GitHub Stuff\project-5-minewalker-WildestPantaloons\TileButton.java

```java
import java.awt.Point;
import javax.swing.JButton;

/**
 * A custom JButton that knows its row,col coordinates, whether it is a mine,
 * whether it is part of a clear path, its number of mine neighbors, and whether
 * it is hidden or revealed.
 *
 * @author mvail
 */
public class TileButton extends JButton {
    private static final long serialVersionUID = 1L;
    private int row;
    private int column;
    private boolean isPath;
    private boolean isMine;
    private boolean isHidden;
    private int numMineNeighbors;

    /**
     * Initialize a new TileButton with given row and column location.
     * By default, this TileButton is hidden, is not part of the path, is not a
     * mine, and has no mine neighbors.
     *
     * @param row
     * @param column
     */
    public TileButton(int row, int column) {
        this.row = row;
        this.column = column;
        this.isPath = false;
        this.isMine = false;
        this.isHidden = true;
        this.numMineNeighbors = 0;
    }

    /**
     * Return true if this TileButton is part of the guaranteed clear path.
     * @return true if this TileButton is part of the clear path
     */
    public boolean isPath() {
        return isPath;
    }

    /**
     * Update this TileButton to be part of the clear path or not.
     * @param isPath true if part of the path, else false
     */
    public void setPath(boolean isPath) {
        this.isPath = isPath;
    }

    /**
```

```
     * Returns true if this TileButton is a mine, else false.
     * @return true if this TileButton is a mine, else false
     */
    public boolean isMine() {
        return isMine;
    }

    /**
     * Set this TileButton to either be or not be a mine.
     * @param isMine true if a mine, false if not a mine
     */
    public void setMine(boolean isMine) {
        this.isMine = isMine;
    }

    /**
     * Return the number of neigtboring TileButtons that are mines.
     * @return number of neighboring TileButtons that are mines
     */
    public int getNumMineNeighbors() {
        return numMineNeighbors;
    }

    /**
     * Update number of neighboring TileButtons that are mines.
     * @param numMineNeighbors number of neighboring TileButtons that are mines
     */
    public void setNumMineNeighbors(int numMineNeighbors) {
        this.numMineNeighbors = Math.max(numMineNeighbors, 0);
    }

    /**
     * Return true if this TileButton is still hidden, false if it is revealed.
     * @return true if hidden, false if revealed
     */
    public boolean isHidden() {
        return isHidden;
    }

    /**
     * Update this TileButton to reflect whether it is hidden or revealed.
     * @param isHidden true if hidden, false if revealed
     */
    public void setHidden(boolean isHidden) {
        this.isHidden = isHidden;
    }

    /**
     * Return a Point with (x=row, y=column) coordinates corresponding to this
     * TileButton's location in a grid.
     * @return a Point with (x=row, y=column) coordinates
     */
    public Point getLocation() {
        return new Point(row, column);
    }
}
```