Alyssa Chan (awc894)

Ethan Chandra (ec42227)

Graphics Final Project Report

Project Overview

This project implements a mass-spring physical model for cloth simulation. We used the same environment that we created back in the virtual mannequin project so that the user can interact with the model, zoom in/out with the camera, and turn to different views. We wanted to specifically highlight the cloth movement, so we decided dropping the cloth on a simple static sphere would be a good visual representation of the simulation.

First, we wanted to create a simulation that was able to showcase cloths with various characteristics. For this, we have preset cloth types: Cotton Basic, Silk Drape, Leather Stiff, and Rubber Stretch. Each perform the same function of dropping cloth on the sphere but with various visual outcomes. Taking this a step further, we decided to create a simulation option with custom parameters so that we could test different cloth characteristics in real time. These parameters include structural stiffness, shear stiffness, bend stiffness, damping, mass, and stretch factor. All of the characteristics were chosen based on real properties of cloth that we see in the actual world.

Next, we wanted to show how cloth responds differently to various environmental factors. We already had gravity implemented for the fundamental “dropping” of the cloth, but we also added wind and pin configurations that affect the cloth’s movement as well as the ability to click and drag on the cloth itself to change its movement. The wind strength and directions can be changed as well as toggled entirely on or off. There are settings for adding pins to the center and/or each of the four corners. Additionally, we implemented a feature that allows the user to manually “take-off” or “add” pins to any of the particles within the cloth. All of these features significantly affect the cloth's movement by adding and taking away degrees of freedom in some or all regions.

To explore how mesh resolution affects cloth flexibility, we created a short animation showing the cloth dropping with face counts ranging from 1 to 4096—higher counts produce more flexible movement. Finally, we also added a toggle (press 'V') to switch between different visualizations of the mesh and particle structures used in the simulation.

Implementation

For the cloth model itself, we wanted to simulate cloth similar to what we see in the actual world. To do this, we started by creating a model of multiple particles arranged in a 2D grid structure. Each particle acts as a point in the cloth with its own energy, position, and collision detection, storing values for mass, velocity, and acceleration. The particles are evenly spaced within the cloth, and thus, cloths with more faces/particles have particles that are positioned closer together. From this, we then created various mesh structures that connect pairs of particles with simulated "springs" using three distinct types: structural springs (horizontal/vertical connections) for basic shape, shear springs (diagonal connections) for preventing unrealistic deformation, and bend springs (skip-one connections) for fabric stiffness. This is to ensure that the cloth maintains its mass and the particles stay connected with each other while reacting to the environment individually. The springs act as springs would in the real world, while operating on a time-stepping loop that integrates forces and updates positions via Verlet integration.

The overall cloth simulation uses fundamental physics equations to calculate the forces of wind, kinetic energy, potential energy, gravity, and collisions with objects/floor. Each particle stores its own values for position and velocity, and the equations are applied to each particle individually using Verlet Integration with position(t+dt) = 2 * position(t) - position(t-dt) + acceleration * dt^2 and velocity = (position(t+dt) - position(t)) / dt. Spring forces follow Hooke's Law with F = -k * (current_length - rest_length) + damping * velocity_difference, while wind forces are calculated per quad using F_wind = direction · normal * $strength^2$ * area. For constraint solving, we employ Position-Based Dynamics with Δp = (current_length - rest_length) / ($mass\_A^{-1}$ + $mass\_B^{-1}$), which satisfies the spring constraint while maintaining mass conservation.

One thing to note is that our particle and mesh system places particles at the corners of each face, which causes an issue in the 1-face cloth animation—it falls through the sphere. This happens because collisions are calculated at the particles, and with only four corner points, none make contact with the sphere. As the mesh resolution increases, more particles are added, allowing proper collision and preventing the cloth from falling through. However, the simulation becomes computationally expensive with high-resolution cloth meshes (>4096 faces).

We implemented a control panel on the right side of the display, with a dropdown at the top to select from various test options—most of which include a short animation and a reset button to replay the simulation. The most complex interface is under the “Custom Parameters” test, which features toggles and sliders for cloth and environment settings. Here, users can also interact with the cloth directly by adding pins or dragging it, with all changes reflected in real time.