Slides for the Week

CS273 Laboratory 7

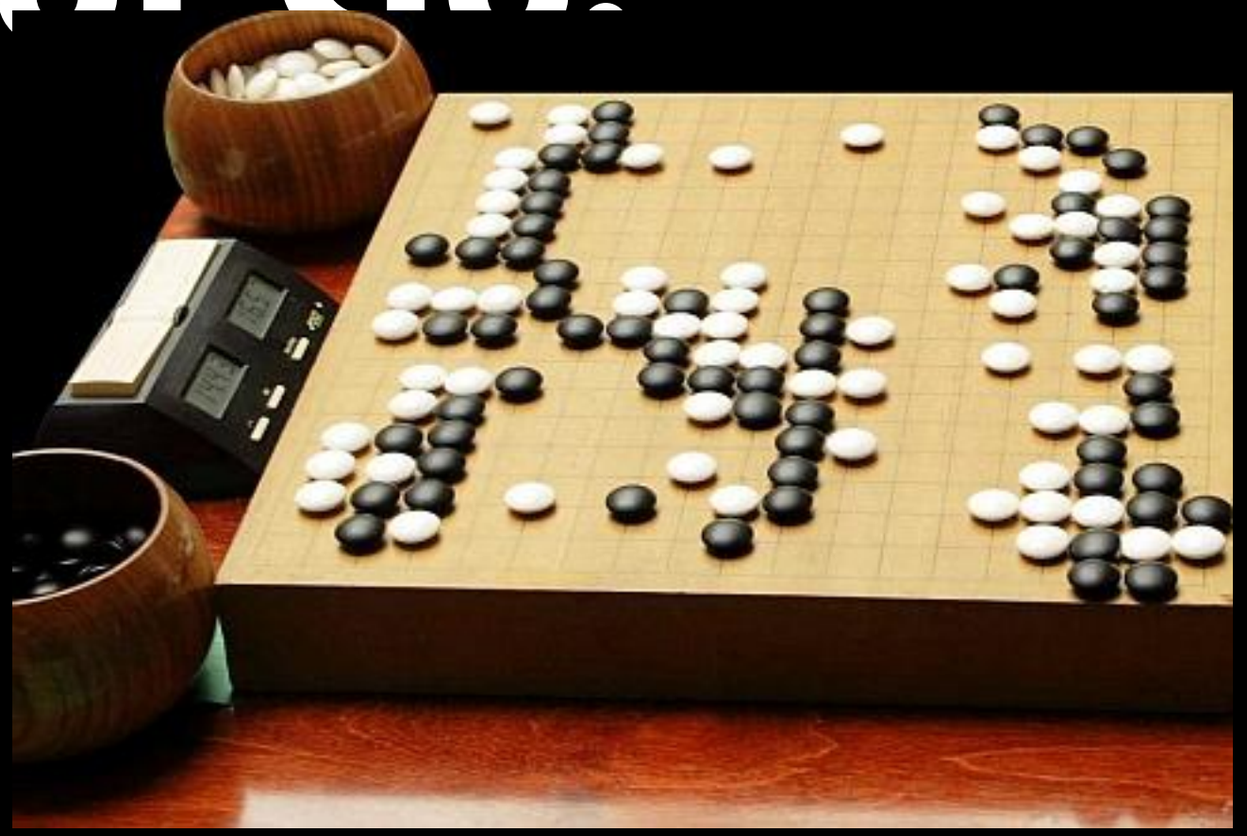
This week's lab focuses on 2-dimensional arrays

You'll be finishing the work you started last time with lab7.zip.

You'll be finishing the work you started last time with lab7.zip.

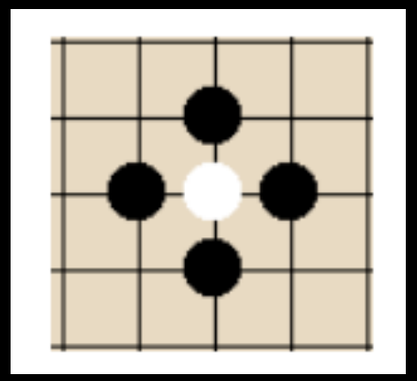
Don't download a new zipfile.
Just work on the same code from last time.

In this lab, you will finish implementing a game of Go.

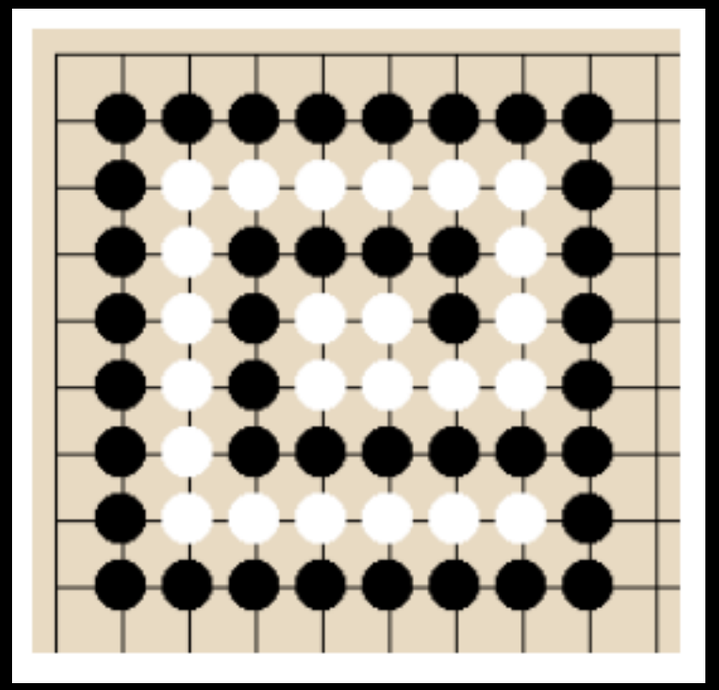


http://gogameguru.com/what-is-go/

Go is a game in which players attempt to capture each other's stones by surrounding each other.



Simple case: Black captures white!



Difficult case: Black captures white!

The board is stored as a two-dimensional array where each element in the array has one of these five values.

```
public static final int WHITE = 0;
public static final int BLACK = 1;
public static final int EMPTY = 2;
public static final int WHITEINPERIL = 3;
public static final int BLACKINPERIL = 4;
```

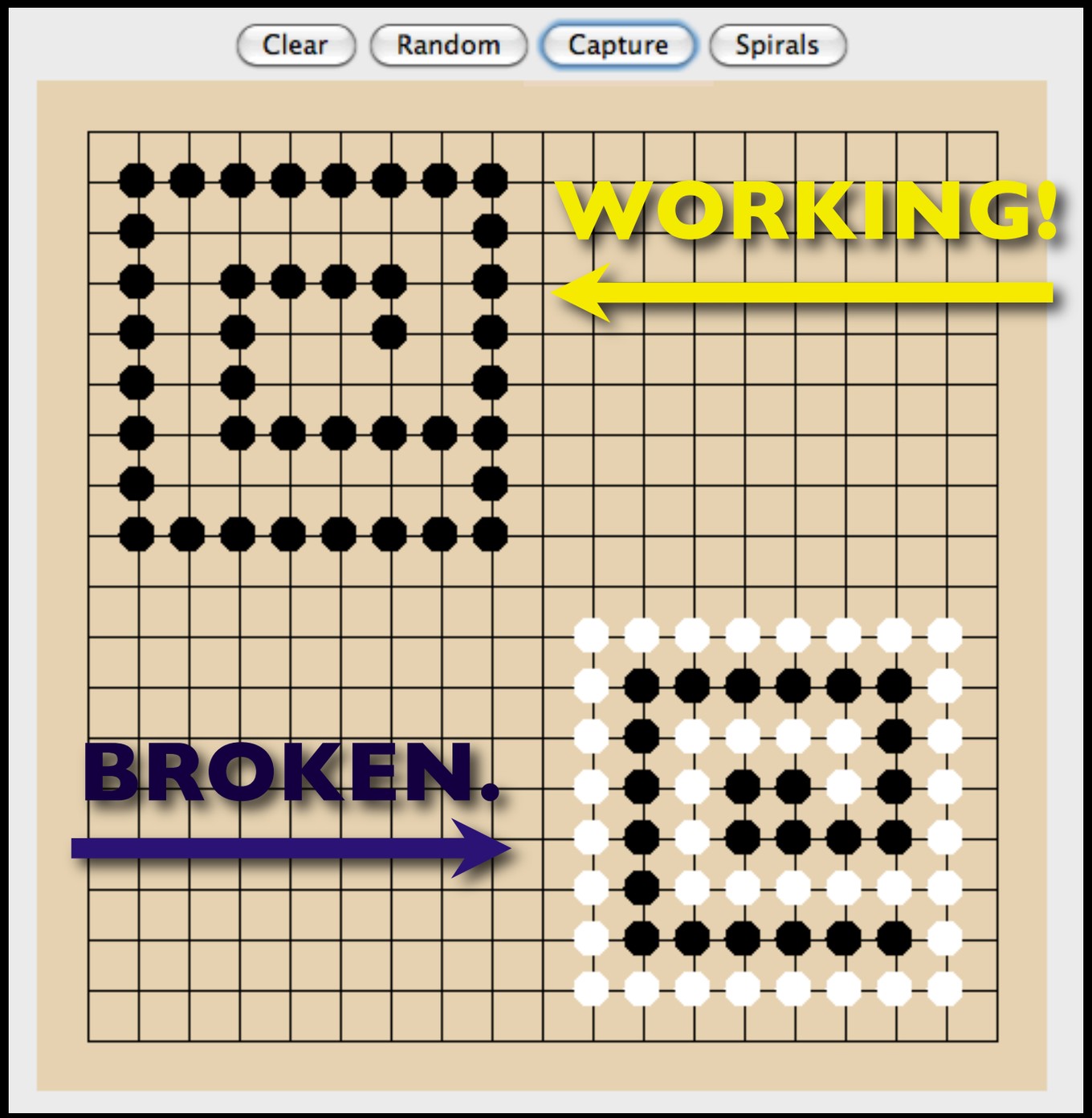
It is strongly recommended that you use the symbolic values and not the integer values. It makes coding easier.

Checkpoint 1 asks you to place a random assortment of stones on the board.

Note that the `Math.random()` method does not work like other functions do.

It gives a **different** number **every** time you call it.

After you complete the remaining checkpoints, the user may press the "Capture" button and the game must be able to detect when pieces are captured and remove them from the board.



The lab will take you through the complex algorithm for doing this step by step:

Mark every stone as "in peril"
Remove "in peril" from every stone that is adjacent to empty or non-"in-peril" stone
If any changes were made in step 2, then goto step 2.
Remove all stones that are still in peril

Step 2 is fairly involved.

Mark every stone as "in peril"
Remove "in peril" from every stone that is adjacent to empty or non-"in-peril" stone
If any changes were made in step 2, then goto step 2.
Remove all stones that are still in peril

For every stone on the board, you must determine if it is next to an empty space or a space of its color.

Mark every stone as "in peril"
Remove "in peril" from every stone that is adjacent to empty or non-"in-peril" stone
If any changes were made in step 2, then goto step 2.
Remove all stones that are still in peril

One approach that can help simplify coding is the use of a `method`.

We recommend that you write a method called `isAdjacencyOkay()`

Given an row position, a column position, and a color, this method returns true if the position on the board is next to an empty space or a space of the same color.

It returns false otherwise.

If you write such a method, you could use it like this:

```
for(int i = 0; i <= ROWS; i++)
        {
            for(int j = 0; j <= COLS; j++)
            {
                if (board[i][j] == WHITEINPERIL)
                {

                    if (isAdjacencyOkay(i, j, WHITE))
                    {
                        board[i][j] = WHITE;
                        changed = true;
                    }
                }


                else if (board[i][j] == BLACKINPERIL)
                {

                    if (isAdjacencyOkay(i, j, BLACK))
                    {
                        board[i][j] = BLACK;
                        changed = true;
                    }
                }
            }
        }
```

You could implement it like this:

```
////////////////////////////////////////////////////////////////////
 // isAdjacentOkay method - Determine if a stone's surroundings
 // make it a safe stone.
 //
 // Given a row, a column, and a color, examine the nine adjacent
 // spaces surrounding this position.  Return true if the position
 // is next to an empty space or a stone of the same color.  Return
 // false otherwise.
 //
 // BEWARE: Do not check beyond the board's boundaries.  For example,
 // if this function is checking a position in row 0, it should not
 // check the row "above" the position.
protected boolean isAdjacencyOkay(int row, int col, int color)
{


    // You implement this part.


}
```

Without a method,
your code might look
like this:

```
if(board[x][y] == WHITEEMPERIL)
        {
            if(x > 0 && x < board.length-1 && (y > 0 && y < board[x].length-1))
            {
                if(board[x+1][y] == WHITE || board[x-1][y] == WHITE || board[x][y+1] == WHITE || board[x][y-1] == WHITE)

                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x+1][y] == EMPTY || board[x-1][y] == EMPTY || board[x][y+1] == EMPTY || board[x][y-1] == EMPTY)

                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(x == 0 && y == 0)
            {
                if(board[x+1][y] == WHITE || board[x][y+1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x+1][y] == EMPTY || board[x][y+1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(x == board.length-1 && y == board[x].length-1)
            {
                if(board[x-1][y] == WHITE || board[x][y-1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x-1][y] == EMPTY || board[x][y-1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(x == 0 && y == board[x].length-1)
            {
                if(board[x+1][y] == WHITE || board[x][y-1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x+1][y] == EMPTY || board[x][y-1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(x == board.length-1 && y == 0)
            {
                if(board[x-1][y] == WHITE || board[x][y+1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x-1][y] == EMPTY || board[x][y+1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(x == 0)
            {
                if(board[x+1][y] == WHITE || board[x][y+1] == WHITE || board[x][y-1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x+1][y] == EMPTY || board[x][y+1] == EMPTY || board[x][y-1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(y == 0)
            {
                if(board[x+1][y] == WHITE || board[x-1][y] == WHITE || board[x][y+1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x+1][y] == EMPTY || board[x-1][y] == EMPTY || board[x][y+1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(x == board.length-1)
            {
                if(board[x-1][y] == WHITE || board[x][y+1] == WHITE || board[x][y-1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x-1][y] == EMPTY || board[x][y+1] == EMPTY || board[x][y-1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(y == board[x].length-1)
            {
                if(board[x+1][y] == WHITE || board[x-1][y] == WHITE || board[x][y-1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x+1][y] == EMPTY || board[x-1][y] == EMPTY || board[x][y-1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
    }
```

```
if(board[x][y] == WHITEEMPERIL)
        {
            if(x > 0 && x < board.length-1) && (y > 0 && y < board[x].length-1))
            {
                if(board[x+1][y] == WHITE || board[x-1][y] == WHITE || board[x][y+1] == WHITE || board[x][y-1] == WHITE)

                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x+1][y] == EMPTY || board[x-1][y] == EMPTY || board[x][y+1] == EMPTY || board[x][y-1] == EMPTY)

                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(x == 0 && y == 0)
            {
                if(board[x+1][y] == WHITE || board[x][y+1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x+1][y] == EMPTY || board[x][y+1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(x == board.length-1 && y == board[x].length-1)
            {
                if(board[x-1][y] == WHITE || board[x][y-1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x-1][y] == EMPTY || board[x][y-1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(x == 0 && y == board[x].length-1)
            {
                if(board[x+1][y] == WHITE || board[x][y-1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x+1][y] == EMPTY || board[x][y-1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(x == board.length-1 && y == 0)
            {
                if(board[x-1][y] == WHITE || board[x][y+1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x-1][y] == EMPTY || board[x][y+1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(x == 0)
            {
                if(board[x+1][y] == WHITE || board[x][y+1] == WHITE || board[x][y-1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x+1][y] == EMPTY || board[x][y+1] == EMPTY || board[x][y-1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(y == 0)
            {
                if(board[x+1][y] == WHITE || board[x-1][y] == WHITE || board[x][y+1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x+1][y] == EMPTY || board[x-1][y] == EMPTY || board[x][y+1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(x == board.length-1)
            {
                if(board[x-1][y] == WHITE || board[x][y+1] == WHITE || board[x][y-1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x-1][y] == EMPTY || board[x][y+1] == EMPTY || board[x][y-1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(y == board[x].length-1)
            {
                if(board[x+1][y] == WHITE || board[x-1][y] == WHITE || board[x][y-1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x+1][y] == EMPTY || board[x-1][y] == EMPTY || board[x][y-1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
        }
```

Actual Student Code.

```
if(board[x][y] == WHITEIMPERIL)
        {
            if(x > 0 && x < board.length-1 && (y > 0 && y < board[x].length-1))
            {
                if(board[x+1][y] == WHITE || board[x-1][y] == WHITE || board[x][y+1] == WHITE || board[x][y-1] == WHITE)

                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x+1][y] == EMPTY || board[x-1][y] == EMPTY || board[x][y+1] == EMPTY || board[x][y-1] == EMPTY)

                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(x == 0 && y == 0)
            {
                if(board[x+1][y] == WHITE || board[x][y+1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x+1][y] == EMPTY || board[x][y+1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(x == board.length-1 && y == board[x].length-1)
            {
                if(board[x-1][y] == WHITE || board[x][y-1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x-1][y] == EMPTY || board[x][y-1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(x == 0 && y == board[x].length-1)
            {
                if(board[x+1][y] == WHITE || board[x][y-1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x+1][y] == EMPTY || board[x][y-1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(x == board.length-1 && y == 0)
            {
                if(board[x-1][y] == WHITE || board[x][y+1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x-1][y] == EMPTY || board[x][y+1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(x == 0)
            {
                if(board[x+1][y] == WHITE || board[x][y+1] == WHITE || board[x][y-1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x+1][y] == EMPTY || board[x][y+1] == EMPTY || board[x][y-1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(y == 0)
            {
                if(board[x+1][y] == WHITE || board[x-1][y] == WHITE || board[x][y+1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x+1][y] == EMPTY || board[x-1][y] == EMPTY || board[x][y+1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(x == board.length-1)
            {
                if(board[x-1][y] == WHITE || board[x][y+1] == WHITE || board[x][y-1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x-1][y] == EMPTY || board[x][y+1] == EMPTY || board[x][y-1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
            else if(y == board[x].length-1)
            {
                if(board[x+1][y] == WHITE || board[x-1][y] == WHITE || board[x][y-1] == WHITE)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
                else if(board[x+1][y] == EMPTY || board[x-1][y] == EMPTY || board[x][y-1] == EMPTY)
                {
                    board[x][y] = WHITE;
                    stop = true;
                }
            }
        }
```

Actual Student Code.
White stones only....

Good Luck!

If you have any questions the TAs and I are happy to help.