
Wave Simulator

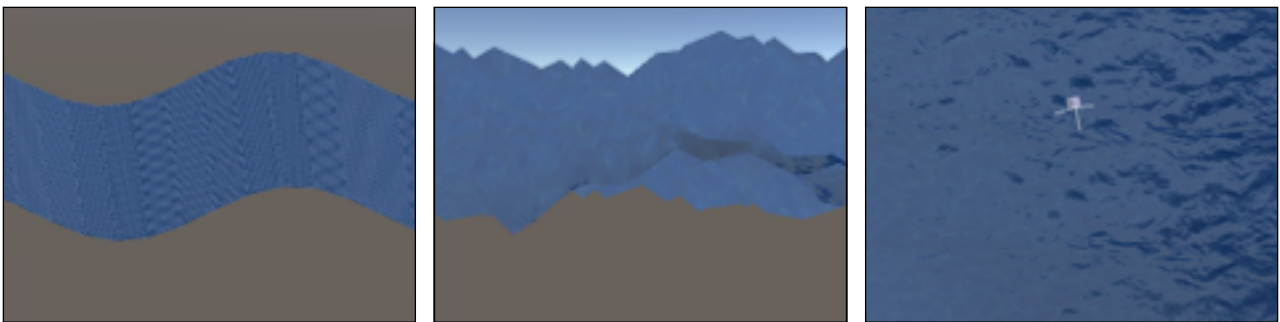
Simulation of Deep Ocean

Jack Prior - 19 April 2016



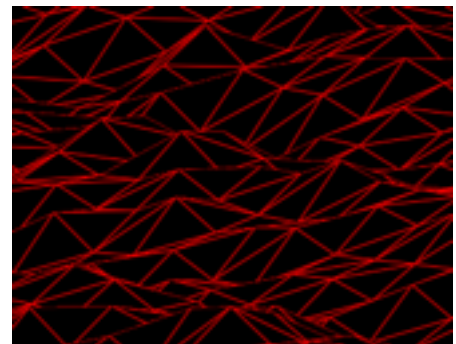
Wave Generation

- My simulation displays deep ocean waves by looping through an array of all of the mesh's vertices and applying a sine wave function.
- It is then passed through a Perlin Noise function to add a degree of randomness.
- A Bias is accounted for to allow for higher tide.
- The simulation can be run continuous or be frozen at any time period throughout a cycle.
- The wave direction can be altered through the UI and the steepness is incorporated by applying the `Mathf.Pow()` function to `Mathf.Sin()`.



Textures

- The texture for the ocean can be swapped by pressing space bar and that changes it from a highly realistic texture with a bump map and fresnel to a black and red wireframe.



Config

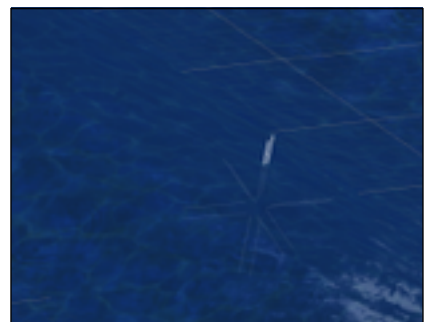
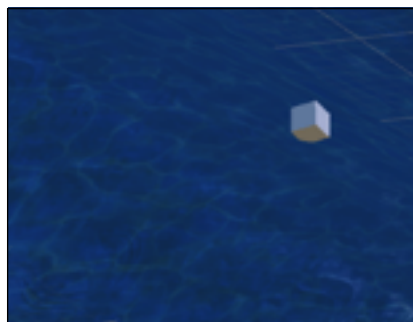
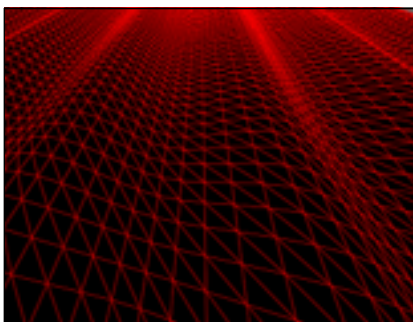
- All of the sine wave parameters are loaded from `config.xml`
- When any changes are made via the UI it is saved back to the config file.
- This is handled by a class called `SineWaveModel` which attempts to encapsulate everything required by the `waveController` class.
- The UI was customised with a guiskin and have a custom font.



Endless Terrain and Flycam

- The 'F' key turns on flycam, allowing the user to traverse the demo easily.
- The ocean segments are spawned dynamically.
- The Infinite waves algorithm looks at the players position vector and appropriately spawns more ocean segments so that which ever way the player travels they will always be on top of an ocean.

```
int xMove = (int)(player.transform.position.x - startPos.x);  
int zMove = (int)(player.transform.position.z - startPos.z);  
  
if (Mathf.Abs (xMove) >= planeSize || Mathf.Abs (zMove) >= planeSize)
```



Boat Physics

The BoatPhysics script looks at the vertices of a given object that are submerged underwater and applies a buoyancy force equal to:

```
Vector3 F = density * Physics.gravity.y * distance_to_surface * area * crossProduct;
```

Wave drift is also considered for horizontal movement:

```
Vector3 F = 0.5f * density * Physics.gravity.y * area * area * normal;
```

- Gravity intensity, the mass of the boat and water density are all configurable dynamic parameters.

References

www.habrador.com
scrawkblog.com
forum.unity3d.com/
VacuumShaders