Eric Mouledoux

I.0: Requirements Documentation

I.1: Description of the Problem

**Name:** Cloth Physics

**Problem Statement:** Implement spring physic to simulate realistic cloth simulation.

**Problem Specification:** The application should simulate cloth using springs and physical forces such as gravity and wind.

I.2: Input Information

Description: Runtime interaction with the cloth

Type: N/A

Range of Acceptable Values: Mouse position and mouse buttons

I.3: Output Information

The program will simulate cloth based on input given before, and during runtime of the application.

I.4: User Interface Information

N/A

II.0: Design Documentation

II.1: System Architecture Description

The program is split into 3 different classes and 2 utilities. The main class of the simulation is Cloth.cs. This will create a grid of nodes, link the nodes with springs, and store apply post generation modifiers to specific variables. Node.cs stores unique values for each node in the system that will be used to move the node in space. Spring.cs links 2 game objects together and applies opposing forces to said game objects. The 2 utilities are CameraMove.cs and UI.cs. UI. cs is only used for exiting the program. CameraMove.cs is used to move the camera during runtime.

II.2: Information about the functions

**File: Node.cs**

Class: Node

Data Members

Name: acl

Description: the current acceleration of this node in space

Name: vel

Description: the current velocity of this node in space

Name: frc

Description: the current force being applied to this node

Name: isLocked

Description: ture or false if this node’s position is locked in spcae

Function: Awake

Parameters: none

Description: sets all initial values associated with this node to zero.

Return: none

III.0: Implementation Documentation

III.1 Program Code

IV.0: Verification and Validation Documentation

IV.1: Test Plan

Test plan includes the following cnf expressions included in the Expressions.txt file

**Tests for functionality**

**Test case:** Simple functionality

**Results:** All pass

(!a \* c) + (!a\*c\*!e)+(!b\*c\*d\*!e)+(a\*!b\*c)+(!e\*f) result:

(a \* b) + (c \* d) + (e \* f) + (g \* h)

(!a \* b) + (c \* d) + (e \* !f) + (g \* h)

(r \* b) + (c \* d) + (c \* a \* !b \* b \* c \* z)

(a \* b \* c) + (c \* d \* e)

(f \* g \* k) + (n \* m)

(q \* z \* l \* v) + (z \* k \* r) + (m \* n \* f) + (b \* r \* t)

(!a \* b) + (c \* d) + (e \* !f) + (g \* h) + (i \* j) + (k \* l) + (m \* n) + (o \* p) + (q \* r) + (s \* t) + (u \* v) + (w \* x) + (y \* z)

(!a \* b \* !c) + (c \* d \* !e \* !g \* !m) + (e \* f) + (g \* h) + (a \* !g \* !f) + (e \* f \* g \* k \* r) + (!e \* a \* b) (a \* b) + (c \* d)

**Tests with Expected results**

**Test case: All values must evaluate to true**

(a) +(b) +(c) +(d) +(e) +(f) +(g) +(h) +(i) +(j) +(k) +(l) +(m)+ (n)+ (o)+(p) +(q) +(r) +(s)+(t)+(u)+(v)+(w)+(x)+(y)+(z)

**Expected: 11111111111111111111111111**

**Result: 11111111111111111111111111**

**Test case first half of cnf must evaluate to true**

(a) +(b) +(c) +(d) +(e) +(f) +(g) +(h) +(i) +(j) +(!k) +(!l) +(!m)+ (!n)+ (!o)+(!p) +(!q) +(!r) +(!s)+(!t)+(!u)+(!v)+(!w)+(!x)+(!y)+(!z**)**

**Expected: 11111111111101100010101011**

**Result: 11111111111101100010101011**

**Results:**

====Solved=====

(!a\*c)+(!a\*c\*!e)+(!b\*c\*d\*!e)+(a\*!b\*c)+(!e\*f)

@ GENERATION 0 with 100110

====Solved=====

(a\*b)+(c\*d)+(e\*f)+(g\*h)

@ GENERATION 0 with 01111010

====Solved=====

(!a\*b)+(c\*d)+(e\*!f)+(g\*h)

@ GENERATION 0 with 10110001

====Solved=====

(r\*b)+(c\*d)+(c\*a\*!b\*b\*c\*z)

@ GENERATION 4 with 011011

====Solved=====

(a\*b\*c)+(c\*d\*e)

@ GENERATION 0 with 10111

====Solved=====

(f\*g\*k)+(n\*m)

@ GENERATION 0 with 10001

====Solved=====

(q\*z\*l\*v)+(z\*k\*r)+(m\*n\*f)+(b\*r\*t)

@ GENERATION 0 with 01100111101

====Solved=====

(!a\*b)+(c\*d)+(e\*!f)+(g\*h)+(i\*j)+(k\*l)+(m\*n)+(o\*p)+(q\*r)+(s\*t)+(u\*v)+(w\*x)+(y\*z)

@ GENERATION 9 with 10111110101011101001011111

====Solved=====

(!a\*b\*!c)+(c\*d\*!e\*!g\*!m)+(e\*f)+(g\*h)+(a\*!g\*!f)+(e\*f\*g\*k\*r)+(!e\*a\*b)

@ GENERATION 0 with 00110111000

====Solved=====

(a\*b)+(c\*d)

@ GENERATION 0 with 1110

@ GENERATION 148 with 11111111111101100010101011

====Solved=====

(a)+(b)+(c)+(d)+(e)+(f)+(g)+(h)+(i)+(j)+(k)+(l)+(m

+(u)+(v)+(w)+(x)+(y)+(z)

@ GENERATION 849 with 11111111111111111111111111

Observations: Most cnfs were solved within the first generation. This is due to the nature of the test cases.

IV.2: Operating Directions

To run this program navigate to the directory that is extracted to and type python driver.py