```
};

#include "stdafx.h"
#include <stdio.h>
#include <iostream>
#include <math.h>
#include <fstream>
#include "SFML\Graphics.hpp"
#include "SFML\System.hpp"
#include "Textury.h"

using namespace std;
using namespace sf;

RenderWindow window(VideoMode(600, 620), "Wahadlo Matematyczne");

class Pendulum
{
public:
	Pendulum()
	{
		PenText.loadFromFile("Data/Pendulum.png");
		PenSprite.setTexture(PenText);
		PenSprite.setPosition(300, 10);
		PenSprite.setOrigin(225.5, 0);
		PenSprite.setScale(0.1, 0.1);
	};
	~Pendulum() {};
	Texture PenText;
	Sprite PenSprite;

	float Mass = 1;

	float Length = 1;

	float Alpha = 0;

	float Force = 0;

	float Acceleration = 0;

	float RotAcceleration = 0;

	float RotVelocity = 0;

	float Velocity = 0;

	float Gravity = 9.80665;

};


void Drawing(Pendulum &NewPendulum);

string Insertingf();

string Insertingf2();

void SetBeginValues();

void SetDeg(Pendulum &NewPendulum);

void SetLength(Pendulum &NewPendulum);

void SetMass(Pendulum &NewPendulum);

void StartSim(Pendulum &NewPendulum);

void SetFileName(Pendulum &NewPendulum);

void SetGravity(Pendulum &NewPendulum);


void main()

{

	Pendulum NewPendulum;

	font.loadFromFile("Data/F2.ttf");
	SetBeginValues();

	InformationText.setString("Aby rozpoczac symulacje wybierz 'Uruchom symulacje'.\nAby zmienic mase wybierz 'Zmien mase'.\nAby zmienic dlugosc nici wybierz 'Zmien dlugosc'.\nAby zmienic kat nachylenia nici wybierz 'Zmien kat'.\nAby zakonczyc wcisnij escape.");

		do

		{

			Vector2f Mouse(Mouse::getPosition(window));



			if (GraphsText.getGlobalBounds().contains(Mouse))

				GraphsText.setTexture(GraphsTexture2);

			else

				GraphsText.setTexture(GraphsTexture);



			if (StartSimText.getGlobalBounds().contains(Mouse))

				StartSimText.setTexture(StartSimTexture2);

			else

				StartSimText.setTexture(StartSimTexture);



			if (SetAngleText.getGlobalBounds().contains(Mouse))

				SetAngleText.setTexture(AngleTexture2);

			else

				SetAngleText.setTexture(AngleTexture);



			if (SetMassText.getGlobalBounds().contains(Mouse))

				SetMassText.setTexture(MassTexture2);

			else

				SetMassText.setTexture(MassTexture);
```

```
if (SetLengthText.getGlobalBounds().contains(Mouse))

            SetLengthText.setTexture(LengthTexture2);

else

            SetLengthText.setTexture(LengthTexture);


if (PomocText.getGlobalBounds().contains(Mouse))

            PomocText.setTexture(PomocTexture2);

else

            PomocText.setTexture(PomocTexture);


if (NazwaSText.getGlobalBounds().contains(Mouse))

            NazwaSText.setTexture(NazwaTexture2);

else

            NazwaSText.setTexture(NazwaTexture);


if (ZapiszText.getGlobalBounds().contains(Mouse))

            ZapiszText.setTexture(ZapiszTexture2);

else

            ZapiszText.setTexture(ZapiszTexture);


if (WczytajText.getGlobalBounds().contains(Mouse))

            WczytajText.setTexture(WczytajTexture2);

else

            WczytajText.setTexture(WczytajTexture);


if (SetGravityText.getGlobalBounds().contains(Mouse))
```

```
                SetGravityText.setTexture(GravityTexture2);
            else
                SetGravityText.setTexture(GravityTexture);


            AngleText.setString("Kat [deg]: " + to_string(NewPendulum.Alpha));
            LengthText.setString("Dlugosc [m]: " + to_string(NewPendulum.Length));
            MassText.setString("Masa [kg]: " + to_string(NewPendulum.Mass));
            GravityText.setString("Przysp. grawitacyjne [m/s]: " +
to_string(NewPendulum.Gravity));


            while (window.pollEvent(WhatYaDoin))
            {
                if (WhatYaDoin.type == Event::MouseButtonReleased &&
WhatYaDoin.key.code == Mouse::Left && SetAngleText.getGlobalBounds().contains(Mouse))
                {
                    SetDeg(NewPendulum);
                    InformationText.setString("Aby rozpoczac symulacje wybierz
'Uruchom symulacje'.\nAby zmienic mase wybierz 'Zmien mase'.\nAby zmienic dlugosc nici wybierz
'Zmien dlugosc'.\nAby zmienic kat nachylenia nici wybierz 'Zmien kat'.\nAby zakonczyc wcisnij
escape.");
                }
                if (WhatYaDoin.type == Event::MouseButtonReleased &&
WhatYaDoin.key.code == Mouse::Left && SetLengthText.getGlobalBounds().contains(Mouse))
                {
                    SetLength(NewPendulum);
                    InformationText.setString("Aby rozpoczac symulacje wybierz
'Uruchom symulacje'.\nAby zmienic mase wybierz 'Zmien mase'.\nAby zmienic dlugosc nici wybierz
'Zmien dlugosc'.\nAby zmienic kat nachylenia nici wybierz 'Zmien kat'.\nAby zakonczyc wcisnij
escape.");
                }
                if (WhatYaDoin.type == Event::MouseButtonReleased && WhatYaDoin.key.code == Mouse::Left && SetGravityText.getGlobalBounds().contains(Mouse))
                {
                    SetGravity(NewPendulum);
                    InformationText.setString("Aby rozpoczac symulacje wybierz 'Uruchom symulacje'.\nAby zmienic mase wybierz 'Zmien mase'.\nAby zmienic dlugosc nici wybierz 'Zmien dlugosc'.\nAby zmienic kat nachylenia nici wybierz 'Zmien kat'.\nAby zakonczyc wcisnij escape.");
                }
                if (WhatYaDoin.type == Event::MouseButtonReleased && WhatYaDoin.key.code == Mouse::Left && SetMassText.getGlobalBounds().contains(Mouse))
                {
                    SetMass(NewPendulum);
                    InformationText.setString("Aby rozpoczac symulacje wybierz 'Uruchom symulacje'.\nAby zmienic mase wybierz 'Zmien mase'.\nAby zmienic dlugosc nici wybierz 'Zmien dlugosc'.\nAby zmienic kat nachylenia nici wybierz 'Zmien kat'.\nAby zakonczyc wcisnij escape.");
                }
                if (WhatYaDoin.type == Event::MouseButtonReleased && WhatYaDoin.key.code == Mouse::Left && StartSimText.getGlobalBounds().contains(Mouse))
                {
                    StartSim(NewPendulum);
                    InformationText.setString("Aby rozpoczac symulacje wybierz 'Uruchom symulacje'.\nAby zmienic mase wybierz 'Zmien mase'.\nAby zmienic dlugosc nici wybierz 'Zmien dlugosc'.\nAby zmienic kat nachylenia nici wybierz 'Zmien kat'.\nAby zakonczyc wcisnij escape.");
                }
                if (WhatYaDoin.type == Event::MouseButtonReleased && WhatYaDoin.key.code == Mouse::Left && NazwaSText.getGlobalBounds().contains(Mouse))
                {
                    SetFileName(NewPendulum);
                    InformationText.setString("Aby rozpoczac symulacje wybierz 'Uruchom symulacje'.\nAby zmienic mase wybierz 'Zmien mase'.\nAby zmienic dlugosc nici wybierz
'Zmien dlugosc'.\nAby zmienic kat nachylenia nici wybierz 'Zmien kat'.\nAby zakonczyc wcisnij
escape.");
                }
                if (WhatYaDoin.type == Event::MouseButtonReleased &&
WhatYaDoin.key.code == Mouse::Left && PomocText.getGlobalBounds().contains(Mouse))
                {
                    Event closewd;
                    do
                    {
                        window.clear();
                        window.draw(Background);
                        Text Help;
                        Help.setFont(font);
                        Help.setCharacterSize(15);
                        Help.setPosition(5, 5);
                        Help.setColor(Color::Yellow);
                        Help.setString("Witaj w instrukcji obslugi
programu!\n1) Podstawowe dzialania:\nAby wprowadzic dlugosc nici, mase ciala lub kat odchylenia
wcisnij odpowiedni przycisk,\nwprowadz z klawiatury nienumerycznej liczbe i zatwierdz ENTERem.
Po wpisaniu wartosci \nuruchom symulacje. Aby zatrzymac symulacje wcisnij ENTER. Po
przeprowadzeniu\nsymulacji mozesz odworzyc wykresy odpowiednim przyciskiem*. Wybierz
\"Nazwa pliku\"\n aby wprowadzic nazwe pliku z ktorego chcesz wczytac lub do ktorego chcesz
zapisac dane.\n\n*aby wyswietlic wykresy wymagane jest zainstalowanie Gnuplota.\n\n2)Zakres
dzialania:\nProgram dziala dla katow od -90 do 90 stopni oraz dlugosci nici i masy\n mieszczacych sie
w zmiennej typu float\n\n\nNACISNIJ ENTER ABY WROCIC");
                        window.draw(Help);
                        window.display();
                        while (window.pollEvent(closewd))
                        {
                            if (closewd.type == Event::Closed)
                                window.close();
                        }
                    } while (!Keyboard::isKeyPressed(Keyboard::Return));
                    InformationText.setString("Aby rozpoczac symulacje wybierz 'Uruchom symulacje'.\nAby zmienic mase wybierz 'Zmien mase'.\nAby zmienic dlugosc nici wybierz 'Zmien dlugosc'.\nAby zmienic kat nachylenia nici wybierz 'Zmien kat'.\nAby zakonczyc wcisnij escape.");
                }
                if (WhatYaDoin.type == Event::Closed || WhatYaDoin.type == Event::KeyPressed && WhatYaDoin.key.code == Keyboard::Escape)
                {
                    window.close();
                }
                if (WhatYaDoin.type == Event::MouseButtonReleased && WhatYaDoin.key.code == Mouse::Left && GraphsText.getGlobalBounds().contains(Mouse))
                {
                    Event Wykwyk;
                    do
                    {
                        window.clear();
                        window.draw(Background);
                        Text Grafy;
                        Grafy.setFont(font);
                        Grafy.setCharacterSize(15);
                        Grafy.setPosition(5, 5);
                        Grafy.setColor(Color::Yellow);
                        Grafy.setString("Wybierz ktory wykres chcesz zobaczyc odpowiednim klawiszem:\n\n1 - Przyspieszenie liniowe w funkcji czasu\n\n2 - Predkosc liniowa w funkcji czasu\n\n3 - Predkosc katowa w funkcji czasu\n\n4 - Przyspieszenie katowe w funkcji czasu\n\n5 - Kat w funkcji czasu\n\n6 - Sila napedzajaca wahadlo w funkcji czasu\n\nENTER - powrot");
                        window.draw(Grafy);
                        window.display();
                        while (window.pollEvent(Wykwyk))
                        {
                            if (Wykwyk.type == Event::KeyReleased && Wykwyk.key.code == Keyboard::Num1)
                                system("start wgnuplot -p -e \"plot 'GraphsData/PrzyspieszenieLiniowe.txt' w lines  \"");
                            if (Wykwyk.type == Event::KeyReleased && Wykwyk.key.code == Keyboard::Num2)
                                system("start wgnuplot -p -e \"plot 'GraphsData/PredkoscLiniowa.txt' w lines  \"");
                            if (Wykwyk.type == Event::KeyReleased && Wykwyk.key.code == Keyboard::Num3)
                                system("start wgnuplot -p -e \"plot 'GraphsData/PredkoscKatowa.txt' w lines  \"");
                            if (Wykwyk.type == Event::KeyReleased && Wykwyk.key.code == Keyboard::Num4)
                                system("start wgnuplot -p -e \"plot 'GraphsData/PrzyspieszenieKatowe.txt' w lines  \"");
                            if (Wykwyk.type == Event::KeyReleased && Wykwyk.key.code == Keyboard::Num5)
                                system("start wgnuplot -p -e \"plot 'GraphsData/Kat.txt' w lines  \"");
                            if (Wykwyk.type == Event::KeyReleased && Wykwyk.key.code == Keyboard::Num6)
                                system("start wgnuplot -p -e \"plot 'GraphsData/SilaNapedzajaca.txt' w lines  \"");
                            if (Wykwyk.type == Event::Closed)
                                window.close();
                        }
                    } while (!Keyboard::isKeyPressed(Keyboard::Return));
                }
```

```
if (WhatYaDoin.type == Event::MouseButtonReleased && WhatYaDoin.key.code == Mouse::Left && ZapiszText.getGlobalBounds().contains(Mouse))
{
	//cout << "Zapisano" << endl;
	string tmp = NazwaText.getString();
	//cout << "TPM: " << tmp << endl;
	ofstream plik(tmp.c_str());
	if (plik.is_open())
	{
		//cout << "Otwarto" << endl;
		plik << NewPendulum.Length << endl;
		plik << NewPendulum.Mass << endl;
		plik << NewPendulum.Alpha << endl;
		plik << NewPendulum.Gravity << endl;
		plik.close();
	}
}
if (WhatYaDoin.type == Event::MouseButtonReleased && WhatYaDoin.key.code == Mouse::Left && WczytajText.getGlobalBounds().contains(Mouse))
{
	string tmp = NazwaText.getString();
	ifstream plik(tmp.c_str());
	string line;
	int i = 0;
	while (!plik.eof())
	{
		getline(plik, line);
		switch (i)
```

```
                        {
                        case 0:
                            NewPendulum.Length = stof(line); break;
                        case 1:
                            NewPendulum.Mass = stof(line); break;
                        case 2:
                            NewPendulum.Alpha = stof(line); break;
                        case 3:
                            NewPendulum.Gravity = stof(line); break;
                        }
                        i++;
                    }
                }
            }
            Drawing(NewPendulum);
        } while (window.isOpen());
}

void Drawing(Pendulum &NewPendulum)
{
    window.clear();

    window.draw(Background);
    window.draw(PenBackground);
    window.draw(InfoBackground);

    window.draw(NewPendulum.PenSprite);
```

```
        window.draw(StartSimText);
        window.draw(GraphsText);
        window.draw(SetAngleText);
        window.draw(AngleText);
        window.draw(SetLengthText);
        window.draw(LengthText);
        window.draw(SetMassText);
        window.draw(MassText);
        window.draw(InformationText);
        window.draw(PomocText);
        window.draw(NazwaSText);
        window.draw(SetGravityText);
        window.draw(GravityText);
        window.draw(ZapiszText);
        window.draw(NazwaText);
        window.draw(WczytajText);
        window.draw(CurrAccText);
        window.draw(CurrAlpha);
        window.draw(CurrRotAccText);
        window.draw(CurrRotVelText);
        window.draw(CurrVelText);
        window.draw(SimTime);
        window.draw(CurrForceText);

        window.display();
    }
```

```
string Insertingf()
{
    string Return;
    while (window.pollEvent(WhatYaDoin))
    {
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::Num1)
            Return = "1";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::Num2)
            Return = "2";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::Num3)
            Return = "3";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::Num4)
            Return = "4";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::Num5)
            Return = "5";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::Num6)
            Return = "6";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::Num7)
            Return = "7";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::Num8)
            Return = "8";
```

```
			if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::Num9)

				Return = "9";

			if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::Num0)

				Return = "0";

			if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::Period)

				Return = ".";

			if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::Dash)

				Return = "-";

	}

	return Return;

}


void SetBeginValues()

{

	Background.setFillColor(Color::Color(128, 128, 128, 125));

	Background.setSize(Vector2f(1024, 720));


	PenBackground.setFillColor(Color::Color(100, 135, 138, 255));

	PenBackground.setSize(Vector2f(600, 300));


	InfoBackground.setFillColor(Color::Color(128, 128, 128, 255));

	InfoBackground.setSize(Vector2f(450, 100));

	InfoBackground.setPosition(0, 620-100);


	StartSimTexture.loadFromFile("Data/button_sym.png");
```

```
StartSimTexture2.loadFromFile("Data/button_sym2.png");
StartSimText.setTexture(StartSimTexture);
StartSimText.setPosition(220, 460);

GraphsTexture.loadFromFile("Data/button_wykresy.png");
GraphsTexture2.loadFromFile("Data/button_wykresy2.png");
GraphsText.setTexture(GraphsTexture);
GraphsText.setPosition(215, 410);

PomocTexture.loadFromFile("Data/button_pomoc.png");
PomocTexture2.loadFromFile("Data/button_pomoc2.png");
PomocText.setTexture(PomocTexture);
PomocText.setPosition(325, 410);

AngleTexture.loadFromFile("Data/button_kat.png");
AngleTexture2.loadFromFile("Data/button_kat2.png");
SetAngleText.setTexture(AngleTexture);
SetAngleText.setPosition(20, 320);

AngleText.setFont(font);
AngleText.setColor(Color::Yellow);
AngleText.setCharacterSize(15);
AngleText.setPosition(20, 360);

LengthTexture.loadFromFile("Data/button_nic.png");
LengthTexture2.loadFromFile("Data/button_nic2.png");
SetLengthText.setTexture(LengthTexture);
SetLengthText.setPosition(140, 320);

LengthText.setFont(font);
LengthText.setColor(Color::Yellow);
LengthText.setCharacterSize(15);
LengthText.setPosition(140, 360);

MassTexture.loadFromFile("Data/button_masa.png");
MassTexture2.loadFromFile("Data/button_masa2.png");
SetMassText.setTexture(MassTexture);
SetMassText.setPosition(295, 320);

GravityTexture.loadFromFile("Data/button_graw.png");
GravityTexture2.loadFromFile("Data/button_graw2.png");
SetGravityText.setTexture(GravityTexture);
SetGravityText.setPosition(20, 390);
SetGravityText.setScale(0.8, 0.8);

GravityText.setFont(font);
GravityText.setColor(Color::Yellow);
GravityText.setCharacterSize(15);
GravityText.setPosition(20, 430);

NazwaTexture.loadFromFile("Data/button_nazwa.png");
NazwaTexture2.loadFromFile("Data/button_nazwa2.png");
NazwaSText.setTexture(NazwaTexture);
NazwaSText.setScale(1, 0.95);
NazwaSText.setPosition(430, 320);

ZapiszTexture.loadFromFile("Data/button_zapisz.png");
ZapiszTexture2.loadFromFile("Data/button_zapisz2.png");
ZapiszText.setTexture(ZapiszTexture);
ZapiszText.setScale(1, 0.95);
ZapiszText.setPosition(430, 390);

WczytajTexture.loadFromFile("Data/button_wczytaj.png");
WczytajTexture2.loadFromFile("Data/button_wczytaj2.png");
WczytajText.setTexture(WczytajTexture);
WczytajText.setScale(1, 0.95);
WczytajText.setPosition(430, 440);

MassText.setFont(font);
MassText.setColor(Color::Yellow);
MassText.setCharacterSize(15);
MassText.setPosition(295, 360);

InformationText.setFont(font);
InformationText.setColor(Color::Yellow);
InformationText.setCharacterSize(15);
InformationText.setPosition(10, 620-105);

NazwaText.setFont(font);
NazwaText.setColor(Color::Yellow);
NazwaText.setCharacterSize(15);
NazwaText.setPosition(430, 355);

CurrVelText.setFont(font);
CurrVelText.setColor(Color::Yellow);
CurrVelText.setCharacterSize(15);
CurrVelText.setPosition(10, 10);

CurrAccText.setFont(font);
CurrAccText.setColor(Color::Yellow);
CurrAccText.setCharacterSize(15);
CurrAccText.setPosition(10, 40);

CurrRotAccText.setFont(font);
CurrRotAccText.setColor(Color::Yellow);
CurrRotAccText.setCharacterSize(15);
CurrRotAccText.setPosition(10, 70);

CurrRotVelText.setFont(font);
CurrRotVelText.setColor(Color::Yellow);
CurrRotVelText.setCharacterSize(15);
CurrRotVelText.setPosition(10, 100);

CurrAlpha.setFont(font);
CurrAlpha.setColor(Color::Yellow);
CurrAlpha.setCharacterSize(15);
CurrAlpha.setPosition(10, 130);
```

```
	SimTime.setFont(font);

	SimTime.setColor(Color::Yellow);

	SimTime.setCharacterSize(15);

	SimTime.setPosition(10, 160);


	CurrForceText.setFont(font);

	CurrForceText.setColor(Color::Yellow);

	CurrForceText.setCharacterSize(15);

	CurrForceText.setPosition(10, 190);


	CurrForceText.setString("Obecna sila napedzajaca [N]: ");

	GravityText.setString("Przysp. grawitacyjne [m/s]: ");

	SimTime.setString("Czas symulacji [s]: ");

	CurrVelText.setString("Obecna predkosc [m/s]: ");

	CurrAccText.setString("Obecne przyspieszenie [m/s^2]: ");

	CurrRotAccText.setString("Obecne przyspieszenie katowe [deg/s^2]: ");

	CurrRotVelText.setString("Obecna predkosc katowa [deg/s]: ");

	CurrAlpha.setString("Obecny kat [deg]: ");

	AngleText.setString("Kat [deg]: ");

	LengthText.setString("Dlugosc [m]: ");

	MassText.setString("Masa [kg]: ");

}


void SetDeg(Pendulum &NewPendulum)

{

	InformationText.setString("Wpisz kat w stopniach i zatwierdz ENTERem.");

	AngleText.setString("Kat [deg]: ");
```

```
while (!Keyboard::isKeyPressed(Keyboard::Return))
{
    string tmp;
    tmp = Insertingf();
    AngleText.setString(AngleText.getString() + tmp);
    Drawing(NewPendulum);
}
string tmp3 = AngleText.getString();
if (tmp3.size() == 11)
{
    AngleText.setString(AngleText.getString() + to_string(NewPendulum.Alpha));
}
string tmp2;
for (int i = 11; i < AngleText.getString().getSize(); i++)
{
    if (i == 11 && AngleText.getString()[i])
        tmp2 = AngleText.getString()[i];
    else
        tmp2 += AngleText.getString()[i];
}
if (stof(tmp2) > 90)
{
    tmp2 = "90";
    AngleText.setString("Kat [deg]: 90");
}
if (stof(tmp2) < -90)
{
        tmp2 = "-90";

        AngleText.setString("Kat [deg]: -90");

    }

    NewPendulum.Alpha = stof(tmp2);

    NewPendulum.PenSprite.setRotation(NewPendulum.Alpha);

}


void SetLength(Pendulum &NewPendulum)

{

    InformationText.setString("Wpisz dlugosc w metrach i zatwierdz ENTERem.");

    LengthText.setString("Dlugosc [m]: ");

    while (!Keyboard::isKeyPressed(Keyboard::Return))

    {

        string tmp;

        tmp = Insertingf();

        LengthText.setString(LengthText.getString() + tmp);

        Drawing(NewPendulum);

    }

    string tmp3 = LengthText.getString();

    if (tmp3.size() == 13)

    {

        LengthText.setString(LengthText.getString() + to_string(NewPendulum.Length));

    }

    string tmp2;

    for (int i = 13; i < LengthText.getString().getSize(); i++)

    {

        if (i == 13 && LengthText.getString()[i])
```

```
            tmp2 = LengthText.getString()[i];
        else
            tmp2 += LengthText.getString()[i];
    }
    if (stof(tmp2) <= 0)
    {
        tmp2 = "0.001";
        LengthText.setString("Dlugosc [m]: 0.001");
    }
    NewPendulum.Length = stof(tmp2);
}

void SetMass(Pendulum &NewPendulum)
{
    InformationText.setString("Wpisz masę w kilogramach i zatwierdz ENTERem.");
    MassText.setString("Masa [kg]: ");
    while (!Keyboard::isKeyPressed(Keyboard::Return))
    {
        string tmp;
        tmp = Insertingf();
        MassText.setString(MassText.getString() + tmp);
        Drawing(NewPendulum);
    }
    string tmp3 = MassText.getString();
    if (tmp3.size() == 11)
    {
        MassText.setString(MassText.getString() + to_string(NewPendulum.Mass));
        }

        string tmp2;

        for (int i = 11; i < MassText.getString().getSize(); i++)

        {

                if (i == 11 && MassText.getString()[i])

                        tmp2 = MassText.getString()[i];

                else

                        tmp2 += MassText.getString()[i];

        }

        if (stof(tmp2) <= 0)

        {

                tmp2 = "0.001";

                MassText.setString("Masa [kg]: 0.001");

        }

        NewPendulum.Mass = stof(tmp2);

}


void StartSim(Pendulum &NewPendulum)

{

        InformationText.setString("Zbieram dane.\nAby przestac zbierac dane wcisnij ENTER.");

        NewPendulum.Force = 0;

        NewPendulum.Acceleration = 0;

        NewPendulum.RotAcceleration = 0;

        NewPendulum.RotVelocity = 0;

        NewPendulum.Velocity = 0;

        float PenTime = 2.f * 3.141592 * (sqrt(NewPendulum.Length / NewPendulum.Gravity)) * (1 +
(1/16)*pow((NewPendulum.Alpha / 180 * 3.141592), 2) + (11/3072)*pow((NewPendulum.Alpha /
180 * 3.141592), 4) + (173/737280)*pow((NewPendulum.Alpha / 180 * 3.141592), 6) +
(22931/1321205760) * pow((NewPendulum.Alpha / 180 * 3.141592), 8));
```

```
if (NewPendulum.Alpha == 0)
	PenTime = 0;
float StartAlpha = NewPendulum.Alpha;
Clock deltaTime;
Time clock1;
Clock risingTime;
Time clock2;

fstream sila;
fstream przysp;
fstream predkosc;
fstream omega;
fstream przyspkatowe;
fstream kat;

sila.open("GraphsData/SilaNapedzajaca.txt", ios::trunc | ios::out | ios::in);
przysp.open("GraphsData/PrzyspieszenieLiniowe.txt", ios::trunc | ios::out | ios::in);
predkosc.open("GraphsData/PredkoscLiniowa.txt", ios::trunc | ios::out | ios::in);
omega.open("GraphsData/PredkoscKatowa.txt", ios::trunc | ios::out | ios::in);
przyspkatowe.open("GraphsData/PrzyspieszenieKatowe.txt", ios::trunc | ios::out | ios::in);
kat.open("GraphsData/Kat.txt", ios::trunc | ios::out | ios::in);

risingTime.restart();
do
{
	deltaTime.restart();
	sleep(milliseconds(15));
```

```
			clock2 = risingTime.getElapsedTime();
			NewPendulum.Force = (NewPendulum.Mass * NewPendulum.Gravity) *
sin(NewPendulum.Alpha / 180 * 3.141592);
			CurrForceText.setString("Obecna sila napedzajaca [N]: " +
to_string(NewPendulum.Force));
			sila << clock2.asSeconds() << " " << NewPendulum.Force << endl;
			clock2 = risingTime.getElapsedTime();
			NewPendulum.Acceleration = NewPendulum.Force / NewPendulum.Mass;
			CurrAccText.setString("Obecne przyspieszenie [m/s^2]: " +
to_string(NewPendulum.Acceleration));
			przysp << clock2.asSeconds() << " " << NewPendulum.Acceleration << endl;
			clock2 = risingTime.getElapsedTime();
			NewPendulum.RotAcceleration = NewPendulum.Acceleration /
NewPendulum.Length;
			CurrRotAccText.setString("Obecne przyspieszenie katowe [deg/s^2]: " +
to_string(NewPendulum.RotAcceleration));
			przyspkatowe << clock2.asSeconds() << " " << NewPendulum.RotAcceleration <<
endl;
			clock2 = risingTime.getElapsedTime();
			clock1 = deltaTime.getElapsedTime();
			NewPendulum.RotVelocity += NewPendulum.RotAcceleration *
(float)clock1.asSeconds();
			CurrRotVelText.setString("Obecna predkosc katowa [deg/s]: " +
to_string(NewPendulum.RotVelocity));
			omega << clock2.asSeconds() << " " << NewPendulum.RotVelocity << endl;
			NewPendulum.Velocity += NewPendulum.Acceleration * (float)clock1.asSeconds();
			CurrVelText.setString("Obecna predkosc [m/s]: " +
to_string(NewPendulum.Velocity));
			predkosc << clock2.asSeconds() << " " << NewPendulum.Velocity << endl;
			NewPendulum.Alpha -= (NewPendulum.RotVelocity * (float)clock1.asSeconds()) *
180 / 3.141592;
```

```
        CurrAlpha.setString("Obecny kat [deg]: " + to_string(NewPendulum.Alpha));
        kat << clock2.asSeconds() << " " << NewPendulum.Alpha << endl;
        NewPendulum.PenSprite.setRotation(NewPendulum.Alpha);
        clock2 = risingTime.getElapsedTime();
        SimTime.setString("Czas symulacji [s]: " + to_string(clock2.asSeconds()));
        Drawing(NewPendulum);
        Vector2f MousePos(Mouse::getPosition(window));
    } while (!Keyboard::isKeyPressed(Keyboard::Return));


    sila.close();
    przysp.close();
    predkosc.close();
    omega.close();
    przyspkatowe.close();
    kat.close();
}


void SetFileName(Pendulum &NewPendulum)
{
    NazwaText.setString("");
    while (!Keyboard::isKeyPressed(Keyboard::Return))
    {
        string tmp;
        tmp = Insertingf2();
        NazwaText.setString(NazwaText.getString() + tmp);
        Drawing(NewPendulum);
    }
```

```
}


string Insertingf2()
{
    InformationText.setString("Wpisz nazwe pliku (alfanumerycznie) i zatwierdz ENTERem.");
    string Return;
    while (window.pollEvent(WhatYaDoin))
    {
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::Num1)
            Return = "1";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::Num2)
            Return = "2";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::Num3)
            Return = "3";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::Num4)
            Return = "4";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::Num5)
            Return = "5";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::Num6)
            Return = "6";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::Num7)
            Return = "7";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::Num8)
```

```
            Return = "8";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::Num9)
            Return = "9";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::Num0)
            Return = "0";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::Period)
            Return = ".";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::W)
            Return = "w";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::Q)
            Return = "q";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::E)
            Return = "e";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::R)
            Return = "r";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::T)
            Return = "t";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::Y)
            Return = "y";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::U)
            Return = "u";
```

```
                if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::I)
                    Return = "i";
                if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::O)
                    Return = "o";
                if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::P)
                    Return = "p";
                if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::A)
                    Return = "a";
                if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::S)
                    Return = "s";
                if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::D)
                    Return = "d";
                if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::F)
                    Return = "f";
                if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::G)
                    Return = "g";
                if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::H)
                    Return = "h";
                if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::J)
                    Return = "j";
                if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code ==
Keyboard::K)
```

```
            Return = "k";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::L)
            Return = "l";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::Z)
            Return = "z";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::X)
            Return = "x";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::C)
            Return = "c";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::V)
            Return = "v";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::B)
            Return = "b";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::N)
            Return = "n";
        if (WhatYaDoin.type == Event::KeyReleased && WhatYaDoin.key.code == Keyboard::M)
            Return = "m";
    }
    return Return;
}

void SetGravity(Pendulum &NewPendulum)
{
```

```
InformationText.setString("Wpisz przyspieszenie grawitacyjne w m/s i zatwierdz ENTERem.");
GravityText.setString("Przysp. grawitacyjne [m/s]: ");
while (!Keyboard::isKeyPressed(Keyboard::Return))
{
    string tmp;
    tmp = Insertingf();
    GravityText.setString(GravityText.getString() + tmp);
    Drawing(NewPendulum);
}
string tmp3 = GravityText.getString();
if (tmp3.size() == 28)
{
    GravityText.setString(GravityText.getString() + to_string(NewPendulum.Gravity));
}
string tmp2;
for (int i = 28; i < GravityText.getString().getSize(); i++)
{
    if (i == 28 && GravityText.getString()[i])
        tmp2 = GravityText.getString()[i];
    else
        tmp2 += GravityText.getString()[i];
}
if (stof(tmp2) <= 0)
{
    tmp2 = "0.001";
    GravityText.setString("Przysp. grawitacyjne [m/s]: " + tmp2);
}
```

```
        NewPendulum.Gravity = stof(tmp2);

}
```