CPSC 478 Spring 2013

Final Project Report

Kerry Clavadetscher and Anthony Tordillos

For our final project we present a first person shooter type game titled “Fireflies.” Below we give a summary of the game and instructions on how to load and play it. We also provide an explanation of its noteworthy components and accomplishments.

**Game Summary**

In our game you become a small child whose toys have hidden themselves for you to find in your backyard near your house in the middle of the night. Your goal is to retrieve all them. You do this by moving around your yard – a square plot of land marked by a forest border, and walking up to the toys. The caveat is that you must do this before running out of light and only one toy likes to show itself at a time. It is a very dark night and the only sources of light are the roaming fireflies that populate the scene. You are able to collect fireflies in the same way you collect toys, and when you do so you capture the firefly light in the jar that you hold, which helps light your way. If you run out of light (these fireflies have very short lifespans!) then you lose. You win when you retrieve all the toys.

**Instructions**

To load our game please go to the following link: . We have hosted our game on Google Drive. To operate the game use the w-key to move forward and s-key to move back. Click and drag to rotate the scene.

**Noteworthy Features**

Models

We modeled our scene in Blender. Most of the elements of our scene we modeled ourselves including the forest border, ground, fireflies, and materials of all the objects. The meshes for the house and the toys we took from publically available models. Our models were much more complex than we had anticipated. For example the forest border is comprised of a particle system where each tree is a particle. There are 300 trees in total and hundreds of vertices per tree. In addition many of the toys have over 1000 triangle faces. Though our main focus was the lighting we spent a good deal of time working with these complex meshes – generating them, altering them, parsing them into readable formats, figuring out how to load them efficiently into the game, rendering them quickly, etc.

Lighting

The signature of our game is the advanced real time lighting. Our moving, attenuated firefly point lights provided a real challenge, but we mastered this by using the object-based shaders in WebGL. We use Phong shading with a constant number of lights whose positions and intensities are updated every rendering cycle. Please take note of the specular reflections on several of the house fixtures, toys, and even the trees (it’s the sap!). The only shortcoming is the lack of shadows, but the real time rendering of the thousands of triangles in our meshes using the Phong lighting model was a real accomplishment.

Camera Movement

Originally we used only keyboard input to move (i.e. wasd for translation and arrow keys for rotation) but we wanted to show the ability to work with mouse events. We changed from using the arrow keys to using the mouse click and drag. Although the arrow keys are more intuitive we kept the mouse control in order to show our game was capable of using such input.