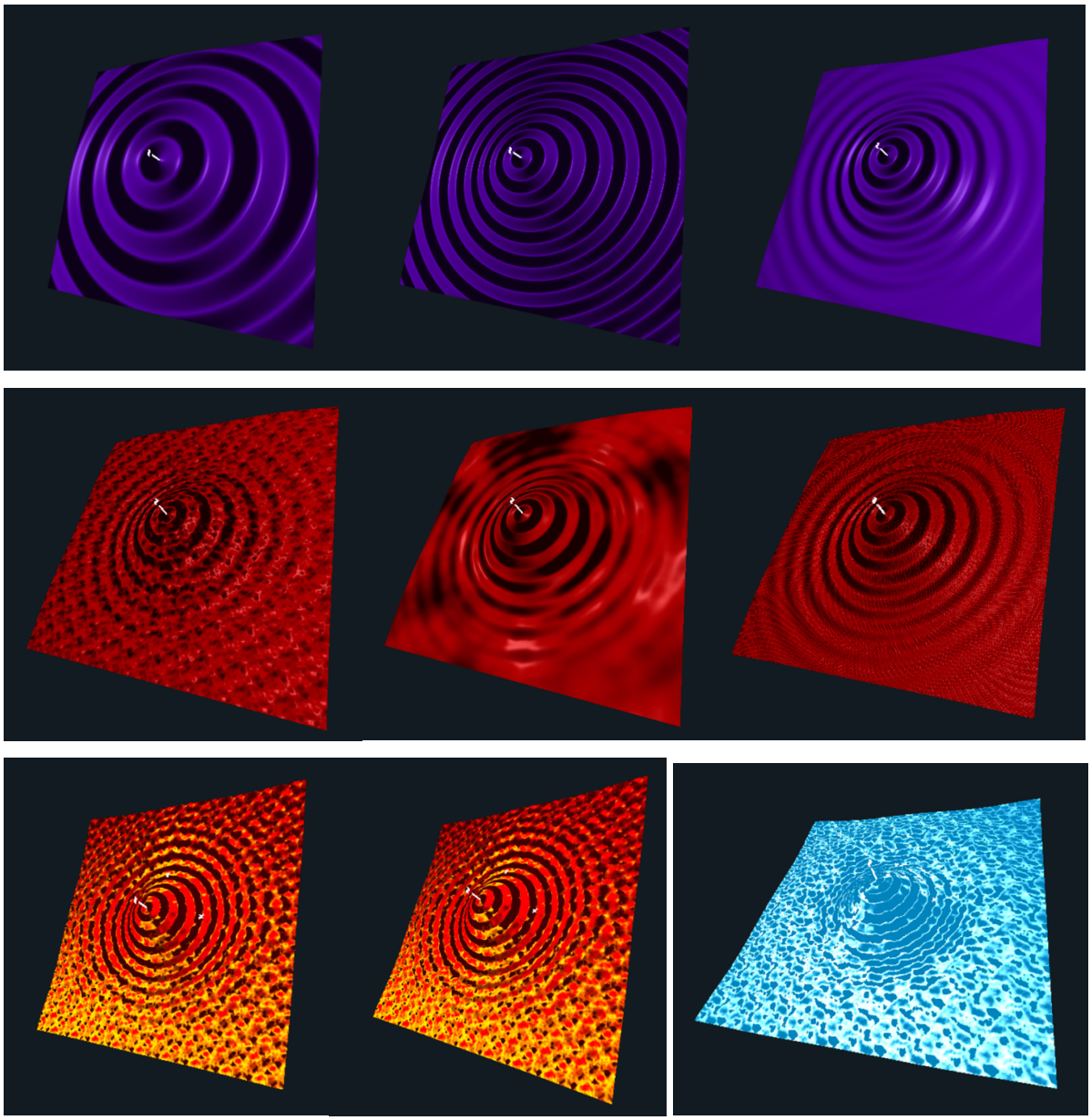
Name : Sanketh Karuturi

Email ID: karutusa@oregonstate.edu

CS 557 *Computer Graphics Shaders*

Project #3: Displacement Mapping, Bump Mapping, and Lighting

1) Project Screenshots



2) **What you did and explaining why it worked this way**

- **Implemented Displacement Mapping**:
    - Used the **rock.vert** vertex shader to alter the geometry of a simple shape by displacing its vertices based on noise functions.
    - This method worked by dynamically adjusting vertex positions before rendering, creating a more complex and detailed surface without needing a more complex model.
- **Re-computed Normals**:
    - Adjusted normals in the vertex shader (**rock.vert**) after displacement to ensure accurate lighting calculations.
    - Recomputing normals is essential after displacement mapping because the original normals of the simple shape would no longer correctly represent the new surface angles, affecting the lighting realism.
- **Applied Bump Mapping**:
    - The **rock.frag** fragment shader was used to simulate surface detail on the displaced geometry through bump mapping, utilizing noise textures.
    - Bump mapping alters the surface normal calculations per-fragment for lighting, giving the illusion of detailed textures.
- **Implemented Per-fragment Lighting**:
    - Calculated lighting in the **rock.frag** fragment shader using the re-computed normals and light source information (including ambient, diffuse, and specular lighting components).
    - Per-fragment lighting provides a more realistic representation of how light interacts with surfaces, especially with dynamically altered normals and bump mapping, by calculating the light interaction for each pixel.

These techniques collectively contributed to the project's goals by enhancing a basic geometric model into a more visually interesting and realistic object, demonstrating a deep understanding of advanced graphics programming concepts.

3) **Media Link : https://media.oregonstate.edu/media/t/1_jp8182cx**