COSC363 Computer Graphics
Lab04: Object Modelling

Aim:

This lab introduces object modeling techniques using quad strips. Such methods are commonly used for generating sweep surfaces. Quad strips also find applications in the modelling of surfaces of revolution.

I. Walls.cpp:

The program `Walls.cpp` displays a simple scene consisting of two vertical quads that are texture mapped with a brick pattern, and a horizontal ground plane that is texture mapped with a floor pattern (Fig. 1). The scene also includes a quad that will be used for displaying the model (image) of a tree, and a polygonal line (a blue curve on the floor plane) that will be used for creating a curved wall using a quad strip.

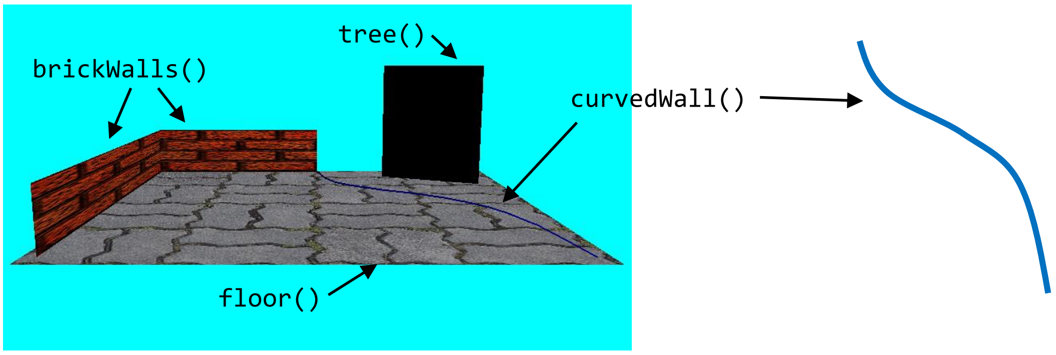

Fig. 1.

1. In the `brickWalls()` function, increase the number of repetitions of the texture on the first quad ("Front Wall") to 4 in the 's' direction, and 2 in the 't' direction (Fig. 2(a)). Similarly, increase the number of repetitions of the texture on the second quad ("Left Wall") to 8 in the 's' direction, and 2 in the 't' direction. In the `floor()` function, set the number of repetitions for the floor texture to 4 in both directions. Fig. 2(b)) shows the output of the program.

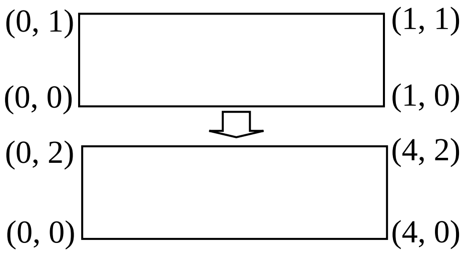

Fig. 2(a)

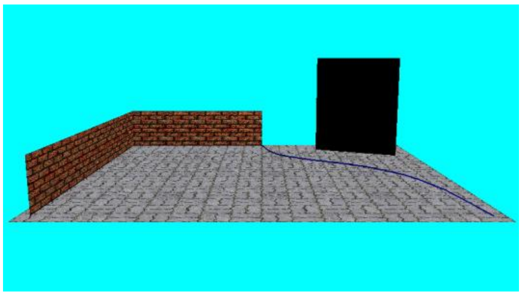

Fig. 2(b)

2. In the `tree()` function, assign texture coordinates (0, 0), (1, 0), (1, 1), (0, 1) to the vertices of the quad. Change the colour of the quad to white. The quad now displays the image of a tree contained in the texture (Fig. 3(a)). This quad serves as a billboard for the tree model. Note that the quad is oriented towards the camera. The background of the tree texture must be removed using the alpha test (Slide [04]:37). Include the following statements in the `intialise()` function:

   ```
   glEnable(GL_ALPHA_TEST);
   glAlphaFunc(GL_GREATER, 0);
   ```

   The output of the program is shown in Fig. 3(b). The specks around the tree image are caused by very small values (noise) in the alpha channel of the image.

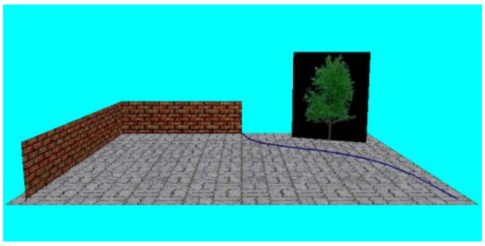

Fig. 3(a)

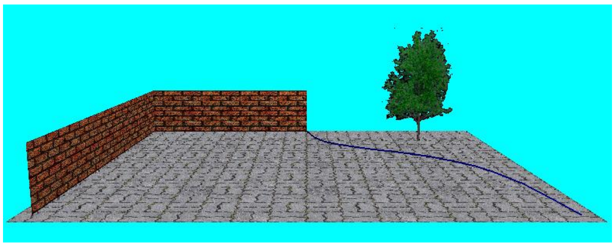

Fig. 3(b)

3. The function `curvedWall()` contains the code for drawing a polygonal line containing 13 vertices (xpts[i], 0, zpts[i]), i = 0..12. Convert this line strip to a quad strip by connecting each vertex (xpts[i], 0, zpts[i]) to its pair (xpts[i], 4, zpts[i]) obtained by translating the curve by 4 units along the y-direction (Fig. 4(a)).

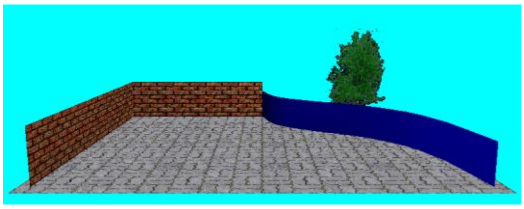

Fig. 4(a)

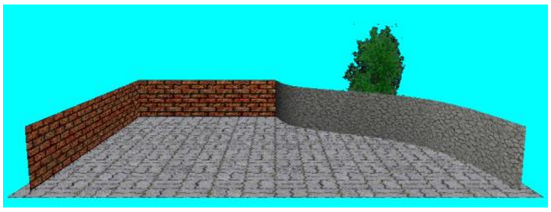

Fig. 4(b)

4. Change the colour of the quad strip to white and assign texture coordinates to each pair of vertices as follows: The index of vertices varies from 0 to N–1 (in our example, N = 13). The index can be directly mapped to the ‘s’ coordinate using the formula $s = i / (N–1)$. If the texture is required to be repeated r times along the quad strip, then $s = i\, r / (N–1)$. The ‘t’ coordinate will have a value 0 along the lower strip and ‘1’ on the upper strip. Assign texture coordinates to the vertices of the quad strip so that the texture is repeated four times along the strip (Fig. 4(b)).

5. In the `display()` function, change the parameters of the `gluLookAt()` function as follows:

   ```
   gluLookAt(eye_x, 5, eye_z,  look_x, 3, look_z,   0, 1, 0);
   ```

   You will now be able to change the camera position and view direction using arrow keys. The program implements the computation of camera position and view direction required for interactive scene navigation.The left and right arrow keys turn the view direction towards left and right of the current direction. The up and down keys move the camera forward and backward in the current direction.

6. Change the background colour to black. This will not only hide the specks around the tree texture but also improve the overall rendering quality of the scene! (Fig. 5). Also note that the `normal()` function included in the program computes per-vertex normal vectors of the base curve using the formula given on Slides [05]:15,16.

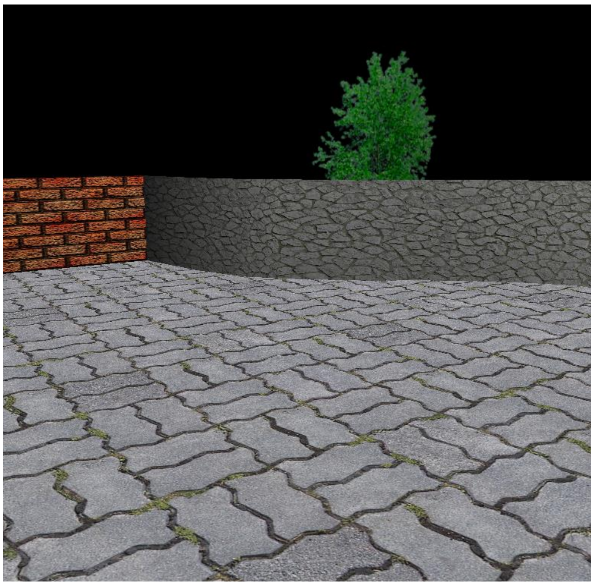

Fig. 5

II. Vase.cpp:

The program `Vase.cpp` contains the vertex coordinates for a 2D base curve on the *xy*-plane in arrays `vx_init[], vy_init[]`,and the vertex normal components (Slide [05]:26) in arrays `nx_init[], ny_init[]`. The base curve contains 50 points (N = 50). The program displays a wireframe model of a flower vase (Fig. 6(a)). The model is generated by revolving a base polygonal curve about the *y*-axis in steps of 10 degrees, and using quad strips to form the bounding surface. The scene can be rotated using left/right arrow keys. The up/down arrow keys change the height of the camera.

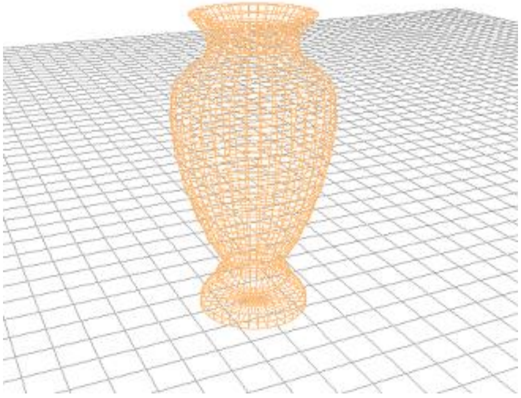

Fig. 6(a)

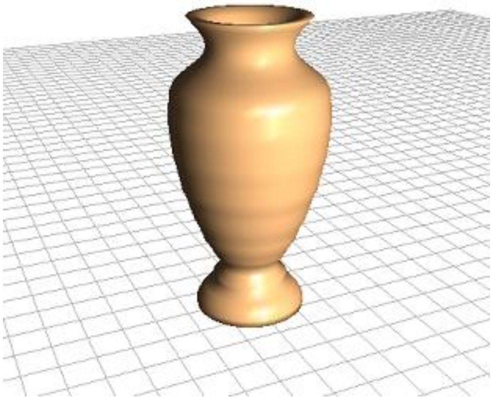

Fig. 6(b)

1. The program uses the code given on Slide [05]:25 to generate quad strips between two consecutives slices. Rendering the model under a light source requires the computation of surface normal vectors. For a surface of revolution, the vertex normal vectors of the base curve can be transformed exactly like vertex coordinates. Use the method outlined on Slide [05]:27 to assign normal vectors to each vertex of the generated quad strips. Change the polygon mode from GL_LINE to GL_FILL, and enable lighting. The output of the program with lighting enabled is given in Fig. 6(b). Note that lighting with vertex normal vectors produces a smooth shading of the surface.

2. The program includes the function to load a bitmap texture "VaseTexture.bmp". We will texture map the image to the whole surface generated above. Please uncomment the function call "`loadTexture();`" in the `initialise()` function. Assign texture coordinates to the vertices of the quad strip (please refer to Slide [05]:28 for more information on computing texture coordinates). The textured output is shown in Fig 7(a).

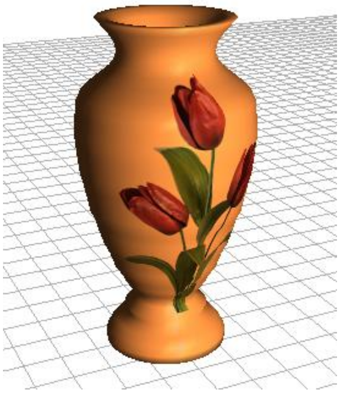

Fig. 7(a)

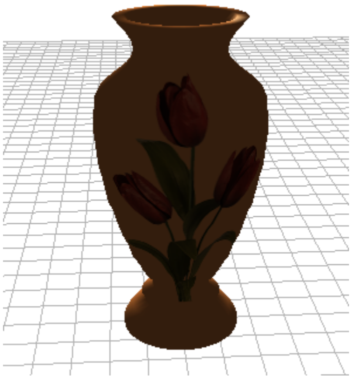

Fig. 7(b)

3. The flower pattern in the shadow region on the vase is barely visible (Fig. 7(b)). OpenGL uses a global ambient light of low intensity (0.2, 0.2, 0.2, 1) to render shadow regions. If we specify an additional ambient color for GL_LIGHT0, the overall intensity of the scene will increase by that factor. Include the following statement in the `initialise()` function:

```
glLightfv(GL_LIGHT0, GL_AMBIENT, grey);
```

The shadow region on the vase will now have brighter shades of colour (Fig. 8)

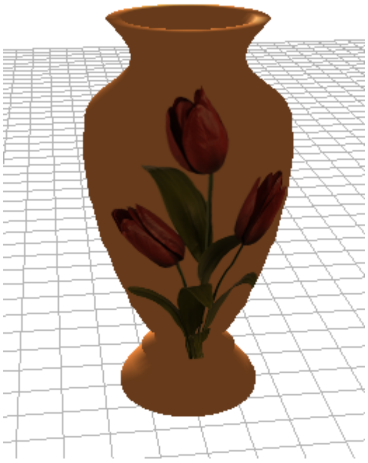

Fig. 8

<u>Ref</u>:

[05]   Lec05_ObjectModelling.pdf (COSC363 Lecture notes)

Textures aquired from Textures.com (https://www.textures.com).

III. Quiz-04

The quiz will remain open until **11:55pm, 31 March 2023**.