

Design Patterns in the AED Software Project

Introduction

The AED (Automated External Defibrillator) software project, primarily encapsulated in the `MainWindow` class, illustrates an effective use of various design patterns. These patterns contribute significantly to the project's structure, making it more maintainable, scalable, and understandable. Key among these are the State and Command design patterns.

State Pattern

The State pattern is employed in managing the different states of the AED simulation. The `MainWindow` class changes its behavior based on its state, which is altered during different scenarios like "good case" and "bad case." These states dictate how the software reacts to user inputs and internal events, closely mirroring how a real-world AED would operate.

Command Pattern

The Command pattern is evident in how actions are queued and executed. Functions such as `goodCase` and `badCase` serve as encapsulated sets of instructions (commands) that the AED needs to perform in respective scenarios. This approach allows for flexibility and easy modification of what each scenario should entail, without changing the underlying execution mechanism.

Design Decisions

The decision to centralize the functionality within a single `MainWindow` class stems from the project's scope and the nature of operations expected of the AED. By encapsulating the functionalities in one class, the design remains straightforward yet effective in handling the various aspects of AED simulation, including UI interactions and operational logic.

Functionality Reliance on `goodCase` and `badCase` Functions

The `goodCase` and `badCase` functions are pivotal in the AED software, simulating different emergency scenarios. They dictate the sequence of actions the AED should perform, effectively guiding the entire simulation process. These functions encompass the core of the AED's operational logic, demonstrating different responses to cardiac events, thus providing a comprehensive training tool.

Importance of Other Functions

- **`updateTimer`**: Essential for maintaining a real-time aspect of the simulation, enhancing the realism of the training tool.
- **`checkPatientResponsiveness` and `callAmbulance`**: Simulate initial response steps, crucial for training users in the protocol of emergency response.
- **`placeElectrodes`, `administerShock`, and `performChestCompressions`**: Core functionalities of an AED, these functions simulate the critical actions taken during cardiac emergencies.
- **`updateBatteryLevel` and `chargeBattery`**: Represent maintenance and operational readiness aspects of the AED, ensuring the trainees understand the importance of device readiness.