CISC106 Fall 2009 Lab09

- Review the code examples from class.
- Some programs below are associated with a question. **Answer the questions** using comments below your code in the m-file.
- The office hours of the TAs and the instructor are on the class website. Visit us!
- **NOTE:** Every function comment section should contain, at a minimum, *three examples* of the function being called and the result of evaluating the call. These examples must include boundary conditions (as discussed in class). Your test files must cover *at least* these exact examples (otherwise, why did you choose them?) and possibly more. Testing is important.
- **Every** M-file you write or modify must be demonstrated, either by running a script test file in a diary or by testing at the command line. Note that if you write function **foo.m** and test script **fooTest.m**, you can demonstrate both by running only fooTest.m (assuming fooTest works).

Problems

1. Write a function that uses **patch** to draw a *regular polygon*. Use 'axis equal' so that it looks regular, too! Your function should take as parameters a lower left corner of a square that contains the polygon, and the length of a side. Your function should return the handle to the drawn polygon. Hint: first see what the orientation of a hex is in the project so you don't do extra work. You can check out the Hex game in wikipedia.

2. In a diary, use the **get** and **set** commands on the handle returned from your polygon drawing function. Specifically, **set** and **get** the 'UserData' property and modify the color of the polygon using the 'FaceColor' property.

3. In a diary, use the Matlab function 'repmat' once each to *rep*eat the *mat*rix shown to create the desired result:

$$\begin{matrix}1\\2\end{matrix} \text{ to } \begin{matrix}1&1&1&1\\2&2&2&2\end{matrix}$$

$$\begin{matrix}1&2\\3&4\end{matrix} \text{ to } \begin{matrix}1&2&1&2\\3&4&3&4\\1&2&1&2\\3&4&3&4\end{matrix}$$

4. Copy your drawTriangle function from lab02. Add a test for when the number of arguments is zero and print a nice error message if the function is called with no arguments. Demostrate in a diary. Hint: Google 'matlab nargin'

5. In a diary, run the Matlab function **bwlabel** on the matrix `[2 2 0 3; 3 0 7 4; 5 0 6 6]` and a second argument of 4. Now run the same function on the same data with argument of 8. What changed? What is the default setting if you call **bwlabel** with only the matrix?

6. In a diary, convert the vector 1:16 to a 4x4 square matrix; then convert the square to a 2x8 matrix. Hint: use the Matlab reshape command.

7. In a diary, run the Matlab **intersect** and **union**:

```
>> intersect([0 1 1 0 3 3 0 7], [0 0 2 0 2 0 3 7])
```

```
>> union([0 1 1 0 3 3 0 7], [0 0 2 0 2 0 3 7])
```

8. In a diary, run the Matlab **intersect** command on the first and last rows of the result from the earlier call to bwlabel. Think about how this could be used to detect a path between two edges in your Hex board on the project.

If your TA requires a paper copy, be sure that you have a printed copy of your function M-files, script M-files,image files, and diary files demonstrating your testing. All must be stapled together, with your name and lab section on the top page.

Be sure that you upload a copy of all the MATLAB function, script, imasge, and diary files to Sakai. Then, click submit ONLY ONCE to send these to your Sakai and your TA.

On the first page of every printed copy for this course, your name, section, and TA's name must appear.