Coursework 2 – COMP3811

Maya Ben Zeev 201897642 | Nadav Goldrat 201897706

Task 1.1 - Matrix/vector functions

Implemented the methods `Mat44f operator*` and `Vec4f operator*` for matrix and vector operations, `make_rotation_` method for generating a rotation matrices for the x, y, and z axes, `make_translation` method for creating a matrix that moves objects in 3D space, and `make_perspective_projection` method for creating a projection matrix that maps 3D points in camera space to their 2D projections on the canvas.

Tests were implemented for each method by comparison to known good values:
Basic Matrix Multiplication: ensured that the resulting matrix has the correct values in s the top left corner and bottom right corner of the resulting matrix.
Basic Matrix-Vector Multiplication: used a diagonal matrix with scaling factors.
Matrix Rotation (X, Y, Z): verified rotations by using the standard rotations (90 degrees for x and y, 60 degrees for z), and ensured that the diagonal values computed correctly (using approximations with margin deviation for float comparisons).
Matrix Translation: confirmed that the translation offsets are correctly added to the matrix on specific values.
Perspective Projection: ensured that the projection matrix correctly projects 3D coordinates into the normalized device coordinates with comparing the matrix diagonal, and additional matrix cells (using approximations or float comparisons).

Task 1.2 - 3D renderer basics

Implemented a 3D rendering application using OpenGL to display a Wavefront OBJ model (langerso.obj), with perspective projection and a simplified directional light model combining ambient and diffuse components. For navigation, a "first person" shooter camera was implemented, controlled byd by WSAD and EQ keys for movement, Shift and Ctrl keys for speed adjustments, and mouse input for activation and orientation.
The application was developed and tested on a system with the following OpenGL specifications:
GL_RENDERER: NVIDIA GeForce RTX 4070/PCIe/SSE2
GL_VENDOR: NVIDIA Corporation
GL_VERSION: 4.3.0 NVIDIA 565.57.01

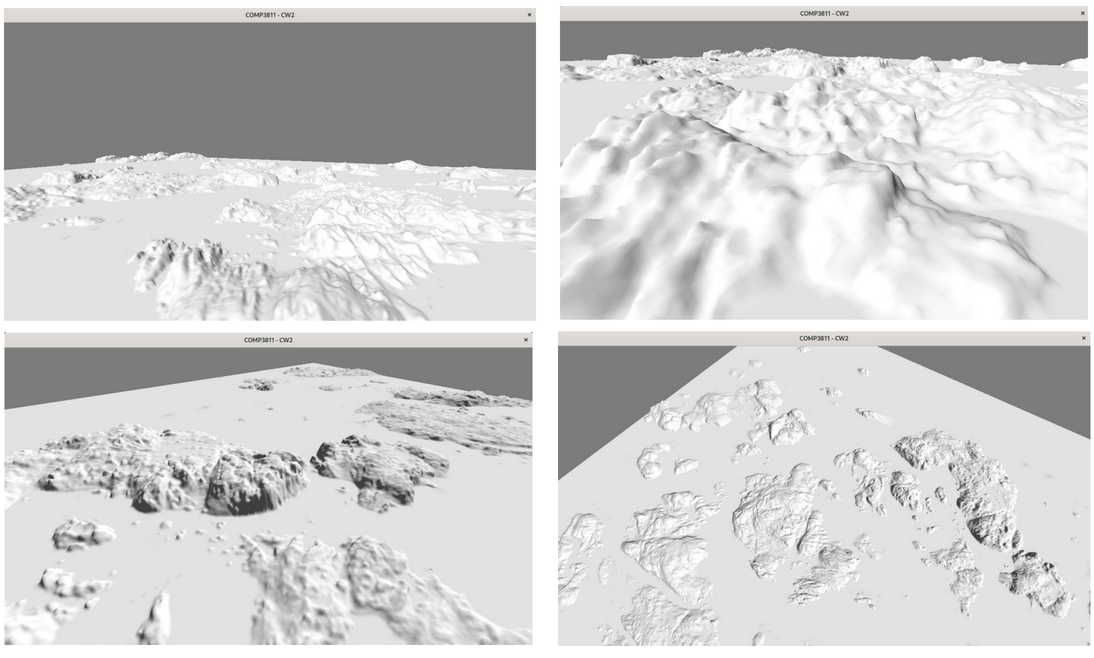


Fig 1

Task 1.3 - Texturing

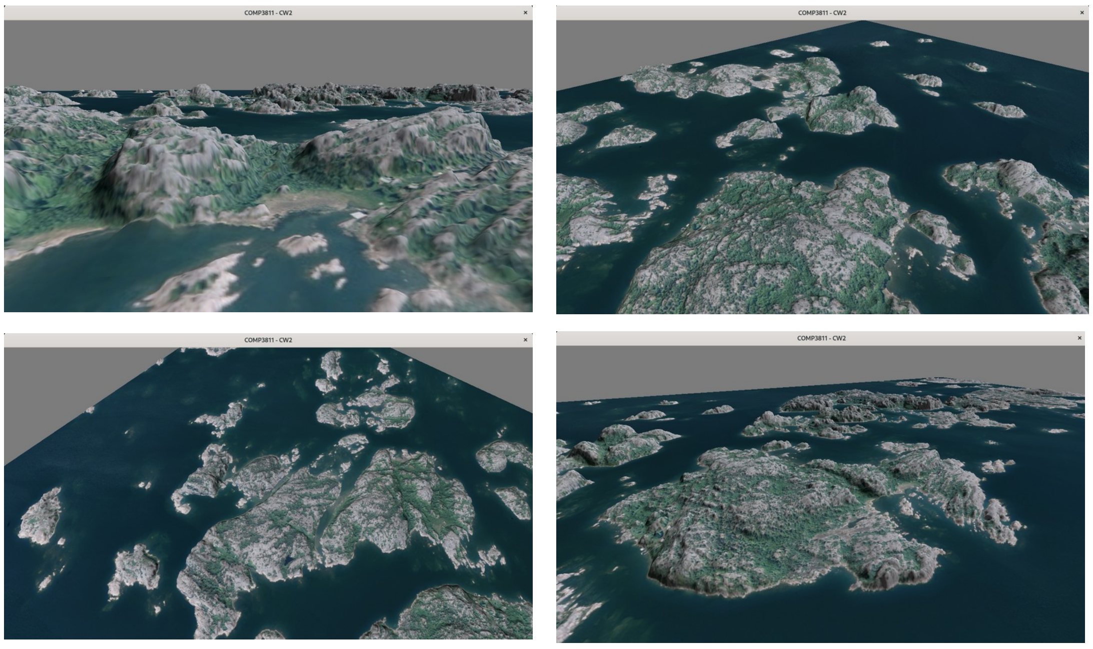


Fig 2

Task 1.4 - Simple Instancing

Launchpads were placed in the following positions:
Launchpad 1: Located at coordinates (5, 0, -5). This launchpad is positioned in the open sea area, in a place that is immediately visible to the camera when the simulation loads.

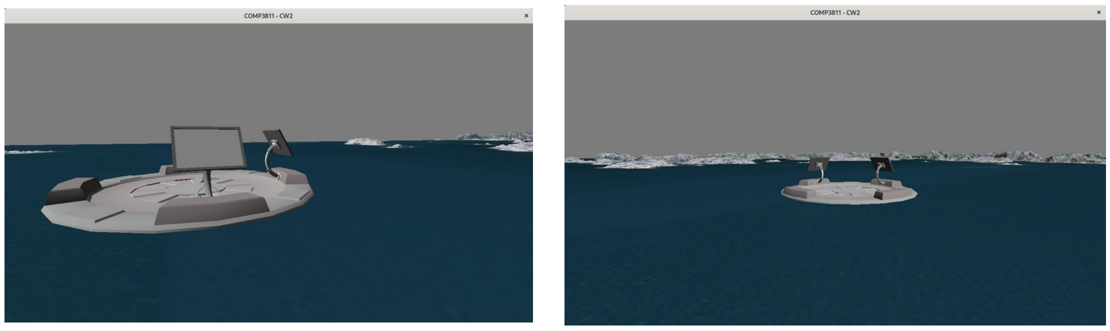


Fig 3

Launchpad 2: Located at coordinates (-2.1, 0, 1.1). This launchpad is positioned further back relative to the initial camera position and is in-between mountains. Reaching this launchpad requires navigation to the water gap. Both launchpads are located on top of the sea surface, making the y coordinate 0.

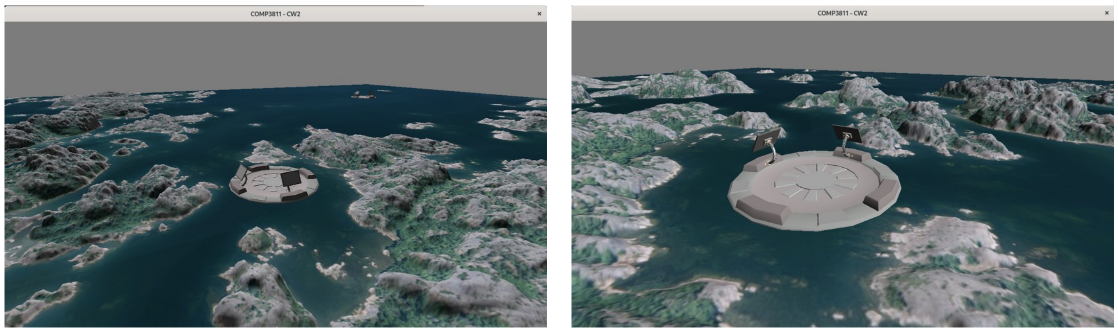


Fig 4

Fig 5 - top-down view of the scene - launchpad positions are marked in red

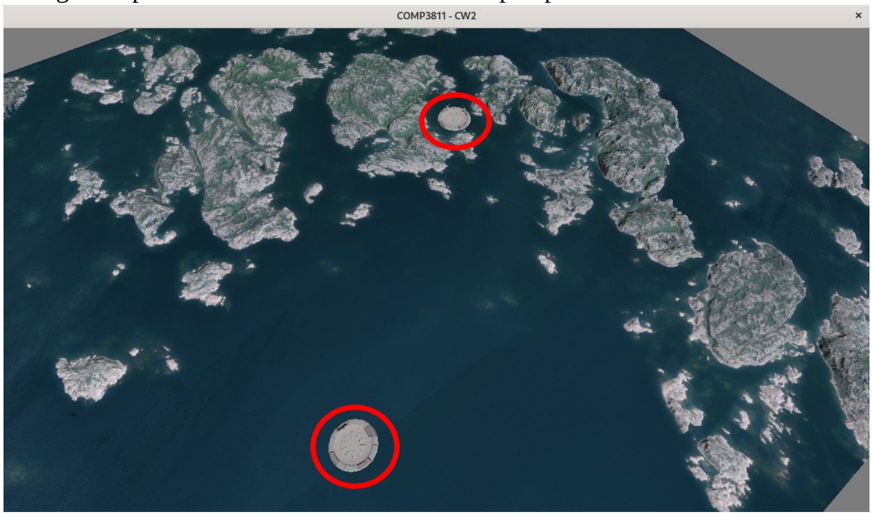
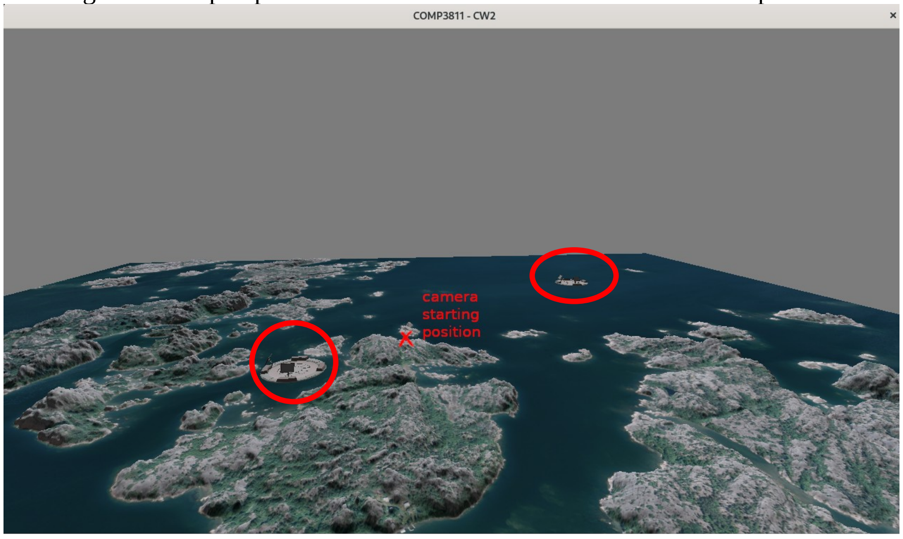


Fig 6 - launchpad positions in relation to the camera initial scene position



Task 1.5 - Custom Model

Created a spaceship vehicle made up of 9 parts: a cylindrical body with a cone-shaped top, 2 additional cylinders with conical tops for the boosters, and 3 cuboid shaped legs.
The Spaceship was placed on top of launchpad 1 (located at coordinates (5, 0, -5)).

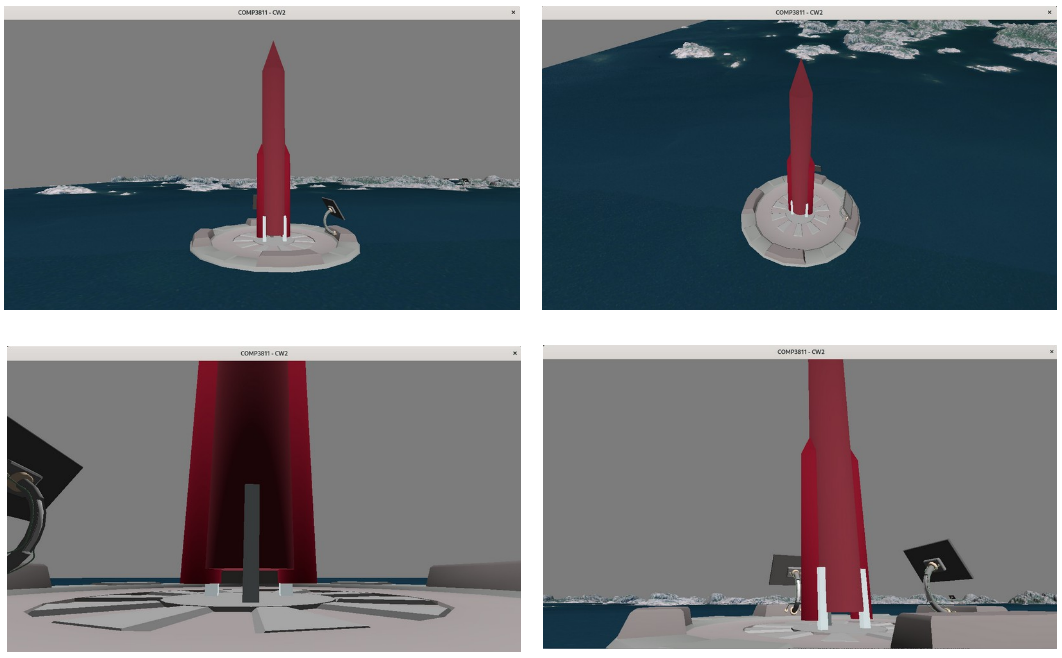


Fig 7

Task 1.7 - Animation

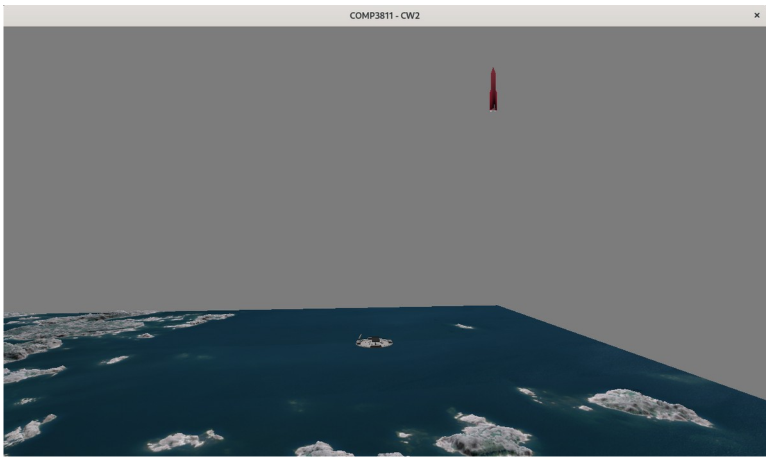
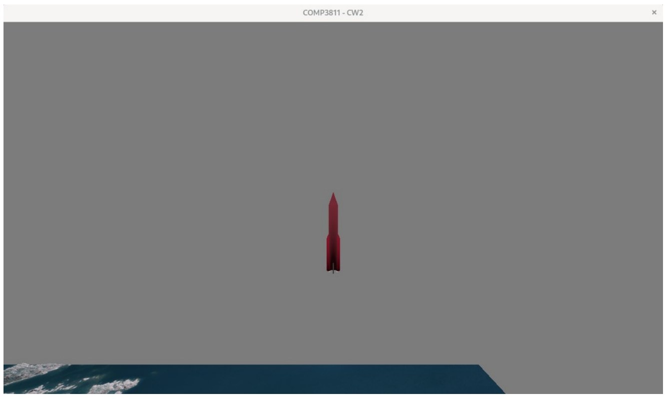


Fig 8

Task 1.8 - Tracking cameras



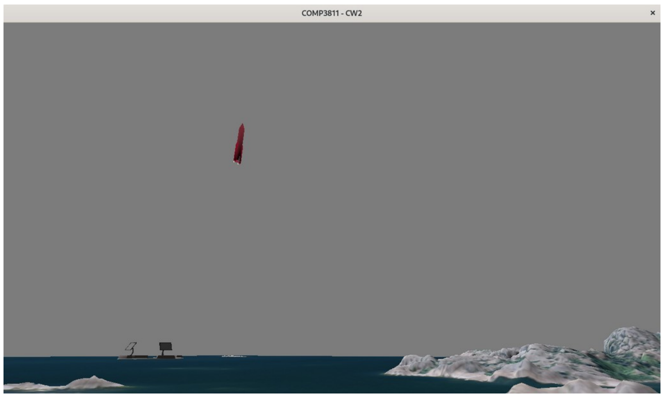


Fig 9.1 - camera 2:
looks at the space vehicle from a fixed
distance and follows it in its flight

Fig 9.2 - camera 3:
looks at the space vehicle from a fixed
distance and follows it in its flight

Appendix

Tasks contribution table

Task	Maya Ben Zeev	Nadav Goldrat
1.1	20%	80%
1.2	50%	50%
1.3	50%	50%
1.4	100%	0%
1.5	80%	20%
1.7	70%	30%
1.8	40%	60%