

CS CAPSTONE DESIGN DOCUMENT

FALL TERM

DECEMBER 1, 2017

KINECT BASED VIRTUAL THERAPY SOLUTION

PREPARED FOR

OSU HEALTHCARE SYSTEMS ENGINEERING
LAB

MEHMET SERDAR KILINC

PREPARED BY

GROUP 24

CIIN S. DIM

LOUIS LEON

Abstract

The purpose of this document is to define and describe a possible solution for physical therapists to utilize when monitoring a patient's prescribed therapeutic movement set. The solution 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 project will be completed once a working prototype is prepared and the clients' requirements are satisfied. The document is structured into three sections which provide a high-level description of the problem, solution, and performance metrics.

CONTENTS

1	Introduction	3
1.1	Purpose	3
1.2	Scope	3
1.3	Summary	3
1.4	Glossary	3
1.5	References	3
	References	3
1.6	Overview	4
1.7	Schedule	4
2	Body Tracking Sensor	5
2.1	Context	5
2.2	Dependencies	5
2.3	Composition	5
2.4	Interactions	5
3	User Interface	5
3.1	Context	5
3.2	Dependencies	5
3.3	Composition	6
3.4	Interactions	6
4	Exercise Recognition	6
4.1	Context	6
4.2	Dependencies	6
4.3	Composition	6
4.4	Interactions	6
5	Data Collection	7
5.1	Context	7
5.2	Dependencies	7
5.3	Composition	7
5.4	Interactions	7
6	Report Generation	7
6.1	Context	7
6.2	Dependencies	7
6.3	Composition	8
6.4	Interactions	8

		2
7	Data Visualization	8
7.1	Context	8
7.2	Interactions	8

1 INTRODUCTION

1.1 Purpose

The purpose of this document is to describe the details of major components of this project. Each component will be depicted from various design viewpoints. The details outlined in each component will define the standards of completion. This document can be used by the client to confirm what will be implemented. The developers will use this document as a guide to implementing the project and for testing the final product to ensure it works as intended. The developers and client can then use this document to determine whether the final product meets the standards defined.

1.2 Scope

The components described in this document are the body tracking sensor, organization of the user interface, exercise recognition, data collection, report generation, and data visualization.

1.3 Summary

The *Kinect Physical Therapy Solution* uses a Kinect body tracking sensor to collect the data of a user. The user can specify which exercise they want to begin using the user interface and menu system. During the exercise, the user's skeletal data is captured and processed by the data collector and exercise recognizer, respectively. A report is generated for the user and acts as a progress chart to see their improvements over time. The data for each exercise session is captured and parsed into a .csv file with the intention of storing locally and sending to the patient's physical therapist.

1.4 Glossary

Term	Definition
.csv File	Stands for "comma-separated values". A file format that is used to store tabular data, such as a spreadsheet or database. They may be imported/exported into different programs that store data in tables[1].
Node	In this context, a node represents a point along the body of the user. Typically associated with a joint in the skeletal system[2].
PC	Personal Computer
Kinect For Windows SDK	A software development kit used to program the Kinect sensor[2].
Data Visualization	A graphical image representing data with graphs, shapes, colors, lines, etc.

1.5 References

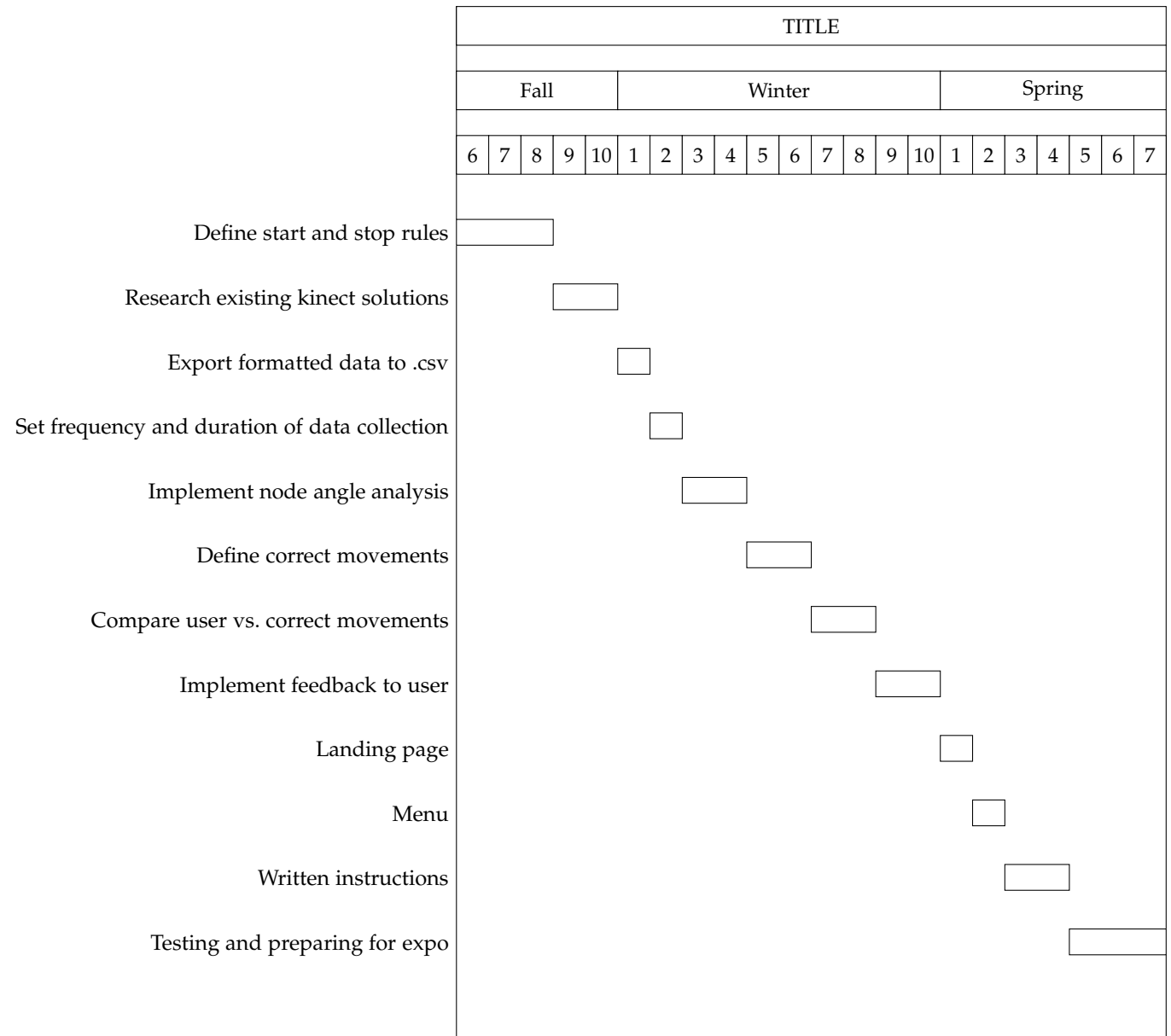
REFERENCES

- [1] "How to create a csv file." <https://www.computerhope.com/issues/ch001356.htm>, September 2017. (Accessed on 10/26/2017).
- [2] "Developing with kinect for windows." <https://developer.microsoft.com/en-us/windows/kinect/develop>. (Accessed on 11/13/2017).
- [3] "Kinect for windows sdk 2.0." <https://www.microsoft.com/en-us/download/developer-tools.aspx>. (Accessed on 11/13/2017).
- [4] "Windows presentation foundation samples." <https://msdn.microsoft.com/en-us/library/hh855377.aspx>, 2017. (Accessed on 11/13/2017).

1.6 Overview

This document will provide an overall description of our project and expected final product. This description will include product perspective, product functions, user characteristics, constraints, and assumptions and dependencies. Following the overall description are the specific requirements that outline our product's functionality as well as a project schedule.

1.7 Schedule



2 BODY TRACKING SENSOR

2.1 Context

The use of the Microsoft Kinect body tracking sensor will be essential for the tracking of users and their exercises. The sensor will act as the main interface between the user and the software.

2.2 Dependencies

A Windows 8 or higher operating system is needed to run the program and support our chosen model of the Kinect sensor (v2). The operating system hosts and runs the program at the users request which will then load the starting menu for exercise selection or data capture configuration. The sensor also requires the Microsoft .NET Framework 4.5 for development [2].

2.3 Composition

The Kinect sensor connects to a PC though USB 3.0 with the addition of standard power input. Development does not directly require the sensor as it functions mainly for a source of input for the software. A Windows device driver will be controlling the device during operation. The sensor is able to track up to six bodies through the use of 25-point skeletal mapping and supports open-hand or closed-hand gesture recognition. The sensor also has a built-in microphone to detect voice commands which may be implemented as a stretch goal for our project which would let users interact with the useron interface and menu systems [3].

2.4 Interactions

The sensor interacts with the software via the *Kinect for Windows* SDK. User skeletal data captured by the sensor is handled and processed by the exercise recognition modules in the software. All of the data being captures is collected via the data collection modules which parse the data out into a formatted .csv file.

3 USER INTERFACE

3.1 Context

The main user interface for this project will be organized in a very simple manner. This will be in an effort to minimize user confusion and make the overall experience more intuitive. The user interface helps with the selection of the desired exercise and report generation. The structure and organization of the user interface will be composed of menu options and visual cues.

3.2 Dependencies

Structuring of the elements for the user interface will be done through the use of Windows Presentation Foundation UI Framework (WPF). WPF provides the ability to create a user interface through Microsoft Visual Studio and supports images, 3D graphics, and video/audio [4].

3.3 Composition

The composition of the UI includes a main menu, exercise selection, settings, report generation, and data exportation. The main menu is where the user will start as they initiate the program. From the main menu, a user will have the option of starting an exercise routine. Once the exercise routine option is selected from the main menu, various exercises will appear for the user to choose from. The exercise starts once selected and on-screen instructions appear to guide the user through the required movements. Once complete, a report of their performance is generated and the exercise routine selection screen appears. The user interface should also include an overlay during the exercise portion which indicates how well the user is doing.

3.4 Interactions

The user interface provides the software with specific inputs and configurations for data collection and exercise recognition. The corresponding verification will need to be active with the currently selected exercise such that the user receives correct feedback and reports are properly generated.

4 EXERCISE RECOGNITION

4.1 Context

The exercise recognition aspect of the project is what reads the output of the sensor and compares it with a baseline version of an exercise. It serves as a comparison for the user and will be helpful when indicating feedback to the user. The user does not directly interact with this piece of the software but instead follows the on-screen instructions for performing the specific exercise they selected for the current session.

4.2 Dependencies

Recognition of user exercises heavily depends on the *Kinect for Windows* SDK for API calls. Specifically, the skeletal and node data of the user's body is required as an input in order to begin evaluating their movements.

4.3 Composition

The elements which make up exercise recognition include the detection of linear gestures or movements that follow a given path. The path is to be defined by the developer and is built into the software. There will be pre-defined movements for each of the available exercises. The following steps describe the method for recognizing linear gestures:

- Check that all necessary points are in progression to the correct direction
- Check if necessary limbs or joints fall within a specified range on the x, y, and z axes
- Check if a minimum distance has been covered with the movement
- Check that the first and last positions were created within a given period of time

4.4 Interactions

The exercise recognition module mainly receives inputs from the sensor data and directly accesses the values of the user's skeletal position in order to process and verify it. If the user is performing a movement or gesture incorrectly, a visual cue will be shown on-screen requesting that the user adjust their body while the exercise recognition continues checking until it deems the movement to be correct. User accuracy will be logged and reported in the data visualization and report generation after they finish performing their exercise.

5 DATA COLLECTION

5.1 Context

The collection of node data is used to analyze and evaluate the user's movements. Data collection will occur behind-the-scenes as the user completes exercises. The user will not be responsible for starting and stopping data collection; collection will automatically begin and end based on indication of the beginning and end of an activity.

5.2 Dependencies

Data collection is dependent on the Kinect sensor to obtain node data from the user's body. Node data is made available for collection by the Kinect SDK which provides an interface for the software to use data captured by the sensor[2].

5.3 Composition

The beginning and end of data collection will be prompted by start and stop rules. The start and stop rules are the node positions or movements that indicate the beginning or end of an activity of interest. These positions and/or movements are to be clearly distinguishable from positions indicating user interface interaction or an idle user.

When data collection is active, node positions (for all nodes) will be recorded at time intervals. The granularity of the time intervals can be customized. The node data will then be saved to a .csv file in which each column contains data for one of the nodes, and the rows contain data for all nodes at a specific time. Each cell, or data entry, will be the position of a node at a specific time. When data collection ends, the .csv will be exported and saved locally to the device running the program. Each .csv file will be a single session, and several sessions will be saved over time.

5.4 Interactions

Data collection interacts with exercise recognition when the positions or movements defined by start and stop rules are detected. When the nodes indicate the beginning or end of data collection, exercise recognition will signal data collection to start or stop saving node data.

6 REPORT GENERATION

6.1 Context

The physical therapist's user interface option will include the ability to view a summary of patients' activity. The summary will show a patient's progress over time, accuracy of movements, duration of exercises, and other various statistics calculated from the collected node data.

6.2 Dependencies

Report generation relies on the output of the data collection component. The data is used to calculate data about several previous sessions. Report generation also relies on data visualization to produce the graphical representations of the calculated data to include in the summary.

6.3 Composition

The main parts of report generation are data analysis and visualization. Node data will be obtained from the .csv files generation in the data collection component. The accuracy of a patient's movements will be calculated by comparing their node data to the expected, or correct, movements' data. Accuracy will be calculated for each session, and it will be plotted on a graph to see the patient's progress over time. Duration of exercises will also be plotted for each session. Various statistics such as average, highest, and lowest accuracy rates and durations will also be calculated. Once these values have been calculated, they will be used by the visualization component to plot the data and represent it in a graphical display. Then, the data visualizations and statistics will be displayed on the data summary page.

6.4 Interactions

Report generation requires interaction from the user interface to display the data summary. This is how the user will demand the output from this component. Report generation also interacts with the data visualization component by providing input for the output needed. The data visualization component then provides the report generation component with the necessary visualizations to deliver the data summary to the user.

7 DATA VISUALIZATION

7.1 Context

Data visualization will be a part of the data summary/report. It will take in information from the data analysis used to generate the report and use it to generate a graphical display of the results. The visualizations will be included in the data summary.

7.2 Interactions

Data visualization is dependent on the analysis of the collected node data. As part of the report generation component, the visualization component will use the results of data analysis in the report generation component to generate visualizations for the summary.