**Package** src.game

# Class CSudokuGame

java.lang.Object
    src.game.CSudokuGame

---

```
public class CSudokuGame
extends Object
```

A complete Sudoku game implementation with GUI using Java Swing. Features include timer, error tracking, hints system, and solution reveal.

**Since:**

2025-01-01

## *Field Summary*

### Fields

| Modifier and Type | Field | Description |
|---|---|---|
| static final Color | BG_BOARD | |
| static final Color | BG_CELL | |
| static final Color | BG_CELL_GIVEN | |
| static final Color | BG_MAIN | |
| static final Border | correctBorder | |
| static final Border | defaultBorder | |
| boolean | gameOver | True when the game ends due to too many errors |
| Timer | gameTimer | Timer for tracking elapsed time during gameplay |
| boolean | gameWon | True when the player has successfully completed the puzzle |
| static final Color | GRID_BORDER | |
| static final Border | gridBorder | |
| static final Color | MSG_RED | |
| static final Color | MSG_TEXT | |
| static final Color | MSG_WIN | |
| final JPanel | numPanel | Panel containing the number selection pad (buttons 1-9) |

| | | |
|---|---|---|
| static final Color | PAD_BTN_BG | |
| static final Color | PAD_BTN_BG_SEL | |
| static final Color | PAD_BTN_FG | |
| final JButton | revealBtn | Button to toggle solution reveal mode |
| int[][] | revealedBtnNumbers | Stores the user's current entries in each cell of the puzzle |
| int | secondsElapsed | Total seconds elapsed since game start |
| static final Border | selectedBorder | |
| static final Color | TEXT_GIVEN | |
| static final Color | TEXT_USER | |
| static final Color | TITLE_BORDER | |
| static final Border | titleBorder | |
| static final Border | wrongBorder | |

## Constructor Summary

### Constructors

| Constructor | Description |
|---|---|
| CSudokuGame() | Constructs a CSudokuGame instance and initializes the GUI. |

## Method Summary

All Methods | Static Methods | Instance Methods | Concrete Methods

| Modifier and Type | Method | Description |
|---|---|---|
| void | checkMove(int row, int col, int value) | Verifies if the user's number placement is correct. |
| boolean | checkWinCondition() | Checks if the player has successfully completed the puzzle. |
| void | disableAllButtons() | Disables all interactive buttons and grid cells. |
| void | flashCell(JButton cell, Color flashColor, int flashes) | Animates a cell by flashing it between its original color and a specified color. |

| | | |
|---|---|---|
| int | getCurrentNumber() | Gets the number currently selected by the player. |
| void | giveHint(JButton hintBtn) | Provides a hint to the player by filling in one randomly-selected empty cell with its correct answer. |
| void | hideSolution() | Hides the solution and restores the puzzle to its previous state. |
| void | loadNearlyCompletePuzzle() | Loads a nearly-complete puzzle configuration for debugging. |
| static void | main(String [] args) | Main entry point for the Sudoku game application. |
| static void | printStatement(int[][] intList, boolean[][] boolList) | Debug helper method to print the puzzle grid to console. |
| void | reEnableAllButtons() | Re-enables interactive buttons and user-editable grid cells. |
| String | requestUserName() | Prompts the user to enter their name via a dialog. |
| void | resetColor(JPanel panel, Color color) | Resets the background color of all buttons in a panel. |
| void | resetGame() | Resets the entire game to its initial state. |
| void | revealSolution() | Reveals the complete solution by filling all cells with correct answers. |
| void | setPlayerName(String name) | Sets the player's name and updates the game title label. |
| void | setSelectedCell(SButton btn) | Sets the currently selected grid cell and updates its visual state. |
| void | setSelectedNumberButton (SButton btn) | Sets the currently selected number button and updates its visual highlighting. |
| void | showGameEndDialog(String message, String title) | Displays a game end dialog with \"Start Over\" and \"Exit\" options. |

## Methods inherited from class java.lang.Object

clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

## *Field Details*

### secondsElapsed

```
public int secondsElapsed
```

Total seconds elapsed since game start

### gameTimer

```
public Timer  gameTimer
```

Timer for tracking elapsed time during gameplay

### gameWon

```
public boolean gameWon
```

True when the player has successfully completed the puzzle

### gameOver

```
public boolean gameOver
```

True when the game ends due to too many errors

### revealBtn

```
public final JButton  revealBtn
```

Button to toggle solution reveal mode

### numPanel

```
public final JPanel  numPanel
```

Panel containing the number selection pad (buttons 1-9)

### BG_MAIN

```
public static final Color  BG_MAIN
```

### BG_BOARD

```
public static final Color  BG_BOARD
```

### BG_CELL

```
public static final Color  BG_CELL
```

### BG_CELL_GIVEN

```
public static final Color  BG_CELL_GIVEN
```

### GRID_BORDER

```
public static final Color  GRID_BORDER
```

### PAD_BTN_BG

```
public static final Color  PAD_BTN_BG
```

### PAD_BTN_BG_SEL

```
public static final Color  PAD_BTN_BG_SEL
```

### PAD_BTN_FG

```
public static final Color  PAD_BTN_FG
```

### TEXT_GIVEN

```
public static final Color  TEXT_GIVEN
```

### TEXT_USER

```
public static final Color  TEXT_USER
```

### MSG_TEXT

```
public static final Color  MSG_TEXT
```

### MSG_WIN

```
public static final Color  MSG_WIN
```

### MSG_RED

```
public static final Color  MSG_RED
```

### TITLE_BORDER

```
public static final Color  TITLE_BORDER
```

### correctBorder

```
public static final Border  correctBorder
```

### wrongBorder

```
public static final Border  wrongBorder
```

### selectedBorder

```
public static final Border  selectedBorder
```

### defaultBorder

```
public static final Border  defaultBorder
```

### gridBorder

```
public static final Border  gridBorder
```

### titleBorder

```
public static final Border  titleBorder
```

### revealedBtnNumbers

```
public int[][] revealedBtnNumbers
```

Stores the user's current entries in each cell of the puzzle

## *Constructor Details*

### CSudokuGame

```
public CSudokuGame()
```

Constructs a CSudokuGame instance and initializes the GUI. Sets up the game board, number pad, control buttons, and starts the timer. Also loads audio resources needed for gameplay.

## *Method Details*

### revealSolution

```
public void revealSolution()
```

Reveals the complete solution by filling all cells with correct answers. Used when the "Reveal Solution" button is toggled on.

### hideSolution

```
public void hideSolution()
```

Hides the solution and restores the puzzle to its previous state. Pre-filled cells show their original values; user cells show user inputs or remain empty.

### loadNearlyCompletePuzzle

```
public void loadNearlyCompletePuzzle()
```

Loads a nearly-complete puzzle configuration for debugging. Fills most cells with correct answers, leaving only a few empty cells.

### getCurrentNumber

```
public int getCurrentNumber()
```

Gets the number currently selected by the player.

**Returns:**

the selected number (1-9), or 0 if no number is selected

### setSelectedNumberButton

```
public void setSelectedNumberButton(SButton btn)
```

Sets the currently selected number button and updates its visual highlighting.

**Parameters:**

`btn` - the button to select (or null to deselect all)

### setSelectedCell

```
public void setSelectedCell(SButton btn)
```

Sets the currently selected grid cell and updates its visual state. Removes highlighting from the previous cell and applies it to the new one.

**Parameters:**

`btn` - the cell button to select

### checkMove

```
public void checkMove(int row,
                      int col,
                      int value)
```

Verifies if the user's number placement is correct. Updates cell border colors (green for correct, red for incorrect), plays appropriate sounds, and increments error count if needed.

**Parameters:**

`row` - the row index of the cell (0-8)

`col` - the column index of the cell (0-8)

`value` - the number the user entered

### disableAllButtons

```
public void disableAllButtons()
```

Disables all interactive buttons and grid cells. Used when the game is won or lost to prevent further input.

### reEnableAllButtons

```
public void reEnableAllButtons()
```

Re-enables interactive buttons and user-editable grid cells. Pre-filled cells remain disabled. Used when exiting reveal mode or after certain game states.

### checkWinCondition

```
public boolean checkWinCondition()
```

Checks if the player has successfully completed the puzzle. Compares all cells against the solution grid.

**Returns:**

true if all cells match the solution, false otherwise

### giveHint

```
public void giveHint(JButton  hintBtn)
```

Provides a hint to the player by filling in one randomly-selected empty cell with its correct answer. Decrements hint count, adds 30-second time penalty, and displays animation on the hinted cell.

**Parameters:**

`hintBtn` - the hint button (disabled when no hints remain)

### showGameEndDialog

```
public void showGameEndDialog(String  message,
                              String  title)
```

Displays a game end dialog with \"Start Over\" and \"Exit\" options. Handles user choice to restart the game or exit the application.

**Parameters:**

`message` - the message to display in the dialog

`title` - the title of the dialog window

### resetColor

```
public void resetColor(JPanel  panel,
                       Color  color)
```

Resets the background color of all buttons in a panel. Used to remove highlighting from number buttons.

**Parameters:**

`panel` - the panel containing buttons to reset

`color` - the color to reset all buttons to

### resetGame

```
public void resetGame()
```

Resets the entire game to its initial state. Clears all user entries, resets timers, error counts, hints, and re-enables all controls. Called when the player selects \"Start Over\" after winning or losing.

### flashCell

```
public void flashCell(JButton  cell,
                      Color  flashColor,
                      int flashes)
```

Animates a cell by flashing it between its original color and a specified color. Used to draw attention to special events like hints or correct answers.

**Parameters:**

`cell` - the button/cell to flash

`flashColor` - the color to flash to

`flashes` - the number of times to flash (alternates between original and flash color)

### printStatement

```
public static void printStatement(int[][] intList,
                                  boolean[][] boolList)
```

Debug helper method to print the puzzle grid to console. Shows given cells as numbers and empty cells as dots.

**Parameters:**

`intList` - the solution array containing all numbers

`boolList` - the configuration array indicating which cells are given

### requestUserName

```
public String  requestUserName()
```

Prompts the user to enter their name via a dialog. Used to personalize the game title.

**Returns:**

the entered name, or \"Chibueze\" as default if cancelled

### setPlayerName

```
public void setPlayerName(String  name)
```

Sets the player's name and updates the game title label.

**Parameters:**

`name` - the player's name to display

### main

```
public static void main(String [] args)
```

Main entry point for the Sudoku game application. Creates the game instance and prompts for player name.

**Parameters:**

`args` - command-line arguments (unused)