Electrocardiogram Rhythm Simulation in Open Source Environment
- a contribution to training in biomedical sciences
A Cardoso Martins, P Dias Costa, J Miguel Marques, R Cruz Correia
Department of Biostatistics and Medical Informatics
Faculty of Medicine - University of Porto, Portugal

Abstract
Considering the importance of training and the need to identify some important electrocardiographic patterns, our aim was to develop a software-based electrocardiogram rhythm simulator, able to generate signals for purposes of training health care professionals, freely available. Additionally we intended to implement a training assessment and the ability to configure specific rhythms/patterns with user-defined settings.

In order to achieve the availability requirement we decided to develop our tool as open source. All the software was developed in Linux platform, but releases are available in binary format for Linux and Microsoft Windows platform.

The result was a functional signal simulator which satisfied all the requirements we proposed in our aim. The final result of our work is presented and available at Sourceforge.

Limitations and future perspectives are also addressed in this paper.

Keywords
Electrocardiogram; ECG; Training; Software; Open Source; Computer simulation.

Address for Correspondence
Prof. Ricardo Cruz-Correia
Faculdade de Medicina da Universidade do Porto
Alameda Prof. Hernâni Monteiro
4200 - 319 Porto, Portugal
Phone: +351 225 513 600
E-mail: rcorreia@med.up.pt
Conflict of Interest
None to declare.

Disclaimer
This project was entirely developed using Open Source software and no copyright laws were violated.

Introduction
Electrocardiographic signal is a graphic representation of the electrical activation (depolarization) and repolarization of cardiac cells, as a result of the expression of main activation vector in each lead. In an electrocardiogram (ECG) strip it is possible to identify several incidents, that concern specific electrical activity. For example, the P wave is a result of the depolarization of both atria and the QRS complex a result of depolarization of both the interventricular septum, right ventricle, and left ventricle. Another incident in the ECG is the T wave, that concerns not the depolarization but rather the repolarization (a “resting” phase in which cells are “recharging” for a new depolarization) of cardiac cells. Additionally it is possible to define a number of intervals and segments between these main incidents, that concern stimuli conduction and that do have clinical correspondence. Also, all of these features have a specific “normal” duration, amplitude and morphology\textsuperscript{1,2}.

Presently, and more than 100 years passed over its introduction in clinical practice by Willem Einthoven\textsuperscript{3}, the ECG is still a valuable diagnostic instrument, in particular in the non-invasive diagnosis of arrhythmias and conduction disturbances. The astonishing technological evolution felt in medical areas in the past 50 years, with particular relevance in Cardiology, created the misconception that newer techniques have replaced electrocardiography. The fact is that the ECG is needed to deliver modern medical care now more than ever, and the correct diagnosis and treatment of arrhythmias strongly depends on the interpretation of the ECG. Therefore, not only is the electrocardiogram an important diagnostic tool, it is used to guide the treatment of common severe cardiac conditions, and to determine the prognosis of patients\textsuperscript{4}.

Training of biomedical professionals is crucial for patient care. The ability to train
procedures, establish diagnosis and define prognosis of a disease prior to applying that knowledge in real-life patients is an enormous advantage for both professionals and patients, with clinical, legal, and ethical implications. In terms of execution electrocardiography is a rather simple technique, but one of the most difficult techniques when it comes to interpretation. The wide variety of normal patterns, sometimes quite different from each other, and the difficulty to identify some subtle abnormal patterns makes prior training not only necessary but also imperative. Recently a committee of the American College of Cardiology, the American Heart Association, and the American College of Physicians/American Society of Internal Medicine came together to develop guidelines for the training in electrocardiography\textsuperscript{5}.

A wide variety of ECG simulators, electronic or software-based, both commercial and non-commercial, are currently available. A series of commercial devices, using just an electronic approach, were built especially for ECG rhythm simulation and training. As electrocardiography itself had been steadily developing, electronic ECG simulators were unable to follow that development, becoming obsolete in specific and advanced rhythm simulations. Nevertheless they are currently in use, but only for specific training. With the explosion of computer technology back in the 1970's and 1980's several computer programs\textsuperscript{6,7} were developed for ECG and pacing rhythm training and simulation. Those systems seem to have been abandoned (at least for what concerns free software implementations), with a few exceptions. Nevertheless some of these programs have severe limitations namely on scientific contents and basis. Conversely, a series of well developed ECG rhythm simulation programs is currently available for commercial use. Additionally, an accounted number of advanced life support (ALS) manikins are also available, allowing the chance to diagnosis and treat malignant arrhythmias\textsuperscript{8-10}. However these training manikins are somewhat limited in their ability to deliver and simulate certain electrocardiographic patterns as they are more emergency training oriented.

As far as we know, currently there is no ECG simulation tool widely available to everyone, able to receive suggestions and improvements from a wide variety of professionals of the medical and other professional areas, that can be freely
improved and reviewed by experts. As stated before, some exceptions are available, but with limitations, like the requirement of a third party software license (Matlab) or lacking usability for everyday usage.

Considering the importance of training and the need to identify some important electrocardiographic patterns, our aim was to develop a software-based ECG rhythm simulator, able to generate ECG signals for purposes of training health care professionals, freely available. Additionally we intended to implement a training assessment and the ability to configure specific rhythms/patterns with user-defined settings. In order to achieve the availability requirement we decided to develop our tool as open source, therefore we have chosen the GNU Public License GPL v3 license model.

Methods
This project was part of an assignment for the Health Information Systems course as part of the Masters Course in Medical Informatics. António Cardoso Martins was responsible for all the programming in the project. Paulo Dias Costa was responsible for the scientific contents and layout. João Miguel Marques was responsible for the design and graphics of the program. All authors were actively involved in compiling, testing, and dissemination the project.

We performed a search in the Pubmed reference database regardless of the type of publication. For addressing the database, we used the query: electrocardiogram [MeSH] AND training [MeSH] AND simulator with limits selected for 'English language' and 'Humans'. The query was performed on the 15th May 2009 and returned 6 articles for analysis. The analysis was performed by all investigators. Selection criteria were defined prior to the query and included: 1) type of study; 2) critical analysis of the results, and 3) suitability to our subject. After applying the above criteria, we selected two articles. This method, despite its scientific reliability, does not take the existence of other not so formal initiatives into account.

To understand the point of development of commercial and non-commercial ECG simulation tools we performed a qualitative survey using the same terms on Google search engine. The result, with the limitations of being a qualitative survey, made clear that a wide variety of tools is available but not with the type of
Results

Implementation
As the first task of the project, we tried to find a simple and flexible way to generate signals that could be similar to those of human ECG with different patterns. Fourier series were selected for representing ECG signals, as any periodic functions which satisfy Dirichlet's condition can be expressed as a series of scaled magnitudes of sin and cos terms of frequencies which occur as a multiple of fundamental frequency. ECG signal is periodic with fundamental frequency determined by the heat beat\textsuperscript{11}. Each significant feature of ECG signal can be represented by shifted and scaled versions of waveforms.

Environment
The Qt 4 development framework was chosen to implement the project, as it is very flexible, user C++ as the programming language, has an easy to use graphical user interface support, and is available in many different operating system platforms. Another reason, was that Qt\textsuperscript{14} version 4 has a licensing model that allows the use of GPL to accomplish an open source software solution. The software was developed using many tools, however the most important ones were Qt Creator\textsuperscript{15} as the integrated development environment (IDE), and Sourceforge\textsuperscript{16} as the project management system that provided a complete software build platform, that includes svn for source code repository, package release file system repository, mailing lists and forum for developer discussion, bug and feature tracking system, and a project web page. All these resources are publicly available in the Sourceforge simECG project pages\textsuperscript{17}. 
Programming

The architecture of the project lead to main C++ classes that implemented the intended block behaviour.

ECG presets

Implemented as ECGmemory class, it saves ECG settings in one object. This class is used to store all the ECG presets, that are called later as a result of the user selecting it from a presets tab, or during an assessment. This class also supports the settings that are displayed by the signal plotter in a given moment.

ECG generator

Implemented as ECGPlotter class, it is the main responsible for the realization of the Fourier series that simulate the ECG signal. This class has methods for the generation of each of the ECG signal components, the P, QRS, S, T and U waves, that can be applied with flexible values. Each wave calculation can be skipped in order no to generate some components in the final signal.

Signal plotter

Implemented as Plotter class, it is responsible for drawing the data vector.
generated by the ECGPlotter. The information is drawn in a scaled grid background appropriate for ECG reading, and similar to the ECG paper used in many medical devices.

**Assessment**

Implemented as AssessmentFrame class, it drives all the assessment logic. In a set of 10 questions, generates three questions randomly from the preset signals, takes one as the correct and displays it. If the user answers correctly, the score is added and another question raised during a 60 second period. After all the questions answered or the time expired, the score is presented to the user. The score is calculated multiplying the number of correct answers by the time remaining.

**Graphical User Interface**

The myMainWindow class, implements the main program window, menu toolbar, docking toolbar with a tab separator, and all the graphical widgets present. It is the class that glues all the code, mainly through the concept of signals and slots, that is very well known in Qt development.

The main window is divided in a menu toolbar, an ECG signal plotter, and a dockable toolbar with tab separators.

The menu toolbar comprises 'File', 'Preferences', and 'Help'. In the 'File' menu is possible to save and load preferences and custom settings. The 'Exit' option is also in this menu. In the 'Preferences' menu we implemented the technical specifications of the signal, such as 'Calibration' (in which the user can set the signal "Speed" and "Amplitude") and 'Filters' (set On or Off). Also in this menu we placed the user interactive signal display options such as 'Display' and 'Background' in which the user can select the 'Static' or 'Rolling' display, or choose between two possible background fillings. The 'Help' menu displays information on the developers, the license, and a disclaimer.

The ECG signal plotter displays the preset or custom settings the user chose. Also, it displays the selected heart rate in the top right corner.

The dockable toolbar is the functional core of simECG and comprises 'Preset', 'Custom settings', and 'Assessment' tabs. This toolbar has additionally a feature that allows it to be drag, therefore making possible users to use two different
monitors, changing the signal in one of the monitors and displaying in the other. The 'Preset' tab we placed some standard rhythm strips (e.g. sinus rhythm and supraventricular tachycardia) which the user can use to identify the usual pattern of such a rhythm. In the 'Custom settings' it is possible change several characteristics of the signal, therefore allowing the user to reconstruct almost every signal he intends. Characteristics such as heart rate, P wave and QRS complex amplitude, duration and morphology, and T wave settings are possible to change here. The introduction of premature beats, and special conditions is also placed on this tab. The 'Assessment' tab allows users to test their knowledge, using friendly interface and classification.

Nevertheless, some features are not currently implemented in this version, and some bugs still persist (see Conclusion and Future Perspectives).

**Discussion and Conclusion**

The simECG is able, as stated in our aim, to generate ECG signals, for purposes of training health care professionals as an open source solution, freely available to the community from now on.

The final result of our work is presented and available at Sourceforge\textsuperscript{16} in the simECG project page\textsuperscript{17} as version 1.0. All the software was developed in Linux platform, but releases are available in binary format for Linux and Microsoft Windows platform.

This project required many hours of research, discussion, and programming from all its authors. Sometimes the amount of work, or difficulty, was such that quitting seemed to be a good idea. Regardless of that, this was certainly one of the projects that the authors appreciated the most in great extent for the challenge it posed. With the writing of this manuscript, in Open Source format, the authors put in practical application their ideal that education (and training, in this case) should not be a commercial tool accessible to some but rather a free collaborative tool available for those who seek knowledge.

**Future Perspectives**

Nevertheless this project has still some limitations. Some special conditions, such as type I and type II second degree atrio-ventricular (AV) block, third degree AV block, and AV dissociation were not implemented due to time constraints. For that
same reason, it was not possible to implement the atrial fibrillation pattern. The implementation of ECG motion (rolling) needed further advanced research and therefore not implemented.

In future versions the authors intend to provide additional features in the training presets, include sino-atrial and AV blocks, and different atrial arrhythmias such as atrial fibrillation and flutter.

All pending bugs and requested features are described in detail in the Sourceforge tracker service.

Acknowledgements
The authors wish to thank R. Karthik, B.E., for providing the ECG simulation and computer programming that supported the initial phase of this project. The authors also wish to thank Sandra Jardim, B.Sc., for her enormous help with all the mathematical-related subjects in our work. Also, we express our gratitude to Ernesto Aranda, B.Sc., for the splash screen design. To all of those anonymous helpers that graciously provided their opinion in Sourceforge for improving this work our sincere acknowledgement.

References


