# Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Section: \_\_\_\_\_\_

CS 211 Programming Exercise 3

2D Graphics Manipulation

100 Points

**Due Date/Time**:

This PEX is due at 1700 on Tuesday 15 April 2014. The progress check for this homework will occur at the beginning of class on lesson 29.

**Help Policy**:

**AUTHORIZED RESOURCES:** Any, except another cadet’s program.

**NOTE:**

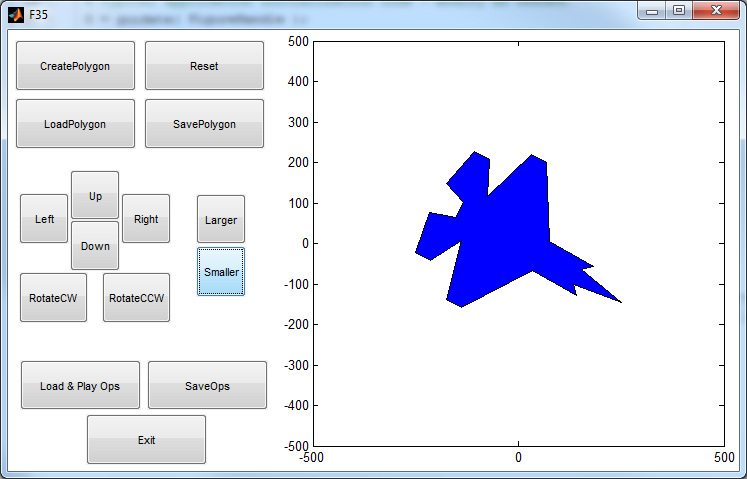
* Never copy another person’s work and submit it as your own.
* Do not jointly work this assignment.
* Do not look at another cadet’s program.
* You must document all help received from sources other than your instructor or instructor-provided course materials (including your textbook).
* DFCS will recommend a course grade of F for any cadet who egregiously violates this Help Policy or contributes to a violation by others.

**Goal:**

Create a graphics manipulation program that uses matrix multiplication and matrix transformations.

**PEX Objectives:**

* Practice matrix manipulation techniques.
* Be able to read and write data to a file.
* Learn how matrix transformations can be used for computer graphics
* Demonstrate good software design principles focused on functional decomposition into small, manageable functions.
* Demonstrate use of parameter passing and use of sub-functions.
* Practice robust coding techniques.
* Demonstrate use of graphics user interface components and dialog boxes.



**Background and Requirements[[1]](#footnote-1):**

Interesting graphical manipulations can be accomplished with matrix multiplication. You will be creating a program which uses matrix multiplication to manipulate and display a filled polygon. You are not allowed to use the built-in matrix multiplier in MATLAB, you must write a sub-function to accomplish this task.

Your program will:

* allow the user to create, save, and load a polygon (as a series of XY coordinates)
* allow the user to perform a series of graphical manipulations on a polygon--scale, rotate, translate (move)
* save a series of operations to a file
* load and replay a series of operations
* include a sub-function that receives as input two matrices and multiplies these matrices and returns the result
* have a graphical user interface

**Implementation Suggestions**:

* Read through this document and run both example programs VERY CAREFULLY and MULTIPLE TIMES so you understand what is expected and what is provided.
* There are two example programs provided. The first is a representation of what you must have completed for the progress check portion of this PEX. The second is a representation of what you must have completed for the final PEX turn-in. The progress check file is named, pex3progcheck.p, and the final is named, pex3.p. You can find these files on the course web site.
* This is a non-trivial program that is best approached using functional decomposition into a hierarchy of sub-functions. The instructor’s solution used the following methods:
  + pex3() – the primary function opens the figure window and displays the menu/GUI, shows the default image (square), processes user commands, and closes the figure window upon a user exit command.
  + DrawPoly() - draws the filled polygon representing the current image.
  + LoadPoly() - loads a polygon from a file.
  + SavePoly() - saves polygon to a file. *(not required for progress check)*
  + CreatePoly() - creates a polygon by allowing the user to specify a series of points via mouse clicks *(not required for progress check)*
  + MXmult() - performs matrix multiplication of two input matrices and returns the resulting product matrix.
  + LoadOps() – loads a series of graphical operations from a file.
  + SaveOps() – saves a series of graphical operations to a file.
  + PlayOps() – plays a series of graphical operations on the current polygon.
  + Reset() – resets the program to its initial state (square polygon with no operations).
  + Exit() – exits the program.
  + GetUserOption() – gets the users selection via a textual menu. *(only for progress check)*
  + There will also be a series of functions that perform the user selected graphical manipulation on the polygon. (e.g. ScaleUP(), ScaleDown(), RotateCW(), RotateCCW() etc).
* It is recommended that you implement this program in an incremental manner, writing and testing at most one function (or part of a function) before going on to the next part.
  + You can do this in the **top-down method** starting with the primary function and then having it call subfunctions that currently do nothing but possibly return a set value (these are called “stubs”). Once that works, implement one of the stubbed methods; creating stubs for other methods that it will need to call.
  + Another approach is the **bottom-up method** were you start with functions such as MXmult() and use “driver” code in the main method to call them for testing.
  + Finally, some prefer the **sandwich method** which combines the top-down approach for some methods with a bottoms-up approach for other. They meet in the middle; “sandwich”ing the results.
* Create transformed images by matrix multiplication of the appropriate transformation matrix times the matrix describing the polygon for the current shape. The transformation matrices are provided in the pex3template.m file on the course web site.
* When doing the transformations, transform the current image replacing the current image with the transformed version.
* Ensure your code is properly commented.
* The MATLAB function to draw a filled polygon is fill().
* The MATLAB function to get a series of points from the user is ginput().
* The MATLAB function to pause your program is pause(). This will be useful during playback of the graphical operations after the operations file is loaded.
* Sample input/output files are provided for your use (look at them!).
  + Saved operations are named \*.ops
  + Saved polygons are named \*.obj
* Carefully inspect the description of the polygon and the transformation matrix provided in the PEX3 template. Notice that the matrix describing the polygon has a 3rd row of values that are only used for calculations. This 3rd row of ones is not part of the polygon and is NOT used to display it.
* Document ALL HELP received other than that from your instructor, the course text, lesson notes/slides, and course example programs.
  + Provide your DOCUMENTATION in a separate area of the header comment block in your MATLAB file. Be sure to include ALL of YOUR DOCUMENTATION.
* Submit your MATLAB file and any object file you wish your instructor to use via the link on the course web site.
  + If for some reason, this submission link does NOT work, simply email the required file(s) to your instructor.

# Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Section: \_\_\_\_\_\_

CS 211 - PEX 3   
Score Sheet

100 Points

|  |  |
| --- | --- |
| **Points** |  |
|  | (10 points) Progress check |
|  | (10 points) Well designed graphical user interface. |
|  | (10 points) Correct implementation of the matrix multiplication sub-function. |
|  | (20 points) Correctly performs the transformations. |
|  | (5 points) Correctly displays the updated image |
|  | (15 points) Creates/Saves/Loads polygon from/to a file. |
|  | (15 points) Saves/Loads/Plays operations on a polygon. |
|  | (5 points) Uses dialog boxes to allow user to select files for load/save. To include proper handling of the cancel button. |
|  | (5 points) Appropriate header comments and all functions and provides in-line comments at least every 3-5 lines. |
|  | (5 points) Uses good descriptive variable names in the proper format, appropriate use of whitespace, code is properly formatted and indented. |
|  | **TOTAL (out of 100)** |

1. The initial idea for this Pex was provided by Dr. Steve Hadfield [↑](#footnote-ref-1)