View Javadoc

```
package model.element.motionless;

import java.awt.Image;
import java.io.IOException;

import javax.imageio.ImageIO;

import contract.ElementType;
import contract.IElement;
import contract.ILevelMap;
import model.element.LevelMap;

/**
 * The Class Diamond.
 *
 * @author Group 5
 */
public class Block implements IElement {

    /** The score */
    private int score = 0;

    /** The exist boolean */
    private boolean exist = true;

    /** The elementType */
    private ElementType elementType = ElementType.BLOCK;

    /** The x */
    private int x;

    /** The y */
    private int y;

    /** The levelmap */
    private ILevelMap levelmap;

    /** The image */
    private static Image image;

    /** The imageName */
    private String imageName = "cobblestone";

    /**
     * constructor to build and place Block
     *
     * @param x
     *              The x.
     * @param y
     *              The y.
     * @param levelMap
     *              The levelMap.
     *
     */
    public Block(final int x, final int y, LevelMap levelMap) {
            this.setX(x);
            this.setY(y);
            this.setImageName(imageName);
            this.loadImage();
            this.setLevelmap(levelMap);

    }

    /**
     * Get x position of Block
     *
     * @return x
     *
     */
    @Override
    public int getX() {

            return this.x;
    }

    /**
     * Set x position of Block
     *
     * @param x
     *              The x.
     *
     */
    @Override
    public void setX(int x) {

            this.x = x;
    }

    /**
     * Get y position of Block
     *
     * @return y
     *
     */
    @Override
    public int getY() {

            return this.y;
    }

    /**
     * Set y position of Block
     *
     * @param y
     *              The y.
     *
     */
    @Override
    public void setY(int y) {

            this.y = y;
    }

    /**
     * Move up the player when up key are pressed
     *
     */
    @Override
    public void moveUp() {

            this.setY(this.getY() - 1);

            this.levelmap.setElement(this.getX(), this.getY(), this);
            this.levelmap.removeElement(getX(), getY()+1);

    }

    /**
     * Move down the player when down key are pressed
     *
     */
    @Override
    public void moveDown() {
            this.setY(this.getY() + 1);

            this.levelmap.setElement(this.getX(), this.getY(), this);
            this.levelmap.removeElement(getX(), getY()-1);
    }

    /**
     * Move left the player when left key are pressed
     *
     */
    @Override
    public void moveLeft() {
            this.setX(this.getX() - 1);

            this.levelmap.setElement(this.getX(), this.getY(), this);
            this.levelmap.removeElement(getX()+1, getY());
    }

    /**
     * Move right the player when right key are pressed
     *
     */
    @Override
    public void moveRight() {
            this.setX(this.getX() + 1);

            this.levelmap.setElement(this.getX(), this.getY(), this);
            this.levelmap.removeElement(getX()-1, getY());
    }

    /**
     * do nothing the player when player don't move
     *
     */
    @Override
    public void doNothing() {
            this.setY(this.getY());

            this.levelmap.setElement(this.getX(), this.getY(), this);
    }

    /**
     * Get image of Block
     *
     * @return image
     */
    @Override
    public Image getImage() {

            return Block.image;
    }
```

```java

        /**
         * Set image of Block
         *
         * @param image
         *                      The image.
         *
         */
        @Override
        public void setImage(Image image) {

                Block.image = image;
        }

        /**
         * Load image of Block
         *
         */
        @Override
        public void loadImage() {

                Image img = null;
                try {
                        img = ImageIO.read(getClass().getClassLoader().getResourceAsStream("images/" + this.getImageName() + ".png"));
                }
                catch(IOException e) {
                        e.printStackTrace();
                }
                this.setImage(img);
        }

        /**
         * Get image name of block
         *
         * @return imageName
         */
        @Override
        public String getImageName() {

                return this.imageName;
        }

        /**
         * Set image name of diamond
         *
         * @param imageName
         *                              The image name.
         *
         */
        @Override
        public void setImageName(String imageName) {

                this.imageName = imageName;
        }

        /**
         * check existing of diamond
         *
         * @return exist
         */
        @Override
        public boolean isExist() {

                return this.exist;
        }

        /**
         * set exist verification of Block
         *
         * @param exist
         *                      The exist state.
         *
         */
        @Override
        public void setExist(boolean exist) {
                this.exist = exist;
        }

        /**
         * Get level
         *
         * @return level map
         *
         */
        public ILevelMap getLevelmap() {
                return levelmap;
        }

        /**
         * Set level
         *
         * @param levelmap
         *                              The levelMap.
```

```java
     *
     */
    public void setLevelmap(ILevelMap levelmap) {
        this.levelmap = levelmap;
    }

    /**
     * Get score of collected diamond
     *
     * @return score
     *
     */
    @Override
    public int getScore() {
        return score;
    }

    /**
     * Set score of collected diamond
     *
     * @param score
     *                       The score.
     *
     */
    @Override
    public void setScore(int score) {
        this.score = score;
    }

    /**
     * Get element type of Block
     *
     * @return element type
     *
     */
    @Override
    public ElementType getElementType() {
        return elementType;
    }

    /**
     * Set element type of diamond
     *
     * @param elementType
     *                               The elementType.
     *
     */
    @Override
    public void setElementType(ElementType elementType) {
        this.elementType = elementType;
    }

}
```