CT404/CT336 Marking Scheme, 2021-22

Q.1. (Graphics)

(i)

<u>If using Canvas2D:</u>

Use of context.translate(x,y).	[3]
Display of some graphics in this transformed coordinate system	[1]
Use of context.rotate(angle)	[2]
Display of some graphics in this transformed coordinate system	[1]
Use of context.scale(x,y).	[2]
Display of some graphics in this transformed coordinate system.	[1]

or

<u>If using Threejs:</u>

Instantiation of a mesh object (or properly explained assumption that it exists)	[3]
object.position.set(x,y,z) or direct assignment of values to object.position.x etc.	[3]
object.rotation.set(x,y,z) or direct assignment of values to object.rotation.x etc., or use of object.rotateOnAxis(axis, angle).	[2]
object.scale.set(x,y,z) or direct assignment of values to object.scale.x etc.	[2]

(ii)

Draw grey box with context.fillRect	[1]
Identification of highest data value in array	[2]
Calculation of x axis scale by dividing canvas width by array length.	[1]
Calculation of y axis scale by dividing canvas height by highest data value	[1]
Calculation of correct x,y position for each data value on the canvas.	[2]
Use of context.beginPath	[0.5]
Use of context.moveTo for the 1st data value.	[1]
Use of context.lineTo for the subsequent data values.	[1]
Use of context.stroke.	[0.5]

Q.2. (Graphics)

(i)

Definition of surface normal [2]
Relevance to calculation of incident light rays angle to surface [2]
Explanation of lambert shading, with diagram and explicit reference to surface normal [2]
Explanation of gourard shading, with diagram and explicit reference to surface normal [2]

(ii)

Radiosity as brute-force ray tracing approach starting from patches of emitted light and involving interaction with surfaces [1]
Radiosity: relevance of surface reflection properties (bouncing and scattering) and materials (alteration of light colour) [1]
Radiosity: pre-runtime calculation of baked texture/shadow map [1]
Ambient occlusion as a simulation of light occlusion via amount of nearby geometry [1]
Ambient occlusion: to include the notion of raycasting/raytracing against world (or local) geometry [1]
Ambient occlusion: pre-runtime calculation of baked AO map [1]

(iii)

Specular colour: colour of specularly reflected light [1]
Definition of specularly reflected light (bouncing on a shiny surface) via diagram [1]
Diffuse colour: colour of diffusely reflected light [1]
Definition of diffuse reflection (scattering on non-shiny surface) via diagram [1]
Ambient lighting: estimation of complex realtime lighting via a summation of 'ambient' values per light source [1]
Ambient light diagram [1]

Q.3. (Graphics)

(i)

<u>Frustum Culling:</u>
Explanation of viewing frustum, with field of view and near/far clipping planes [2]
Helping the polygon budget by only rendering if inside viewing frustum [1]
<u>Bump Mapping:</u>
Mention of Lambertian lighting (angle of light incidence to surface normal), and modification of surface normal from data in raster image [1]
Mention of mapping via raster image to pixels on a polygonal surface [1]
Helping the polygon budget by faking (via lighting) a complex surface [1]
<u>Back Face Culling:</u>
Do not render polygons that face away from the camera [1]
Applicable to Convex objects [0.5]
Method of calculation via vector dot product. [0.5]
Helping the polygon budget by only rendering those that can be seen. [1]
<u>Billboards:</u>
Single polygons always facing the camera [1]
Textures with opacity. [1]
Helping the polygon budget by being just 1 polygon which looks like something more complex [1]
<u>Levels-of-Detail (LODs):</u>
Reduced polygon-count versions of a mesh. [1]
Selected at runtime according to size of object on screen [1]
Helps the polygon budget by using less polygons for distant objects [1]

(ii)

Flat Shading: Lambertian calculations (cosine of angle of incident ray to surface normal) [1]
Gourard Shading: Bilinear interpolation of shading across surface, from Lambertian calculations performed from normal at vertices [1.5]
Phong Shading: Interpolation of normal across surface, with a separate shading calculation at each interpolated normal [1.5]
Diagrams [1]

Q.4. (Image Processing)

(i)

At least one example of an erosion template [1]
Explanation of how an erosion template is applied to each pixel in the input image, to produce an output image [2]
Erosion effect on binary images [1]
Dilation as the inverse of erosion [1]
Dilation effect on binary images [1]
Opening as a concatenation of erosion+dilation, and effect on binary images [1]
Closing as a concatenation of dilation+erosion, and effect on binary images [1]

(ii)

Dealing with noise: why and how [3]
Isolation using (edge detection, thresholding, Hough Transform: why and how) or using (morphology approaches: why and how – to include shine in centres) [6]
Final counting [3]

Q.5. (Image Processing)

(i)

Reference image definition as an image with known objects on [1]
Identification of these known objects as control points in reference image as captured by camera [2]
Explanation that we can define the movement (translation) of pixels from input to output [1]
Interpolation of all other pixels in captured image, via nearest control points [1]
Explanation of pixel carry-over/pixel filling to construct geometrically corrected image of the real world [1]
Interpolation of values during construction of corrected image [1]
Specific explanation of the relevance of high contrast markings to the image processing algorithm that will segment the reference image [3]

(ii)

Hough Transform: accumulation of votes for simple shapes, using an edge map (including definition of an edge map) [2]
The specific process of casting votes from each edge pixel, given a range of candidate circle radii [2]
How the accumulator array is queried, to produce output circles – including dealing with circles of different radius requiring different acceptance thresholds [2]
Why this approach is robust to noise, occlusion, and low contrast as indicated in the supplied image [2]
Specifically why other approaches (such as edge tracing) would fail in these situations [2]