```cpp
/*
    Marshall Lindsay
    Max Houck
    Formula SAE Tire Temperature Visualization
    ECE 3220 Final Project
    main.cpp
*/
#include<windows.h>
#include "tires.h"
#include "SerialClass.h"
#include "CarGraphics.h"

#define LOGGING_RATE .1
#define ACCEPTABLE_PERCENTAGE .05

void programStart(void);
void mainOptionMenu(void);
void dataVisualization(void);
void dataVisualizationWelcomeMessage(void);
void simulation(Car* car, CarGraphics*);
void realTimeDemo(void);
Car* carSetup(void);
void graphicsSetup(CarGraphics*);
void graphicsSetupOptionMenu(void);
void realTimeGraphicsSetup(RealTimeGraphics*);
void realTimeGraphicsSetupOptionMenu(void);


int main() {
    try{
        programStart();
    }
    catch(...){
        cout<<"\nProgram Failed. Catch all Main()"<<endl;
        return(1);
    }
}

void programStart(void){
    string userInput;
    for(int i = 0; i < 60; i++){
        cout<<"#";
    }
    cout<<"\n\nFSAE TIRE TEMPERATURE DATA VISUALIZATION SOFTWARE\n"<<endl;
    for(int i = 0; i < 60; i++){
        cout<<"#";
    }
    cout<<"\n\n\nWelcome to version 1 of the FSAE Tire Temperature Visualization
    Software!"<<endl;
    while(1){
        mainOptionMenu();
        getline(cin, userInput);

        if(userInput == "1"){
            realTimeDemo();
        }else if(userInput == "2"){
            dataVisualization();
        }else if(userInput == "3"){
            return;
        }else{
            cout<<"\n Invalid input!"<<endl;
        }
    }
}

void mainOptionMenu(void){
    cout<<"\n\nPlease select from the following options: "
        <<"\n   1)Real time data sensor visualization"
        <<"\n   2)Data visualization from file and video"
```

```
        <<"\n   3)Quit"<<endl;
}

void realTimeDemo(void){
    string userInput;
    HWND myconsole = GetConsoleWindow();
    HDC mydc = GetDC(myconsole);
    RealTimeGraphics* realTime = new RealTimeGraphics();
    realTimeGraphicsSetup(realTime);
    cout<<"\n Real Time graphics setup complete... Press any key to begin"<<endl;
    getline(cin, userInput);
    Serial* SP;

    while(1){
        try{
            SP = new Serial("COM4"); //We may need to adjust this as necessary
            break;
        }
        catch(const int x){
            if(x == 1){
                cout<<"\n Please connect the arduino!"<<endl;
                Sleep(3000);
            }else if(x == 2){
                cout<<"\n Unknown error!"<<endl;
            }
        }
        catch(...){
            cout<<"\n Catch all realTimeDemo. Should not be here!"<<endl;
        }
    }

    if (SP->IsConnected()){
        cout<<"\n Ardiuno Connected!"<<endl;
    }else{
        cout<<"\n Arduino not connected?.. Terminating!"<<endl;
    }
    char* token;
    string incomingDataString;
    char incomingData[256] = "";
    int dataLength = 255;
    int readResult = 0;
    int data1 = 0;
    int data2 = 0;
    int data3 = 0;
    int bufferSize = 5;
    int bufferHold1,bufferHold2;
    int buf1[bufferSize], buf2[bufferSize], buf3[bufferSize];
    int data1Hold, data2Hold, data3Hold;
    int average1,average2,average3, i;

    int slope1, slope2, slope3;
    int estimatedPoint1, estimatedPoint2, estimatedPoint3;
    double percentDiff1, percentDiff2, percentDiff3;
    double averageDelta;
    while(1){

        data1Hold = data1;
        data2Hold = data2;
        data3Hold = data3;
        readResult = SP->ReadData(incomingData,dataLength);

        token = strtok(incomingData,","); //parsing serial output from arduino
        data1 = atoi(token);
        token = strtok(NULL, ",");
        data2 = atoi(token);
        token = strtok(NULL, ",");
        data3 = atoi(token);

        if(buf1[0] == 0) { //if buffers are empty
```

```
138             for(i=0;i<bufferSize;i++) {
139                 buf1[i] = data1; //fill them completely with first value
140                 buf2[i] = data2;
141                 buf3[i] = data3;
142             }
143         }
144         int sum = 0; //calculate averages of buffers and error check
145         /*
146         for(int i=0;i<bufferSize;i++){
147             sum+=buf1[i];
148         }
149         average1 = sum / bufferSize;
150         sum = 0;
151         for(int i = 0; i < bufferSize; i++){
152             sum += buf1[i] - average1;
153         }
154         averageDelta = sum / buffersize;
155         percentDiff1 = (data1 - average1) / averageDelta;
156         if(percentDiff1 > ACCEPTABLE_PERCENTAGE && percentDiff1 <
            (ACCEPTABLE_PERCENTAGE + .01)
157             || percentDiff1 < ACCEPTABLE_PERCENTAGE  && percentDiff1 >
                (ACCEPTABLE_PERCENTAGE - .01)){
158             data1 *= .9;
159         }
160         if(percentDiff1 > (ACCEPTABLE_PERCENTAGE + .01) && percentDiff1 <
            (ACCEPTABLE_PERCENTAGE + .05)
161             || percentDiff1 < (ACCEPTABLE_PERCENTAGE - .01)  && percentDiff1 >
                (ACCEPTABLE_PERCENTAGE - .05)){
162             data1 *= .8;
163         }
164         if(percentDiff1 > (ACCEPTABLE_PERCENTAGE + .05) && percentDiff1 <
            (ACCEPTABLE_PERCENTAGE + .1)
165             || percentDiff1 < (ACCEPTABLE_PERCENTAGE - .05)  && percentDiff1 >
                (ACCEPTABLE_PERCENTAGE - .1)){
166             data1 *= .6;
167         }
168         if(percentDiff1 > (ACCEPTABLE_PERCENTAGE + .1) && percentDiff1 <
            (ACCEPTABLE_PERCENTAGE + .2)
169             || percentDiff1 < (ACCEPTABLE_PERCENTAGE - .1)  && percentDiff1 >
                (ACCEPTABLE_PERCENTAGE - .2)){
170             data1 *= .;
171         }
172         */
173         sum = 0;
174         for(int i=0;i<bufferSize;i++)
175             sum+=buf1[i];
176         average1 = sum / bufferSize;
177         if (data1 > average1 + 10 | data1 < average1 - 10)
178             data1 = average1;
179         sum = 0;
180         for(int i=0;i<bufferSize;i++)
181             sum+=buf2[i];
182         average2 = sum / bufferSize;
183         if (data2 > average2 + 10 | data2 < average2 - 10)
184             data2 = average2;
185         sum = 0;
186         for(int i=0;i<bufferSize;i++)
187             sum+=buf3[i];
188         average3 = sum / bufferSize;
189         if (data3 > average3 + 10 | data3 < average3 - 10)
190             data3 = average3;
191
192
193          //rotate it in the new values to each buffer
194         bufferHold1 = buf1[0];
195         buf1[0] = data1;
196         for(int i=1;i<bufferSize;i++) {
197             bufferHold2 = buf1[i];
198             buf1[i] = bufferHold1;
```

```
199             bufferHold1 = bufferHold2;
200         }
201         bufferHold1 = buf2[0];
202         buf2[0] = data2;
203         for(int i=1;i<bufferSize;i++) {
204             bufferHold2 = buf2[i];
205             buf2[i] = bufferHold1;
206             bufferHold1 = bufferHold2;
207         }
208         bufferHold1 = buf3[0];
209         buf3[0] = data3;
210         for(int i =1;i<bufferSize;i++) {
211             bufferHold2 = buf3[i];
212             buf3[i] = bufferHold1;
213             bufferHold1 = bufferHold2;
214         }
215 
216         realTime->setData(data1, data2, data3); //output to graphics
217         realTime->draw(&mydc);
218         Sleep(150);
219         }
220     delete realTime;
221 }
222 
223 
224 
225 void realTimeGraphicsSetup(RealTimeGraphics* graphics){
226     string userInput;
227     HWND myconsole = GetConsoleWindow();
228     HDC mydc = GetDC(myconsole);
229     int value;
230     while(1){
231 
232         realTimeGraphicsSetupOptionMenu();
233         getline(cin,userInput);
234         if(userInput == "1"){
235             graphics->draw(&mydc);
236             getline(cin, userInput);
237         }else if(userInput == "2"){
238             int value2;
239             cout<<"\n Please enter the new X value: "<<endl;
240             cin>>value;
241             cout<<"\n Please enter the new Y value: "<<endl;
242             cin>>value2;
243             graphics->move(value, value2);
244         }else if(userInput == "3"){
245             cout<<"\n Please enter the value you wish to vertically shift the graph
                by:"<<endl;
246             cin>>value;
247             graphics->verticalShift(value);
248         }else if(userInput == "4"){
249             cout<<"\n Please enter the value you wish to horizontally shift the graph
                by:"<<endl;
250             cin>>value;
251             graphics->horizontalShift(value);
252         }else if(userInput == "5"){
253             int value2;
254             cout<<"\n Please enter the new length:"<<endl;
255             cin>>value;
256             cout<<"\n Please enter the new width:"<<endl;
257             cin>>value2;
258             graphics->resize(value, value2);
259         }else if(userInput == "6"){
260             int value2;
261             cout<<"\n Please enter the lowest temperature:"<<endl;
262             cin>>value;
263             cout<<"\n Please enter the highest temperature:"<<endl;
264             cin>>value2;
265             graphics->setRange(value, value2);
```

```
266         }else if(userInput == "7"){
267             break;
268         }else{
269             cout<<"\n Invalid Input!"<<endl;
270         }
271 
272         fflush(stdin);
273     }
274     return;
275 }
276 
277 void realTimeGraphicsSetupOptionMenu(void){
278     cout<<"\n Graphics Setup Options:"
279         <<"\n 1)Print"
280         <<"\n 2)Move"
281         <<"\n 3)Vertical Shift"
282         <<"\n 4)Horizontal Shift"
283         <<"\n 5)Resize graph"
284         <<"\n 6)Set temperature range"
285         <<"\n 7)Finish"<<endl;
286 }
287 
288 void dataVisualization(void){
289     string userInput;
290     CarGraphics* graphicsCar = new CarGraphics();
291     graphicsSetup(graphicsCar);
292     Car* car = carSetup();
293 
294     cout<<"\n\n\n\n Car creation and GUI setup complete.\nPress any key to
        continue"<<endl;
295     getline(cin, userInput);
296     simulation(car, graphicsCar);
297     cout<<"\n Simulation complete!"<<endl;
298     while(1){
299 
300             cout<<"\n What would you like to do?"
301             <<"\n 1)Run the simulation"
302             <<"\n 2)Change car data"
303             <<"\n 3)Change graphics settings"
304             <<"\n 4)Quit to main menu"<<endl;
305         getline(cin,userInput);
306 
307         if(userInput == "1"){
308             simulation(car, graphicsCar);
309         }else if(userInput == "2"){
310             delete car;
311             car = carSetup();
312         }else if(userInput == "3"){
313             graphicsSetup(graphicsCar);
314         }else if(userInput == "4"){
315             break;
316         }
317     }
318     delete car;
319     delete graphicsCar;
320 
321 }
322 
323 void graphicsSetup(CarGraphics* car){
324     string userInput;
325     HWND myconsole = GetConsoleWindow();
326     HDC mydc = GetDC(myconsole);
327     int value;
328     while(1){
329 
330         graphicsSetupOptionMenu();
331         getline(cin,userInput);
332         if(userInput == "1"){
333             car->drawGraphics(&mydc);
```

```
334             car->drawGraphics(&mydc);
335             getline(cin, userInput);
336         }else if(userInput == "2"){
337             int x, y;
338             cout<<"\n Please enter the new X value:"<<endl;
339             cin>>x;
340             cout<<"\n Please enter the new Y value:"<<endl;
341             cin>>y;
342             car->move(x, y);
343         }else if(userInput == "3"){
344             cout<<"\n Please enter the value you wish to shift the object vertically
                by:"<<endl;
345             cin>>value;
346             car->verticalShift(value);
347         }else if(userInput == "4"){
348             cout<<"\n Please enter the value you wish to shift the object horizontally
                by:"<<endl;
349             cin>>value;
350             car->horizontalShift(value);
351         }else if(userInput == "5"){
352             cout<<"\n Please enter the new size for the tires:"<<endl;
353             cin>>value;
354             car->resize(value);
355         }else if(userInput == "6"){
356             cout<<"\n Please enter the new size for the front left tire:"<<endl;
357             cin>>value;
358             car->flResize(value);
359         }else if(userInput == "7"){
360             cout<<"\n Please enter the new size for the front right tire:"<<endl;
361             cin>>value;
362             car->frResize(value);
363         }else if(userInput == "8"){
364             cout<<"\n Please enter the new size for the rear left tire:"<<endl;
365             cin>>value;
366             car->rlResize(value);
367         }else if(userInput == "9"){
368             cout<<"\n Please enter the new size for the rear right tire:"<<endl;
369             cin>>value;
370             car->rrResize(value);
371         }else if(userInput == "10"){
372             cout<<"\n Please enter the new offset value:"<<endl;
373             cin>>value;
374             car->updateOffset(value, value);
375         }else if(userInput == "11"){
376             int value2;
377             cout<<"\n Please enter the low temperature:"<<endl;
378             cin>>value;
379             cout<<"\n Please enter the high temperature:"<<endl;
380             cin>>value2;
381             car->setRange(value, value2);
382         }else if(userInput == "12"){
383             break;
384         }else{
385             cout<<"\nInvalid Input!"<<endl;
386         }
387         fflush(stdin);
388     }
389 
390     return;
391 }
392 
393 void graphicsSetupOptionMenu(void){
394     cout<<"\n Graphics Setup Options:"
395         <<"\n 1)Print"
396         <<"\n 2)Move"
397         <<"\n 3)Vertical Shift"
398         <<"\n 4)Horizontal Shift"
399         <<"\n 5)Resize Every Tire"
400         <<"\n 6)Resize Front Left Tire"
```

```
401         <<"\n 7)Resize Front Right Tire"
402         <<"\n 8)Resize Rear Left Tire"
403         <<"\n 9)Resize Rear Right Tire"
404         <<"\n 10)Increase Offset From MidPoint"
405         <<"\n 11)Set temperature range"
406         <<"\n 12)Finish"<<endl;
407 }
408
409
410 void simulation(Car* car, CarGraphics* graphicsCar){
411     HWND myconsole = GetConsoleWindow();
412     HDC mydc = GetDC(myconsole);
413     graphicsCar->setRange(0, 110);
414     for(int i = 0; i < car->numDataPoints(); i++){
415         graphicsCar->setDataFrontLeft(car->getTemperature(frontLeft, outer, i),
416                                       car->getTemperature(frontLeft, middle, i),
417                                       car->getTemperature(frontLeft, inner, i));
418
419         graphicsCar->setDataFrontRight(car->getTemperature(frontRight, outer, i),
420                                       car->getTemperature(frontRight, middle, i),
421                                       car->getTemperature(frontRight, inner, i));
422
423         graphicsCar->setDataRearLeft(car->getTemperature(rearLeft, outer, i),
424                                       car->getTemperature(rearLeft, middle, i),
425                                       car->getTemperature(rearLeft, inner, i));
426
427         graphicsCar->setDataRearRight(car->getTemperature(rearRight, outer, i),
428                                       car->getTemperature(rearRight, middle, i),
429                                       car->getTemperature(rearRight, inner, i));
430         graphicsCar->drawGraphics(&mydc);
431         Sleep(20);
432
433
434     }
435 }
436
437 void dataVisualizationWelcomeMessage(void){
438     string input;
439     for(int i = 0; i < 60; i++){
440         cout<<"-";
441     }
442     cout<<"\n Setup Wizard"<<endl;
443     cout<<"\n This wizard will guide you through the steps to set up your data for"
444         <<"\n visualization. Please follow the steps carefully."
445         <<"\n Press any key to continue"<<endl;
446
447     getline(cin,input);
448 }
449
450 Car* carSetup(void){
451     string userInput;
452     dataVisualizationWelcomeMessage();
453     ifstream dataFile1, dataFile2, dataFile3, dataFile4;
454
455     cout<<"\n Please enter the name of the car : "<<endl;
456     getline(cin, userInput);
457
458     Car* car = new Car(userInput);
459     try{
460
461         while(1){
462             cout<<"\n Please enter the name of the data file for the Front Left Tire:
                "<<endl;
463             getline(cin, userInput);
464             if(userInput == "quit"){
465                 throw(1);
466             }
467             dataFile1.open((userInput.c_str()), ios::in);
468             if(!dataFile1.is_open()){
```

```
469                 cout<<"\n Could not open the file! Please try again or enter 'quit' to
                    exit"<<endl;
470             }else{
471                 break;
472             }
473         }
474 
475         while(1){
476             cout<<"\n Please enter the name of the data file for the Front Right Tire:
                "<<endl;
477             getline(cin, userInput);
478             if(userInput == "quit"){
479                 dataFile1.close();
480                 throw(1);
481             }
482             dataFile2.open((userInput.c_str()), ios::in);
483             if(!dataFile2.is_open()){
484                 cout<<"\n Could not open the file! Please try again or enter 'quit' to
                    exit"<<endl;
485             }else{
486                 break;
487             }
488         }
489 
490         while(1){
491             cout<<"\n Please enter the name of the data file for the Rear Left Tire:
                "<<endl;
492             getline(cin, userInput);
493                 if(userInput == "quit"){
494                     dataFile1.close();
495                     dataFile2.close();
496                     throw(1);
497                 }
498             dataFile3.open((userInput.c_str()), ios::in);
499             if(!dataFile3.is_open()){
500                 cout<<"\n Could not open the file! Please try again or enter 'quit' to
                    exit"<<endl;
501             }else{
502                 break;
503             }
504         }
505 
506         while(1){
507             cout<<"\n Please enter the name of the data file for the Rear Right Tire:
                "<<endl;
508             getline(cin, userInput);
509                 if(userInput == "quit"){
510                     dataFile3.close();
511                     throw(1);
512                 }
513             dataFile4.open((userInput.c_str()), ios::in);
514             if(!dataFile4.is_open()){
515                 cout<<"\n Could not open the file! Please try again or enter 'quit' to
                    exit"<<endl;
516             }else{
517                 break;
518             }
519         }
520         cout<<"\n Please wait while the Car is created and the data imported..."<<endl;
521         string line;
522         char* token;
523 
524         //Front left tire creation
525         Tire FL(frontLeft, "FrontLeft");
526         Sensor FL_Outer("FL_Outer", outer, 0x5A);
527         Sensor FL_Middle("FL_Middle", middle, 0x5A);
528         Sensor FL_Inner("FL_Inner", inner, 0x5A);
529         //Data for the sensor objects.
530         while(getline(dataFile1, line)){
```

```
            token = strtok((char*)line.c_str(),",");
            FL_Outer.addTemperature(atoi(token));
            token = strtok(NULL, ",");
            FL_Middle.addTemperature(atoi(token));
            token = strtok(NULL, ",");
            FL_Inner.addTemperature(atoi(token));
        }
        dataFile1.close();
        //Add the sensors to the tire
        FL.addSensor(FL_Outer);
        FL.addSensor(FL_Middle);
        FL.addSensor(FL_Inner);

        //Front right tire creation
        Tire FR(frontRight, "FrontRight");
        Sensor FR_Outer("FR_Outer", outer, 0x5A);
        Sensor FR_Middle("FR_Middle", middle, 0x5A);
        Sensor FR_Inner("FR_Inner", inner, 0x5A);
        while(getline(dataFile2, line)){
            token = strtok((char*)line.c_str(),",");
            FR_Outer.addTemperature(atoi(token));
            token = strtok(NULL, ",");
            FR_Middle.addTemperature(atoi(token));
            token = strtok(NULL, ",");
            FR_Inner.addTemperature(atoi(token));
        }
        dataFile2.close();

        //Add the sensors to the tire
        FR.addSensor(FR_Outer);
        FR.addSensor(FR_Middle);
        FR.addSensor(FR_Inner);

        //Rear left tire creation
        Tire RL(rearLeft, "RearLeft");
        Sensor RL_Outer("RL_Outer", outer, 0x5A);
        Sensor RL_Middle("RL_Middle", middle, 0x5A);
        Sensor RL_Inner("RL_Inner", inner, 0x5A);
        while(getline(dataFile3, line)){
            token = strtok((char*)line.c_str(),",");
            RL_Outer.addTemperature(atoi(token));
            token = strtok(NULL, ",");
            RL_Middle.addTemperature(atoi(token));
            token = strtok(NULL, ",");
            RL_Inner.addTemperature(atoi(token));
        }
        dataFile3.close();

        //Add the sensors to the tire
        RL.addSensor(RL_Outer);
        RL.addSensor(RL_Middle);
        RL.addSensor(RL_Inner);

        //Rear right tire creation
        Tire RR(rearRight, "RearRight");
        Sensor RR_Outer("RR_Outer", outer, 0x5A);
        Sensor RR_Middle("RR_Middle", middle, 0x5A);
        Sensor RR_Inner("RR_Inner", inner, 0x5A);
        while(getline(dataFile4, line)){
            token = strtok((char*)line.c_str(),",");
            RR_Outer.addTemperature(atoi(token));
            token = strtok(NULL, ",");
            RR_Middle.addTemperature(atoi(token));
            token = strtok(NULL, ",");
            RR_Inner.addTemperature(atoi(token));
        }
        dataFile4.close();
        //Add the sensors to the tire
        RR.addSensor(RR_Outer);
```

```
600             RR.addSensor(RR_Middle);
601             RR.addSensor(RR_Inner);
602 
603             //Add all of the tires to the car
604             car->addTire(FL);
605             car->addTire(FR);
606             car->addTire(RL);
607             car->addTire(RR);
608 
609             return(car);
610         }
611         catch(...){
612             return(NULL);
613         }
614     }
615 
```