Appendix - Player

```
package src;


import java.awt.*;

/*
 * Controlling Entities of ships in the game, or the actual person "Playing" the game.
 *
 */
public class Player extends Actor {

        private final double ACCELERATION = 0.09;

        private boolean thrustingUp;

        private boolean thrustingDown;

        private boolean thrustingLeft;

        private boolean thrustingRight;

        private int frame;

        public Player(Dictator d) {
                super(d, new Position(d.SIZE_X / 2.0, d.SIZE_Y / 2.0), new Movement(0.0,
                                0.0), 10.0);
                speed = 0;
                rotation = 0;
                this.frame = 0;
                radius = 10;

                thrustingUp = false;
                thrustingDown = false;
                thrustingLeft = false;
                thrustingRight = false;
        }

        public void update(Dictator d) {
                super.update(d);
                this.frame++;
                checkAcceleration();
                this.getPosition().add(this.getVelocity());
                rotationization(d.mouse.getPoint());
}
public void rotationization(Point e){
	double firstx = this.getPosition().getX();
	double firsty = this.getPosition().getY();

	double tox = e.getX();
	double toy = e.getY();

	double ydiff = toy-firsty;
	double xdiff = tox-firstx;
	if(xdiff<0){
		super.setRotation(Math.atan(ydiff/xdiff)-Math.PI);
	}else{
		super.setRotation(Math.atan((ydiff/xdiff)));
	}


}

public String toString() {
	String stringy = super.getPosition().toString();

	return stringy;
}

public Position getPosition() {
	return super.getPosition();
}

public int getFrame() {
	return frame;
}

public void thrustingLeft(boolean inc) {
	// TODO Auto-generated method stub
	this.thrustingLeft = inc;

}

public void thrustingRight(boolean inc) {
	// TODO Auto-generated method stub
	this.thrustingRight = inc;

}
```

```
public void thrustingDown(boolean inc) {
	// TODO Auto-generated method stub
	this.thrustingDown = inc;

}

public void thrustingUp(boolean inc) {
	// TODO Auto-generated method stub
	this.thrustingUp = inc;

}

public boolean getThrustingLeft() {
	// TODO Auto-generated method stub
	return thrustingLeft;

}

public boolean getThrustingRight() {
	// TODO Auto-generated method stub
	return thrustingRight;

}

public boolean getThrustingDown() {
	// TODO Auto-generated method stub
	return thrustingDown;

}

public boolean getThrustingUp() {
	// TODO Auto-generated method stub
	return thrustingUp;

}

public void checkAcceleration() {
	if (thrustingUp) {
		this.getVelocity().addY(-ACCELERATION);
	}
	if (thrustingDown) {
		this.getVelocity().addY(ACCELERATION);
	}
	if (thrustingLeft) {
		this.getVelocity().addX(-ACCELERATION);
	}
```

```
        if (thrustingRight) {
            this.getVelocity().addX(ACCELERATION);
        }

        this.getVelocity().scale(.990);

    }

    public void draw(Graphics2D g, Dictator dic) {

        // TODO add check for pause, or thrust, or what ever, i don't really
        // know

        g.setColor(getColor());
        if (true) {
            g.drawLine(-10, -8, 10, 0);
            g.drawLine(-10, 8, 10, 0);
            g.drawLine(-6, -6, -6, 6);
        }
    }

    public void collided(Actor a, Dictator dic) {
        if(a.getClass() == Asteroid.class){
            dic.crash();
        }
    }

    public void center(Dictator d) {
        this.getVelocity().set(0.0,0.0);
        this.getPosition().set(d.SIZE_X/2, d.SIZE_Y/2);

    }

}
```