I am working on making a City Builder simulator game in C++. The game is broken down into sections which are complete and each have their own main which runs. What I need help with is incorporating these main files into one big main which will run the whole program. In this prompt I will supply you with a main file, which runs an interface for transportation part of the program, and needs to be incorporated into the big main. In the next prompt I will give you the second main to also be incorporated into the big main, and so on. I will supply you with the main, what its for, and how I want it incorporated into the big main.

Here is a quick rundown of the program. The game works on a year-to-year basis in which the user has to make upgrades and modifications to the city in order to keep it running and achieve the highest satisfaction rate possible. The game will have a main menu which allows the user to navigate to different sections of the city which they can modify, progress to the next year, and quit the game. The main menu should look as follows:

============================

City Simulation Main Menu

============================

Choose an action:

1. Manage Transport

2. Manage Tax

3. Manage Buildings

4. Manage Utilities

5. Manage Citizens

6. Manage Government

7. View Resources

8. View Statistics

9. Progress Year

10. Quit

Enter your choice:

The main will have to keep track of statistics by having the following members: `Satisfaction`, `City Budget`, `Productivity`

Firstly, we will start with the Transportation section of the project. The user will need to enter `1` in order to enter this section. This section makes use of the State and Observer design patterns. The State pattern is used for three subsections (transportation systems), namely `Road`, `Public Transit`, and `Train System`. Each of these subsections have different states to represent the transportation system, which have different effects on productivity and satisfaction of the city. Below is the current main for this transportation section:

#include "Road.h"

#include "NotOperational.h"

#include "Bad.h"

#include "Okay.h"

#include "Good.h"

#include "Excellent.h"

#include "TrainSystem.h"

#include "Star1.h"

#include "Star2.h"

#include "Star3.h"

#include "Star4.h"

#include "Star5.h"

#include "PublicTransit.h"

#include "TransportObserver.h"

#include <iostream>

#include <memory>

#include <string>

#include <iomanip>

#include <sstream>

// Function to format money

std::string formatMoney(double amount) {

std::ostringstream oss;

if (amount < 1000.0) {

oss << "$" << static\_cast<long long>(amount);

}

else if (amount < 1000000.0) {

double thousands = amount / 1000.0;

oss << "$" << std::fixed << std::setprecision(1) << thousands << "k";

}

else {

double millions = amount / 1000000.0;

oss << "$" << std::fixed << std::setprecision(1) << millions << "m";

}

return oss.str();

}

// Management function for TrainSystem

void trainM(TrainSystem& train) {

bool runningT = true;

while (runningT) {

std::cout << "============================\n";

std::cout << "Current Train System State: " << train.getCurrentStateName() << "\n";

std::cout << "City Budget: " << formatMoney(train.getCityBudget()) << "\n";

std::cout << "Train Satisfaction: " << train.getSatisfaction() << "\n";

std::cout << "============================\n";

std::cout << "Choose an action:\n";

std::cout << "1. Upgrade Train System\n";

std::cout << "2. Exit\n";

std::cout << "Enter your choice: ";

int choiceT;

std::cin >> choiceT;

switch (choiceT) {

case 1:

train.upgradeTrainSystem();

break;

case 2:

std::cout << "Exiting Train.\n";

runningT = false;

break;

default:

std::cout << "Invalid choice. Please try again.\n";

}

std::cout << "\n";

}

}

// Management function for Road

void roadM(Road& road) {

bool runningR = true;

while (runningR) {

std::cout << "============================\n";

std::cout << "Current Road State: " << road.getCurrentStateName() << "\n";

std::cout << "City Budget: " << formatMoney(road.getCityBudget()) << "\n";

std::cout << "Road Satisfaction: " << road.getSatisfaction() << "\n";

std::cout << "============================\n";

std::cout << "Choose an action:\n";

std::cout << "1. Upgrade Road\n";

std::cout << "2. Exit\n";

std::cout << "Enter your choice: ";

int choiceR;

std::cin >> choiceR;

switch (choiceR) {

case 1:

road.upgradeRoad();

break;

case 2:

std::cout << "Exiting Road.\n";

runningR = false;

break;

default:

std::cout << "Invalid choice. Please try again.\n";

}

std::cout << "\n";

}

}

// Management function for Public Transit

void publicTransitM(PublicTransit& transit) {

bool runningP = true;

while (runningP) {

std::cout << "============================\n";

std::cout << "Current Public Transit State: " << transit.getCurrentStateName() << "\n";

std::cout << "City Budget: " << formatMoney(transit.getCityBudget()) << "\n";

std::cout << "Public Transit Satisfaction: " << transit.getSatisfaction() << "\n";

std::cout << "============================\n";

std::cout << "Choose an action:\n";

std::cout << "1. Buy Uber ($250k)\n";

std::cout << "2. Buy Bolt ($150k)\n";

std::cout << "3. Exit\n";

std::cout << "Enter your choice: ";

int choiceP;

std::cin >> choiceP;

switch (choiceP) {

case 1:

transit.buyUber();

break;

case 2:

transit.buyBolt();

break;

case 3:

std::cout << "Exiting Public Transit.\n";

runningP = false;

break;

default:

std::cout << "Invalid choice. Please try again.\n";

}

std::cout << "\n";

}

}

int main() {

double budget = 2000000.00;

Road road(budget);

TrainSystem train(budget);

PublicTransit transit(budget);

TransportObserver observer(road, train, transit);

road.attach(&observer);

train.attach(&observer);

transit.attach(&observer);

bool running = true;

while(running){

std::cout << "============================\n";

std::cout << "City Simulation Main Menu\n";

std::cout << "============================\n";

std::cout << "Choose an action:\n";

std::cout << "1. Manage Road\n";

std::cout << "2. Manage Train\n";

std::cout << "3. Manage Public Transit\n";

std::cout << "4. Progress Year\n";

std::cout << "5. View Statistics\n";

std::cout << "6. Exit\n";

std::cout << "Enter your choice: ";

int choice;

std::cin >> choice;

switch (choice) {

case 1:

roadM(road);

break;

case 2:

trainM(train);

break;

case 3:

publicTransitM(transit);

break;

case 4: {

std::cout << "Progressing to the next year...\n\n";

road.progressYear();

train.progressYear();

transit.progressYear();

break;

}

case 5:

observer.update();

std::cout << "============================\n";

std::cout << "City Statistics:\n";

std::cout << "============================\n";

std::cout << "Total Transport Satisfaction: " << observer.getTransportSatisfaction() << "\n";

std::cout << "Total Transport Productivity: " << std::fixed << std::setprecision(1)

<< observer.getTransportProductivity() \* 100 << "%\n";

std::cout << "Total City Budget: " << formatMoney(budget) << "\n";

std::cout << "============================\n\n";

break;

case 6:

std::cout << "Exiting simulation.\n";

running = false;

break;

default:

std::cout << "Invalid choice. Please try again.\n";

}

}

return 0;

}

For now could you please provide me with the big main which makes use of this Transportation system. Do what you think is necessary to get this done (for example, if you believe that making separate classes which are used in the big main is better than one giant main then do so)