

Movement testing and training Application

Project Number: 21-1-D-6



# Capstone Project Phase B

**Actors:**

Ayman Odeh - AimanOuda2309@gmail.com

Mhemmad Abdalla - Mhemmad01@gmail.com

**Supervisor:**

Dr. Anat Dahan

E-mail: [anatdhn@gmail.com](mailto:anatdhn@gmail.com)

Table of Contents

[1. Project Review and Process Description 3](#_Toc56702708)

[1.1. Project Review 3](#_Toc56702709)

[1.2. Process 3](#_Toc56702709)

1.3. [Verifications 14](#_Toc56702709)

2. User Documentation 15

2.1. User's Guide 15

2.1.1. General Description 15

2.2. Operation Instructions 16

2.2.1. GUI 16

3. Maintenance Guide 24

3.1. Package Diagram 24

3.2. Use Case 25

3.3. Data Base Tables 25

3.4. Running Instructions 27

4. References 27

# Project Review and Process Description

# Project Review

# Our project is an application that help diagnosis to discover people with motor problems and to help them to overcome these problems buy training modes, our modes in the project is Interpersonal Synchronization and Motor mode the diagnosis could choose which type he want to diagnose the diagnosed, the diagnosed could choose which mode he want to train to overcome his problems.

# The diagnosed will train by drawing a shape that the app display for him in each level in each stage and in training section he will get features like stars of his result and will move forward to the next level.

# Process

# Last semester we prepared the interface of our project in android studio and we saved it to use it this semester at the code section of the project, so we didn’t have to rebuild the interface this semester.

# In our project we worked in iteration which mean that we created a working version and each iteration we added features and tested it.

# One of the iterations was include studying and evaluating the synchronization evaluation algorithm.

# Also, we have tested our project on diagnosed to check if there is something missing in our project and we gave our project to diagnostic to get a feedback about the project.

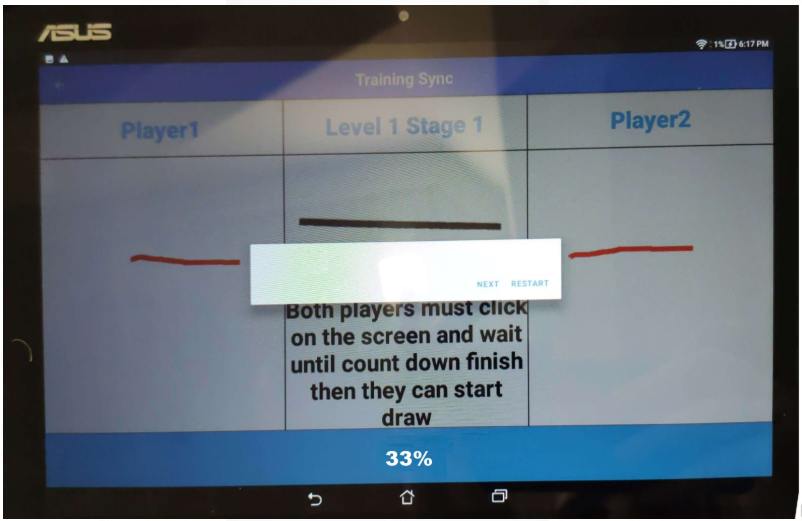
# Our main challenge in the code section of the project was to build the algorithm of synchronization, and make it work with a high accuracy.

# Last semester we displayed the algorithm of synchronization that we are going to build on our project, but after we build the algorithm and try it on our project, we were receiving an unexpected synchronization percent, so we decided to build a new algorithm.

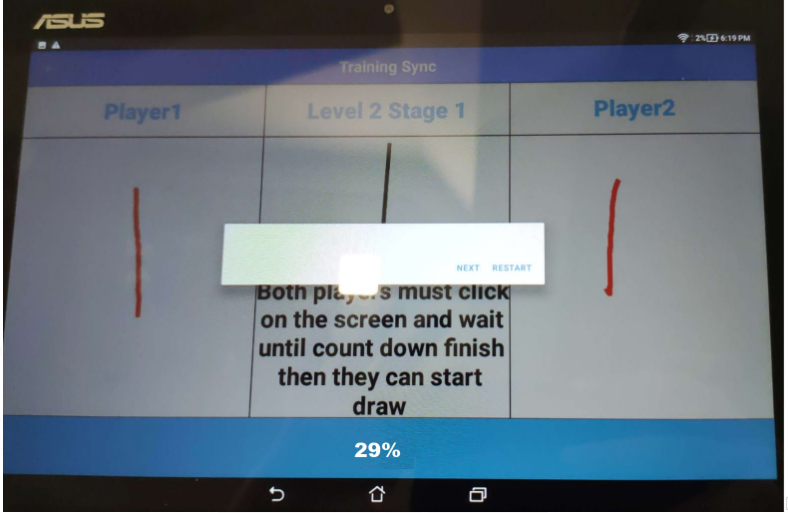
# Our last Algorithm (CVVAlgorithm Background 2.1.2 on phase 1):

**Results:**

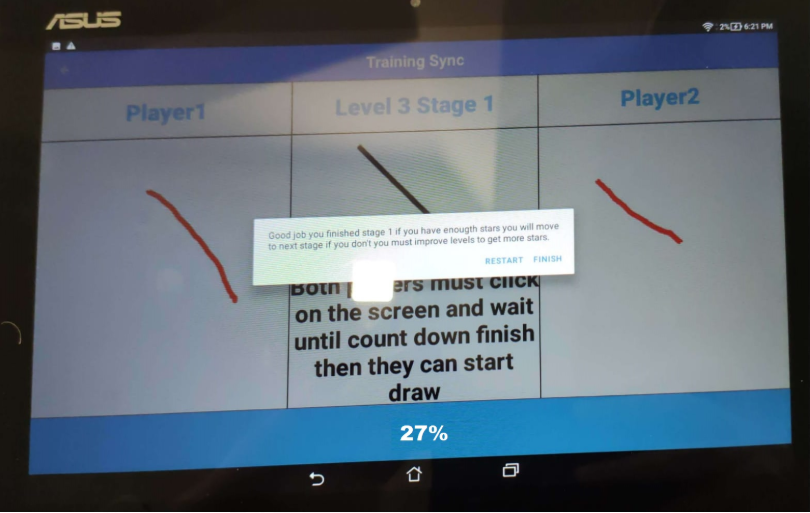
Here we are going to display the results of our last algorithm (CVV Algorithm), you can recognize that the percent is not appropriate to the drawing pictures:



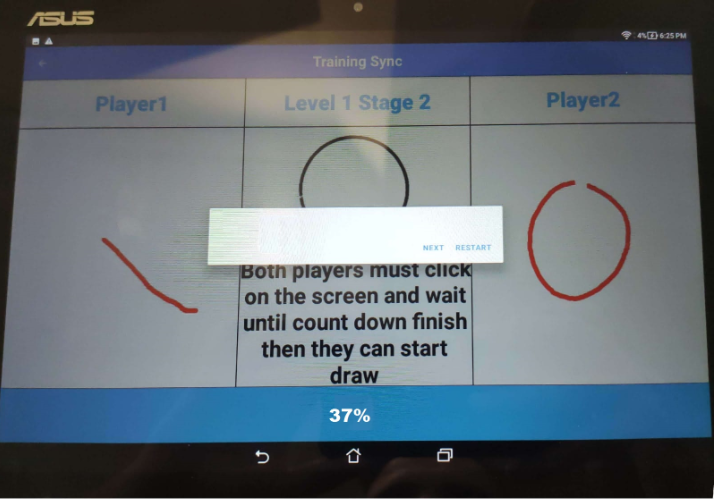
# Figure 1



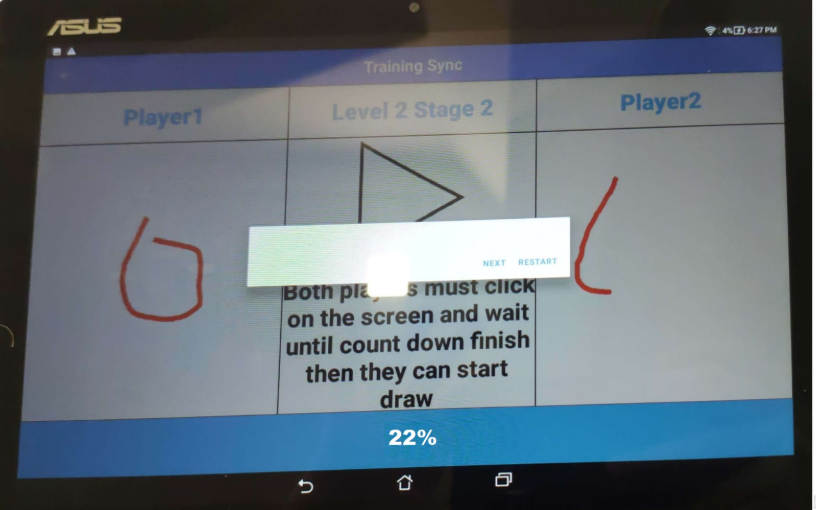
**Figure 2**

****

**Figure 3**

****

**Figure 4**

****

**Figure 5**

# In (Figure 1, Figure 2) the two players had to draw the displayed shape and they had to be synchronized with each other, in other word they had to draw the shape with as much as possible similar velocity and similar position at time t (Synchronization).

# As we can see in (Figure 1, Figure 2) the result of the two players was very similar, also they were drawing the shape with very similar velocity and Synchronize with each other while drawing with very similar position at time t, Although the result that we received shows us that the two players were not synchronize with each other (10% and 17% synchronize with each other).

# Our new Algorithm (Bool-Sync Algorithm):

# Each 100 (ms) we get the position of user1 and user2 and we saved it on array, so in this case we made the algorithm also related to the velocity of two players.

# We calculate the distance between each to point at the same index using the distance between two points formula:

# Distance

# Here we had a challenge how to convert the array of distances to percent, after a lot of thinking we decided to create new array of Boolean and we go over the distance array and if the distance is less the 55 we will add true to the Boolean array and if the distance if bigger than 55 we will add false to the Boolean array, at the end we sum the number of true of the Boolean array and we divided it by the size of the Boolean array then we multiple the result by 100.

# In this algorithm we have got a Synchronization with high accuracy, and each stage we make the true condition distance less than the last stage to make the application more challenge and more accuracy. For example: stage 1 the condition distance=55, stage 2 the condition distance= 50 and stage 3 the condition distance =45.

# Examples (All of this levels the two players were synchronize as much as possible):

# Stage 1 Level 1 (distance=55):

# 

# Figure 6

# Stage 2 Level 1 (distance=50):

# 

# Figure 7

# Stage 3 Level 1 (distance=45):

# 

# Figure 8

# As you can see the sync result and the sync picture of each stage becomes harder, because of the reduce of distance and the complex shapes.

# Another challenge that we have put on our app is that the user must draw the shape without move his finger up, if one player moved his finger he can’t continue the drawing and this could effect on the synchronization result, when the two players move their fingers up the application will calculate the synchronization and display it for the player at the training, if the user was doing a diagnosis of synchronization he can’t see the percent result only the diagnostic could see the result of the synchronization, also when the player want to pass to the next stage of motor or synchronization training he must have enough stars of the previous levels, the diagnostic will check the training result of the diagnosed and will give him the number of stars that he deserve.

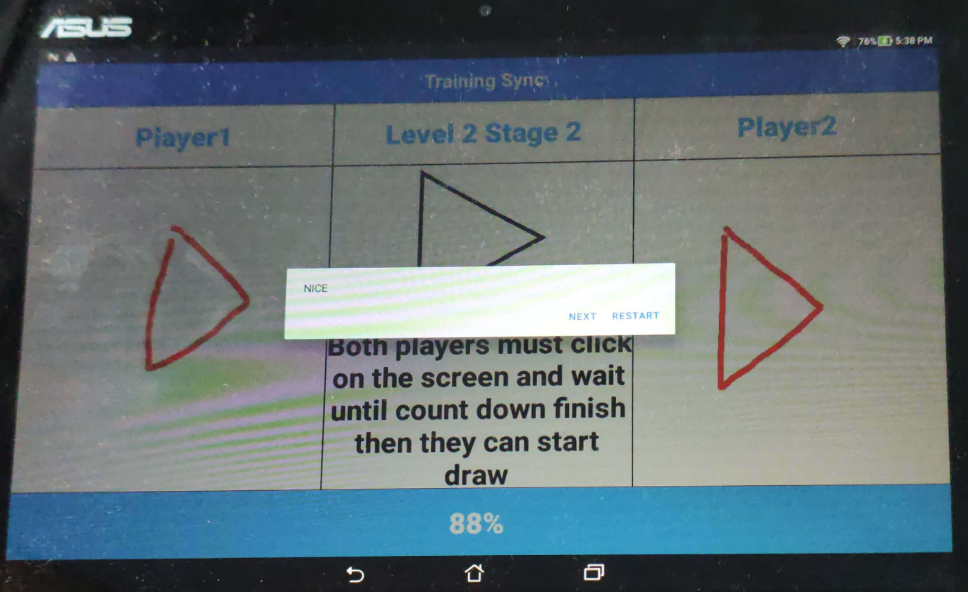
# Also, after the user pass each stage, he will get harder pictures to draw at the next stage, so this makes our application more interesting and more fun.

On our project we used DB MySQL and we connect it directly to the client side so we don’t need server side because there are no requirements for a server, we build the application on android studio and we uploaded it on GitHub so we can work in parallel.

GitHub Link:

[**https://github.com/mhemmad01/FinalProject.git**](https://github.com/mhemmad01/FinalProject.git)

**Here you can see Results of training sync with as much as possible synchronization drawing:**



# Figure 9

# 

# Figure 10

# 

# Figure 11

**Here you can see Results of training sync without synchronization, in other word they draw with different speeds and different positions:**

# 

# Figure 12

# 

# Figure 13

# As we can see here the result of the synchronization by using our new algorithm displaying the expected result when two players drawing with as much as possible synchronization and without synchronization.

# We divided the work on the app, each of as work on his part and if someone had a problem, we were trying to solve it together and after each of as finished his part we start working on the algorithm of synchronization together.

# When we tried our new algorithm and the demo app with diagnostics and diagnoseds, we got a feedback from the two type of users and we improve the app to make it more suitable to their requirements, the main problem that the users had is that the application must include more challenges and more fun, so because of that we added all of the challenges that we mentioned above and we make our app more interesting, after the users saw our final application they impressed and wanted to download the app on their tablets.

# So, we are so glad of our app result and during our work we had a lot of bugs and challenges on our app (challenges that we explained above) but we resolve all of it until we had this amazing app that we hope will help the diagnostic and people with motor and synchronization problems.

# In this table we can see the result of our last Algorithm (CVV Algorithm) and our new algorithm (Bool-Sync Algorithm):

|  |  |  |
| --- | --- | --- |
| CVV Algorithm | Bool-Sync Algorithm | Expected |
| 33% (Figure 1) | 88% (Figure 9) | Synchronized (80-90)% |
| 29% (Figure 2) | 76% (Figure 10) | Synchronized (70-80)% |
| 27% (Figure 3) | 63% (Figure 11) | Synchronized (60-70)% |
| 37% (Figure 4) | 30% (Figure 12) | Not Synchronized (30-40)% |
| 22% (Figure 5) | 16% (Figure 13) | Not Synchronized (10-20)% |

# 

# As we can see the result of Bool-Sync Algorithm is in our expectation result space and the CVV algorithm is out of expectation space.

# 

# Motor Section:

# In the motor section we used the berry VMI algorithm, where the user will draw a shape and will move to the next level until he finish all of the stage levels, then the user must wait until the diagnostic give him a stars for the shape that he drew, when the diagnostic Login in he can display the diagnosed drawing shapes and will give him a stars for the shape.

# Example motor section for the diagnosed:

# 

# Example motor section for the diagnostic, where he can give starts for the diagnosed shapes:

# 

# Verifications

# In our project we have been working in iteration which mean that we were creating a working version and each iteration we added features and test it, also we have tested our application with diagnostic and diagnosed to check if it is work correctly and if he has any comment to give us.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Test ID** | **Description** | **Excepted Result** | **Precondition** | **Comments** | **Result** |
| **1** | **AddDiagnosed Profile** | **The Diagnostic add new diagnosed profile for his diagnosis list** | **The System adds the profile to the list** | **User is logged in** | **Checking if the profile is already in the system** | **Pass** |
| **2** | **EnterDiagnosis** | **The Diagnostic Enter new diagnosis** | **The System allow for him to select diagnosed** | **User is logged in The Diagnostic has a diagnosed profile** | **Checking if the diagnostic has diagnosed to**  **diagnose them** | **Pass** |
| **3** | **ViewMyDiagnosed** | **The Diagnostic open My diagnosed list** | **The system shows for him his diagnosed** | **User is logged in There are already diagnosed in the system** | **Checking if the diagnostic already has diagnosed** | **Pass** |
| **4** | **SelectDiagnosedForResults** | **The Diagnostic open My diagnosed list then choose diagnosed profile** | **The system shows him results for this diagnosed and allow for him to score the results** | **The user is logged in, there are diagnosed profiles and there are results for this profile** | **Checking profile if exists and if there are results for this profile** | **Pass** |
| **5** | **Login** | **The user Enter details and then click on the login button** | **The system checks his details and if they are correct, show him the main page if not show him error message** | **The user enters the username and the password without any blank field** | **Checking if the user in the system** | **Pass** |
| **6** | **Registration** | **The user open registration screen, then fills the details and click on registration button** | **The System check if the details are matching rules and this username and email are not already in the system, then account will create successfully** | **The user enters all the details without any blank field** | **Check if Details (Email, Username, Password) are matching our roles** | **Pass** |
| **7** | **ImproveResults** | **The user opens previous training results and choose a level to improve** | **The system checks if there is a previous training to improve** | **User is logged in the user has trained before** | **Checking if there are a training results for this user** | **Pass** |
| **8** | **EnterTraining** | **The user decides to start new training** | **The System check if there a picture to display for the user** | **User is logger in** | **Check if there are photos in the database** | **Pass** |
| **9** | **LastAlgCorrectSync** | **Both users start drawing together with a high synchronization** | **The algorithm should return a high percent of synchronization** | **The users put their fingers on the screen and waited until count down finish to start drawing.** | **Checking validity of the algorithm (High Synchronization)** | **Fail** |
| **10** | **LastAlgBadSync** | **Both users start drawing with a low synchronization** | **The algorithm should return a low percent of synchronization** | **The users put their fingers on the screen and waited until count down finish to start drawing.** | **Checking validity of the algorithm (Low Synchronization)** | **Fail** |
| **11** | **BlankField** | **The user tries to login/register without filling one required field or more** | **The system shows him a message "Fill all the required fields"** | **The user clicks on login/registration button** | **Checking if there is a blank required field** | **Pass** |
| **12** | **LowStars** | **The user tries to move to the next stage in the training and he do not have enough stars** | **The system shows him a message "You don’t have enough starts you must improve some previous levels"** | **The user clicks on the arrow to move for the next level** | **Checking the sum of the stars for the user to decide if he can move for the next stage** | **Pass** |
| **13** | **NextLevel** | **The user moves to the next level of training** | **The system must upgrade the number of the current level and the shape of the new level** | **The user clicks on the arrow to move for the next level** | **Check the number of the current level and increase it by one and update the shape of the level** | **Pass** |
| **14** | **NewAlgCorrectSync** | **Both users start drawing together with a high synchronization** | **The algorithm should return a high percent of synchronization** | **The users put their fingers on the screen and waited until count down finish to start drawing.** | **Checking validity of the algorithm (High Synchronization)** | **Pass** |
| **15** | **NewAlgBadSync** | **Both users start drawing with a low synchronization** | **The algorithm should return a low percent of synchronization** | **The users put their fingers on the screen and waited until count down finish to start drawing.** | **Checking validity of the algorithm (Low Synchronization)** | **Pass** |
| **16** | **TestTheAppOnUsers** | **We test our app on diagnostic and diagnoseds** | **Feedback and comments.** | **Demo app that we can test the users on it.** |  | **Pass with some comments.** |

# User Documentation

# User’s Guide

# General Description

In our project we created an app that includes training and diagnosis section and each section contain synchronization part and motor part, the synchronization part has co-actors drawing a shape together and our application recognize if they are moving together (IS),the motor part include 1 actor and he will draw a shape, at the end our app will display for the diagnostic the pictures of the drawing and he will decide if the picture is scored or not scored after that our application will calculate the grade of the diagnosis.

## Unique features:

* Our project is helping people to recognize if they have problems with drawing or handling the pen, it is an application that will be used in experimental settings and assessed for therapeutic purposes, the app can also discover people with motor movements problems and help them by training to improve their skills.
* Also, the app can discover how much two people are related or similar with the way of drawing the shape.
* The project includes research and comparison of methods for evaluating Interpersonal Synchrony.

# Operation Instructions

# GUI

# 

# Fig. 1: Login Page

# 

# Fig. 2: Registration Page

# (Click on Create Account on Login Page to move to Registration Page)

# (The user must fill all the fields to register for the system)

# 

# Fig. 3: Diagnostic Main Page

# (Click on Login on Login Page to move to Diagnostic Main Page)

# (The Diagnostic could start diagnosis or display list of his patients and their diagnosis result or add new diagnosed to the list)

# 

# Fig. 4: Diagnosed Main Page

# (Click on Login on Login Page to move to Diagnosed Main Page)

# (The Diagnosed could start training or display list of his previous training levels and improve result of previous level)

# 

# Fig. 5: Diagnostic Select Page

# (Click on Start Diagnosis on Diagnostic Main Page to move to Diagnostic Select Page)

# (The Diagnostic must choose one diagnosed from the diagnosed list and must select mode of the diagnosis)

# 

# Fig. 6: Diagnosed Select Page

# (Click on Start Training on Diagnosed Main Page to move to Diagnostic Select Page)

# (The Diagnosis must choose mode of the training which he wants to train on it)

# 

# Fig. 7: Diagnosis Synchronize Page

# (Click on Synchrozation on Diagnostic Select Page to move to Diagnosis Synchronize Page)

# (This is the Synchronize diagnosis page where the diagnostic checks the diagnosed synchronization skills)

# 

# Fig. 8: Training Synchronize Page

# (Click on Synchrozation on Diagnosed Select Page to move to Training Synchronize Page)

# (The diagnosed train on synchronize with another player and he will get Synchronize percent result)

# 

# Fig. 9: Diagnosis Motor Page

# (Click on Motor on Diagnostic Select Page to move to Diagnosis Motor Page)

# (This is the Motor diagnosis page where the diagnostic checks the diagnosed Motor skills)

# 

# Fig. 10: Training Motor Page

# (Click on Motor on Diagnosed Select Page to move to Training Motor Page)

# (This is the diagnosed motor train page where the diagnosed Train on his Motor skills)

# 

# Fig. 11: My Diagnosed Page

# (Click on My Diagnoseds on Diagnostic Main Page to move to My Diagnosed Page)

# (In this page the diagnostic could see his diagnosed list, add diagnosed to his diagnosed list and click one of his diagnosed to display “Fig 12”)

# 

# Fig. 12: Diagnostic Checking Train Page

# (Click on Aiman Ouda on My Diagnosed Page to move to Diagnostic Checking Train Page)

# (In this page the diagnostic see the training results of Aiman Ouda and give him a stars)

# 

# Fig. 13: Diagnostic Checking Diagnosis Page

# (Click on diagnosis on Diagnostic Checking Page to move to Diagnostic Checking Diagnosis Page)

# (In this page the diagnostic see the diagnosis results of Aiman Ouda and give him a score)

# 

# Fig. 14: Improve Level Page

# (Click on my levels on Diagnosed Main Page to move to Improve Level Page)

# (If the diagnosed to see the num of stars or need to collect more stars to move for the next level of training he will enter this page to improve one of his previous training levels to gain more stars for his training)

# 

# Fig. 15: About Page

# (Click on About on Diagnosed/Diagnostic Main Page to move to About Page)

# (Information about the application and about the application Engineers)

# Maintenance Guide

# Package Diagram

# 

# Use case

# 

# Data Base Tables

# users:

# 

# sync:

# 

# 

# stars:

# 

# diagnosissync:

# 

# diagnosismotor:

# 

# 

# diagnoseds:

# 

# Running Instructions:

# 

# You must install android studio and import the project.

# (currently our GUI fits to Nexus 10 API 30).

# If you will use your own MySQL server you have to initials all the DB tables, otherwise you don’t have to install the DB because our remote SQL server is currently running, so you only need to run the app on android studio.

# You must be connecting to the internet while using the app.

# 

# References:

1. *Measuring the Dynamics of Interactional Synchrony*

*R. C. Schmidt • Samantha Morr • Paula Fitzpatrick • Michael J. Richardson*

1. *The mirror game as a paradigm for studying the dynamics of two people improvising motion together*

*Lior Noya,b, Erez Dekela , and Uri Alona,1*

1. *A time-varying measure of dyadic synchrony for three-dimensional motion*

*Philip T. Reiss,∗,1 Hila Z. Gvirts,∗,2,3 Rotem Bennet,4*

1. *Stallings, Loretta M. (1973). Motor Skills: Development and Learning.*

*Boston: WCB/McGraw-Hill.*

1. *"Fine motor control: MedlinePlus Medical Encyclopedia".*

*medlineplus.gov. Retrieved 2019-05-06.*

1. *"Fine Motor Outcomes in Preschool Children Who Receive Occupational Therapy Services".*

*Retrieved 26 October 2014.*

1. *"Fine Motor Skills & Activities for Infants & Toddlers". Early Intervention Support.*