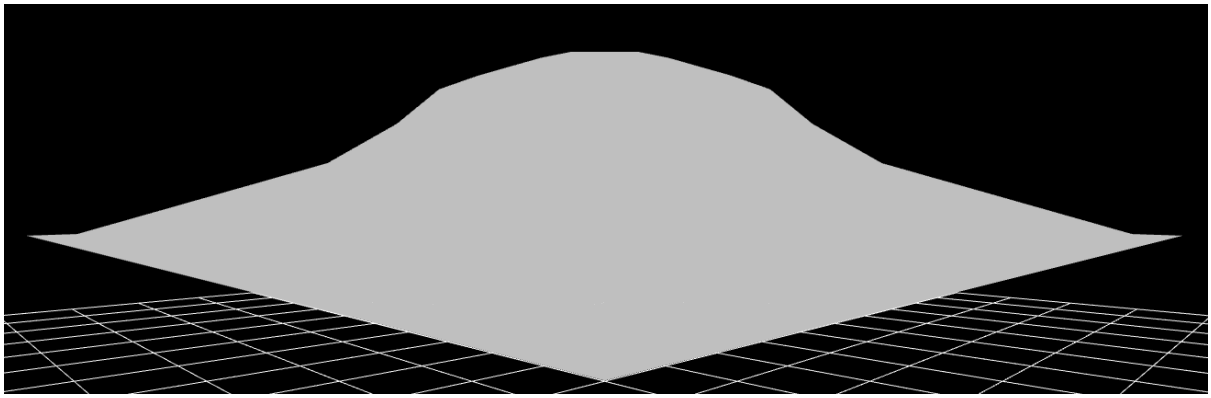
COMP330 Assignment 1 Report

Name: Benjamin Rendell
Student ID: 44655010

Features implemented in this assignment:

Feature	**Mark**	**Check if used**
Height map modelling	20%	X
Trees	5%	X
Textured terrain	10%	X
Multiple textures	5%	
Player movement (simple)	10%	X
Player movement (complex)	15%	
First-person perspective camera	10%	X
+ zoom	5%	X
Third person orthographic camera	5%	
+ zoom	10%	
Directional light	5%	X
Point light	5%	
Smooth shading	10%	
Screen-space effects	5%	
Transparency	5%	
TOTAL (max 100%)		

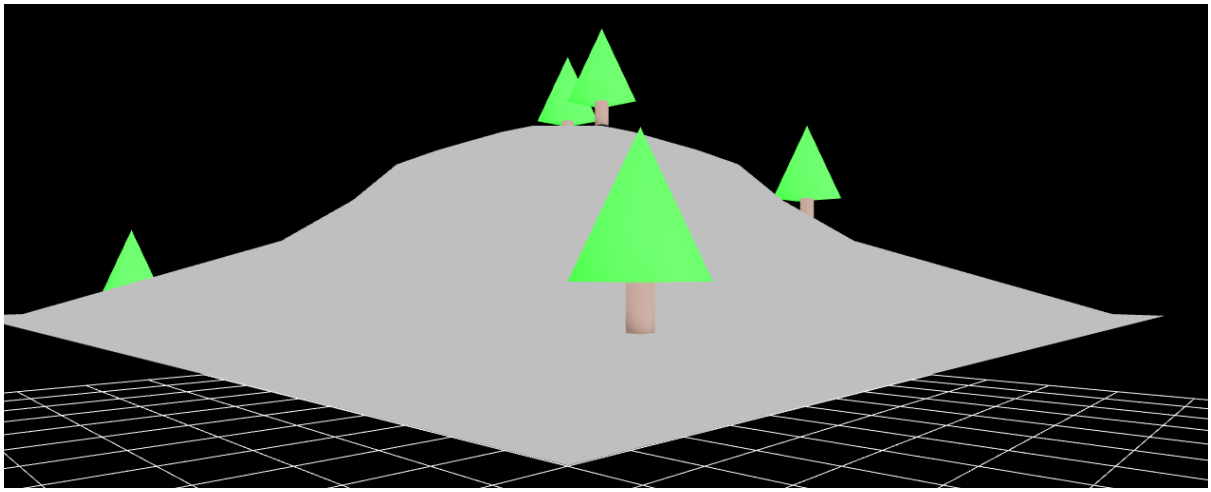
Heightmap Terrain



Implemented in:

- terrain.js:1-111 – initialisation and render code for terrain and heightmap

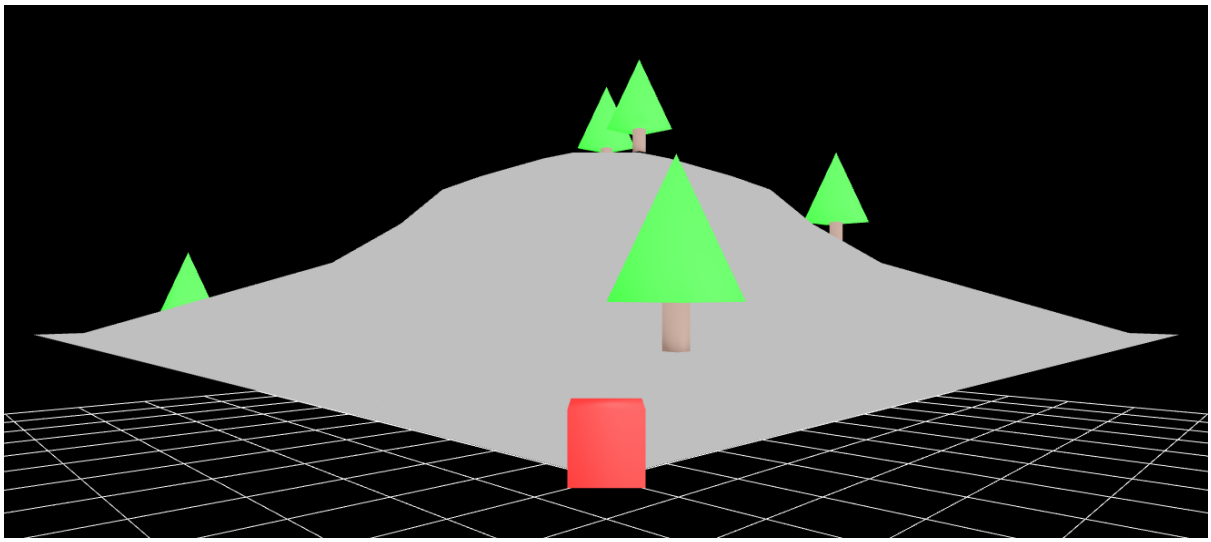
Trees

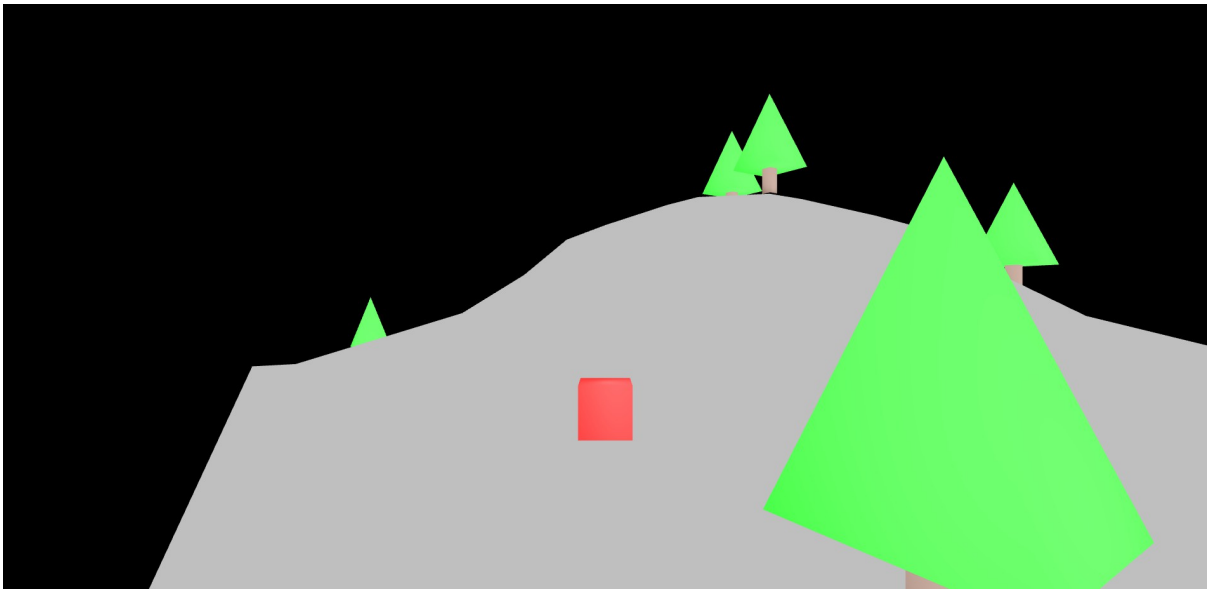


Implemented in:

- tree.js:1-179 – initialisation and render code for a single tree
- main.js:166-178 – store tree count in array and set their y positions on heightmap
- main.js:260-262 – render tree array

Simple Player Movement

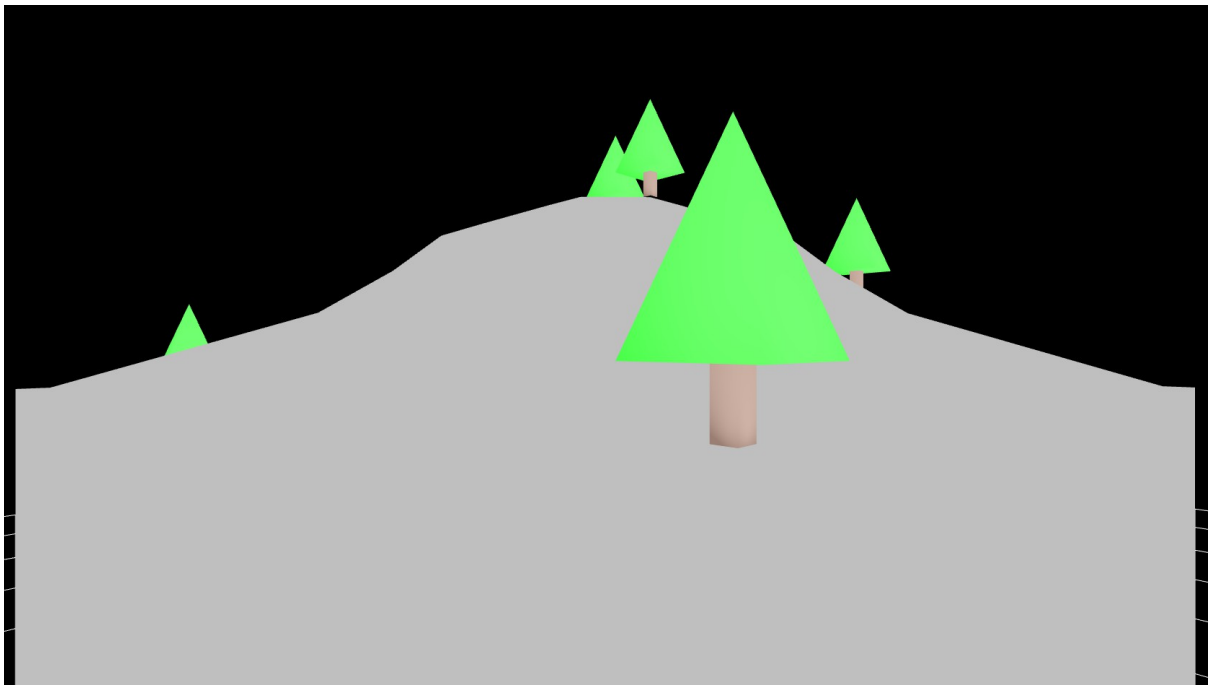




Implemented in:

- player.js:1-95 – initialisation code for player
- player.js:88-95 – initialises player's current position & heading using .json file
- player.js:140-167 – render code for player
- input.js:1-87 – Registers keyboard inputs using a global superclass inputManager for inserting controls to the scene.
- player.js:100-114 – player control logic & math, player translates based on current yaw rotation.
- player.js:116-128 – level bounds logic, prevents player from moving outside of terrain.
- player.js:130-140 – calculate player's height based on their current position using the heightmap.

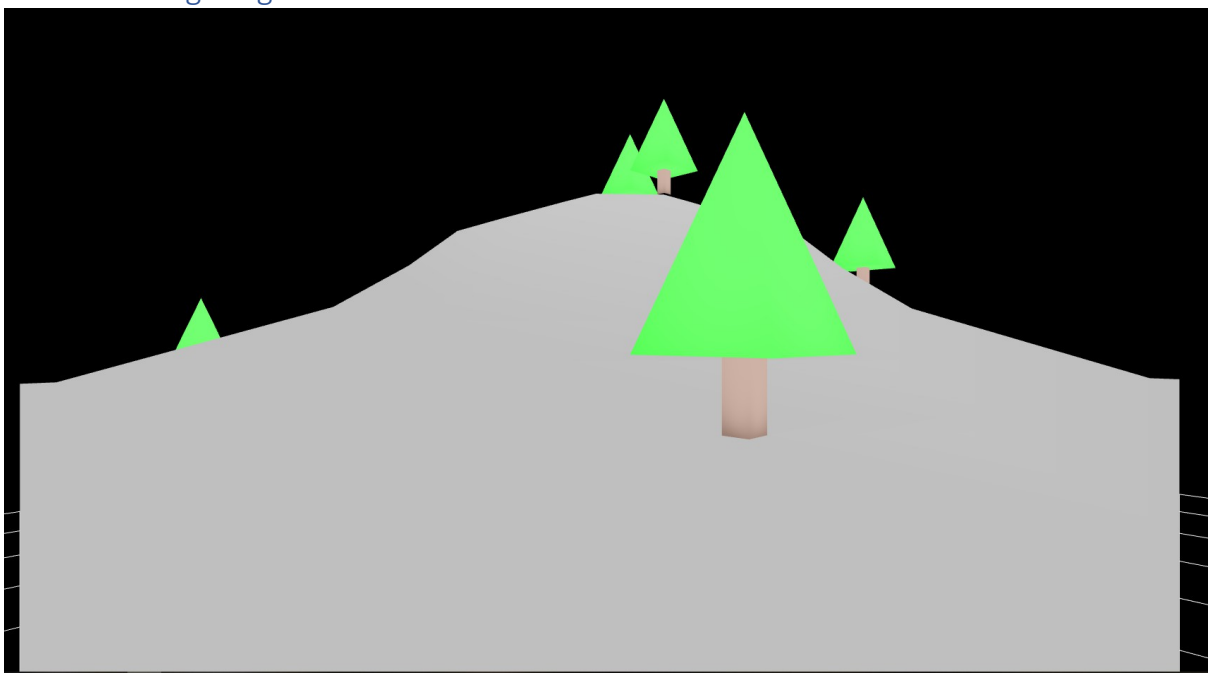
First-Person Camera



Implemented in:

- player.js:173-198 – initialisation and render code for an empty object
- main.js:194-207 – camera update math & logic to move camera based on player
- main.js:232-241 – calculate camera's projection matrix
- main.js:244-251 – calculate player's view matrix; bind cameras focus point to the empty object.
- player.js:181-186 – math and update logic to move empty object at center of the player's view port.

Directional Lighting

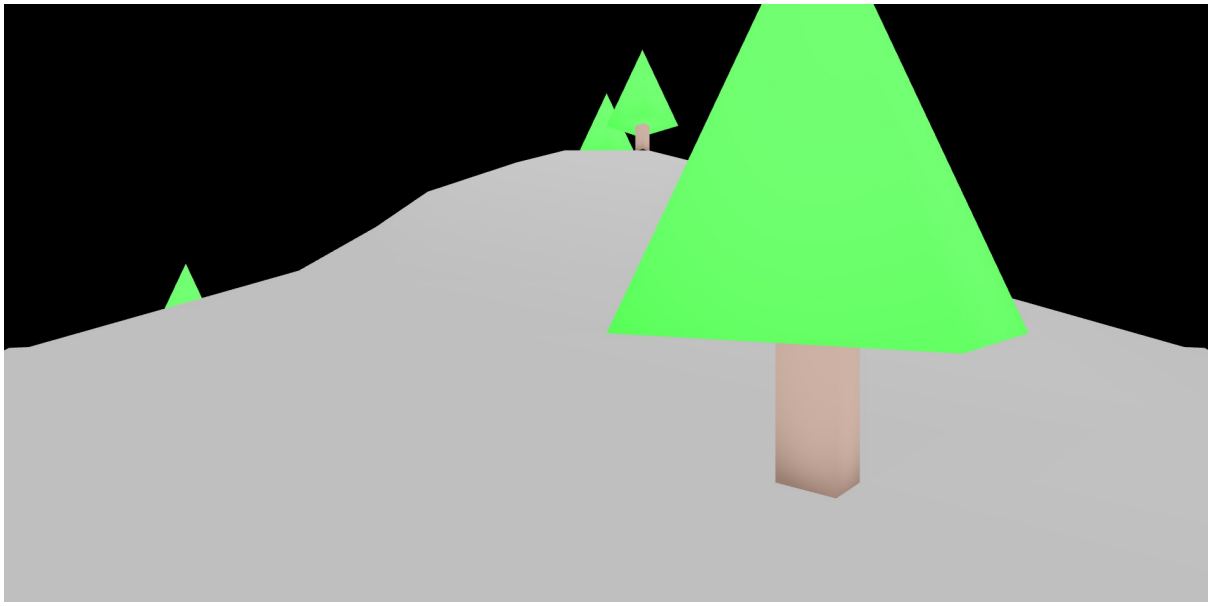


All the screenshots were taken after lighting was implemented so all the screenshots are displaying my final lighting model

Implemented in:

- main.js:41-70 – fragment shader code
- main.js:56-62 – basic math for implementing directional lighting.
- main.js:146-152 – initalises shader location, intensity, reflection and ambience variables. These control where and how the directional light is displayed. I've chosen to shine the light across the vector(-1, 5, 0), this angles the light as seen in the image above.

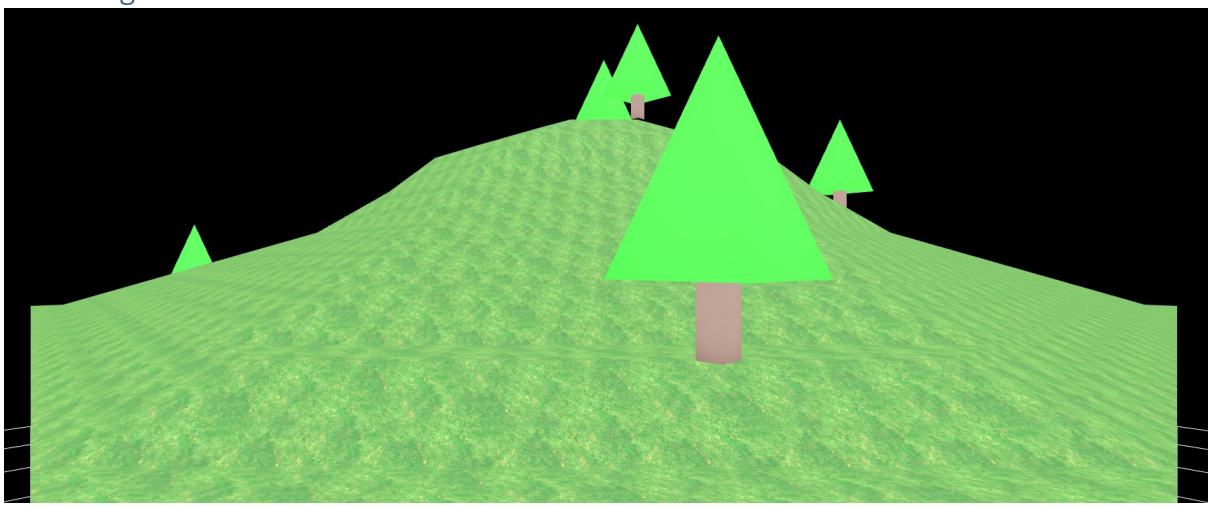
Zoom



Implemented in:

- input.js:66-84 – onwheel event detection
- player.js:181-186 – empty object position offset
- main.js:194-214 – alters camera distance variable based on mousewheel flags and emptyOffset value.

Texturing



Implemented in:

- main.js:5-23 – 2D uv coordinates
- main.js:41-70 – parse 2D uv coordinates to texture2D then apply colour and shading
- texture.js:10-25 – fixed variable names and values to enable both textures and colours to render.
- terrain.js:12-109 – applies uv mapping to each terrain tile to enable the terrain to render.
- main.js:154-263 – instantiates and parses textures to render functions

- tree.js:173-176 – applies blank texture, to enable rendering
- grid.js:72-75 – applies blank texture, to enable rendering
- player.js:165-168 – applies blank texture, to enable rendering