Virtual Reality Physio Monitor

BME 300/200
University of Wisconsin-Madison
Department of Biomedical Engineering
October 26, 2011

Team:
Hope Marshall – *Team Leader*
Roland Pomfret – *Communicator*
Gabriel Bautista – *BSAC*
Jiaquan Yu – *BWIG*

Clients:
Professor Robert Radwin-Department of Biomedical Engineering
Dr. Benjamin Mandel-Department of Surgery

Advisor:
Professor John Webster
University of Wisconsin-Madison Department of Biomedical Engineering

Table of Contents

Abstract……………………………………………………………………………………….3
Background…………………………………………………………………………...…....3
Motivation …………………………………………………………………………......….....4
Problem Statement………………………………………………………......………….....5
Client Requirements……………………………………………………………………….5
Existing Devices…………………………………………..……………………………....5
Ethics ………………………………………………………………………………..…....5
Ergonomics……………………………………………………………………..………....6
Design Proposal Overview……………………………………………..………………..….6
Alternative Design 1: MATLAB……..……………………………………………….…6
Alternative Design 2: C++……….………………………………………….……………..6
Alternative Design 3: Java..…………..…………………………….………………..7
Design Criteria…….………………………………...……………………………….......7
Final Design……..…………………...………………………………………………….…8
Design Development………………………….………………..…………………………..9
Testing……………………………………….…………………….…………………..........11
Quantitative Testing…………………………………………...……………...……11
Qualitative Testing………………………………………………………………………12
Future Work…………………………….………………..…………………….………......12
Works Cited……………………………….………………………..…………..……………..14
Appendix A: Product Design Specification Report…………………………...……………..15
Appendix B: Original User Interface and Monitor Design……………………………………...18
Appendix C: Final Flowchart………………..……………………………………………..19
Appendix D: Java Class Diagram………………………………………………………….22
Appendix E: Guide to Creating a New Scenario (Text File)……………………………………23
Appendix F: Tension Pneumothorax Text File…………………………………………………27
Appendix G: Qualitative Testing Responses……………………………..…………………..28
Appendix H: Quantitative Testing Data………………………………………………………30
Appendix I: Virtual Physio Monitor……………………………………………………….32

Abstract

Nurse education currently consists of theoretical learning and practice with patient models. In the ER Trauma Bay simulation in the Cave Automatic Virtual Environment (CAVE) at the Wisconsin Institutes of Discovery, one goal is to allow nurses to practice proper skills and techniques in a realistic scenario. Professor Robert Radwin and Dr. Benjamin Mandel are working with the CAVE to produce an ER tension pneumothorax (severe oxygen shortage and low blood pressure, progressing to cardiac arrest) scenario[1]. Currently, the vitals monitor in the simulation displays vitals that do not change. In order to simulate a real-world clinical environment for training purposes, a dynamic virtual monitor display system must be created for the CAVE. The monitor should display electrocardiographs (EKGs or ECGs), respiration rate, blood pressure and *S*pO2 levels that respond to changing physiological conditions in the virtual ER trauma bay. In order to complete the monitor design, three alternative programming languages were evaluated. The programming languages examined were Java, MATLAB, and C++. Based on chosen design criteria, the advantages of Java significantly outweighed the advantages of the alternative programming languages. Currently, the team has implemented a user interface for the monitor, which includes an ECG signal data and graph, a photoplethysmograph (PPG), *S*p02, respiration numerical readings and graphical representation, temperature, blood pressure and patient condition descriptions and images. The team has developed an efficient way to create and load scenarios onto the monitor. The tension pneumothorax case has been researched and implemented so that the monitor responds to vitals changes and nurse actions for the tension pneumothorax scenario in the CAVE. The team has tested the monitor with students in the nursing program and has received approval from the clients. Although the generated vitals could be fine-tuned, the program is ready for implementation into the CAVE.

Background

The CAVE (Figure 1) is a 10 x 10 x 10 ft. cube capable of rendering 3-dimensional virtual environments. Each side of the cube has two 1080p projectors which allow a given simulation to be displayed in high-definition. Audio in the CAVE is provided by a full surround sound system. The environment is viewed by wearing a special set of 3-D glasses, specially equipped with infrared sensors, which determine the direction the user is facing and adjust the image accordingly. The user interacts with the environment through a handheld device that allows him/her to point at certain objects in the simulation and move or use the object. Simulations for the CAVE are written in the CAVELib programming language (Radwin, Interview).

The scenario being simulated involves a patient suffering from tension pneumothorax (Figure 2), commonly known as a collapsed lung. In the simulation, the user must follow a specific set of steps in order to bring the virtual patient to a stable condition. One element of the simulation is the physio monitor (Figure 3), or vitals monitor. The current monitor in the

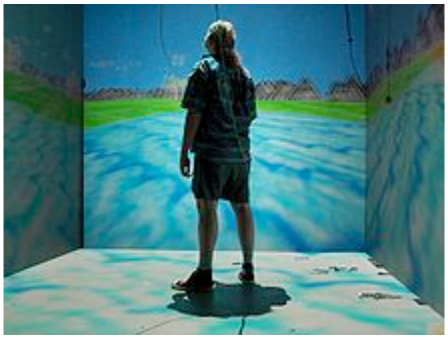


Figure 1: Image of a CAVE

http://en.wikipedia.org/wiki/Cave_Automatic_Virtual_Environment

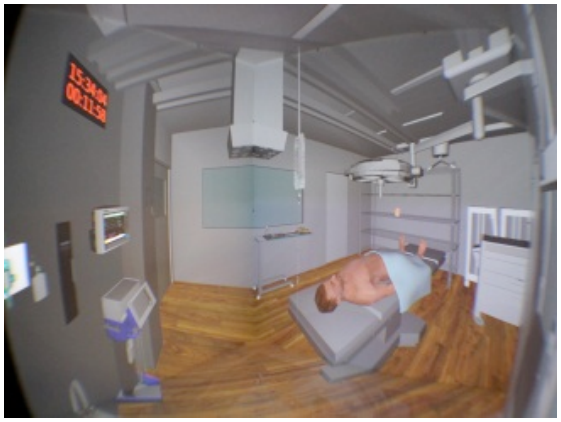


Figure 2: Tension pneumothorax scenario

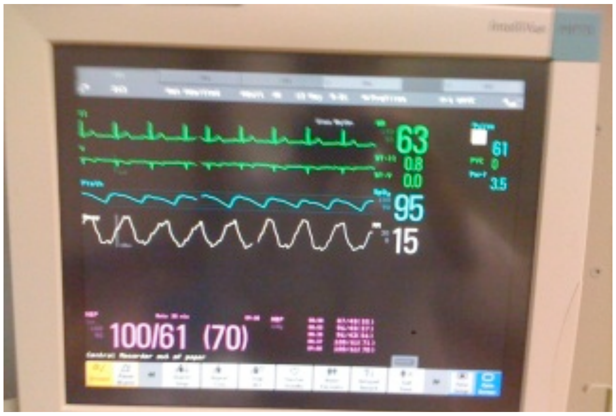


Figure 3: Philips Intellivue MP70

simulation is unrealistic in that it picks static values for all vitals. A more realistic monitor is necessary in order to bring the simulation to the level of realism required in order to train nurses.

ERs are regularly equipped with physio monitors, although the types and brands vary widely. The clients have asked that the Philips Intellivue MP70 be used as a model for the monitor. Physio monitors can display many different readings, including but not limited to: heart rate, respiration rate, blood oxygen levels, carbon dioxide blood levels, blood pressure, and temperature.

Tension pneumothorax occurs when air accumulates in the chest cavity. This places pressure on the lungs, inhibiting their ability to expand, resulting in the patient experiencing symptoms such as shortness of breath, chest pain, fatigue, and a blue hue to the skin (Jain et al 2011). Air can enter the chest cavity for numerous reasons such as chest trauma from a car accident. This can also be caused by outdoor activities such as scuba diving or high-altitude climbing (Collapsed Lung 2011). From a vitals perspective, heart rate and respiration rate increase, while blood pressure and blood oxygen decrease (Mandel, Interview). Tension pneumothorax is typically identified using an x-ray. However, there is usually not enough time to wait for these results in an emergency situation. If tension pneumothorax is suspected, prompt treatment with a needle and chest tube is performed, which allows air to escape from the chest cavity, returning the patient to a stable condition (Chen et al. 2011).

Motivation

The tension pneumothorax 3-D scenario is being developed for nurse training. It is important for nurses to be trained in real cases and the CAVE is capable of simulating a realistic trauma scenario. Currently, nurse education consists of theoretical learning and practice with patient models. Neither of these methods provides an experience that is close to a real ER situation. The goal of the CAVE program is to allow nurses to practice proper skills and techniques in a realistic scenario.

In order to simulate real conditions in the ER trauma bay, the patient monitor must be displayed clearly and it should be identical to a real one both in appearance and function. Specifically, the monitor should display dynamic signals that change according to the patient's physiological conditions and nurses' reactions.

Problem Statement

The clients, Professor Robert Radwin and Dr. Benjamin Mandel, are working with the CAVE to produce an ER scenario that, in the future, will be used to train nurses. Currently, the vitals monitor in the simulation displays vitals that do not change. In order to simulate a real-world clinical environment for training purposes, a dynamic virtual monitor display system must be created for the CAVE. The monitor should display dynamic EKGs, respiration rate, blood pressure and $S\text{p}O_2$ levels that respond to changing physiological conditions in the virtual ER trauma bay. These levels should also fluctuate based on general variability in humans. The virtual display must sound an alarm if it detects that the virtual patient's vitals are passing a dangerous threshold.

Client Requirements

The final monitor design must resemble a monitor used to train medical students. In the tension pneumothorax case, the monitor must simulate vitals that respond to the series of steps detailed by the flow chart provided by the clients. The monitor must display at least four dynamic vitals: heart rate, blood pressure, $S\text{p}O_2$, and respiration rate. The signals should look like the filtered signals commonly displayed by modern monitors in the ER. The vitals displayed must change based on user action, but they should also fluctuate based on general variance among the human race. In addition, the program must be flexible to allow for modification and implementation of other patient ailments in the future. The design should be based on the Philips Intellivue MP70. The program should also have the option for the user to enter vitals manually. Since this is a training program, a system should be in place to notify the user if the patient's vitals are declining at dangerous rates. This is meant to attract the user's attention to the monitor in order to properly address the situation.

Existing Designs

There are programs in existence that simulate patient vitals, but many of these programs are specific to certain scenarios. For example, the ANGIO Mentor product line from Simbionix is a series of hands-on simulators for cardiovascular surgeries. The user performs actions with their hands, such as inserting an endovascular balloon, that are reflected on a computer generated image of the heart. Simulated patient vitals appear on a large screen next to this image (ANGIO Mentor et al 2011).

Anesoft, a medical software company, has numerous programs available that simulate various medical emergencies in areas such as anesthesiology, Advanced Cardiovascular Life Support, and even bioterrorism (Anesoft 2011). These programs include patient vitals simulators tied to the specific scenario the program is meant for.

Neither of these options, nor any other existing products found, have tension pneumothorax scenarios or are compatible with the CAVE.

Ethics

It is important to make this monitor simulation as realistic as possible so nurses who train in the CAVE will be better prepared for real surgery. If the simulated monitor is unrealistic, when it comes time for the user to perform in a real ER situation, wrong diagnoses may result which could cause harm to or death of the patient. Even if a realistic simulation is created, the user must be aware that it is merely a simulation. After practice in a CAVE scenario, the user should by no means consider his or her self ready to operate on a real person where a real life is at risk. In addition, any signal data acquired for programming use from real people will have to be made anonymous. Any programming assistance received will have to be properly documented and cited.

Ergonomics

Since the user does not physically interact with the monitor, the only relevant ergonomics are the aesthetics of the monitor and the audio it produces. The monitor will be designed after a Philips Intellivue MP70 image provided by the clients. The visual and audio components of this monitor should provide the user with an easy transition from the CAVE to the ER. Audio cues for heart rate and to indicate failing vitals must be implemented to attract the user's attention.

Design Proposal Overview:

The nature of the project is programming-oriented, so the alternative designs represent different programming languages. The clients have specific requirements for how the monitor should look and function, which leaves little flexibility for creativity in designing the monitor. The three alternative designs are described below.

Alternative Design 1-MATLAB

The first alternative design is the programming language MATLAB, which is intended for advanced numerical computing. The most notable feature of MATLAB is that each element is based on matrices. The advantages of MATLAB are that it is available as tethered software and the advanced numerical computing will be useful in simulating vitals (Mathworks). In addition, a realistic EKG simulator is available as a MATLAB program. MATLAB has good options for graphics, but the advanced graphics options are used primarily for mathematical applications (Mathworks). A disadvantage of MATLAB is that only a limited number of functions support multithreading, which means that multiple parts of the program can't be running at once (Mathworks). For example, in the program, each digital and graphical representation of the vitals will represent a separate thread. Since MATLAB doesn't have a multithreading option for the types of functions we would be using, the vitals would not update concurrently. For the program to run effectively, all vitals must update independently and concurrently.

Figure 4: MATLAB logo

http://media.photobucket.com/image/matlab+logo+/cacing_95/logo.jpg

Alternative Design 2-C++

The second alternative design is the programming language C++, which stems from the language C. C++ is an object-oriented version of C (Stroustrup). The advantages of C++ include that it uses less memory, and therefore is a fast language. In addition, C++ is widely used so many people have the skills to add on or modify the program. A disadvantage of C++ is that the language is more advanced and complex than languages that the team has experience with (Morris). Also, to program the necessary graphics, the team may need to purchase additional software for Graphical User Interface (GUI) applications. C++ has a limited standard library

Figure 5: C++ logo

http://www.pgpnet.dk/?Programming

compared to Java, especially concerning graphics, which means there aren't as many *built-in* options for graphics (Programmers Stack Exchange).

Alternative Design 3-Java

The third alternative design is the programming language Java. Although Java is capable of robust applications, it is basic enough that it is regarded as a beginner language. Java is multithreaded, so multiple aspects of the program can update independently and concurrently. Java is the most accessible of the three alternative designs, as it runs on Eclipse, which is free and platform independent. In addition, an EKG simulator is available in Java. A disadvantage of Java is that it is slower than C++ and MATLAB and uses more memory (Jelovic). Therefore, the vital signals may not update as quickly as would be necessary.

Figure 6: Java logo

http://geektech.in/archives/1865

Design Criteria:

Design criteria were determined based on the goal to create a monitor that meets client requirements. Figure 7 lists evaluation criteria for the alternative programming languages. The categories that were used to evaluate the designs are Universality, Accessibility and Cost, Ease of Use for Programming, and Graphics. The three designs were given a score for each category and then the total score was summed (Figure 8). The Universality category evaluates how common the programming language is. The more common the programming language is, the more likely a programmer will be able to implement new trauma scenarios in the future. Accessibility and Cost is important as the program for the monitor isn't copyrighted or protected. There is potential for other uses for the monitor outside of the uses for the CAVE in training, teaching or outreach. Ease of Use for Programming is the most heavily weighted category. As this design project involves relatively complex programming, the ability for the team to satisfactorily complete the project is contingent on programming skills and experience. The final category, Graphics, is significant because the virtual monitor must look realistic and the graphs must update precisely.

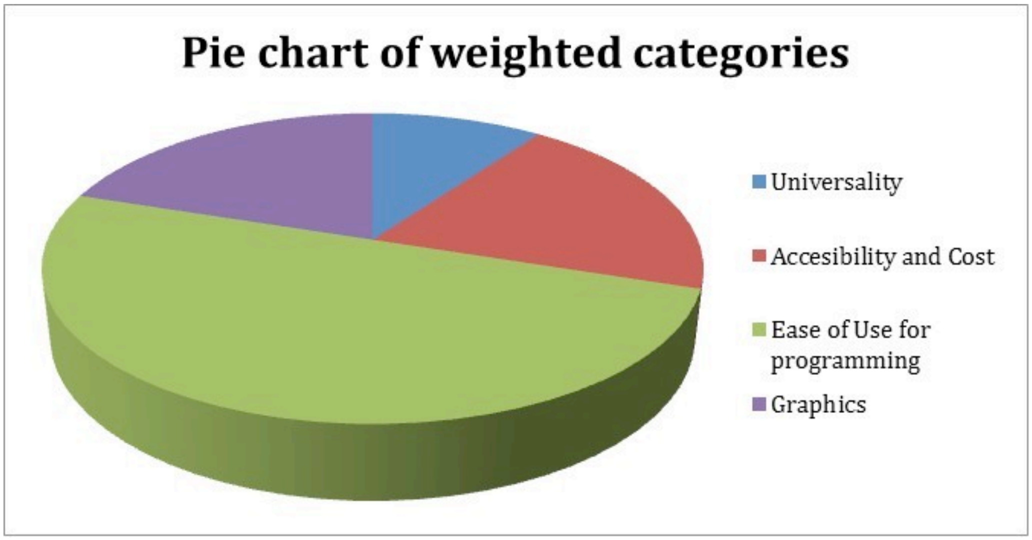
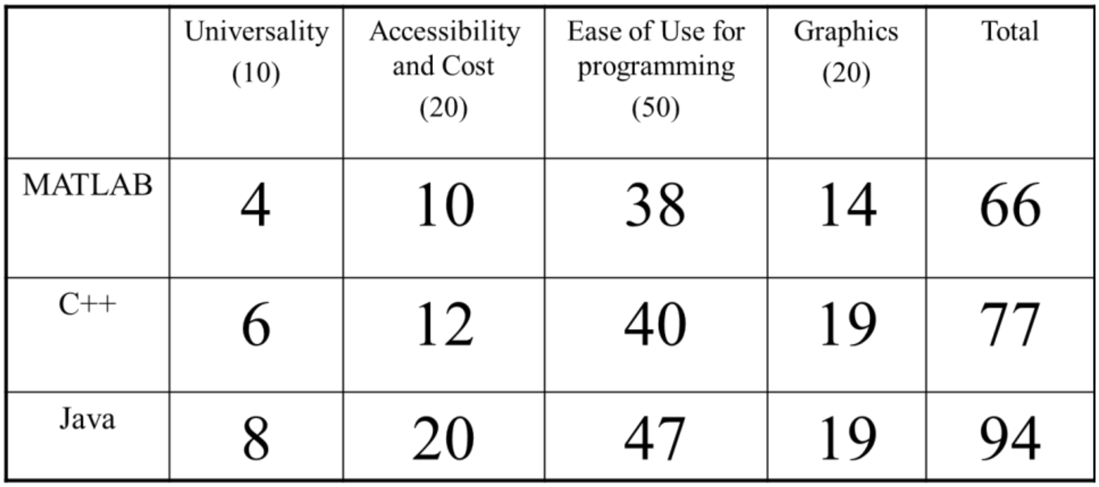


Figure 7: Pie chart of weighted categories



	Universality (10)	Accessibility and Cost (20)	Ease of Use for programming (50)	Graphics (20)	Total
MATLAB	4	10	38	14	66
C++	6	12	40	19	77
Java	8	20	47	19	94

Figure 8: Decision Matrix

For the Ease of Use for Programming category, MATLAB scored the lowest because only one team member has had experience programming in MATLAB. C++ has similar syntax to Java, so it scored higher than MATLAB in this category. Java scored the highest since two team members have experience programming in Java and it is regarded as a beginner language.

MATLAB is extremely costly and therefore not accessible to the public, so it scored the lowest in the Accessibility and Cost category. In contrast, Java is free and platform independent, so it got a perfect score in this category.

After analyzing the Universality of each alternative design, MATLAB scored the lowest because it is the least common programming language out of the three chosen languages. Java scored the highest as it is the most commonly used programming language in comparison to the other two.

In researching the quality of graphics for each programming language, it was found that MATLAB has the fewest options for graphics while Java and C++ both have a large variety of options (Mathworks). Overall, MATLAB scored 66 points, C++ scored 77 points and Java scored 94 points. Evidently, Java is the strongest alternative design and was chosen as the final design in programming the Virtual Reality Physio Monitor.

Final Design:

The final design will utilize a combination of Java and MATLAB in order to create a realistic monitor. Since the CAVE software cannot be run on most computers, a GUI will be created in Java to "simulate" the CAVE environment. This GUI (Figure 9) will display patient vitals, pictures of the scenario and a text description of the patient's condition. It will also accept input via the "Actions" menu so that users can navigate through each step of the procedure. Vital signals will be simulated using MATLAB and these values will be stored in text files. The text files will then be read by the GUI software and stored in array lists. Various "Updater" and "Grapher" objects which run concurrently in the program will utilize the values in the array lists along with random number generators and user input to determine the signals the monitor needs to display. All of these values will have thresholds to make sure they do not stray from the specific simulated range. In order to efficiently add new scenarios to the software, an "Ailment" interface has been created. For the program to run, new scenario objects must adhere to the methods defined in the "Ailment" interface. Finally, the program will be translated into CAVELib in order to be implemented in the CAVE.

Design Development:

Heart rate, respiration rate, and the PPG all have corresponding waveforms on the monitor. In order to generate these vitals, mathematical equations were used to produce realistic waves in the simulation. To produce the ECG wave, a set of equations in a program called ECGSYN was used (McSharry et al 2003). The output of the program is best described by Figures 9 and 10 below. ECGSYN loops a set of equations together in order to create a repeating and realistic heart rate wave.

The PPG and the respiration waves were modeled after Figures 11, 12, 13 and 14 using a combination of sinusoidal functions, as seen below. The complexity of these functions comes from making the waves simulate real vitals. One particularly important aspect is the random Gaussian number generation. This is what allowed variability to exist in the program. In the equations shown in Figures 15 and 16, the Gaussian number is multiplied by a constant, which fits the random number to the scale of the vital in question. This allows the program to maintain variability without the vital signs becoming highly erratic. All waveforms update one cycle at a time and are then displayed on the monitor at a rate of 100Hz.

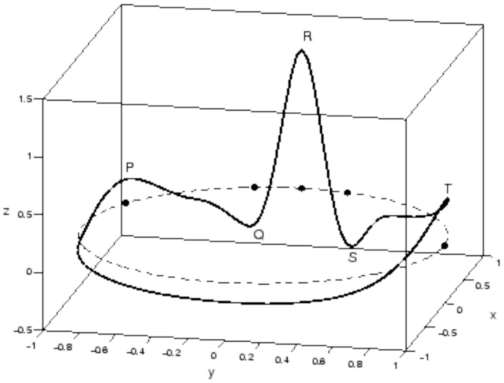


Figure 9: Wave generated by ECGSYN in 3D space (Program one is at 94 bpm)

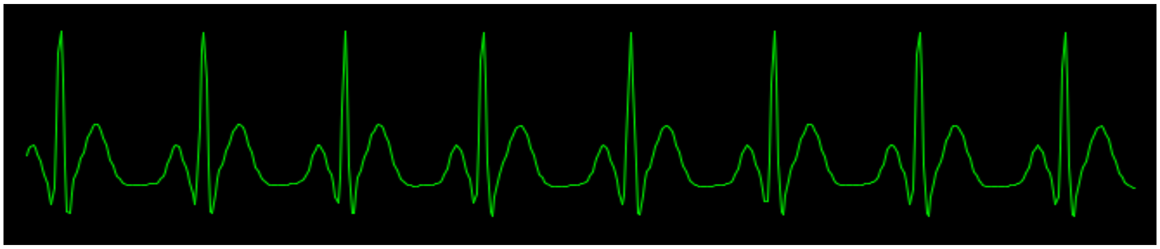


Figure 10: ECG waveform generated by ECGSYN in 2D space

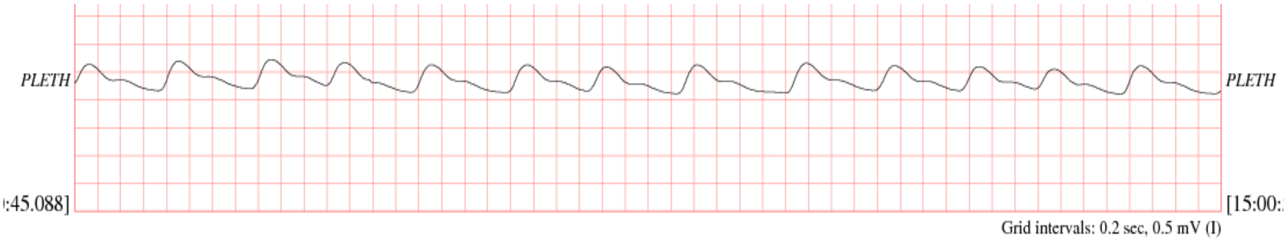


Figure 11: Photoplethysmograph Selected Input: record mimic2wdb/36/3600087/3600087, from [15:00:45.088] to [15:00:55.088]

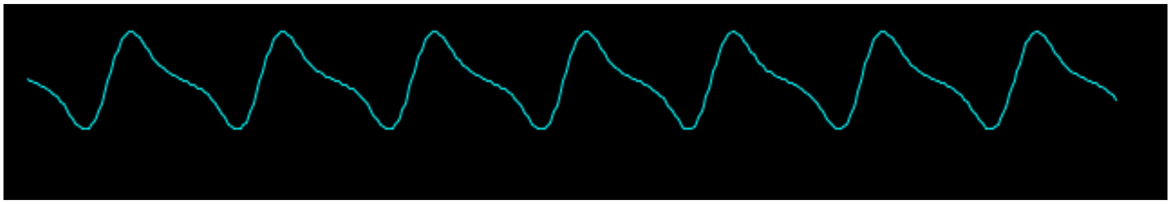


Figure 12: Photoplethysmograph generated by the program using sinusoidal functions.

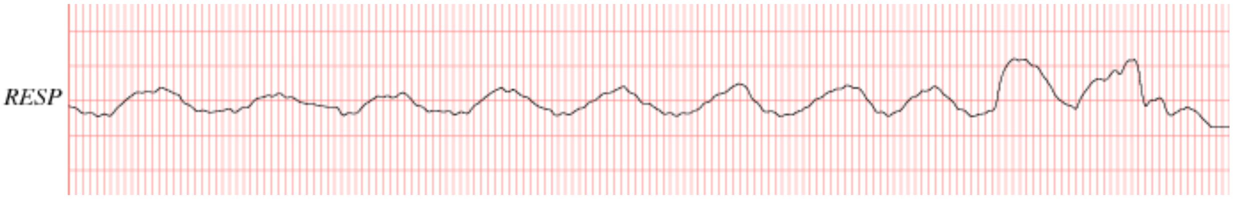


Figure 13: Respiration Selected input: record mimic2wdb/36/3600087/3600087, from [14:39:25.088 1] to [14:40:25.088 1]

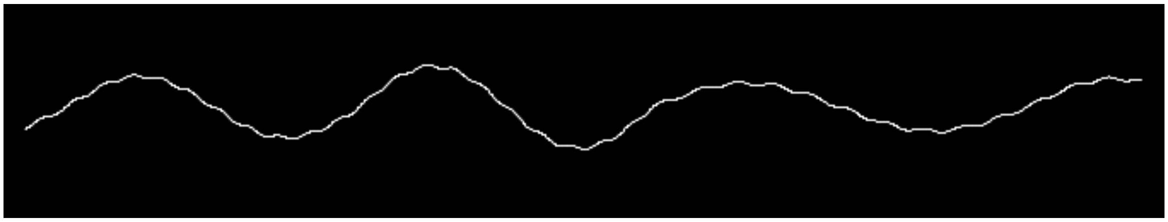


Figure 14: Respiration wave in program at 20 b/m

$$
\begin{gathered}
Pleth = 1 * \sin(t\omega) + .5 * \sin(2 * t\omega) + .1 * \sin(3 * t\omega) \\
\omega = ((HeartRate + HeartRateStandardDeviation * y) * 2\pi)/60 \\
0 \leq t \leq (Length\ of\ ECG\ waveform\ cycle - 1)/100 \\
y = Random\ number\ from\ Gaussian\ distribution\ with\ \sigma^2 = 1.0\ and\ \mu = 0
\end{gathered}
$$

Figure 15: Plethysmograph equations

$$
\begin{gathered}
Resp = a * \sin(t\omega) + .02 * \sin(10 * t\omega) + 1 \\
\omega = ((RespirationRate + RespirationRateStandardDeviation * y) * 2\pi)/60 \\
0 \leq t \leq (3600\omega/2\pi) \\
a = .3 + .075 * z \\
y, z = Random\ number\ from\ Gaussian\ distribution\ with \\
\sigma^2 = 1.0\ and\ \mu = 0
\end{gathered}
$$

Figure 16: Respiration equations

Testing:

Quantitative Testing

The program was run successively using different values for vital signs in the default setting and changes in vital signs for specific actions in the tension pneumothorax scenario. It was noticed that the rates of change for the heart rate and respiration rate given as input from the tension pneumothorax text file were not consistent with the output rates of change in the program. After each respiration or heartbeat output cycle, the program compares the current output rates to the change for the given action and adjusts the output rates according to the input rate of change for both heart rate and respiration. Therefore, a faster heart rate or respiration rate will output a faster rate of change. Theoretically, the rate of change should match the input rates.

Testing was first done to develop an equation to adjust the rate of change. Both the heart rate and respiration rate were tested using multiple input rates of change. A test initial condition and action were created in which the heart rate would change from 150 beats per minute to 10 beats per minute at a variable rate. A test initial condition and action were created in which the respiration rate would change from 50 breaths per minute to 10 breaths per minute at a variable rate. The numerical data from the tests are shown in Appendix H. The graphs showing the data from the rate of change in respiration rate and heart rate and the linear regressions are represented in Figures 17 and 18, respectively. The high rates of change that stray from the two regressions are unrealistic according to physiological principles. Hence, the equations were utilized in the program.

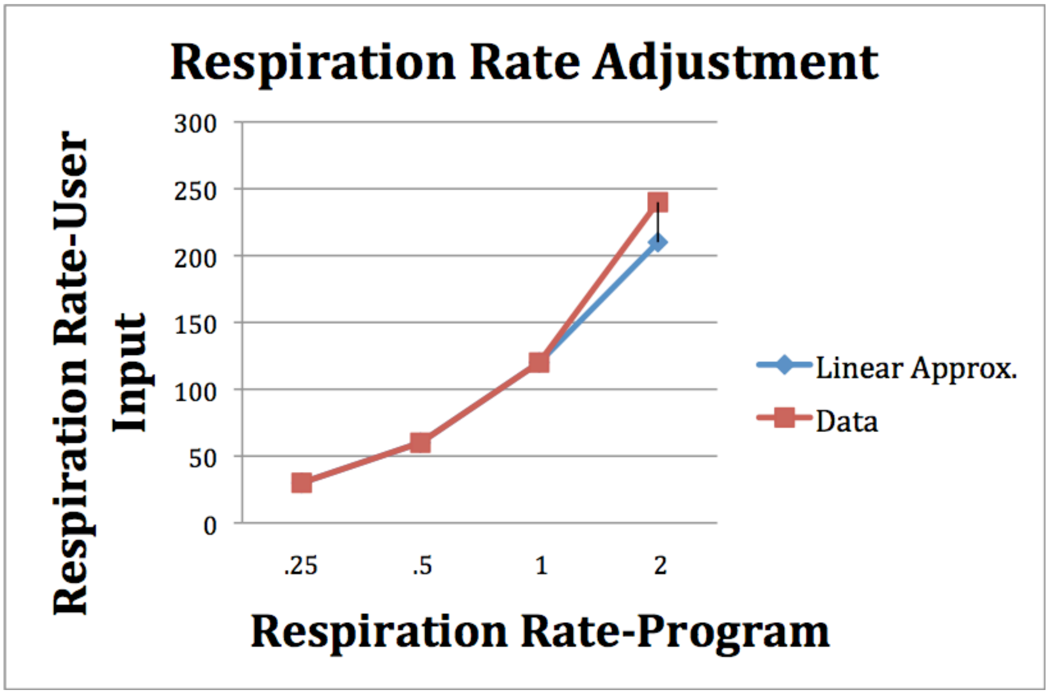


Figure 17: Respiration Rate Adjustment Graph

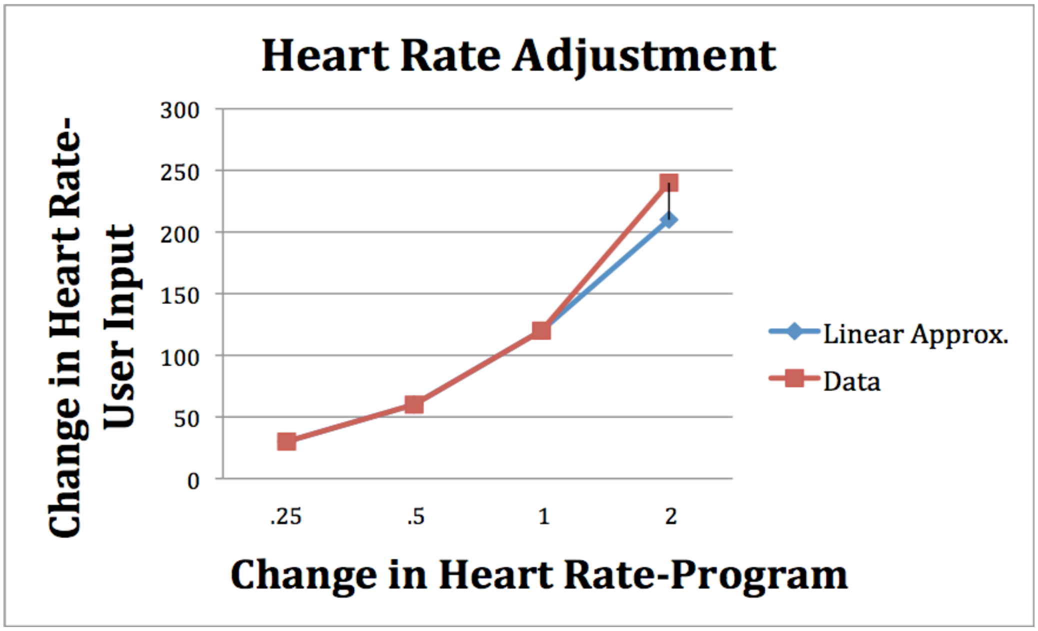


Figure 18: Respiration Rate Adjustment Graph

Qualitative Testing

Fourteen surveys with nursing students and health-care professionals were conducted. The responses are shown in Appendix G. General approval was received regarding the quality of the monitor simulation. However, certain improvements were suggested. A nursing student suggested that labels be added to the vital readings on the monitor. Two health-care professionals suggested that the scale of the graphical representation of the ECG signal be modified so that the individual units are farther apart. Furthermore, the spread of the ECG unit from the p wave to the t wave should not vary as the heart rate increases or decreases, but rather, the length between t wave and the p wave of the next ECG unit should vary. Several other recommendations regarding the respiration wave indicated a need for improvement of the wave so that it looks more realistic. These changes were implemented in the final program.

Future Work:

There are several areas for improvement of the current monitor. The variability in the EKG signal is limited to the heart rate and does not include variability in wave morphology. For scenarios other than tension pneumothorax, changes in the waveform other than the rate may be necessary. For example, in a scenario where tachycardia or arrhymia is involved, the current monitor wouldn't readily show the changes in EKG that correspond to these abnormalities. The data generated by Matlab and read by Java that creates the EKG waveform would have to be manipulated directly or the equation itself would have to be altered. The team has made it possible to manually change the P, Q, R, S, and T components of the EKG signal within the program, but these variations aren't an option in the reading of the text file. So, the user could modify the code of the program to change these components of the EKG signal, but not as easily as other variables that are modified by creating a text file that the program reads. The ability to easily change the P, Q, R, S and T components should be an option in the text file that the program reads.

Realistic respiration curves are difficult to model as they have small yet noticeable variations with no given pattern and include types of outputs that are difficult to model. Our respiration equation is a relatively simplistic representation of a true respiration wave, which is effective for the purposes of the project, but could still be improved so that the monitor looks more realistic. In addition, the user cannot easily change the amplitude of the respiration wave in the text file. The user has to change the amplitude of the respiration wave in the code of the program. To make the program more user-friendly, there should be a variable in the text file that changes the amplitude of the respiration wave.

Aside from improvements that should be made, our clients must translate the code into CAVELib before it can be implemented in the CAVE. Our clients must also make any additions necessary to the monitor program so that it is able to react based on the actions of the user in the CAVE. In addition, the team would like to make the monitor program publicly available for educational purposes. The team will have to work with the clients to include an explanation for how the monitor works as well as marketing the information so that it is a known tool for students studying physiology, medicine, and nursing.

Works Cited

Chen, Chen, and Cheng. "Delayed Tension Pneumothorax During Surgery." Mackay Memorial Hospital, 2005. Web. 20 Oct. 2011. <http://homepage.vghtpe.gov.tw/~jcma/68/10/491.pdf>.

"Collapsed Lung - PubMed Health." *PubMed Health*. U.S. National Library of Medicine, 2011. Web. 22 Oct. 2011. <http://www.ncbi.nlm.nih.gov/pubmedhealth/PMH0001151/>.

"Introduction to Nursing Theories." *Nursing Theories.* Web. 23 Oct. 2011 <http://currentnursing.com/nursing_theory/introduction.html>

Jain and Jain. "Understanding and Managing Tension Pneumothorax." Indian Academy of Clinical Medicine, Jan. 2008. Web. 20 Oct. 2011. <http://medind.nic.in/jac/t08/i1/jact08i1p42.pdf>.

Jelovic. "Why Java Will Always Be Slower than C++." Web. 23 Oct. 2011. <http://www.jelovic.com/articles/why_java_is_slow.htm>.

"MATLAB-The Language of Technical Computing." Mathworks. Web. 10 Oct. 2011. <http://www.mhttp://www.mathworks.com/products/matlab/athworks.com/products/matlab/>.

McSharry, Clifford, Tarassenko, Smith. " A Dynamical Model for Generating Synthetic Electrocardiogram Signals. " *IEEE Transactions on Biomedical Engineering*50(3): 289-294; March 2003.

Morris (2 July 2009). "Niklaus Wirth: Geek of the Week". Retrieved 8 August 2009. "C++ is a language that was designed to cater to everybody's perceived needs. As a result, the language and even more so its implementations have become complex and bulky, difficult to understand, and likely to contain errors for ever."

Programmers Stack Exchange. Web. 10 Oct. 2011. <http://programmers.stackexchange.com/questions/80439/why-is-c-still-preferred-to-build-heavy-gui-apps-over-the-latest-dynamic-langu>.

Radwin and Mandel. Personal interview. 7 Sept. 2011.

Stroustrup (1997). "1". The C++ Programming Language (Third ed.). ISBN 0201889544. OCLC 59193992.

Anesoft.com - Screen-based Medical Simulation Software. Anesoft, 2010. Web. 26 Oct. 2011. <http://www.anesoft.com/>.

"ANGIO Mentor." *Simbionix - To Advance Clinical Performance*. Simbionix, 2011. Web. 26 Oct. 2011. <http://simbionix.com/simulators/angio-mentor/>.

Appendix A: Product Design Specification Report

Product Design Specification Report
Virtual Reality Physio Monitor

Client: Professor Robert Radwin
Dr. Benjamin Mandel
Advisor: John Webster
Team: Hope Marshall, hkmarshall@wisc.edu (Leader)
Roland Pomfret, pomfret@wisc.edu (Communicator)
Jiaquan (Jason) Yu, yu53@wisc.edu (BWIG)
Gabriel Bautista, gbautista@wisc.edu (BSAC)

Date: September 15th, 2011

In order to simulate a real-world clinical environment for training purposes, a dynamic virtual monitor display system must be improved for the 3D CAVE. The monitor should display dynamic EKG's, respiration rate, blood pressure and $S\mathrm{pO}_2$ levels that respond to changing physiological conditions in the virtual ER trauma bay. The virtual display must sound an alarm if it detects that the avatar patient's vitals are passing a dangerous threshold.

Client requirements:

- Compatible with current virtual reality ER bay trauma room in CAVE
- Respond to a changing virtual physiological environment
- Clear and legible display
    - Display dynamic, realistic graphical readings for heart rate, respiration rate and $S\mathrm{pO}_2$
    - Display dynamic, realistic digital readings for blood pressure, heart rate, respiration rate and $S\mathrm{pO}_2$
- Sound an alarm if vitals cross a dangerous threshold
- Expandable to accommodate physiological scenarios that aren't already built in

Design requirements:

1. Physical and Operational Characteristics

a. Performance requirements: The monitor should display dynamic EKG's, respiration rate, blood pressure and $S\mathrm{pO}_2$ levels that respond to changing physiological conditions and user actions in the virtual ER trauma bay.

b. Safety: The safety hazards are minimal as the environment in question is virtual.

c. Accuracy and Reliability: Blood pressure, SpO_2, respiration rate and heart rate should be accurate to the single digit.

d. Life in Service: The virtual monitor should last as long as the CAVE is in service.

e. Shelf Life: There are no shelf-life requirements.

f. Operating Environment: The virtual monitor must be compatible with Windows XP and CAVELib.

g. Ergonomics: All users must be able to understand and interact with the CAVE virtual monitor. The virtual device must be easy to use and see from the virtual distance of the user.

h. Size: The virtual monitor should be 49 cm x 38 cm.

i. Weight: This is not applicable.

j. Materials: Google SketchUp, Java, CAVELib, CAVE.

k. Aesthetics: The virtual monitor must appear as a real monitor does in an ER trauma bay. The monitor must be labeled clearly and concisely.

2. Production Characteristics

a. Quantity: 1

b. Target Product Cost: $0.00

3. Miscellaneous

a. Standards and Specifications: Not applicable

b. Customer: The clients require software that responds to changing physiological conditions and is expandable to accommodate physiological scenarios that aren't already built in.

c. Patient-related concerns: Not applicable

d. Competition: Similar scenarios have been designed. However, the software is not available on the market

Appendix B: Original User Interface and Monitor Design

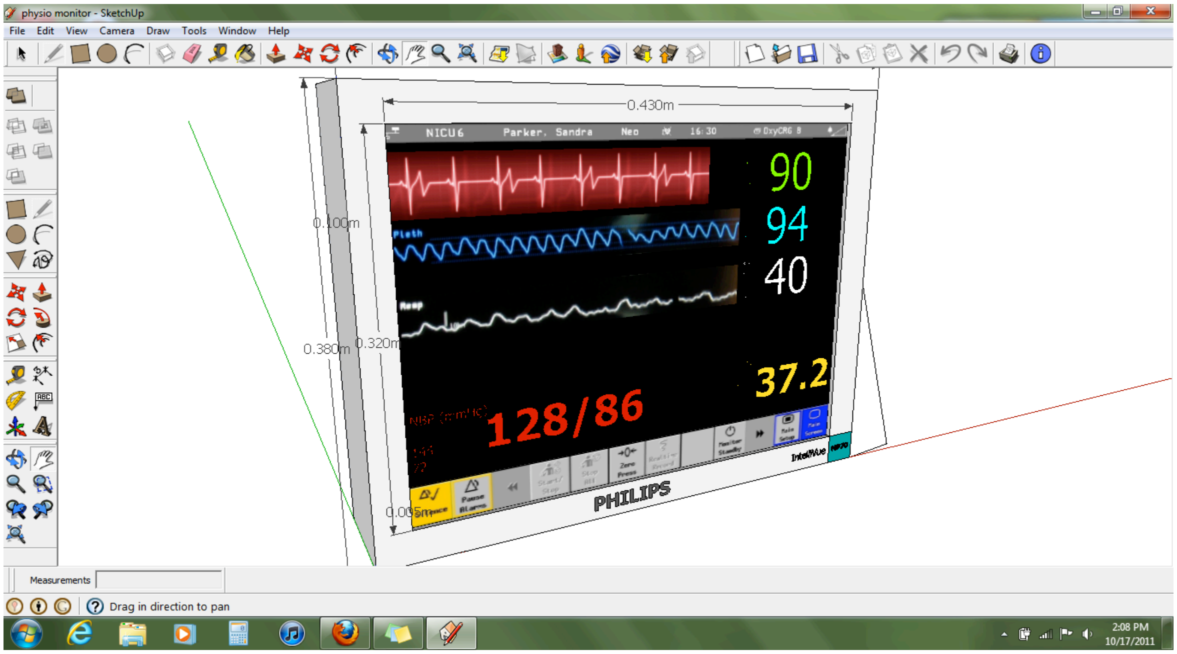


Figure 19: The above monitor design was made using Google SketchUp. The monitor design is a suggestion of what the monitor in the CAVE would look like according to the primary design in the middle of the semester.

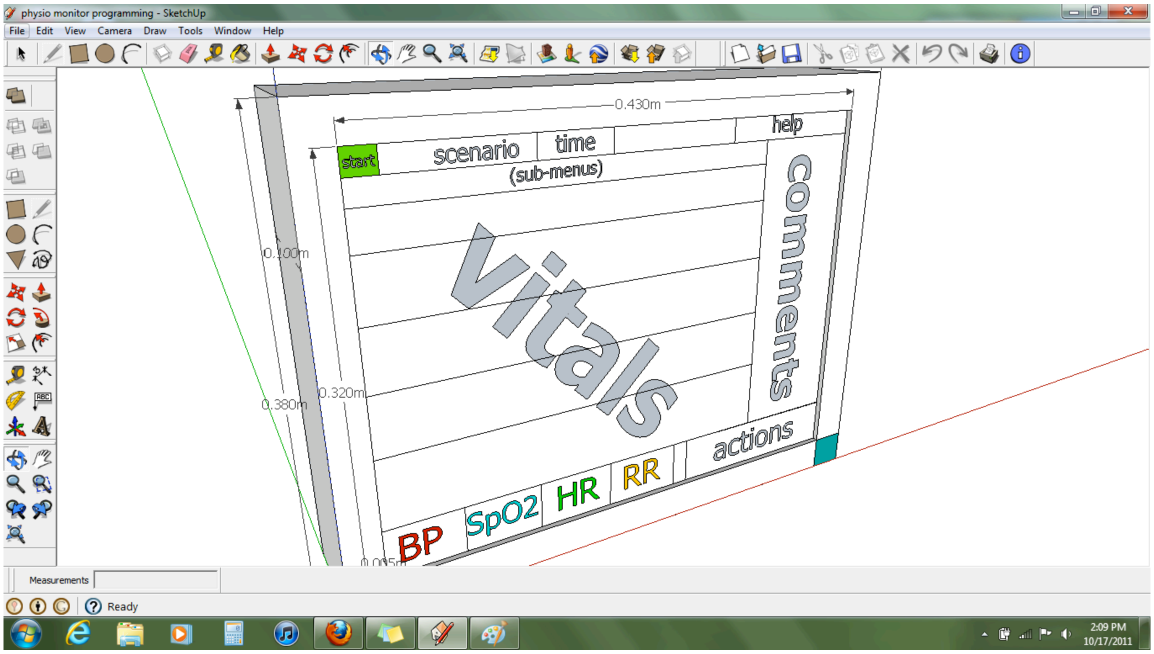


Figure 20: The above user interface design was made using Google SketchUp. The user interface design is a depiction of how the program would be run without actually being in the simulation environment from the middle of the semester.

Appendix C: Final Flowchart

Key: Each text box represents one action in the word file (which will be a text file)

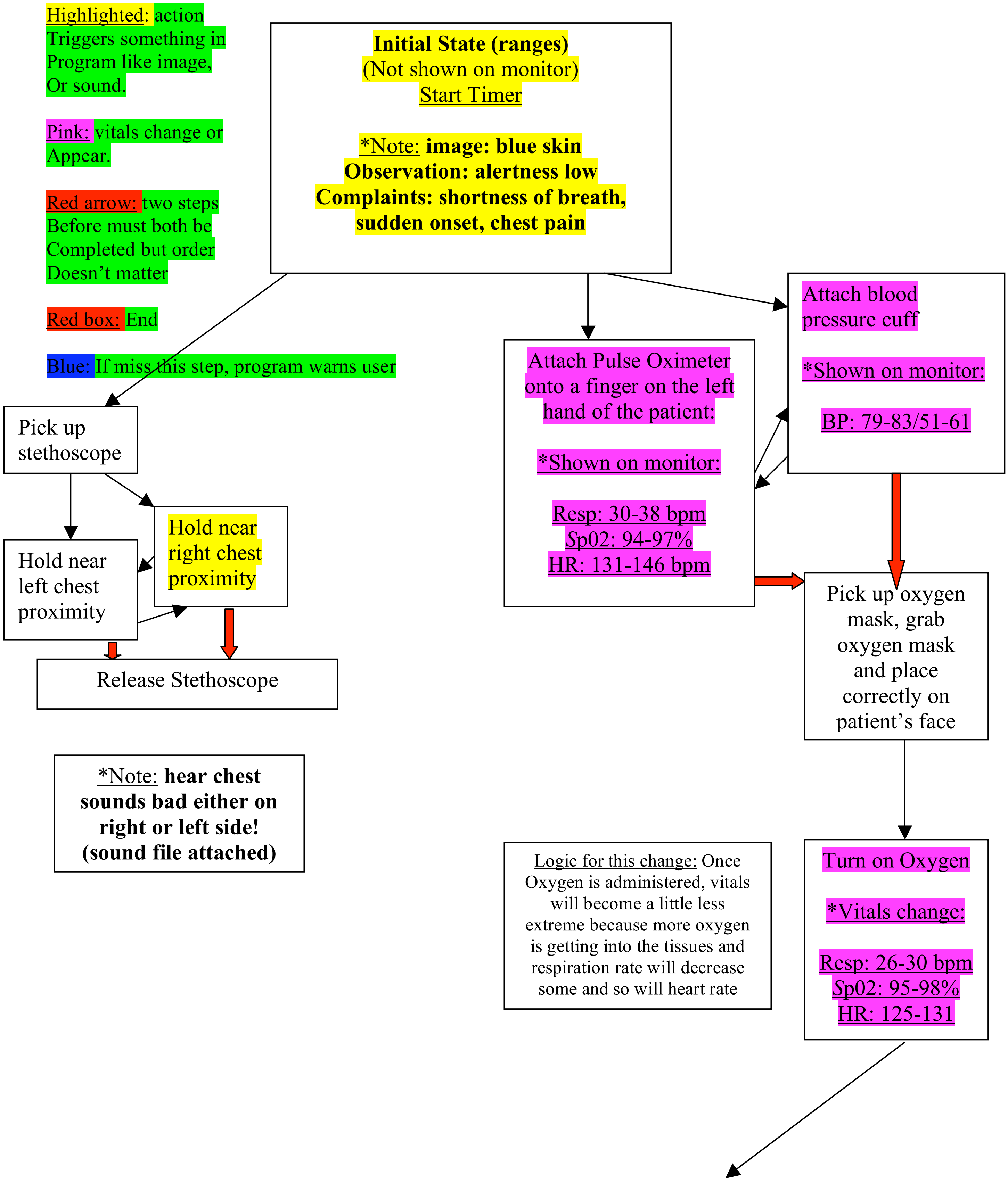
Highlighted: action Triggers something in Program like image, Or sound.

Pink: vitals change or Appear.

Red arrow: two steps Before must both be Completed but order Doesn't matter

Red box: End

Blue: If miss this step, program warns user



Initial State (ranges)
(Not shown on monitor)
Start Timer

*Note: **image: blue skin**
Observation: alertness low
Complaints: shortness of breath, sudden onset, chest pain

Pick up stethoscope

Hold near left chest proximity

Hold near right chest proximity

Release Stethoscope

*Note: **hear chest sounds bad either on right or left side! (sound file attached)**

Attach Pulse Oximeter onto a finger on the left hand of the patient:

*Shown on monitor:

Resp: 30-38 bpm
Sp02: 94-97%
HR: 131-146 bpm

Attach blood pressure cuff

*Shown on monitor:

BP: 79-83/51-61

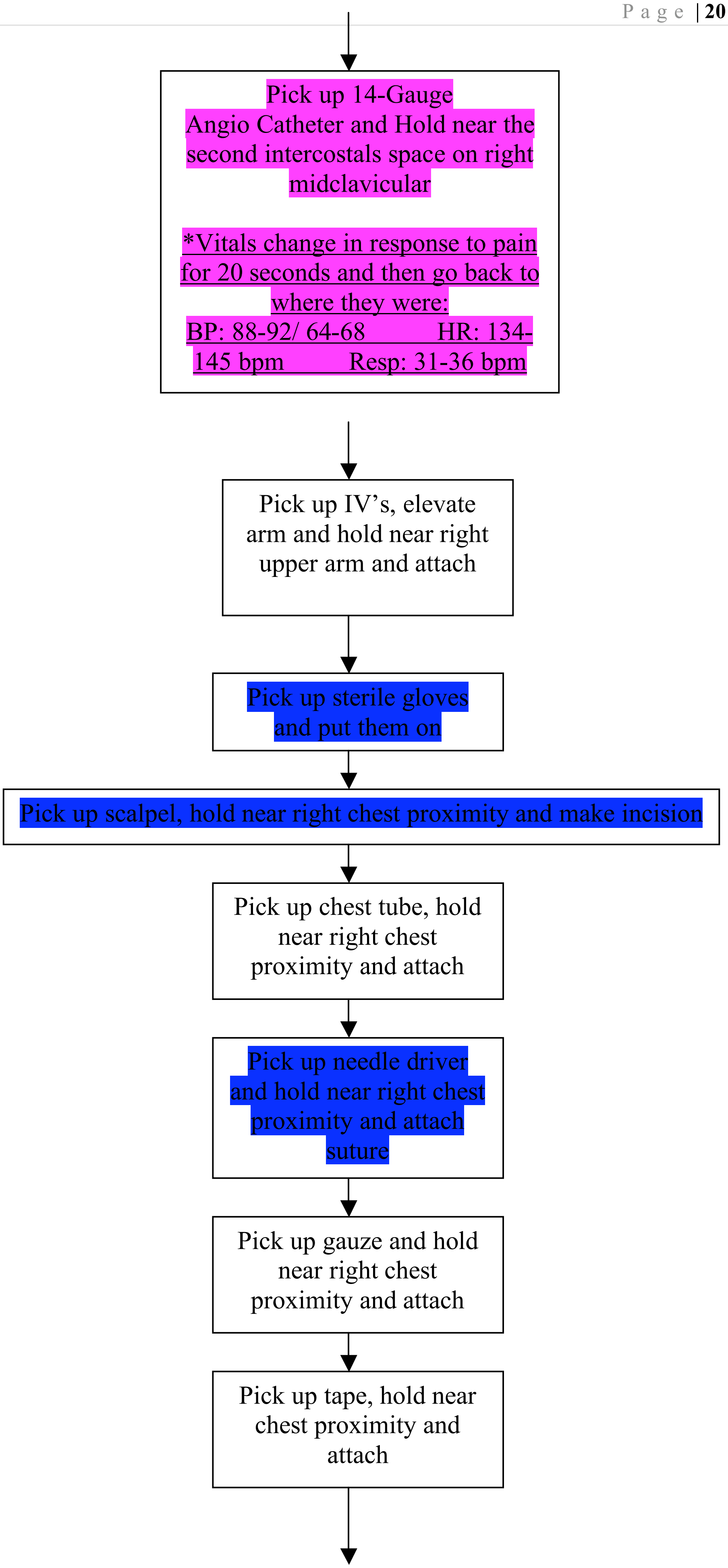
Pick up oxygen mask, grab oxygen mask and place correctly on patient's face

Turn on Oxygen

*Vitals change:

Resp: 26-30 bpm
Sp02: 95-98%
HR: 125-131

Logic for this change: Once Oxygen is administered, vitals will become a little less extreme because more oxygen is getting into the tissues and respiration rate will decrease some and so will heart rate



Pick up 14-Gauge Angio Catheter and Hold near the second intercostals space on right midclavicular

*Vitals change in response to pain for 20 seconds and then go back to where they were:
BP: 88-92/ 64-68 HR: 134-145 bpm Resp: 31-36 bpm

Pick up IV’s, elevate arm and hold near right upper arm and attach

Pick up sterile gloves and put them on

Pick up scalpel, hold near right chest proximity and make incision

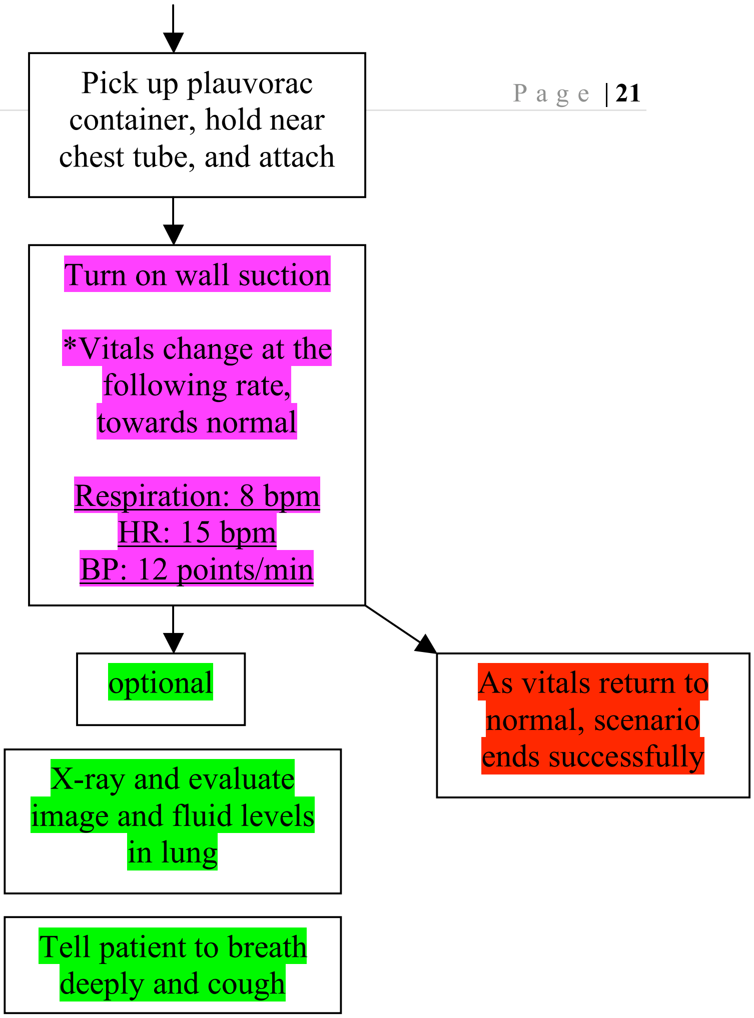
Pick up chest tube, hold near right chest proximity and attach

Pick up needle driver and hold near right chest proximity and attach suture

Pick up gauze and hold near right chest proximity and attach

Pick up tape, hold near chest proximity and attach

Logic for the rate of change: Should quickly start moving towards normal ranges. I figure after about a minute with the suction, vitals should be significantly close to normal ranges, hence the rates. I couldn't find research on exactly how long to return to normal ranges, but I figure by 2 min, they should be pretty close. Do these rates make sense?



Pick up plauvorac container, hold near chest tube, and attach

Turn on wall suction

*Vitals change at the following rate, towards normal

Respiration: 8 bpm
HR: 15 bpm
BP: 12 points/min

optional

X-ray and evaluate image and fluid levels in lung

Tell patient to breath deeply and cough

As vitals return to normal, scenario ends successfully

Appendix D: Class Diagram

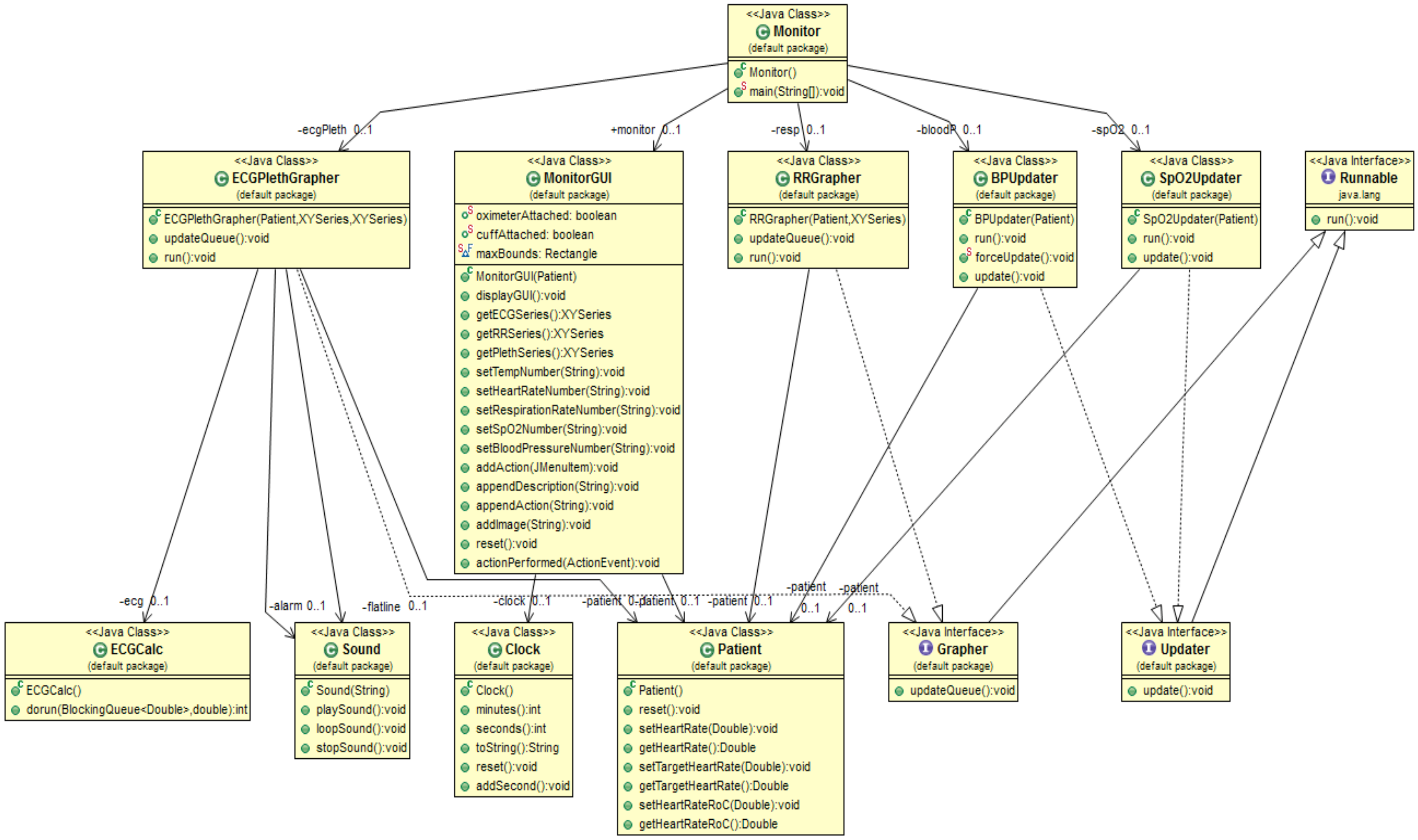


Figure 21: Class diagram of the Virtual Reality Physio Monitor Java program. Each box represents a unique class, its public variables and public methods. The classes are organized in hierarchical order, with the main class, Monitor, at the top

Appendix E: Guide to Creating a New Scenario (Text File)

Where and How to Create the Text File:

Creating a text file in Microsoft Word

The scenario text file can be created in Microsoft Word. After creating the document, click “Save” or “Save As” and then instead of “filename.doc”, type “filename.txt” and the file will be saved as a text file. Make sure that the text file is saved in your workspace.

For example, if the newly created scenario were saved as: heartfailure.doc, I would “Save As”: heartfailure.txt in my workspace and preferably in the folder for the Monitor Program within my workspace.

Creating a text file in Notepad

If you use the program Notepad, your work will automatically be saved as a text file. So, just make sure your text file is saved in your workspace, preferably in the folder for the Monitor Program within my workspace.

How to Format and Structure the Text File:

1. If an action is to be performed, it is represented by # a.
   i. “c” represents the initial condition
   ii. “t” represents the timer
   iii. “i” represents the image
   iv. “x” represents the text displayed on the monitor as an aid for patient diagnosis.

2. The format for the action should be represented in the following manner:

a Name of Action; Max HR; Min HR; HR SD; HR Rate of Change; Max RR; Min RR; RR SD; RR Rate of Change; Max *S*pO2; Min *S*pO2; *S*pO2 SD; Max systolic; Min systolic; Max diastolic; Min diastolic; Blood Pressure SD; if kill timer, number of timer to kill (first timer is 1, second is 2 and so on)

3. The format for the initial condition is as follows:

c Max HR; Min HR; HR SD; Max RR; Min RR; RR SD; Max *S*pO2; Min *S*pO2; *S*pO2 SD; Max systolic; Min systolic; Max diastolic; Min diastolic; Blood Pressure SD;

4. Special notes for actions:

a. If there is no change in a given vital for the corresponding action, replace either the Min or the Max for that vital with “n”. If there is an “n” in the spot for Min or Max, the vital will not change.

b. If the change is immediate, put an “l” in for rate of change instead of a value. Note that the units for rate of change are units of change/minute.

i. HR rate of change: change in **beats/minute**
ii. RR rate of change: change in **breaths/minute**
iii. Since BP updates every minute, there is no rate of change for BP

5. The format for setting up the timer is as follows:

t Description of What Happened; Max HR; Min HR; HR SD; HR Rate of Change; Max RR; Min RR; RR SD; RR Rate of Change; Max *S*pO2; Min *S*pO2; *S*pO2 SD; Max systolic; Min systolic; Max diastolic; Min diastolic; Blood Pressure SD; Time Delay in Milliseconds;

6. To insert an image, type an “i” on a new line and the filename, which should be saved in your workspace. An example follows:

```
i lung.png
```

Example: An example of a correctly formatted scenario is as follows:

```
#Initial Condition
c 146;131;2;38;30;2;97;94;.5;83;79;61;51;2
x Patient complains of shortness of breath and chest pain that occurred suddenly. You observe that the patient is not especially alert and that their skin tone is slightly blue.
t Patient has died due to lack of Oxygen;0;0;0;i;0;0;0;i;0;0;0;0;0;0;0;0;20000
#Steps in procedure
a Turn on Oxygen;131;125;n;5;30;26;n;5;98;95;n;n;n;n;n;n;1
a Make Incision in Chest;145;134;n;i;36;31;n;i;n;n;n;92;88;68;64;n
t Patient has died out of neglect :( ;0;0;0;i;0;0;0;i;0;0;0;0;0;0;0;0;20000
i lung.png
```

Explanation of Example:

In the above example, the tension pneumothorax scenario, the initial conditions would be:

Heart rate:

Max heart rate: 146 beats per minute
Min heart rate: 131 beats per minute
Heart rate standard deviation: 2 beats

Therefore, the heart rate would be randomly chosen between 131 bmp and 146 bpm and would vary randomly according to a normal distribution, staying within 2 beats of the chosen value 67% of the time.

Respiration:

Max respiration: 38 breaths per minute
Min respiration: 30 breaths per minute
Respiration standard deviation: 2 breaths

Therefore, the respiration rate would be randomly chosen between 30 and 38 breaths per minute and would vary randomly according to a normal distribution, staying within 2 breaths of the chosen value 67% of the time.

*S*pO2:

Max *S*pO2: 97%
Min *S*pO2: 94%
*S*pO2 SD: 0.5%

Therefore, the *S*pO2 would be chosen between 94% and 97% and would vary randomly according to a normal distribution, staying within 0.5% of the chosen value 67% of the time. *S*pO2 updates every 10 seconds.

Blood Pressure:

Max systolic blood pressure: 83 mm Hg
Min systolic blood pressure: 79 mm Hg
Max diastolic blood pressure: 61 mm Hg
Min diastolic blood pressure: 51 mm Hg
Blood Pressure standard deviation: 2 mm Hg

Therefore the systolic blood pressure (top number) would be randomly chosen between 79 and 83 mm Hg and would vary randomly according to a normal distribution, staying within 2 mm Hg of the chosen value 67% of the time.

The diastolic blood pressure (bottom number) would be randomly chosen between 51 and 61 mm Hg and would vary randomly according to a normal distribution, staying within 2 mm Hg of the chosen value 67% of the time. Blood Pressure updates every minute on the monitor

Text:

Patient complains of shortness of breath and chest pain that occurred suddenly. You observe that the patient is not especially alert and that their skin tone is slightly blue.

Timer:

Next, the time constraint is shown. It is shown that all vitals will become zero and images will be shown (represented by “i”), in addition, at 20000 milliseconds (20 seconds) if the action action: “Turn on Oxygen” hasn’t been performed. Also, it tells the program to stop the first timer, timer 1, indicated by the last value.

Actions:

The actions are formatted similar to the initial condition. The new values represent the changes in the vitals from the initial condition or a previous action. “n” represents No Change, so all of the minimums, maximums and standard deviations with “n” in place of a number do not change based on the action performed.

Appendix F :Tension Pneumothorax Text File

```
i lung.png
c 146;131;2;38;30;2;97;94;.5;83;79;61;51;2
x Patient complains of shortness of breath and chest pain that occurred suddenly. You observe that the patient is not especially alert and that their skin tone is slightly blue.
t Patient has died due to failure to place angiocatheter;0;0;0;i;0;0;0;i;0;0;0;0;0;0;0;0;300000
#Steps in procedure
a Turn on Oxygen;131;125;n;20;30;26;n;10;98;95;n;n;n;n;n;n
a Insert 14-Gauge Angiocatheter;112;105;n;40;27;24;n;18;100;98;n;108;98;80;69;n;1
x Your patient is no longer in a life-threatening condition.
a Insert chest tube;78;70;n;30;18;16;n;14;100;98;n;116;108;87;80;n
a Turn on suction;78;59;n;30;16;13;n;12;100;98;n;116;108;87;80;n
x Congratulations, your patient is quickly recovering!
i betterlung.png
```

Appendix G: Qualitative Testing Responses

Testing: pre-health/nursing majors

Name: Jared
Qualitative Comments: Looks realistic. Was extremely interested in learning how it worked and what each word meant and how the program runs. Interested in learning about tension pneumothorax. Thought the monitor looked realistic and had no suggestions.

Name: Jamie
Qualitative Comments (Nursing Assistant):
Hr too pretty
Respiration doesn't have baseline
Otherwise awesome

Name: Anyi Wang
Qualitative Comments: Units on Monitor are needed

Name: Ashleigh
Qualitative Comments: Thinks it looks realistic and doesn't have any suggestions off the bat

Name: Jason
Qualitative Comments: Colors look, realistic, deviates for respirations positive (baseline doesn't matter as much)

Name: Hayley
Qualitative Comments: EKG scale looks weird. Modify this.

Name: Jordan
Qualitative Comments: EKG units look too close together.

Name: Sarah
Qualitative Comments: Need units on Monitor. Other wise looks really nice

Name: Dr. Miranpuri
Qualitative Comments: Respiration doesn't look realistic. Need units on Monitor.

Name: Sasha
Qualitative Comments: Not sure if Respiration curves look realistic or not. ECG seems too close together or has weird scaling and looks too perfect.

Name: John
Qualitative Comments: need units on the monitor

Name: Dr. Silay

Qualitative Comments: Need units on the monitor, ECG looks too perfect and doesn't vary enough per unit, respiration doesn't look realistic. However, I think this will be good enough for the application you are trying to achieve with it.

Name: Angie
Qualitative Comments: Units on monitor are needed. Otherwise, looks very realistic

Name: Lauren
Qualitative Comments: Looks really good! I think it looks

Appendix H: Quantitative Testing Data

Test rate: Final data relation

Resp:
5 breaths/min=.25
10 breaths/min=.5
20 breaths/min=1
34 breaths/min=2 (unrealistic)

y1=(1/20)(x1) where x1 is the breaths per minute that they entered.

So whatever value the user enters for the rate of change for respiration should be multiplied by 1/20 and then the program will go at approximately the rate the entered.

Hr:
30 beats/min=.25
60 beat/min=.5
120 beats/min=1
210 beats/min=2 (unrealistic)

y2=(1/120) (x2) where x2 is the beats per minute change that they entered

So whatever value the user enters for the rate of change for heart rate should be multiplied by 1/120 and then the program will go at approximately the rate the entered.

Data used to generate the above conclusions:

Hr
Rate=1
Started at 30 seconds
150-125: t=42
125-100: t=55

Rate: 2 beat/sec=120 beats/min

Resp
Rate=1
Started at 30
30-25: t=40
25-20: t=53
20-15: t=110
15-10: t=134

Rate: 20 breaths/min

Hr
Rate=2
Started at 10 seconds
150-125: t=16
125-100: t=24

Rate: 210 beats/min

Resp
Rate=2
Started at 10 seconds
30-25: t=17
25-20: t=24
20-15: t=32
15-10: t=46

Rate: 34 breaths/min

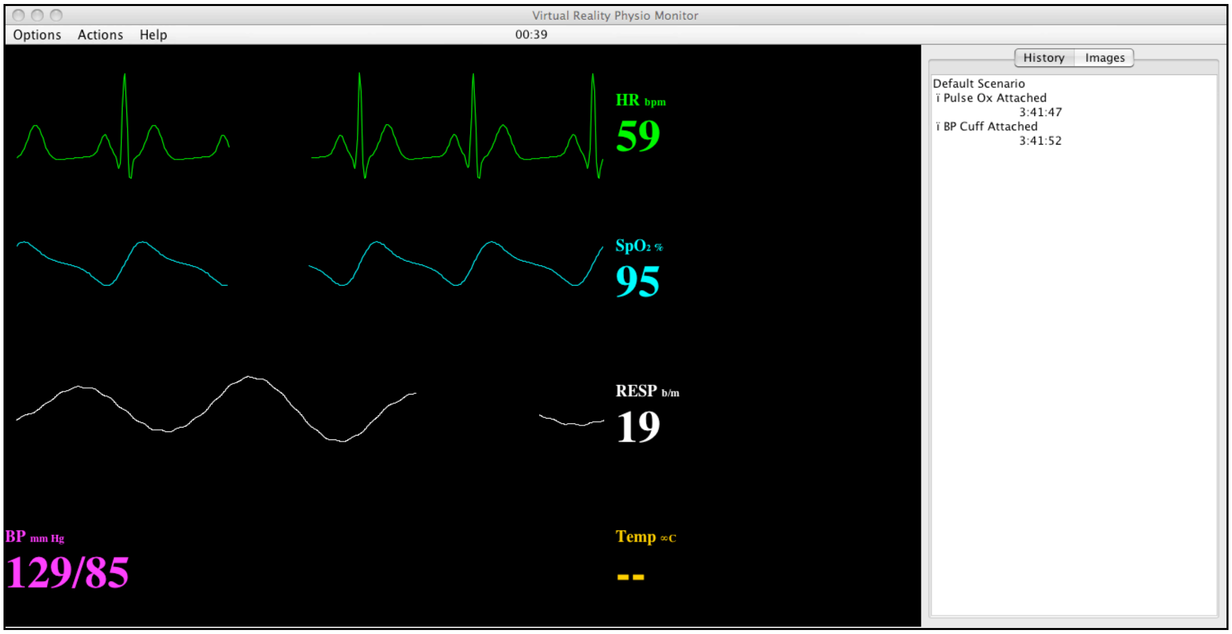
Hr
Rate=.5
Started at 10 seconds
150-125: t=33
125-100: t=59

Rate=1 beat/sec

Resp
Rate=.5
Started at
30-25: t=30
25-20: t=56
20-15: t=130
15-10: t=223

Rate: 10 breaths/min

Appendix I: Virtual Physio Monitor



HR bpm
59

SpO_2 %
95

RESP b/m
19

BP mm Hg
129/85

Temp ∞C
--

```
Default Scenario
ï Pulse Ox Attached
            3:41:47
ï BP Cuff Attached
            3:41:52
```

Figure 22: Vitals Monitor Final Design and User Interface. The sweeping bar updating the vitals is visible as a dark gap in the graphics. Blood pressure in mm Hg is displayed on the bottom left and the ECG, photoplethysmograph and respiration graphics and values in beats per minute, percent oxygen and breaths per minute are displayed respectively. The history log is displayed right above the monitor and the timer, options menu, actions menu and help menu are displayed above the monitor.