Chomp

A board game program with someone playing against the computer.

- **Author:** Rudd Fawcett
- **Course:** Period 6 AP CS, Ms. Litvin
- **Files:**
  - `CharMatrix.java`
- **Due:** 12/11/2015

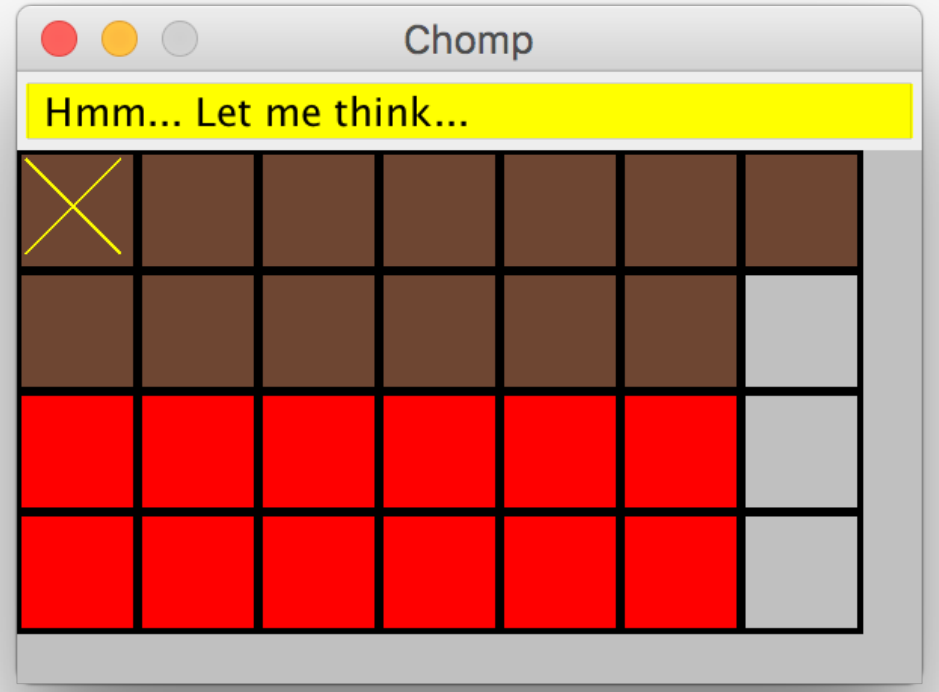
`CharMatrix.java`

```
/**
 * CharMatrix
 * Implements a 2-D array of characters.
 *
 * @author  Rudd Fawcett
 * @course  Period 6 AP CS, Ms. Litvin
 * @due     2015-12-10
 */

public class CharMatrix {
  private char[][] matrix;
  private static final char SPACE = ' ';

  /**
   * Constructor: creates a grid with dimensions rows, cols,
   * and fills it with spaces
   */
  public CharMatrix(int rows, int cols) {
    this(rows, cols, SPACE);
  }

  /**
   * Constructor: creates a grid with dimensions rows , cols ,
   * and fills it with the fill character
   */
  public CharMatrix(int rows, int cols, char fill) {
    this.matrix = new char[rows][cols];
    fillRect(0, 0, rows - 1, cols - 1, fill);
  }
```

```
  /**
   * Returns the number of rows in grid
   */
  public int numRows() {
    return matrix.length;
  }

  /**
   * Returns the number of columns in grid
   */
  public int numCols() {
    return matrix[0].length;
  }

  /**
   * Returns the character at row, col location
   */
  public char charAt(int row, int col) {
    return matrix[row][col];
  }

  /**
   * Sets the character at row, col location to ch
   */
  public void setCharAt(int row, int col, char ch) {
    matrix[row][col] = ch;
  }

  /**
   * Returns true if the character at row, col is a SPACE,
   * false otherwise
   */
  public boolean isEmpty(int row, int col) {
    return matrix[row][col] == SPACE;
  }

  /**
   * Fills the given rectangle with the fill characters.
   * row0, col0 is the upper left corner and row1, col1 is the
   * lower right corner of the rectangle.
   */
  public void fillRect(int row0, int col0, int row1, int col1, c
    for (int row = row0; row <= row1; row++) {
      for (int col = col0; col <= col1; col++) {
        setCharAt(row, col, fill);
```

```
        }
    }
}

/**
 * Fills the given rectangle with the SPACE characters.
 * row0, col0 is the upper left corner and row1, col1 is the
 * lower right corner of the rectangle.
 */
public void clearRect(int row0, int col0, int row1, int col1)
    fillRect(row0, col0, row1, col1, SPACE);
}

/**
 * Returns the count of all non-SPACE characters in row.
 */
public int countInRow(int row) {
    int count = 0;

    for (int cols = 0; cols < numCols(); cols++) {
        if (!isEmpty(row, cols)) {
            count++;
        }
    }

    return count;
}

/**
 * Returns the count of all non-SPACE characters in col.
 */
public int countInCol(int col) {
    int count = 0;

    for (int row = 0; row < matrix.length; row++) {
        if (!isEmpty(row, col)) {
            count++;
        }
    }

    return count;
}
}
```

`chomp.jar`


Chomp
Hmm... Let me think...


Chomp
Hurray, I won!