

#### Week 4 Homework – CMSC405

1. Show the results and transformation matrices used for a translation of  $(-45, 68)$ , a rotation of 65 degrees and a scale of  $(0.5, 2.2)$  applied to a starting point of  $(80, -100)$ . You should use 3x3 matrix math. Note: Your results should not reset to the starting point of  $80, -100$ . The starting point of your subsequent transformation should be the output of the previous transformation. Also, you do not need to show matrix math calculations, but you should show the matrix and the starting points and the results for each geometric transformation. Finally, show that you receive the same answer regardless of whether you perform each step individually, or if you create a composite matrix for the 3 geometric transformations and multiply that by the starting point.
2. Using existing OpenGL functions including `glRotatef()`, `glTranslatef()`, `glScalef()` along with the `glPushMatrix()` and `glPopMatrix()` functions provided in the Week 4 – 2D transformation notes and associated source code, create your own **unique** computer graphic that is comprised of multiple polygons, multiple colors and multiple geometric transformations. There are many an endless number of different output results that are possible. Be creative and try to output an interesting and unique geometric design. Feel free to add text and lines to enhance your graphic.
3. Using Visual C++ and your OpenGL configured environment, write an application that provides 3 windows for listing help commands, object inventory and a map for a simple game of your design. There is much flexibility in what you do here. There is no required functionality in your game at this point- just the graphical interfaces. Think of this as 3 windows showing some text and graphics that would make sense to have in your game. Your help window would have a list of 5-10 commands you would use in the game. The object inventory would show 5-10 graphical objects your users might have in their inventory. These objects should be created by you and use the polygons, bitmaps and other primitives studied so far, to create more complex shapes such as guns, swords, hats, pants and other objects that would make sense to have as inventory in your game. The map window should provide a top down view of the game map. It should consist of geometric shapes and text labels showing an overview of game map area. You are welcome to make it more complex but you may run out of time if you get too fancy. Have fun and be creative.

**Deliverables:** You should submit a well-organized, word document that includes the results for the first question of this assignment along with a screen captures and descriptions of the output resulting from the running of your Visual C++ code. For your game interface, you should describe the game concept to include the theme of your game and discuss how the inventory and game map will support your theme. You should submit your C++ source code for each of your 3 applications. You should name your word assignment “yournamehw4.doc” (or .docx, or .PDF). You should name your C++ source code yournamehw4\_1.cpp, yournamehw4\_2.cpp, and yournamehw4\_3.cpp respectively for your three C++ applications.