CSci 5607, Spring 2022 Name _____
Assignment 2a: Programming Interactive 2D Graphics with OpenGL and GLFW
Due: Friday April 8th Score (out of 100) _____

____ The student has convincingly demonstrated their ability to compile and run the provided template code. (5 pts)

____ The keyboard callback function correctly recognizes when the arrow keys have been pressed, and takes an appropriate action (such as updating a variable that controls the scaling of the model). (5 pts)

____ The program enables the user to robustly adjust the width and/or height of a 2D geometric object according to input provided via the keyboard. The rate of change is reasonable: the object does not get too big or too small "too fast", and it never flips over as a result of scaling commands. Although it is allowable for extreme scaling to result in some objects becoming too small to see, the model never enters a state from which recovery (short of a full reset) is impossible. (10 pts)

____ Re-sizing the object does not cause a noticeable shift in the object's central position. Re-sizing is always applied with respect to the model's intrinsic horizontal and vertical axes. Pressing the left arrow key always causes the object to become thinner, and pressing the right arrow key always causes it to become wider, with respect to its original state; likewise pressing the up arrow key always causes the object to become taller and pressing the down arrow key always causes the object to become shorter. This specified scaling behavior occurs irrespectively of the orientation or position of the object in the viewing window. (10 pts)

____ The program is capable of sensing the movement of the cursor and interpreting it appropriately. Cursor movement doesn't lead to a change in the model's position or orientation until the user signals (e.g. via a button press) that they are intending to use the cursor to rotate or translate the model. The program is able to successfully differentiate between when the user is seeking to change the object's position versus its orientation. For instance, when the 'control' key (or an appropriate alternative) is pressed at the same time as the left mouse button (or equivalent) is pressed, the movement of the cursor is used to modify parameters pertaining to the object's position; if the 'control' key (or its designated alternative) is *not* pressed at the same time as the left mouse button (or equivalent) is pressed, the cursor's movement is used modify parameters pertaining to the object's rotation. (5 pts)

____ The program successfully enables the user to interactively rotate a 2D geometric object via user-controlled cursor motion. The mapping between the amount of cursor movement and the amount of object rotation is reasonable (i.e. a small amount of cursor movement leads to a small amount of rotation and larger amounts of cursor movement cause larger amounts of object rotation); the entire object should <u>not</u> spin around more than once or twice for an amount of cursor motion that is comfortable to achieve in one continuous movement. (10 pts)

____ If the model is a square, it always rotates about its centroid; if it is an imported model, it always rotates about a point whose location is fixed in model coordinates. A clear and consistent relationship between cursor movement and object rotation is enforced. When the cursor moves a small amount to the right, the object rotates by a small amount in the clockwise direction; when the cursor moves a small amount to the left, the object rotates by a small amount in the counterclockwise direction. Up and down movements of the cursor (within reason) do not change the rotational state of the object. This specified behavior occurs irrespective of the position of the object within the viewing window and irrespective of its orientation. The object never rotates clockwise when the mouse moves left, or counter-clockwise when the mouse moves right, and it never spins in an arc whose radius depends on the position of the object with respect to the display window. (20 pts)

____ The program successfully enables the user to interactively reposition a 2D geometric object within a display window via user-controlled cursor motion in conjunction with an appropriate key press. (10 pts)

____ While the object is being re-positioned, its movement follows the cursor movement relatively faithfully: when the cursor moves in a given direction, the object moves in that same direction; when the cursor moves by a given amount, the object moves by a roughly similar amount. This behavior is maintained irrespective of the scale of the object or its orientation within the viewing window (15 pts)

____ The program prevents the user from moving the object irretrievably "off screen". Parts of the object may be "clipped" by the edges of the screen but the object never disappears entirely; some portion of the object always remains visible within the display window. (5 pts)

____ The program is reasonably well-written and well-structured; comments are used to explain what is going on at various critical stages in the code, and there are no glaring inefficiencies in the implementation. The code is written in a reasonably platform-independent manner and is straightforward to compile given the students' provided instructions/makefiles. The execution of the program is robust; there are no unexpected glitches in performance. The student has turned in all of their source code, along with a video or sequence of images demonstrating the successful performance of their program. (5 pts)

Extra Credit:

____ The program includes a keypress option that allows resetting the state of the geometric transformations, so that the object returns to its original size and position, from which it can be manipulated again as before. (3 pts)

____ The program is capable of reading a model description from an input file and working with that input instead of the two hard-coded triangles provided in the template code. (5 pts)