Rotating Wind Turbine Project

by Ata Gowani

Problem Summary

The problem that I am working on is that of building a wind turbine simulation that will respond to wind particles that hit the wind turbine. This simulation will be able to demonstrate how the interaction of wind particles with the turbine causes rotations for the turbine and the energy that would be generated due to the rotation.

Description of Work

My project presents a wind turbine simulation programmed using the JavaScript graphics library Three.js. The simulation uses a raycaster-based collision detection algorithm and physical concepts of wind turbine power generation to set the rotation of the turbine based on the wind particles. When wind particles collide with the wind turbine, it leads to rotation in the turbine and energy generation based on the properties of the wind particles such as velocity, density, and size/coverage. The simulation allows the user to change the velocity and density of the wind particles. The simulation also considers friction in the wind turbine components and adjusts the turbine speed to account for that.

Major Challenges Encountered

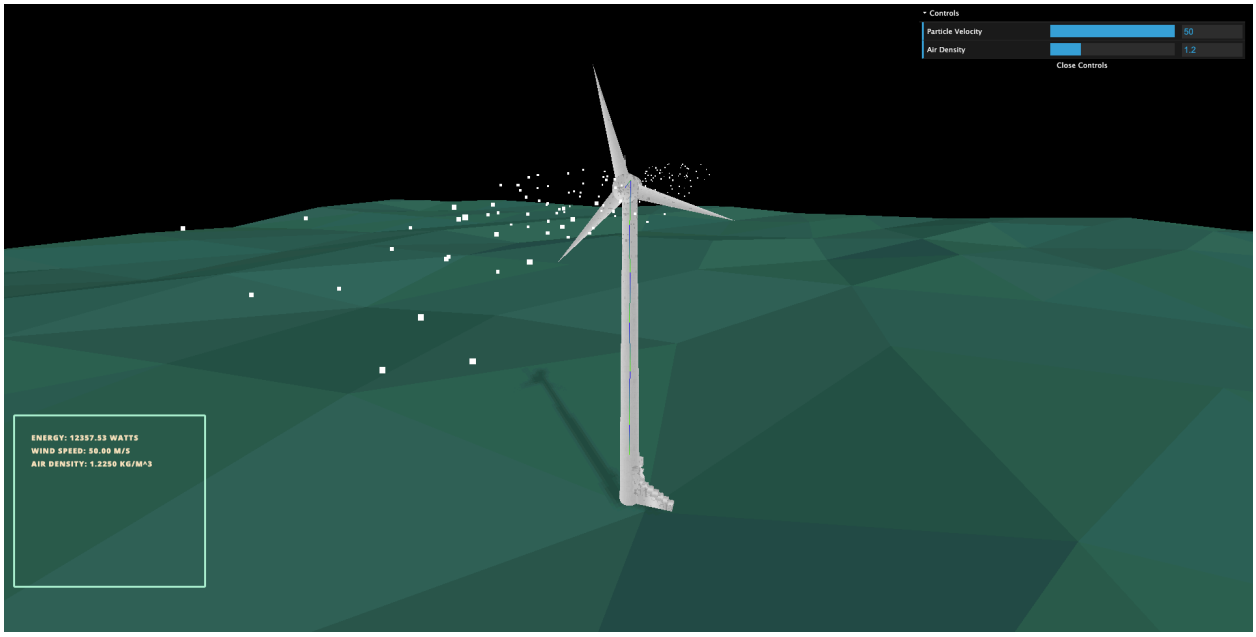
Some of the major challenges that I encountered were:

- There were major issues with my collision detection algorithm not working as intended when trying to detect collisions with a particle system. In order for my program to work as intended my algorithm would've needed to detect each particle separately as it collided with the turbine, but due to the nature of particle system being treated as one big mass, a single object could not be separated for detection. I fixed this by creating separate objects that acted as a particle of system yet were independent of each other. This allowed me to perform a single particle detection at a time.
- The creation of more than 500 wind particles led to very slow animations due to the massive overhaul from particle creation. I knew I needed lots of wind particles in order to have a good wind turbine simulation, but I realized that at any given time only a few particles will be colliding with the wind turbine. So, I decided to solve this issue by recycling particles once they had collided. This means is that after a particle has passed the wind turbine is it set back in position so that it can pass again once more this process is repeated allowing me to achieve the desired effect with just a 100 particles instead of 500.

- An algorithmic deficiency that I ran into was that the collision system detected multiple hits for one wind particle collision as it kept staying in contact with the wind turbine while passing through it. This led to an exaggerated and inaccurate turbine rotation and energy count. After spending some time trying to modify the collision detection algorithm, I discovered that Three.js provides metadata that can be stored alongside objects. I instead decided to store metadata containing details of whether a certain particle had already collided, and therefore been accounted for in the rotation speed, to avoid duplication of a particle's effect on the rotation. This metadata is reset when the particle is recycled as described before.

Results

I was able to finish a simulation that contains wind particles flowing perpendicular to a wind turbine that is initially at rest. There is a raycaster-based collision detection algorithm in place that checks to see if any of the wind particles have collided with the wind turbine. At collision, properties about the wind particle such as velocity, density, and area covered are then used to determine the rotational impact of the wind particle on the turbine and also leads to updates in the energy stats box. The simulation also allows for the user to control the wind speed and density to see the effects of different wind conditions on the rotation and energy generated. A screenshot of the project is attached below and the completed project can also be access online at atagowani.com/wind-turbine-simulation/.



Analysis of Work

I set out create a program that could allow for a better understanding of how a wind turbine would function in different wind conditions through wind particle and turbine collision detection. I think that I was able to accomplish what I set out to accomplish in the most basic sense, however, there are major improvement that could help deliver better results for different and more intricate wind turbine design. Currently, my simulation assumes the most ideal wind turbine with no loss of energy. In a more real example wind turbines will not be generated as such. My program also does not allow for there to be different turbine types nor does it consider aerodynamics even if different turbine types were put in place of the current turbine. These concepts were not implemented in this project because they are beyond the scope of this project due to their complexities.

This project does not introduce any new results or findings.