

Task 1.5: Programmatic Space Vehicle Model

COMP3811 Computer Graphics – Coursework 2

Overview. This section documents the procedural construction of a 3D space vehicle composed of geometric primitives. The model incorporates eleven components from four primitive types (cylinder, cone, box, sphere), employs translation, rotation, and non-uniform scaling, and positions components hierarchically. The vehicle is placed on **Launchpad A** at world coordinates $(75, -1, 20)$.

Primitive Generation. Four mesh generators in `primitive_shapes.cpp` produce the geometric primitives: `generate_cylinder` creates closed cylinders with outward-facing normals; `generate_cone` computes face normals via cross products; `generate_box` produces cuboids with per-face normals; `generate_sphere` uses latitude-longitude tessellation with normalised position vectors as normals. All generators output positions, normals, and indices in `SimpleObjMesh` format.

Vehicle Composition. The vehicle comprises eleven interconnected parts generated by `generate_space_vehicle()`:

Component	Primitive	Dimensions	Colour
Main body	Cylinder	$r=0.8, h=8.0$	Light grey
Nose cone	Cone	$r=0.6, h=3.0$	Red-orange
Engine nozzle	Cylinder	$r=1.0, h=1.5$	Dark grey
Fins ($\times 4$)	Box	$0.15 \times 2.5 \times 1.0$	Blue
Window	Sphere	$r=0.5, y\text{-scaled } 0.6$	Cyan
Antenna	Cylinder	$r=0.1, h=1.5$	Yellow
Thruster pods ($\times 2$)	Cylinder	$r=0.3, h=1.5$	Orange

The main fuselage serves as the structural reference. The nose cone sits atop at $y=9.5$, demonstrating relative placement. Four fins are distributed radially at 90 intervals via composed rotation-translation matrices. The window applies non-uniform y -axis scaling (60%) before translation. Thruster pods rotate $\pi/2$ about the z -axis before lateral positioning.

Transformations and Normals. The `transform_mesh` function applies 4×4 homogeneous matrices to vertex positions. Normals are transformed using the inverse-transpose matrix, preserving perpendicularity under non-uniform scaling. This ensures correct Blinn-Phong shading response across all components.

Launchpad Placement. The vehicle is positioned on **Launchpad A** (sea platform from Task 1.4) with vertical offset $y=0.2$ so the engine base rests on the platform surface. Position stored in `state.vehicleOriginalPos` for animation support.

Screenshots.

[Figure 1.5.1: Front-oblique view of the space vehicle on Launchpad A]

[Figure 1.5.2: Elevated view showing radial fin arrangement and thruster pods]