COSC363 Computer Graphics
Lab05: Sweep Surfaces

Aim:

This lab is a continuation of Lab04 on object modelling. In this lab, we will consider another example of sweep surface obtained by successive transformations of a polygonal shape.

I. Tower.cpp:

The "Turning Torso" (Fig. a), a twisted tower in Malmo, Sweden is an excellent architectural example of a sweep surface/structure (For more information on the tower, see: http://en.wikipedia.org/wiki/Turning_Torso). We can generate a model of the tower using the shape shown in Fig (b) as the base polygon. The program `Tower.cpp` contains a representation of the polygonal shape using vertex coordinates stored in arrays vx[], vy[], vz[] inside the `display()` function. The polygon has 18 vertices as shown in Fig. (c). Since this is a closed polygon, the first vertex is appended to the end of the list again, to get a closed quad strip. The vertex list therefore contains 19 vertices.

Fig. (a)

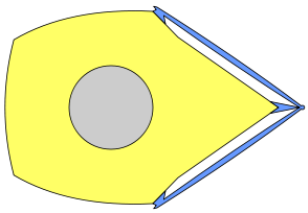

Fig. (b)

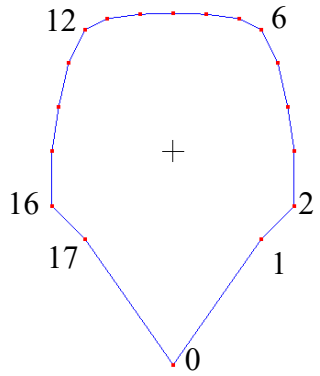

Fig. (c)

The tower consists of 9 "blocks", each block having a height of approx. 20 meters, and turned *clockwise* about the vertical axis by 10 degrees relative to the lower block.

1. The program displays only the base polygon (Fig. c) at the bottom part of the screen. The camera can be moved around the scene using the left and right arrow keys.

2. Delete the code segment that draws the base polygon inside the `display()` function and implement the following algorithm:

   - Generate a new set of transformed coordinates wx[i], wy[i], wz[i], i = 0.. *N*–1 (already declared in the program) by rotating the points (vx[i], vy[i], vz[i]) by -10 degs (clockwise) about the *y*-axis and translating along *y*-axis by 20 units. Since we require the transformed points, we cannot use the OpenGL function `glRotatef()` to perform the rotation. Instead, we use the following equations:

     $w_x = v_x \cos\theta + v_z \sin\theta$, $\theta = -10$ Degs (Convert to radians!)
     $w_y = v_y + \mathbf{20}$
     $w_z = -v_x \sin\theta + v_z \cos\theta$

   - Join the points V_i = (vx[i], vy[i], vz[i]) and the transformed points W_i = (wx[i], wy[i], wz[i]) using a quad strip to generate the surface of the extruded shape. Refer to the code given on Slide [5]-12.

   - After drawing the quad-strip, replace the values in (vx[], vy[], vz[]) with the transformed values in (wx[], wy[], wz[]).

   The above three steps generate the surface of one "block" of the tower. Repeat the steps 8 more times.

   The light and camera positions have already been defined in the program. Change the value of the global variable `viewAngle` to -160 degs. The program should generate an output similar to the one given in Fig. (d).

Fig. (d)

3. The program includes the necessary functions for loading a bitmap texture "TowerTexture.bmp". Please uncomment the three lines inside the

`initialise()` function to enable texture mapping. The texture has 5 parts corresponding to 5 sides of the tower (Fig (e)).

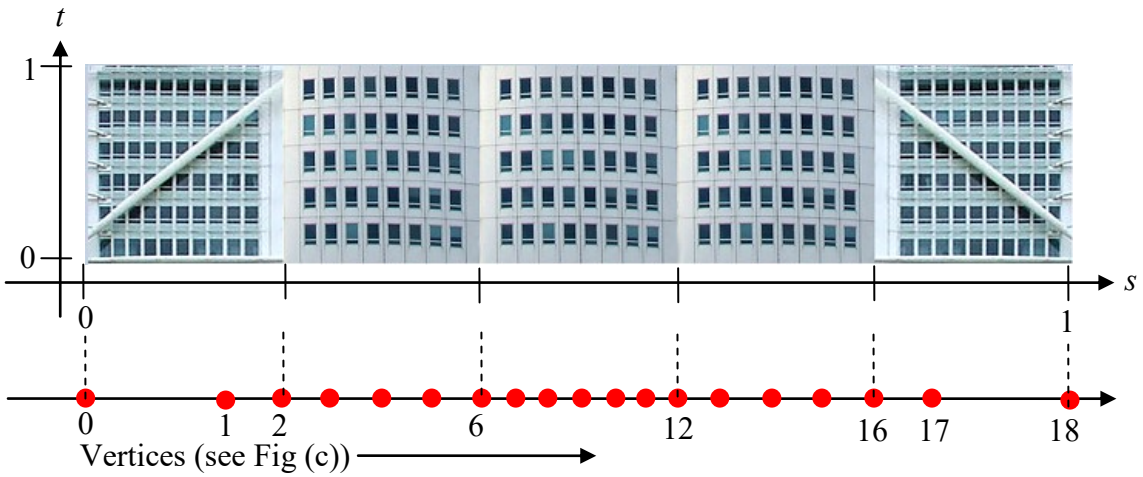

Fig. (e)

4. We need to assign texture coordinates to vertices of the quad strip. If a vertex V_i has texture coordinates $(s_i, 0)$, then the corresponding vertex W_i will have texture coordinates $(s_i, 1)$. The values s_i must be carefully chosen so that the image sections in the texture are properly mapped to the 5 sides of the polygon (see Fig. (c)). Fig(e) shows how this mapping must be done. The table below gives the texture s-coordinates for some of the vertices. Compute the remaining coordinates, assuming that the points are uniformly distributed within each of the 3 middle sections. Your program should then produce an output similar to the one given in Fig. (f).

Vertex index i	Tex Coord s_i
0	0
1	0.15
2	0.2
:	:
6	0.4
:	:
12	0.6
:	:
16	0.8
17	0.85
18	1

Fig. (f). Textured tower

Ref:

[5] Lec05_ObjectModelling.pdf (COSC363 Lecture notes)

II. Quiz-05

The quiz will remain open until **5pm, 3 April, 2020**.