for checking against recently spawned ship.

\* @return true if they overlap or flase if they don't.

\*/

public boolean checkIfOverlap(EnemyShip e)

{

if(((e.getX()+imgX)<x||e.getX()>(x+imgX)))

return false;// they are not overlapped

return true;//they are overlapped

}

/\*\*

\* This will set the flight pattern on the X axis.

\* @param suddenChange true if ship should change its pattern due to limit

\* or pre-determined change situation.

\*/

public void flightPattern(boolean suddenChange)

{

if(!suddenChange){

int pattern = 1 + rand.nextInt(6);

moveX = pattern;

}

else

setMoveX(getMoveX()\*(-1));

}

private class TimeListener implements ActionListener{

public void actionPerformed(ActionEvent e) {

setY(getY()+getMoveY());

setX(getX()+getMoveX());

if(((x+enemyShip.getIconWidth())>= 600)||x<=0) flightPattern(true);

//with small problem of abrupt change of flight pattern (to be fixed)

//if(rand.nextInt(400)==ticker) setMoveX(getMoveX()\*(-1));

if(ticker>=limit){ticker=0;setFireModeOn();}

else

ticker++;

}

}

/\*\*

\* Checks if enemy ship has reached end of screen, thus firing "you lose" message

\* for the player.

\* @return true if it has reached end of screen.

\*/

public boolean isGameOver()

{

if((y+enemyShip.getIconHeight())>=600) return true;

return false;

}

}