Telephone Agenda Application

Toasca Adrian

May 2024

Problem Statement

Telephone Agenda Application

- **Strategy:** This is a utility application for managing phone book records with dynamic memmory allocation(8/pointer, 1/character). Users can add, delete, and view contacts.
- **Number of users:** This is a single-user application.
- **Reason:** A better understanding about how dynamic memory allocation works.

Project Description

Implementation of Telephone Agenda:

This project implements a simple telephone agenda (phone book) using C++ and the ImGui library for the graphical user interface. The application allows users to add, delete, and view contacts. The data is managed using dynamic arrays with manual memory management. Key functionalities include:

- **Adding Contacts:** Users can add new contacts with a name and phone number.
- **Deleting Contacts:** Users can delete existing contacts.
- **Viewing Contacts:** Users can view the list of all contacts.
- **Memory Management:** The application dynamically allocates and deallocates memory for contact records.
- **Advanced usage:** If you are running the app from IDE then you can use the advanced informations about the memorry allocated.

Data Structures

Structure	Description
`phone_record`	This structure represents a single phone record. It contains two fields: • `char* name` - A pointer to a character array storing the name of the contact. • `char* phone_number` - A pointer to a character array storing the phone number of the contact.
`phone_book`	This structure represents the phone book. It contains three fields: • `int size` - The current size of the allocated memory for the records. • `int record_count` - The number of records currently stored in the phone book. • `phone_record* records` - A pointer to an array of `phone_record` structures.

Table 1: Data Structures Used in Telephone Agenda

Demonstration

Figure 1: Imgui interface

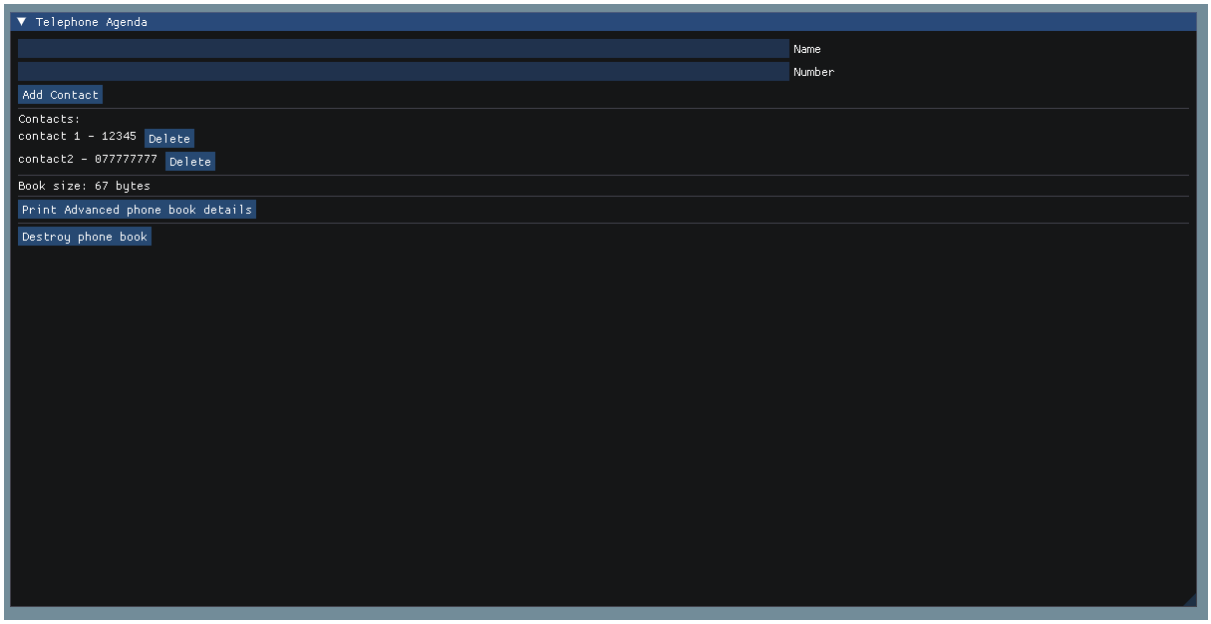

Figure 2: Adding contacts

```
"C:\Users\Adrian\Desktop\uni\Anul 3 Sem. 2\SO_lab\project_SO\cmake-build-debug\project_SO.exe"
allocated_size: 32 = 2 * 16 (sizeof phone_record)
used: 32 = 2 * 16 (sizeof phone_record)
start address: 0000013c638e3ca0
end address: 0000013c638e3cc0
item 1 starts at address 0000013c638e3ca0 ends at 0000013c638e3cb0 name: contact 1 phone: 12345
30(size) = 2 *8 (sizeof char*) + 9 (len("contact 1")) * 1 (sizeof char) + 5 (len("12345")) * 1 (sizeof char)
item 2 starts at address 0000013c638e3cb0 ends at 0000013c638e3cc0 name: contact2 phone: 077777777
33(size) = 2 *8 (sizeof char*) + 8 (len("contact2")) * 1 (sizeof char) + 9 (len("077777777")) * 1 (sizeof char)
```

Figure 3: Advanced details