Task 1: Build a Menu Navigation System

Objective: Implement a basic menu navigation system similar to what is found in automotive HMI clusters (e.g., settings menu, media options).

Requirements:

Create a hierarchical menu structure using classes and data structures.

Example:

Main Menu

Settings

Display Settings

Audio Settings

Media

Radio

Bluetooth Audio

Use a tree structure (std::vector or custom tree data structure) to represent the hierarchy.

Implement the following functionalities:

Navigate through menu levels (up/down/enter/back).

Display the current menu options on the console.

Add keyboard-based interaction (e.g., 1 to navigate down, 2 to navigate up, 3 to enter).

Deliverables:

A C++ program that uses OOP to create the menu structure.

Console output demonstrating navigation and interactions

Program:

#include <iostream>

#include <vector>

#include <string>

using namespace std;

class MenuItem

{

public:

string name;

vector<MenuItem\*> subMenus;

MenuItem(string itemName) : name(itemName) {}

void addSubMenu(MenuItem\* menuItem)

{

subMenus.push\_back(menuItem);

}

void display()

{

cout << name << endl;

}

};

class MenuSystem

{

private:

vector<MenuItem\*> menuHierarchy;

MenuItem\* currentMenu;

public:

MenuSystem() : currentMenu(nullptr) {}

void addMenu(MenuItem\* menuItem)

{

menuHierarchy.push\_back(menuItem);

}

void displayMenu()

{

cout << "Current Menu: " << currentMenu->name << endl;

for (size\_t i = 0; i < currentMenu->subMenus.size(); ++i)

{

cout << (i + 1) << ". " << currentMenu->subMenus[i]->name << endl;

}

cout << "\nEnter your choice (1 to navigate down, 2 to go back, 3 to enter): ";

}

void navigateDown(int choice)

{

if (choice >= 1 && choice <= currentMenu->subMenus.size())

{

currentMenu = currentMenu->subMenus[choice - 1];

}

else

{

cout << "Invalid choice! Try again." << endl;

}

}

void navigateUp()

{

for (size\_t i = 0; i < menuHierarchy.size(); ++i)

{

for (auto submenu : menuHierarchy[i]->subMenus)

{

if (submenu == currentMenu)

{

currentMenu = menuHierarchy[i];

return;

}

}

}

}

void start()

{

currentMenu = menuHierarchy[0];

while (true)

{

displayMenu();

int choice;

cin >> choice;

switch (choice)

{

case 1:

navigateDown(choice);

break;

case 2:

navigateUp();

break;

case 3:

cout << "You have entered the " << currentMenu->name << " submenu.\n";

break;

default:

cout << "Invalid choice! Please select again.\n";

}

}

}

};

int main()

{

MenuItem\* mainMenu = new MenuItem("Main Menu");

MenuItem\* settingsMenu = new MenuItem("Settings");

MenuItem\* displaySettingsMenu = new MenuItem("Display Settings");

MenuItem\* audioSettingsMenu = new MenuItem("Audio Settings");

MenuItem\* mediaMenu = new MenuItem("Media");

MenuItem\* radioMenu = new MenuItem("Radio");

MenuItem\* bluetoothMenu = new MenuItem("Bluetooth Audio");

mainMenu->addSubMenu(settingsMenu);

mainMenu->addSubMenu(mediaMenu);

settingsMenu->addSubMenu(displaySettingsMenu);

settingsMenu->addSubMenu(audioSettingsMenu);

mediaMenu->addSubMenu(radioMenu);

mediaMenu->addSubMenu(bluetoothMenu);

MenuSystem menuSystem;

menuSystem.addMenu(mainMenu);

menuSystem.start();

return 0;

}

Output:

Current Menu: Main Menu

1. Settings

2. Media

Enter your choice (1 to navigate down, 2 to go back, 3 to enter): 1

Current Menu: Settings

1. Display Settings

2. Audio Settings

Enter your choice (1 to navigate down, 2 to go back, 3 to enter): 1

Current Menu: Display Settings

Enter your choice (1 to navigate down, 2 to go back, 3 to enter): 2

Current Menu: Display Settings

Enter your choice (1 to navigate down, 2 to go back, 3 to enter): 2

Current Menu: Display Settings

Enter your choice (1 to navigate down, 2 to go back, 3 to enter): 3

You have entered the Display Settings submenu.

Current Menu: Display Settings