COMPUTER GRAPHICS (CCG3013)

LESSON 12

SURFACE SHADINGS

COURSE OUTLINE

Lesson	Topic
1	Introduction to computer graphics
2	Graphics hardware and software
3	Geometry in 2D graphics
4 & 5	Geometry in 3D graphics
6 & 7	User interfaces and interactions
8	Colour
9 & 10	Motion and animation
11	Lighting and rendering
12	Surface shadings

REVIEW ON EXERCISE 3

Explain ray casting applications in terms of the features and its computations for rendering.

Ray casting	Features	Computations
Ray tracing	1. A rendering technique. 2. First presented by Arthur Appel in year 1968. 3. Open source and free.	It traces path of light as pixels, which generate an image. It is recursive and thus slow in rendering process.
Mental ray	1. Product quality rendering application. 2. Developed by Mental Images. 3. Acquired by Nvidia in year 2007. 4. It is a cross-platform application. 5. Proprietary license.	Computations included global illumination, photon mapping, depth of field, ambient occlusion, reflection, and soft shadows.
V-ray	1. Computer-generated imagery (CGI) rendering software. 2. Developed by Chaos Group. 3. It is cross-platform application. 4. Proprietary commercial license.	Computations included global illumination, photon mapping, and irradiance maps.

REVIEW ON EXERCISE 4

Identify the initial values for GL_DIFFUSE, GL_SPECULAR, and GL_AMBIENT modes.

Mode of lightings	Initial values
GL_DIFFUSE	(1, 1, 1, 1) for GL_LIGHT0, while (0, 0, 0, 1) for others.
GL_SPECULAR	(1, 1, 1, 1) for GL_LIGHT0, while (0, 0, 0, 1) for others.
GL_AMBIENT	(0, 0, 0, 1)

ADDITIONAL QUESTIONS

Refer to web colour website, convert the following special colours to HSI.

(a) Orange, (r, g, b) = (255, 165, 0).

(b) Pink, (r, g, b) = (255, 192, 203).

(c) Brown [Homework],

(d) Purple [Homework].

ADDITIONAL QUESTIONS

Orange , (r, g, b) = (255, 165, 0). b > g? No.

$$h = \cos^{-1}\left[\frac{2(255)-165-0}{2\sqrt{(255-165)^2+(255-0)(165-0)}}\right]$$

$$= \cos^{-1}\left[\frac{345}{2\sqrt{90^2+42075}}\right]$$

$$= \cos^{-1}\left[\frac{345}{2\sqrt{50175}}\right]$$

$$= \cos^{-1}\left[\frac{345}{2(224)}\right]$$

$$= \cos^{-1}(0.7701)$$

$$= 39.64^{\circ}$$

$$I = \frac{r+g+b}{3} \text{ (2 marks)}$$

$$= \frac{255+165+0}{3} \text{ (1 mark)}$$

$$= 140 \text{ (1 mark)}$$

$$s = 1-\frac{\min(r,g,b)}{I} \text{ (2 marks)}$$

$$= 1-\frac{0}{140} \text{ (1 mark)}$$

$$= 1 \text{ (1 mark)}$$

ADDITIONAL QUESTIONS

Pink, (r, g, b) = (255, 192, 203). b > g? Yes.

$$h = 360° - \cos^{-1}\left[\frac{2r-g-b}{2\sqrt{(r-g)^2+(r-b)(g-b)}}\right] \text{(2 marks)}$$

$$= 360° - \cos^{-1}\left[\frac{2(255)-192-203}{2\sqrt{(255-192)^2+(255-203)(192-203)}}\right] \text{(1 mark)}$$

$$= 360° - \cos^{-1}\left[\frac{115}{2\sqrt{3397}}\right]$$

$$= 360° - \cos^{-1}\left[\frac{115}{116.5676}\right]$$

$$= 360° - \cos^{-1}(0.986552)$$

$$= 360° - 9.4071$$

$$= 350.5929° \text{(1 mark)}$$

$$I = \frac{r+g+b}{3} \text{ (2 marks)}$$

$$= \frac{255+192+203}{3} \text{ (1 mark)}$$

$$= 217 \text{ (1 mark)}$$

$$s = 1 - \frac{\min(r,g,b)}{I} \text{ (2 marks)}$$

$$= 1 - \frac{192}{217} \text{ (1 mark)}$$

$$= 0.1152 \text{(1 mark)}$$

ASSESSMENTS

Structure	Marks (%)	Hand-out	Hand-in
Assignment 1	30	RELEASED	DUE
Assignment 2	30	RELEASED	DUE
Final examination	40	Exam week	

LEARNING OUTCOMES

1. Determine front faces and back faces for 2D/3D models.

2. Apply materials on the surfaces of 2D/3D models based on Phong's reflection model.

CONTENT

No.	Topics	Duration (Minutes)
1	Mini lecturer 1: Texture mapping	15
2	Exercise 1	10
3	Mini lecturer 2: Texture coordinates	15
4	Exercise 2	10
5	Break	10
6	Mini lecturer 3: Texture functions in OpenGL	15
7	Exercise 3	10
8	Mini lecture 4: Texture binding using OpenGL	15
9	Exercise 4	10

MINI LECTURE 1
TEXTURE MAPPING

. . .

TEXTURE MAP

1. Sometimes it refers to diffuse map.

2. It is a two-dimensional (2D) image.

3. Main purpose is to bind this map onto a face of a polygon.

4. There are two ways to render texture map; load from image and generate pixel by pixel.

PHOTOREALISTIC MAPPING

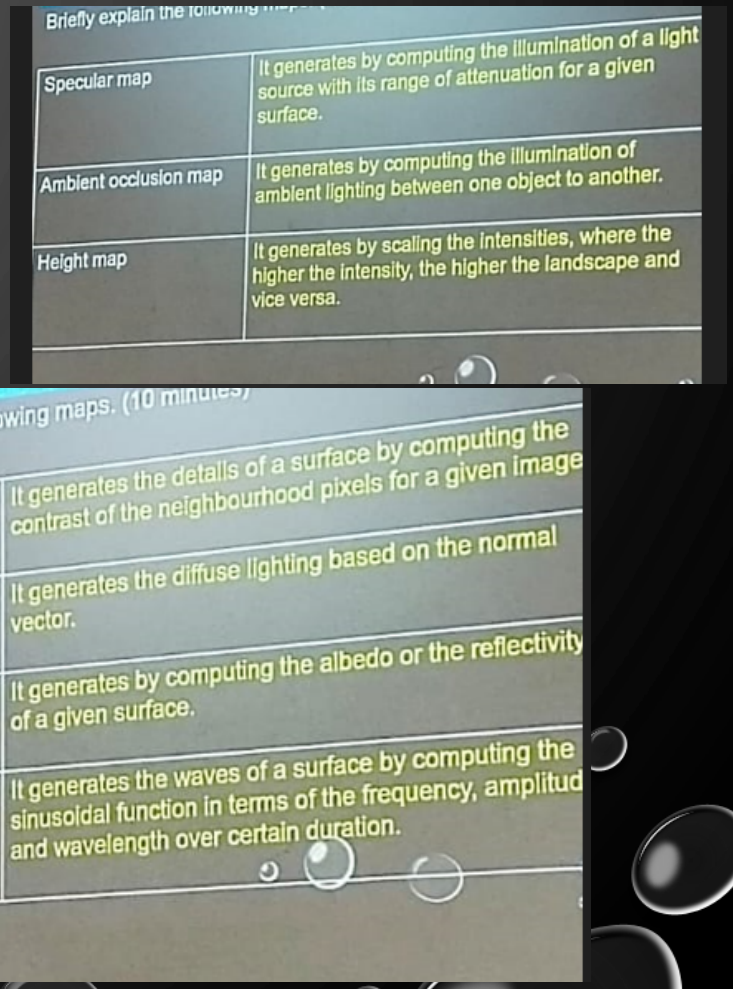
1. To simulate photorealism for a scene, texture map has been expanded into an integrated material mapping system.

2. Mappings included specular mapping, ambient occlusion mapping, height mapping, bump mapping, normal mapping, displacement mapping, reflection mapping and so on.

EXERCISE 1 (15 MINUTES)

Briefly explain the following maps. (10 minutes)

Specular map	Ambient occlusion map
Height map	Bump map
Normal map	Displacement map
Reflection map	



MINI LECTURE 2
TEXTURE COORDINATES

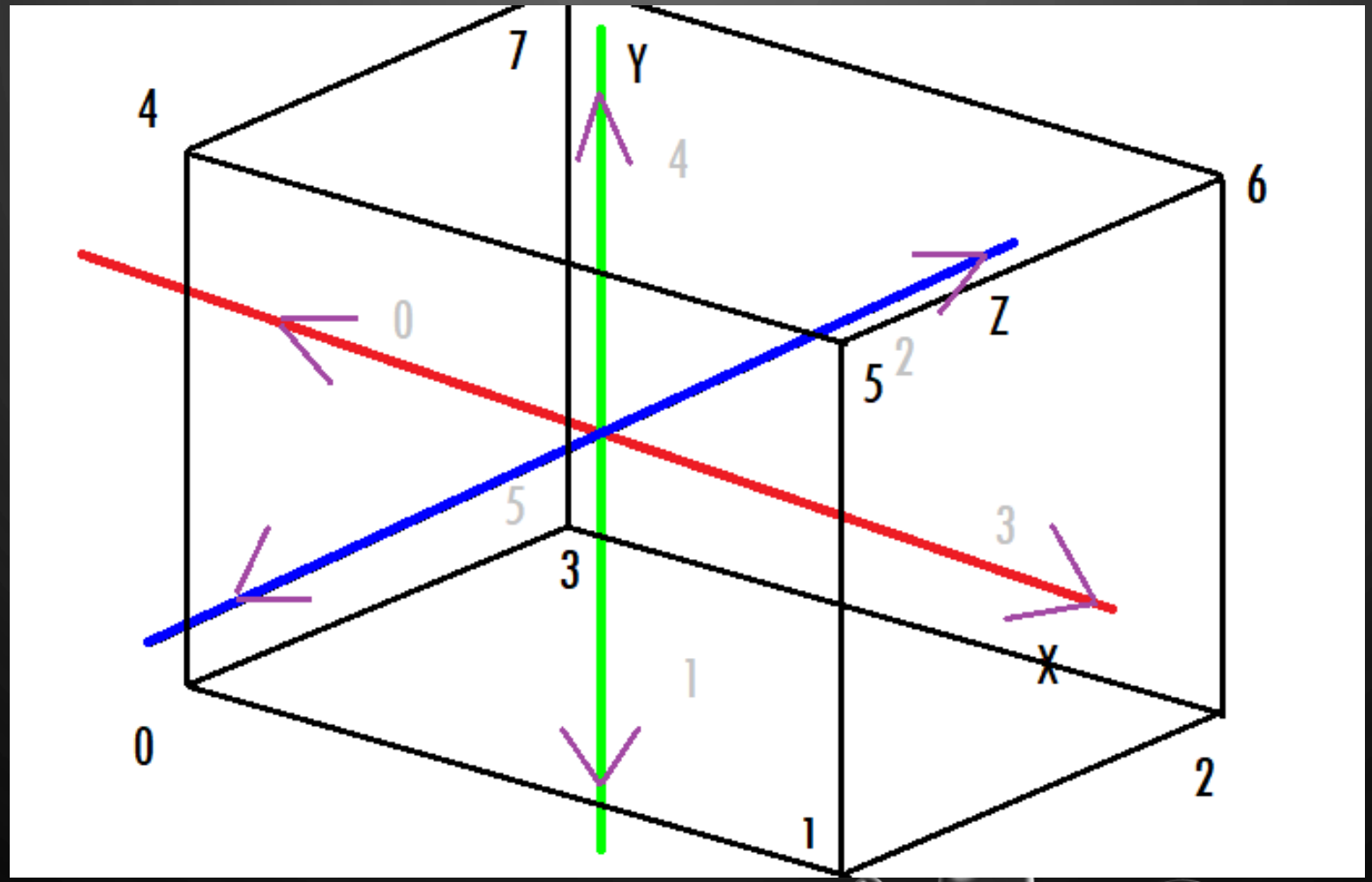
. . .

TEXTURE BINDING

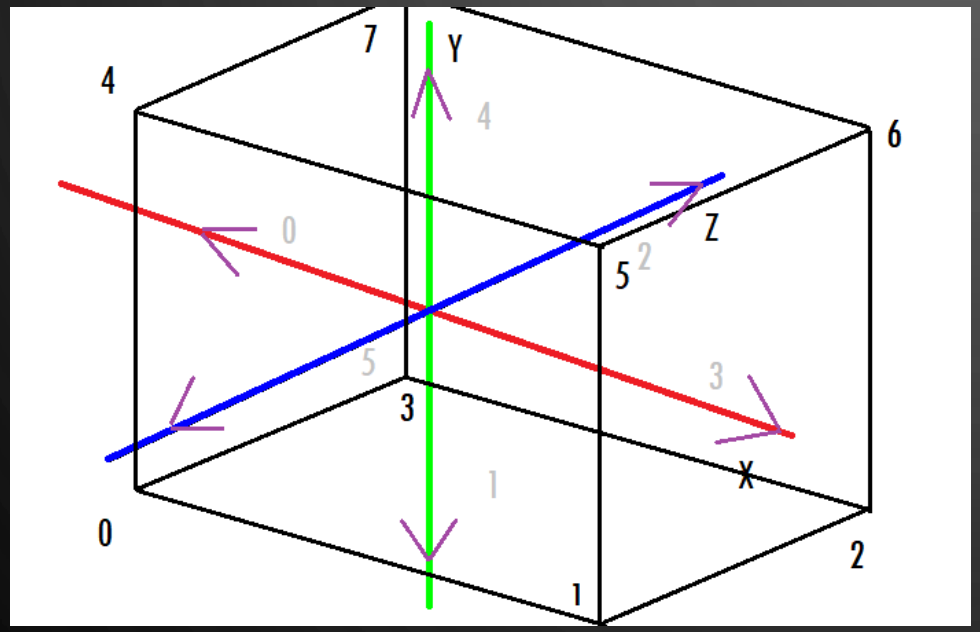
1. To bind a texture onto a surface of a polygon, each corner of a texture has to render together with each vertex of a polygon.

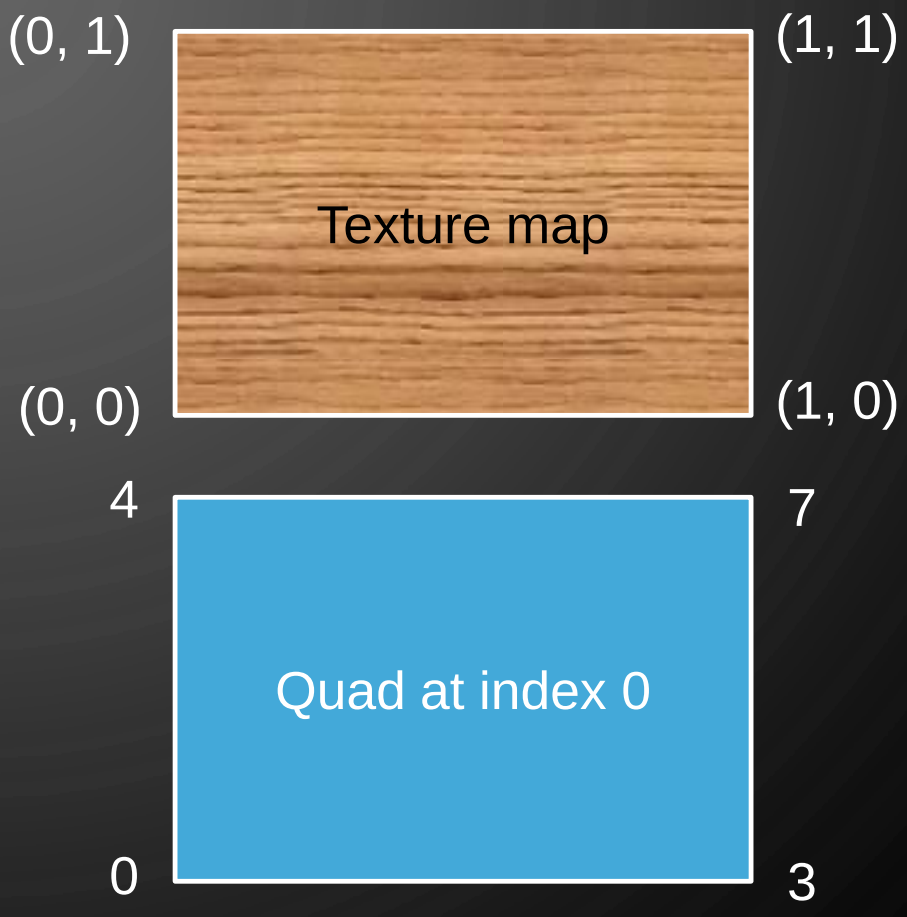
2. There is a need to specify either clockwise (CW) winding or counter-clockwise (CCW) winding as front face.

BOX



BOX

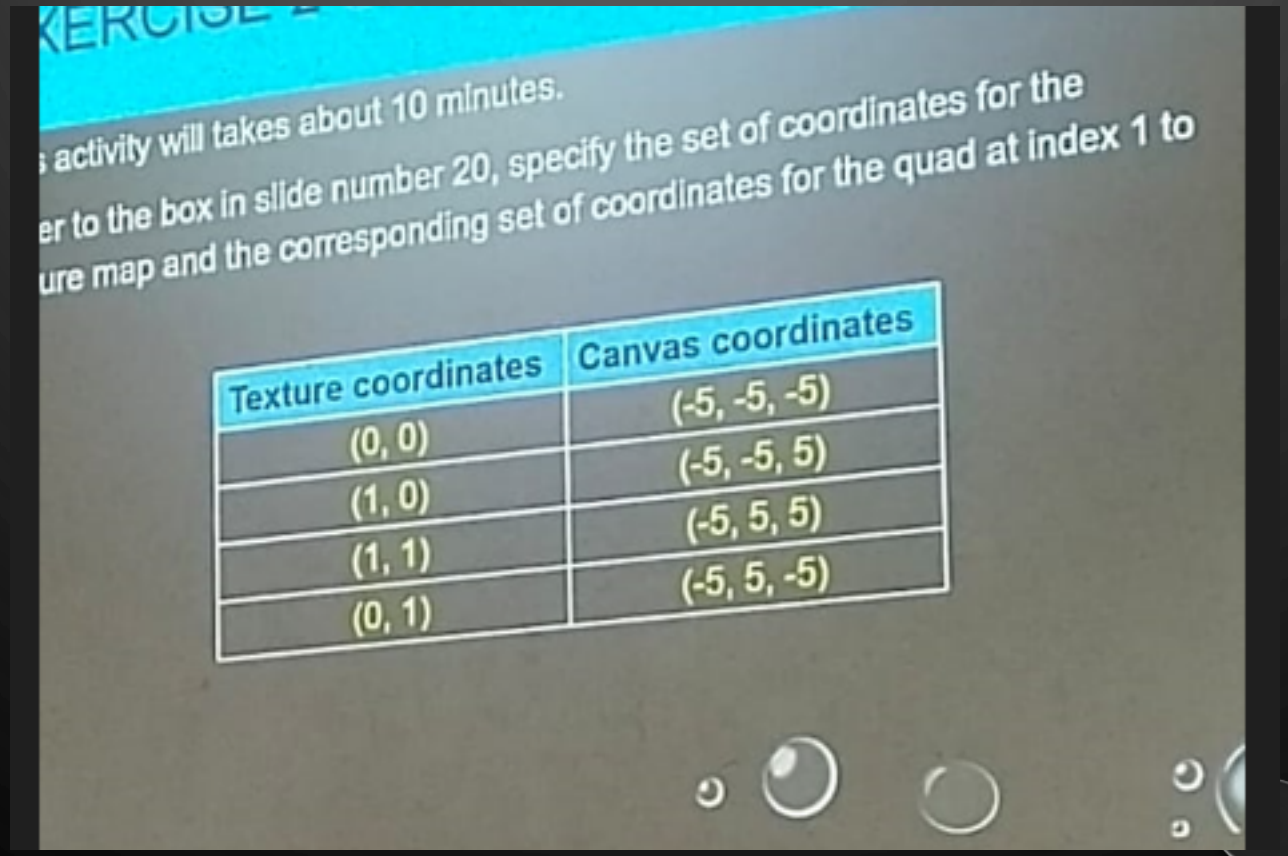

7
Y
4
6
0
Z
5
2
5
3
3
X
1
0
1
2


(0, 1)
(1, 1)
Texture map
(0, 0)
(1, 0)
4
7
Quad at index 0
0
3

EXERCISE 2 (15 MINUTES)

This activity will takes about 10 minutes.

Refer to the box in slide number 20, specify the set of coordinates for the texture map and the corresponding set of coordinates for the quad at index 1 to 7.



BREAK

. . .

MINI LECTURE 3
TEXTURE FUNCTIONS IN OPENGL

. . .

TEXTURE IMAGE

Function name	glTextImage2D
Purpose	Specify a two-dimensional (2D) texture image.
Arguments or parameters	Target; Level; BitDepth; Width; Height; Border; Format; DataTypes; Data;
Return value	None.

TEXTURE BINDING

Function name	glBindTexture
Purpose	Bind a named texture to a texturing target.
Arguments or parameters	Target; Texture, specify the name of a texture;
Return value	None.

EXERCISE 3 (15 MINUTES)

This activity will takes about 10 minutes.

1. State all the target values to be used in glTexImage2D.

2. State all the target values to be used in glBindTexture.

MINI LECTURE 4
TEXTURE BINDING USING OPENGL

. . .

CODE TO SETUP TEXTURE MAPS

To setup a texture map.

```
glBindTexture(targetValue, textureMapName);

glTexImage2D(targetValue, bitDepth, width, height, 0, format,
datatype, imageData);

glEnable(targetValue);
```

EXAMPLE CODE TO BIND TEXTURE

To bind a specify texture map.

```
glBindTexture(targetValue, textureMapName);
glBegin(GL_QUADS); // Quad defines each vertex in CCW.
  // glNormal2f(1.0f, 0.0f); //Normal vector not in 2D context!
  glTexCoord2f(0, 0);  glVertex2f(250, 50);
  glTexCoord2f(1, 0);  glVertex2f(50, 50);
  glTexCoord2f(1, 1);  glVertex2f(50, 250);
  glTexCoord2f(0, 1);  glVertex2f(250, 250);
glEnd();
```

EXERCISE 4 (15 MINUTES)

This activity will takes about 10 minutes.

1. Write a segment of code in C++ OpenGL to load an image for texture binding.

2. Write a segment of code in C++ OpenGL to generate a checker image for texture binding.

REFERENCES

Main reference:

Hajek, D. (2019). Introduction to Computer Graphics 2019 Edition. Independently Published.

Additional reference:

Marschner, S. and Shirley, P. (2021). Fundamentals of Computer Graphics, 5th Edn. CRC Press: Taylor's & Francis.