BIL 421/BIL 539:Spring 2020

Assignment 2

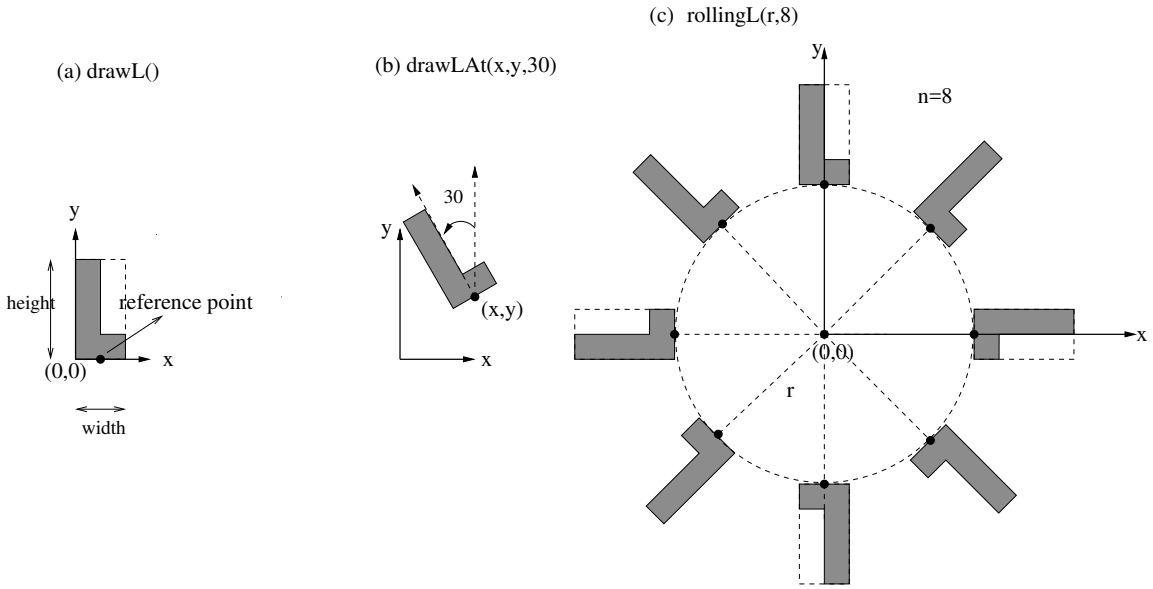
Assigned Friday, January 31. The program must be submitted to our grader (sksueksioglu@gmail.com) by Friday, February 14 (any time up to midnight). Subject of the email should say "Windows: BIL421 Assignment 2". Replace "Windows" with "Linux" or "MacOS" if your code is to be compiled in those operating systems. Name your Project directory as "YournameAssg2". Before submitting, **zip** your Project directory, and email a single zipped file as attachment. Do NOT include any .exe files in your submission. Remember that your code should be fully documented. Check the course syllabus for the late policy. I also would like to remind you once again about the academic honesty rules stated in the syllabus. The implementation can be done in immediate mode(glVertex) or shader-based OpenGL (VAOs and VBOs). I suggest the shader-based so that you are also prepared for the next homework. **In any case, you are NOT allowed to use glTranslate, glRotate, glScale.** But, you are allowed to use Angel's mat.h and vec.h. Do all drawing within the default rectangle/viewing volume as we did in sierpinski and rotating cube examples.

Overview: You are given 3 problems involving transformations below. First do these on paper, then code it up.

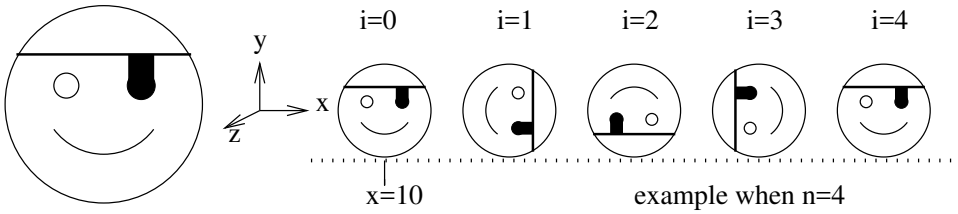
Problem 1. For this problem, your graphics window must be of size 500 by 500. Your idealized drawing window must be a square centered at (0,0) with a sidelength of 500.

- Initially draw the character *'L'* so that the bottom and left sides of its bounding rectangle are aligned with x and y axes, respectively (the lower left corner of its bounding rectangle coincides with (0,0)) as shown in figure below part (a). You can adjust the height and width of the character as you wish (but the whole shape must stay in drawing window). The character has an implicit *reference point* which is the midpoint of the bottom edge of the character's bounding rectangle. Store the vertices of this shape in an array of vertices called Lshape.
- Single Rotation Mode: On right mouse click you switch to single rotation mode. Let (x, y) denote the mouse coordinates and Θ be an angle initialized as 30 degrees. In single rotation mode, draw the character *'L'* rotated counterclockwise by angle Θ with its reference point located at (x, y). See part (b) of the figure below. **You must do this by transforming the original vertices that you stored in the Lshape array.** If the 'r' key is pressed in this mode, the angle Θ must be incremented by 5 degrees and the shape must be redrawn accordingly. 'r' can be pressed as many times as we want when in this mode.
- Animation Mode: When the key 'a' is pressed, you switch to animation mode. In this mode, produce an animation of drawings of the character *'L'* rolling around a circle of radius 100 centered at (0,0). There will be n drawings around the circle such that the reference points are evenly spaced. (See figure below, part(c)). You may assume n is at least 2. Try with different values of n. **You must do this part by transforming the**

vertices that you stored in the Lshape array. Note that 'r' should have no effect in animation mode.



Problem 2. Initially, draw a 2-dimensional pirate face centered at the origin and lying on the x-y plane as shown in the figure below, part(a). Store the vertices of this pirate in an array called pirateface. You are free to set the radius of the circle forming the face. When the key 'a' is pressed, produce a sequence of drawings of the face rolling along the x-axis, but scaled down to a radius of 1/2 of its original radius. (See figure below, part(b)). This animation will look like as if the face is rolling.



You can do this by drawing the pirate $n+1$ times, for $i = 0, 1, 2, \ldots, n$. When $i = 0$, the pirate will be displayed upright at some initial position, say at $x = 10$. As i increases, the face rotates and translates to its next position. When $i = n$, it will undergo a full 360° rotation clockwise. **You must do this part by transforming the vertices that you stored in your original pirateface array.**

Problem 3. Now, write a program where a planet object (a sphere) is revolving around the sun (another sphere) in a circular orbit. The spheres should be drawn as wireframes. The sun is stationary, but the planet moves. To do this, first specify vertices to draw a sphere centered at the origin. Then scale/translate/rotate these vertices appropriately to draw the sun and the planet. In addition make the planet rotate around its own axis. You will use the deafult viewing volume.