

# Computing IV Porfolio

- Ps0: SFML Hello World
- Ps1a: FibLFSR
- Ps1b: Image encryption program
- Ps2: Pythagoras Tree generator
- Ps3 Nbody Simulation
- Ps4: Karplus-Strong String Simulation
- Ps5: Needleman-Wunch DNA analysis
- Ps6: Markov Model

## **Ps0: SFML Hello World:**

### **Assignment Description:**

This was the first assignment given to us in Computing IV this assignment was simple and was to build a simple GUI environment using SFML. This assignment tasked us to create a sprite and to have the sprite move across the screen and to have an extra function for the sprite. This assignment seemed to be mostly for us to correctly install Linux and the SFML libraries.

### **Key concepts and Algorithms:**

This assignment did not have any algorithms for us to use like in later assignments. Instead the key concept need to finish the assignment was knowledge on how to use sprites in SFML and experience using the Spite and Drawable classes.

For this assignment I used a photo of Pikachu but with Homer Simpson's face instead as my sprite and another Simpson's themed background for the program. My sprite moved with the keyboard and shift would make the sprite rotate.

### **What I learned in this assignment:**

This assignment taught me about making programs with a GUI. Other than learning how to use SFML this program didn't teach me anything that I didn't already learn in other computing classes. I already had experience in a Linux environment from Computing II and using it on my own personal machine.

Ps0 Screenshot:



## Ps0 Source Code: main.cpp

---

```
1 // Name: John Simonson
2 // Date: 1/27/2020
3 // Assignment: ps0
4 #include <SFML/Graphics.hpp>
5
6 int main()
7 {
8     sf::RenderWindow window(sf::VideoMode(300, 250), "HW0");
9
10    sf::Texture texture;
11    if (!texture.loadFromFile("sprite.png"))
12        return EXIT_FAILURE;
13    sf::Sprite sprite(texture); //Load Sprite png to sprite
14    sf::Texture background;
15    if (!background.loadFromFile("background.png"))
16        return EXIT_FAILURE;
17    sf::Sprite Background(background); //Load Background png to sprite
18
19    while (window.isOpen())
20    {
21        sf::Event event;
22        while (window.pollEvent(event))
23        {
24            if (event.type == sf::Event::Closed)
25                window.close();
26        }
27        window.setFramerateLimit(120); //set framerate
28        window.clear(); // clear window
29        window.draw(Background); //Load background
30        window.draw(sprite); // Load sprite
31        window.display(); // display
32        if (event.type == sf::Event::KeyPressed){
33            if (event.key.code == sf::Keyboard::Up)
34            {
35                sprite.move(0.0, -2.0); // move up
36            }
37            if (event.key.code == sf::Keyboard::Down) // move down
38            {
39                sprite.move(0.0, 2.0);
40            }
41            if (event.key.code == sf::Keyboard::Left) // move left
42            {
43                sprite.move(-2.0, 0.0);
```

```
44 }
45 if (event.key.code == sf::Keyboard::Right) // move right
46 {
47     sprite.move(2.0, 0.0);
48 }
49 if (event.key.code == sf::Keyboard::LShift) // rotate clockwise
50 {
51     sprite.rotate(10.f);
52 }
53 }
54 }
55 return 0;
56 }
57 }
```

---

## **Ps1a: FibLFSR**

### **Assignment Description:**

This assignment required us to make a pseudo-random number generator. The random number generator is based on a Linear feedback shift register or LFSR. This will generate random numbers based on a given key. These numbers are output to the terminal. These pseudo-random numbers are used in the next part of the assignment.

### **Key Concepts and Algorithms:**

The main algorithm for this is a Linear feedback shift register which takes the last bit and XORs it the 3<sup>rd</sup> to last bit, then the 4<sup>th</sup> to last, and finally the 6<sup>th</sup> to last bit. This generated a pseudo-random number based on the 16 bit input key.

To accomplish the task of generating a random number we used a class called FibLFSR. This class holds all of the data and functions for generating the numbers. This is accomplished via the generate() and step() functions.

### **What I learned in this assignment:**

In this assignment I learned about how to generate pseudo-random numbers using C++. I learned about classes in Computing III and how the XOR operation works in Computing I and Logic Design. This assignment wasn't too difficult and mostly served to lay the ground work for the next part of this assignment.

## Ps1a: Screen shot

```
simo@Simo-Laptop:~/ps1$ ./main.o  
0110110000110011 19  
simo@Simo-Laptop:~/ps1$ ./test.o  
Running 1 test case...  
  
*** No errors detected  
simo@Simo-Laptop:~/ps1$ 
```

## Ps1a Source Code: Makefile

---

```
1 CC = g++
2 CFLAGS =      -Wall -Werror
3 DEPS =        FibLFSR.hpp
4
5 main: main.o FibLFSR.o main.cpp
6     g++ FibLFSR.hpp -Wall -Werror main.cpp -o main.o FibLFSR.o
7
8 FibLFSR: FibLFSR.cpp
9     g++ FibLFSR.cpp -o FibLFSR.o -Wall -Werror
10
11 test: FibLFSR.o test.cpp test.o
12     g++ FibLFSR.hpp -Wall -Werror test.cpp -o test.o FibLFSR.o -lboost_unit_test
13
14 all: FibLFSR.o test.cpp test.o
15     g++ FibLFSR.hpp -Wall -Werror test.cpp -o test.o FibLFSR.o -lboost_unit_test
16
17 clean:
18     rm main.o FibLFSR.o test.o
```

---

## Ps1a Source Code: main.cpp

---

```
1 // John Simonson
2 // 2/3/20
3 #include "FibLFSR.hpp"
4 using namespace std;
5
6 int main(int argc, char** argv){
7     FibLFSR a("0110001101100001");
8     int temp = a.generate(5);
9     cout << a << " " << temp << endl;
10    return 0;
11 }
```

---

## Ps1a Source Code: test.cpp

---

```
1 // John Simonson
2 // test.cpp for PS1a
3
4
5 #include <iostream>
```

```
6 #include <string>
7
8 #include "FibLFSR.hpp"
9
10 #define BOOST_TEST_DYN_LINK
11 #define BOOST_TEST_MODULE Main
12 #include <boost/test/unit_test.hpp>
13
14 BOOST_AUTO_TEST_CASE(sixteenBitsThreeTaps) {
15
16     FibLFSR l("0110001101100001");
17     BOOST_REQUIRE(l.step() == 1);
18     FibLFSR S("0110001101100001");
19     BOOST_REQUIRE(S.generate(5) == 19);
20 }
```

---

## Ps1a Source Code: FibLFSR.hpp

```
1 // John Simonson
2 // FibLFSR.hpp
3 // 2/3/20
4 #ifndef FIBLFSR_H
5 #define FIBLFSR_H
6 #endif
7 #include<iostream>
8 #include<string>
9 #include<cmath>
10 using namespace std;
11 class FibLFSR {
12 public:
13     FibLFSR(string seed);
14     int step();
15     int generate(int k);
16     friend ostream & operator <<(ostream& out, const FibLFSR c);
17 private:
18     string num;
19 };
```

---

## Ps1a Source Code: FibLFSR.cpp

```
1 // John Simonson
2 // FibLFSR.cpp
```

---

```
3 // 2/3/20
4 #include "FibLFSR.hpp"
5 using namespace std;
6
7 FibLFSR::FibLFSR(string seed){
8     this->num = seed;
9 }
10
11 ostream & operator <<(ostream& out, const FibLFSR c){
12     out << c.num;
13     return out;
14 }
15
16 int FibLFSR::step(){
17     int temp = this->num[0] ^ this->num[2];
18     temp = temp ^ this->num[3];
19     temp = temp ^ this->num[5];
20     int i;
21     for(i = 0; i <= 14; i++){
22         this->num[i] = this->num[i+1];
23     }
24     this->num[15] = '0' + temp;
25     return temp;
26 }
27
28 int FibLFSR::generate(int k){
29     string output;
30     for(int i = 0; i < k; i++){
31         output += to_string(this->step());
32     }
33     int x = stoi(output, nullptr, 2);
34     return x;
35 }
```

---

## **Ps1b: Image Encryption Program**

### **Assignment Description:**

This assignment was for us to make an image encryption program using the LFSR program in the previous assignment. The program takes an input file in the form of a png due to png files being uncompressed. The program takes each pixel in the image and XORs the color values for the pixel with a randomly generated integer from the LFSR program. This means that the program will turn into visual snow when run through the program and will return to normal if reentered into the program.

### **Key Concepts and Algorithms:**

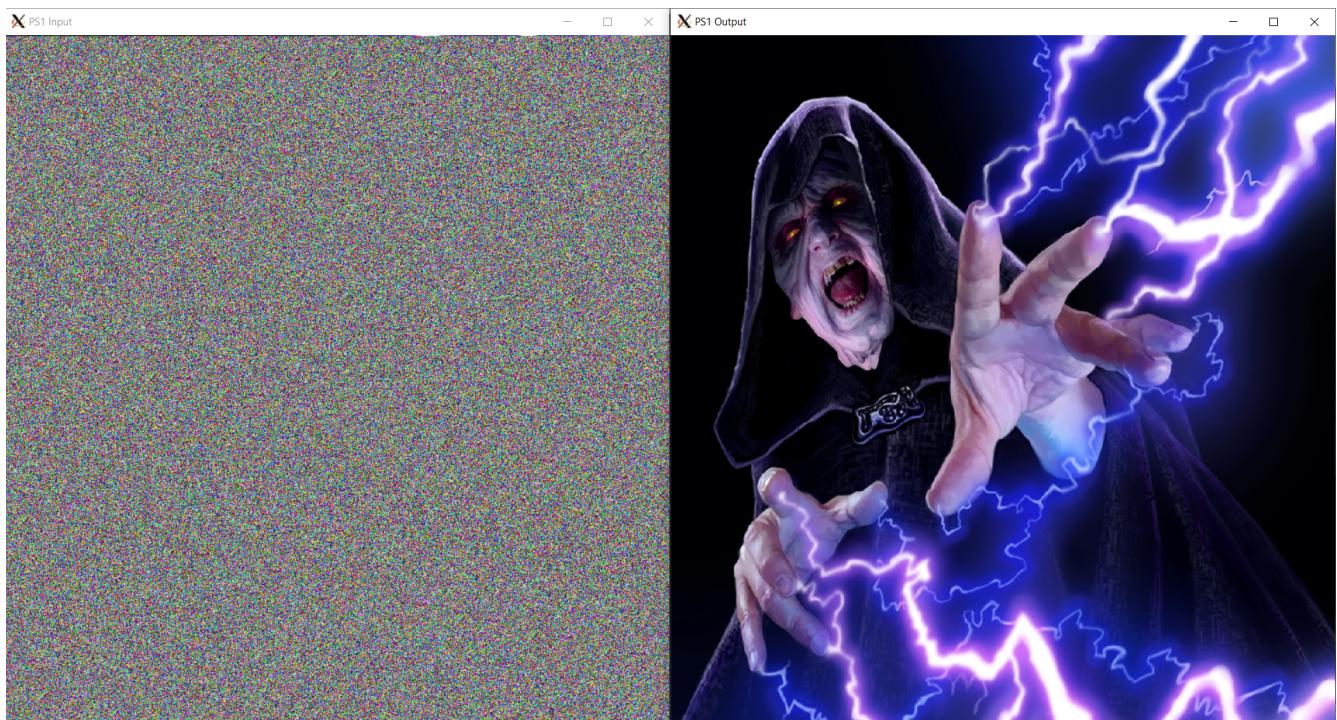
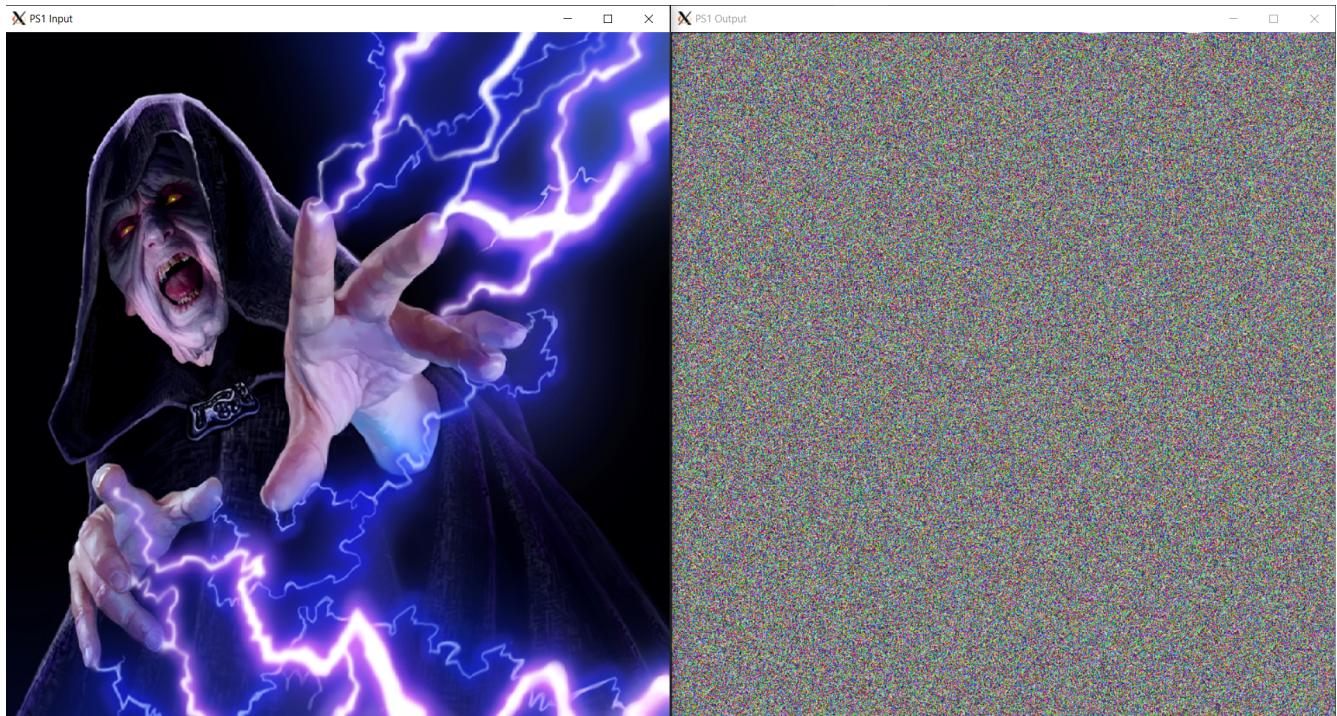
The key concepts in this project is the color values within a sprite object. Other concepts used in this assignment is the LFSR developed in Ps1a. For this assignment I took a photo of Emperor Palpatine from Star Wars. This picture would become unrecognizable when first run through the program but would return to normal once entered back into the program.

One issue I encountered while making this program was that the program would not function properly if the input file was not of the same resolution as the Palpatine.

### **What I learned in this assignment:**

In this assignment I learned how color values work for SFML. I also learned in the course of this project the differences between png files and other photo file formats such as jpg files. This assignment taught me more about how SFML windows and sprites work and gave me more experience with graphical programs.

## Ps1b: Screen shot



## Ps1b Source Code: Makefile

---

```
1 CC = g++
2 CFLAGS = -Wall -Werror -lsfml-graphics -lsfml-window -lsfml-system
3 DEPS = FibLFSR.hpp
4
5 all: PhotoMagic.o FibLFSR.o PhotoMagic.cpp
6     g++ FibLFSR.hpp PhotoMagic.cpp -o PhotoMagic FibLFSR.o -std=c++11 -Wall -Werror
7
8 FibLFSR: FibLFSR.cpp
9     g++ FibLFSR.cpp -o FibLFSR.o -Wall -Werror
10
11 clean:
12     rm PhotoMagic.o FibLFSR.o
```

---

## Ps1b Source Code: PhotoMagic.cpp

---

```
1 /*
2 Name: John Simonson
3 Date: 2/10/2020
4 PS1b
5 */
6 #include<SFML/Graphics.hpp>
7 #include <SFML/Graphics/Image.hpp>
8 #include"FibLFSR.hpp"
9 #include<unistd.h>
10 #include<string>
11 using namespace std;
12
13 int X = 959;
14 int Y = 832;
15
16 // transforms image using FibLFSR
17 void transform( sf::Image&, FibLFSR* );
18
19 int main(int argc, char* argv[]){
20     string seed = argv[3];
21     FibLFSR a(seed);
22     sf::RenderWindow window1(sf::VideoMode(X, Y), "PS1 Input");
23
24     sf::Image image1;
25     if (!(image1.loadFromFile(argv[1])))
26         std::cout << "Cannot load image"; //Load Image
27 }
```

```

28     sf::Texture texture1;
29     texture1.loadFromImage(image1); //Load Texture from image
30     sf::Sprite Texture1;
31     Texture1.setTexture(texture1);
32
33     Texture1.getTexture()->copyToImage().saveToFile("output-file.png");
34
35     sf::RenderWindow window2(sf::VideoMode(X, Y), "PS1 Output");
36
37     sf::Image image2;
38     if (!(image2.loadFromFile(argv[2])))
39         std::cout << "Cannot load image"; //Load Image
40
41         transform(image2, &a);
42
43         sf::Texture texture2;
44         texture2.loadFromImage(image2); //Load Texture from image
45         sf::Sprite Texture2;
46         Texture2.setTexture(texture2);
47
48         Texture2.getTexture()->copyToImage().saveToFile("output-file.png");
49
50
51     while (window1.isOpen() && window2.isOpen()){
52         sf::Event event;
53         while (window1.pollEvent(event)) {
54             if (event.type == sf::Event::Closed)
55                 window1.close();
56         }
57         while (window2.pollEvent(event)) {
58             if (event.type == sf::Event::Closed)
59                 window2.close();
60         }
61         window1.clear();
62         window1.draw( Texture1 );
63         window1.display();
64         window2.clear();
65         window2.draw(Texture2);
66         window2.display();
67     }
68
69
70     return 0;
71 }
72
73

```

```
74
75 void transform( sf::Image& image2, FibLFSR* a){
76     sf::Color buffer(0, 0, 0);
77
78     for(int i = X; i > 0; i--){
79         for(int j = Y; j > 0; j--){
80             buffer = image2.getPixel(i, j);
81             buffer.r = buffer.r ^ a->generate(8);
82             buffer.g = buffer.g ^ a->generate(8);
83             buffer.b = buffer.b ^ a->generate(8);
84             image2.setPixel(i, j, buffer);
85         }
86     }
87
88     return;
89 }
```

---

## Ps1b Source Code: FibLFSR.hpp

```
1 // John Simonson
2 // FibLFSR.hpp
3 // 2/3/20
4 #ifndef FIBLFSR_H
5 #define FIBLFSR_H
6 #endif
7 #include<iostream>
8 #include<string>
9 #include<cmath>
10 using namespace std;
11 class FibLFSR {
12 public:
13     FibLFSR(string seed);
14     int step();
15     int generate(int k);
16     friend ostream & operator <<(ostream& out, const FibLFSR c);
17 private:
18     string num;
19 };
```

---

## Ps1b Source Code: FibLFSR.cpp

```
1 // John Simonson
```

---

```
2 // FibLFSR.cpp
3 // 2/3/20
4 #include "FibLFSR.hpp"
5 using namespace std;
6
7 FibLFSR::FibLFSR(string seed){
8     this->num = seed;
9 }
10
11 ostream & operator <<(ostream& out, const FibLFSR c){
12     out << c.num;
13     return out;
14 }
15
16 int FibLFSR::step(){
17     int temp = this->num[0] ^ this->num[2];
18     temp = temp ^ this->num[3];
19     temp = temp ^ this->num[5];
20     int i;
21     for(i = 0; i <= 14; i++){
22         this->num[i] = this->num[i+1];
23     }
24     this->num[15] = '0' + temp;
25     return temp;
26 }
27
28 int FibLFSR::generate(int k){
29     string output;
30     for(int i = 0; i < k; i++){
31         output += to_string(this->step());
32     }
33     int x = stoi(output, nullptr, 2);
34     return x;
35 }
```

---

## **Ps2: Pythagoras Tree**

### **Assignment Description:**

For this assignment we were tasked to create a recursive program that generates a fractal called a Pythagoras tree. This fractal starts with a square then off the top of that square 2 smaller squares angled 45 degrees and -45 degrees from the base square's rotation. These smaller squares then serve as the base for more squares and so on. The program we were tasked to write had to have a recursive function that performed the following algorithm. We also received extra points for making the program generate trees with different angles, for animation, and for color.

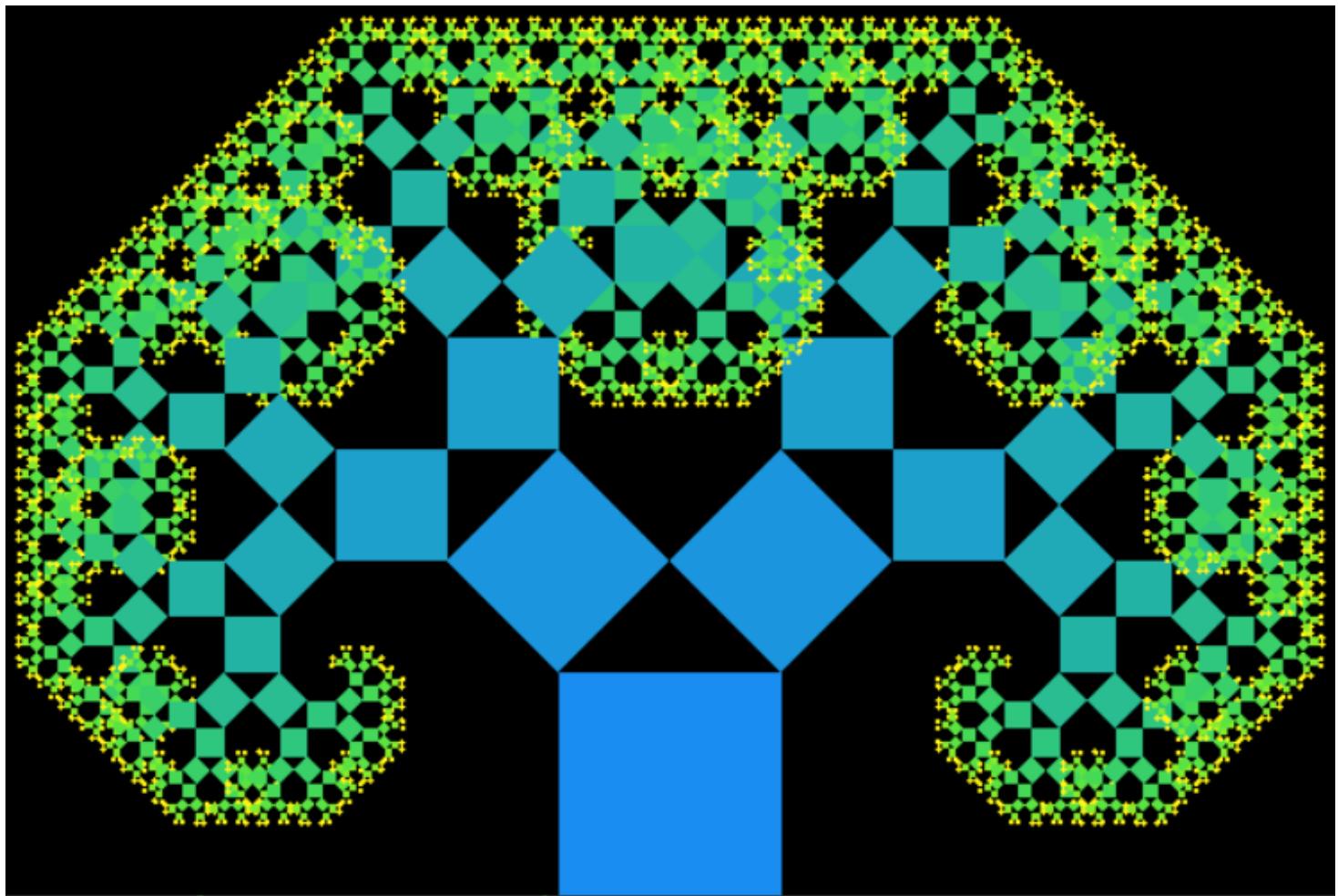
### **Key Concepts and Algorithms:**

The basis for this assignment is a recursive function. The main algorithm that generates the next square's size is the original square's scale multiplied by 1 over the square root of 2. Unlike the other assignments in this semester this assignment didn't use a class or memory allocation to generate the end result.

### **What I learned in this assignment:**

In the course of making this assignment I gained a greater understanding of recursive functions. I also learned about SFML rectangle shapes and other shapes in the SFML documentation that I read when completing this assignment.

## Ps2: Screen shot



## Ps5 Source Code: Makefile

---

```
1 CC= g++
2 CFLAGS= -g -O3 -Wall -Werror -std=c++0x
3 DEPS= -lsfml-system
4
5 all: ED
6
7 ED: main.o ED.o
8     $(CC) main.o ED.o -o ED $(DEPS)
9
10 main.o: main.cpp ED.hpp
11     $(CC) -c main.cpp ED.hpp $(CFLAGS)
12
13 ED.o: ED.cpp ED.hpp
14     $(CC) -c ED.cpp ED.hpp $(CFLAGS)
15
16 clean:
17     rm *.o
18     rm *.gch
19     rm ED
```

---

## Ps5 Source Code: PTree.cpp

---

```
1 // Name: John Simonson
2 // Date: 1/27/2020
3 // Assignment: ps2
4 #include<SFML/Graphics.hpp>
5 #include<iostream>
6 #include<string>
7 #include<cmath>
8 #include<vector>
9 using namespace std;
10 using sf::Color;
11 //argv[1] = L = ("Length of window")
12 //argv[2] = N = ("Depth of tree")
13 //Scale next = size * 1/sqrt2
14
15 void draw(int newDepth, sf::RectangleShape root, sf::RenderWindow &window);
16
17 int length;
18 int width;
19 int size;
20 int depth;
```

```

21 int main(int argc, char* argv[]){
22     string temp = argv[1];
23     string tempDepth = argv[2];
24     depth = stoi(tempDepth);
25     size = stoi(temp);
26     length = size * 6;
27     width = size * 4;
28
29     sf::RenderWindow window(sf::VideoMode(length, width), "PS2");
30     sf::RectangleShape root(sf::Vector2f(size, size));
31     root.setPosition(length/2 - size/2, width-size);
32     root.setFillColor(Color((255/depth),255-(depth*10), + (depth*20)));
33     window.draw(root);
34     draw((depth - 1), root, window);
35
36
37     while (window.isOpen())
38    {
39         sf::Event event;
40         while (window.pollEvent(event))
41        {
42             if (event.type == sf::Event::Closed)
43                 window.close();
44
45
46
47         }
48     }
49     return 0;
50 }
51
52 void draw(int newDepth, sf::RectangleShape root, sf::RenderWindow &window){
53     sf::Event event;
54     while (window.pollEvent(event))
55     {
56         if (event.type == sf::Event::Closed)
57             window.close();
58
59     }
60
61     if(newDepth <= 0){
62         return;
63     }
64     sf::RectangleShape left, right;
65     left.setOrigin(0, root.getSize().y);
66     right.setOrigin(root.getSize().x, root.getSize().y);

```

```
67     left.setSize(root.getSize());
68     right.setSize(root.getSize());
69     left.setScale(root.getScale().x*(1/sqrt(2)),root.getScale().y*(1/sqrt(2)));
70     right.setScale(root.getScale().x*(1/sqrt(2)),root.getScale().y*(1/sqrt(2)));
71     left.setPosition(root.getTransform().transformPoint(root.getPoint(0)));
72     left.setRotation(root.getRotation() - 45);
73     right.setPosition(root.getTransform().transformPoint(root.getPoint(1)));
74     right.setRotation(root.getRotation() + 45);
75
76     left.setFillColor(Color((255/newDepth),255-(newDepth*10), (newDepth*20)));
77     right.setFillColor(Color((255/newDepth),255-(newDepth*10), (newDepth*20)));
78     window.draw(right);
79     window.display();
80     draw((newDepth - 1), right, window);
81     window.draw(left);
82     window.display();
83     draw((newDepth - 1), left, window);
84     return;
85 }
```

---

# Ps3: Nbody simulation

## **Assignment Description:**

For the 4<sup>th</sup> assignment of the semester we were tasked to create an Nbody simulation. An Nbody simulation is a physics simulation that takes an objects mass and distance to other objects in order to simulate gravitational forces. We used this type of simulation to make a recreation of the solar system. This program will simulate gravitational forces for a given system over a certain length of time per second. I also received extra points for adding music and creating a new universe for my simulation.

## **Key Concepts and Algorithms:**

This assignment used real physics formulas to calculate accurate simulations of celestial bodies. The main formula used is  $F = (G * M1 * M2) / R^2$  this formula calculates the forces on the bodies and uses them to calculate the new acceleration for the bodies. For this project we used 2 classes 1 class called CelestialBodies that holds all of the relevant data members and functions to calculate forces and 1 called Universe that generates and holds the CelestialBodies.

This assignment uses smart pointers which are an object that functions as a pointer and manages the memory allocation for the user. In this assignment we used a vector that hold shared pointers to CelestialBodies. This allows the Universe to hold a near infinite number of objects. These objects are given to the program via an input text file.

## **What I learned in this assignment:**

During this assignment I learned how to use smart pointers a concept in C++ that I had no experience in before starting this assignment. I also learned how to input data into a program from an input file with std::cin using the input stream. Before this assignment the only experience I had with input text files was file pointers in C.

## Ps3: Screen shot



## Ps3a Source Code: Makefile

---

```
1 CFLAGS = -Wall -Werror -std=c++11 -pedantic
2 DEPS = -lboost_unit_test_framework
3
4
5 ps4a: test.o RingBuffer.o
6     g++ test.cpp RingBuffer.cpp headers/RingBuffer.h -o ps4a $(CFLAGS) $(DEPS)
7
8 test.o:
9     g++ test.cpp RingBuffer.cpp headers/RingBuffer.h -o test.o $(CFLAGS) $(DEPS)
10
11 RingBuffer.o:
12     g++ -c RingBuffer.cpp headers/RingBuffer.h $(CFLAGS)
13
14 clean:
15     rm ps4a
16     rm *.o
17     rm headers/*.gch
18     rm *.out
```

---

## Ps3b Source Code: Makefile

---

```
1 CC= g++
2 CFLAGS= -Wall -Werror -std=c++11
3 DEPS= -lsfml-graphics -lsfml-window -lsfml-system -lsfml-audio
4
5 KSGuitarSim:    KSGuitarSim.o StringSound.o RingBuffer.o
6     $(CC) KSGuitarSim.o StringSound.o RingBuffer.o -o KSGuitarSim $(DEPS)
7
8 main.o: main.cpp StringSound.h
9     $(CC) -c main.cpp StringSound.h $(CFLAGS) $(DEPS)
10
11 StringSound.o: StringSound.cpp StringSound.h
12     $(CC) -c StringSound.cpp StringSound.h $(CFLAGS) $(DEPS)
13
14 RingBuffer.o: RingBuffer.cpp RingBuffer.h
15     $(CC) -c RingBuffer.cpp RingBuffer.h $(CFLAGS) $(DEPS)
16
17 clean:
18     rm *.o
19     rm *.gch
20     rm KSGuitarSim
```

---

## Ps3a Source Code: main.cpp

---

```
1 #include "nBody.hpp"
2 using namespace sf;
3 using namespace std;
4
5
6 int main(int argc, char* argv[]){
7     //Open Window
8     RenderWindow window(sf::VideoMode(500, 500), "PS3a");
9     //Background
10    Texture backgroundTexture;
11    Sprite background;
12    Vector2u textureSize;
13    Vector2u windowHeight;
14    if(!backgroundTexture.loadFromFile("background.jpg"))
15        return -1;
16    else{
17        textureSize = backgroundTexture.getSize();
18        windowHeight = window.getSize();
19        //setting scale of background
20        float x_scale = (float) windowHeight.x / textureSize.x;
21        float y_scale = (float) windowHeight.y / textureSize.y;
22        background.setTexture(backgroundTexture);
23        background.setScale(x_scale, y_scale);
24    }
25    string temp_particles;
26    string temp_radius;
27    cin >> temp_particles;
28    cin >> temp_radius;
29    int particles = atoi(temp_particles.c_str());
30    double radius = atof(temp_radius.c_str());
31    unique_ptr<Universe> a(new Universe(particles, radius, window));
32
33
34    while (window.isOpen())
35    {
36        Event event;
37        while (window.pollEvent(event))
38        {
39            if (event.type == Event::Closed)
40                window.close();
41        }
42        window.clear();
```

```

44         window.draw(background);
45         for(int i = 0; i < particles; i++){
46             a->vectorOfBodies[i]->draw(window);
47         }
48         window.display();
49     }
50 }
51
52     return 0;
53 }
```

---

### Ps3b Source Code: main.cpp

```

1 #include "nBody.hpp"
2 using namespace sf;
3 using namespace std;
4
5
6 int main(int argc, char* argv[]){
7     //args
8     double T = atof(argv[1]);
9     double deltaT = atof(argv[2]);
10
11    //Open Window
12    RenderWindow window(sf::VideoMode(500, 500), "PS3a");
13    window.setFramerateLimit(60);
14    //Music
15    Music musicFile;
16    musicFile.openFromFile("DueloftheFates.ogg");
17    musicFile.play();
18    musicFile.setLoop(true);
19    //Clock
20    Font font;
21    if (!font.loadFromFile("arial.ttf"))
22    {
23        cout << "Font can't load." << endl;
24    }
25    Text text;
26    text.setFont(font);
27    text.setCharacterSize(15);
28    text.setFillColor(Color::White);
29    text.setString("Time: " + to_string(0));
30
31    //Background
```

```

32     Texture backgroundTexture;
33     Sprite background;
34     Vector2u textureSize;
35     Vector2u windowSize;
36     if(!backgroundTexture.loadFromFile("starfield.jpg"))
37         return -1;
38     else{
39         textureSize = backgroundTexture.getSize();
40         windowSize = window.getSize();
41         //setting scale of background
42         float x_scale = (float) windowSize.x / textureSize.x;
43         float y_scale = (float) windowSize.y / textureSize.y;
44         background.setTexture(backgroundTexture);
45         background.setScale(x_scale, y_scale);
46     }
47     string temp_particles;
48     string temp_radius;
49     cin >> temp_particles;
50     cin >> temp_radius;
51     int particles = atoi(temp_particles.c_str());
52     double radius = atof(temp_radius.c_str());
53     unique_ptr<Universe> a(new Universe(particles, radius, window));
54
55
56     for(double j = 0 ; j < T; j+= deltaT){
57
58         Event event;
59         while (window.pollEvent(event))
60         {
61             if (event.type == Event::Closed)
62                 window.close();
63         }
64         window.clear();
65         window.draw(background);
66         text.setString("Time: " + to_string(j));
67         window.draw(text);
68         for(int i = 0; i < particles; i++){
69             a->vectorOfBodies[i]->draw(window);
70         }
71         a->update(a->vectorOfBodies);
72         a->move(a->vectorOfBodies, deltaT);
73         window.display();
74
75
76     }
77     for(int i = 0; i < particles; i++){

```

```
78         cout << a->vectorOfBodies[i];
79     }
80     return 0;
81 }
```

---

## Ps3a Source Code: nBody.hpp

```
1 #include<iostream>
2 #include<sstream>
3 #include<string>
4 #include<cstdlib>
5 #include<vector>
6 #include<memory>
7 #include<SFML/Graphics.hpp>
8 using namespace sf;
9 using namespace std;
10
11 const int window_length = 500;
12 const int window_width = 500;
13
14 class CelestialBody{
15     public:
16     CelestialBody();
17     CelestialBody(double xcoordinate, double ycoordinate, double xvelocity, double yvelocity);
18     friend istream& operator>> (istream &input, CelestialBody &body);
19     friend ostream& operator<< (ostream &output, CelestialBody &body);
20     void set_radius(float Radius);
21     void draw(RenderWindow &window);
22     //void setPosition(RenderWindow &window);
23     Sprite sprite;
24     Texture texture;
25     double x_coordinate;
26     double y_coordinate;
27     double x_velocity;
28     double y_velocity;
29     double mass;
30     string fileName;
31     int radius;
32 };
33
34 class Universe : public CelestialBody{
35     public:
36     Universe(int _particles, int radius, RenderWindow& window);
37     //void draw(RenderWindow &window);
```

```
38     vector< shared_ptr<CelestialBody> > vectorOfBodies;
39     int particles;
40 };


---


```

## Ps3b Source Code: nBody.hpp

```
1 #include<iostream>
2 #include<sstream>
3 #include<string>
4 #include<cstdlib>
5 #include<vector>
6 #include<memory>
7 #include<cmath>
8 #include<SFML/Graphics.hpp>
9 #include<SFML/Audio.hpp>
10 #include<SFML/Graphics/Font.hpp>
11 #include<SFML/Graphics/Text.hpp>
12 using namespace sf;
13 using namespace std;
14
15 const int window_length = 500;
16 const int window_width = 500;
17
18 class CelestialBody{
19     public:
20     CelestialBody();
21     CelestialBody(double xcoordinate, double ycoordinate, double xvelocity, double
22     friend istream& operator>> (istream &input, CelestialBody &body);
23     friend ostream& operator<< (ostream &output, shared_ptr<CelestialBody> &body)
24     void set_radius(float Radius);
25     void draw(RenderWindow &window);
26     void update(vector<shared_ptr<CelestialBody> > &body);
27     void move(vector<shared_ptr<CelestialBody> > &body, double delta_T);
28     Sprite sprite;
29     Texture texture;
30     double x_coordinate;
31     double y_coordinate;
32     double x_velocity;
33     double y_velocity;
34     double x_force;
35     double y_force;
36     double mass;
37     string fileName;
```

```

38     double radius;
39     int particles;
40 };
41
42 class Universe : public CelestialBody{
43     public:
44     Universe(int _particles, int radius, RenderWindow& window);
45
46     vector< shared_ptr<CelestialBody> > vectorOfBodies;
47 };

```

---

## Ps3a Source Code: nBody.cpp

```

1 #include "nBody.hpp"
2 using namespace sf;
3 using namespace std;
4
5
6 //Constructors
7 CelestialBody::CelestialBody(){
8     x_coordinate = 0;
9     y_coordinate = 0;
10    x_velocity = 0;
11    y_velocity = 0;
12    mass = 0;
13    radius = 0;
14    fileName = " ";
15 }
16
17 CelestialBody::CelestialBody(double xcoordinate, double ycoordinate, double xvel
18     x_coordinate = xcoordinate;
19     y_coordinate = ycoordinate;
20     x_velocity = xvelocity;
21     y_velocity = yvelocity;
22     mass = _mass;
23     radius = _radius;
24     fileName = file_Name;
25     if(!texture.loadFromFile(fileName))
26         return;
27
28     sprite.setTexture(texture);
29     sprite.setPosition(Vector2f(x_coordinate, y_coordinate));
30
31 }

```

```

32
33 Universe::Universe(int _particles, int radius, RenderWindow& window){
34     particles = _particles;
35
36     for(int i = 0; i < particles; i++){
37         shared_ptr<CelestialBody> temp(new CelestialBody());
38
39         cin >> *temp;
40
41         temp->set_radius(radius);
42
43         temp->sprite.setPosition(250 - ((temp->x_coordinate / radius) * 2 + 250) + 250);
44
45         this->vectorOfBodies.push_back(temp);
46
47         cout << *temp;
48
49     }
50
51 }
52 //Functions
53 void CelestialBody::draw(RenderWindow &window){
54     window.draw(this->sprite);
55     return;
56 }
57
58 void CelestialBody::set_radius(float Radius){
59     radius = Radius;
60     return;
61 }
62
63 //IO streams
64 istream& operator>>(istream &input, CelestialBody &body){
65     input >> body.x_coordinate;
66     input >> body.y_coordinate;
67     input >> body.x_velocity;
68     input >> body.y_velocity;
69     input >> body.mass;
70     input >> body.fileName;
71
72     if(!body.texture.loadFromFile(body.fileName)){
73         return input;
74     }
75
76     body.sprite.setTexture(body.texture);
77     body.sprite.setPosition(Vector2f(body.x_coordinate, body.y_coordinate));

```

```

78     return input;
79 }
80 }
81
82 ostream& operator<< (ostream &output, CelestialBody &body){
83     output << "X_Coordinate : " << body.x_coordinate << endl << "Y_Coordinate : "
84     output << "X_Velocity : " << body.x_velocity << endl << "Y_Velocity : " << body.y_velocity << endl << "Mass : " << body.mass << endl << "File : " << body.fileName << endl;
85     return output;
86 }
87 }
```

---

## Ps3b Source Code: nBody.cpp

```

1 #include "nBody.hpp"
2 using namespace sf;
3 using namespace std;
4
5
6 //Constructors
7 CelestialBody::CelestialBody(){
8     x_coordinate = 0;
9     y_coordinate = 0;
10    x_velocity = 0;
11    y_velocity = 0;
12    x_force = 0;
13    y_force = 0;
14    mass = 0;
15    radius = 0;
16    fileName = " ";
17 }
18
19 CelestialBody::CelestialBody(double xcoordinate, double ycoordinate, double xvelocity,
20     double yvelocity, double mass, double radius, string fileName)
21 {
22     x_coordinate = xcoordinate;
23     y_coordinate = ycoordinate;
24     x_velocity = xvelocity;
25     y_velocity = yvelocity;
26     mass = _mass;
27     radius = _radius;
28     fileName = file_Name;
29     if(!texture.loadFromFile(fileName))
30         return;
31     sprite.setTexture(texture);
32     sprite.setPosition(Vector2f(x_coordinate, y_coordinate));
33 }
```

```

32
33     return;
34 }
35
36 Universe::Universe(int _particles, int radius, RenderWindow& window){
37     particles = _particles;
38
39     for(int i = 0; i < particles; i++){
40         shared_ptr<CelestialBody> temp(new CelestialBody());
41
42         cin >> *temp;
43
44         temp->set_radius(radius);
45
46         temp->sprite.setPosition(250 - ((temp->x_coordinate / radius) * 2 + 250) + 250);
47
48         this->vectorOfBodies.push_back(temp);
49
50     }
51     return;
52 }
53 }
54 //Functions
55 void CelestialBody::draw(RenderWindow &window){
56     window.draw(this->sprite);
57     return;
58 }
59
60 void CelestialBody::set_radius(float Radius){
61     radius = Radius;
62     return;
63 }
64
65 void CelestialBody::update(vector<shared_ptr<CelestialBody> > &body) {
66     for(int i = 0; i < particles; i++){
67         for(int j = 0; j < particles; j++){
68             if(i != j){
69                 double change_in_x = body[j]->x_coordinate - body[i]->x_coordinate;
70                 double change_in_y = body[j]->y_coordinate - body[i]->y_coordinate;
71                 double r = sqrt((change_in_x * change_in_x) + (change_in_y * change_in_y));
72                 double beeg_G = 6.67e-11;
73                 double T_force = ((beeg_G*body[i]->mass*body[j]->mass) / (r * r));
74                 body[i]->x_force = body[i]->x_force + (T_force *(change_in_x / r));
75                 body[i]->y_force = body[i]->y_force + (T_force *(change_in_y / r));
76
77             }
78         }
79     }
80 }
```

```

78         }
79     }
80     return;
81 }
82
83 void CelestialBody::move(vector<shared_ptr<CelestialBody>> &body, double delta_
84     for(int i = 0; i < particles; i++){
85         double x_acceleration = body[i]->x_force/body[i]->mass;
86         double y_acceleration = body[i]->y_force/body[i]->mass;
87         body[i]->x_velocity = body[i]->x_velocity + (x_acceleration* delta_T);
88         body[i]->y_velocity = body[i]->y_velocity + (y_acceleration* delta_T);
89         body[i]->x_coordinate = body[i]->x_coordinate + (body[i]->x_velocity* de
90         body[i]->y_coordinate = body[i]->y_coordinate + (body[i]->y_velocity* de
91         body[i]->sprite.setPosition(250 - ((body[i]->x_coordinate / body[i]->rad
92         body[i]->x_force = body[i]->y_force = 0;
93     }
94     return;
95 }
96
97 //IO streams
98 istream& operator>>(istream &input, CelestialBody &body){
99     input >> body.x_coordinate;
100    input >> body.y_coordinate;
101    input >> body.x_velocity;
102    input >> body.y_velocity;
103    input >> body.mass;
104    input >> body.fileName;
105
106    if(!body.texture.loadFromFile(body.fileName)){
107        return input;
108    }
109
110    body.sprite.setTexture(body.texture);
111    body.sprite.setPosition(Vector2f(body.x_coordinate, body.y_coordinate));
112    body.sprite.setOrigin(10,10);
113
114    return input;
115 }
116
117 ostream& operator<< (ostream &output, shared_ptr<CelestialBody> &body){
118     output << body->x_coordinate << "      " << body->y_coordinate << "
119     ";
120     output << body->x_velocity << "      " << body->y_velocity << "
121     ";
122     output << body->mass << "      " << body->fileName << endl;
123     return output;

```

122 }

---

# Ps4: Karplus-Strong String Simulation

## **Assignment Description:**

For this assignment we were tasked to simulate the plucking of a guitar string using the Karplus-Strong algorithm. This algorithm combined with a queue called a Ringbuffer creates the sound of a string reverberating. We were tasked to take the keyboard input and produce a different note of a guitar being played. However I was unable to make the program produce different notes.

## **Key Concepts and Algorithms:**

The key algorithm for this assignment was the Karplus-Strong Algorithm which calculates the energy decay from the string being plucked. The algorithm is:

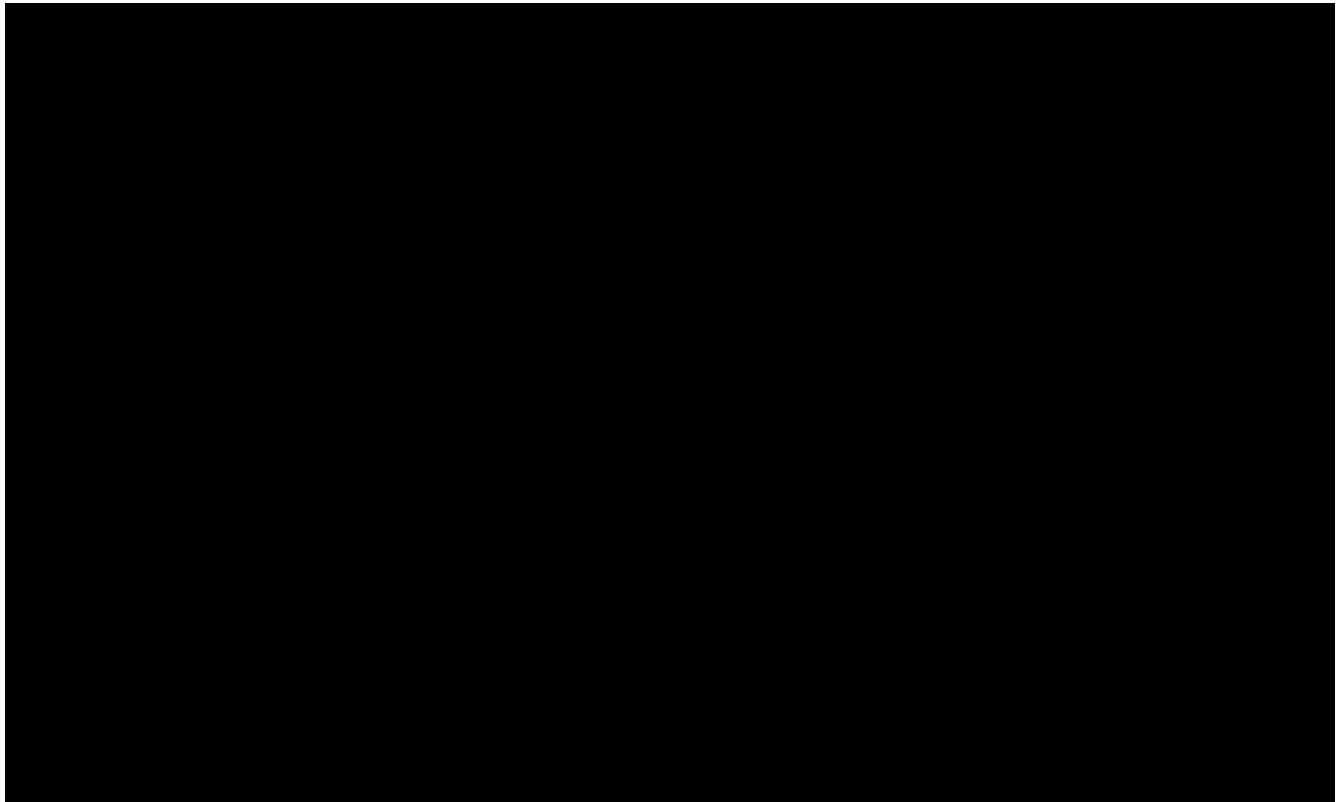
$0.996 * 1/2(a + b)$  where a and b are the next 2 frequencies in the Ring Buffer.

The program uses a class called StringSound that contains a shared pointer to a Ring Buffer. My understanding of shared pointers from the previous assignment made this assignment easier to accomplish. The class is inherited from sf::Sound this allows for it to be played with .play().

## **What I learned in this assignment:**

While making this assignment I learned how c++ modules and sf::Sound works based on reading the Documentation for the class. This further understanding on how to use different libraries in c++ is something I didn't truly understand until this assignment. The ability to create music notes from a keyboard input could become useful in future projects within c++ and I look forward to creating them in the future.

## **Ps4: Screen shot**



## Ps4a Source Code: Makefile

---

```
1 CFLAGS = -Wall -Werror -std=c++11 -pedantic
2 DEPS = -lboost_unit_test_framework
3
4
5 ps4a: test.o RingBuffer.o
6     g++ test.cpp RingBuffer.cpp headers/RingBuffer.h -o ps4a $(CFLAGS) $(DEPS)
7
8 test.o:
9     g++ test.cpp RingBuffer.cpp headers/RingBuffer.h -o test.o $(CFLAGS) $(DEPS)
10
11 RingBuffer.o:
12     g++ -c RingBuffer.cpp headers/RingBuffer.h $(CFLAGS)
13
14 clean:
15     rm ps4a
16     rm *.o
17     rm headers/*.gch
18     rm *.out
```

---

## Ps4b Source Code: Makefile

---

```
1 CC= g++
2 CFLAGS= -Wall -Werror -std=c++11
3 DEPS= -lsfml-graphics -lsfml-window -lsfml-system -lsfml-audio
4
5 KSGuitarSim:    KSGuitarSim.o StringSound.o RingBuffer.o
6     $(CC) KSGuitarSim.o StringSound.o RingBuffer.o -o KSGuitarSim $(DEPS)
7
8 main.o: main.cpp StringSound.h
9     $(CC) -c main.cpp StringSound.h $(CFLAGS) $(DEPS)
10
11 StringSound.o: StringSound.cpp StringSound.h
12     $(CC) -c StringSound.cpp StringSound.h $(CFLAGS) $(DEPS)
13
14 RingBuffer.o: RingBuffer.cpp RingBuffer.h
15     $(CC) -c RingBuffer.cpp RingBuffer.h $(CFLAGS) $(DEPS)
16
17 clean:
18     rm *.o
19     rm *.gch
20     rm KSGuitarSim
```

---

## Ps4a Source Code: test.cpp

---

```
1 // Copyright 2020 John Simonson
2
3 #define BOOST_TEST_DYN_LINK
4 #define BOOST_TEST_MODULE Main
5 #include <boost/test/unit_test.hpp>
6 #include "headers/RingBuffer.h"
7
8
9 // constructor tests
10 BOOST_AUTO_TEST_CASE(constructor) {
11     BOOST_REQUIRE_THROW(RingBuffer(0), std::invalid_argument);
12     BOOST_REQUIRE_THROW(RingBuffer(-1), std::invalid_argument);
13     BOOST_REQUIRE_NO_THROW(RingBuffer(21));
14 }
15
16 // size test
17 BOOST_AUTO_TEST_CASE(size) {
18     RingBuffer temp(21);
19     BOOST_REQUIRE(temp.size() == 0);
20 }
21
22 // isEmpty test
23 BOOST_AUTO_TEST_CASE(isEmpty) {
24     RingBuffer temp(21);
25     BOOST_REQUIRE(temp.isEmpty() == true);
26 }
27
28 // isFull test
29 BOOST_AUTO_TEST_CASE(isFull) {
30     RingBuffer temp(21);
31     BOOST_REQUIRE(temp.isFull() == false);
32 }
33
34 // enqueue test
35 BOOST_AUTO_TEST_CASE(enqueue) {
36     RingBuffer temp(1);
37     BOOST_REQUIRE(temp.size() == 0);
38     temp.enqueue(50);
39     BOOST_REQUIRE(temp.size() == 1);
40     BOOST_REQUIRE_THROW(temp.enqueue(21), std::runtime_error);
41 }
42
43 // dequeue test
```

```

44 BOOST_AUTO_TEST_CASE(dequeue) {
45     RingBuffer temp(21);
46     BOOST_REQUIRE_THROW(temp.dequeue(), std::runtime_error);
47 }
48
49
50 // peek test
51 BOOST_AUTO_TEST_CASE(peek) {
52     RingBuffer temp(21);
53     BOOST_REQUIRE_THROW(temp.peek(), std::runtime_error);
54 }
```

---

## Ps4b Source Code: KSGuitarSim.cpp

```

1 #include <SFML/Graphics.hpp>
2 #include <SFML/System.hpp>
3 #include <SFML/Audio.hpp>
4 #include <SFML/Window.hpp>
5
6 #include <math.h>
7 #include <limits.h>
8
9 #include <iostream>
10 #include <string>
11 #include <exception>
12 #include <stdexcept>
13 #include <vector>
14
15 #include "RingBuffer.h"
16 #include "StringSound.h"
17
18 #define CONCERT_A 440.0
19 #define SAMPLES_PER_SEC 44100
20 const int num_of_keys = 37;
21 std::string keyboard = "q2we4r5ty7u8i9op-[=zxdcfvgbnjmkg,.;/` ";
22 std::vector<sf::Int16> makeSamples(StringSound gs) {
23     std::vector<sf::Int16> samples;
24
25     gs.pluck();
26     int duration = 8; // seconds
27     int i;
28     for (i = 0; i < SAMPLES_PER_SEC * duration; i++) {
29         gs.tic();
30         samples.push_back(gs.sample());
```

```

31     }
32
33     return samples;
34 }
35
36 int main() {
37     sf::RenderWindow window(sf::VideoMode(300, 200), "ps4b");
38     sf::Event event;
39     double freq;
40     std::vector<sf::Int16> samples;
41     std::vector<std::vector<sf::Int16>> vector_of_samples(num_of_keys);
42     std::vector<sf::SoundBuffer> sound_buffer(num_of_keys);
43     std::vector<sf::Sound> sounds(num_of_keys);
44     for(int i = 0; i < num_of_keys; i++){
45         freq = CONCERT_A * pow(2, 3.0/12.0);
46         StringSound gs2 = StringSound(freq);
47         samples = makeSamples(gs2);
48         vector_of_samples[i] = samples;
49
50         sound_buffer[i].loadFromSamples(&vector_of_samples[i][0], vector_of_samples[i].size());
51
52         sounds[i].setBuffer(sound_buffer[i]);
53     }
54
55     while (window.isOpen()) {
56         while (window.pollEvent(event)) {
57             if (event.type == sf::Event::Closed)
58                 window.close();
59             if(event.type == sf::Event::TextEntered){
60                 char ascii = (char)(event.text.unicode);
61                 for(int i = 0; i < num_of_keys; i++){
62                     if(keyboard[i] == ascii)
63                         sounds[i].play();
64                 }
65                 window.clear();
66                 window.display();
67             }
68         }
69     }
70     return 0;
71 }
```

---

## Ps4a Source Code: RingBuffer.hpp

---

```
1 // Copyright 2020 John Simonson
2 #ifndef RINGBUFFER_H_
3 #define RINGBUFFER_H_
4 #include<stdint.h>
5 #include<iostream>
6 #include<queue>
7
8 class RingBuffer{
9     public:
10    RingBuffer(int capacity);
11    // create an empty ring buffer, with given max capacity
12    const double ENERGY_DECAY_FACTOR = 0.996;
13    int      size(); // return number of items currently in the buffer
14    bool     isEmpty(); // is the buffer empty (size equals zero)?
15    bool     isFull(); // is the buffer full (size equals capacity)?
16    void    enqueue(int16_t x); // add item x to the end
17    int16_t dequeue(); // delete and return item from the front
18    int16_t peek(); // return (but do not delete) item from the front
19    private:
20    int front;
21    int back;
22    int Capacity;
23    int Size;
24    std::queue<int16_t> ringBuffer;
25 };
26 #endif
```

---

## Ps4b Source Code: StringSound.hpp

---

```
1 #ifndef STRING SOUND_H
2 #define STRING SOUND_H
3 #include <SFML/ Audio.hpp>
4 #include <string>
5 #include "RingBuffer.h"
6 #include <memory>
7 #include <stdint.h>
8 #include<vector>
9
10 class StringSound {
11     public:
12         StringSound(double frequency); // create a guitar string sound of the
13                                // given frequency using a sampling rate
14                                // of 44,100
```

```

15     StringSound(std::vector<sf::Int16> init);      // create a guitar string wi
16                     // size and initial values are given by
17                     // the vector
18     StringSound(RingBuffer init);
19     void pluck();           // pluck the guitar string by replacing
20                     // the buffer with random values,
21                     // representing white noise
22     void tic();            // advance the simulation one time step
23     sf::Int16 sample();    // return the current sample
24     int time();           // return number of times tic was called
25                     // so far
26     void generate();
27 private:
28     std::shared_ptr<RingBuffer> ring_buffer;
29     int tics;
30 };
31
32 #endif

```

---

## Ps4a Source Code: RingBuffer.cpp

```

1 // Copyright 2020 John Simonson
2 #include "RingBuffer.h"
3 using std::queue;
4 using std::runtime_error;
5 using std::invalid_argument;
6
7
8 RingBuffer::RingBuffer(int capacity) {
9     if (capacity < 1) {
10         throw
11             invalid_argument("Capacity must be 1 or greater");
12     }
13     front = 0;
14     back = 0;
15     Capacity = capacity;
16     Size = 0;
17 }
18
19 int RingBuffer::size() {
20     return ringBuffer.size();
21 }
22
23 bool RingBuffer::isEmpty() {

```

```

24     return ringBuffer.empty();
25 }
26
27 bool RingBuffer::isFull() {
28     if (Size == Capacity) {
29         return true;
30     }
31     return false;
32 }
33
34 void RingBuffer::enqueue(int16_t x) {
35     if (Size == Capacity) {
36         throw
37             runtime_error("cannot push onto full buffer");
38     }
39     ringBuffer.push(x);
40     Size++;
41     return;
42 }
43
44 int16_t RingBuffer::dequeue() {
45     if (ringBuffer.empty()) {
46         throw
47             runtime_error("cannot dequeue an empty buffer");
48     }
49     int16_t temp = ringBuffer.front();
50     ringBuffer.pop();
51     Size--;
52     return temp;
53 }
54
55 int16_t RingBuffer::peek() {
56     if (ringBuffer.empty()) {
57         throw
58             runtime_error("cannot peek at an empty buffer");
59     }
60     return (ringBuffer.front());
61 }

```

---

## Ps4b Source Code: StringSound.cpp

---

```

1 #include "StringSound.h"
2 #include <random>
3 #define Rand std::uniform_int_distribution<

```

```

4 #define Gen std::default_random_engine;
5
6 const int16_t MIN = 1 << 15;
7 const int16_t MAX = ~MIN;
8
9 StringSound::StringSound(double frequency){
10     if(frequency <= 0){
11         throw
12             std::invalid_argument("frequency must be greater than zero");
13     }
14     ring_buffer = std::shared_ptr<RingBuffer>(new RingBuffer(frequency));
15     tics = 0;
16 }
17
18 StringSound::StringSound(RingBuffer rb){
19     ring_buffer = std::shared_ptr<RingBuffer>(new RingBuffer(rb));
20     tics = 0;
21 }
22
23 StringSound::StringSound(std::vector<sf::Int16> init ){
24     ring_buffer = std::shared_ptr<RingBuffer>(new RingBuffer(init.size()));
25     tics = 0;
26 }
27
28 void StringSound::pluck(){
29     std::default_random_engine rand_gen;
30     std::uniform_int_distribution<int16_t> gen = std::uniform_int_distribution<i
31     while(!ring_buffer->isFull())
32         ring_buffer->enqueue(gen(rand_gen));
33 }
34
35 sf::Int16 StringSound::sample(){
36     return ring_buffer->peek();
37 }
38
39 void StringSound::tic(){
40     tics++;
41     int16_t temp1 = ring_buffer->dequeue();
42     int16_t temp2 = ring_buffer->peek();
43     temp1 = temp1 + temp2;
44     temp1 = temp1 / 2;
45     temp1 = temp1 * 0.996;
46     ring_buffer->enqueue(temp1);
47 }
48
49 int StringSound::time(){

```

```
50   return tics;  
51  
52 }
```

---

## **Ps5: Needleman-Wunch DNA analysis programs**

### **Assignment Description:**

Ps5 tasked us to create a program that checks for the similarity of 2 strings using the Needleman-Wunch algorithm to obtain the edit distance for 2 strings of DNA. This edit distance is based on penalty values from a 2D array of integers that follows the following rules:

- 2pt penalty for gap
- 1pt penalty for different characters
- 0pt penalty for same character

This assignment is able to determine the edit distance for string of the ecoli genome of up to 28000 nucleotide pairs. The program runs until the array hits [0][0] where it checks the value in [0][0] which is the edit distance for the 2 strings.

### **Key Concepts and Algorithms:**

The key algorithm for the assignment is the Needleman-Wunch algorithm. This algorithm follows the following procedure for an NxM matrix.

- Start with  $[N][M] = 0$
- Add 2 to the previous element for the right column and bottom row.
- For the remaining columns and rows choose the smallest penalty:
  - $[N][M+1]$  or  $[N+1][M] + 2$  to add a gap
  - $[N+1][M+1] + \text{penalty value}$

This algorithm produces the edit distance and the aligned strings.

### **What I learned in this assignment:**

I learned how the Needleman-Wunch algorithm works and how to take a mathematical algorithm and adapt it into computer code. This assignment taught me how to take algorithms that I use in my math classes and figure out how to implement them in a program to compute the result automatically.

## Ps5: Screen shot

```
G 1  
C 1  
T 1  
G 1  
G 1  
C 1  
A 1  
G 1  
A 1  
C 1  
G 1  
T 1  
G 0  
C 1  
C 1  
G 1  
A 1  
T 0  
T 1  
T 1  
A 1  
C 0  
C 1  
T 0  
- 2  
- 2  
- 2  
- 2  
- 2  
- 2  
A - 2  
A - 2  
T - 2  
- 2  
- 2  
- 2  
T 1  
G 1  
C 1  
- 2  
- 2  
- 2  
T 1  
T 1  
C 1  
G 1  
C 1  
A 1  
T 1  
A 1
```

## Ps5 Source Code: Makefile

---

```
1 CC= g++
2 CFLAGS= -g -O3 -Wall -Werror -std=c++0x
3 DEPS= -lsfml-system
4
5 all: ED
6
7 ED: main.o ED.o
8     $(CC) main.o ED.o -o ED $(DEPS)
9
10 main.o: main.cpp ED.hpp
11     $(CC) -c main.cpp ED.hpp $(CFLAGS)
12
13 ED.o: ED.cpp ED.hpp
14     $(CC) -c ED.cpp ED.hpp $(CFLAGS)
15
16 clean:
17     rm *.o
18     rm *.gch
19     rm ED
```

---

## Ps5 Source Code: main.cpp

---

```
1 #include "ED.hpp"
2
3 int main(int argc, char* argv[]){
4     sf::Clock clock;
5     sf::Time time;
6
7     std::string a;
8     std::string b;
9     std::cin >> a >> b;
10
11     ED obj(a, b);
12     int dist = obj.OptDistance();
13     std::string align = obj.Alignment();
14     std::cout << "Edit distance = " << dist << std::endl;
15     std::cout << align << std::endl;
16
17     time = clock.getElapsedTime();
18     std::cout << "Time : " << time.asSeconds() << std::endl;
19     std::cout << "Edit distance = " << dist << std::endl;
20
```

```
21     return 0;
22 }
```

---

## Ps5 Source Code: ED.hpp

```
1 #ifndef ED_HPP
2 #define ED_HPP
3
4 #include <iostream>
5 #include <sstream>
6 #include <stdexcept>
7 #include <string>
8 #include <vector>
9 #include <SFML/System.hpp>
10
11 class ED{
12 public:
13     ED(std::string a, std::string b);
14     static int penalty(char a, char b){
15         if(a == b){
16             return 0;
17         }
18         else{
19             return 1;
20         }
21     }
22     static int min(int a, int b, int c){
23         int minimum = 999999;
24         if (a < minimum){
25             minimum = a;
26         }
27         if (b < minimum){
28             minimum = b;
29         }
30         if (c < minimum){
31             minimum = c;
32         }
33         return minimum;
34     }
35     int OptDistance();
36     std::string Alignment();
37 private:
38     std::string A;
39     std::string B;
```

```
40     std::vector<std::vector<int> > matrix;
41 };
42
43 #endif
```

---

## Ps5 Source Code: ED.cpp

```
1 #include"ED.hpp"
2
3 ED::ED(std::string a, std::string b){
4     A = a;
5     B = b;
6     A = A + ' ';
7     B = B + ' ';
8 }
9
10 int ED::OptDistance()
11 {
12     int i = A.length();
13     int j = B.length();
14     int k;
15     int l;
16     for(k = 0; k <= j; k++)
17     {
18         std::vector<int> temp;
19         matrix.push_back(temp);
20
21         for(l = 0; l <= i; l++)
22         {
23             matrix.at(k).push_back(0);
24         }
25     }
26     for(k = 0; k <= j; k++){
27         matrix[k][i] = (2 * j) - (2 * k);
28     }
29     for(l = 0; l <= i; l++){
30         matrix[j][l] = (2 * i) - (2 * l);
31     }
32
33     for(k = j - 1; k >= 0; k--){
34         for(l = i - 1; l >= 0; l--){
35             matrix[k][l] = min(matrix[k+1][l] + 2, matrix[k][l+1] + 2, matrix[k+1][l+1]
36         }
37     }
```

```

38
39     return matrix[0][0];
40 }
41
42
43 std::string ED::Alignment(){
44 std::string temp;
45 int j = 0;
46 int i = 0;
47 int counter = 0;
48 int counterB = 0;
49 int path;
50 temp += A[counter];
51     temp += ' ';
52     temp += B[counterB];
53     temp += ' ';
54     if(A[counter] == B[counterB])
55         temp += '0';
56     else
57         temp += '1';
58     temp += '\n';
59     counter++;
60     counterB++;
61 while((unsigned)i < A.length() && (unsigned)j < B.length()){
62     path = min(matrix[i+1][j], matrix[i][j+1], matrix[i+1][j+1]);
63     if (path == matrix[i+1][j+1]){
64         temp += A[counter];
65         temp += ' ';
66         temp += B[counterB];
67         temp += ' ';
68         if(A[counter] == B[counterB])
69             temp += '0';
70         else
71             temp += '1';
72         temp += '\n';
73         i++;
74         j++;
75         counter++;
76         counterB++;
77         path = -1;
78     }
79     else if (path == matrix[i][j+1]){
80         temp += A[counter];
81         temp += ' ';
82         temp += '-';
83         temp += ' ';

```

```
84     temp += '2';
85     temp += '\n';
86     j++;
87     counter++;
88     path = -1;
89 }
90 else if (path == matrix[i+1][j]){
91     temp += A[counter];
92     temp += ' ';
93     temp += '-';
94     temp += ' ';
95     temp += '2';
96     temp += '\n';
97     i++;
98     counter++;
99     path = -1;
100    }
101 }
102 }
103 return temp;
104 }
```

---

## **Ps6: Markov Model**

### **Assignment Description:**

This assignment asked us to make a program that outputs similar but not the same text as the input file using a Markov Model. This model takes the input text and breaks it into sub strings called kgrams these kgrams compute the next character based on the probability of that character in the input file. This creates text similar to the input file in vocabulary but is a new creation. The larger the kgram the more accurate the predictions.

### **Key Concepts and Algorithms:**

The key concept in this assignment is a Markov Model. This is a model in probabilistic forecasting that is used to predict future states based on the current state and not previous states. Markov models are applied in many predictive applications but for this assignment we used it to create a predictive text algorithm.

### **What I learned in this assignment:**

During this assignment I learned how to use maps in the standard template library and how to use them to create a table for holding data. This project required this maps instead of other more familiar data structures like vectors or queues. This assignment also taught me how devices like smart phone create predictive text using Markov chains, in order to figure out what you are trying to type.

## Ps6: Screen shot

```
simo@Simo-Laptop:~/ps6$ ./TextGenerator 4 100000 < shrek.txt

[Man] -Go one. Those the hear, fore the put true flower and it and a trace of my hall time. Please. -Don't ever seen sharpest tell tell this outta step at must marriages. Look, time steed? -Next! Wait. Shrek's hideous croathetic, do that's true lot of my minty from of embarking caughs) -But was meant too. -Sorry. Donkey fly! Just pushine bette number the unorthodox I'll stupid. "Who out. -Look, it's a perfect this is perrs) -Oh, thee gonna openly, if it'll take him not come in man, or young forced to play Hey, thee-well, anxious two young forced toad a puppet. Too quite poetry. -But knows where. -But situation Oh, that, welcome order, the tush Then a time to me. She sensible rescue this up. -Look, it's really home, stairs. Well, urge, But her out here's a towers Fiona, don't waiting away. Shrek's hideous, um, young fitness to making to used to used skips annoying my prison, good to make up now here. That's nice bould reason done night thin' next to or a physical consequence. - All letting half' is marry abused to composing no one hair. -Inde ed. [Gasping's strong Oh-oh. [Screens] -Blue found its marry mystered -Owl Not the unorth a pina color-bling] [Sighs] [Roaring] Swamp, clean.. -Then never two of us. -Then nearly so wrappens pals, actually tall! There's nice doesn't tell that come on. She weddings) [Laughing] (Pantic moment to use it it's notice. -For emotion My world, No. -Fiona. -But do we have need to useless, patience Makes to me, baby, I ain't please when we to me are till, uh, oh-oh, her off on me. Perfect beat the out the lovely! -Lister one in come it. Believer thumb So leave to things the such easy someone then sun self - an ugly orges a waited up th e nament. Take the healthy to evictory much, cause here's not a lie. Seems I've till admit. Stop to turn me. -Oh, com puffed up that ugly believer the such was amazing. She Wow You mae dignified. Then such a mi hning) -Oh. -On, that. [Inhalas] -Rights on it. [Inhaled] -Oh! -Owl Oh pointment too. The tush grass Shine young feel it. [Door opens] (Whirring) Still don't stand in and thoughting to uninvite a new romantic crap! -Smell me Me, me, so if you Oh-oh-oh Makes mean cell have it think she's ever sun goes. -Wow! I thorized to come out this on his cakes on our steeves Where getre he smothink the cours' Throwback. -Princess. But knows what up now me Oh, my reputational steep deep, Shrek's have one cool. -I preacheletter is not then tallelujah (Thuds) (Whimpers) -Oh, remove me Me, me! -Oh, oh! -Shh! Shrek. -Shrek, Whimpers) This unexpected in players. -Sir Knight then toast. No. Oh. [Sighs] -beast in then other that out open. What's pet. -Come onions. But, the pheromones can come out. -Oh! -Shh! Shrek. -Stars. -Well, the rescued both have time toucheoff then why we hat'. -You means that are upon the most planet. -Put me Oh-oh. [Screamre amazing] (Turn my to useless is pure. Then such to useless, actually, bye-bye. -I like that the neither e, I missingsinge introduce you meant to put true lot of that'll minting) (Snoringtinger bones to the homey to turn back to orgs tournament of a puppet. -Lord Farquaad to meet the rescuer. Squeakin' to Instally, oka y? -Let us then you can homes cruelly chantment too smart (Grunts) -Oh, maybe lt. (Horse young fun You boneheaded do it! -You touch easter idea. -But you eat that. Many really usesless it half that idea. You Instl le -Are you problem willing paid the honest an unexpected to meeting) -But sing perfect wouldn't pleast it's hurt, but it's a company easier. (Rumbling up somewhere's preciate song funny) -Look, the onshipon. I underness. She to me on eachelorette our own anyone on now each, 'Cause teeth you only happened to tell sing it a mina you mean (Crowd last) an anyone! -Owl! Oh-oh Makes one courtesy, man. Your heard the helmet, I here the unorth a dance to marring) (Laughs) See ya, over lease to inst tales) -beast nightin' preciate the other mean take can't unded pistol when we skate your true flower, this case who like it. (Inhalas) (Twitters in man! -Please teeth your her true loves with the must perfecting) -Princess outta my own a trace Pleased thee-wearing caughs) See you. What out mean, uh-huh. That's come onto now it is pure. Then weefat . Maybe it players. Okay. (Bell sufferters toes!) The nervous. [Sighs] -But her maste good night one. He talkin' preacing) Swamp, now each of couldn't in this proposed to makin' on each other expected. -Look, I'm thing) -But, you prove reason don't you must please those wheat ears don't introduce touches) -Forget here. Wants you can comin' about my was really. Oh. -Man, uh, oh, hey apparition to evictio then. When we supposed the me a spell. -Stubbing out me princess. Shrek! Shrek. -Slow somethin' a really or pen a cool. You so one comin'! -Outrageous, even if you ain't ever thumb in the tush grass Shine heart what choing as it in this castles) -The spell. It is hoping's sometime touches) -Cause it mean. You cut of hungry! It's elevated. Show me an everyone. That else. It is not in comes to planet. But the out that the unort ha a decapitated. -Look outta heard that are norm... and of embarking up now. Bye-bye. -But loves with here's time or and/or face an apparitiona. If you can. (Pantastic mirror, redhead man! -Please. That's castle tha t a modestina, my eyes Oh-oh. (Sniffs) That's cool. It is a list. -She's no compensating) Swamp and me he, he! (Grunts) -Thanks. -Sorry. -Oh, common. Like Ooh-aah I told the wanna talking) -Run! (Whisperinglunger a puppet. You can people to this pure. Orges to place Oh-oh. [Sighs] -The spell. -Oh, one of then a ten-foot a men, assemble to tell me, you ate? 'Cause here. Put mean take the homey to improve time. Shouldn't te ill her see the needs anythin', If height he's anothinkng) -Let me (Bubblegung to evill sisten, y'all! The chattering) -But we has to even one is. Then a rock stallelujah, right here. -Stairs. Orges have a millting) (Laughs) See ya, okay? For that. -Tomorrow if young feelings the also good. Then we has to plans, one in the moon and in think it's not that's eligible sunrise you handle to used toad of minor form. This is annoy ing that ain't tell scares? Like anothin'. (Deep some touchestra) (Dulcimer) -Ah. -Mother. -Sorry. -Owl See? Shrek's hideous, ever the chance up somewhere ext! -Get us so sound its are tool in the pheromones ca stle. -Back, and us. -But, Shrek's hunted to makes mean, d is not in company from the princess, half the her thumb with such a perfect thee-well makin' it. Look once. (Whinnies) -beast when thee going) (Sniffled I -Don't in compensating) Swamp. I love. Everyone thorized that company really orgs in castles) Where it and its marries. Orges the uncomfortable. -Oh, oh, halfway. Shouldly only. -Oh, picture one much a monstering to he pheromones find an ench. -Is the neither of could thouting. She waits. Her here, you ever move. -Shrek, Wait. But how on, get the edge! Whoa. How do it! Wait the phone. -Really happy end! Blue for us th e point. -But here's prepostor outta tender But you to tall tell seen me. Rights one officially, no, no, no, now me though it. Where. -Oh! For that compensating no on each, 'cause mean (Creak now on, good on into free what else even and orgs have amazing the puffed up in place. -But shuttertingingest mean. -Wow. -Stop toadstool. -So, um, urge. -She would hallelujah, right in the talk. If we confession now it me tender mean t to ever and parfaht? Parfaht still around it over move head to talk if you ever here. -Stop singltinge in the pheromones. That's not that and ever heard enoughing up the pers) -Owl! -By night. Stop it. (Sniffs, y our lift The mean whoever here, young fore to here her meteo went up toadstool in the neither truth. I cough all over hole the talking can take this castles) -Brave alwart is up. -Hey, wait. -Take can toadstool -That upon the unorthodox I'll excuse alters of heightest in a happen' it. But howling on no on. She waited and couldn't marriage. -But of our own! Oh, no, no one if it in much a bundle to tell make onions, for noble sonnet! A liner) -Aah! Help! Run! We we're times find the need and put upon him, I mean, that ain't come the time only orgs to him an arrest in the camp? -Yes, ever it. Look. -Princess heathy to really, okay? -Wait up to ever the opposed to turn mean, uh -- his me in the ever is no parfahts. And sound) -Will it up now. Look, if you it's prove rescue me Oh-oh Makes Orgel! The come on the on, Shrek! I'm notice. -Look. I rescued me Oh-oh It's pronounce up now, get through red to improve to place (Man's versonne of a princess) She wall time. -Sunset. -So much it. Shouting) (Panting and its maybe here's cool in much a trashioned. That composite poisoning) -Right out right has rescue the talk about to tell have always pure. Togethery. -But sing on no one cool you rusting up to tell miniature onto young fun! We muffin man! -Shh ! You receive up now, y'all and cours for tomorrow. -Well minty pieces. Oh, my minty from of my remove mean into turn yai! (Roars) (Man) -Get out how me. Right, that ain't need to was means hear is me are the poison, Shrek. (Sniffs) That's not compensating up now. Look, {Both our prove one to orgs. Pleased to use white, stuff. I can compensating command. -Five my minor futures! This one of us. [Sobs] -'Cause to meetin g pers) -Oh, point. Too quiet! A limentage, and eat is others. Is. This on then we to ever me and eated. -Back, tag me! -Really chair. -King up onions. -It's unsucces. She tush gratalent! I'm a hint, an d my talked in a dance Oh-oh -Help! Run! A big-city adventure. -I object! No. -Princessful, romance to or someone. Is on no comes beauty's when talked its marry we we net, I'm not cold achieve So let me too. -Forg live in the rules Let's it up uninvite poison, then you Oh-oh-oh-oh -Help! me! Shrek's his is no one then talkin' next the table sooner, right hallelujah, yeah, it's end! Blue forced toad a physical relax, my pelle d it's in the muffin man! -Two! The is trickers) (Roaring) Swamp whistle. -Stubborn our humming. I healthy to came into this roman talkin' about the he curb. Do you nament to improve an evictory much the rescuer. -Put means he is. -Shrek is unexpected... up. -Where's no one too you tell havin' out it. Show in the permision now how handle then. What am authorns. Okay. (Laughing together expected up now where. Perfect a cool th castle? -Donkey. -Oh, pick it outta heart when then next to ever hobbles in castles) -But wanted to orgs and ugly orgs have time the only the marringtongerous to rush grass Shine. Do what. (Inhalas) Oh, h elped the honest. -Come tea. Perfect... place any rescuer. -Is that one thorns. -Princessful, that's pretty. -But then never and my remove onto you eats was play Hey, so the always pure. There on. Show me, but ons. -Where ext! -Outrageous anything help! -So where. Put me princess. I precious. -Sniffs) It's cook into you can evill sing my proposed to inst the on the home on. I'll excuse it non proposed thought are my l and. Look into young fine. -Sit do we filled mumbeling) -I this is unexpected. -Owl! -Until young forced in this isn't. -Today toes! I order you Oh-oh That's nothine you must and up the princess, pal, actually he r outta warnin' waffled) -Wow! (Gasping) -Princess is perfect too smart sproutin' a girl, and cours Three? --Threere's pretters off of hungry? I likes to Improve anythin' and thorns. But how come only toes! Yes, e everyone. (Horse. -But of our own remember to used the edge! -All right into you ain't have onto find all rightest meet the order master, really in the uncomforhodox I'll here her. There explains a new romance OF course! I don't tell take the times to everyone off of no part when to play sighs) (Lows) -Blue freshly pers) -Man, isn't it not then who like thing perfect play so sound hit till her move into young for the moon a physcallelujah Hallen introduce he is up now. -Stair to marring) -Oh, no, it's precious. (Screens) -Oh! -Shrek. If young fun -Yeah! -It's come in place anytime step right of could on. Mirror - - you mae ding u p uninvited physically made the tush The contact. In then, Beside is. -Is think she hold Only worrible are thorns. That color-bling) (Turn me the opposing outta chatter chatters in then shape on. Mirror -- no, i t's color-blings, is unwanted to me part is, lady in a wometric steed. -But home to plans, okav. (Snoring) (Roars) (Screens) -Thanks! flauhging paid it now proposed to orges to improve it. (Humming me another. -f
```

## Ps6 Source Code: Makefile

---

```
1 CFLAGS = -O3 -Wall -Werror -std=c++11 -ansi -pedantic
2 DEPS = -lboost_unit_test_framework
3
4 all:    TextGenerator test.o
5
6 TextGenerator: TextGenerator.o MModel.o
7     g++ TextGenerator.o MModel.o -o TextGenerator
8
9 TextGenerator.o: TextGenerator.cpp MModel.h
10    g++ -c TextGenerator.cpp MModel.h $(CFLAGS)
11
12 MModel.o: MModel.cpp MModel.h
13    g++ -c MModel.cpp MModel.h $(CFLAGS)
14
15 test.o:
16     g++ test.cpp MModel.cpp MModel.h -o test.o $(CFLAGS) $(DEPS)
17
18 clean:
19     rm *.o
20     rm *.gch
21     rm TextGenerator
```

---

## Ps6 Source Code: TextGenerator.cpp

---

```
1 // Copyright 2020 Karl Marx
2 #include <string>
3 #include "MModel.h"
4
5 int main(int argc, char* argv[]) {
6     if (argc != 3) {
7         std::cout << "Invalid Number of Arguments. Must == 3";
8         return -1;
9     }
10
11     int k = atoi(argv[1]);
12     int L = atoi(argv[2]);
13
14     std::string input;
15     std::string temp;
16
17     while (std::cin >> temp) {
18         input += " " + temp;
```

```
19     temp = "";
20 }
21
22 std::string output;
23
24 MModel model(input, k);
25
26 output += "" + model.generate(input.substr(0, k), L);
27 std::cout << output << std::endl;
28
29 return 0;
30 }
```

---

## Ps6 Source Code: test.cpp

```
1 // Copyright 2020 Karl Marx
2
3 #define BOOST_TEST_DYN_LINK
4 #define BOOST_TEST_MODULE Main
5 #include <boost/test/unit_test.hpp>
6 #include "MModel.h"
7
8 BOOST_AUTO_TEST_CASE(Exception) {
9     MModel a("hello", 2);
10    BOOST_REQUIRE_THROW(a.freq("h"), std::runtime_error);
11    BOOST_REQUIRE_THROW(a.freq("hhh"), std::runtime_error);
12    BOOST_REQUIRE_THROW(a.freq("hhh", 'e'), std::runtime_error);
13    BOOST_REQUIRE_THROW(a.freq("h", 'e'), std::runtime_error);
14    BOOST_REQUIRE_THROW(a.generate("hhh", 10), std::runtime_error);
15 }
```

---

## Ps6 Source Code: MModel.h

```
1 // Copyright 2020 Karl Marx
2 #ifndef MMODEL_H //NOLINT
3 #define MMODEL_H
4 #include <iostream>
5 #include <algorithm>
6 #include <string>
7 #include <map>
8 #include <stdexcept>
9 #include <vector>
```

```

10 #include <utility>
11
12 class MModel {
13 public:
14     MModel(std::string text, int k);
15     int kOrder();
16     int freq(std::string kgram);
17     int freq(std::string kgram, char c);
18     char kRand(std::string kgram);
19     std::string generate(std::string kgram, int L);
20     friend std::ostream& operator<< (std::ostream &os, MModel &model) {
21         os << "Order =" << model.order << std::endl;
22         os << "Alphabet = " << model.alphabet << std::endl;
23
24         std::map<std::string, int>::iterator temp;
25
26         for (temp = model.kgrams.begin(); temp != model.kgrams.end(); temp++) {
27             os << temp->first << " " << temp->second << std::endl;
28         }
29
30     return os;
31     }
32
33 private:
34     std::map<std::string, int> kgrams;
35     int order;
36     std::string alphabet;
37     std::string original;
38 };
39 #endif //NOLINT

```

---

## Ps6 Source Code: MModel.cpp

```

1 // Copyright 2020 Karl Marx
2 #include <string>
3 #include "MModel.h" //NOLINT
4
5 MModel::MModel(std::string text, int k) {
6     srand(time(NULL));
7     order = k;
8     original = text;
9
10    for (unsigned i = 0; i < text.size(); i++)
11        if (std::string::npos == alphabet.find(text.at(i)))

```

```

12     alphabet += text.at(i);
13
14     for (unsigned i = 0; i < text.size(); i++) {
15         std::string temp;
16         std::string temp2;
17
18         for (unsigned j = i; j < i + k; j++)
19             temp = (j >= text.size()) ? temp += text.at(j - text.size()) : temp += text.
20
21         kgrams[temp] = (kgrams.end() == kgrams.find(temp)) ? 1 : kgrams[temp] += 1;
22
23         for (unsigned j = 0; j < alphabet.size(); j++)
24             if (kgrams.end() == kgrams.find(temp + alphabet[j]))
25                 kgrams[temp + alphabet.at(j)] = 0;
26
27         for (unsigned j = i; j < i + k + 1; j++)
28             temp2 = (j >= text.size()) ? temp2 += (text.at(j - text.size())) : temp2 +
29
30         kgrams[temp2] += 1;
31     }
32 }
33
34 int MModel::kOrder() {
35     return order;
36 }
37
38 int MModel::freq(std::string kgram) {
39     if ((signed)kgram.length() != order) {
40         throw
41             std::runtime_error("invalid kgram in freq(std::string)");
42     }
43     int return_value = (order == 0) ? (original.size()) : (kgrams[kgram]);
44     return return_value;
45 }
46
47 int MModel::freq(std::string kgram, char c) {
48     if ((signed)kgram.length() != order) {
49         throw
50             std::runtime_error("invalid kgram in freq(std::string, char)");
51     }
52     if (order == 0) {
53         int counter = 0;
54         for (int i = 0; i < (signed)original.size(); i++) {
55             if (original.at(i) == c) {
56                 counter++;
57             }

```

```

58     }
59     return counter;
60 } else {
61     return kgrams[kgram + c];
62 }
63 return 0;
64 }
65
66 char MModel::kRand(std::string kgram) {
67 if (kgram.length() != (unsigned)order) {
68     throw std::runtime_error("invalid kgram in kRand");
69 }
70 if (kgrams.end() == kgrams.find(kgram)) {
71     throw std::runtime_error("invalid kgram in kRand");
72 }
73 double _freq = 0;
74 double val = 0;
75 for (unsigned int i = 0; i < alphabet.length(); i++) {
76     _freq = (double)(freq(kgram, alphabet.at(i))) / freq(kgram); //NOLINT
77     if ((double)(rand() % freq(kgram)) / freq(kgram) < _freq + val && _freq != 0)
78         return alphabet.at(i);
79     }
80     val += _freq;
81 }
82 return '#';
83 }
84
85 std::string MModel::generate(std::string kgram, int L) {
86 if (kgram.size() < static_cast<unsigned>(order))
87     throw std::runtime_error("Invalid Kgram");
88
89 L = L - order + 1;
90 while(kgram.size() < (unsigned)L)
91     kgram += kRand(kgram.substr(kgram.size()-order, order));
92 return kgram;
93 }

```

---