<u>Cellular Automata Based Tsunami Simulation</u>

Final Report
Team 1
New Mexico Supercomputing Challenge
April 4th, 2012

Max Benjamin Bond	Erik William Nelson	Arlo James Barnes	William Christopher Fong
Monte del Sol Charter School	The Academy for Technology and the Classics	The MASTERS Program Charter School	The Academy for Technology and the Classics

Mentor: Nick Bennett

Executive Summary

Tsunamis are powerful disasters that can wreak havoc on cities. In order to protect critical infrastructure, such as food storage or power plants, our project seeks to create a tsunami simulator that will locate the areas that will be most grievously damaged. These structures can then be constructed away from these areas, in the safest possible location. This will be accomplished using a Cellular Automata model.

Problem Statement

The destructive power of tsunamis has been demonstrated in a series of disasters in the past few years. Recently, we have had catastrophic and lethal incidents in Chile and Haiti, and also in Japan, the inspiration for this project. On March 11th of 2011, a tsunami hit the coast at Sendai. The water caused not only extensive flooding and structural damage, but hit the Fukushima Daiichi nuclear power plant directly. The compromised nuclear plant leaked radiation, which irradiated and displaced citizens and damaged the environment; it will take years of time and incalculable expense to clean up. In future urban planning for cities affected by tsunamis, structures susceptible to such risks will need to be placed where the waves will have as little effect as possible, such as areas shielded by hilly terrain, or far enough on shore that they waves do not reach them.

Our project will seek to create a simulator that will approximate the tsunami event, in order to locate such areas. Our model will approximate the inundation, as well as record various forms of damage to structures, such as damage from exposure to water, and damage caused by the wave's kinetic energy. We plan to use GIS data to simulate the Fukushima Daiichi tsunami, as a case study.

Approach

In order to accomplish this, we must simulate a tsunami and the damage it inflicts, so that areas where less harm was incurred will be revealed. There are several things that are necessary for this task. We must be able to model water, in a manner which can be shown to mimic its behavior, and we must be able to detect and record damage to structures, as well as the tsunami's inundation.

At the outset, we decided that while accuracy is important to a simulation such as this, precision is not. The detailed aspects of waves, such as shoaling and cresting, and the finer parts of a tsunami, like the transport of debris, are extraneous to our goal, which consists of finding the gross impact of a tsunami. We therefore deemed it appropriate to attempt a very minimalistic simulation.

In this spirit, we abandoned our initial study of an existing method for modelling water, the Shallow Water Equations[1], and instead used it as a source of inspiration for developing our own simplified Cellular Automata implementation.

Cellular Automata models use discrete cells following simple rules in relation to the states of their direct neighbors, to mimic complex behaviour.

Imagining a body of water as a matrix of discrete cells, we observed there to be two basic phenomena at work. Firstly, gravity is constantly pulling this body of water down. But as water is an incompressible fluid, rather than making the water more dense in areas containing more mass, the force of gravity causes the water to move from high areas to low areas. The effect is that the body of water adjusts itself to remain at the same elevation throughout.

The second observed phenomena is that, because of the Conservation of Energy, momentum propagates itself through matrix, moving water from one cell to another and creating waves.

Using these, we designed a model that we termed the "Two-Vector Method."

The world of our model is a two-dimensional array, with each element being a cell. These cells hold various values, including a representation of their height in the third dimension.

The first vector represents the momentum that travels through bodies of water, and is therefore called a Momentum Vector. It holds the values for momentum on each axis. Each time the simulation runs, all of the vectors entering a cell are summed together, and this vector as well as the water it displaces are distributed to neighboring cells in a certain proportion based on the direction the vector is travelling.

The next vector, the Transfer Vector, approximates the way by which bodies of water tend towards a uniform height. The Transfer Vector facilitates motion from high areas to low areas. The Transfer Vector holds two values; the time it was created, and the cell it is transferring to. On each iteration of the simulation, it creates momentum towards this cell, and the amount of momentum is dependent on when it was created and the acceleration of gravity on Earth.

Momentum vectors select the neighbors they will propagate to and the proportion in which to do so in the following manner;

1. Obtain the *destination*, which is the vector summed with the coordinates of the current cell
2. Of all the current cell's neighbors, those which are closer to the destination than to the current cell will be propagated to
3. To each of these cells, give a portion of your momentum, the proportion being determined via to following equation:

$$m_a = (1 - (\frac{d_{neighbor}}{d_{all}})) \bullet v_a$$

Where m is momentum, a is axis, $d_{neighbor}$ is the distance to the neighbor we are distributing to, d_{all} is the sum of all distances of the neighbors being distributed to, and v is the vector we are distributing.

When a momentum vector is distributed, it carries with it some amount of water. The amount is determined by the same proportion as momentum is distributed with, and by the resistance of the weight of the water to be carried. As the weight of the water can be thought of as a vector acting directly perpendicular to the vector (on the z axis), it does not directly counteract the momentum vector. Rather, it has the effect of stretching it diagonally through the water, so that the deeper the water, the less direct the path and the less mass transmitted. More shallow water will yield a greater percentage of water transmitted.

The exact value of the water transmitted, *w*, is calculated by subtracting the distance from the cell distributing the vector to the destination, from the distance of the destination to the bottom of the cell. The equation is:

$$w = \sqrt{H^2 + (P \bullet H)^2} - (P \bullet H)$$

With *H* being the height of the cell, and *P* being an abbreviation of $1 - d_{neighbor} / d_{all}$.

Every iteration of the simulation causes Transfer Vectors to create momentum in the direction of a lower cell. As the water this represents is falling, it is also accelerating. The vector it produces, then, is given by:

$$t_a = (G \bullet [\frac{s}{i} \bullet T]) \bullet c_a$$

Where *t* is the vector produced, *G* is the acceleration of gravity, *s / i* is the seconds per iteration, *T* is the number of iterations since the start of the transfer vector, and *c* is the cell it originates from.

There are special 'structure' cells, and they represent man made buildings. When they are hit, they record the kinect impact they receive. If they are partly or fully submerged in water, the depth to which they are submerged and for how long is also recorded.

Conclusion

We are finishing up refinements to the way our simulation models water (the momentum vector aspect of our project is complete, and what remains is the code to handle transfer vectors), and starting on an SVG/SMIL visualisation; our next step after that is to compare it to the data[6] we collected from a wave tank we made, and to run the model on our case study to demonstrate its accuracy charting the path of tsunamis.

We plan to refine our model beyond its purpose in the Challenge.

Acknowledgements

We'd like to give a very special thanks to
Nick Bennett, who has been our (very patient) mentor for four consecutive years,
The Challenge consult team:
Drew Einhorn
Celia Einhorn
David Kratzer
That Guy Who Does All The Tech Stuff
Nico Marrero, a New Mexico tech student who reviewed our interim
Jeff Fenchel,
Cali Olsen, a local student who critiqued a presentation we made of our product
Robert Bond, who gave us good advice

Works Cited

1. The Shallow Water class at the kickoff - http://challenge.nm.org/kickoff/classes/#shallow
2. A paper on Shallow water by the MathWorks founder and employee Cleve Moler - http://www.mathworks.com/moler/exm/chapters/water.pdf
3. Navier-Stokes equations - http://mathworld.wolfram.com/Navier-StokesEquations.html
4. Navier-Stokes equations - http://icon.enes.org/current/SW-a/index.html
5. A New Kind of Science, by Stephen Wolfram, 2002, ISBN 1-57955-008-8
6. Total Volume: 816.75 cubic inches
   Displacement: 389.8125 cubic inches (at 20 degree incline)
   Max Wave Height: Approx. 7 inches
   Avg. Wave Interval: .8 seconds
   Max-Min Wave Time: Approx. 26 seconds
   Minimum wave height is around 1/8 inch during this interval.