Task 1.6: Blinn-Phong Shading with Point Lights

COMP3811 Computer Graphics – Coursework 2

Overview. This section documents the implementation of the full Blinn-Phong shading model for point lights with standard $1/r^2$ distance attenuation. Three coloured point lights are positioned around the space vehicle, affecting all scene objects (terrain, launchpads, and vehicle). The directional light from Section 1.2 is preserved.

Blinn-Phong Shading Model. The fragment shaders (`default.frag` and `material.frag`) implement the complete Blinn-Phong reflection model. For each point light, the lighting contribution is computed as:

$$L = (k_d \cdot \max(N \cdot L, 0) + k_s \cdot \max(N \cdot H, 0)^\alpha) \cdot \text{attenuation}$$

where N is the surface normal, L is the light direction, $H = \text{normalize}(L + V)$ is the half-vector between light and view directions, and α is the shininess exponent. The diffuse term k_d provides base illumination, while the specular term k_s creates highlights on glossy surfaces.

Distance Attenuation. Point lights employ standard inverse-square distance attenuation:

$$\text{attenuation} = \frac{1}{1 + d^2}$$

where d is the distance from the light source to the fragment. The constant term prevents division by zero when $d \approx 0$. This physically-based falloff ensures realistic light intensity decay with distance.

Light Configuration. Three point lights with distinct colours are positioned around the space vehicle:

Light	Offset from Vehicle	Colour (RGB)	Purpose
Point Light 1	$(-3, 2, 0)$	$(2.0, 0.6, 0.6)$ Red	Left side accent
Point Light 2	$(+3, 2, 0)$	$(0.6, 2.0, 0.6)$ Green	Right side accent
Point Light 3	$(0, 3, -3)$	$(0.6, 0.6, 2.0)$ Blue	Rear/top accent

The lights are offset slightly from the geometry (2–3 units) to avoid numerical issues with the attenuation calculation and to provide visible coloured illumination on vehicle surfaces. Light positions are stored as offsets from `state.vehicleOriginalPos` and updated dynamically during animation (Task 1.7).

Directional Light Preservation. The directional light from Section 1.2 remains active with direction $(0, 1, -1)$ normalised and colour $(1.0, 1.0, 1.0)$. This light uses the simplified ambient-plus-diffuse model (no specular) as specified in the original requirements, providing consistent base illumination across the entire scene independent of point light effects.

Shader Implementation. The `blinnPhong()` function in both fragment shaders computes per-light contributions. Uniforms include camera position (`uCameraPosition`) for view vector calculation, light properties (`uPointLights[3]`), and material shininess (`uShininess`). The helper function `set_lighting_uniforms()` in `main.cpp` uploads all lighting parameters to both shader programs, ensuring consistent illumination across textured terrain and material-coloured objects.

Global Application. All three point lights affect every rendered object: the terrain mesh (via `default.frag`), both launchpad instances, and the space vehicle (via `material.frag`). This unified lighting approach creates cohesive scene illumination with coloured highlights visible on nearby surfaces.

Screenshot.

[Figure 1.6.1: Space vehicle and launchpad illuminated by three coloured point lights]