

# Sudoku-5: Game

CSCI 4526 / 6626 Fall 2016

## 1 Goals

- To create a simple text-based interface for Sudoku Helper
- To provide an interactive way to test Board and the other classes.

## 2 The Game Class

Instantiate this controller class from main.

- At this time, there is only one data member, a Board. Later we will add Stacks for undo and redo and streams for save and restore.
- The Game constructor should call the Board constructor.
- Define a `run()` function for Game. In this function, write a menu-loop that will ask the user what to do next. End the menu loop when the user selects *Quit* or when the puzzle is completely solved (*isDone()*).

### 2.1 The Menu

We will implement two menu entries this week, more later. Use the `menu_c()` function in tools to implement the menu. To use this function, you must supply a title, the number of menu choices, an array of `const char*` for the options, and a `const char*` c-string that lists the first letter of each legal choice. This string will be used by `menu_c` to validate user selections. The menu should display these options:

- Move: Input row, column, and value, then call `Board::move()`.
- Undo
- Redo
- Save game
- Restore game
- Quit and discard game: End and print a message.

## 3 Changes to the Board Class

Board needs to provide a way for Game to find out when the puzzle is solved.

- Add an integer counter to count the number of squares that have non-dash values. Initially 0, this counter will be incremented in the Board constructor and in `Board::move()` whenever a value is stored in a square.
- Add an *isDone()* function that will return *true* when all 81 squares have values, *false* otherwise.