11. Sudoku Helper: Sixy Sudoku

CSCI 4526 / 6626 Fall 2022

1 Goals

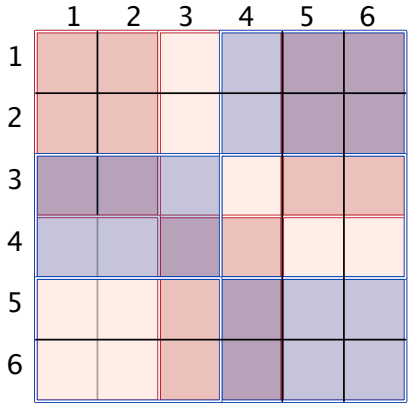
- To and another variant of Sudoku to the game.
- To make sure that everything in parts 1–10 can handle a different size board.

2 Classes Board and Cluster.

Very little in parts 1–5 of this project should change except the Board constructor and lines that call a Board constructor. However, it now becomes essential that N and N^2 are used consistently everywhere – not 9 and 81.

The SixyBoard class

- You already have one class derived from Board (DiagBoard) Now derive two more: TradBoard and SixyBoard, and change DiagBoard to be derived from TradBoard. Now Board becomes a true polymorphic class that is not complete by itself, and all actual board objects will be allocated
- The base class constructor should do all parts of construction up to building the row and column Clusters. Move the Box construction to the constructor of TradBoard. Add new code to the Sixy constructor to create Clusters for the six VBoxes and the six Hboxes. Note that now each Square in the board has *four* constraints, not three.



HBoxes: lilac

hb1: [1,1] ..[2,3]
hb2: [1,4] ..[2,6]
hb3: [3,1] ..[4,3]
hb4: [3,4] ..[4,6]
hb5: [5,1] ..[6,3]
hb6: [5,4] ..[6,6]

VBoxes: salmon

vb1: [1,1] ..[3,2]
vb2: [1,3] ..[3,4]
vb3: [1,5] ..[3,6]
vb4: [4,1] ..[6,2]
vb5: [4,3] ..[6,4]
vb6: [4,5] ..[6,6]

- Except for the graphic interface, the Game should now operate correctly for the Sixy version. Because time is short, you don't need to adapt the graphic interface. You can still play a sixy game from the list of clusters that is displayed after every move.

3 Submission.

Submit the source code and proof that it works in the form of a text file that contains a copy of your screen output. I will eventually compile the program and test this version.