



Ghulam Ishaq Khan Institute of Engineering Sciences and Technology

CS 221 Fall 23 Lab Project

Team:

Usama Sadiq	2022609
Abdullah Shabbir	2022030
Syed Faiq Haider Naqvi	2022562

Submitted to: Sir Usama Arshad Janjua

Event Simulation System

1. Introduction:

The Event Simulation System is a sophisticated program meticulously crafted to simulate and manage a myriad of events seamlessly. Leveraging a meticulously designed combination of data structures, the system employs a linked list to adeptly manage the event queue, ensuring optimal event sequencing and execution. Additionally, the implementation incorporates the efficiency of an AVL tree for event scheduling, enhancing the overall performance and responsiveness of the system. This comprehensive report endeavors to provide a detailed overview of the code structure, multifaceted functionality, and practical usage of this innovative event simulation system.

2. Code Overview:

The system is implemented in C++ and consists of following classes:

2.1. EventType Class

This class represents predefined event types, storing a type ID and type name.

2.2. Event Class

The **Event** class represents individual events, storing details such as event ID, name, description, type, date and time, location, address, organizer, number of participants, capacity, and status.

2.3. EventNode and EventQueue Classes

These classes implement a linked list to manage the event queue. The queue is used to enqueue and dequeue events.

2.4. AVLNode and AVLTree Classes

The AVL tree is used for efficient event scheduling based on event timestamps. The AVL tree supports operations such as insertion, deletion, and traversal.

2.5. SimulationSystem Class

The **SimulationSystem** class manages the overall simulation, including the event queue, AVL tree, event counters, predefined event types, and simulation menu options.

3. Key Features

3.1. Event Management

- **Event Creation:** Users can create new events by providing details such as name, description, type, date, time, location, address, organizer, number of participants, and capacity.
- **Event Queue:** Events are enqueued into a linked list, representing the order in which they occur.

3.2. AVL Tree for Efficient Scheduling

- **AVL Tree:** An AVL tree is used to efficiently schedule events based on their timestamps.
- **Venue Availability:** The system checks the availability of the venue at the specified time before inserting an event into the AVL tree.

3.3. Simulation Control

- **Menu Interface:** The system provides a menu interface for users to add events, process the next event, display the simulation state, and exit the simulation.

4. Usage

1. **Compile and Run:** Compile the C++ code and run the executable.
2. **Menu Options:** Choose from the following options:
 - **Add Event (Option 1):** Add a new event to the simulation.
 - **Process Next Event (Option 2):** Simulate the processing of the next event.
 - **Display Simulation State (Option 3):** View the current state of the simulation.
 - **Exit (Option 4):** Exit the simulation.

5. Conclusion

In conclusion, the Event Simulation System stands as a robust and flexible solution for the intricate task of managing and simulating diverse events. By judiciously integrating a linked list for the event queue and an AVL tree for efficient event scheduling, the system not only ensures the seamless processing of events but also contributes to the overall organization and optimization of event simulations.

The utilization of a linked list allows for the orderly enqueueing and dequeuing of events, preserving the chronological order of their occurrence. This feature is pivotal in accurately mirroring real-world scenarios where events unfold in a temporally sequential manner.
