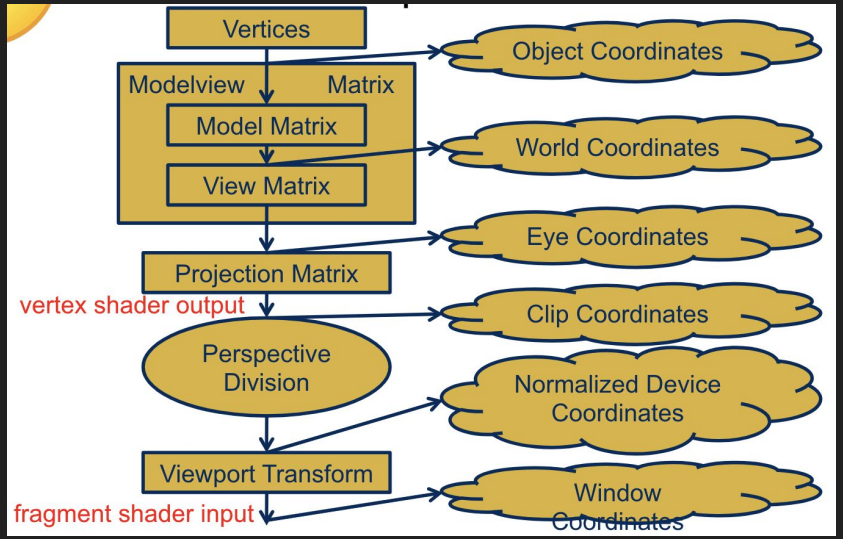
Four Seasons Island Screensaver - CG

By Thapelo Masasa, Jerry Migwi, Ray Bae, Jesus Robles

Project Overview

- ❏ Create a Screensaver 3D graphic
- ❏ Encapsulate the course concepts learned in class through the Computer Graphics Pipeline
  - ❏ Create Elements
  - ❏ Lighting
  - ❏ Texture
- ❏ Explore working in Three.js



Creation framework of the project

- Project inspired by: https://medium.com/@curry_is_nice/summer-3d-island-3d0be731ce1b

Creation framework of the project

- Changes made from our team:
  - Multiple and changing seasons with time variable (not only summer)
  - Created various townscapes with buildings and borders
  - Implementation of weather (e.g., snow)
  - Animal change to seagulls that fly across the screen unlike the static version in the inspiration post.
  - Shapes and sizes of the island to mimic a 90s video game aesthetic.
  - Addition of music to match the changing seasons.

Components

- Body of water and 3D islands instantiated in JavaScript with Three.js
- Tone mapping, water class, sky class, shading, shaderMaterial within Three.js
- Various functions to tackle time processes to match the season/weather
  - E.g., rainbow with clear skies, snow in winter, respective lighting on the objects

```
function addRainbow() {
  if (hasRainbow) {
    return;
  }

  hasRainbow = true;

  const material = new THREE.ShaderMaterial({
    side: THREE.DoubleSide,
    transparent: true,
    uniforms: {},
    vertexShader: document.getElementById('vertex_shader').textContent,
    fragmentShader: document.getElementById('fragment_shader').textContent,
  });
  const geometry = new THREE.TorusGeometry(200, 10, 50, 100);
  rainbow = new THREE.Mesh(geometry, material);
  rainbow.opacity = 0.1;
  rainbow.position.set(0, -50, -400);
  scene.add(rainbow);
}
```

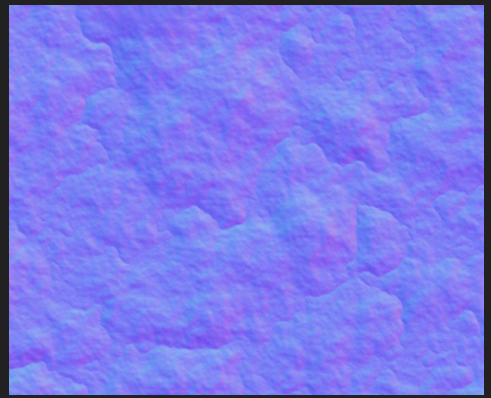
```
function fadeOutSnowIsland() {
  if (isSnowing) {
    return;
  }

  const currentTime = new Date();
  const timeElapsed = currentTime - timeSinceLastFade;
  if (timeElapsed < fadeTimeThreshMs) {
    return;
  }
  timeSinceLastFade = new Date();

  island_meshes &&
    island_meshes.forEach((island) => {
      if (island.name == 'green_island') {
        if (green_island_opacity < 1) {
          green_island_opacity += 0.01;
        }
        island.traverse((child) => {
          if (child.isMesh) {
            child.material.opacity = green_island_opacity;
          }
        });
      } else {
        if (snow_island_opacity > 0) {
          snow_island_opacity -= 0.01;
        }
        island.traverse((child) => {
          if (child.isMesh) {
            child.material.opacity = snow_island_opacity;
          }
        });
      }
    });
}
```

Components - Sun

- Created by making use of a Three.js skybox.
- We dynamically updated the sun position and sun direction to mimic different times of the day e.g sunrise and sunset.
- We added a Three.js point light to the sun to give the effect of a lens flare on the sun in order to make it more realistic.

Components - Water

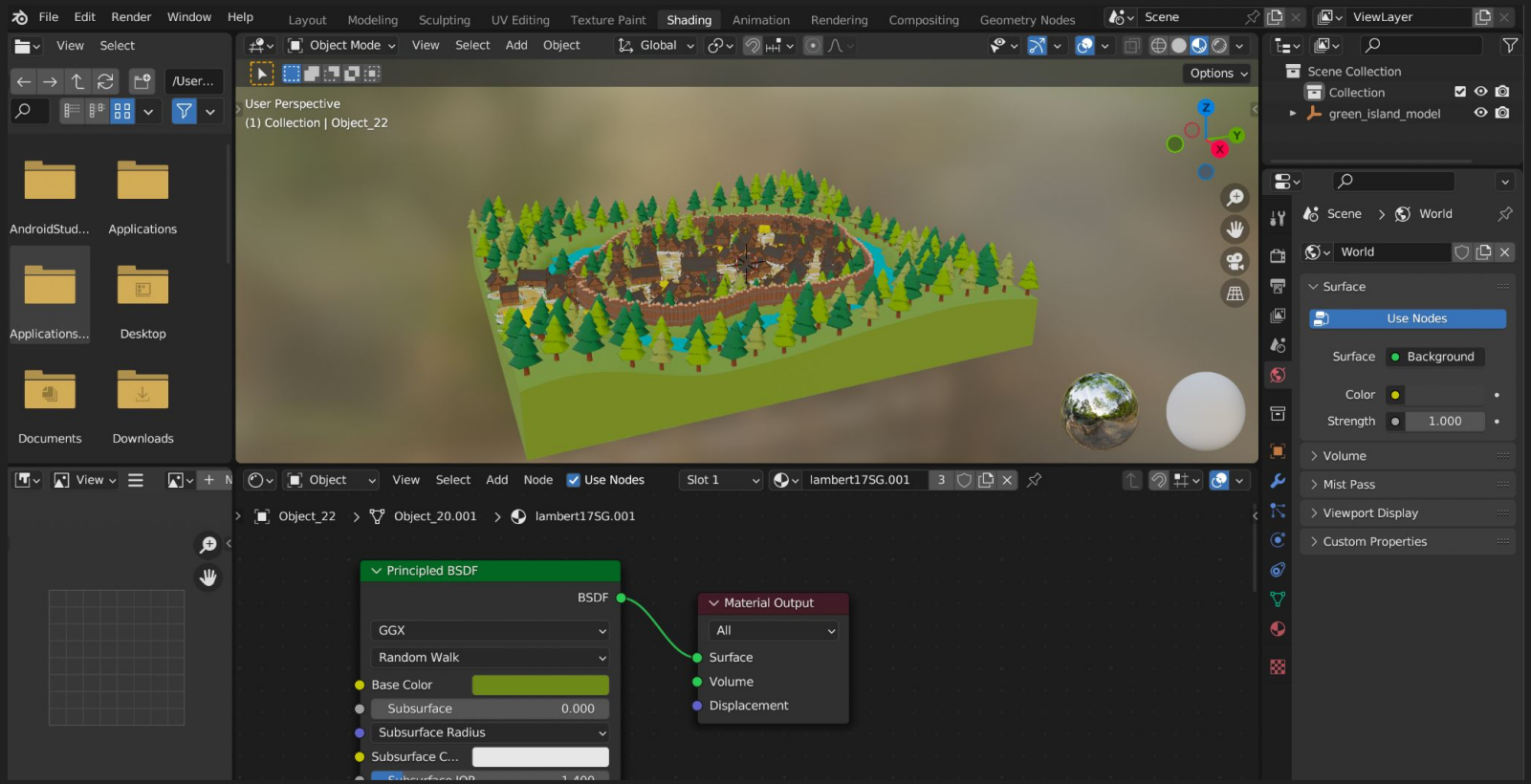
- Created by making use of a Three.js plane geometry and adding a water texture to it.
- The water component is dynamically rotated on the x-axis to give the illusion of having waves.
- The water plane geometry also allows for reflections on its surface which makes our scene appear more realistic.



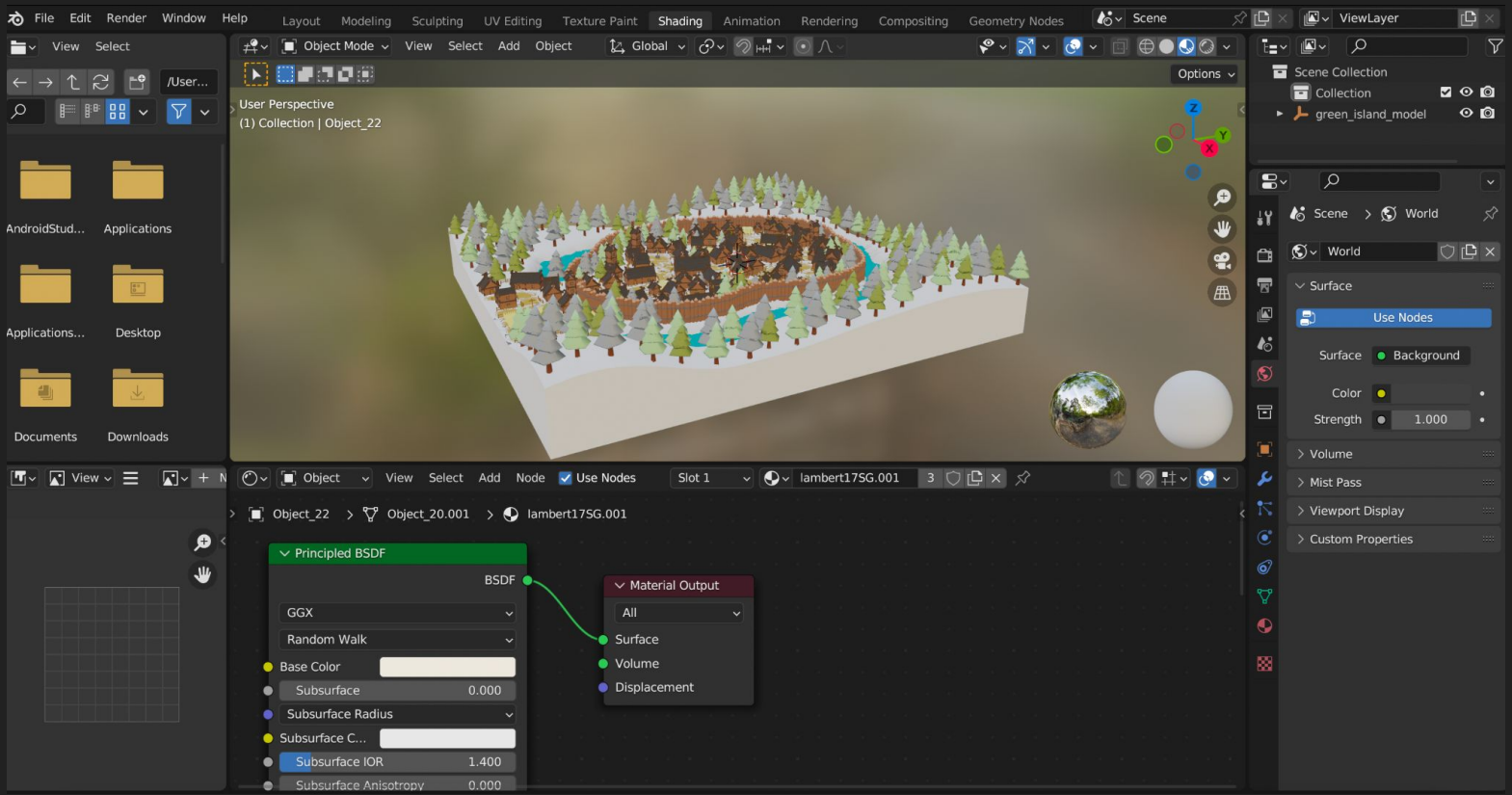
Components - Island

- 3D model downloaded from sketchfab.
- We used Blender to modify the 3D model to match our desired use case by:
    a. Removing unnecessary components from the original model e.g lights and cameras.
    b. Adding a different model shader to mimic an island during winter time.
- In Three.js, we constantly updated the island's opacity to be able to switch from the green island during summer time to the white island during winter time.

Summer Island Model



Winter Island Model



Components - Rainbow

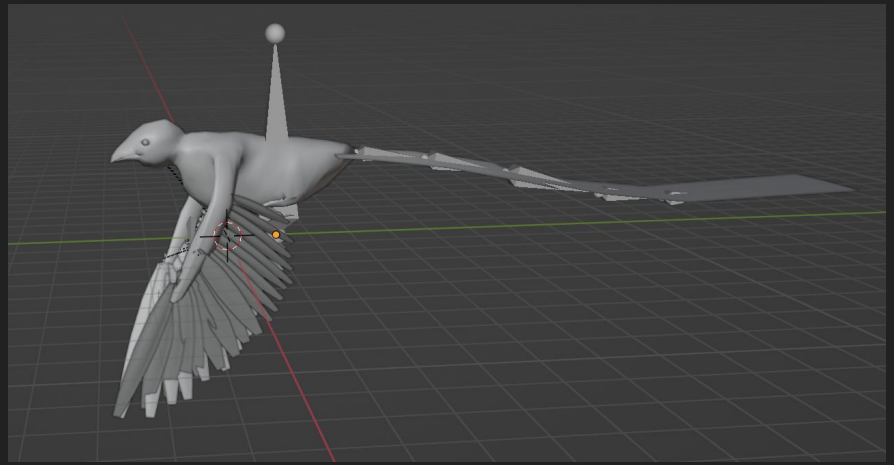
- Created by making use of a Three.js mesh object.
- A Three.js torus geometry was added to the rainbow mesh object to mimic the iconic rainbow arc.
- A Three.js shader material was added to the rainbow mesh in addition to a vertex-shader and fragment-shader that provided the rainbow mesh with its color.

Components - Snow

- Created by making use of a Three.js buffergeometry and adding a custom texture to it.
- Each instantiated snow particle has a random and unique position and velocity to make the snow appear dynamic and realistic.

Components - Bird

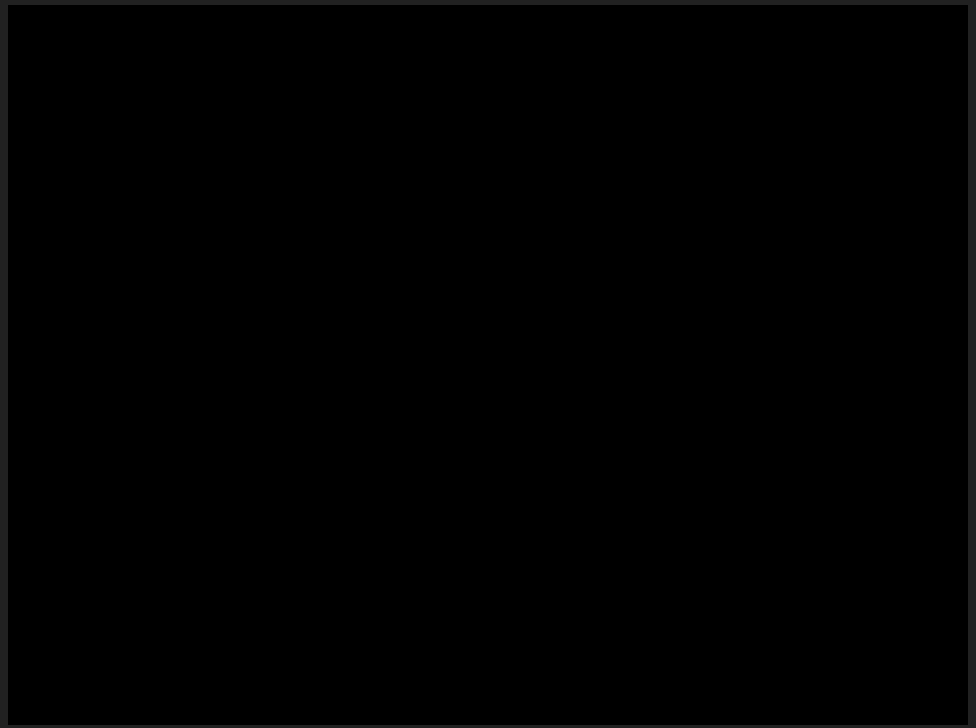
- The bird used is an animated 3D model downloaded from sketchfab.
- Each bird's x,y, and z positions are dynamically updated to mimic a bird flying across the island in real time.



Components - Music

- Added different songs that change dynamically depending on the current season.
- Having background music ensures our screensaver remains lively and engaging.
- Used music from YouTube Audio Library

Demo



Improvements & Challenges

- Challenges
  - The learning curve for Three.js was a challenge → new to all of us
    - Core Knowledge helped with the transfer
- Future Improvements
  - Add scenes and possibly interchange within them like an option
  - Increase intractability
    - Changing music, interacting with the elements further

- Q & A -