

Jared Best, #101188009
Matthew Parsons, #101187027
Alejandro Ferreira, #101232439
Marcus Moquin, #101198659

Our program runs using a timer called mainCount, which increments by 1 every second, up to a maximum of 100. Depending on what the value of mainCount is, the simulator will perform different actions.

When the AED is turned on, it creates a victim to use for the simulation. It then runs a self test to assure that it is properly functioning. If no errors are detected, it declares the unit ok and begins the simulation.

The AED will then show several voice prompts corresponding to actions for the user to take, and at around mainCount = 20, will ask the user to attach defib pads

At maincount = 30, the AED will check the victims age and what pads are attached. If there is a mismatch (adult w child pads or vice versa) the AED will decrement mainCount to 21 and inform the user to check the pads.

Once mainCount = 35, the AED will perform a scan of the victim to determine if a shock is needed. At mainCount = 40, the scan will complete.

If a shot is advised, the AED will say so, advising everyone to back away from the victim. At mainCount = 45 the AED will shock the victim, Then decrement mainCount to 32 to perform another scan.

If no shock is advised, the mainCount will increment to 55 to skip the shock, and inform the user to perform CPR when mainCount = 60. Once mainCount = 80, the AED will tell the user to stop performing CPR, and decrement mainCount to 32 to perform another scan.

After 5 shocks or 5 instances of CPR, the AED will state that an ambulance has arrived, and the victim should be left to them, and shut down.

If the victim's heart rate does not require CPR, then perform another scan. If the victim's heart rate is normal after 5 scans, the simulation will jump to the ambulance arriving, as it is not needed.

The AED also has a battery level that drains over its use. If during a scan or shock, falls below 35% or 40% respectively, it will inform the user to change its batteries. Performing a shock drains some of the battery from the AED, and it also slowly drains overtime.

We decided to have mainCount run via a timer because the CPR is supposed to take 20 seconds according to the rubric, and C++ does not have a very reliable way to pause it's runtime for a certain period of time, so we decided to use QTimer to act as our timer by sending a signal every second, which is received by a slot which increments mainCount by 1. The use of the timer also allows us to refresh the display window every second, changing things like the battery level and current voice prompt as needed.

Whenever the AED decides to shock the victim, it calls the successofShock() function, then increases the shock count by 1. This function starts by creating 3 random numbers, from 1

to 4, 1 to 3, and 1 to 2 respectively. Depending on how many times the victim has been shocked already, it will choose one of the three numbers.

If the shock count is 0, it will use the first value, checking if it is equal to 1, a 25% chance.

If the shock count is 1, it uses the second value, which has a 33% chance of success.

If the shock count is 2 or 3, it uses the third value, which has a 50% chance of success.

If the shock count is 4, it is a guaranteed success.

NOTE: if there are any issues with compilation, run qMake first