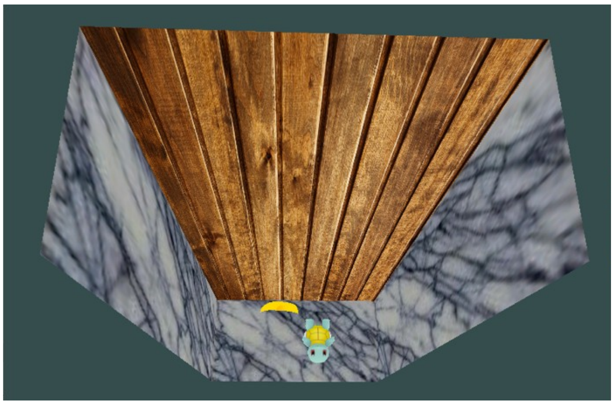
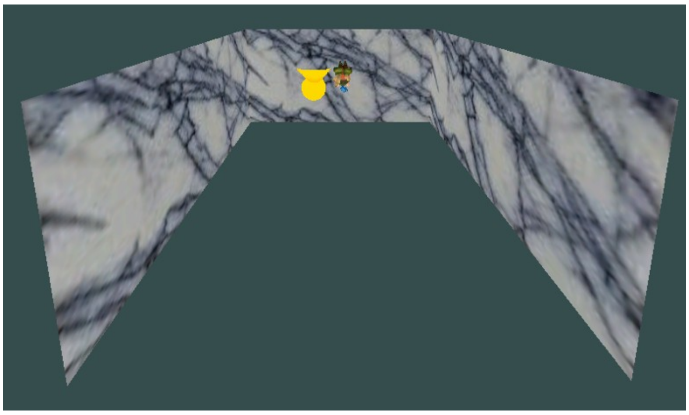
Water Simulation using OpenGL

- Group members:
  唐仕超 0656639
  周彥儒 0656151

- Goal:
  Render rippling, reflective, refractive water using OpenGL. We have completed all of our original goals.

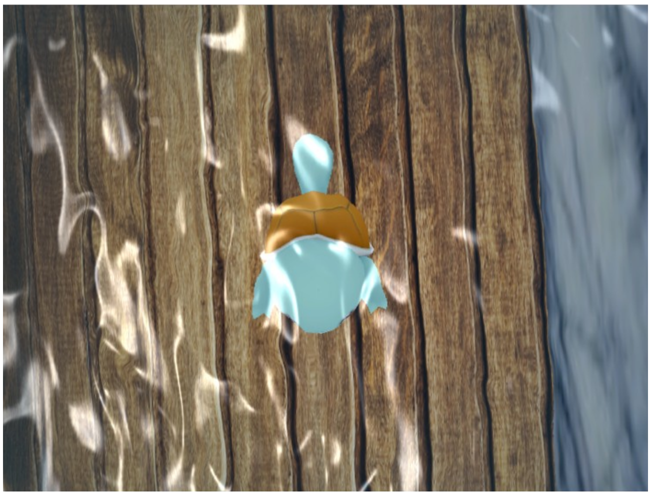
- Implementation:
  1. **Render quad:**
     First, the water is rendered as a flat quad.
  2. **Get reflection and refraction textures:**
     Next, create two frame buffer objects (FBOs) and attach a color and depth buffer to each FBO. These two FBOs will be used to render the reflection and refraction texture. We render the refraction FBO from the original camera position, and clip everything above the water surface to capture the image beneath the water. For the reflection FBO, we move the camera underneath the water's surface by its position above the water while inverting the camera pitch. Then, we clip everything below the water surface and render the image above the water. Note that although we do not use the depth buffer texture in this project, it can be used to create some depth effects, such as murky water.

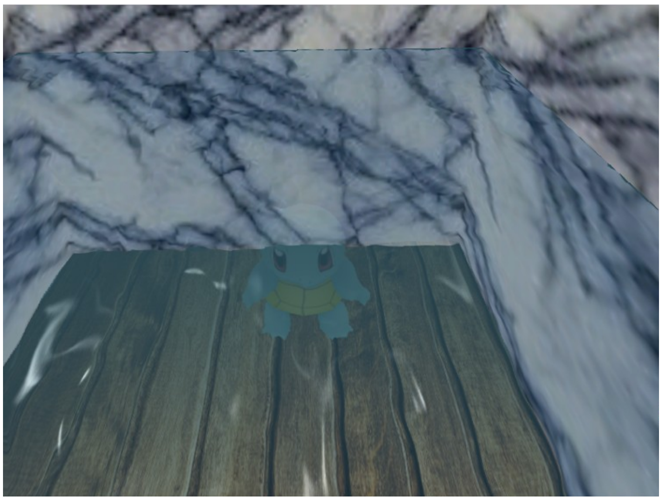




3. **Fresnel effect:**

   We mix the reflection and refraction textures according to the Fresnel effect. If the camera is viewing the water from a lower angle, then we should see more of the reflection texture. Similarly, if the camera is viewing the water nearly perpendicularly, then we should see more of the refraction texture.





4. **Projective texture mapping:**

   Now that we have both the reflection and refraction textures, we need to do projective texture mapping to map the textures onto the surface of the water. For each point on the water's surface, to see the location of where we should sample from the reflection and refraction textures, we first need the point of the water's surface in normalized device space. After that, we need to transform the coordinates we just got to the texture's coordinate system.

5. **Rippling effect:**

   DuDv maps are textures with red and green colors which can represent 2D vectors. Since all color components are between 0 to 1, we will map it to between -0.5 and 0.5 since we do not want to only add offsets in one direction. To create the effect that the water has a slightly distorted surface, we add it as an offset to the final coordinates we got from the previous step. This changes the texture sampling location. However, in order to simulate the rippling effect instead of only static distortion, we continuously change where we sample from the DuDv map by adding different offsets in both dimensions. Finally, in order to make sure that the coordinates that we are sampling the reflection and refraction textures are not larger than one, which causes the texture to wrap around, the clamp the edges to 0 and 1.

6. **Specular highlights:**

   Similar to homework 3, we use normal mapping to add specular highlights to the water's surface.

- Final results: