

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond  
Tarkvarateaduse instituut

Jan-Joonas Bernstein  
142316

# **KONTEKSTIPÕHINE INIMLIIGUTUSTE REGISTREERIMINE JA ANALÜÜS**

Bakalaureusetöö

Juhendaja: Sven Nõmm  
PhD

Tallinn 2017

## **Autorideklaratsioon**

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Jan-Joonas Bernstein

.....  
(kuupäev)

.....  
(allkiri)

## **Annotatsioon**

Käesoleva bakalaureusetöö eesmärgiks on inimliigutuste registreerimist ning analüüsi võimaldava tarkvara väljatöötamine. Rakendus võimaldab registreerida kolme erinevat tüüpi neuroloogias Parkinsoni tõve diagnoosimiseks ja haiguste modelleerimiseks kasutatavat harjutust. Rakendus suudab eristada ja tuvastada nende harjutuste korduseid ning faase. Arendatav tarkvara vähendab meditsiinipersonali sekkumise vajadust patsiendi motoorikat kontrollivate harjutuste juures. Lisaks on võimalik kontekstide registreerimise järgselt saadud tulemusi analüüsida – arvutada erinevad massiparameetreid, liigete vahelisi nurki, nurkade ekstreemumeid ning massiparameetrite suhteid. Kontekstide faasid ning harjutuste kordused, mida rakendus implementeerib, tuvastatakse harjutuse salvestamise käigus automaatselt. Lisaks võimaldab rakendus salvestatud harjutusi järgi vaadata ning analüüsitulemusi üles laadida. Rakendus kasutab inimliigutuste salvestamiseks Kinect for Xbox One välisseadet. Töö tulemusena valmis rakendus, mis täidab püstitatud eesmärke ja funktsionaalsuseid. Rakenduse valiidsust kontrolliti viies läbi programmi poolt kontrollitavaid harjutusi, võrreldes neid samaaegselt salvestatud vanema süsteemi tulemustega, mis tõestasid programmi töötavust.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 67 leheküljel, 3 peatükki, 34 joonist.

## **Abstract**

### **Context based registration and analysis of human motions**

The aim of this bachelor's thesis is the development of a software for context based registration and analysis of human motions, which would reduce the need of intervention of medical personnel while testing human motility skills.

The software enables the user to register three different types of exercises and their phases used in neurology for the diagnosis Parkinson's disease. In addition, after recording the contexts, it is possible to analyze the results – to calculate motion mass parameters, angles between different, joints' angular extremes and the ratios of different motion mass parameters.

During the recording session, the contexts and its phases, which the software implements, are detected automatically. Furthermore, the software enables the user to replay previously recorded exercises and to upload analysis reports to Amazon Cloud Services. For detection and recording of human motions, the application uses Microsoft's Kinect for Xbox One.

The result is an application that meets the previously set requirements and objectives. The validity of the software was verified and checked by carrying out the context based exercises in parallel with an older, but similar piece of software, which proved the reliability of this application.

The thesis is in Estonian and contains 67 pages of text, 3 chapters, 34 figures.

## Lühendite ja mõistete sõnastik

<b>Kinect for Xbox One</b>	Microsoft'i välisseade, mis võimaldab inimkeha kolmemõõtmelist tuvastamist kasutades selleks infrapuna sensorit.
<b>Up and go</b>	Ehk „tõuse ja mine“ on kontekstitüüp, mis seisneb inimese toolilt tõusmisel, kõndimisel otse Kinect kaamera poole, ümber pööramisel, tagasi kõndimisel ning uuesti maha istumisel.
<b>Up and sit</b>	Ehk „tõuse ja istu“ on kontekstitüüp, mis seisneb inimese toolilt püsti tõusmisel ning uuesti toolile istumisel.
<b>Go</b>	Ehk „mine“ on kontekstitüüp, mis seisneb inimese kõndimisel Kinect kaamera suunas, ümber pööramisel, tagasi kõndimisel ning uuesti ümber pööramisel.
<b>Visual Studio</b>	Microsoft'i poolt loodud arenduskeskkond Windows'i rakenduste loomiseks.
<b>WPF</b>	<i>Windows Presentation Foundation</i> on arenduskeskkonna Visual Studio alamsüsteem, mis võimaldab mugavalt luua modernseid kasutajaliideseid.
<b>Repetition</b>	Ehk kordus on harjutuse korduste arv ühe salvestamise käigus.
<b>Phase</b>	Ehk faas on vastavalt harjutusele selle harjutuse jooksul toimunud liikumise osa. Näiteks, toolilt püsti tõusmine või edasi kõndimine.
<b>Motion mass</b>	Parameeter, mis kirjeldab mingit liikumisega seotud protsessi parameetri (näiteks kiirendus) absoluutväärtuste summat.
<b>Velocity mass</b>	Parameeter, mis kirjeldab mingis faasis mingi liigese kõikide salvestatud kaadrite jooksul toimunud kiiruste absoluutväärtuste summat. Tähistus (V).
<b>Acceleration mass</b>	Parameeter, mis kirjeldab mingis faasis mingi liigese kõikide salvestatud kaadrite jooksul toimunud kiirenduste absoluutväärtuste summat. Tähistus (A).
<b>Jerk mass</b>	Parameeter, mis kirjeldab mingis faasis mingi liigese kõikide salvestatud kaadrite kiirenduste muutumiste kiiruste absoluutväärtuste summat. Tähistus (J).

<b>Combined trajectory</b>	Parameeter, mis kirjeldab mingis faasis mingi liigese kõikide salvestatud kaadrite kogu teepikkust. Tähistus (T).
<b>Euclidean distance</b>	Parameeter, mis kirjeldab mingi faasi algus- ning lõpppunkti vahelist teepikkust. Tähistus (E).
<b>Time</b>	Parameeter, mis kirjeldab mingi faasi läbimiseks kulunud aega. Tähistus (t).
<b>T/E</b>	Parameeter, mis kirjeldab Combined trajectory ning Euclidean distance'i suhet.
<b>V/E</b>	Parameeter, mis kirjeldab Velocity mass'i ning Euclidean distance'i suhet.
<b>A/E</b>	Parameeter, mis kirjeldab Acceleration mass'i ja Euclidean distance'i suhet.
<b>J/E</b>	Parameeter, mis kirjeldab Jerk mass'i ja Euclidean distance'i suhet.
<b>Hip-Knee-Ankle(Right)</b>	Nurk inimese puusa, põlve ning pahkluu vahel paremal pool keha.
<b>Hip-Knee-Ankle (Left)</b>	Nurk inimese puusa, põlve ning pahkluu vahel vasakul pool keha.
<b>Head-SpineBase-Knee (Right)</b>	Nurk inimese pea, ristluu ning põlve vahel paremal pool keha.
<b>Head-SpineBase-Knee (Left)</b>	Nurk inimese pea, ristluu ning põlve vahel vasakul pool keha.
<b>Knee-SpineBase-Knee</b>	Nurk inimese vasaku põlve, ristluu ning parema põlve vahel.
<b>Head-SpineShoulder-SpineMid</b>	Nurk inimese pea, rinnaluu ning selgroo keskpunkti vahel.
<b>SpineShoulder-SpineMid-SpineBase</b>	Nurk inimese rinnaluu, selgroo keskpunkti ning ristluu vahel.
<b>Head-SpineShoulder-SpineBase</b>	Nurk inimese pea, rinnaluu ning ristluu vahel.
<b>Start</b>	Esimene nurgaväärtus faasi jooksul, mis vastab eelnevalt nimetatud nurkade loetelust valitud nurgatüüpidele.

<b>End</b>	Viimane nurgaväärtus faasi jooksul, mis vastab eelnevalt nimetatud nurkade loetelust valitud nurgatüüpidele.
<b>Min</b>	Minimaalne nurgaväärtus faasi jooksul, mis vastab eelnevalt nimetatud nurkade loetelust valitud nurgatüüpidele.
<b>Max</b>	Maksimaalne nurgaväärtus faasi jooksul, mis vastab eelnevalt nimetatud nurkade loetelust valitud nurgatüüpidele.
<b>Avg</b>	Keskmine nurgaväärtus faasi jooksul, mis vastab eelnevalt nimetatud nurkade loetelust valitud nurgatüüpidele.
<b>Sum</b>	Faasi jooksul iga kahe järgneva kaadri vaheliste nurkade erinevus absoluutväärtusena ning summeritult, mis vastab eelnevalt nimetatud nurkade loetelust valitud nurgatüüpidele.
<b>XAML</b>	<i>Extensible Application Markup Language</i> on Microsoft'i poolt arendatud XML tüüpi märgistuskeel.
<b>C#</b>	Microsoft'i poolt arendatud üldotstarbeline objektorienteeritud programmeerimiskeel.
<b>SDK</b>	<i>Software Development Kit</i> ehk tarkvaraarenduskomplekt on tarkvaraarenduse tööriistade komplekt.
<b>RGB</b>	<i>Red-Green-Blue</i> on liitvärvimudel, kus kolme põhivärvuse, punase, roheline ja sinise, liitumisel saadakse erinevaid värvitoone.
<b>csv</b>	Comma-separated-values. On failitüüp, kus andmed on eraldatud komaga. Sobib näiteks Microsoft Excelis töötlemiseks ja vaatamiseks.

## Sisukord

1 Sissejuhatus .....	13
2 Kirjanduse ülevaade .....	14
3 Probleemi püstitus .....	15
4 Teoreetiline osa.....	17
4.1 Tehnoloogiad .....	17
4.1.1 Tarkvara.....	17
4.1.2 Riistvara.....	17
4.2 Analüüsimeetodid.....	18
4.2.1 Motion mass .....	18
4.2.2 Velocity mass .....	19
4.2.3 Acceleration mass.....	20
4.2.4 Jerk mass .....	20
4.2.5 Combined trajectory .....	21
4.2.6 Euclidean distance .....	21
4.2.7 Time.....	22
4.2.8 Motion masside suhted .....	22
4.2.9 Nurgad .....	23
5 Praktiline osa .....	29
5.1 Faasituvastus.....	29
5.2 Kontekst Up and go .....	29
5.2.1 Istumine konteksti alguses.....	30
5.2.2 Püsti tõusmine .....	30
5.2.3 Seismine peale püsti tõusmist.....	30
5.2.4 Sammumise alustamine .....	31
5.2.5 Ümber pööramine kaamera juures.....	32
5.2.6 Tagasi kõndimine .....	32
5.2.7 Ümber pööramine tooli ees .....	33
5.2.8 Seismine tooli ees peale ümber pööramist .....	33
5.2.9 Maha istumine .....	34



5.2.10 Istumine konteksti lõpus.....	34
5.3 Kontekst Up and sit.....	35
5.3.1 Istumine .....	35
5.3.2 Püsti tõusmine .....	35
5.3.3 Seismine .....	35
5.3.4 Maha istumine .....	36
5.3.5 Istumine konteksti lõpus.....	36
5.4 Kontekst Go .....	36
5.4.1 Seismine .....	36
5.4.2 Sammumise alustamine .....	37
5.4.3 Ümber pööramine kaamera juures.....	37
5.4.4 Tagasi kõndimine .....	37
5.4.5 Ümber pööramine alguses .....	37
5.4.6 Seismine alguses.....	38
5.5 Kontekstide salvestamine.....	38
5.6 Analüüs ja analüüsitulemused .....	39
5.6.1 Analüüsitulemused tabeli kujul .....	39
5.6.2 Analüüsitulemused tavalisel kujul.....	40
5.7 Analüüsitulemuste salvestamine.....	40
5.8 Andmetöötlusprotsess.....	41
6 Kasutajaliides .....	42
6.1 Peamenüü.....	42
6.2 Uue konteksti vaade (new) .....	42
6.3 Salvestuste järgi vaatamise vaade (open) .....	43
6.4 Kontekstide analüüsivaade (analyze).....	43
6.5 Üles laadimise vaade (upload).....	44
6.6 Sätete vaade (settings) .....	44
7 Kokkuvõte .....	45
Kasutatud kirjandus .....	46
Märkmed.....	48
Lisa 1 – Inimese püsti tõusmise kontrollmeetodid .....	49
Lisa 2 – Inimese seismise kontrollmeetod.....	49
Lisa 3 – Inimese sammumise kontrollmeetod .....	50
Lisa 4 – Inimese kaamera juures ümber pööramise kontrollmeetod .....	50

Lisa 5 – Inimese tagasi kõndimise kontrollmeetod .....	51
Lisa 6 – Inimese alguses ümber pööramise kontrollmeetod.....	51
Lisa 7 – Inimese tagasi kõndimise järgse seismise kontrollmeetod .....	52
Lisa 8 – Inimese maha istumise kontrollmeetod .....	52
Lisa 9 – Inimese konteksti lõpus istumise kontrollmeetod.....	53
Lisa 10 – Näide salvestatavast konteksti logifailist.....	53
Lisa 11 – Näide analüüsitulemustest tabeli kujul .....	54
Lisa 12 – Näide analüüsitulemustest tavalisel kujul.....	55
Lisa 13 – Failide Amazon’i S3 Bucketisse üles laadimise meetod .....	56
Lisa 14 – Kasutajaliidese peamenüü .....	57
Lisa 15 – Kontekstitüübi valik ning kasutaja andmed.....	58
Lisa 16 – Harjutuse salvestamise vaade .....	59
Lisa 17 – Salvestuste järgi vaatamise vaade.....	60
Lisa 18 – Analüüsitava faili valimise vaade .....	61
Lisa 19 – Analüüsitavate liigete valimise vaade .....	62
Lisa 20 – Analüüsitüüpide valimine .....	63
Lisa 21 – Analüüsitavate korduste ja faaside valimise vaade .....	64
Lisa 22 – Analüüsitulemuste kokkuvõtte vaade .....	65
Lisa 23 – Failide üles laadimise vaade .....	66
Lisa 24 – Sätete vaade .....	67

## Jooniste loetelu

Joonis 1. Kinect for Xbox One .....	18
Joonis 2. Nurk Hip-Knee-Ankle (Right) .....	24
Joonis 3. Nurk Hip-Knee-Ankle (Left).....	25
Joonis 4. Nurk Head-SpineBase-Knee (Right).....	25
Joonis 5. Nurk Head-SpineBase-Knee (Left) .....	26
Joonis 6. Nurk Knee-SpineBase-Knee .....	26
Joonis 7. Nurk Head-SpineShoulder-SpineMid .....	27
Joonis 8. Nurk SpineShoulder-SpineMid-SpineBase.....	27
Joonis 9. Nurk Head-SpineShoulder-SpineBase .....	28
Joonis 10. Andmetöötlusprotsess .....	41
Joonis 11. Inimese püsti tõusmise kontrollmeetodid.....	49
Joonis 12. Inimese seismise kontrollmeetod .....	49
Joonis 13. Inimese sammumise kontrollmeetod.....	50
Joonis 14. Inimese kaamera juures ümber pööramise kontrollmeetod.....	50
Joonis 15. Inimese tagasi kõndimise kontrollmeetod .....	51
Joonis 16. Inimese alguses ümber pööramise kontrollmeetod .....	51
Joonis 17. Inimese tagasi kõndimise järgse seismise kontrollmeetod.....	52
Joonis 18. Inimese maha istumise kontrollmeetodid.....	52
Joonis 19. Inimese konteksti lõpus istumise kontrollmeetod .....	53
Joonis 20. Näide salvestatavast konteksti logifailist .....	53
Joonis 21. Näide analüüsitulemustest tabeli kujul.....	54
Joonis 22. Näide analüüsitulemustest tavalisel kujul .....	55
Joonis 23. Failide üles laadimine Amazon'i S3 Bucketisse .....	56
Joonis 24. Kasutajaliidese peamenüü .....	57
Joonis 25. Kontekstitüübi valiku kasutajaliides .....	58
Joonis 26. Harjutuse salvestamise kasutajaliides .....	59
Joonis 27. Salvestuste järgi vaatamise kasutajaliides.....	60
Joonis 28. Analüüsitava faili valimise kasutajaliides .....	61
Joonis 29. Analüüsitavate liigete valimise kasutajaliides .....	62

Joonis 30. Analüüsitüüpide valimise kasutajaliides .....	63
Joonis 31. Analüüsitavate korduste ja faaside valimise kasutajaliides.....	64
Joonis 32. Analüüsitulemuste kokkuvõtte kasutajaliides .....	65
Joonis 33. Failide üles laadimise kasutajaliides .....	66
Joonis 34. Sätete kasutajaliides .....	67

# 1 Sissejuhatus

Inimese mootorikavõimete jälgimine üle mingi ajaperioodi on väga aktuaalne viis erinevate haiguste, näiteks Parkinsoni tõve diagnoosimiseks. Siiani on Eestis ja mujal maailmas inimmotoorika registreerimise ja analüüsimise läbiviimine toimunud üldiselt kas ainult arsti vaatluse all või lisaks kasutades mõnda vanemat ja vähesemate funktsionaalsustega rakendust. Antud töö on seotud Tallinna Tehnikaülikooli projektiga B37 „Raamistik inimmotoorika kvantitatiivseks analüüsiks“. Käesoleva bakalaureusetöö eesmärgiks on arendada välja selliste protsesside registreerimiseks sobiv rakendus, mis lihtsustaks oluliselt meditsiinipersonali tööd ning tõstaks inimese mootorikat kontrollivate harjutuste läbiviimise efektiivsust, kasutades selleks üsnagi odavat ning kättesaadavat Microsofti poolt arendatud välisseadet Kinect for Xbox One.

Kontekstiks nimetatakse formaalseoste hulka. Tavaliselt luuakse kontekstid selleks, et kitsendada keskkonna võimalikke muutuseid või määrata vaadeldava objekti käitumist. Kontekst kirjeldab käesolevas töös teatud harjutusi, mida patsiendid arstide vaatluse all läbi viivad. Kontekstide abil kirjeldatakse harjutuse faaside järjekorda. Niimoodi on saab lihtsustada erinevate faaside tuvastust – ei tuvastata mitte kõikvõimalikke harjutuse jooksul toimuvaid faase, vaid oodatakse millal lõppeb ja algab kindel ja teadaolev tuvastatav faas. Rakendus võimaldab salvestada ja analüüsida kolme erinevat tüüpi harjutust ning nende jooksul toimuvaid faase. Lisaks, võimaldab rakendus analüüsi tulemusena saadud faile üles laadida, et need oleksid kättesaadavad ning et neid saaks soovi korral edasi analüüsida.

Selleks, et analüüsida mootorikavõimete halvenemist või paranemist, viivad arstid patsientide peal läbi kolme tüüpi katseid: Up and go (tõuse ja mine), Up and sit (tõuse ja istu), Go (mine). Eelnevalt nimetatud harjutusi viiakse läbi üle teatud ajaperioodi, korrates neid kindel arv kordi nädalas ning võrreldes erinevate harjutuste tulemusi. Arendatud rakendus lihtsustab eelnevalt nimetatud protsesside läbiviimist ning võimaldab mugavalt salvestada ja hoiustada harjutuste ja analüüsitude tulemusi.

## 2 Kirjanduse ülevaade

Inimese mootorika ja liigutuste analüüsiks kasutatakse tänapäeval üldiselt keerukaid masinõppe algoritme ning liigutuste mustrite äratundmissüsteeme. Selleks, et mõõta üldist inimese mootorikat on toodud mõiste *motion mass*. Motion mass on kriteerium liigete sujuvuse kirjeldamiseks [1].

Sõna *motion* kirjeldab liigutuste hulka, kus liigutused kirjeldavad skeleti mistahes liigese asukoha muutust üksteise suhtes [1]. Motion mass'i kasutati esimest korda palli viskamise sujuvuse ning õppimise erinevate staadiumide analüüsiks kasutades selleks sarnaselt käesolevas töös kasutatavat Kinect välisseadet [1].

Kinect'i on kasutatud varem ka Parkinsoni tõve patsientide erinevate liigutuste mõõtmiseks: seismise, erinevate esemete haaramise, kõndimise ja väiksemate liigutuste analüüsiks [2]. Tulemuseks avastati, et Kinect on väga täpne suuremate liigutuste mõõtmisel nagu kõndimine, kuid mitte väga täpne väikeste liigutuste mõõtmisel, nagu käe rusikaks tegemine [2].

Võrreldes käesolevas töös kasutatud välisseadet Kinect for Xbox One selle eelkäija, Kinect v1-ga, on uus variant usaldusväärsem ja korduvalt täpsem, eriti Kinect'i sügavuskaamera, mille täpsus on käesolevas töös väga suure tähtsusega [3]. Kinect kaamera on liigutuste tuvastamiseks kõige täpsem vahemikus 0.5m – 3m kaamerast ning suurim ebatäpsus tekib alates 3.5st meetrist [3]. Seetõttu oleks optimaalne viia läbi kõiki harjutusi võimaluse korral optimaalses täpsuse vahemikus.

Inimliigutusi on võimalik mõõta ka mitme Kinect kaameraga korraga. Kasutates mitut kaamerat (kaamerad peaksid asuma nii salvestatava inimese ees kui ka taga) on võimalik luua iga kaamera andmete põhjal inimese liigete asukohtade „pilv“, mis võimaldab kordades täpsemat liigete asukohtade tuvastust. Mitme kaamera kasutamisel on võimalik vabaneda üksiku Kinecti seljaga kaamera poole seisva inimese ebatäpse tuvastamise probleemist [4].

### 3 Probleemi püstitus

Tarkvara peab võimaldama kasutajal mugavalt salvestada valitud harjutuse jooksul toimuvaid inimliigutusi ning nende inimliigutuste analüüsi. Lisaks, peab rakendus võimaldama kasutajal varem salvestatud kontekstifaile järgi vaadata ning soovi korral üles laadida. Rakendus peab töötama koos Kinect for Xbox One välisseadmega. Nende eesmärkide saavutamiseks on vaja lahendada järgmised alamülesanded:

- Kuna Kinect for Xbox One on üsna odav (võrreldes alternatiivsete seadmetega), kättesaadav, mugavalt kasutatav ning üsnagi täpne [2], siis tuleb arendada rakendus, mis on võimeline oma töös kasutama nimetatud välisseadet.
- Rakendust haktakse kasutama meditsiinasutustes kontrollimaks inimese mootorikaoskuseid- ja võimeid. Selleks kasutatakse kolme põhilist harjutust – tõuse ja mine (Up and go), tõuse ja istu (Up and sit) ning mine (Go). Harjutused on kirjeldatud kontekstidena. Harjutused seisnevad formaliseeritud faaside kogumis. Käesolev rakendus peab võimaldama nimetatud harjutuste läbiviimist ning salvestamist.
- Selleks, et muuta kasutajakogemus paremaks ning meditsiinipersonali sekkumist üldiselt vähendada, siis võrreldes varem Eestis kasutatud rakendusega, kus faaside eristamiseks oli vajalik inimespoolne (arsti) nupuvajutus, tuleks arendada rakendusse automaatne faaside ning korduste tuvastus ja eristus vastavalt valitud kontekstile.
- Harjutusi viiakse patsientide peal läbi üle teatud ajaperioodi ning selleks on vajalik erinevate harjutuste tulemuste salvestamine ning võrdlemine. Rakendus peab võimaldama tulemusi faili salvestada ning neid järgi vaadata.

- Lihtsalt salvestamisest ei piisa, vajalik on lisaks sellele lisada ka analüüsimise funktsionaalsus. Analüüsida peaks saama varem salvestatud kontekste. Analüüs peaks võimaldama arvutada valitud liigete jaoks motion mass parameetreid ning erinevate liigete vahelisi nurki, ekstreemumeid ning suhteid.
- Soovitud analüüsitulemusi peab rakendus võimaldama üles laadida pilvesse, et neid oleks hiljem mugavam kätte saada ning tulemusi võrrelda. Selleks peab rakendus suutma valitud faile laadida Amazoni pilveteenustesse.
- Loodud rakenduse kasutajaliides peaks olema võimalikult kasutajasõbralik. Kasutajaliides peaks edukalt peitma rakenduse sisemuses oleva keerukuse ning nii muutma selle kasutamise mugavaks.



## **4 Teoreetiline osa**

### **4.1 Tehnoloogiad**

#### **4.1.1 Tarkvara**

Rakendus on kirjutatud keeles C# ning kasutades selleks arenduskeskkonda Visual Studio 2015. Rakenduse modernse välimuse saavutamiseks kasutati Visual Studio alamsüsteemi WPF, mis võimaldab luua kasutajaliideseid kasutades selleks XAML märgistuskeelt programmiakende välimuse ja selle komponentide kirjeldamiseks.

Selleks, et rakendus oleks võimeline suhtlema Microsoft'i välisseadmega Kinect for Xbox One, kasutatakse samuti Microsoft'i poolt pakutavat SDK'd Kinect for Windows SDK 2.0 [17]. Amazoni pilveteenustega suhtlemiseks kasutab rakendus tarkvaraarenduskomplekti AWS SDK for .NET [16].

#### **4.1.2 Riistvara**

Käesolev rakendus kasutab inimese liigete salvestamiseks ja tuvastamiseks Microsoft'i poolt arendatud välisseadet Kinect for Xbox One. Kinect on kaamera, mis sisaldab endas infrapunasensorit ehk sügavuskaamerat, mis võimaldab jälgida inimest kolmemõõtmelises ruumis olenemata valgustingimustest. Lisaks, sisaldab Kinect mikrofoni ning RGB värvikaamerat. Võrreldes oma vanema versiooniga (Kinect v1), võimaldab selles töös kasutatav Kinect'i uuem versioon salvestada korraga 25te liigest, võrreldes varasema 20ga [18]. Personaalse arvutiga (arenduse jooksul kasutati operatsioonisüsteeme Windows 8.1 ning Windows 10) ühendamiseks ja suhtlemiseks vajab Kinect for Xbox One lisaadapterit.



Joonis 1. Kinect for Xbox One

## 4.2 Analüüsimeetodid

Rakendus võimaldab salvestatud kontekstifailide peal läbi viia analüüsi. Analüüs seisneb motion mass parameetrite, motion mass parameetrite suhete, liigestevaheliste nurkade ning nurkade ekstreemumite arvutamisel, kusjuures võimaldab rakendus kasutajal läbi kasutajaliidese analüüsitavaid liigeseid, analüüsimeetodeid, korduste numbrit ning faase valida.

### 4.2.1 Motion mass

[1] Motion mass on kriteerium, mis kirjeldab inimese liigutuste sujuvust. Liigutuste sujuvus ja motion mass võimaldab kirjeldada inimliigutuste tehnika efektiivsust. Kui *liigutus* (movement) tähendab inimese mistahes liigese füüsilist asukoha muutust mõne teise inimese liigese suhtes, siis sõna *motion* (sõnast *motion mass*) kirjeldab nende sündmuste hulka. Selline klassifikatsioon võimaldab motion mass andmete kui hulga peal viia läbi mitmeid analüüse, mille tulemusena on võimalik hinnata motion mass'i kriteeriumi põhilist eesmärk-parameetrit – sujuvust.

Iga kaader, mida Kinect rakendusele tagastab sisaldab endas X, Y ja Z koordinaate iga 25 liigese kohta. Järgnevate arvutuste jaoks olgu punktid  $P$  liigese koordinaatide ning ajaga inimkehal defineeritud järgmiselt:

$$P_n = \{t_n, x_n, y_n, z_n\}$$

$$P_{n+1} = \{t_{n+1}, x_{n+1}, y_{n+1}, z_{n+1}\}$$

kus  $t$  on aeg millisekundites alates salvestamise algusest,  $x$  on koordinaat  $x$ -teljel,  $y$  on koordinaat  $y$ -teljel,  $z$  on koordinaat  $z$ -teljel ning kus  $n$  tähistab vaadeldava kaadri numbrit ja  $n+1$  sellele vahetult järgneva kaadri numbrit. Punkt  $P$  tähistab mingit kindlat liigest.

Iga kahe järjestikuse punkti vaheline aeg  $\Delta t_n$  arvutatakse:

$$\Delta t_{n+1,n} = \frac{t_{n+1} - t}{1000} (s)$$

Jagamine 1000ga toimub seetõttu, et programm salvestab aega millisekundites alates salvestamise algusest ning kuna velocity mass on antud kujul  $m/s$ , acceleration mass kujul  $m/s^2$  ja jerk mass kujul  $m/s^3$ , siis sekunditeks saamisel tuleb saadud tulemus läbi jagada 1000ga.

Järgnevalt kirjeldatakse analüüsiks kasutatud parameetrite arvutusi.

#### 4.2.2 Velocity mass

**Velocity mass** on motion mass'i parameeter, mis kirjeldab ühe korduse ja ühe selle korduse faasi jooksul toimunud kõikide kaadrite mingi kindla liigese koordinaatide ning aja põhjal arvutatud kiiruste kogusumma absoluutväärtust. Lisaks, tuleb mainida, et iga faasi alguses arvutab programm esimese kaadri kiiruseks 0 m/s. Velocity mass'i tähistuseks on  $V$ .

Olgu meil vaatluse all üks suvaline liiges koos oma kõikide punktide ning nende salvestusaegadega mingi faasi  $F$  jooksul, kus kaadrite koguarv on  $m$  ning vaatluse all olev kaader on  $n$ , siis velocity mass arvutatakse järgmiselt:

Esmalt arvutatakse iga kahe järjestikuse punkti vaheline kaugus  $d$ :

$$d_{n+1,n} = \sqrt{(x_{n+1} - x_n)^2 + (y_{n+1} - y_n)^2 + (z_{n+1} - z_n)^2}$$

Teades igat kahe järjestikuse punkti vahelist kaugust ning aega, arvutatakse velocity mass:

$$V_F = \sum_{n=0}^{m-1} \left| \frac{d_{n+1,n}}{\Delta t_{n+1,n}} \right| \left( \frac{m}{s} \right)$$

### 4.2.3 Acceleration mass

Sarnaselt velocity mass'ile, kirjeldab **acceleration mass** samuti ühe korduse ja ühe selle korduse faasi jooksul toimunud mingi kindla liigese koordinaatide ning aja põhjal arvutatud parameetrit – kiirenduste kogusumma absoluutväärtust. Acceleration mass'i tähistus on A.

Olgu meil teada kahe järjestikuse punkti kiirused  $v_n$  ja  $v_{n+1}$  ning aeg  $\Delta t_{n+1,n}$  mingi faasi F jooksul, kus faasi F kaadrite koguarv on  $m$  ning vaatluse all olev kaadri number on  $n$ , siis acceleration mass arvutatakse järgmiselt:

$$A_F = \sum_{n=0}^{m-1} \left| \frac{v_{n+1} - v_n}{\Delta t_{n+1,n}} \right| \left( \frac{m}{s^2} \right)$$

### 4.2.4 Jerk mass

**Jerk mass** kirjeldab samuti ühe korduse ja ühe selle korduse faasi jooksul toimunud mingi kindla liigese koordinaatide ning aja põhjal arvutatud parameetrit – kiirenduse muutumise kiiruste absoluutväärtuste summat. Jerk mass'i tähistus on J.

Olgu meil teada kahe järjestikuse punkti kiirendused  $a_n$  ja  $a_{n+1}$  ning aeg  $\Delta t_{n+1,n}$  mingi faasi F jooksul, kus faasi F kaadrite koguarv on  $m$  ning vaatluse all olev kaadri number on  $n$ , siis jerk mass arvutatakse järgmiselt:

$$J_F = \sum_{n=0}^{m-1} \left| \frac{a_{n+1} - a_n}{\Delta t_{n+1,1}} \right| \left( \frac{m}{s^3} \right)$$

#### 4.2.5 Combined trajectory

**Combined trajectory** kirjeldab ühe korduse ja ühe selle korduse faasi jooksul toimunud mingi kindla liigese koordinaatide põhjal arvutatud iga kahe järjestikuse koordinaadi vahelise kauguse summat. Combined trajectory tähistus on  $T$ .

Olgu meil teada mingi faasi  $F$  jooksul toimunud mingi kindla liigese kõik koordinaadid. Olgu vaatluse all oleva kaadri number  $n$ , siis võime tähistada vaatluse all oleva kaadri liigese punkti ning sellele järgneva kaadri liigese punkti järgnevalt:

$$P_n = \{x_n, y_n, z_n\}$$

$$P_{n+1} = \{x_{n+1}, y_{n+1}, z_{n+1}\}$$

ning arvutada nende kahe punktide vahelise kauguse:

$$d_{n+1,n} = \sqrt{(x_{n+1} - x_n)^2 + (y_{n+1} - y_n)^2 + (z_{n+1} - z_n)^2}$$

Olgu faasi  $F$  jooksul toimunud kaadrite koguarv  $m$ . Combined trajectory arvutatakse siis järgnevalt:

$$T_F = \sum_{n=0}^{m-1} d_{n+1,n}(m)$$

#### 4.2.6 Euclidean distance

**Euclidean distance** kirjeldab ühe korduse ja ühe selle korduse faasi mingi liigese faasi esimese punkti ning faasi viimase punkti vahelist kaugust. Euclidean distance tähistus on  $E$ .

Olgu meil teada faasi  $F$  mingi liigese faasi  $F$  esimene punkt koos oma koordinaatidega ning viimane punkt koos oma koordinaatidega. Olgu faasis  $F$  kokku kaadreid  $m$ :

$$P_0 = \{x_0, y_0, z_0\}$$

$$P_m = \{x_m, y_m, z_m\}$$

Euclidean distance arvutatakse siis järgmiselt:

$$E_F = d_{0,m} = \sqrt{(x_m - x_0)^2 + (y_m - y_0)^2 + (z_m - z_0)^2} (m)$$

#### 4.2.7 Time

**Time** kirjeldab ühe korduse ja ühe selle korduse faasi toimumise aega. Time'i tähistus on  $t$ . Olgu meil teada mingi faas  $F$ , milles on kokku  $m$  kaadrit ning mille jooksul salvestatud suvalise liigese esimese kaadri salvestusaeg on  $t_0$  ja viimase kaadri salvestusaeg on  $t_m$ , siis saame arvutada time'i  $t$ :

$$t_F = \frac{t_m - t_0}{1000} (s)$$

1000ga jagamine on vajalik selleks, et konverteerida aeg millisekunditest sekunditeks.

#### 4.2.8 Motion masside suhted

Lisaks motion mass'ele võimaldab rakendus arvutada ka nende parameetrite suhteid. Järjekordselt, käivad need suhted ühe korduse ühe selle korduse faasi  $F$  mingi liigese parameetrite kohta. Täpsemalt saab arvutada järgnevaid suhteid:

1. Combined trajectory ja Euclidean distance suhe **T/E**

$$F_{\frac{T}{E}} = \frac{T_F}{E_F}$$

2. Velocity mass ja Euclidean distance suhte **V/E**

$$F_{\frac{V}{E}} = \frac{V_F}{E_F}$$

3. Acceleration mass ja Euclidean distance suhe **A/E**

$$F_{\frac{A}{E}} = \frac{A_F}{E_F}$$

4. Jerk mass ja Euclidean distance suhe **J/E**

$$F_{\frac{J}{E}} = \frac{J_F}{E_F}$$

#### 4.2.9 Nurgad

Rakendus võimaldab arvutada ka erinevate liigete vahelisi nurkasid ning nende nurkade eksteermumeid. Nurgad arvutatakse ühe korduse terve ühe faasi kohta. Nurgad esitatakse kraadides.

Olgu meil mingi suvaline faas  $F$  ja vaadeldava nurga tüüp  $r$ . Lisaks, olgu meil iga selle faasi  $F$  kaadri jaoks arvutatud nurga  $r$  väärtuste komplekt  $N$ . Olgu selles faasis kokku  $m$  kaadrit ning viidaku  $n$  nurkade komplektis  $N$  hetkel vaadeldava nurga numbrile. Seega vastab  $n$  ka hetkel vaadeldavale kaadri, kus  $n \leq m$ , siis ekstreemumid, mida rakendus võimaldab iga arvutatud nurga kohta arvutada on järgmised:

1. **Start** – kirjeldab faasi esimese kaadri andmete põhjal arvutatud nurga väärtust.

$$F_{r_{start}} = N_0$$

2. **End** – kirjeldab faasi viimase kaadri andmete põhjal arvutatud nurga väärtust.

$$F_{r_{end}} = N_m$$

3. **Min** – kirjeldab faasi kõikide kaadrite andmete põhjal arvutatud nurkade minimaalset väärtust.

$$F_{r_{min}} = \text{Min}(N)$$

4. **Max** – kirjeldab faasi kõikide kaadrite andmete põhjal arvutatud nurkade maksimaalset väärtust.

$$F_{r_{\max}} = \text{Max}(N)$$

5. **Avg** – kirjeldab faasi kõikide kaadrite andmete põhjal arvatatud nurkade keskmist väärtust.

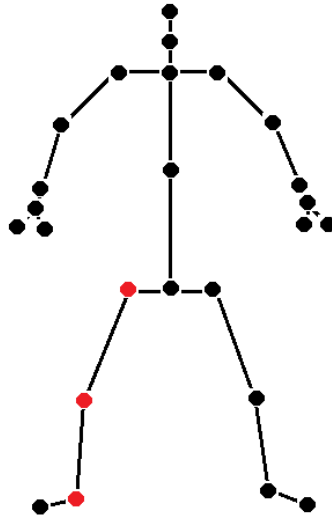
$$F_{r_{\text{avg}}} = \frac{\sum_{n=0}^m N_n}{m}$$

6. **Sum** – kirjeldab faasi kõikide nurkade andmete põhjal arvatatud iga kahe järgneva nurga vahelist absoluutväärtuste summat.

$$F_{r_{\text{sum}}} = \sum_{n=0}^{m-1} |N_{n+1} - N_n|$$

Nurki, mida rakendus võimaldab arvutada ühe korduse ja ühe selle korduse faasi kohta on 8. Järgnevalt on välja toodud nurga nimed ning joonised iga arvutatava nurga kohta, kus punased täpid tähistavad nurga arvutuseks kasutatud liigeseid:

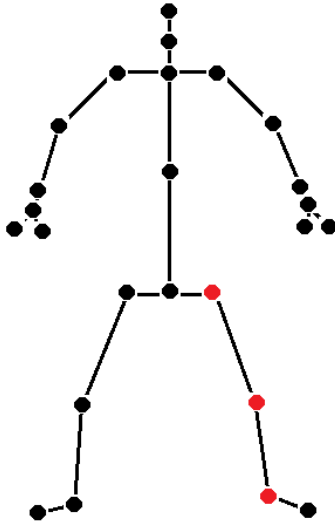
1. **Hip-Knee-Ankle (Right)**. Nurk puusa, põlve ning pahkluu vahel paremal pool.



Joonis 2. Nurk Hip-Knee-Ankle  
(Right)

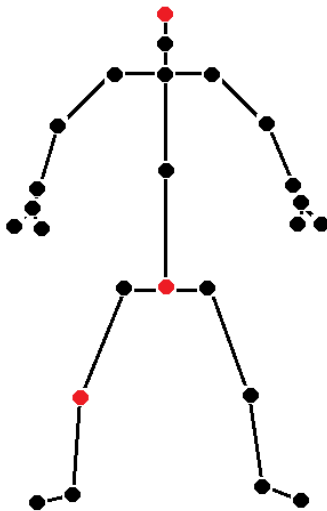


2. **Hip-Knee-Ankle (Left)**. Nurk puusa, põlve ning pahkluu vahel vasakul pool.



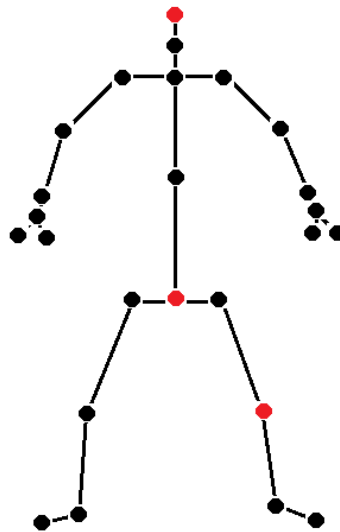
Joonis 3. Nurk Hip-Knee-Ankle (Left)

3. **Head-SpineBase-Knee (Right)**. Nurk pea, ristluu ning põlve vahel paremal pool.



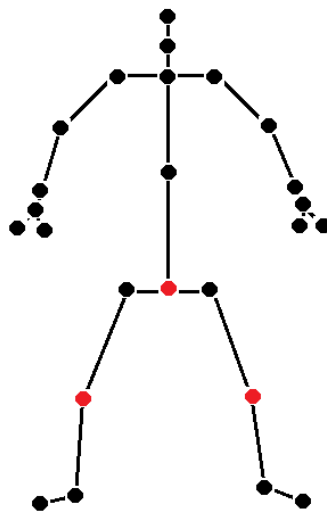
Joonis 4. Nurk Head-SpineBase-Knee (Right)

4. **Head-SpineBase-Knee (Left)**. Nurk pea, ristluu ning põlve vahel vasakul pool.



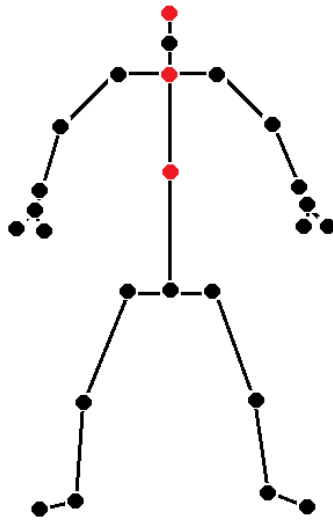
Joonis 5. Nurk Head-SpineBase-Knee (Left)

5. **Knee-SpineBase-Knee**. Nurk parema põlve, ristluu ning vasaku põlve vahel.



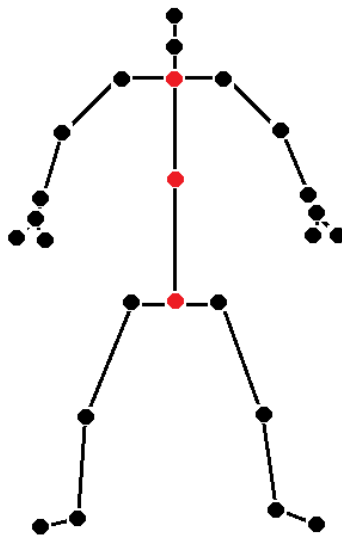
Joonis 6. Nurk Knee-SpineBase-Knee

6. **Head-SpineShoulder-SpineMid.** Nurk pea, rinnaluu ning selgroo keskpunkti vahel.



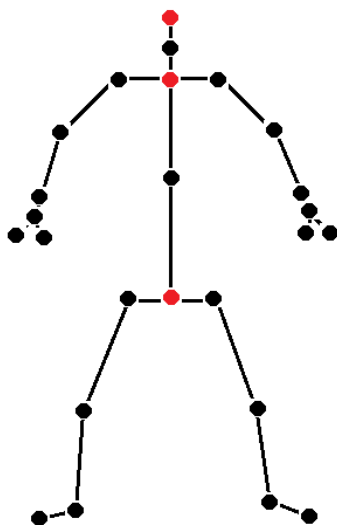
Joonis 7. Nurk Head-SpineShoulder-SpineMid

7. **SpineShoulder-SpineMid-SpineBase.** Nurk rinnaluu, selgroo keskpunkti ning ristluu vahel.



Joonis 8. Nurk SpineShoulder-SpineMid-SpineBase

8. **Head-SpineShoulder-SpineBase.** Nurk pea, rinnaluu ning ristluu vahel.



Joonis 9. Nurk Head-SpineShoulder-SpineBase

## **5 Praktiline osa**

### **5.1 Faasituvastus**

Võrreldes vana rakendusega, võimaldab käesolev rakendus konteksti salvestamise jooksul tuvastada kontekstist sõltuvaid faaside alguseid automaatselt. Igal harjutusel on erinev arv faase, mida rakendus suudab tuvastada. Faaside muutus kirjutatakse salvestatavasse faili, iga kaadri ja selle andmete rea lõppu erilise märkenumbri, kus igale numbrile vastab mingi kindel faas. Harjutused seisnevad formaliseeritud faaside hulgas, kus on teada mis faas eelneb ning järgneb käesolevale. Rakendus võimaldab harjutusi korrata ning korduste jooksul samuti faase tuvastada. Tuvastamine toimub mitte kõikvõimalike faaside järgi, vaid kontekstipõhiselt. Lisaks, suudab rakendus eristada harjutuse korduseid (tavaliselt viiakse ühte konteksti läbi 3 korda). Kuna harjutused „Up and sit“ ja „Go“ kõik faasid sisalduvad harjutuses „Up and go“, siis esmalt kirjeldataksegi faasi „Up and go“ ning selle põhjal järgnevalt ülejäänud kaks faasi. Järgnevates alampeatükkides on kirjeldatud igale harjutusele vastavad faasid koos nende tuvastusmeetoditega.

### **5.2 Kontekst Up and go**

Up and go (tõuse ja mine) kontekst sisaldab üheksat faasi (sulgudesse on lisatud vastavale faasile omane märkenumber, mis kirjutatakse faili): istumine (0), püsti tõusmine(1), seismine (7), sammumise alustamine (2), ümber pööramine (3), tagasi kõndimine (4), ümber pööramine (5), seismine (8), maha istumine (6), istumine (0). Nende üheksa faasi läbimine tähendab, et harjutus on ühe korra läbitud. Harjutust võib läbida ühe salvestuse käigus kui mitu kordi tahes. Faasituvastus meetodeid kutsutakse välja vastavalt hetkel toimuvale faasile iga Kinecti poolt saadetud kaadri jaoks. Järgnevalt kirjeldatakse iga faasi tuvastusmeetodeid.

### **5.2.1 Istumine konteksti alguses**

Harjutuse alguses olevat istumise faasi otseselt ei tuvastata, sest rakendus eeldab, et Up and go tüüpi konteksti alguses on inimene istumisfaasis. Küll aga salvestatakse istumise faasi ajal inimese ristлуу Y koordinaat. See on vajalik istumisele järgneva faasi korrektseks tuvastuseks. See tähendab, et tegelikult hakatakse kohe kontrollima, kas inimene on alustanud püsti tõusmist.

### **5.2.2 Püsti tõusmine**

Püsti tõusmise kontroll toimub terve istumise faasi jooksul. Esimese saadud kaadri jaoks arvutatakse ja salvestatakse nurgad inimese mõlema jala pahkluu, põlve ning puusa vahel. Lisaks salvestatakse minimaalne nurk, et programm arvaks, et toimub püsti tõusmine. See nurk on eelnevalt arvutatud nurk + 20 kraadi.

Järgnevalt hakatakse kontrollima igat kaadrit, mis toimub, kui hetkefaas on istumise olekus. Selleks arvutatakse iga kaadri jaoks uuesti keskmine nurk inimese mõlema jala pahkluu, põlve ning puusa vahel ning kontrollitakse, kas see nurk on suurem kui minimaalne püsti tõusmise nurk. Lisaks, kasutatakse eelnevalt salvestatud inimese ristлуу Y koordinaati. Selleks, et programm märgiks hetkefaasiks "püsti tõusmine", peab lisaks jala nurga kontrollile olema järgnev kontroll samuti tõene: arvutuste põhjal on teada, et iga +1 kraadi kohta (eelnevalt arvutatud jalgade keskmise nurga kasvul) liigub inimese ristлуу koordinaat y teljel positiivses suunas 0.35 cm. Programm kontrollib, kas hetkel saadud kaadrist arvutatud ristлуу Y koordinaat on suurem kui eelnevalt nimetatud ristлуу Y koordinaadi kasvu konstant, mis on korrutatud minimaalse püsti tõusmise nurgaga (20 kraadi) ning millele on liidetud eelmises faasis arvutatud esialgne ristлуу Y koordinaat.

Kui eelnevad kontrollid (Vt. Lisa 1) on tõesed, siis on väga tõenäoline, et inimene on alustanud püsti tõusmist ning seetõttu vahetatakse ka hetkefaas vastavale vaasile.

### **5.2.3 Seismine peale püsti tõusmist**

Seismise faasi kontrolli hakatakse tegema siis, kui inimene on püsti tõusmise faasis. Seismise faasi kontroll seisneb pidevalt eelnevas faasis nimetatud jalgade nurkade keskmiste arvutamisel ning nende võrdlemisel erinevate konstantidega. Nimelt, salvestatakse iga kaadri jaoks arvutatud jalgade nurk massiivi. Kui massiivi suurus on

vähemalt 5 elementi (seda väärtust võib vabalt muuta, kuid selle suurendamine pikendab faasi tuvastuse protsessi ajakulu, mis see-eest muudab kontrolli täpsemaks) ning kui praeguse kaadri jala nurk on suurem või võrdne kui minimaalne seismise jala nurk (160 kraadi), siis hakatakse arvutama ning võrdlema keskmiseid. Nimelt, arvutatakse eelnevalt nimetatud massiivis oleva viie viimase elemendi keskmine väärtus. Seda keskmist võrreldakse praeguse kaadri jaoks arvutatud jala nurgaga. Kui praeguse kaadri jaoks arvutatud jalgade nurk jääb keskmise väärtuse +/- erinevuskonstandi (1 kraad) vahemikku, siis see tähendab, et viimase viie kaadri jooksul ei ole toimunud jala nurgas suuremaid muutuseid, mis tähendab, et võib eeldada, et inimene on jõudnud stabiilsesse asendisse nagu seismine.

Nende kontrollide (Vt. Lisa 2) täitmisel muudetakse inimese hetkefaas faasi „seisab“. Lisaks, kui esimest korda muudetakse hetkefaas faasi „seisab“, siis sellel hetkel salvestatakse jällegi ka seismise hetkel olev inimese ristluu Y koordinaat ning inimese selgroo keskpunkti Z koordinaat, mida kasutatakse edaspidisteks kontrollideks.

#### **5.2.4 Sammumise alustamine**

Sammumise alustamise faasi hakatakse kontrollima siis, kui inimene on jõudnud faasi “seisab”. Esimese asjana (esimese salvestatud kaadri jaoks, kus patsient on “seisab” faasis) salvestatakse inimese parema ja vasaku pahkluu ning jalalaba Z koordinaadid. Järgnevalt hakatakse läbi viima esimese sammu kontrolli, kusjuures nimetatud kontroll viiakse läbi kahe juhtumi jaoks: kui inimene alustab esimest sammu parema jalaga või vasaku jalaga. Selleks arvutatakse mõlema jala jaoks nende “keskmine” Z koordinaat, mis seisneb mõlema jala (jalalaba ning pahkluu) Z koordinaatide keskmises.

Sellele järgnevalt vaadatakse eelnevalt nimetatud kahe juhtumi jaoks järgnevaid kontrole (kirjeldus on toodud juhtumi kohta, kus inimene alustab sammumist parema jalaga. Vasaku jalaga sammumise alustamise kontroll on identne, kuid vastupidine): esmalt kontrollitakse kas arvutatud vasaku jala keskmine väärtus jääb algse jala Z koordinaadi +/- seismise konstandi (3 cm) vahemikku.

Järgnevalt kontrollitakse sammumise alustamiseks kasutatavat jalga (paremat). Kontrollitakse, kas parema jala arvutatud keskmine Z koordinaadi väärtus on väiksem (liikudes Kinect'i suunas z koordinaat väheneb [10]) kui seismise hetkel salvestatud parema jala z koordinaadi väärtus + minimaalne esimese sammu distants (8 cm).

Need kontrollid (Vt. Lisa 3) kindlustavad, et üks jalg on jäänud seisvasse asendisse, kuid teine jalg on alustanud liikumist Kinecti suunas. Sellist tüüpi kontroll kindlustab, et inimese tagurpidi kõndimisel ei arvataks, et toimub edasi kõndimine. Nüüd muudetakse hetkefaas faasi “kõnnib”.

### **5.2.5 Ümber pööramine kaamera juures**

See faas kirjeldab inimese ümber pööramist Kinect kaamera juures pärast selle poole kõndimist. Tegelikult otseselt ümber pööramist ei kontrollitagi, sest Kinect'i jaoks on probleemne tuvastada korrektselt pööravat või seljaga kaamera poole seisvat isikut. Selle asemel on aga teada, et kõndimist sisaldavates kontekstides viiakse kõndimine läbi ~1.5m kauguseni Kinectist. See on ka kaugus, kus Kinect veel suudab inimest korrektselt tuvastada. Seetõttu kontrollitakse ümber pööramise faasi jaoks inimese selgroo keskpunkti Z koordinaadi kaugust Kinect'i kaamerast. Täpsemalt, selleks, et hetkefaas muudetaks faasi “ümber pööramine”, peab inimese selgroo keskpunkti Z koordinaat olema vahemikus  $1.5\text{m} + 0.1\text{m}$  ja  $1.5\text{m} - 1\text{m}$ . Ehk  $1.4\text{m}$  ja  $0.5\text{m}$  vahemikus kaugusel  $1.5\text{m}$  Kinect kaamerast (Vt. Lisa 4).

Lisaks, kui esimest korda tuvastatakse, et eelnevalt nimetatud kontroll on tõene, siis salvestatakse inimese selgroo keskpunkti Z koordinaat edaspidisteks arvutusteks.

### **5.2.6 Tagasi kõndimine**

Tagasi kõndimise faasi kontroll algab siis, kui on tuvastatud, et inimene on jõudnud eelmisesse, ümber pööramise, faasi. Jällegi, kuna Kinectil on probleeme seljaga kaamera poole seisva inimese korrektses tuvastamises, siis kontrollitakse selles faasis selgroo keskpunkti Z koordinaati. Eelmises faasis salvestati ümber pööramise hetkel olev selgroo keskpunkti Z koordinaat. Nüüd, selleks, et hetkefaas muutuks faasi “tagasi kõndimine”, kontrollitakse iga kaadri korral inimese hetkelist selgroo keskpunkti Z koordinaati ja võrreldakse seda eelnevalt salvestatud selgroo keskpunkti Z koordinaadiga.

Nimelt, kui hetkel salvestatud selgroo keskpunkti Z koordinaat on suurem kui ümber pööramise hetkel salvestatud selgroo keskpunkti Z koordinaat  $+ 0.1\text{m}$  ( $0.1\text{m}$  on distant, mida kasutati ümber pööramise kontrollis vahemiku kontrolliks ja Kinectist eemale



kõndides see koordinaat kasvab) (Vt. Lisa 5), siis muudetakse hetkefaas faasi “kõnnib tagasi”.

### **5.2.7 Ümber pööramine tooli ees**

Ka selles faasis on inimene seljaga kaamera poole ning mistõttu peab kontrollima ainult üldiseid parameetreid. Seetõttu toimub ka tooli ees oleva ümber pööramise kontroll inimese selgroo keskpunkti Z koordinaadi põhjal. Selleks, et ümber pööramise kontrolli tehtaks, peab eelnevalt olema inimene jõudnud eelmisesse, tagasi kõndimise, faasi. Eelnevalt püsti tõusmisele järgnevas seimise faasis salvestati inimese tooli ees seismise selgroo keskpunkti Z koordinaat. Selle faasi kontroll seisneb iga kaadri jaoks leitud selgroo keskpunkti Z koordinaadi võrdlemine eelnevalt salvestatud Z koordinaadi vastu.

Täpsemalt, kui praegusel hetkel arvatud selgroo keskpunkti Z koordinaat jääb vahemikku eelnevalt salvestatud Z koordinaat  $- 0.1\text{m}$  ja  $+ 0.35\text{m}$ , siis muudetakse hetkefaas faasi “pöörab ümber tooli ees” (Vt. Lisa 6).

### **5.2.8 Seismine tooli ees peale ümber pööramist**

Seismise kontrolli viiakse läbi algoritmiliselt sarnaselt seismise kontrollile, mis toimus peale püsti tõusmist, kuid hetkese faasi tuvastamiseks ei ole mõttekas kontrollida mitte põlvede vahelisi nurki, vaid keskmist vasaku ja parema jala pahkluu vahelist X koordinaatide vahet. Nimelt, seisneb algoritmi mõte selles, et kui inimese mõlema jala pahkluu X koordinaatide vahe ei ole mingi aja jooksul drastiliselt muutunud, siis võib eeldada, et inimene on seisvas (stabiilses) asendis ning eelmine, pööramise faas on lõppenud. Selleks, et vastava faasi tuvastus algaks, peab inimene olema faasis “pöörab tooli ees”. Täpsemalt, hakatakse arvutama iga kaadri jaoks parema ning vasaku jala pahkluu X koordinaate. Nende koordinaatide põhjal arvutatakse nende vahe ning lisatakse massiivi. Iga kaadri jooksul, kui massiivi suurus on vähemalt 5 elementi (jällegi, valides selleks kontrolliks suurema arvu, saab suurendada algoritmi täpsust, kuid mille tulemusena suureneb esimese kontrolli jaoks kuluv aeg), arvutatakse viimase viie massiivielemendi keskmine ning võrreldakse seda praeguse kaadri jaoks arvatud jalgade pahkluu X koordinaatide vahega.

Kui praeguse kaadri jaoks arvatud väärtus jääb vahemikku keskmine  $\pm 0.0001\text{m}$ , siis muudetakse hetkefaas faasi “seisab” (Vt. Lisa 7).

### **5.2.9 Maha istumine**

Maha istumise faasituvastuse kontrolle hakatakse läbi viima, kui inimene on faasis kas “seisab” või “pöörab tooli ees”. Seda seetõttu, kuna mõned inimesed hakkavad istuma juba pööramise hetkel, samas kui teised jäävad alguses mõneks hetkeks seisma ning alles siis alustavad istumist. Sarnaselt püsti tõusmisega, kontrollitakse siin faasis inimese mõlema jala pahkluu, põlvede ning puusa vahelist nurka. Lisaks, on salvestatud minimaalne nurk, et vastav kontroll hakkaks toimuma – minimaalne seismise nurk (160 kraadi) + 5 kraadi ehk 165 kraadi. Iga kaadri jaoks arvutatakse eelnevalt nimetatud nurk ning võrreldakse seda minimaalse istumise nurgaga. Kui hetkel arvutatud nurk on väiksem kui minimaalne istumise nurk, siis kontrollitakse lisaks ka inimese selgroo keskpunkti Y koordinaati sarnaselt püsti tõusmise faasile. Nimelt, on jällegi teada kui palju väheneb inimese selgroo keskpunkti Y koordinaat iga ühe eelnevalt nimetatud nurga kraadi võrra. Teada on, et minimaalne istumise nurk on 160 kraadi, siis saame arvutada väärtuse, millest peab praegune inimese selgroo keskpunkti Y koordinaat olema väiksem. Lisaks, teame eelnevast (seismise faasist) faasist inimese selgroo keskpunkti Y koordinaadi väärtust seismise hetkel. Nüüd kontrollitakse, kas inimese hetkene selgroo keskpunkti Y koordinaadi väärtus on väiksem või võrdne kui inimese seismise faasi hetkel arvutatud selgroo keskpunkti Y koordinaadi väärtus, millest on lahutatud minimaalse istumise nurga põhjal arvutatud eelnevalt nimetatud parameter.

Eelmiste kontrollide läbimisel muudetakse hetkefaas faasi “istub maha” (Vt. Lisa 8).

### **5.2.10 Istumine konteksti lõpus**

Seda faasi hakatakse kontrollima siis, kui inimene on jõudnud eelmisesse, maha istumise, faasi. Kasutatakse jällegi massiivi viimase  $n$  elemendi arvutamise keskmise algoritmi. Iga kaadri jaoks arvutatakse ja lisatakse massiivi nurk inimese mõlema jala pahkluu, põlve ning puusa vahel. Kontrollitakse, et hetkel arvutatud nurk on väiksem või võrdne minimaalsest istumise nurgast (95 kraadi). Jällegi, kui massiivi suurus on vähemalt 5 elementi, siis võetakse selle massiivi viimased 5 elementi ning arvutatakse nende keskmine. Kui hetkel arvutatud eelnevalt nimetatud nurk jääb vahemikku arvutatud keskmine  $\pm 5$  kraadi, siis on tõenäoline, et viimase 5 kaadri jooksul ei ole toimunud suuremaid muutuseid inimese pahkluu, põlve ja puusa vahelises nurgas. Seetõttu muudetakse hetkefaas faasi “istub” (Vt. Lisa 9).

## **5.3 Kontekst Up and sit**

Up and sit (tõuse ja istu) kontekst sisaldab viite faasi (sulgudesse on lisatud vastavale faasile omane märkenumber, mis kirjutatakse faili): istumine (0), püsti tõusmine(1), seismine (7), maha istumine (2), istumine konteksti lõpus (3). Nimetatud viie faasi läbimine tähendab, et kontekst on ühe korra läbitud. Konteksti võib läbida ühe salvestuse käigus kui mitu kordi tahes. Faasituvastus meetodeid kutsutakse välja vastavalt hetkel toimuvale faasile iga Kinecti poolt saadetud kaadri jaoks. Järgnevalt kirjeldatakse iga selle konteksti faasi tuvastusmeetodeid.

### **5.3.1 Istumine**

Sarnaselt kontekstile Up and go, algab kontekst Up and sit inimese istuvast asendist. Eeldatakse, et faasi alguses on inimene juba istuvas olekus. Selles faasis midagi otseselt ei tuvastata, vaid hakatakse kohe kontrollima püsti tõusmiseks vajalike kontrollide täitmise meetodeid.

### **5.3.2 Püsti tõusmine**

Püsti tõusmise kontroll algab tegelikult juba kohe peale harjutuse alustamist st. istumise faasis. Esmalt salvestatakse sarnaselt faasile Up and go edaspidisteks arvutusteks vajalikud parameetrid – inimese esialgne pahkluu põlve ning puusa nurk, minimaalne püsti tõusmise nurk (eelnevalt arvatud nurk + 20 kraadi) ning algne inimese selgroo keskpunkti Y koordinaadi väärtus. Järgnevalt toimub konteksti Up and go püsti tõusmise faasile algoritmiliselt identne püsti tõusmise kontroll (Vt. 5.2.2 Püsti tõusmine). Kontrolli läbimisel muudetakse hetkefaas faasist “istub” faasi “tõuseb püsti”.

### **5.3.3 Seismine**

Seismise kontrolli viiakse läbi siis, kui inimese hetkefaas on “tõuseb püsti”. Seismise kontrolli meetodi algoritm on jällegi identne kontekstiga Up and go (Vt. 5.2.3 Seismine peale püsti tõusmist) – võrreldakse minimaalse seismise nurga ja hetkelise jalgade nurga vahelist nurka ning arvutatakse ja lisatakse keskmiseid massiivi, et kontrollida kas viimase n kaadri jooksul on toimunud suuremaid muutuseid. Kontrolli läbimise tulemusena muudetakse inimese hetkefaas faasi “seisab”.

### **5.3.4 Maha istumine**

Erinevalt kontekstile Up and go, hatakase kontekstis Up and sit peale seismist kohe kontrollima maha istumise faasi. See ka eristab konteksti Up and sit teistest kontekstidest. Jällegi, on maha istumise kontrolli algoritm identne konteksti Up and go maha istumise faasi kontrollile (Vt. 5.2.9 Maha istumine). Võrreldakse inimese pahkluu, põlve ja puusa nurka minimaalse istumise nurga vastu ning kontrollitakse inimese selgroo keskpunkti Y koordinaadi muutumist vastavalt eelnevalt nimetatud jalgade nurkade muutumise põhjal. Kontrollide tulemuste läbimisel muudetakse hetkefaas faasist “seisab” faasi “istub maha”.

### **5.3.5 Istumine konteksti lõpus**

Konteksti lõpus oleva istumise faasi kontrolli viiakse läbi siis, kui patsient on jõudnud faasi "istub maha". Faasikontroll on identne kontekstis Up and go läbi viidud viimase faasi kontrollile (Vt. 5.2.10 Istumine konteksti lõpus). Kontrollide läbimise tulemusena on inimene faasis “istub konteksti lõpus” ning seetõttu on ka nimetatud kontekst läbitud.

## **5.4 Kontekst Go**

Erinevalt kontekstidest Up and go ning Up and sit algab kontekst Go faasist, kust inimene on seisvas asendis. Kontekst Go sisaldab kuute faasi (sulgudesse on lisatud vastavale faasile omane märkenumber, mis kirjutatakse faili): seismine (0), sammumise alustamine (1), ümber pööramine kaamera juures (2), tagasi kõndimine (3), ümber pööramine alguses (4) ning seismine alguses (5). Nimetatud kuue faasi läbimine tähendab, et kontekst on ühe korra läbitud. Konteksti võib läbida ühe salvestuse käigus kui mitu kordi tahes. Faasituvastus meetodeid kutsutakse välja vastavalt hetkel toimuvale faasile iga Kinecti poolt saadetud kaadri jaoks. Järgnevalt kirjeldatakse iga selle konteksti faasi tuvastusmeetodeid.

### **5.4.1 Seismine**

Sarnaselt kontekstidele Up and go ning Up and sit, eeldab kontekst Go, et inimene on konteksti alguses juba esimeses faasis – seismise faasis. See tähendab, et juba seismise faasis alustatakse esimese sammu kontrolle.

#### **5.4.2 Sammumise alustamine**

Sammumise alustamise kontroll algab juba tegelikult konteksti alustamise hetkest. Esmalt, esimese kaadri jooksul, salvestatakse inimene selgroo keskpunkti Z koordinaat ning mõlema jala pahkluu ja jalalaba Z koordinaadid. Need on vajalikud edaspidisteks arvutusteks. Jalgade Z koordinaadid on vajalikud juba praeguse faasi tuvastamiseks. Järgnevalt hakatakse kontrollima esimest sammu ning seda mõlema juhtumi vastu – kui inimene alustab sammumist parema jalaga või vasaku jalaga. Sammumise alustamise faasi kontroll toimub jällegi identselt Up and go faasis toimuva sammumise alustamise faasi kontrollile (Vt. 5.2.4 Sammumise alustamine). Tulemuseks on inimene faasis “sammumise alustamine”.

#### **5.4.3 Ümber pööramine kaamera juures**

Vastavat faasi kontrolli hakatakse läbi viima siis, kui inimene on jõudnud edasi kõndimise faasi. Sarnaselt kontekstile Up and go, on teada, et ümber pööramine peaks toimuma siis, kui inimene on jõudnud kaamerast 1.5m kaugusele. Jällegi, kuna Kinectil on probleeme inimese korrektse tuvastamisega, kui inimene ei seisa näoga kaamera poole, siis on sarnaselt kontekstile Up and go valitud vastava faasi tuvastuseks inimese selgroo keskpunkti Z koordinaadi võrdlemine teatud väärtustega. Faasi tuvastuse algoritm on jällegi identne kontekstiga Up and go (Vt. 5.2.5 Ümber pööramine kaamera juures). Sarnaselt, salvestatakse pööramise hetkel inimese selgroo keskpunkti Z koordinaat järgmise faasi tuvastamiseks. Tulemusena on inimene jõudnud faasi “pöörab kaamera juures”.

#### **5.4.4 Tagasi kõndimine**

Tagasi kõndimise kontrolli hakatakse läbi viima siis, kui inimene on faasis “ümber pööramine kaamera juures”. Jällegi, sarnaselt kontekstile Up and go, kontrollitakse inimese selgroo keskpunkti Z koordinaadi asukohta ning võrreldakse seda eelmises faasis salvestatud sama koordinaadiga (Vt. 5.2.6 Tagasi kõndimine). Tulemusena on inimene faasis “kõnnib tagasi”.

#### **5.4.5 Ümber pööramine alguses**

Vastavat faasi hakatakse kontrollima, kui inimene on faasis “kõnnib tagasi”. Vastava faasi põhiline võrdlusobjekt on inimese selgroo keskpunkti Z koordinaat, mida võrreldakse sammumise alustamise faasis salvestatud sama koordinaadiga. Kontroll on

jällegi sarnane konteksti Up and go ümber pööramise kontrolliga (Vt. 5.2.7 Ümber pööramine tooli ees). Nimetatud kontrolli läbimise tulemusena on inimene faasis “ümber pööramine alguses”.

#### **5.4.6 Seismine alguses**

Seismise faasi kontrolli viiakse läbi siis, kui inimene on faasis “ümber pööramine alguses”. Selleks kontrollitakse sarnaselt kontekstile Up and go inimese pahklude X koordinaatide vahet, lisades neid massiivi ja arvutades keskmiseid (Vt. 5.2.8 Seismine tooli ees peale ümber pööramist). Kontrolli mõte seisneb selles, et kui massiivist pidevalt viimase n elemendi keskmist arvutada ja kontrollida, kas hetkel arvutatud kaadri pahklude X koordinaatide vahe väärtus jääb selle keskmise +- mingi konstandi vahemikku, siis tõenäoliselt on lõppenud inimese jalgade vaheline aktiivne liikumine (ümber pööramine) ning inimene on jõudnud faasi “seisab”. Lisaks on kontekst korra läbitud ning nüüd võib soovi korral alustada konteksti läbimist uuesti algusest.

### **5.5 Kontekstide salvestamine**

Selleks, et rakendus saaks kontekste analüüsida või soovi korral järele vaadata, peab rakendus võimaldama inimese liigeste koordinaate faili salvestada, arvestades valitud kontekstitüüpi ning faasituvastusest saadud andmeid inimese hetkefaasi kohta.

Selleks, kirjutab rakendus reaajas konteksti läbi viimise jooksul andmeid pidevalt csv faili ning märkides pidevalt hetkefaasi igale faasile vastava märkenumbriga (kirjeldatud faasituvastuse peatükkides). Rakendus salvestab korraga kõiki 25e liigest, mida Kinect for Xbox One suudab tuvastada.

Esmalt, kirjutab rakendus logifailile päise, mis seisneb valitud harjutusetüübi nimes, inimese nimes, nädala numbris ning harjutuse korra numbris. Sellele järgnevalt hakkab rakendus iga liigese kohta kirjutama selle liigese X, Y ja Z koordinaate, millele eelnevad salvestatud kaadri number ning ajamärge. Ajamärge on number millisekundites alates salvestamise alguse hetkest. See tähendab, et iga salvestatud kaadri kohta kirjutab rakendus faili  $75(x, y \text{ ja } z \text{ koordinaadid iga } 25\text{e liigese kohta}) + 2$  (kaadri number ning ajamärge) + 1 (hetkefaasi märkenumber) komaga eraldatud väärtust (Vt. Lisa 10).

Tuleb märkida, et nagu näites on toodud, siis esimeste kaadrite jooksul ei pea kohe olema viimaseks väärtuseks hetkefaasi märkenumber, sest faasituvastus algab alles siis, kui kasutaja vajutab nupule. Niimoodi on võimalik salvestada faile ka ilma faasituvastuseta, kui selleks peaks tekkima soov.

## **5.6 Analüüs ja analüüsitulemused**

Eelmises peatükis kirjeldatud kontekstide logifailide põhjal viiakse läbi inimese liikumise analüüsi. Analüüsi kasutajaliides võimaldab kasutajal valida mitmeid parameetreid, mille põhjal analüüs läbi viiakse. Näited analüüsi kasutajaliidesest on toodud kasutajaliidese peatükis. Nimetatud kasutajaliides võimaldab kasutajal valida: liigeseid, mille peal analüüsi läbi viiakse, analüüsi tüüpe ning lisaks võimaldab analüüsi kasutajaliides valida milliste korduste ja milliste faaside peal eelnevalt valitud analüüse läbi viia. Valitud analüüsi viiakse läbi valitud liigestel ning valitud kontekstidel ja valitud nende kontekstide faasidel.

Selleks, et rakendus saaks läbi viia nimetatud analüüse, töötleb esmalt rakendus eelmises peatükis kirjeldatud kontekstide logifailid objektideks, mis on tulemusena analüüsiks sobivalt struktureeritud.

Analüüsi tulemused esitatakse sarnaselt kontekstide logifailile csv failina. Analüüsi tulemusi on võimalik esitada kahel kujul – tabeli kujul olev viis, mis on mugavam edaspidiseks töötamiseks või tavalisel kujul, mis on mugavam kohe lugemiseks. Analüüsitulemuste kuju saab muuta sätetes.

### **5.6.1 Analüüsitulemused tabeli kujul**

Tabeli kujul analüüsi tulemust esitamine on mugav mitme kordusega kontekstide kontrollimiseks ja võrdlemiseks. See tähendab seda, et sarnaste liigeste ning faaside tulemused on kokku grupeeritud mitme korduse peale. Tabeli kujul olev tulemuste fail sisaldab faili päist, kus on kirjas inimese nimi, konteksti tüüp, nädala number, testi number, konteksti salvestamise kuupäev ning kellaaeg, valitud korduste numbrid ning valitud faaside nimed. Sellele järgnevad tabeli kujul esitatavad ning eelnevalt kirjeldatud grupeeritud kujul olevad analüüsiandmed. Kusjuures, iga tabeli päisesse on lisatud mida iga tabeli veerg tähistab. Tabelite read vastavad kordustele. Näide analüüsitulemustest tabeli kujul (Vt. Lisa 11) on nii motion mass analüüside kohta kui

ka nurkade analüüsi kohta, kus vahepealsed osad on lõigatud (mahtuvuse tõttu) ning asendatud punktidega.

Toodud tabelite sisu võib muutuda vastavalt kasutaja analüüside valikule. Values nimeline veerg nurkade tabelis sisaldab kõiki faasi jooksul arvatud nurki, näitese mahtumise tõttu on see lõigatud nii, et see sisaldab ainult ühte väärtust. Muidugi, on csv failide loetavus ja analüüsimine optimaalsem, võrreldes siinse tavatekstiga kasutades selleks näiteks programmi Microsoft Excel.

### **5.6.2 Analüüsitulemused tavalisel kujul**

Tavalisel kujul esitatavad analüüsitulemused ei ole grupeeritud korduste kaupa, vaid sellel kujul olevad tulemused esitatakse iga korduse, iga faasi ja iga liigese kohta eraldi. Lisaks, esitatakse eraldi ka nurkade analüüsitulemused, mida samuti ei grupeerita. Tavalisel kujul olevad analüüsitulemused esitatakse ridade kaupa, kus iga rea esimene element kirjeldab analüüsitüüpi ning sellele järgnev(ad) analüüsitulemusi. Ka tavalisel kujul olevad analüüsitulemused sisaldavad esiteks päist, mis kirjeldab sama informatsiooni salvestuse kohta, nagu kirjeldas tabeli kujul olev tulemus. Iga tulemuse päises on samuti kirjeldatud, millise korduse (esitatud numbriga), millise faasi ja millise liigese kohta vastavad tulemused käivad (Vt. Lisa 12).

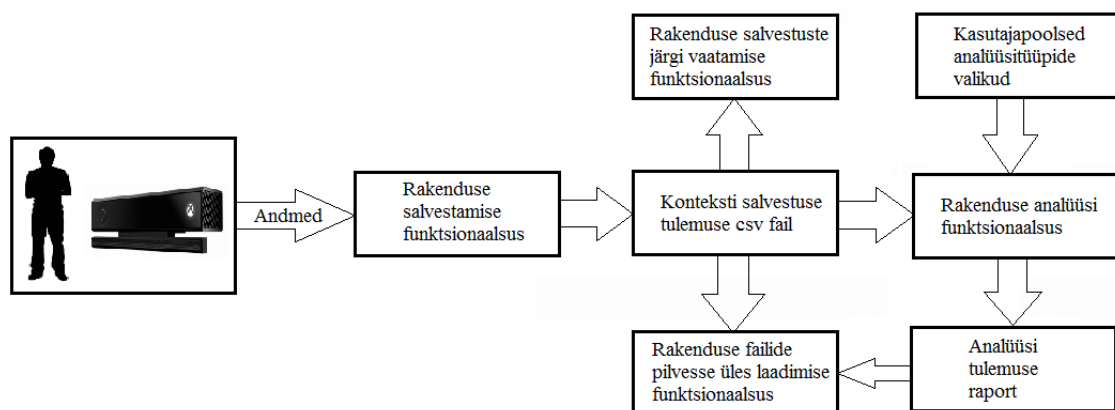
## **5.7 Analüüsitulemuste salvestamine**

Arvatud analüüsitulemuste salvestamiseks ja hoiustamiseks võimaldab rakendus laadida vastavaid faile üles Amazoni S3 Bucket'isse [15]. Analüüsitulemuste failide nimed salvestavad suvalist 9 sümboli pikkust stringi, seda seetõttu, et vältida mingi põhjusel tekkivaid duplikaate, kui samal ajahetkel peaksid mitu inimest sama konteksti salvestama (Vt. Lisa 13).

Failide üles laadmine toimub kasutaja soovil, kasutades selleks kasutajaliideses vaadet "Upload", mida on kirjeldatud kasutajaliideste peatükis.



## 5.8 Andmetöötlusprotsess



Joonis 10. Andmetöötlusprotsess

Salvestamise hetke algusest hakatakse kohe kontekstilogi csv faili kirjutama keha iga 25e liigese koordinaate iga saabunud kaadri kohta. Paralleelselt hakatakse tuvastama (harjutuse alustamise nupu vajutuse järgselt) inimese hetkefaase ning ka nende faaside märkenumbreid kirjutatakse csv faili iga saabunud kaadri koordinaatide lõppu. Faili kirjutamine toimub kuni kasutaja vajutab stop nuppu. Selle tulemusena on valminud kogu salvestust kirjeldav csv fail, mida saab nüüd kas pilvesse üles laadida, järgi vaadata või edasi analüüsida. Valides edasiseks toiminguks analüüsimise, tuleb kasutajal avada kasutajaliidesest “Analyze” aken. Järgnevalt toimub analüüsitüüpide valik, mis on täpsemalt kirjeldatud kasutajaliidese peatükis. Valiku lõpus vajutab kasutaja nuppu “Generate report”, mille tulemusena genereeritakse vastavalt kasutaja valikutele konteksti analüüsiraport. Jällegi, on kasutajal võimalus üles laadida genereeritud tulemusi.

## 6 Kasutajaliides

Kasutajaliidese tegemiseks kasutati Visual Studio arenduskeskkonnaga kaasa tulevat WPF alamsüsteemi, mis võimaldab mugavalt ja lihtsalt ehitada modernseid Windowsi platvormile omase väljanägemisega rakendusi.

Kasutajaliidese kasutamine põhineb erinevate otstarbetega akende vahel liikumise põhimõttel. Aknad sisaldavad tavalisi Windowsi platvormile ehitatud rakendustele omaseid elemente nagu: nupud, tekstiväljad, checkboxid jne. Lisaks, on enamus akende päises menüüriba.

### 6.1 Peamenüü

Rakenduse avamisel avatakse rakenduse peamenüü. Peamenüü kaudu toimub põhiline rakenduse sisene kasutajapoolne navigatsioon. Selleks, on peamenüüs iga tegevuse jaoks menüünupp, mille valjutamise tulemusena avaneb vastava tegevuse jaoks omane aken.

Menüünuppudeks on: *new* (uus konteksti salvestus), *open* (võimaldab järgi vaadata varem salvestatud kontekstifaili), *analyze* (salvestatud konteksti analüüsimise tegevus), *upload* (failide Amazon'i S3 Bucketisse laadimise vaade), *logs* (avab kasutaja poolt sätetest valitud logifailide salvestamise asukoha kasutaja arvutis), *settings* (rakenduse sätted) ja *exit* (sulgeb rakenduse) (Vt. Lisa 14).

### 6.2 Uue konteksti vaade (new)

Valiku *new* valimisel saab rakenduse kasutaja alustada uue konteksti salvestamist. Vastavale menüünupule vajutamise tulemusena avaneb esimese asjana väike lisaaken, mis võimaldab salvestada testitava inimese andmed ning valida kontekstitüüpi (Vt. Lisa 15).

Sellele järgnevalt avatakse rakenduse üks põhilisi vaateid, mis hakkab pidevalt välja joonistama Kinect kaamera ees seisva inimese eest-, pealt- ning kõrvaltvaadet. Eelnev vaade sisaldab nuppu „Start recording“, mille vajutamise tulemusena hakatakse faili salvestama inimese koordinaate. Lisaks, sisaldab vastav vaade ka nuppu „Start exercise“, mille tulemusena algab faasituvastus vastavalt valitud kontekstitüübile. „Start exercise“ nupp muutub aktiivseks pärast „Start recording“ nupu vajutamist. Sarnaselt nuppude vajutamisele on võimalik harjutusi alustada ja lõpetada kasutades selleks juhtmevabasid esitlusteks kasutatavaid pulte (testitud puldiga TRACER Showman 400). Faaside tuvastamise jooksul on nupu „Start exercise“ kõrval tekst, mis kirjeldab inimese hetkefaasi. Lisaks, kõlab iga harjutuse alguses helifail, mis seisneb inimese „Mine“ ütlemises. Selle helifaili keelt on võimalik muuta sätetes (Vt. Lisa 16).

### **6.3 Salvestuste järgi vaatamise vaade (open)**

Valides peamenüüst *open*, saab kasutaja eelnevalt salvestatud kontekste järgi vaadata. Nupule *open* klikkides avaneb kasutaja arvutis *file-explorer* tüüpi vaade, mis võimaldab kasutajal valida vaadatavat .csv faili. Faili valimise järgselt avaneb konteksti salvestamise vaatele sarnane vaade, mis sisaldab nuppe „play/replay“ ja „close“. Nupule „play/replay“ vajutamisel hakatakse valitud salvestust kasutajale välja joonistama (Vt. Lisa 17).

### **6.4 Kontekstide analüüsivaade (analyze)**

Valides peamenüüst valiku *analyze* avaneb kasutajale aken, mis võimaldab varem salvestatud .csv faili põhjal analüüsida inimese liikumist. Avanenud aknas palutakse kasutajal esmalt valida fail, mille põhjal analüüsi läbi viima hakatakse (Vt. Lisa 18).

Sellele järgnevalt saab kasutaja valida liigeseid, mille peal analüüsi läbi viia. Seda tehakse inimese skeleti peal olevate checkbox’ide märgistamisega (Vt. Lisa 19).

Olles valinud soovitud liigesed, avaneb vaade, mis võimaldab kasutajal valida soovitud analüüsitüüpe. Analüüsitüüpide hulka kuuluvad motion-mass analüüsid, motion-mass’ide suhted, nurgad ning nurkade ekstreemumid (Vt. Lisa 20).

Järgnevalt tuleb valida, milliste korduste ja milliste korduste faaside peal analüüsi läbi viia. Selleks võimaldab rakendus kasutajal valida nelja erineva juhu vahel: analüüsida kõiki korduseid ning kõiki selle korduse faase või analüüsida kindlaid korduseid, kuid

kõiki nende korduste faase või analüüsida kõiki korduseid, kuid kindlaid nende korduste faase või analüüsida kindlaid korduseid ja kindlaid faase (Vt. Lisa 21).

Järgnevalt esitatakse kokkuvõtte kasutaja valikutest ning eelseisvast analüüsiraportist. Vajutades nupule „Generate report“ luuakse kasutaja poolt sätetest valitud asukohta kasutaja valikute põhjal tehtud analüüsiraport (Vt. Lisa 22). Analüüsiraport luuakse kas tabeli kujul või tavalisel kujul (Vt. 5.6.1 Analüüsitulemused tabeli kujul ja 5.6.2 Analüüsitulemused tavalisel kujul). Seda on võimalik valida sätetest.

## **6.5 Üles laadimise vaade (upload)**

Valides peamenüüst valiku *upload* avaneb kasutajale aken, mis võimaldab varem salvestatud .csv faili Amazoni S3 Bucketisse üles laadida. Selleks, palutakse kasutajal esimese asjana valida soovitud fail. Faili valimise tulemusena kuvatakse valitud faili andmeid. Nüüd saab kasutaja vajutada nupule „upload“, mille tulemusena alustatakse üles laadimist. Kui eelnev tegevus oli edukas, muutub vaate alumises osas asuv progress-bar roheliseks ning kuvatakse teadet eduka üles laadimise kohta. Vastasel juhul kuvatakse veateadet (Vt. Lisa 23).

## **6.6 Sätete vaade (settings)**

Validest peamenüüst valiku *settings* avaneb kasutajale aken, mis võimaldab muuta rakendusega seotud sätteid. Sätete hulka kuuluvad: failide salvestamise asukoht kasutaja arvutis, valik automaatselt avada genereeritud analüüsifaile, kontekstide salvestamise ajal "Start Excercise" nupule vajutades kõlava „Mine!“ heli keelt (eesti, inglise, saksa ja itaalia) ning analüüsiraporti kuju (tabeli kujul või tavalisel kujul) (Vt. Lisa 25).

## 7 Kokkuvõte

Bakalaureusetöö eesmärgiks oli välja töötada rakendus, mis võimaldab salvestada, registreerida, analüüsida ning säilitada inimliigutustel põhinevaid kontekstist sõltuvaid harjutusi. Arendatud rakendus implementeerib kolme reaalselt inimliigutuste modelleerimiseks ja analüüsiks kasutatavat harjutust.

Loodud rakendus tõestab, et kasutades ühte üsnagi odavat Kinect kaamerat on võimalik salvestada, registreerida ja analüüsida inimese motoorikat üsnagi täpselt. Arendatud tarkvara töötab edukalt Windows'i operatsioonisüsteemidega arvutites. Rakendus kirjutati keeles C# kasutades selleks Visual Studio arenduskeskkonda. Mugava kasutajaliidese loomiseks kasutati WPF alamsüsteemi.

Tuleb märkida, et analüüsitulemused, mis on arvatud faaside põhjal, kus inimene ei seisa näoga kaamera poole võivad olla ebatäpsed. See juhtub Kinecti suutmatuse tõttu korrektselt tuvastada vastavas faasis olevat inimkeha. Implementeeritud automaatne faasituvastus töötab parimalt siis, kui Kinect kaamera kõrgus on umbes inimese kaela või rindkere kõrgusel ning kui inimene ei ole Kinectist liiga kaugel (umbes 4m).

Töö käigus omandas autor hulgaliselt uusi programmeerimisoskuseid ning sai esimese kogemuse liigutuste registreerimisel ja analüüsimisel põhineva teadusvaldkonnaga.

Tulemuseks saadi tarkvara, mis püstitab kõiki bakalaureusetöös esitatud eesmärke. Rakenduse efektiivsust kontrolliti erinevate inimeste peal, viies läbi implementeeritud harjutusi. Rakendus sobib hästi edasiseks arenduseks, näiteks masinõppega faaside tuvastamise funktsionaalsuse lisamine.

## Kasutatud kirjandus

- [1] S. Nõmm and A. Toomela, „An alternative approach to measure quantity and smoothness of the human limb motions.“, *Estonian Journal of Engineering*, 2013.
- [2] B. Galna, G. Barry, D. Jackson, D. Mhiripiri, P. Olivier and L. Rochester, „Accuracy of the microsoft kinect sensor for measuring movement in people with parkinson's disease.“ *Gait & Posture*, 2014.
- [3] L. Yang, L. Zhang, H. Dong, A. Alelaiwi and A. El Saddik, „Evaluating and Improving the Depth Accuracy of Kinect for Windows v2.“, *IEEE*, 2015.
- [4] L. Zhang, J. Sturm, D. Cremers and D. Lee, „Real-time Human Motion Tracking using Multiple Depth Cameras.“. *IEEE*, 2012.
- [5] “pterneas.com“ [Online]. Available: <http://pterneas.com/2014/03/13/kinect-for-windows-version-2-body-tracking/>  
[Accessed 9 February 2017]
- [6] “paulbourke.net“ [Online]. Available: <http://paulbourke.net/geometry/hyperspace/>  
[Accessed 13 February 2017]
- [7] “stackoverflow.com“ [Online]. Available:  
<http://stackoverflow.com/questions/4019831/how-do-you-center-your-main-window-in-wpf>  
[Accessed 15 February 2017]
- [8] “dotnetperls.com“ [Online]. Available: <https://www.dotnetperls.com/textfieldparser>  
[Accessed 19 February 2017]
- [9] “msdn.microsoft.com“ [Online]. Available: <https://msdn.microsoft.com/en-us/library/microsoft.kinect.jointtype.aspx>  
[Accessed 19 February 2017]
- [10] “stackoverflow.com“ [Online]. Available:  
<http://stackoverflow.com/questions/38206972/what-is-the-nature-of-xyz-coordinates-of-skeleton-with-kinect>  
[Accessed 20 February 2017]
- [11] “stackoverflow.com“ [Online]. Available:  
<http://stackoverflow.com/questions/21483999/using-atan2-to-find-angle-between-two-vectors>  
[Accessed 26 February 2017]

[12] “social.msdn.microsoft.com“ [Online]. Available: <https://social.msdn.microsoft.com/Forums/vstudio/en-US/fa0cf6b6-70b7-4181-bc9b-fe625cd5e159/angle-between-two-lines?forum=csharpgeneral>  
[Accessed 26 February 2017]

[13] “stackoverflow.com“ [Online]. Available: <http://stackoverflow.com/questions/39333217/grab-last-n-items-from-c-sharp-list>  
[Accessed 26 February 2017]

[14] “stackoverflow.com“ [Online]. Available: <http://stackoverflow.com/questions/19729831/angle-between-3-points-in-3d-space>  
[Accessed 20 March 2017]

[15] “stackoverflow.com“ [Online]. Available: <http://stackoverflow.com/questions/25814972/how-to-upload-a-file-to-amazon-s3-super-easy-using-c-sharp>  
[Accessed 2 April 2017]

[16] “aws.amazon.com“ [Online]. Available: <https://aws.amazon.com/sdk-for-net/>  
[Accessed 1 April 2017]

[17] “microsoft.com“ [Online]. Available: <https://www.microsoft.com/en-us/download/details.aspx?id=44561>  
[Accessed 9 February 2017]

[18] “zugara.com“ [Online]. Available: <http://zugara.com/how-does-the-kinect-2-compare-to-the-kinect-1>  
[Accessed 5 April 2017]

[19] “ionicons.com“ [Online]. Available: <http://ionicons.com/>  
[Accessed 12 February 2017]

[20] “blogspot.com“ [Online]. Available: [http://3.bp.blogspot.com/-MQFY-d0MV7c/UTOfFhg-U3I/AAAAAAAAABE/s4\\_toGdSVpI/s1600/esqueleto.png](http://3.bp.blogspot.com/-MQFY-d0MV7c/UTOfFhg-U3I/AAAAAAAAABE/s4_toGdSVpI/s1600/esqueleto.png)  
[Accessed 18 March 2017]

[21] “extremetech.com“ [Online]. Available: <https://www.extremetech.com/wp-content/uploads/2013/09/Kinect-640x353.png>  
[Accessed 14 April 2017]

[22] “richardhopton.blogspot.com.ee“ [Online]. Available: <http://richardhopton.blogspot.com.ee/2009/08/responding-to-mediapresentation-buttons.html>  
[Accessed 19 May 2017]

## Märkmed

Tööle on kaasa lisatud:

- Rakenduse lähtekood (Visual Studio projektifailina)
- Rakenduse installer
- Näitelogid
  - Up and go näitesalvestus
  - Up and sit näitesalvestus
  - Go näitesalvestus
  - Analüüsiraportid kõikide valikutega (liigesed, analüüsimeetodid, faasid ja kordused) eelnevalt nimetatud näitesalvestuste kohta tavalisel kujul ning tabeli kujul.



## Lisa 1 – Inimese püsti tõusmise kontrollmeetodid

```
public static void CheckPatientSittingUp(Body body) {
    double currentLegAngle = ProfileAnalyzer.GetLegAngle(body);
    if (currentLegAngle > minSittingUpAngleValue &&
        CheckSpineBaseDifForSittingUp(body)) {
        angleAtSittingUp = currentLegAngle;
        spineBaseAtSittingUp =
            ProfileAnalyzer.GetSpineBaseYCoordinate(body);
        sittingUp = true;
    }
}

public static bool CheckSpineBaseDifForSittingUp(Body body) {
    double dif = sittingUpAngleDif * spineBaseDifAvgPer1Degree;
    return (ProfileAnalyzer.GetSpineBaseYCoordinate(body)
        >= (dif + initialSpineBase)) ? true : false;
}
```

Joonis 11. Inimese püsti tõusmise kontrollmeetodid

## Lisa 2 – Inimese seismise kontrollmeetod

```
public static void CheckPatientStanding(Body body) {
    double calculatedAngle = ProfileAnalyzer.GetLegAngle(body);
    if (calculatedAnglesList.Count >
        lenOfElementsToCalculateAvgWithStanding
        && calculatedAngle >= minStandingAngleValue) {
        double avg = Enumerable.Reverse(calculatedAnglesList)
            .Take(lenOfElementsToCalculateAvgWithStanding)
            .Reverse().ToList().Average();
        if ((calculatedAngle > avg - avgDif)
            && (calculatedAngle < avg + avgDif)) {
            standing = true;
            standingInFrontOfChairSpineBaseValue =
                ProfileAnalyzer.GetSpineBaseYCoordinate(body);
        }
        standingInFrontOfChairLegAngle = calculatedAngle;
    }
    calculatedAnglesList.Add(calculatedAngle);
}
```

Joonis 12. Inimese seismise kontrollmeetod

## Lisa 3 – Inimese sammumise kontrollmeetod

```
public static void CheckPatientFirstStep(Body body) {
    double leftFootZ = ProfileAnalyzer.GetCurrentLeftFootAvgZ(body);
    double rightFootZ = ProfileAnalyzer.GetCurrentRightFootAvgZ(body);
    // Right foot goes first
    if ((leftFootZ < initialLeftFootZ + maxStationaryDistance) &&
        (leftFootZ > initialLeftFootZ - maxStationaryDistance)) {
        if (rightFootZ < initialRightFootZ - minFirstStepDistance) {
            firstStep = true;
            standingInFrontOfChairSpineBaseValue =
                ProfileAnalyzer.GetSpineBaseYCoordinate(body);
        }
    }
    // Left foot goes first
    else if ((rightFootZ < initialRightFootZ + maxStationaryDistance) &&
        (rightFootZ > initialRightFootZ - maxStationaryDistance)) {
        if (leftFootZ < initialLeftFootZ - minFirstStepDistance) {
            firstStep = true;
            standingInFrontOfChairSpineBaseValue =
                ProfileAnalyzer.GetSpineBaseYCoordinate(body);
        }
    }
}
```

Joonis 13. Inimese sammumise kontrollmeetod

## Lisa 4 – Inimese kaamera juures ümber pööramise kontrollmeetod

```
public static void CheckPatientTurningAtEnd(Body body) {
    float spineMidZ = ProfileAnalyzer.GetSpineMidZValue(body);
    if ((spineMidZ > turnAroundDistanceFromKinect -
        turningAtEndDifTowards) &&
        (spineMidZ < turnAroundDistanceFromKinect +
        turningAtEndDifAway)) {
        turningAtEnd = true;
        turningAroundAwaySpineMidZValue = spineMidZ;
        turningAroundAwaySpineMidZValueSet = true;
    }
}
```

Joonis 14. Inimese kaamera juures ümber pööramise kontrollmeetod

## Lisa 5 – Inimese tagasi kõndimise kontrollmeetod

```
public static void CheckPatientWalkingBack(Body body) {  
    float spineMidZ = ProfileAnalyzer.GetSpineMidZValue(body);  
    if (spineMidZ > turningAroundAwaySpineMidZValue +  
        walkingBackMinDif) {  
        walkingBack = true;  
    }  
}
```

Joonis 15. Inimese tagasi kõndimise kontrollmeetod

## Lisa 6 – Inimese alguses ümber pööramise kontrollmeetod

```
public static void CheckPatientTurningAtStart(Body body) {  
    float spineMidZ = ProfileAnalyzer.GetSpineMidZValue(body);  
    if ((spineMidZ >  
        standingInFrontOfChairSpineMidZValue - turningAtStartDifTowards)  
        && (spineMidZ < standingInFrontOfChairSpineMidZValue +  
            turningAtStartDifAway)) {  
        turningAtStart = true;  
    }  
}
```

Joonis 16. Inimese alguses ümber pööramise kontrollmeetod

## Lisa 7 – Inimese tagasi kõndimise järgse seismise kontrollmeetod

```
public static void CheckPatientStandingAfterWalkingBack(Body body) {
    double avgAnklesX = ProfileAnalyzer.GetAnklesXCoordinateDif(body);
    if (standingAnklesXCoordinatesList.Count >
        lenOfElementsToCalculateAvgWithStanding) {
        double avg = Enumerable
            .Reverse(standingAnklesXCoordinatesList)
            .Take(lenOfElementsToCalculateAvgWithStanding)
            .Reverse().ToList().Average();
        if ((avgAnklesX > avg - anklesXDifPercision)
            && (avgAnklesX < avg + anklesXDifPercision)) {
            standingAtStartAfterWalkingBack = true;
        }
    }
    standingAnklesXCoordinatesList.Add(avgAnklesX);
}
```

Joonis 17. Inimese tagasi kõndimise järgse seismise kontrollmeetod

## Lisa 8 – Inimese maha istumise kontrollmeetod

```
public static void CheckPatientSittingDown(Body body) {
    double currentLegAngle = ProfileAnalyzer.GetLegAngle(body);
    if (currentLegAngle < minSittingDownAngleValue
        && CheckSpineBaseDifForSittingDown(body)) {
        sittingDown = true;
    }
}

public static bool CheckSpineBaseDifForSittingDown(Body body) {
    double dif = sittingDownAngleDif * spineBaseDifAvgPer1Degree;
    return (ProfileAnalyzer.GetSpineBaseYCoordinate(body) <=
        (standingInFrontOfChairSpineBaseValue - dif)) ? true : false;
}
```

Joonis 18. Inimese maha istumise kontrollmeetodid

## Lisa 9 – Inimese konteksti lõpus istumise kontrollmeetod

```
public static void CheckPatientSittingAfterExcercise(Body body) {
    double calculatedAngle = ProfileAnalyzer.GetLegAngle(body);
    if (calculatedAnglesListAfterExcercise.Count >
        lenOfElementsToCalculateAvgWithAfterExcercise) {
        if (calculatedAngle <= minSittingAngleValueAfterExcercise) {
            double avg = Enumerable
                .Reverse(calculatedAnglesListAfterExcercise)
                .Take(lenOfElementsToCalculateAvgWithAfterExcercise)
                .Reverse().ToList().Average();
            if ((calculatedAngle > avg - avgDifAfterExcercise)
                && (calculatedAngle < avg + avgDifAfterExcercise)) {
                sittingAfterExcercise = true;
            }
        }
    }
    calculatedAnglesListAfterExcercise.Add(calculatedAngle);
}
```

Joonis 19. Inimese konteksti lõpus istumise kontrollmeetod

## Lisa 10 – Näide salvestatavast konteksti logifailist

```
Jan-Joonas Bernstein,
Up and go,
8,
1,
0,0,-0.1013,-0.9567,3.9781, .... 0.0403,-0.9318,3.7367
1,32,-0.1017,-0.9565,3.9778, .... 0.0408,-0.9369,3.7313
2,66,-0.1012,-0.9550,3.9797, .... 0.0429,-0.9424,3.7197
3,99,-0.1008,-0.9555,3.9787, .... 0.0420,-0.9415,3.7261
....
210,7045,-0.1003,-0.5441,2.3612, .... 0.2025,-0.6104,2.3208,2
211,7084,-0.1019,-0.5404,2.3276, .... 0.2021,-0.6160,2.2881,2
212,7108,-0.1030,-0.5396,2.2946, .... 0.2053,-0.6068,2.2706,2
....
370,12375,-0.0499,-0.6108,3.6130, .... 0.1704,-0.6786,3.4860,5
371,12404,-0.0500,-0.6109,3.6132, .... 0.1634,-0.6830,3.5038,5
372,12452,-0.0500,-0.6111,3.6134, .... 0.1605,-0.6782,3.5163,5
....
```

Joonis 20. Näide salvestatavast konteksti logifailist

## Lisa 11 – Näide analüüsitulemustest tabeli kujul

```
Sitting,Head
CombinedTrajectory,VelocityMass, .... ,Time,T/E, .... ,J/E,
0.4625517598,14.94532411, .... ,1.526000022,1.214528161, .... ,16045.36745,
0.8720804765,30.88303539, .... ,1.66900002,115.2441271, .... ,539511.4610,
0.847043314,28.0683335, .... ,1.480000019,,115.3137575, .... ,101764.9977,
. . . .
StandingUp,ElbowRight
CombinedTrajectory,VelocityMass, .... ,Time,T/E, .... ,J/E,
0.3270391457,9.630921496, .... ,1.125909467,33.1567209, .... ,27551.5100,
0.2700728804,8.79091710, .... ,1.048404995,34.12575667, .... ,24437.12963,
0.3961244563,13.40676350, .... ,1.214111870,41.09140561, .... ,39720.11021,
. . . .
StandingUp,HeadSpineBaseKneeRight
Start,End, .... ,Max,Avg,Sum,Values
92.1915870,162.100068, .... ,162.100068,131.706995,69.9084816,92.1915870, ....
92.752592,163.7024261, .... ,163.702426,133.269354,70.9498332,92.752592, ....
85.0305019,140.1337173, .... ,158.26333,130.616431,91.36245,85.0305019, ....
```

Joonis 21. Näide analüüsitulemustest tabeli kujul

## Lisa 12 – Näide analüüsitulemustest tavalisel kujul

```
0,Sitting,Head
CombinedTrajectory,0.462551759842729
VelocityMass,14.9453241115801
AccelerationMass,249.070148485966
JerkMass,6110.86114728468
EuclideanDistance,0.380848937414719
Time,1.52600002288818
T/E,1.21452816169746
V/E,39.2421315733004
A/E,653.986722863676
J/E,16045.3674592516
....
2,StandingUp,AnkleRight
CombinedTrajectory,0.342424927756359
VelocityMass,11.4977623649164
AccelerationMass,235.735650733153
JerkMass,10843.8414742167
EuclideanDistance,0.269219693230931
Time,0.700999975204468
T/E,1.27191634329155
V/E,42.7077314699034
A/E,875.62558260158
J/E,40278.7825217343
....
0,WalkingForwards,HipKneeAnkleLeft
Values,175.735127593,176.394954386,176.576322046,176.976049481, ....
Min,131.015682161941
Max,179.228660023334
Start,175.735127593738
End,174.359706342192
Avg,166.693262711221
Sum,174.615468902923
```

Joonis 22. Näide analüüsitulemustest tavalisel kujul

## Lisa 13 – Failide Amazon'i S3 Bucketisse üles laadimise meetod

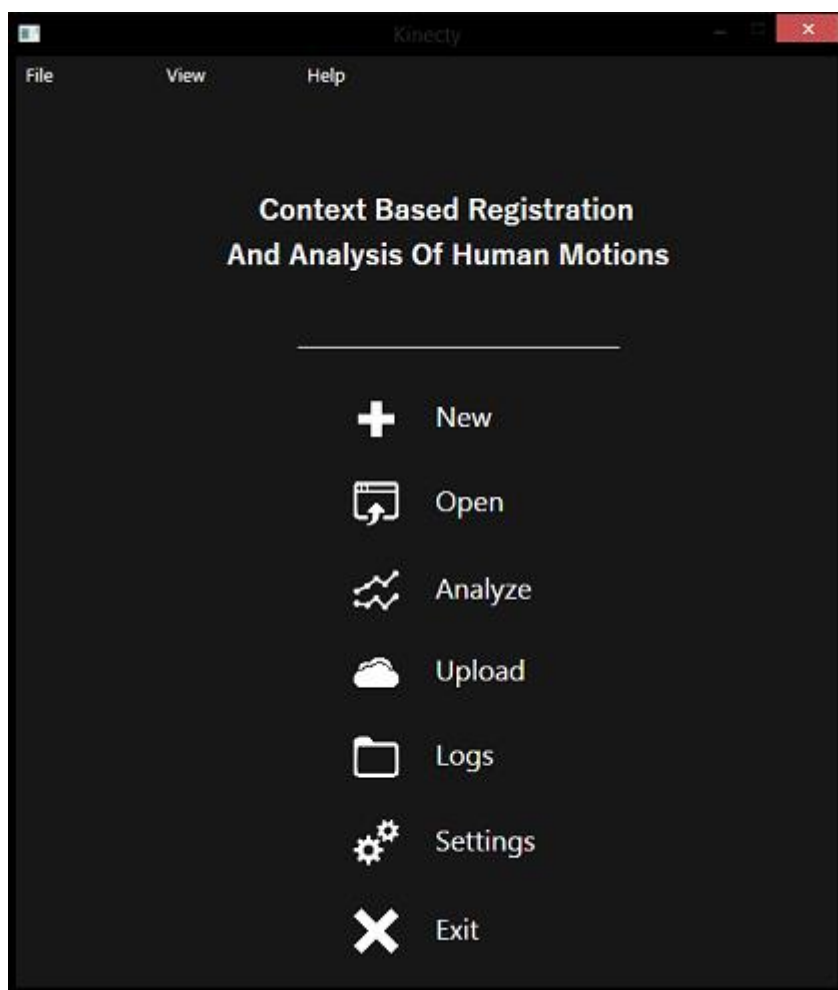
```
public bool UploadFileToS3(string localPath,
    string bucketSubDirectory, string newFileName) {
    AmazonS3Client client = new
    AmazonS3Client(Amazon.RegionEndpoint.EUCentral1);
    TransferUtility transferUtility = new TransferUtility(client);
    TransferUtilityUploadRequest request = new
    TransferUtilityUploadRequest();
    request.BucketName = (bucketSubDirectory.Equals(""))
        ? Properties.Resources.BucketName
        : (Properties.Resources.BucketName + @"/" + bucketSubDirectory);
    request.Key = newFileName;
    request.FilePath = localPath;
    request.UploadProgressEvent += new EventHandler<UploadProgressArgs>
    (UploadWindow.DisplayUploadInformation);
    transferUtility.Upload(request);
    return true;
}

Uploader.AWSUploader uploader = new Uploader.AWSUploader();
uploader.UploadFileToS3(fileToUpload, subDirectory, fileNameInAWS);
```

Joonis 23. Failide üles laadimine Amazon'i S3 Bucketisse

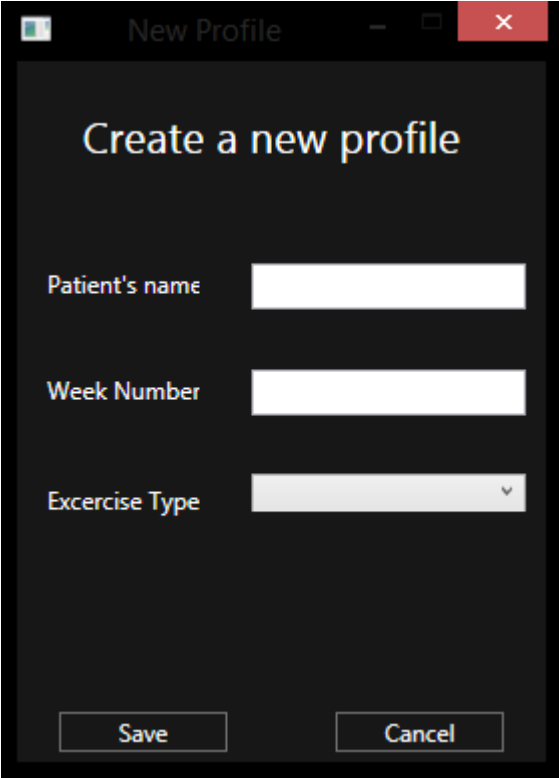


## Lisa 14 – Kasutajaliidese peamenüü



Joonis 24. Kasutajaliidese peamenüü

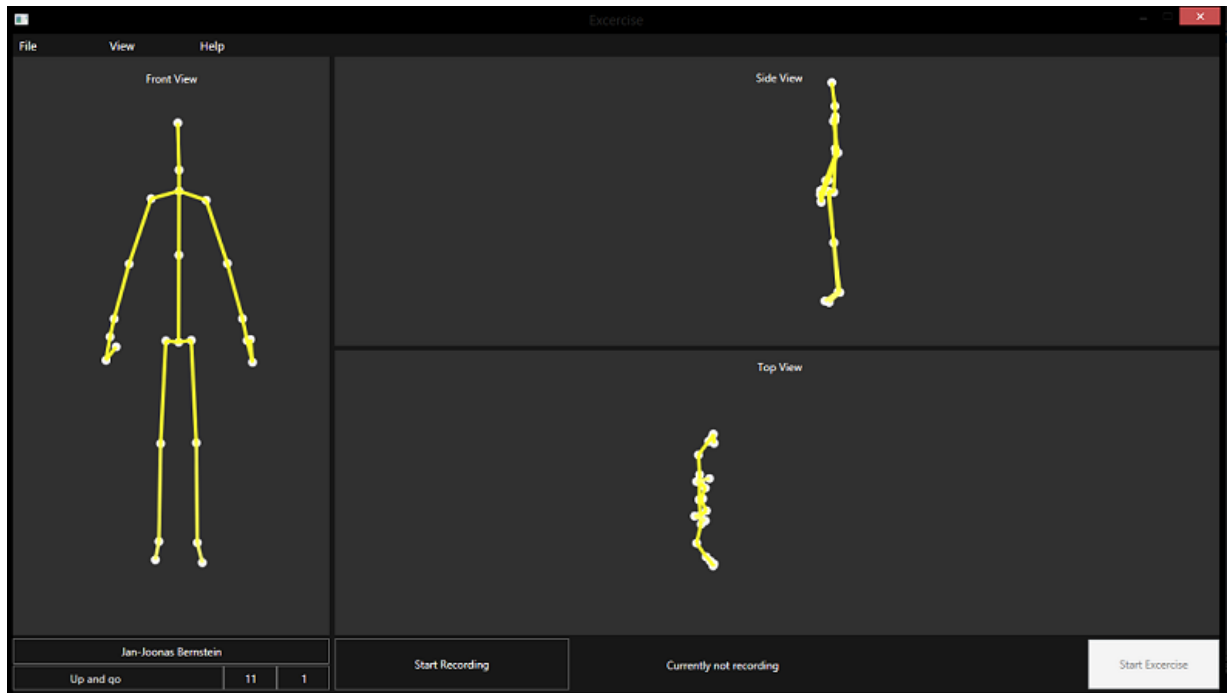
## Lisa 15 – Kontekstitüübi valik ning kasutaja andmed



The image shows a software dialog box titled "New Profile". The dialog has a dark background with white text. At the top, the title "New Profile" is displayed in a light gray font. Below the title, the main heading "Create a new profile" is centered in a larger white font. There are three input fields: "Patient's name" with a white text box, "Week Number" with a white text box, and "Exercise Type" with a white dropdown menu. At the bottom of the dialog, there are two buttons: "Save" and "Cancel", both with white text on a dark background.

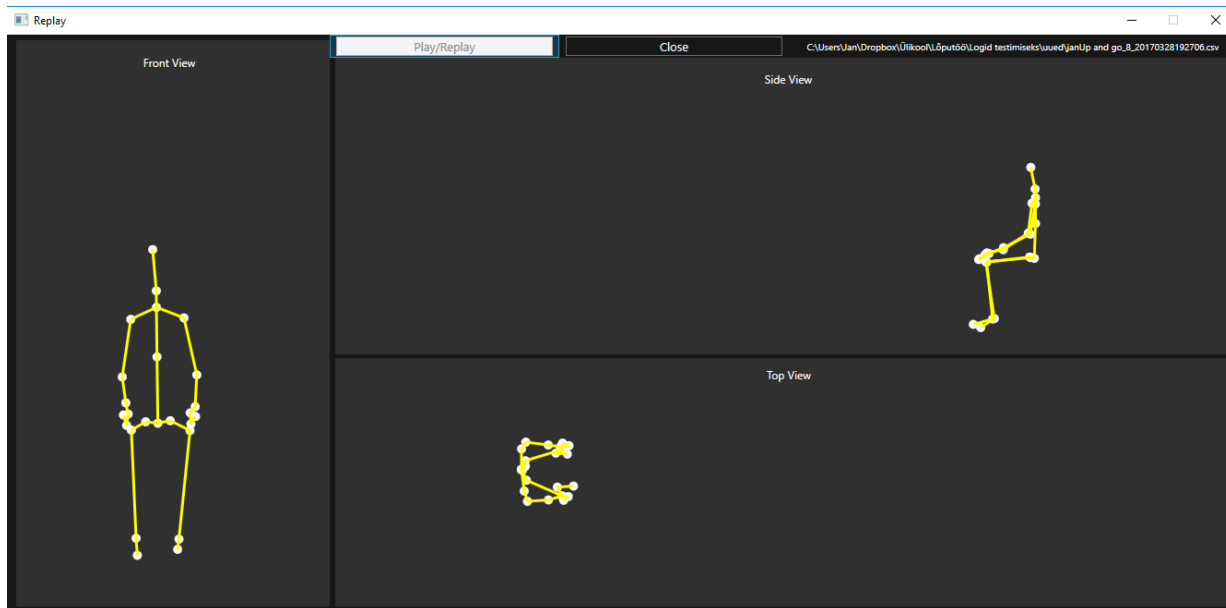
Joonis 25. Kontekstitüübi valiku kasutajaliides

## Lisa 16 – Harjutuse salvestamise vaade



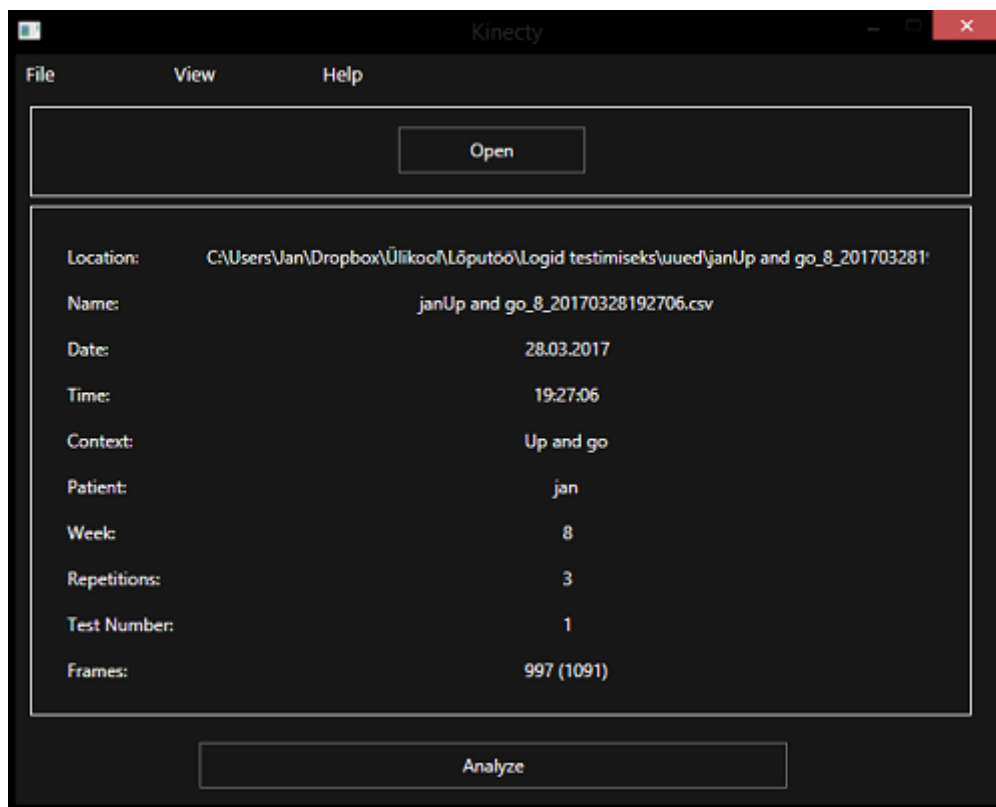
Joonis 26. Harjutuse salvestamise kasutajaliides

## Lisa 17 – Salvestuste järgi vaatamise vaade



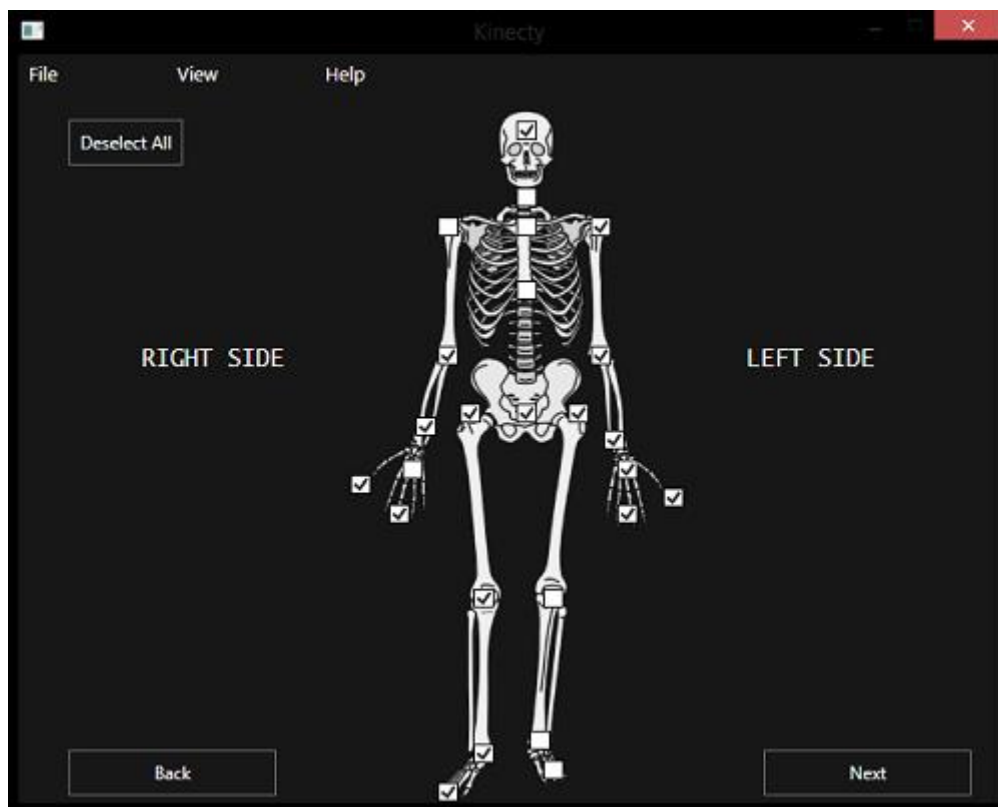
Joonis 27. Salvestuste järgi vaatamise kasutajaliides

## Lisa 18 – Analüüsitava faili valimise vaade



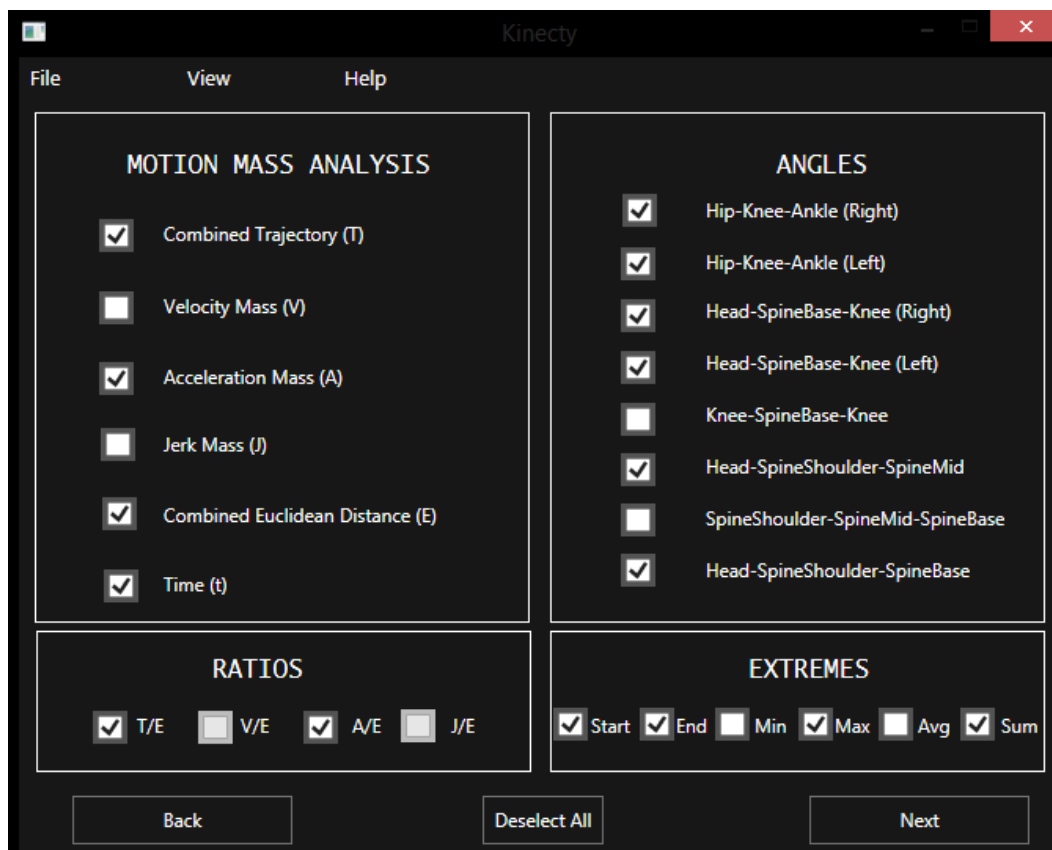
Joonis 28. Analüüsitava faili valimise kasutajaliides

## Lisa 19 – Analüüsitavate liigeste valimise vaade



Joonis 29. Analüüsitavate liigeste valimise kasutajaliides

## Lisa 20 – Analüüsitüüpide valimine



Joonis 30. Analüüsitüüpide valimise kasutajaliides

## Lisa 21 – Analüüsitavate korduste ja faaside valimise vaade

Kinecty

File View Help

### REPETITION(S) AND PHASE(S) SELECTION

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
All repetitions and all phases	All phases, but specific repetition	All repetitions, but specific phase	Specific repetition and specific phase

Repetitions: 9      Number of phases in one repetition: 3

Specific Repetition Specific Phase

Select a repetition: 1

Select a phase: WalkingForwards

Back      Next

Joonis 31. Analüüsitavate korduste ja faaside valimise kasutajaliides

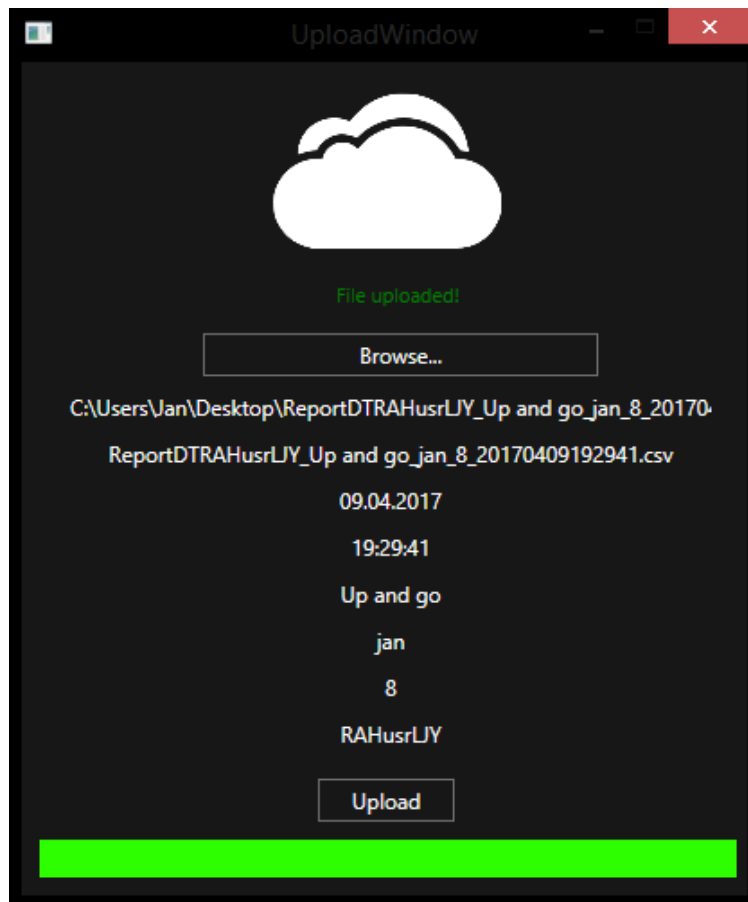


## Lisa 22 – Analüüsitulemuste kokkuvõtte vaade

Kinecty				
File		View		Help
Location: C:\Users\Jan\Dropbox\Ülikool\Lõputöö\Logid testimiseks\uuued\janUp and go_8_20170328192706.csv				
Date:	28.03.2017	<b>Joints</b> Head ShoulderLeft ElbowRight ElbowLeft SpineBase HipRight HipLeft WristLeft WristRight HandLeft ThumbRight ThumbLeft HandTipRight HandTipLeft KneeRight AnkleRight FootRight	<b>Motion Masses</b> CombinedTrajectory AccelerationMass Time CombinedEuclideanDis	<b>Angles</b> HipKneeAnkleLeft HipKneeAnkleRight HeadSpineBaseKneeLeft HeadSpineBaseKneeRight HeadSpineShoulderSpineM HeadSpineShoulderSpineBa
Time:	19:27:06			
Context:	Up and go			
Patient:	jan			
Week:	8		<b>Angular Extremes</b> Start End Max Sum	<b>Divisions</b> TdivE AdivE
Frames:	997 (1091)			
Repetitions:	1			
Phases:	WalkingForwards			

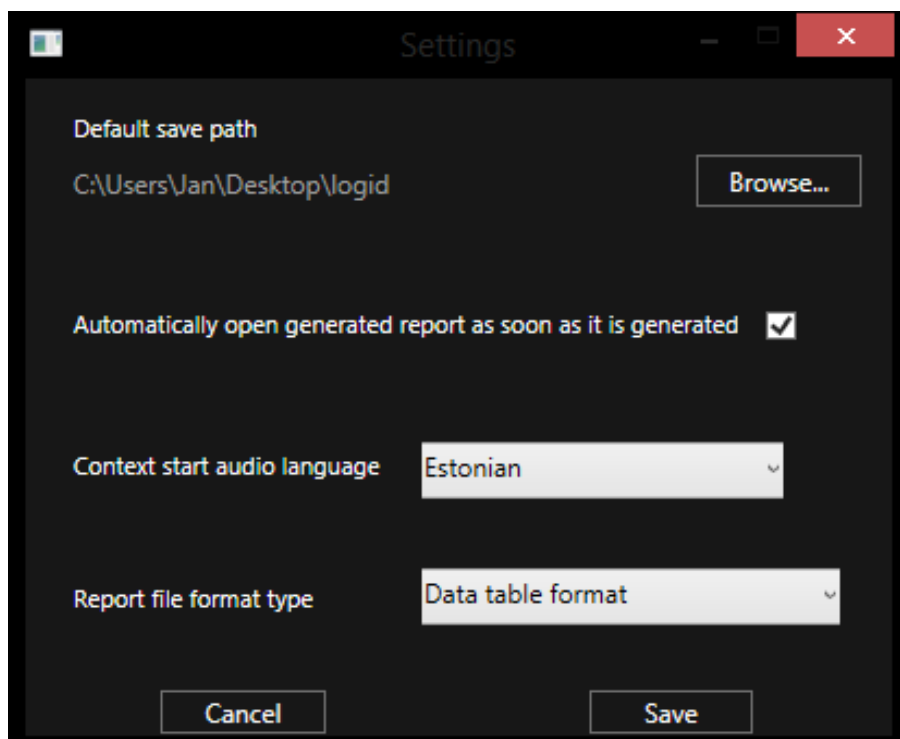
Joonis 32. Analüüsitulemuste kokkuvõtte kasutajaliides

## Lisa 23 – Failide üles laadimise vaade



Joonis 33. Failide üles laadimise kasutajaliides

## Lisa 24 – Sätete vaade



Joonis 34. Sätete kasutajaliides