CS Capstone Technology Review

Fall Term

November 21, 2017

Kinect Based Virtual Therapy Solution

Prepared for

OSU Healthcare Systems Engineering Lab

Mehmet Serdar Kilinc

Prepared by

Group 24

Ciin S. Dim

Abstract

The purpose of this document is to analyze the possible implementations of the components that our project is comprised of. The project involves the use of a Kinect sensor to track a patient's movements when performing exercises. The data that the sensor records will be stored and sent to their physical therapist to allow them to monitor their patient's progress. The task is to develop software that includes an interface for patients and physical therapists to interact with. Pre-defined exercises will be implemented in the software and compared against a patient's movements to determine the accuracy of the therapy. The document is organized into pieces, or components, of the project, and it is further divided into the options for implementing each piece.

Contents

1 Piece One: Development Environment

1.1 Overview

A development environment is a set of software that enables the writing of programs for a particular language or platform. This software often includes a text editor, shell, and a compiler/interpreter[1].

1.2 Criteria

The development environment we end up choosing needs to be well suited for the language we will be programming in, as well as the operating system the program will be running on. It must also optimize the ease of developing with the hardware we are using.

1.3 Potential Choices

1.3.1 Oracle Netbeans

Netbeans is a free, open source IDE that is optimized for Java development. It also supports other languages such as C/C++, Ruby, HTML5, PHP. The plugin-centric design of this IDE makes it extensible by allowing additional language syntax support and libraries to be added[1].

1.3.2 Eclipse

Similar to Netbeans, Eclipse is an IDE with a plugin architecture that is well suited for Java development. It can also be used for many different workflows and languages[1]. There are several possible versions of Eclipse that could be used by installing one or more packages. These packages include one or more features such as C/C++ development tools, git integration, Maven integration, PHP development tools, Java development tools, JavaScript development tools, and many more[2].

1.3.3 Microsoft Visual Studio

Visual Studio is an Integrated Development Environment (IDE) used for creating windows applications. It includes a text editor and compiler that covers languages such as C++, C#, and .NET among many others. Visual Studio also enables development for various Microsoft products such as Windows x86, Windows RT, and Windows Phone[1]. It can also easily integrate Microsoft Azure cloud services.

1.4 Discussion and Conclusion

Since we will be programming in C++, any of the mentioned development environment options would be suitable because they all support C++ development. However, Visual Studio is the best choice for implementing our project because it fits the criteria for the right development environment. We plan to develop our program on a Windows operating system. We have decided this because the system requirements of our chosen hardware, the Microsoft Kinect Sensor, states that the supported operating systems for the Kinect Software Development Kit (SDK) are Windows 8 (x64), Windows 8.1 (x64), Windows 8 Embedded Standard (x64), Windows 8.1 Embedded Standard (x64)[3]. We have chosen the Kinect because it is what our client has chosen for us to use. The system requirements also state that the supported development environments are Visual Studio 2012, and Visual Studio 2013, which means that our chosen hardware is compatible with this development environment[3]. Unlike Netbeans and Eclipse, Visual Studio is best suited for developing projects for Windows platforms. Since the Kinect SDK was developed by Microsoft, Visual Studio can easily integrate it. Although it is possible to write C++ programs in Eclipse and Netbeans, the Kinect SDK requires Visual Studio to work.

2 Piece Two: Data Collection

2.1 Overview

For our project, we will need to save the user's node movement data to be exported and sent to their physical therapist. The data that is saved will consist of coordinates of 25 nodes at every (specified) interval of time.

2.2 Criteria

The method of data collection we use for this project must be best suited for this specific type of data. A unit of data would be coordinates. At each time interval, there will be 25 units of data. After a data collection period, there will be 25 multiplied by the number of time intervals in that period. The data collected must be able to be shared and viewed easily.

2.3 Potential Choices

2.3.1 Database

A database is a collection of information that is organized so that it can be easily accessed, managed and updated[4]. Data is organized into rows, columns, and tables and are indexed for easy reference. The tables can update themselves based on changes in the database by querying the database. There a several different types of databases that are optimal depending on the type of data that will be stored in it. They are often appealing for very large data sets as they are more easily scalable and manageable than CSV files or file systems. They are mainly used for storing related, structured data.

2.3.2 File System

File systems are mainly used for unstructured data that may not necessarily be related. There is no indexing of entries or well-defined data format; files only contain text. Since there is no data format, finding data requires reading text files and iterating through them to search for that data. Inserting data only requires appending to a file. Using a file system can get more complex if there needs to be operations done to the data such as searching or sorting. They could be more efficient in cases where the data is unstructured and accessed infrequently.

2.3.3 Comma-Separated Values (CSV) File

A CSV file contains data in plain text that is separated by commas. Each line of data in the file is an entry, and each entry contains one or more fields separated by commas. This file format is used to exchange data between different applications that are similar. A CSV file can be viewed in KSpread, OpenOffice Calc and Microsoft Excel spread-sheet applications[5]. The data exported to a CSV file would be saved locally to the device running the program and could then be sent to anyone. Anyone receiving the file would have many options to view this data as it is readable in plain text and compatible with many software.

2.4 Discussion and Conclusion

Since the collected data will be structured into coordinates, nodes, and time intervals, a file system would be the least suitable implementation of data collection in our case. Unlike databases and CSV files, file systems have no indexing, which make it difficult to find and view a specific unit of data based on the time interval and/or node. While databases provide easy access and reference to data, it contains far more complexity than what is needed for our data. One main reason for this is that the data collected will not be stored for extended periods of time. They are only meant to be collected briefly to be exported and shared. If the data were to be stored, they could be stored in different categories, or tables, that would be used in a database. Since data is only collected for specific periods of time, it is unnecessary to store each period of data collection this way. The only thing we require is a method that is structured and enables easy sharing of the data. That is why CSV files are the best option for implementing this. Each line of a CSV file will correspond to a time interval, and each field in an entry will correspond to a node. That way, a specific node position at a specific time can be easily found. It is also organized conveniently so that one CSV file is one period of data collection.

3 Piece Three: Visualization

3.1 Overview

Once data has been successfully collected, our program will display a summary of the results. This summary will include visualizations of the data collection from a user's activity session. The visualizations will be in the form of a line graph, bar chart, and/or pie chart.

3.2 Criteria

The data visualization tool used for the summary will have to be compatible with most browsers. It must be able to generate graphs with the least amount of dependencies so that the users will not need to download additional libraries or programs.

3.3 Potential Choices

3.3.1 D3.js

D3.js is a free and open source data visualization tool that uses HTML, CSS, and SVG to render charts and diagrams[6]. This tool provides the capability of creating extremely complex visualizations both static and dynamic. It has a steep learning curve and is only compatible with modern browsers[6].

3.3.2 Chart.js

Chart.js is an open source library that supports just six chart types: line, bar, radar, polar, pie and doughnut[6]. It uses HTML5 canvas element for rendering charts. All the charts are responsive and use flat design.

3.3.3 Google Charts

Google Charts renders charts in HTML5/SVG to provide cross-browser compatibility and cross-platform portability to iPhones and Android. It has the capability to render complex graphs and diagrams as well as simple ones. It does not require any plugins or software to view the visualizations and is compatible across browsers and devices[6].

3.4 Discussion and Conclusion

All of the mentioned options use HTML and JavaScript to render visualizations from data. While all of these tools are fully capable of using to implement visualization, D3.js may be too complex to learn and use for our case. While Chart.js is simple, it requires installation for development and generation of visualizations; we would like a tool that can produce visualizations on all or most browsers and platforms. Were using Google Charts because it fulfill our needs of generating the types of visualizations we want to generate, and it is more compatible than Chart.js. Although D3.js is just as capable of generating the visualizations we want, the documentation and support isnt as detailed or readily available as Google's documentation. All of these tools are quite similar, but using Google Charts will be the most convenient and easiest experience when integrating visualization into our program.

References

[1] "Life after codecademy: setting up your development environment." http://www.shubhro.com/2014/05/29/development-environment/, May 2014. (Accessed on 11/13/2017).

[2] "Compare eclipse packages." http://www.eclipse.org/downloads/compare.php. (Accessed on 11/13/2017).

[3] "System requirements." https://msdn.microsoft.com/en-us/library/dn782036.aspx. (Accessed on 11/13/2017).

[4] M. Rouse, "database (db)." http://searchsqlserver.techtarget.com/definition/database, February 2017. (Accessed on 11/13/2017).

[5] "Comma separated values (csv) standard file format." http://edoceo.com/utilitas/csv-file-format. (Accessed on 11/13/2017).

[6] N. Sharma, "The 14 best data visualization tools." https://thenextweb.com/dd/2015/04/21/the-14-best-data-visualization-tools/, April 2015. (Accessed on 11/13/2017).