SKYSCRAPER / TOWERS GAME

PLANNING

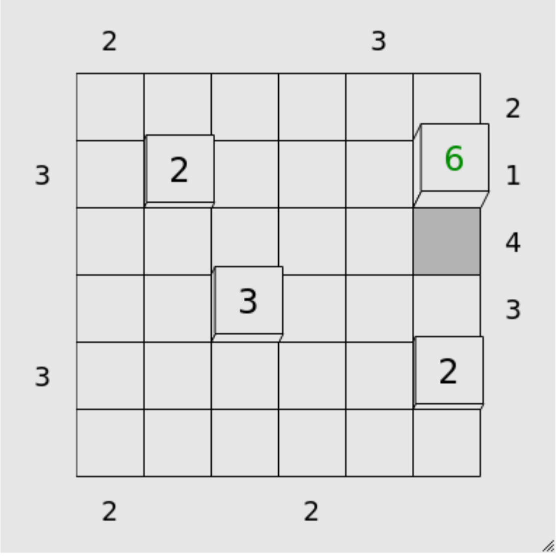
Description of Game

This is a logic game that I found on my Android (Simon Tatham's Tower puzzle), which I am having trouble winning. I'd like to write a Java program to make it easier to figure out the logic of how to win.

Rules:

- this version is on a 6x6 grid (5x5 is too easy to win)
- The numbers 1-6 must be placed in each row and column.
  There can be no duplicate numbers in a row or column **(rule 1)**
- These numbers denote the height of a tower or skyscraper in that square.
- Along the edges of the board are some numbers that indicate how many skyscrapers you can see if you look along that row or column (see diagram below)
  Example: [5][1][3][6][4][2][1]
  - from the left, you would see two (the 5 and the 6)
  - from the right you would see four (the 1,2,4,6).
  - The edge numbers would be: 2 [ ][ ][ ][ ][ ][ ][ ]4
- Any towers placed must ensure that the edge numbers are correct (ie. if it is a 2, then you cannot see any other number of towers). **(rule 2)**

Diagram of board :



From https://www.chiark.greenend.org.uk/~sgtatham/puzzles/js/towers.html

User Interface

There will be an opening screen with instructions. There will also be a text box on the right which has the logic tips that I've worked out so far on how to solve these puzzles.

You will be able to select a particular game (board layout to solve) numbered 1 to 10, or get a random one.
There will be at least 10 games, most of which will be “hard” level, but maybe a few “extreme” as well. I don’t know how to generate solvable games so I’ll just be loading games that I’ve copied from online.

The user will click on the square to select it and then also click on a number in order to fill in the tower for that square. Right clicking on a number can change the “pencil marks” in that square.

The game will tell you if you have won. It will also show you if you make an error when you place a tower: if you violate rule 1 or 2, then your move will be shown in red.

There will be a button for “give up” as well as one for “play a new game”.

Program Planning

The program will be written using Swing (instead of HSA2). (What?? That means no gameloop ...)

- Global variables:
    - board (a 6x6 array of ints)
    - edges (a 4x6 array of ints to hold the edge numbers for a certain puzzle)
    - int gameNum : which game you are playing
    - some way to keeping track of pencil marks for each square.
      I think I'll do it like this:
      int[][]pencilMarks = new int[6][6]
      Each one of these squares will contain an integer that represents the pencil marks:
      eg. pencilMarks[1][2] = 12346; //so all pencil marks except for 5 are in that square.
      This will make it more complicated to retrieve and change the pencil marks, but I think it can be done without too much trouble.
    - int sx, sy : the x,y of the square that is selected. If no square is selected, these are set to -1

- Constants:
    - Screen width and height
    - gridSize

- Methods:
  Here I am thinking about how to break the program up into methods and what each of them will do. One of the harder things is to figure out the main methods that control the others.
  You will also want to use only global variables for things that are absolutely necessary, so some methods may have parameters and return values to transfer data.

    - void loadGame(int n) : this will load a particular game from a text file. If n<1, it will pick a random game). I don't know how to generate Tower games, so I'll just be copying the ones that I can't solve from my Android. This method is called from newGame().
    - various graphics methods that are called from the main graphics method
        - drawBoard() //this draws the board, the grid, the buttons 1-6 to click on.
        - drawTowers()
        - drawEdges()
        - drawSelected()
        - drawPencilMarks()
    - void mouseClicked() : this method will either select a square, place a number or change the pencil mark, depending on what exactly is clicked and with which mouse button.
    - boolean selectSquare() : select a square on the board
      it returns false if the square has a fixed number in it (from when the game was loaded).
    - void placeNumber(int n) : place number n in the selected square
    - removePencilMark()
    - boolean checkMove() : This checks the latest move and sees if it causes an error by violating rule 1 or rule 2. (no duplicate rule, and view from side rule).
      Somehow, if the move is invalid, the drawTowers() method needs to draw your move in red.

- pencilMarkAutoRemove() : this will automatically remove the appropriate pencil marks when you place a tower.
- boolean gameOver() : this checks to see if the game is over: are all squares filled? Are rules 1 and 2 met for all squares?
- void restartGame() : restart the current game
- void newGame() : this resets everything and loads a new game (asking the user which # game they want).