algorithm.java

```
        /* Surrounding code hidden for readability for part 2c. */
        this.submitButton.addActionListener(e -> {
            switch (this.submitButtonState) {
                case 0:
                    Move move = new Move(this.moveColors);
                    this.responseColors = this.game.submitMove(move);
                    this.moveButtonsEnabled = false;
                    updateMoveButtons();
                    if (this.game.wasWon()) {
                        gameOverWindow = new GameOverWindow(this, game);
                        this.submitButton.setText("New Game");
                        this.submitButtonState = 2;
                    } else {
                        this.submitButton.setText("Next Guess");
                        this.submitButtonState = 1;
                    }
                    break;
                case 1:
                    this.submitButton.setText("Submit");
                    this.responseColors = null;
                    this.moveButtonsEnabled = true;
                    updateMoveButtons();
                    this.submitButtonState = 0;
                    Box box = createPreviousMoveBox(this.game.lastMove);
                    addToHistoryPanel(box);
                    break;
                case 2:
                    if (gameOverWindow != null) gameOverWindow.dispose();
                    newGame();
                    break;
            }
        });
        /* Surrounding code hidden for readability for part 2c. */

        /* Surrounding code hidden for readability for part 2c. */
        private void updateMoveButtons() {
            for (int i = 0; i < 4; i++) {
                this.moveButtons[i].setIcon(this.moveColors[i].icon_large);
                this.moveButtons[i].setEnabled(this.moveButtonsEnabled);
            }
            if (this.responseColors != null) {
                for (int i = 0; i < 4; i++) {
                    if (this.responseColors[i] == null) continue;
                    this.responseButtons[i].setIcon(this.responseColors[i].icon_small);
                }
            } else {
                for (int i = 0; i < 4; i++) {
                    this.responseButtons[i].setIcon(PegIcon.WHITE_ICON_SMALL);
                }
            }
        }
        /* Surrounding code hidden for readability for part 2c. */

        /* Surrounding code hidden for readability for part 2c. */
        private void addToHistoryPanel(JComponent component) {
            component.setAlignmentX(LEFT_ALIGNMENT);
            this.historyPanel.setPreferredSize(new Dimension(520, this.historyPanel.getPreferredSize().height + 100));
            this.historyPanel.add(component);
            this.revalidate();
            this.repaint();
            JScrollBar bar = historyScrollPane.getVerticalScrollBar();
            bar.setValue(bar.getMaximum());
        }
        /* Surrounding code hidden for readability for part 2c. */
```