Birla Institute of Technology & Science, Pilani

Work Integrated Learning Programmes Division
M. Tech. Software Engineering at Wipro Technologies (WASE)
First Semester 2016 - 2017
Comprehensive Examination (Regular)

Course Number	:	SEWP ZC472
Course Title	:	COMPUTER GRAPHICS
Type of Exam	:	Open Book
Weightage	:	50 %
Duration	:	3 hours
Date of Exam	:	12.02.2017

No. of Pages : 2
No. of Questions : 5

Session: FN

Note:

1. Please read and follow all the instructions given on the cover page of the answer script.
2. Start each answer from a fresh page. All parts of a question should be answered Consecutively.

Q1 a. With appropriate block diagrams differentiate between the two basic display types. [4M]

b. Explain the usage of the following OpenGL callback functions in the graphics programming:

i. glutMouseFunc();

`glutMouseFunc`: registers callback handler for mouse click.

void glutMouseFunc(void (*_func_)(int _button_, int _state_, int _x_, int _y_)

`glutMouseFunc` sets the mouse callback for the *current window*. When a user presses and releases mouse buttons in the window, each press and each release generates a mouse callback. The `button` parameter is one of `GLUT_LEFT_BUTTON`, `GLUT_MIDDLE_BUTTON`, or `GLUT_RIGHT_BUTTON`. For systems with only two mouse buttons, it may not be possible to generate `GLUT_MIDDLE_BUTTON` callback. For systems with a single mouse button, it may be possible to generate only a `GLUT_LEFT_BUTTON` callback. The `state` parameter is either `GLUT_UP` or `GLUT_DOWN` indicating whether the callback was due to a release or press respectively. The `x` and `y` callback parameters indicate the window relative coordinates when the mouse button state changed. If a `GLUT_DOWN` callback for a specific button is triggered, the program can assume a `GLUT_UP`callback for the same button will be generated (assuming the window still has a mouse callback registered) when the mouse button is released even if the mouse has moved outside the window.

ii. glutReshapeFunc();

Usage

```
void glutReshapeFunc(void (*func)(int width, int height));
```

`func`

The new reshape callback function.

Description

`glutReshapeFunc` sets the reshape callback for the *current window*. The reshape callback is triggered when a window is reshaped. A reshape callback is also triggered immediately before a window's first display callback after a window is created or whenever an overlay for the window is established.
The `width` and `height` parameters of the callback specify the new window size in pixels. Before the callback, the *current window* is set to the window that has been reshaped.

If a reshape callback is not registered for a window or `NULL` is passed to `glutReshapeFunc` (to deregister a previously registered callback), the default reshape callback is used. This default callback will simply call `glViewport(0,0,width,height)` on the normal plane (and on the overlay if one exists).

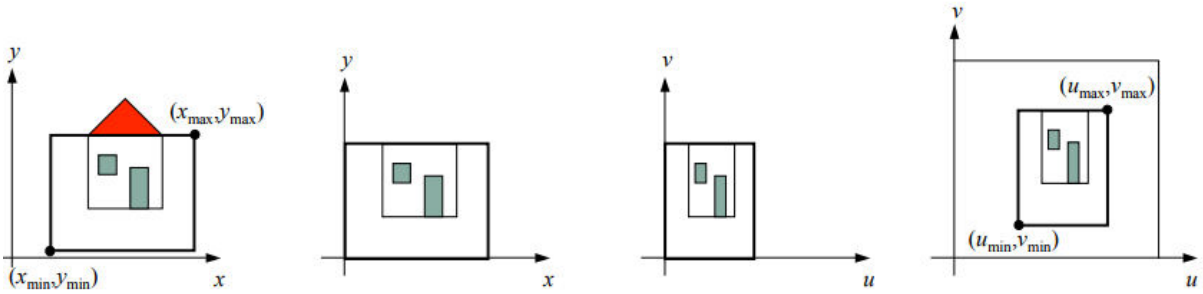
If an overlay is established for the window, a single reshape callback is generated. It is the callback's responsibility to update both the normal plane and overlay for the window (changing the *layer in use* as necessary).

When a top-level window is reshaped, subwindows are not reshaped. It is up to the GLUT program to manage the size and positions of subwindows within a top-level window. Still, reshape callbacks will be triggered for subwindows when their size is changed using `glutReshapeWindow`.

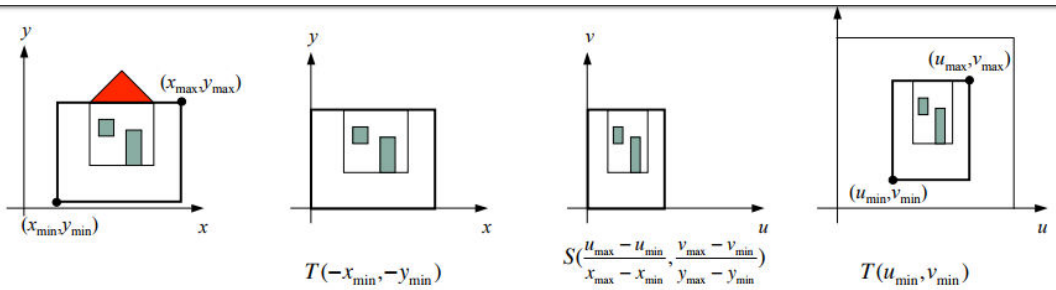
c. Differentiate between the types of Orthographic projection and Perspective projection.[3M]

Q2. Given a Window and Viewport, what is the transformation matrix that maps the window from world coordinates into the viewport in screen coordinates? [10M]

Given a window and viewport, what is the transformation matrix that maps the window from world coordinates into the viewport in screen coordinates? This matrix can be given as a three-step transformation composition as suggested by the following sequence of pictures:



window in world coordinates	window translated to origin	window scaled to size of viewport	translated by (u_{min}, v_{min}) to final position
	$T(-x_{min}, -y_{min})$	$S(\frac{u_{max}-u_{min}}{x_{max}-x_{min}}, \frac{v_{max}-v_{min}}{y_{max}-y_{min}})$	$T(u_{min}, v_{min})$

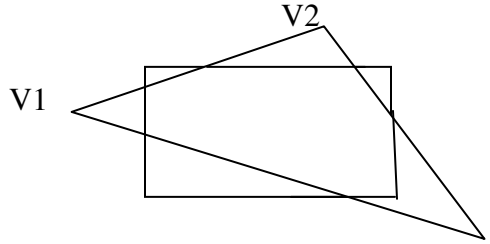


$T(-x_{min}, -y_{min})$ $S(\frac{u_{max}-u_{min}}{x_{max}-x_{min}}, \frac{v_{max}-v_{min}}{y_{max}-y_{min}})$ $T(u_{min}, v_{min})$

$$M_{wv} = T(u_{min}, v_{min}).S(\frac{u_{max}-u_{min}}{x_{max}-x_{min}}, \frac{v_{max}-v_{min}}{y_{max}-y_{min}}).T(-x_{min}, -y_{min})$$

$$= \begin{bmatrix} 1 & 0 & u_{min} \\ 0 & 1 & v_{min} \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} \frac{u_{max}-u_{min}}{x_{max}-x_{min}} & 0 & 0 \\ 0 & \frac{v_{max}-v_{min}}{y_{max}-y_{min}} & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & -x_{min} \\ 0 & 1 & -y_{min} \\ 0 & 0 & 1 \end{bmatrix}$$

Q3. Clip the given polygon below using Sutherland_Hodgeman polygon clipping technique. [10M]



V3

Q4. A) In the Z-Buffer algorithm, show that depth calculation at each pixel on a scan line can be done incrementally if the plane equation for each polygon is available. [5M]

WKT the polygon is planar. We can simplify the calculation of z values for each point on a scan line by using Depth Coherence concept. At (x, y) compute z by Z=(-D-Ax-By)/C At (x+dx, y) value of z= z1-{A/C}(dx) Only one subtraction is need to compute (z(x+1), y) given z(x, y) as dx=1.

B) In the Painters algorithm / the depth sorting method, indicate the tests that are to be carried out to determine if two surfaces R and S need not be ordered. **[5M]**

Tests that are to be carried out are

- Do the polygon's x extent overlap
- Do the polygon's y extent overlap
- Is R entirely on the opposite side of S's plane from the viewpoint
- Is S entirely on the opposite side of R's plane from the viewpoint
- Do the projections of the polygons onto the (x,y) plane not overlap

Q5. A) Explain the important characteristics of Bezier curve. A cubic Bezier curve segment is described by control points P_0 (2,3), P_1(6,6), P_2(8,1),P_3 (4,-3). Find the points on the curve at u=0.2 and u=0.5. [5M]

B) Shape is an important visual feature of an image. Explain briefly about the various shape representation techniques. [5M]

XXXXXX