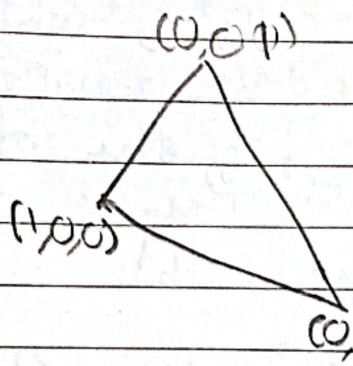
Graphics Supo 2 work

1.



$0 \leq \alpha, \beta, \gamma \leq 1, \quad \alpha + \beta + \gamma = 1$

Each vertex of the triangle has one of the barycentric coordinates as 1 and the other two as 0.

2. Diffuse and Specular as both use the surface normal in their calculations.

3.

	OpenGL	DirectX	Vulkan
Platform	Multi-platform	Windows / Xbox	Multi-platform
API type	Open Standard	Proprietary	Open Standard
Focus	General 3D applications	Games	High performance 3D graphics

Vulkan also reduces CPU load, gives better support for multi-CPU-core architectures, has finer GPU control when compared with OpenGL and DirectX. It's intended for game engines and other code that must be highly optimised since even drawing a few primitives can take 1000s of lines of code.

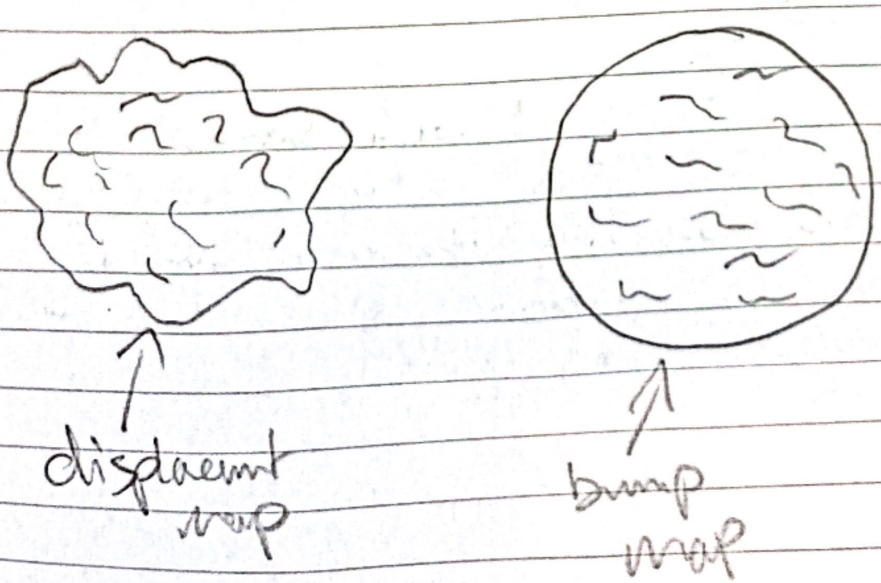
4. Using GPUs for general purpose computing by carry out tasks normally done by the CPU, on the GPU. It takes advantage of GPUs large number of cores to excel at parallel computing. OpenCL and CUDA are two APIs that can be used for GPGPUs.

5. "in" variables are inputs to the shader,
"out" are outputs from the shader
"uniform" variables are constants for the drawn geometry that are passed to the shader by the application (Java, C/C++).

The "in" variables are passed to the shader either by the application or by another shader's outputs.
The "out" values are set within the shader.
The "uniform" values are set by the application.

6. Blocky artefacts if using nearest-neighbour upsampling.
Blurry artefacts if using bilinear upsampling.

7. A sphere shaped object that's very bumpy.



8. Light spectra that are different but appear the same colour to us due to us only having 3 photoreceptor types to sense colour differences.

9. Displays can only reproduce a small subspace of all visible colours. Gamma correction reduces the number of bits for each colour channel from 12 to 8.

10.
- ITU-R 709
- HSV
- ITU-R 2020

11. Luma is the gamma corrected brightness of the pixels.
Luminance is the linear brightness of the pixels.

12. They are fast to compute and mimic the response of analog film which produces good tone-reproduction.

13. To trick the eye into thinking some areas are brighter than they actually are which by adding variance among intensities that are greater than what the monitor can display.

Longer Qs

1. It could be computed N times if all triangles overlapped and we went from furthest away to closest.

Render triangles in random order??.

2. Vertex shader -> Primitive assembly -> Clipping
↓
Fragment Shader <- Rasterization

Vertex shader: Processes vertices, normals and u, v texture co-ordinates

Primitive Assembly: Organizes vertices into primitives and prepares them for rendering

Clipping: Remove/modify vertices so they all lie within the view frustum.

Rasterization: Generates fragments to be drawn for each primitive. Interpolates vertex attributes

Fragment shader: Calculates colour for each fragment.

3. Array Buffer

It stores all the vertex data (colours, normals, positions, texture co-ordinates etc)

Element Array Buffer

Stores the indices of the vertex data, for each primitive, in the array buffer.

Vertex array

Old version of Array buffer that is less efficient since geometry information needs to be passed to the renderer every frame.

4. Without: 48, 36 for all triangle vertices and 18 for all triangle normals

With: 14, 8 for all vertices of cube and 6 for all normals.

5. 2D: Map each u,v co-ordinate to a spherical co-ordinate and wrap the sides.

3D: have the texture as a sphere and uniquely map each texture co-ordinate to a cartesian co-ordinate.

5. Cube map: map each texture co-ordinate from the cube to a spherical co-ordinate to simulate reflections from an environment.

6. More: Interpolate values using nearest-neighbour or bilinear interpolation. This is up-sampling.

Less: Average the texture across the area covered by the pixel. This is down-sampling. Use mipmaps to get closest resolution of texture to pixels.

7. Use a double buffer. The front buffer is always shown on screen and the back buffer is where the GPU draws the next frame. They swap after drawing is finished.
Triple buffering can increase speed but needs more memory and has a higher delay between drawing and displaying a frame.
V-Sync can prevent tearing by waiting for all rows of pixels to be copied from the colour buffer to the monitor before switching front and back buffers.

8. The colour matching functions create metamers of colours that our eyes see but only using Red, green and blue. This works because we only have l, m, s photo receptors for colour.

9. Linar gives the luminance which is perceptually less uniform of a brightness distribution.
Gamma-corrected gives the luma which is perceptually more uniform of a brightness distribution so requires less bits for storage of each colour channel.

10. The colour space. It is what the colour of "pure" red, green, and blue are.

11. Tone-mapping converts scene-referred colours into display-referred colours (colours the monitor can display). Display encoding converts the colours from a linar colour space to a gamma corrected colour space that has a perceptually more uniform brightness distribution.