import java.awt.Polygon;

import java.awt.Rectangle;

/\*\*

\*

\*Description This class is made to create a ship. It will be an object that can move, turn, and respond to player stimulus. It will also interact with asteroids and fire lasers.

@Author Eric Feng

@Version 5.26.2020

\*

\*/

public class SpaceShip extends Polygon {

// Set constants for ship

private double xVelocity = 0, yVelocity = 0;

int heightOfWindow = AsteroidGame.windowHeight;

int widthOfWindow = AsteroidGame.windowWidth;

// Set center for polygon

private double shipX = heightOfWindow / 2, shipY = widthOfWindow / 2;

public static int[] asteroidXValues = { -12, 20, -12, -10, -12 };

public static int[] asteroidYValues = { -15, 7, 15, 7, -15 };

// Set the height and width of ship

private int wOfShip = 27, hOfShip = 30;

private double newXPos = getXShip() + SpaceShip.asteroidXValues[0];

private double newYPos = getYShip() + SpaceShip.asteroidYValues[0];

// Set rotational angle

private double rotatingAngle = 0, shipAngle = 0;

public static boolean interaction = false;

public SpaceShip() {

super(asteroidXValues, asteroidYValues, 5);

}

// Getters and setters for the center are made

/\*\*

\* Description updates the X - values of the ship

\* @return shipX

\*/

public double getXShip() {

return shipX;

}

/\*\*

\* Description updates the Y - values of the ship

\* @return shipX

\*/

public double getYShip() {

return shipY;

}

/\*\*

\* Description sets the value of X of the ship as xShip

\* @return shipX

\*/

public void setXShip(double xShip) {

this.shipX = xShip;

}

/\*\*

\* Description sets the value of Y of the ship as xShip

\* @return shipY

\*/

public void setYShip(double yShip) {

this.shipY = yShip;

}

// updates x position and y position

/\*\*

\* Description Creates a new X position of the ship

\* @return newXpos

\*/

public double getNewXPos() {

return this.newXPos;

}

/\*\*

\* Description creates a new y position of the ship

\* @return newXpos

\*/

public double getNewYPos() {

return this.newYPos;

}

/\*\*

\* Description sets a new X position of the ship

\* @return newXpos

\*/

public void setNewXPos(double nXPos) {

this.newXPos = nXPos;

}

/\*\*

\* Description Updates Y position of the ship

\* @return newXpos

\*/

public void setNewYPos(double nYPos) {

this.newYPos = nYPos;

}

/\*\*

\* Description: Gets the weight of the ship

\* @ return wOfShip

\*/

public int getWofShip() {

return this.wOfShip;

}

/\*\*

\* Description Gets the height of the ship

\* @return hOfShip

\*/

public int getHofShip() {

return this.hOfShip;

}

// updates ship weight and height (if changed at all)

// updates velocity

/\*\*

\* Description Updates the XVelocity

\* @return xVelocity

\*/

public double getXVelocity() {

return this.xVelocity;

}

/\*\*

\* Description Updates the Y velocity of the ship

\* return yVelocity

\*/

public double getYVelocity() {

return this.yVelocity;

}

// Increase and decrease methods for position and velocity

/\*\*

\* Description Updates Y velocity of the ship

\* @return yVelocity

\*/

public void setXVelocity(double xVel) {

this.xVelocity = xVel;

}

/\*\*

\* Description Updates Y velocity of the ship

\* @return yVelocity

\*/

public void setYVelocity(double yVel) {

this.yVelocity = yVel;

}

// updates moving angle

/\*\*

\* Description

\* @param nYPos

\*/

public void setShipAngle(double sAngle) {

this.shipAngle = sAngle;

}

/\*\*

\* Description Updates the ship angle of the ship in question

\* @return shipAngle

\*/

public double getShipAngle() {

return this.shipAngle;

}

/\*\*

\* Description Updates the speed at which the ship is rotating

\* @return rotatingAngle

\*/

public double getRotatingAngle() {

return this.rotatingAngle;

}

// Formula is quantity + the increased amount. quantities include x, y,

// velocity, and angle

/\*\*

\* Description Increases or decreases the X position by velocity

\* @param increaseAmount: The amount the position is increased by

\*/

public void increaseXPosition(double increaseAmount) {

this.shipX += increaseAmount;

}

/\*\*

\* Description Increases or decreases the Y position by velocity

\* @param increaseAmount The amount the position is increased by

\*/

public void increaseYPosition(double increaseAmount) {

this.shipY += increaseAmount;

}

/\*\*

\* Description Increases the x velocity by a certain amount

\* @param increaseAmount The amount the x velocity is increased by

\*/

public void increaseXVelocity(double increaseAmount) {

this.xVelocity += increaseAmount;

}

/\*\*

\* Description Increases the y velocity by a certain amount

\* @param increaseAmount The amount the y velocity is increased by

\*/

public void increaseYVelocity(double increaseAmount) {

this.yVelocity += increaseAmount;

}

/\*\*

\* Description Decreased the x velocity by a certain amount

\* @param decreaseAmount The amount the x velocity is decreased by

\*/

public void decreaseXVelocity(double decreaseAmount) {

this.xVelocity -= decreaseAmount;

}

/\*\*

\* Description Decreased the y velocity by a certain amount

\* @param decreaseAmount The amount the y velocity is decreased by

\*/

public void decreaseYVelocity(double decreaseAmount) {

this.yVelocity -= decreaseAmount;

}

/\*\*

\* Description increases the ship's angle by a certain amount

\* @param sAngle The amount the angle is increased by

\*/

public void increaseShipAngle(double sAngle) {

this.shipAngle += sAngle;

}

public void stopShip() {

this.xVelocity = 0.0;

this.yVelocity = 0.0;

this.shipAngle = rotatingAngle;

// stops the ship

}

/\*\*

\* Descprition: Doubles the x angle change by a certain amount

\* @param xShipAngle: The angle that the ship is currently traveling at (x-value)

\* @return xShipAngle

\*/

public double shipXAngle(double xShipAngle) {

return (double) (Math.cos(xShipAngle \* Math.PI / 180));

// doubles x moving angle with cosine and pi operations

}

/\*\*

\* Description Doubles the y angle change by a certain amount

\* @param shipYAngle The angle that the ship is currently traveling at (y-value)

\* @return yShipAngle

\*/

public double shipYAngle(double yShipAngle) {

return (double) (Math.sin(yShipAngle \* Math.PI / 180));

// doubles y moving angle with sine and pi operations

}

public void increaseRotatingAngle() {

if (getRotatingAngle() >= 355) {

this.rotatingAngle = 0;

} else {

this.rotatingAngle += 5;

}

// function for increasing rotational angle by 5 degrees

}

public void decreaseRotatingAngle() {

if (getRotatingAngle() <= 0) {

this.rotatingAngle = 355;

} else {

this.rotatingAngle -= 5;

}

// function for decreasing rotational angle by 5 degrees

}

/\*\*

\* Description: Gets the new position and orientation of the ship.

\* @return Rectangle

\* @return getXShip

\* @return getYShip

\* @return getWofShip

\* @return getHofShip

\*/

public Rectangle getBounds() {

return new Rectangle((int) getXShip() - 14, (int) getYShip() - 15, getWofShip(), getHofShip());

// updates a bound of a rectangle, factors in center, width, and height.

}

/\*\*

\* Description Increase the x rotational angle by a certain amount

\* @return getXShip

\* @return rotatingAngle

\*/

public double getShipFrontX() {

return this.getXShip() + Math.cos(rotatingAngle) \* 14;

// updates the x-center of the ship's nose

}

/\*\*

\* Description Increase the y rotational angle by a certain amount

\* @return getXShip

\* @return rotatingAngle

\*/

public double getShipFrontY() {

return this.getYShip() + Math.sin(rotatingAngle) \* 14;

// updates the y-center of the ship's nose

}

/\*\*

\* not sure how this works

\*

\*/

public void move() {

// moves x position of the ship via the velocity and the center

// moves y position of the ship via the velocity and the center

this.increaseXPosition(this.getXVelocity());

if (this.getXShip() < 0) {

this.setXShip(widthOfWindow);

} else if (this.getXShip() > widthOfWindow) {

this.setXShip(0);

}

this.increaseYPosition(this.getYVelocity());

if (this.getYShip() < 0) {

this.setYShip(heightOfWindow);

} else if (this.getYShip() > heightOfWindow) {

this.setYShip(0);

}

}

}