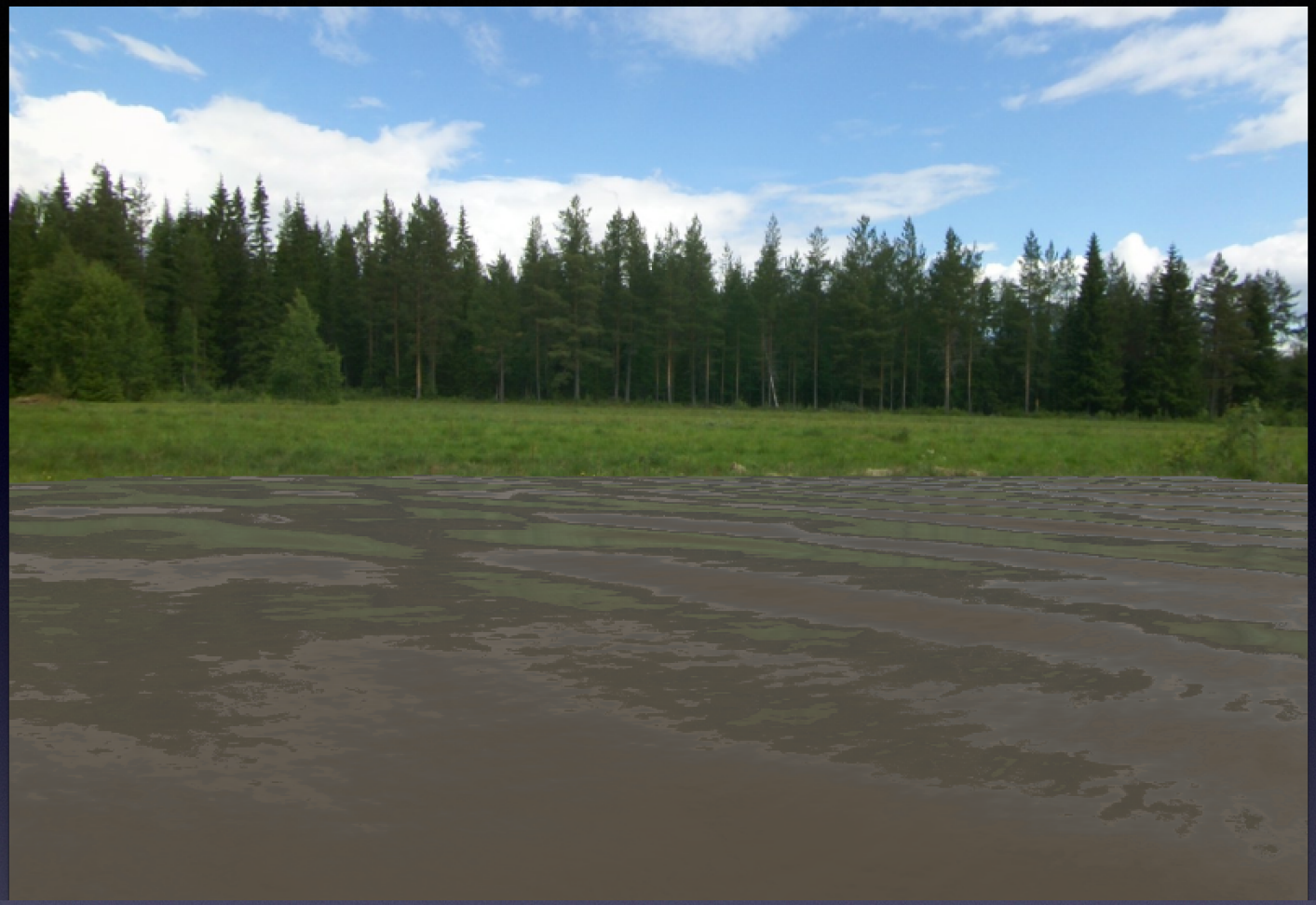


Realtime Water Simulation

Benjamin Harry
CS148 Final Project

Project Goal

Create a realtime water simulation that exhibits characteristics of a real body of water.

- Generate believable waves in realtime
- Light reflection and transmission
- Caustics on bottom surface

Relevance to Course

Using OpenGL

Lighting Effects:

- Reflection - Fresnel Effect
- Refraction - Snell's Law

Implementing Waves

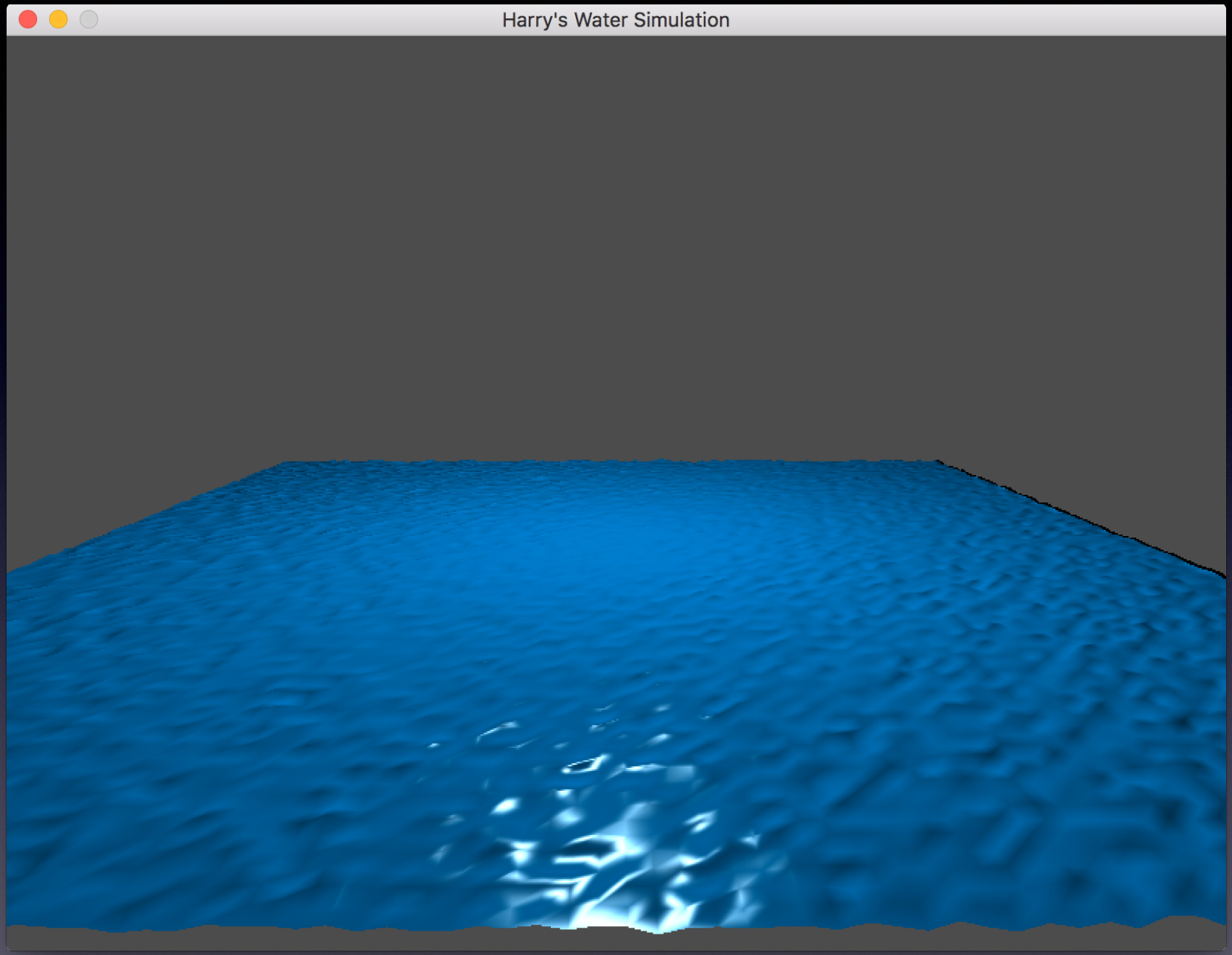
Method for wave animation from Matthias Müller-Fischer.

Initialized vertex heights for waves using sum of sines function.

Initialize horizontal velocities to zero.

For each time step update the vertex positions based upon forces acting on waves.

```
// Implementing the waves using height field.
for all i, j {
  f = c^2 * (u[i+1,j] + u[i-1,j] + u[i,j+1] +
      u[i,j-1] - 4u[i,j]) / h^2;
  v[i, j] = v[i, j] + f * Δt;
  u_new[i, j] = u[i, j] + v[i] * Δt;
}
for all i, j: u[i, j] = u_new[i, j];
```


Harry's Water Simulation

Height Field Conclusion

Very easy to implement.

Generated waves had decent animation, although could see triangles when looked closely.

Data updated in CPU, which requires a lot of processing and copying data in for loops.

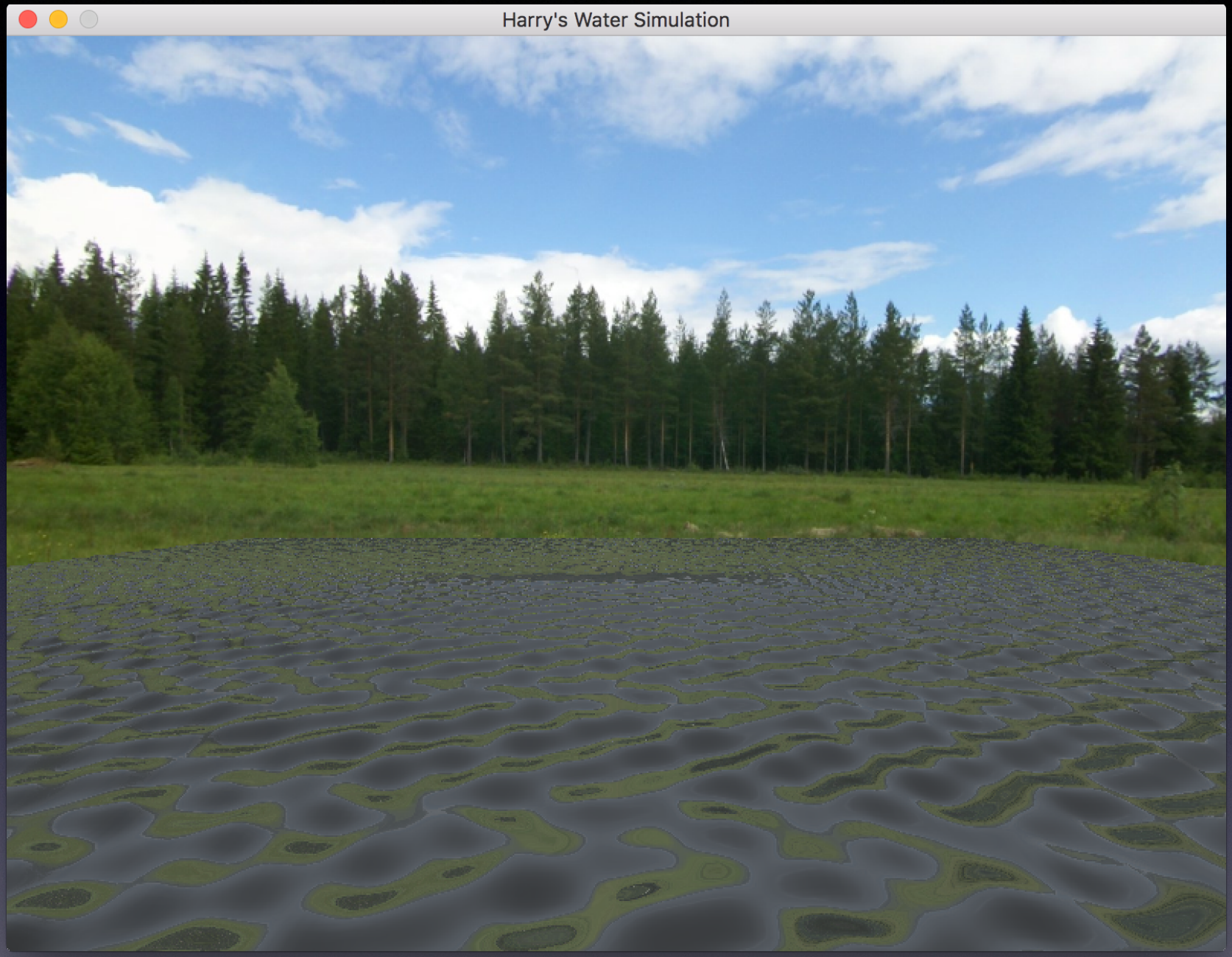
Degraded frame rate.

Summing Sine Waves

Sum sine waves with varying amplitudes, wavelengths, and directions to get height at each vertex.

Need to also have the derivative to get the normal at each vertex.

Perform the calculations in the GPU.


Harry's Water Simulation

Summing Sine Waves

Waves have "blobby" appearance. Sometimes sine wave shape is obvious (using random directions).

Improved by adding high frequency normal map in the fragment shader.

Lighting Effects

Reflections using OpenGL cube map feature.

Improved the look of reflections using Fresnel effect.

- Viewing angle affects how much light is reflected and transmitted.

Caustics added using Snell's law and wave normals.

Challenges

Figuring out which method to use to generate waves.

- Overwhelming amount of information on the internet.
- Implement wave calculations on CPU or GPU?

Overcoming blobby appearance of waves.

- Adding animated high frequency normal map.

References

Bouny, Jeremy. "Realistic Water Shader for Three.js." GitHub. github.com/jbouny/ocean

de Greve, Bram. "Reflections and Refractions in Ray Tracing." Nov 13, 2006. graphics.stanford.edu/courses/cs148-10-summer/docs/2006--degreve--reflection_refraction.pdf

de Vries, Joey. Learn OpenGL. www.learnopengl.com

Finch, Mark. "Effective Water Simulation from Physical Models." GPU Gems. 2004. developer.nvidia.com/gpugems/GPUGems/gpugems_ch01.html

Guardado, Juan. "Rendering Water Caustics." GPU Gems. 2004. developer.nvidia.com/gpugems/GPUGems/gpugems_ch02.html

Hollasch, Steve. Steve's Web Pages. Nov 4, 2007. steve.hollasch.net/cgindex/render/refraction.txt

Müller-Fischer, Matthias. "Fast Water Simulation for Games Using Height Fields." GDC 2008. matthias-mueller-fischer.ch/talks/GDC2008.pdf

Tessendorf, Jerry. "Simulating Ocean Water." 1999 - 2001. graphics.ucsd.edu/courses/rendering/2005/jdewall/tessendorf.pdf