# Task 1: Non-Power-of-Two Texture Mapping

## Implementation

The setTexture function was updated to support textures of any size. WebGL typically requires textures to have dimensions that are powers of two to utilize certain features like mipmapping and wrapping. To accommodate this, we adjusted the texture parameters:

* For non-power-of-two textures, we used gl.CLAMP\_TO\_EDGE for the texture wrapping mode and gl.LINEAR for the texture filtering.
* For power-of-two textures, we maintained the existing configuration, enabling mipmaps for improved performance and visual quality at the cost of additional memory.

A computer screen shot of a program code

Description automatically generated

# Task 2: Basic Lighting Model

## Implementation

For the basic lighting model, we modified the MeshDrawer class and the fragment shader:

* A screen shot of a computer program

  Description automatically generatedThe MeshDrawer constructor was augmented with uniforms for the light position and color.
* A screen shot of a computer program

  Description automatically generatedThe setMesh function was updated to include normal data in its buffer setup.
* A screen shot of a computer program

  Description automatically generatedThe draw function was extended to update these uniforms before rendering.
* A computer screen shot of a black screen

  Description automatically generatedIn the fragment shader (meshFS), we added ambient and diffuse lighting calculations. The ambient light was implemented as a constant color multiplied by the base color of the mesh, and the diffuse light was calculated using Lambert's cosine law, depending on the angle between the light direction and the mesh surface normal.