TECHNICAL DOCUMENTATION

Klara Renström

Version 0.1

Status

|  |  |  |
| --- | --- | --- |
| Reviewed |  |  |
| Approved |  |  |

PROJECT IDENTITY

Project group number 2, MED3/term 7, RunKinect

Tekniska högskolan at Linköping University

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Responsibility** | **Telephone** | **E-mail** |
| Erika Drageryd | Project manager | 073-140 01 40 | eridr093@student.liu.se |
| Klara Renström | Secretary | 073-050 38 22 | klare519@student.liu.se |
| Nima Behnam | Developer Manager | 073-683 00 90 | nimbe760@student.liu.se |
| Martin Bäckström | Developer | 072-233 75 53 | marba607@student.liu.se |
| Alexander Börjesson | Developer | 073-071 37 10 | alebo916@student.liu.se |
| Johan Tidholm | Developer | 076-894 09 16 | johti626@student.liu.se |

**Customer:** Minovi, Teknikringen 7, 583 30  LINKÖPING,   
customer phone 073-383 90 82, info@minovi.se  
**Customer contact:** Göran Salerud, office 401, tel. 013-286755, goran.salerud@liu.se

**Course examiner**: Göran Salerud, office 401, tel. 013-286755, goran.salerud@liu.se  
**Mentor:** Marcus Larsson, tel. 013-286751, marcus.larsson@liu.se

**Abstract**

Contents

[1 Introduction 6](#_Toc355789284)

[1.1 Background 6](#_Toc355789285)

[1.1.1 How a Kinect works 6](#_Toc355789286)

[1.1.2 The ECG monitor 8](#_Toc355789287)

[1.2 Purpose and aim 9](#_Toc355789288)

[1.3 Limitations 10](#_Toc355789289)

[1.4 Method 10](#_Toc355789290)

[1.4.1 Before phase 10](#_Toc355789291)

[1.4.2 During phase 11](#_Toc355789292)

[1.4.3 After phase 11](#_Toc355789293)

[2 Result 11](#_Toc355789294)

[2.1 Finding angles in joints 11](#_Toc355789295)

[2.2 Finding the speed and incline of the treadmill 12](#_Toc355789296)

[2.3 Connecting and disconnecting to the ECG monitor via Bluetooth 13](#_Toc355789297)

[2.4 Presentation of test results 14](#_Toc355789298)

[3 Discussion 14](#_Toc355789299)

[3.1 Development potential 14](#_Toc355789300)

[4 sources 14](#_Toc355789301)

[4.1 Printed sources 14](#_Toc355789302)

[4.2 Electronic sources 14](#_Toc355789303)

Document history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Eventual changes** | **Done by** | **Reviewed** |
| 0.1 | 2013-01-24 | First version | NB, MB, AB, ED, KR, JT |  |

# Introduction

Skriv något o nämn att Kinect används.

## Background

Every person has a different way of moving around, a different moving pattern. It is influenced from parents to their children and from other inputs that we are exposed to. Ways of doing a simple thing, as carrying a handbag, can vary a lot between different people. One way can seem simple and normal to someone while it seems odd and sometimes even painful to someone else.

By analyzing the movement pattern of an individual, interesting information can be gained. This information can help the individual to a better way of moving. A better way can be more relaxed, less painful or more efficient. These ways of moving can have impact on a persons heart rate and breathing pattern. But changing a moving pattern is changing a habit and that can be hard[[1]](#footnote-1) and feel unnatural. To establish a new pattern can take over 300 repetitions depending on how complex it is, which might seem quite remarkable. Also, the age of the individual has a big impact on the difficulty. It is easier for a young person, a child for example, to change his/hers behavior and way of moving, than for an adult.[[2]](#footnote-2)

Analyzing a movement and finding out if changing it is necessary can be hard to do, especially when the movement occurs in a high speed. The timing in the body segments stays the same no matter how fast the move is and that gives us the opportunity to slow down the movement and analyse it when it is easier to spot the issues and fix them.[[3]](#footnote-3) When the speed and the load of an imperfect movement increases, the more the imperfection will be shown.[[4]](#footnote-4) A good way to see this is by using a camera. A Kinect camera can take up to 30 frames per second[[5]](#footnote-5) and this can help spotting the imperfect movement. When a person is jumping right up it can be hard to see any imperfections but if you have several pictures of it, it is easier to see what is happening. With this way you can see if the person is knocking his/her knees together, which can be a sign of a weak hip. With this information, new helpful exercises can help the person and prevent injuries in the knees and back.

Analyzing the movement pattern can have huge impact on individuals. It can help them in their daily life by making different movements more efficient and preventing them from getting injuries.

### How a Kinect works

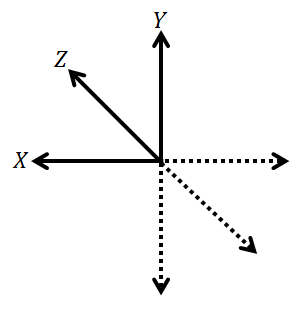
In november 2010 Microsoft first released, after years of speculations, rumors and a prereleases, a new accessory to the Xbox 360 that would change the way gamers played. This was a way to play that did not require a controller, instead you would control the game with your body. This new accessory was called Kinect.[[6]](#footnote-6) The Kinect camera uses a complex system of sensors, transmitters and cameras to detect and show how a person’s moves. This is done with the help of an infrared (IR) projector, two cameras and four microphones. Starting from the right, no 1 an IR projector which emits infrared light into the room. Second comes the color camera used to collect RGB-data. It functions as a regular webcam and supports a resolution of 1280 x 960 pixels. The third one is an IR-camera. This camera collects the depth data, it detects the infrared light when it bounces back towards the camera. It supports resolution of 640 x 480 pixels.[[7]](#footnote-7) The Kinect camera also has a microphone array that consists of four microphones placed on the bottom of the camera. One is underneath the IR transmitter and the other three is evenly spaced on the right side of the depth camera[[8]](#footnote-8). However, the microphones were not used in this project.

#### The Kinect SDK

Microsoft have created a Software Development Kit (SDK). The SDK is a set of libraries that allow us to program applications with the Kinect as input. With the help of these libraries it is relatively easy to access the Kinect cameras image-, depth- and skeleton streams. With this data you can to develop any number of applications.[[9]](#footnote-9)

#### The Kinect skeleton space

From the Kinect camera’s point of view, the room coordinates for the skeleton space looks like this:



The positive z-axis is pointing out of the camera, the positive x-axis is pointing to the left and the positive y-axis is pointing upwards. The boundaries (in m) are [-2.2, 2.2] for the x-plane, [-1.6, 1.6] for the y-plane and [0, 4] for the z-plane.

### The ECG monitor

Our heart is, among other things, a complex electrical system called the cardiac conduction system. This system consists of three different parts: the sinoatrial node (S-V node), the atrioventricular node (A-V node) and the His-Purkinje system. These parts cooperate to create and spread the necessary electrical potential through the heart in the appropriate manner. The heart contraction begins with an electrical build-up in the S-V node. The signal is then spread through the heart with the help of the A-V node and the His-Purkinje system that makes sure that the different valves and walls contract in the right order. (<http://www.bostonscientific.com/lifebeat-online/heart-smart/electrical-system.html> 2013-04-25)

The electrical signals in the heart can be measured in different ways e.g. electrodes attached on the person's body, a transmitter secured around the chest with a strap. The detected signal is then transmitted to a watch or some other equipment that can detect, process and display the desired data. (<http://wiki.answers.com/Q/How_does_a_heart-rate_monitor_work> 2013-04-25 eller <http://www.freescale.com/files/microcontrollers/doc/app_note/AN4059.pdf>)

The pulse monitor provided to the group is a two-lead ECG-monitor secured with a strap around the chest. The leads are connected by wire to a pad where a module is placed. The module transmits the acquired signals via Bluetooth to a server/nearby computer. The ECG data is then, if necessary, filtered and presented on a chart by a program.  


*Image 1 - Back of pulse monitor with highlighted ECG leads*



*Image 2 - Pad where transmitter module is placed*  


*Image 3 - Transmitter module*

## Purpose and aim

The projects main purpose was to create a running lab where different running styles, related to the runners physical health, were exposed. The project should result in a model that could help analyzing and improving running movements. The aim was to set up a model around a treadmill with optical registration of body movements and wireless recording of physiological measures. These two was to be connected to one single system for easy usage for any operator. The result should be presented in real-time.

Important body movements was how different angles between joints in the body changed during running. The runners pulse and ECG were important physiological measures and should be shown in the program.

The projects outcome was uncertain because project group had such limited programming skills and that the project customer had little experience in these type of projects involving mainly software development. Nobody involved in the project knew how the problems were going to be solved and exactly what level of knowledge was required to solve them.

## Limitations

Limitations in the project, disregarding the project groups competence, were (among other things) limitations in the optical instrument. The number of instruments needed were limited to two and the instruments hardware and software also had limitations. The access of experts were limited and their knowledge as well. This depends on that the required knowledge was uncertain in the beginning. Other limitations were the maximum speed and incline of the treadmill. The sensitivity of the pulse band also contributed limitations to the project.

## Method

### Before phase

*The project didn’t have any given question but questions raised was if it was even possible to create such a system and in that case, what was the best way to do it? What optical instrument was the best to use? How could the runners pulse and ECG be presented in the most optimal way?*

In the before phase of the project the group began with the creation of a group contract. This document defines the distribution of responsibilities among the group members and how the group works together and it can be seen in *appendix A*. There were meetings with the customer about the idea of the project and what the group was supposed to do. This resulted in the writing of a project plan, including a timeplan *(appendix B)*, a specification of requirements *(Appendix C)* and a design specification *(appendix D)*. These documents were approved by the customer and in certain cases the mentor.

The first thing the group did was decide on how the requirements would be resolved. A programming language and optical instrument was chosen. Since the pulse monitor used Bluetooth for data exchange, a programming language with a Bluetooth standard library or a Bluetooth API would have to be chosen for the project. Suggested languages would then be C, C++, C#, Java, Python, VB.NET and many more. The customer suggested using Kinect sensors as part of the solution. After research of alternatives lead by the project developer manager, the Kinect sensor was chosen as part of the solution because of its superiority over other solutions by also providing depth and motion aside from a video feed[[10]](#footnote-10). Since parts of the Kinect project are closed source[[11]](#footnote-11), the most effective way of programming with the sensors would be by using the official Microsoft Kinect SDK. This would limit the number of choices to the .NET languages C#, C++/CLI and Visual Basic .NET. The project developer manager then decided to write the code in C# because of its C-like syntax and familiar object-orientation seen in previous courses taken by the group members.

### During phase

The during phase began with a literary study. Everybody in the group read about the c# language and did tutorials to learn how the programing language was build and how it functioned. Different tutorials were done to learn different code basics7.These included operators, types, classes and other vital basics*.[[12]](#footnote-12)* Then the group read about the hardware of the Kinect, how the camera is build, how it works and the basics concerning how to program it. The basics of the available SDK was studied with the help of the book *Beginning Kinect programming with the Microsoft Kinect SDK* by Jarret Webb and James Ashleyand the interesting parts were read more thorough. The group also studied how to connect two cameras into the same system.

Then the group discussed what different components that had to be developed to make a system that fulfilled the specification of requirements. A plan describing who would be working with which parts of the project was also formed and the group started working with different parts of the system simultaneously.

The group had a meeting with the customer and renegotiated some of the requirements. This was done late in the project, with only five weeks until deadline. Though it was necessary since the project group discovered that some requirements was impossible to fulfill depending on the equipment itself. Other changes in the requirements was the level of priority and relevance.

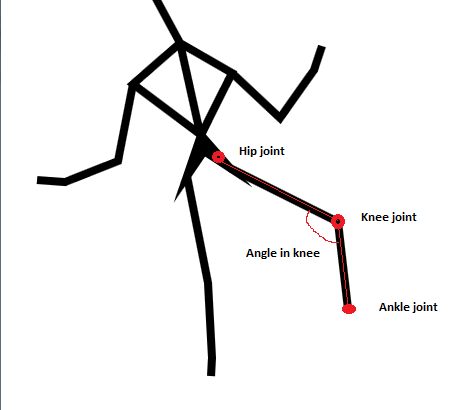
### After phase

After the project is finished the delivery of the product will take place, after which necessary documentation will be supplied to the customer. A technical report, a reflection report and a user manual will be written. The project will be tested in regard to the specification of requirements. The project is then finished after an evaluation by each of the group members.

# Result

A big part of this project was solving different problems using c#. Below there is a short description showing how we have solved the bigger problems in the project. The result is also shown in the commented source code in appendix **XX**.

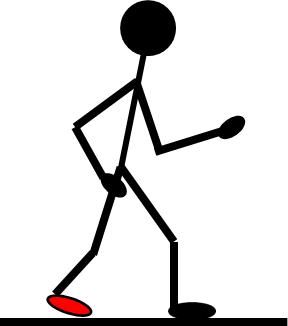
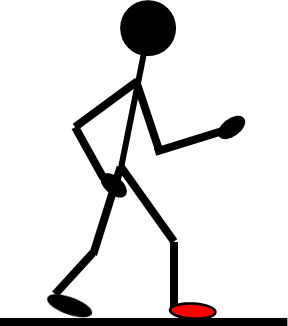
## Finding angles in joints

To find the angle between two joints in the runners lower body we used data from the Kinect skeletonstream.The data needed is the X, Y and Z position of three joints. If you e.g. want the angle in the knee you use knee, ankle and hip. There is a vector drawn between knee and ankle as well as knee and hip and scalar product is used to find the angle between the vectors. This is described in figure XX below

*Figure XX Image describing how the angle in the knee is found*

## Finding the speed and incline of the treadmill

To find the speed and incline of the treadmill, the Kinect skeleton stream was used which allowed comparison of joint position between different frames. The development of the method for this purpose was based on the assumption that the feet have the same velocity as the band of the treadmill as long they are connected. Therefore it seemed feasible to determine the speed if knowing when and where a foot hits and takes off from the band during a step. The speed of the foot (and therefore the treadmill) should then be the distance between these two points divided with the time elapsed between their recording. The concept is illustrated below:

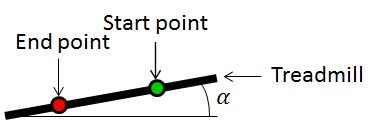


*Figure X. The starting position for the measurement is highlighted in red in the left figure. The same is done close to half a step cycle later for the right figure with the end position for measurement.*

The first approach was to determine momentaneous velocities by comparing the current frame with the previous frame. When the x-velocity of the foot under consideration turns positive, the foot should be moving backward (see section The Kinect skeleton space).This indicates that the foot is about to or has hit the band. When the y-velocity of the foot turns positive, the foot should be moving upwards. Which indicates that the foot is about to or has took off from the band. However, this approach gave unreliable, though reasonable, results. This is probably because of the skeleton data being too “jumpy” for comparison between a frame and the preceding one. Since the time between frames are so short (mostly around 30 ms), even slightly misgiven coordinates might have indicated velocities that in reality would have the opposite sign.

The final approach was to base the decision of when to start and stop measuring by investigating the distance between the feet. This turned out to be a significantly more reliable solution. The code performing this determines when the distance between the feet goes below 0.5 m, and also checks if the foot considered is in front of the other and if the distance between the feet in the previous frame was longer. The considered foot was at first the runner’s right foot (closest to the Kinect) but was changed to the left foot since this proved to give better results. Possibly because the Kinect being more accurate at that distance. When the code has found this starting point, the position of the left foot in this frame is recorded and the built-in Timestamp property is used to record the time the frame was created. The code then checks each preceding frame until a frame shows feet distance longer than 0.25 m, left foot being behind the right and the feet distance from the previous frame being shorter. This indicates that the foot has followed the treadmill band backwards, the position of the left foot (the end point) is then recorded along with the Timestamp of the frame. With simple linear algebra functions the distance between the start point and the end point can then be calculated alongside with the time difference between when the frames they have been collected from. There are also some filtering functions in the code to get rid of very unlikely results. Also, the value shown on the screen is a mean value from ten measurements with the two lowest and two highest values not included.

The plan for determining the angle of the treadmill was essentially the same as for the speed in the section above. Apart from not using the Timestamp property and just using trigonometric functions to get the angle between the start point and end point. The concept is shown in the image below:



*Figure X. The angle of the treadmill, α, relative to the horizontal plane*

However, determining the angle proved to be more tricky than determining the speed. The uncertainty in the position coordinates seemed more

## Connecting and disconnecting to the ECG monitor via Bluetooth

The Bluetooth address of the ECG monitor was found and used to create an endpoint. A Bluetooth service is declared in the program and the endpoint is used to connect to the specific device, in this case the ECG monitor, by utilizing Connect method. Disconnection is achieved by using the Disconnect method, which disposes the Bluetooth service.

## Presentation of test results

# Discussion

**Other options**

One other method would be using a regular camera. Then the group would have to write a software that could recognize different parts of the body via some kind of object recognition. This software would have to be able to recognize body parts and specific joints in a persons body and be able to create a skeleton representation from this data.

Without the depthstream (IR transmitter and receiver) available in the Kinect it would be hard to create a 3D representation of a skeleton. To find how far away an object is you would have to detect shaded areas and how they move in respect to a known light source.[[13]](#footnote-13) This is very complicated due to several reasons, one being that the reference light always has to be the same for you to get a correct comparison.

## Subsystem 1

Most parts of subsystem 1 works as intended as can be seen in the test protocoll. The thing that the group has been unable to achieve is detecting the incline of the treadmill in a satisfactory way. The solution the group has tried is similar to detecting speed in the treadmill but comparing the distance in z as well. This is shown in the result as well. Another approach ha been using object tracking to find to specified objects located on the treadmill and then using code similar to the one that detects angles in joints to find the angle.

Finding two specified objects is difficult. You would have to specify how the object looked and this could change dependeing on e.g. the light in the room. You also had to know the exact distance between the object and the camera. Also, this type of programming is advanced and on a knowledge level way above the one in the project group.

The group also wanted to create a 3D representation using data from both Kinect cameras simultaneously. One thing we are missing is an algorithm that can mearge the data from the two cameras. We can create figures and skeletons but it is difficult to update the graphic because there is something wrong in the event handler. It would have been possible to complete this if the group had more time and resources. If this 3D-represantation would have worked properly all our other functions would have given better and more exact measurements.

In the beginning the group only had one Kinect camera available. We used the strategy that everybody that would be using the camera learned how to do the basic programming on a Kinect. This resulted in five people trying to use the same camera to test their written code which lead to a lot of waiting time and the project did not progress. In this stage it would have been good if the mentor had some experience in how to do this kind of programming project. As an example the plotting needed to display angles in joints did not need any knowledge of the Kinect. the same can be said about finding an object and parts of the 3D-programming.

From more than this aspect it would have been beneficial to have a mentor more familiar with c# programming so the group had someone to ask about different programming issues.

## Subsystem 2

As for subsystem 2, one could discuss why plotting of the ECG couldn’t be achieved. The subsystem can handle connection and disconnection from the pulse monitor, but the problem is somewhere within the area of handling the data stream. The data stream is stored within a variable in the program which in turn is sent to other methods for manipulation and representation. At first, methods were written to examine whether any data was received at all. This examination was successful, with the result being data chunks being received, which varied in size, more specifically between 2-4 bytes each. After then trying to convert these individual parts of the stream to bytes which would be stored in an array for further processing, something went wrong. After threading the methods manipulating the parts and still getting no results, the program was rewritten, and the data stream was stored in a text file for examination. After examining the output text file, it was discovered that there was no output. One could argue that there was something wrong with the storage/conversion methods, however these were tested before use. One could then suspect that the pulse monitor was sending empty bytes. And that would limit the power of what the project group could do.

The problem may have been solved by using for instance, another API or more research. But given the time, this would limit the resources for programming, thus it wouldn’t have been the optimal solution. Another noteworthy fact is the prior knowledge of members of the group when entering the project. Basically, one course in object-oriented programming was taken by the majority of the group before entering the project. One could then lobby for better preparation and education of group members before entering the project, which today seems like something that would’ve been useful. Also, taking the spent time on the subsystem in fact, the resources given to the project group should be reconsidered into giving more. There is much room for improvement in the software, which will be discussed later. Given time, many improvements can be made, which of course will lead to a more optimal software as a result.

*Is there a better way to approach/solve the problem?*

The project has provided an extensive insight into the C# programming language and object-oriented programming, specifically of the Kinect- and the 32feet.NET Bluetooth application programming interfaces. The project has also provided extensive insight into that of working in a group, with every member having to take an area of responsibility and everyone depending on one another, emphasizing cooperation within the group; and of course, experience of spending given resources as effectively as possible.

## 

## Development potential

* Hur långt har vi hunnit?
* Varför har det inte gått som vi tänkte oss?
* Hur kan man gå vidare med detta?
* Är det ett krav som vi inte lyckats uppfylla?

*Making a 3D representation of the runner*

**Discussion**

* Is there a better way to approach/solve the problems in the project?
* What have we learned?

**Stroke rehabilitation - a development**

* Movement improvement - training balance and coordination
  + Make rehabilitation fun with Kinect games
* Help people who are unable to e.g. type to use computers.

# sources

## Printed sources

## Electronic sources

1. <http://www.sund.nu/docs/artikel.asp?tem=30&art=282>, 25/4 2013 [↑](#footnote-ref-1)
2. <http://www.ptdirect.com/training-design/training-fundamentals/movement-mechanics-and-motor-learning/the-primary-movement-patterns>, 25/4 2013 [↑](#footnote-ref-2)
3. <http://www.ptdirect.com/training-design/training-fundamentals/movement-mechanics-and-motor-learning/the-primary-movement-patterns>, 25/4 2013 [↑](#footnote-ref-3)
4. <http://www.ptdirect.com/training-design/training-fundamentals/movement-mechanics-and-motor-learning/the-primary-movement-patterns>, 25/4 2013 [↑](#footnote-ref-4)
5. <http://msdn.microsoft.com/en-us/library/jj131033.aspx> 2013-05-01 [↑](#footnote-ref-5)
6. <http://news.cnet.com/8301-10805_3-20035039-75.html> 2013-04-23 [↑](#footnote-ref-6)
7. *Beginning Kinect programming with the Microsoft Windows SDK* sid 10 [↑](#footnote-ref-7)
8. B*eginning Kinect programming with the Microsoft Windows SDK* sid 9 kan vara ibid. 9 om de ligger i samma ordning i slutskedet [↑](#footnote-ref-8)
9. B*eginning Kinect programming with the Microsoft Windows SDK* sid 9 [↑](#footnote-ref-9)
10. <http://www.microsoft.com/en-us/kinectforwindows/discover/features.aspx> 2013-05-07 [↑](#footnote-ref-10)
11. <http://www.bit-tech.net/news/2013/03/13/microsoft-reveals-kinect-source-code/> 2013-05-07 [↑](#footnote-ref-11)
12. Tutorials taken from <http://csharp-station.com/> [↑](#footnote-ref-12)
13. [http://s](http://stackoverflow.com/questions/7232400/object-shape-recognition-from-webcam)[t](http://www.ptdirect.com/training-design/training-fundamentals/movement-mechanics-and-motor-learning/the-primary-movement-patterns)[ackoverflow.com/questions/7232400/object-shape-recognition-from-webcam](http://stackoverflow.com/questions/7232400/object-shape-recognition-from-webcam) [↑](#footnote-ref-13)