Cedar Wood 03/08/24

# Satogaeri Puzzle Generator

---

## Setup Directions:

Compile and run the main file, SatogaeriGen.java; once running, follow the following prompts:

- ***Run program in debug mode? y/n***

  If the user enters ‘y’, the program will print preliminary board attempts for each iteration of the board; this may increase the run time

- ***Input board dimensions, separated by a comma and space:***

  Here, dimensions should be entered; larger sizes may increase the run time. Recommended sizes:
  - 7, 9
  - 8, 10
  - 10, 15

- ***Select difficulty (1-3):***

  Enter an integer between 1 and 3 – 1 being easiest and 3 being most difficult. This difficulty increases the maximum distance a circle may be required to move and adjusts the calculations for factors like *numRegions* and *perRegion.*

*After the prompts have been finished, the puzzle will generate and output in this format (example):*

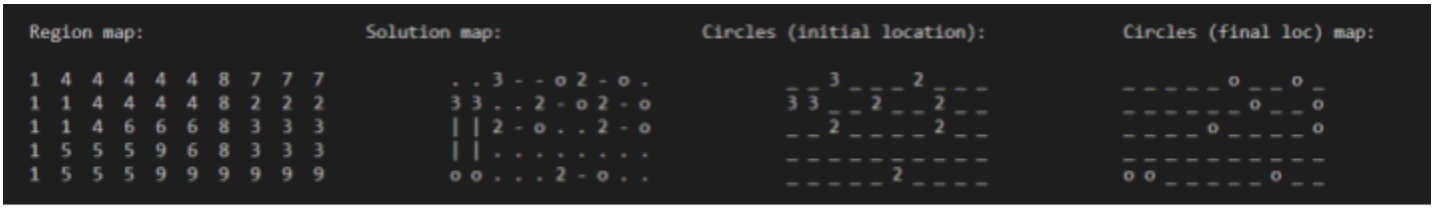
```
Run program in debug mode? y/n
(preliminary boards will be printed as they are generated, this may increase runtime))
n
Input board dimensions, seperated by a comma and space:
- Eg: 7, 9
7, 9
Select difficulty  (1-3):

2
board of: 7 by 9
Generating ...


Region map:                   Solution map:               Circles (initial location):          Circles (final loc) map:

9  9  9  9  9  9  9  9  9            4 . 3 . . 2 - o .           4 _ 3 _ _ 2 _ _ _                _ _ _ _ _ _ _ o _
9  9  5  5  5  5  5  5  5            | . | . 3 - - o .           _ _ _ _ 3 _ _ _ _                _ _ _ _ _ _ _ o _
9  9  9  5  5  5  5  5  5            | 2 | . . . . . .           _ 2 _ _ _ _ _ _ _                _ _ _ _ _ _ _ _ _
2  1  6  6  6  4  10 10 10           | | o 3 - - o . 3           _ _ _ 3 _ _ _ _ 3                _ _ o _ _ _ o _ _
2  1  1  1  8  4  4  10 10           o o 4 - - - o . |           _ _ 4 _ _ _ _ _ _                o o _ _ _ _ _ o _ _
2  11 1  1  8  4  4  7  3            . 3 - - o 2 - o |           _ 3 _ _ _ 2 _ _ _                _ _ _ _ o _ _ o _
2  11 11 11 8  8  8  7  3            2 - o . . . . . o           2 _ _ _ _ _ _ _ _                _ _ o _ _ _ _ _ o

Previous file deleted: newPuzzle.txt
File created: newPuzzle.txt
Saved puzzle to newPuzzle.txt
```




**Region Map:** outputs the puzzle regions – each notated with a number for their specified region.

***Solution Map:** outputs the puzzle's solution by moving the circles to their solution location (numbers represent the initial circle, while an 'o' is that circles' final location )

**Circles Initial**: outputs the puzzle's circle locations and their numeric value (the number of spaces they must be moved)

**Circles Final:** outputs the puzzle's circle locations after they have been moved into their respective final regions.

** The Solution Map mimics the format outputted by the solver, Satogaeri-solver.py by showing the horizontal or vertical movement path of each circle. Solution Map is a combination of Circles Initial/Final and their connecting (linear) paths.*

---

## Difficulty Factors:

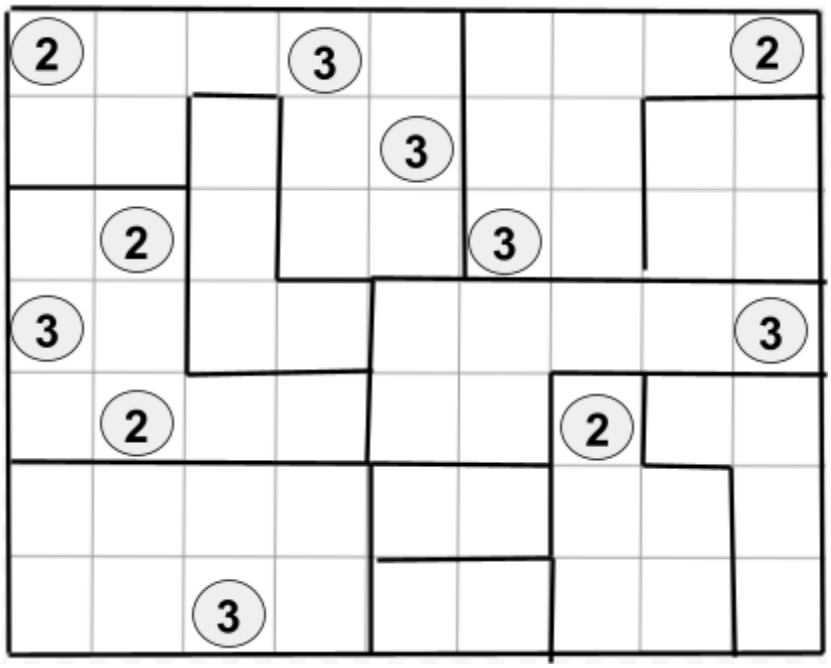
While size is an obvious factor in increasing difficulty (since the board will automatically generate more regions and circles for larger boards) I decided to leave the size completely up to the user and use other factors for my adjustable measures of difficulty:

- **MAX_CIRCLE_VAL/MIN_CIRCLE_VA**: these variables affect the range of acceptable values for circles (ie: how far a circle is expected to be moved for the solution). By allowing larger MAXes at higher difficulties, there are further distances required to move each circle – which can increase visual confusion. In addition, longer distances create more challenge when unable to overlap lines.
- **numRegions/perRegion**: these variables were adjusted as a combination of board size and the previously mentioned circle values. Increasing the number of regions creates more areas of the board that need a circle to satisfy them – ie more puzzle complexity and difficulty.

## Puzzle Examples:

**Puzzle1.txt (Easy)**

```
9 7 11
2 . . 3 . . . . 2
. . . . 3 . . . .
. 2 . . . 3 . . .
3 . . . . . . . 3
. 2 . . . . 2 . .
. . . . . . . . .
. . 3 . . . . . .
1 1 1 1 1 11 11 11 11
1 1 10 1 1 11 11 8 8
2 2 10 1 1 11 11 8 8
2 2 10 10 5 5 5 5 5
2 2 2 2 5 5 7 6 6
3 3 3 3 9 9 7 7 6
3 3 3 3 4 4 7 7 6
```




**Puzzle2.txt (Medium)**

```
8 10 15
. 3 . . 2 . 2 3
. . 3 2 . 2 . .
2 . . . . . . .
. . . . . . . .
. 2 . . . . . .
. . 2 . . . . .
. . . . 3 . . .
2 3 . . . . . .
. . . 3 . . . .
. . 4 . . . . .
15 15 15 15 15 15 15 15
15 11 11 11 11 3 3 3
15 15 11 11 11 4 3 3
15 8 8 2 2 4 4 7
15 6 6 2 2 2 4 7
15 1 1 14 14 14 14 7
12 1 10 10 14 13 13 5
12 12 10 10 10 13 13 5
12 12 10 10 10 13 13 5
12 12 12 12 9 9 9 9
```