View Javadoc

```
1   package model.element.motionless;
2
3   import java.awt.Image;
4   import java.io.IOException;
5
6   import javax.imageio.ImageIO;
7
8   import contract.ElementType;
9   import contract.IElement;
10  import contract.ILevelMap;
11  import model.element.LevelMap;
12
13  /**
14   * The Class Diamond.
15   *
16   * @author Group 5
17   */
18  public class Block implements IElement {
19
20      /** The score */
21      private int score = 0;
22
23      /** The exist boolean */
24      private boolean exist = true;
25
26      /** The elementType */
27      private ElementType elementType = ElementType.BLOCK;
28
29      /** The x */
30      private int x;
31
32      /** The y */
33      private int y;
34
35      /** The levelmap */
36      private ILevelMap levelmap;
37
38      /** The image */
39      private static Image image;
40
41      /** The imageName */
42      private String imageName = "cobblestone";
43
44      /**
45       * constructor to build and place Block
46       *
47       * @param x
48       *              The x.
49       * @param y
50       *              The y.
51       * @param levelMap
52       *              The levelMap.
53       *
54       */
55      public Block(final int x, final int y, LevelMap levelMap) {
56              this.setX(x);
57              this.setY(y);
58              this.setImageName(imageName);
59              this.loadImage();
60              this.setLevelmap(levelMap);
61
62      }
63
64      /**
65       * Get x position of Block
66       *
67       * @return x
68       *
69       */
70      @Override
71      public int getX() {
72
73              return this.x;
74      }
75
76      /**
77       * Set x position of Block
78       *
79       * @param x
80       *              The x.
81       *
82       */
83      @Override
84      public void setX(int x) {
85
86              this.x = x;
87      }
88
89      /**
90       * Get y position of Block
91       *
```

```
92       * @return y
93       *
94       */
95      @Override
96      public int getY() {
97
98              return this.y;
99      }
100
101     /**
102      * Set y position of Block
103      *
104      * @param y
105      *              The y.
106      *
107      */
108     @Override
109     public void setY(int y) {
110
111             this.y = y;
112     }
113
114     /**
115      * Move up the player when up key are pressed
116      *
117      */
118     @Override
119     public void moveUp() {
120
121             this.setY(this.getY() - 1);
122
123             this.levelmap.setElement(this.getX(), this.getY(), this);
124             this.levelmap.removeElement(getX(), getY()+1);
125
126     }
127
128     /**
129      * Move down the player when down key are pressed
130      *
131      */
132     @Override
133     public void moveDown() {
134             this.setY(this.getY() + 1);
135
136             this.levelmap.setElement(this.getX(), this.getY(), this);
137             this.levelmap.removeElement(getX(), getY()-1);
138     }
139
140     /**
141      * Move left the player when left key are pressed
142      *
143      */
144     @Override
145     public void moveLeft() {
146             this.setX(this.getX() - 1);
147
148             this.levelmap.setElement(this.getX(), this.getY(), this);
149             this.levelmap.removeElement(getX()+1, getY());
150     }
151
152     /**
153      * Move right the player when right key are pressed
154      *
155      */
156     @Override
157     public void moveRight() {
158             this.setX(this.getX() + 1);
159
160             this.levelmap.setElement(this.getX(), this.getY(), this);
161             this.levelmap.removeElement(getX()-1, getY());
162     }
163
164     /**
165      * do nothing the player when player don't move
166      *
167      */
168     @Override
169     public void doNothing() {
170             this.setY(this.getY());
171
172             this.levelmap.setElement(this.getX(), this.getY(), this);
173     }
174
175     /**
176      * Get image of Block
177      *
178      * @return image
179      */
180     @Override
181     public Image getImage() {
182
183             return Block.image;
184     }
```

```
185
186     /**
187      * Set image of Block
188      *
189      * @param image
190      *                      The image.
191      *
192      */
193     @Override
194     public void setImage(Image image) {
195
196             Block.image = image;
197     }
198
199     /**
200      * Load image of Block
201      *
202      */
203     @Override
204     public void loadImage() {
205
206             Image img = null;
207             try {
208                     img = ImageIO.read(getClass().getClassLoader().getResourceAsStream("images/" + this.getImageName() + ".png"));
209             }
210             catch(IOException e) {
211                     e.printStackTrace();
212             }
213             this.setImage(img);
214     }
215
216     /**
217      * Get image name of block
218      *
219      * @return imageName
220      */
221     @Override
222     public String getImageName() {
223
224             return this.imageName;
225     }
226
227     /**
228      * Set image name of diamond
229      *
230      * @param imageName
231      *                              The image name.
232      *
233      */
234     @Override
235     public void setImageName(String imageName) {
236
237             this.imageName = imageName;
238     }
239
240     /**
241      * check existing of diamond
242      *
243      * @return exist
244      */
245     @Override
246     public boolean isExist() {
247
248             return this.exist;
249     }
250
251     /**
252      * set exist verification of Block
253      *
254      * @param exist
255      *                      The exist state.
256      *
257      */
258     @Override
259     public void setExist(boolean exist) {
260             this.exist = exist;
261     }
262
263     /**
264      * Get level
265      *
266      * @return level map
267      *
268      */
269     public ILevelMap getLevelmap() {
270             return levelmap;
271     }
272
273     /**
274      * Set level
275      *
276      * @param levelmap
277      *                              The levelMap.
```

```
278      *
279      */
280     public void setLevelmap(ILevelMap levelmap) {
281             this.levelmap = levelmap;
282     }
283
284     /**
285      * Get score of collected diamond
286      *
287      * @return score
288      *
289      */
290     @Override
291     public int getScore() {
292             return score;
293     }
294
295     /**
296      * Set score of collected diamond
297      *
298      * @param score
299      *                      The score.
300      *
301      */
302     @Override
303     public void setScore(int score) {
304             this.score = score;
305     }
306
307     /**
308      * Get element type of Block
309      *
310      * @return element type
311      *
312      */
313     @Override
314     public ElementType getElementType() {
315             return elementType;
316     }
317
318     /**
319      * Set element type of diamond
320      *
321      * @param elementType
322      *                              The elementType.
323      *
324      */
325     @Override
326     public void setElementType(ElementType elementType) {
327             this.elementType = elementType;
328     }
329
330 }
```