Peg Board Strategy (Rehabilitation)

Devraj Nataraj

School of Computing Science

Newcastle University, Newcastle upon Tyne NE1 7RU, UK

Email:d.nataraj@ncl.ac.uk

Abstract—**In this paper I construe how simulating the test help us to find the strategy the patients used to perform the test faster. I also depict the graph showing the results of the test which can help the predicting the results. I also discuss about the simulators which are being used and how it has helped the patient and the doctor. Also the challenges faced when trying to implement it and the benefits of using this simulator.**

Keywords— **Nine-hole Pegboard test, Virtual simulation, Dexterity test, Hemiplegia test, Pegboard**

I. Introduction

Hemiplegia (sometimes called hemiparesis) is a condition affecting one side of the body (Greek 'hemi' = half). We talk about a right or left hemiplegia, depending on the side affected. It is caused by damage to some part of the brain, which may happen before, during or soon after birth, when it is known as congenital hemiplegia, or later in childhood as a result of injury or illness, in which case it is called acquired hemiplegia. Generally, injury to the left side of the brain will cause a right hemiplegia and injury to the right side a left hemiplegia. Childhood hemiplegia is a relatively common condition, affecting up to one child in 1,000. About 80% of cases are congenital, and 20% acquired. To assess we use the 9-hole pegboard test (2 adjacent boards; 3 peg sizes; electronically timed)[1].

The Nine Hole Pegboard Test (NHPT) was developed to measure dexterity also known as manual dexterity. Also the NHPT was first introduced by Kellor, Frost, Silberberg, Iversen, and Cummings in 1971. In 1985, norms for the NHPT in healthy individuals were established by Mathiowetz, Weber, Kashman, and Volland[3].

The NHPT test is done in our project to minimize or cure hemiplegia. This test can be classified into to fields namely

- Unilateral
- Bimanual

Unilateral is using a single hand to move the pegs; Bimanual is using two hands to move the pegs. The test is performed on both male and female with age group 4-90 years, and on the Game Genre.

When the test is performed its mainly categorized into 2 main parts and they are based on the Game Genre which are action games playing patients and non-action games playing patients and the data is recorded like the name, age, size of pegboard, gender, time taken to pick and place the peg, total time, which peg did he pick and to which peg did he place and the distance between them and lastly the dominant hand and the direction in which the pegs are moved are also recorded. After the test was performed there was a significant difference in the test between the action games playing patients vs non-action games playing patients which was the test was performed faster by action games playing patient but why did they perform faster and which strategy they used to perform was a mystery so to find out which kind of strategy or path they used to perform faster ,I am creating a simulator which reads the recorded data from the test and recreates the test which helps us to find which kind of strategy or path they used.

II. Background and Related Work

Numerous virtual simulations have been carried out on different test in different fields in this section we discuss various other virtual simulator done by other people.

A. Virtual Reality-Based Orthopaedic Tele-rehabilitation

The numbers of patients that need rehabilitation have increased in recent years and also have been problematic for the patients to get to travel to and fro to hospitals. The doctors would lend them the exercising devices but they couldn't track the patient progress so they came up with an idea of Tele-rehabilitation which is remote monitoring the patient progress by installing a home rehabilitation.

Timeliness and duration of rehabilitative therapy are especially problematic for those in remote rural locations or living in depressed urban areas. In such instances there are no clinics in the vicinity of the patient's home. Patients with orthopaedic impairments, such as those that have had hand or knee surgery, typically follow a regimen of combined clinic and home rehabilitation exercises. Home exercises are currently done on simple mechanical systems that are loaned to the patient. Since these mechanical devices are not networked, there is no way a therapist can either monitor patient's progress at a distance, or change exercise effort (difficulty) levels from the clinic. There is also no way to

Verify that indeed the patient has done the prescribed home rehabilitation exercises, and some patients feel less motivated to exercise at home without direct medical supervision.

Station in turn consists of a force feedback glove called the "Rutgers Master," a multipurpose control interface connected to the PC, a net camera, and a microphone array . The sensing glove measures finger grasping and abduction/adduction motion and sustains resistive forces up to 16 N at each fingertip. It weighs only 100 grams due to the use of pneumatic actuators. These are custom glass-graphite pistons that have extremely low friction and high dynamic range. The actuators can sustain high forces without overheating and damage. The similar device with fewer graphic is placed in the doctor's location [6].

In the home station while the patient is performing the exercise connecting to the station the doctor can simultaneously check the patient progress and keep records of it. This simulation has helped a lot of patients by letting the patients stay home and take the test as it suits for them and also helped the doctors to keep track of the patient's progress. Also it has minimized the time to travel to and fro to the hospital and stress to patient by making them feel at home and still have the therapy with doctor examining them.

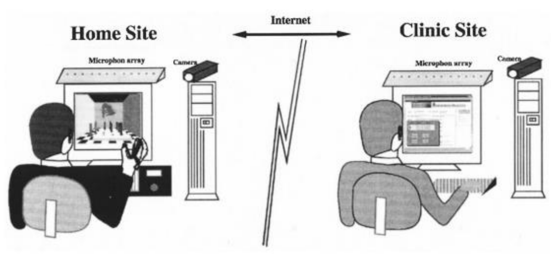


Figure 1: The tele-rehabilitation system architecture

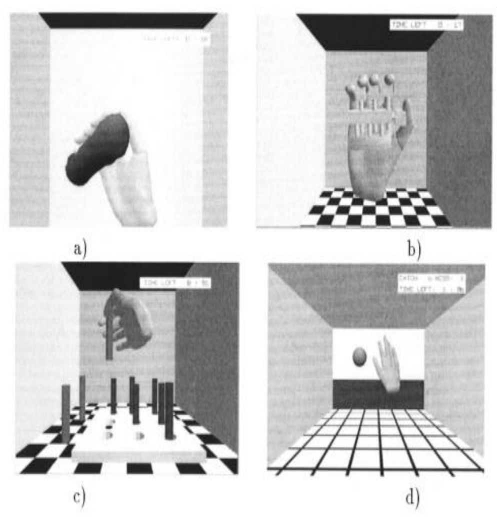


Figure 2: Virtual rehabilitation exercises. (a) Power putty. (b) DigiKey (c) Peg board. (d) Hand ball

The rehabilitation software library has three main components, the3-D graphics environment, the patient database, and the graphical user interface (GUI). The GUI is the combination of windows and menu buttons that allow a computer novice to operate the system with a mouse. The 3-D graphics library was built using the World Toolkit commercial software library and contains virtual reality modules, each dedicated to a given exercise. Each module currently depicts a virtual hand which mimics the patient's right hand. The hand 3-D model was ported from the Viewpoint DataLabs library, and segmented to allow independent finger motion. The virtual hand interacts with other virtual objects, within an exercise room. The room has a tiled floor for better 3-D perspective, since the image displayed is monoscopic. The overall scene complexity is approximately 1500 polygons, allowing a graphics refresh rate of about 30 frames/s.

The VR simulations are designed for physical rehabilitation and for functional rehabilitation. The physical rehabilitation modules are rubber ball squeezing, the virtual power putty and the virtual DigiKey[see Fig. 2(a), (b)]. These are designed to increase finger force exertion and range of motion. The functional rehabilitation exercises are the ball game and the peg board [Fig. 2(c), (d)]. Each exercise has several levels of difficulty, and has an allowed amount of time. For example, the virtual DigiKey is color-coded (similar to the real one) to indicate maximum force exertion levels. The peg board level of difficulty relates to the peg-hole tolerance, while the ball game has "fast" or "slow" ball velocity levels.

B. Virtual Reality Simulator for Robotic Surgery

This is an open surgery simulator and the simulator is equipped with 8 motor and brake controls, and 2 grip encoders. The user can view the virtual environment in 3dimensions using a simulator-mounted stereo eyepiece that emulates the surgical system by which the surgery can be performed. It is classified in to 2 stages novice and experienced and at the beginning of the test the user can select which type of level he is from and can select it, usually the students go for the novice and the experienced are taken by the doctors and test is performed.

After a standardized introduction and 10 minutes of practice, each subject completed 2 virtual EndoWrist modules ("Pick and Place" and "Peg Board") and 2 virtual needle-driving modules ("Dots and Numbers" and "Suture Sponge") in sequence (Fig. 2). "Pick and Place" requires the user to manipulate a ring over a series of cone-shaped targets. During "Peg Board" training, the user picks up a sequence of rings from pegs, performs a transfer from 1 instrument to another, and places the ring on a new peg. "Dots and Numbers" entails loading a needle and driving it at several oblique angles around a circle. "Suture Sponge," a module still in development, involves driving a needle into a sponge from foreground to background [7].

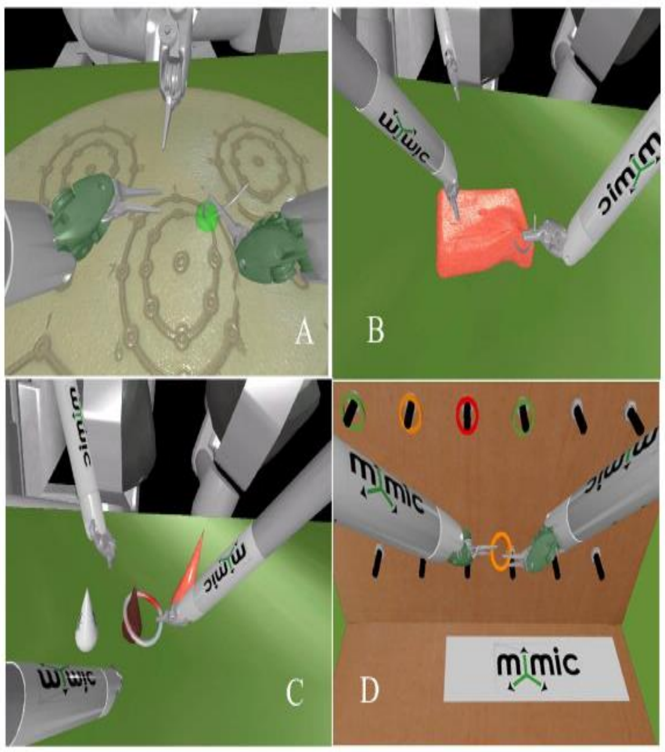


Figure3 :(A)Dots and Numbers (B) Suture Sponge (C) Pick and Place and (D) Pegboard

The scores or performance is recorded using the scoring algorithm this measure the total number of seconds he has taken to complete the task ,maximum force applied when operating and various other factors are taken it into consideration and a total score percentage is reported.
When using this kind of approach it helps the doctors or the students to get a visual training before performing the operation and get used to the things which can encounter during the operation and be prepared for it. The scoring algorithm will also give them the score by which they can judge how they performed in the open surgery and improvise.

C. *Unity*

Unity is the ultimate tool for video game development and is cross platform game engine with built-in IDE developed by Unity Technologies. It is used to develop games in all platforms like consoles, web plugins, mobile devices etc.
Unity supports art assets and file formats from 3ds Max, Maya, Softimage, Blender, modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks and Algorithmic Substance. These assets can be added to the game project, and managed through Unity's graphical user interface.
Unity is an organized, accessible platform brimming with tools that make getting up and running pretty easy. It features three scripting languages: UnityScript (similar to ActionScript — however, they just call it JavaScript), C# and Boo.

Unity tools help us to deliver reliable performance, smooth frame rate, reduce graphics bottlenecks also with unity it renderers thing which can only be seen thing which can be seen are not drawn. To improve performance Unity automatically combines your small geometry into batches at runtime. You can also batch larger static objects by creating geometry batches at build-time.

With an emphasis on portability, the graphics engine targets the following APIs: Direct3D on Windows and Xbox 360; OpenGL on Mac, Windows, and Linux; OpenGL ES on Android and iOS; and proprietary APIs on video game consoles. Unity allows specification of texture compression and resolution settings for each platform the game supports, and provides support for bump mapping, reflection mapping, parallax mapping, screen space ambient occlusion (SSAO), dynamic shadows using shadow maps, render-to-texture and full-screen post-processing effects. Unity's graphics engine's platform diversity can provide a shader with multiple variants and a declarative fallback specification, allowing Unity to detect the best variant for the current video hardware; and if none are compatible, fall back to an alternative shader that may sacrifice features for performance.

The game engine's scripting is built on Mono, the open-source implementation of the .NET Framework. Programmers can use UnityScript (a custom language with ECMAScript-inspired syntax, referred to as JavaScript by the software), C#, or Boo (which has a Python-inspired syntax).

Unity has a built-in path finding which helps you to bring your scene to life with automatic navigation .Also it contains a powerful 3D physics engine powered by Nvidia.

1) *Reading from file*

In unity we cannot import the dll files directly and use them as we do in the visual studio they have to be converted into specific MonoDevelop support dll file and then make it compatible so that the MonoDevelop understands it and from there we can pass values generate the required results and store them or return the values to the mono develop so that it can read these values and access them in the function according the user needs.

2) *Precision Timing*

In unity we can get precision timing by using the scripting API:Time[8] which is associated with the unity SDK so by this we can work on the timing in a more accurate way and have no problems related to timing.

3) *Blender*

Blender is free and open-source 3D computer graphics software product used for creating animated films, models, animation, visual effects, art ,3D Printed models.

The features of Blender are

- Support for a variety of geometric primitives, including polygon meshes, fast division meshes etc.
- Real-time control during physics simulation and rendering.

Blender makes it possible to perform a wide range of 3d-content-creation-oriented tasks. Therefore it may seem daunting when first trying to grasp the basics. However, with a bit of motivation and the right learning material, it is possible to be productive with Blender after a few hours of practice. If you're reading this wiki, it is a good start, though it serves more as a reference. You also have online video tutorials (free and paid) from specialized websites, and several books in the Blender store.

Despite everything Blender can do, it remains a tool. Great artists create masterpieces, not only by pressing buttons or manipulating brushes, but also by learning and practicing human anatomy, color theory, composition, lighting, traditional animation, photography, psychology and many other areas. 3D content creation software have the added technical complexity and jargon associated with the underpinning technologies. CPUs, GPUs, memory, algorithms, vectors, materials, meshes are the mediums of the digital artist, and understanding them, even broadly, will help you using Blender to its best.

In unity the blender works so well that once you create an object in blender we can just drag and drop into the asset and directly use them on to project.

III. PROJECT APPROACH

In this section of the paper we discuss the benefits of our project and how the simulation of the project helps us to determine the strategy the action games playing patients vs the non-action games playing patients.

In the NHPT we use a pegboard which consist of small holes for pegs on one part also it consist of pegs which is placed on the top of the hole and on the other part of the board holes are empty to place the pegs from the first part of the board. Pegboard can be of different sizes and different patterns and which is used for different purposes but the pegboard which we are discussing consist of 9 holes on one part of the board and 9 holes on other part of the board also it has a barrier or a partition through which the pegs cans be passed from one board from one hand to the other hand to the other board and place it to the hole. There is another scenario or method in which the partition or barrier is removed and the pegs have to be moved in one hand without using the other.

In the Unilateral test the middle barrier is removed and the test is performed. But in the Bimanual the pegs are transferred from one hand to the other through the middle barrier so in the bimanual test the barrier is placed in between the boards. The analysis is made using General Linear Model Analysis of Variance (MANOVA).

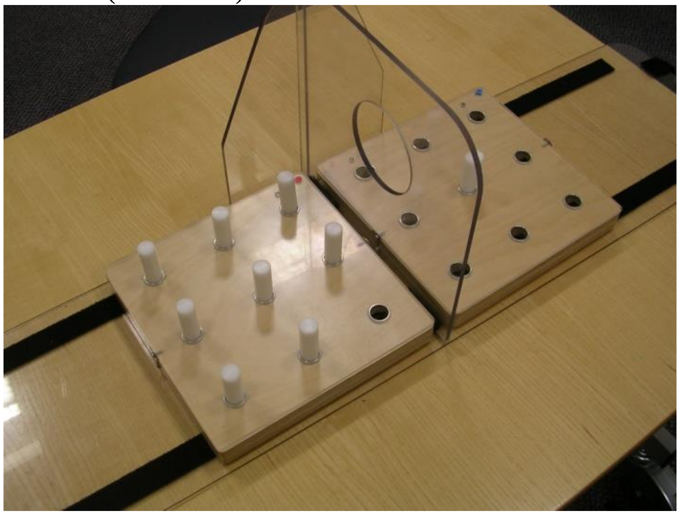


Figure 4:Pegboard

The pegboard which we are discussing is connected to the computer and the readings are recorded using the custom written software. The peg board design has two pegboards placed side by side so that the pegs are upright and easier to access for grasping. There are three peg sizes (small, medium and large). At the bottom of each hole there is an infra-red light source opposite a photovoltaic detector, allowing detection and electronic coding (including timing) of events in which pegs are removed or replaced.

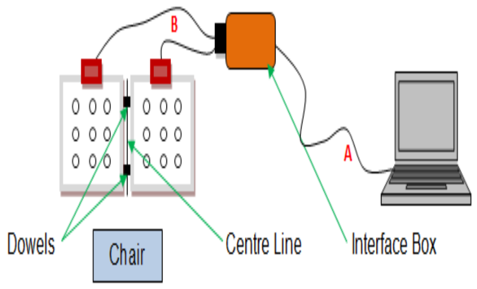
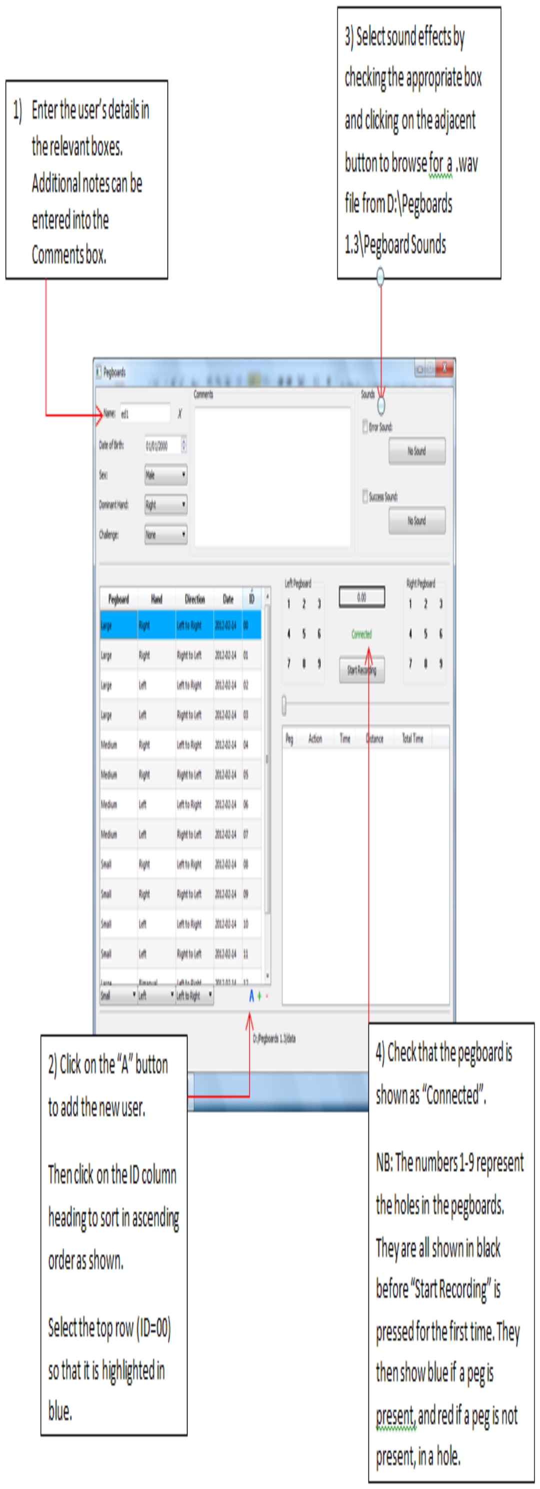


Figure 5: Circuit Connenection of pegboard

In the custom written software we can enter number of details which helps us to get details of the patient and once the test is conducted the results are stored and can be accessed through which the physician can know how the patient is and prescribe medication to the patient.


1) Enter the user's details in the relevant boxes. Additional notes can be entered into the Comments box.

3) Select sound effects by checking the appropriate box and clicking on the adjacent button to browse for a .wav file from D:\Pegboards 1.3\Pegboard Sounds

2) Click on the "A" button to add the new user.

Then click on the ID column heading to sort in ascending order as shown.

Select the top row (ID=00) so that it is highlighted in blue.

4) Check that the pegboard is shown as "Connected".

NB: The numbers 1-9 represent the holes in the pegboards. They are all shown in black before "Start Recording" is pressed for the first time. They then show blue if a peg is present, and red if a peg is not present, in a hole.

Figure 6: Software version of pegboard which records the data

We use this pegboard to assess dexterity [4] among the hemiplegia [2] patients. The test was performed on all age group between 4-90 years for all peg sizes like the pegs with small, medium and large and also 2 methods or task are performed the bimanual task and unilateral task which is using both the hands to transfer the pegs and using single hand to transfer the pegs. When they considered these two there wasn't much effect in the test but there was lot of effect in the test and values began to vary when they considered the Game Genre [5] for the large pegs for both hands and for the medium pegs for the non-dominant hand. In the test the patients who played action games completed the test faster than the patient who doesn't play action games. There was a highly significant main effect for Age Group for all peg sizes and the bimanual task which is depicted in the graph below. There was a main effect for Game Genre for the large pegs for both hands and for the medium pegs for the non-dominant hand which can be noticed in the table and the graph. There were no Age*Genre interactions.

	df	F	p
Large Pegs Dominant Hand	1	7.12	0.008
Large Pegs Non Dominant Hand	1	12.76	<0.001
Medium Pegs Dominant Hand	1	2.38	0.125
Medium Pegs Non Dominant Hand	1	6.71	0.011
Small Pegs Dominant Hand	1	2.39	0.124
Small Pegs Non Dominant Hand	1	2.93	0.09

Table 1:Dexterity and Game Genre

It was strange because the action games playing patients completed the test faster and it left a mystery behind what kind of strategy they performed in order to complete the test quicker. So in order to reveal the mystery and study on the strategy I am creating a simulator which recreates the test and give us good understanding which kind of strategy or path the patient used to perform the test faster.

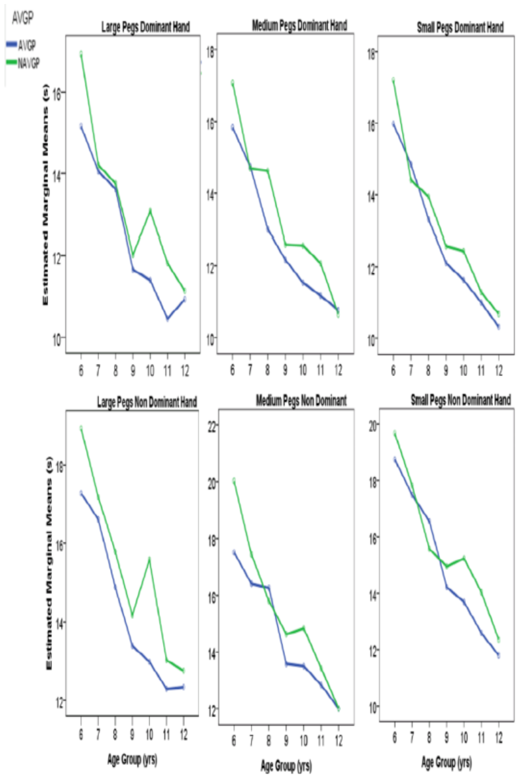


Figure 7: Graph of different sizes of peg board.

In this paper two approaches are considered: Virtual Reality-Based Orthopedic Tele-rehabilitation presented by Grigore Burdea, Viorel Popescu, Vincent Hentz, and Kerri Colbert and Virtual Reality Simulator for Robotic Surgery presented by Patrick A. Kenney, Matthew F. Wszolek, Justin J. Gould, John A. Libertino, Alireza Moinzadeh. In the first approach they have discussed how the test can be done remotely without stressing the patients and thereby helping the doctors to keep track of the patients. In the second approach they have discussed how to virtually recreate the surgical system and perform the test.

The first approach is not that useful for my project because I have the data to recreate the test so I don't need to create 2 station which is home and clinic to record the data and find the strategy but this approach can be used for future implementation by reducing the patient stress of travelling to and fro to the hospital.
The second approach outstands because they have recreated the surgical simulator which is similar to my project which is I need to recreate the test from the data which was stored when the test was performed and also the virtual 3d environment can be used as a future implementation of my project.

In my project I use unity as a tool for development because of Multiplatform compatibility, scripting language is organized and gives us more flexibility to use and control, user interface is pretty good the basic stuffs can be got by drag and drop. The scripting language is so well structured if you want to work on other libraries we have to store the.dll files in the asset folders which later can be accessed in the script. The precision timing which is the most important factor in the project can be easily achieved with the Scriping: API present in the unity library. Also to create the replica of the pegboard which is used in the actual test I use Blender to draw the model and add textures to it. Blender and unity works very well together.

IV. Implementation

In this project I have virtually simulated the pegboard and the GUI look as simple as possible and can be accessed and used as user friendly as possible. I have also implemented different view modes in which the user can look at the pegboard on any view angle as possible. Also I have implemented graphs to get suitable results and compare against other results and get to a conclusion.

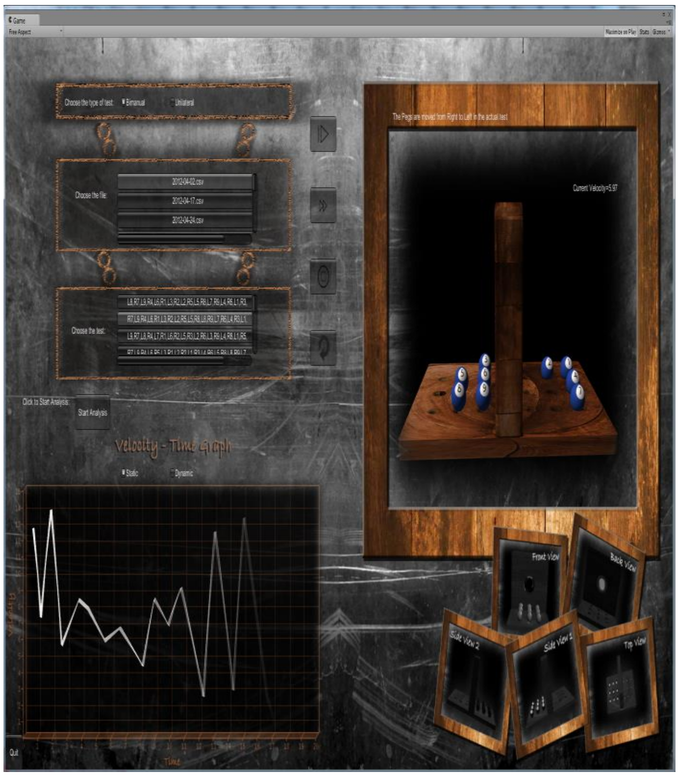


Figure 8: Bimanual Test.

Current Velocity of the Pegs is also displayed so that the user can know at which velocity the patient has moved the pegs. In our project the test Is mainly classified into two types of test namely

- Bimanual
- Unilateral

In these two tests the pegboard will not remain the same, In Bimanual test the pegs are picked up and passed through the hole which is in the center of the pegboard to the other hand and placed on the respected position, In Unilateral test the center part is removed and the pegs are picked and placed to the respective position. The Figure 8 is an example of Bimanual test and Figure 9 is the example of Unilateral test.

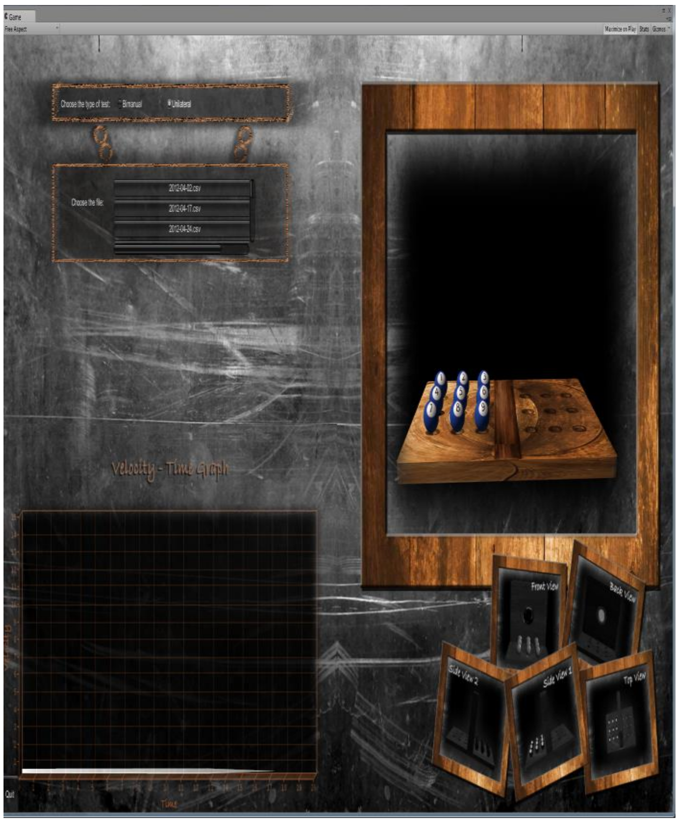
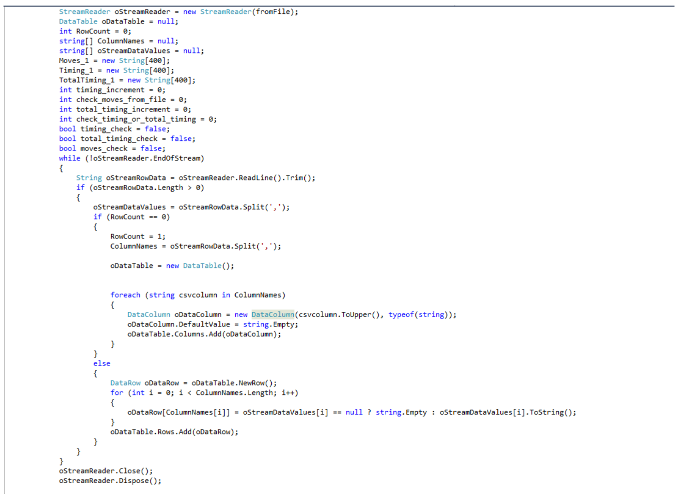


Figure 9:Unilateral Test.

A. *Reading from the file*

To read the from the file I use the StreamReader which read the each line from the file and store it in the DataTable in the form of rows and columns as stored in the file.



```
StreamReader oStreamReader = new StreamReader(fromFile);
DataTable oDataTable = null;
int RowCount = 0;
string[] ColumnNames = null;
string[] oStreamDataValues = null;
Moves_1 = new String[400];
Timing_1 = new String[400];
TotalTiming_1 = new String[400];
int timing_increment = 0;
int check_moves_from_file = 0;
int total_timing_increment = 0;
int check_timing_or_total_timing = 0;
bool timing_check = false;
bool total_timing_check = false;
bool moves_check = false;
while (!oStreamReader.EndOfStream)
{
    String oStreamRowData = oStreamReader.ReadLine().Trim();
    if (oStreamRowData.Length > 0)
    {
        oStreamDataValues = oStreamRowData.Split(',');
        if (RowCount == 0)
        {
            RowCount = 1;
            ColumnNames = oStreamRowData.Split(',');

            oDataTable = new DataTable();


            foreach (string csvcolumn in ColumnNames)
            {
                DataColumn oDataColumn = new DataColumn(csvcolumn.ToUpper(), typeof(string));
                oDataColumn.DefaultValue = string.Empty;
                oDataTable.Columns.Add(oDataColumn);
            }
        }
        else
        {
            DataRow oDataRow = oDataTable.NewRow();
            for (int i = 0; i < ColumnNames.Length; i++)
            {
                oDataRow[ColumnNames[i]] = oStreamDataValues[i] == null ? string.Empty : oStreamDataValues[i].ToString();
            }
            oDataTable.Rows.Add(oDataRow);
        }
    }
}
oStreamReader.Close();
oStreamReader.Dispose();
```

Code Segment 1: StreamReader

In the above piece of code I am using the StreamReader to read till the end of the file and if the row count is 0 then these are named as columns and stored in DataColumn's and when row count is greater than 0 then it is stored in the form of DataRow. Once the StreamReader is done executing it is closed and disposed.

The values stored in these temporary DataTable are used to perform the test and is extracted and used where ever it is required. These values are extracted from the DataRow's and DataColumn's which are stored using StreamReader.

Also I couldn't use unity 3d for SteamReader because it didn't allow me load in external libraries so I had to use visual studio to create a DLL file [10].As all the data is stored in the CSV [11] file format so I had to import the data using file reader so I had to use the above method in the visual studio and make it compatible to the dll files used by unity3d.

B. *Built using unity3d*

The project is built in unity and on the windows version and the C# scripting to interact with the objects in the game. The game engine is quite simple and is user friendly everything is visible in the same screen and can perform all the operation staying in the same screen without moving to other.

1) *Game Engine*

The game engine is quite simple and is user friendly everything is visible in the same screen and can perform all the operation staying in the same screen without moving to other. The Sphere collider is used as to check the collision.

2) *Model Loading*

Model loading in unity is simple as you can just drag and drop the objects and use them and perform operation on them.

3) *Text Mesh*

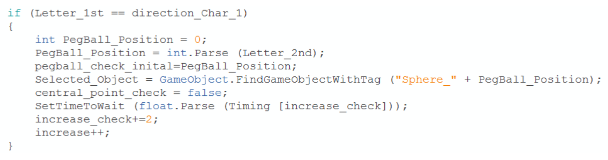
Test meshes are used for displaying the values on the screen and are changed depending on the test as it progresses.

4) *Lighting, Shadows and Line Renderer*

Basic lighting and shadows are used in the project. The Line Render is used for joining the points and to create a graph.

C. *Peg ball Movements*

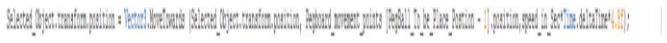
The Peg ball movements are extracted from the values after reading from the file. Then the values are got in the form of 'LX' and 'RX', The first letter indicate from which side to which side is moved and the second letter indicate the values from 1-9 which shows the peg ball number to the peg to be place position.



```
if (Letter_1st == direction_Char_1)
{
    int PegBall_Position = 0;
    PegBall_Position = int.Parse (Letter_2nd);
    pegball_check_inital=PegBall_Position;
    Selected_Object = GameObject.FindGameObjectWithTag ("Sphere_" + PegBall_Position);
    central_point_check = false;
    SetTimeToWait (float.Parse (Timing [increase_check]));
    increase_check+=2;
    increase++;
}
```

Code Segment 2: Pegball Movements.

In the above piece of code the first letter of the first move is checked that is it ‘L’ or ‘R’ and depending on that its told that the pegs are moved from left or right then the second letter is taken as the peg ball in the above code peg ball position is the number on the pegboard which the peg ball is named after. After the peg ball position or the peg ball is found the respective peg ball is picked.



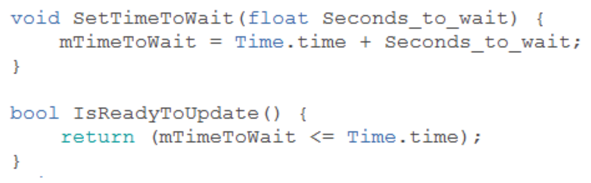
```
Selected_Object.transform.position = Vector3.MoveTowards (Selected_Object.transform.position, Pegboard_movement_points [PegBall_To_be_Place_Postion - 1].position,speed_in_Sec*Time.deltaTime*4.0f);
```

Code Segment 3: Move selected pegball.

After selecting the peg ball the selected peg ball is moved to the specific position using the above piece of code is used for moving the ball.

D. Peg ball Timing

The Peg ball timing is one of the important part which I had to pay a lot of attention to so that the pegs had to be moved to that particular position in a fixed amount of time and this fixed time can be found in the file and when the file is read the values are taken and read so that it will be moved in that amount of time and also wait till the user come and grabs the ball back.



```
void SetTimeToWait(float Seconds_to_wait) {
    mTimeToWait = Time.time + Seconds_to_wait;
}

bool IsReadyToUpdate() {
    return (mTimeToWait <= Time.time);
}
```

Code Segment 4: Wait timing

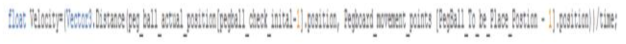
In the above piece of code the first function is used in the figure X so the time it has to wait till the patient picks the peg in the actual pegboard test. In the first function we set the time it has to wait and then in the second function is called in every update and check it has set and then waits for it and then it performs the test.

E. Calculating the Velocity

Velocity of the pegs to be moved has to be calculated because you cannot just move the pegs in some random speed it has to match with the time taken by the patient to perform the test so the velocity had to be calculated so I used a formula

Velocity = Distance/Time

Before we calculate the velocity I need to calculate the distance between the peg ball actual position to peg ball to be placed position so I used the below piece of code which helped me to calculate the distance.



```
float Velocity=(Vector3.Distance(peg_ball_actual_position[pegball_check_inital-1].position, Pegboard_movement_points [PegBall_To_be_Place_Postion - 1].position))/time;
```

Code Segment 5: Calculate the Velocity.

After calculating the distance it is divided by the time to get the velocity and the time is the time taken by the patient in the actual test this time is recorded in the test file which we read in the earlier stages of the project.

Also this is one of the key bits in our project where this velocity is used for check if the patient has performed in a higher speed rate or at a slower also depending on this velocity we can also tell if the patient is getting better or no with the help of this velocity.

F. GUI

The GUI in the previous version was a bit confusing and lot of cumbersome to use and understand so I had to make an efficient user friendly simulator so that it would give a proper demonstration of the actual simulation this was challenging task and taking everything into consideration this was I implemented and user friendly and easy to use simulator.

The GUI in the current is made user friendly and each and every elements can be accessed and used in the same screen and it is very easy to use. To achieve that I had to use 3 camera’s one to project the pegboard and one camera to display the graph and other to display the file to be selected and type of test and also the other elements.

Also in order to do all those I need to use static variable which doesn’t change even if the scene is reloaded and also I had to put a check point which indicates that the simulation has started and shouldn’t start from the beginning so to do that I had to save all these data in a text file and access it again once it is loaded back again from that check point.

Different view modes are also been implement so that the user doesn’t miss any part of the simulation and can look into every part of detail in test as it conducted in the actual test and come up with a proper analysis. Also there are Pause/Play, Fast Forward, Stop and Rotation is been implement so that the user can use these thing to pause the simulation and take down any results and again play it from the current pause position and also can fast forward the test and also Stop the Simulation in any point of time and can run the next test and the last rotation which rotates the pegboard where the user can get 360 degree view of the pegboard when the simulation is being performed so everything is happening in the runtime. You don’t have to stop the simulation or re-run the simulation in any particular point of time. Everything is done in real time which makes its outstanding with other simulators which are available in market.

To make the user even more easily the graphs are also plotted against the velocity-time graph so that the user can get clear picture and idea of how fast the patient is actually moving the pegs in the actual test. Also there are two types of graph is drawn one is a static graph which is drawn once calculating prior to the simulation and then drawn and the other one is the dynamic graph which is implemented in the real time as the simulation is performed which makes the user even more easy to look at the result.

G. Custom Graphs

In unity 3d there is no option to create the graph so I had to create the graph manually which is done by placing the quad and matching the texture to it so that it looks like it looks like the actual graph plotter. Also I had to put a set of points which are space equally and then raise them depending on the value exactly to the value obtained from the result. Then these points had to be connected using a LineRender[12] so that it looks like the graph is actually getting plotted depending on the result this was a challenging task so that I had to scale it and move it in a specified measurement and move or else the values wouldn't match.

In my project I have implanted two types of graphs

- Static
- Dynamic

In the Static graph the values are calculated prior to the simulation and then plotted before the simulation could start. The problem with this was to calculating the results before the simulation so I had to do the expensive calculation and alter a few things in the simulation so that it could create graphs in that way. Figure 10 is a graph which is created static before even the test is actually performed.

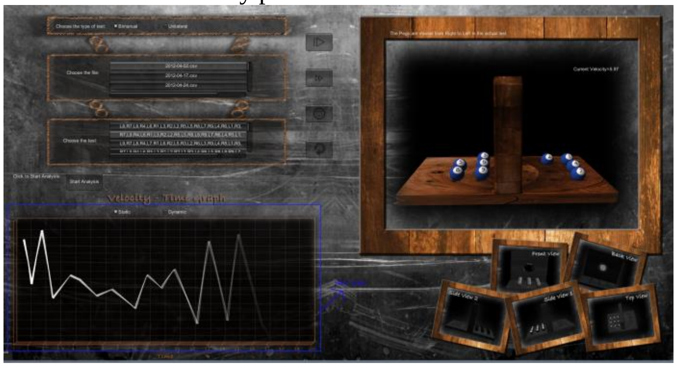


Figure 10:Static Graph

In the Dynamic graph the values are calculated as the test is progressed and as the pegs are moved. To create a dynamic graph wasn't that difficult because the test was performed as the peg moved so I just needed to calculate the velocity and just use them to plot the values. The Figure 11 is created as the pegs are moved.

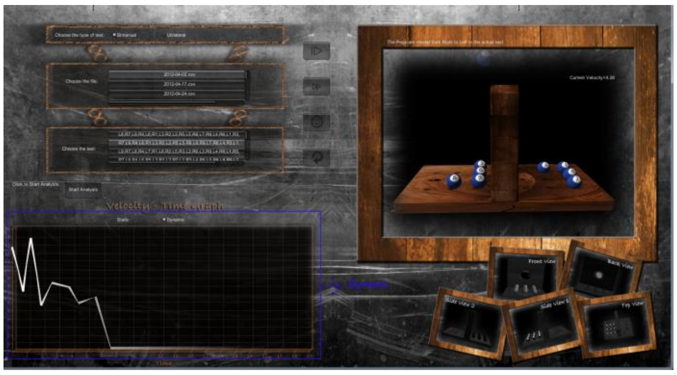


Figure 11:Dynamic Graph

The white lines used in the above graph is the line renderer which is used join these points together and looks like they are actually getting plotted.

H. Challenges

The challenges faced during the creation of this Virtual Simulator are

- Precision timing is one of the most important bit I need to concentrate when picking up the peg ball and placing the peg ball at the right timing so I had to take various measures that u don't pick the ball at the wrong time and place it in a different time so I had to various conditions and place different check points in the simulator to check if its working fine.
- Custom graphs creation had to precise and exactly match the values as they were no graph maker in unity 3d so I had to manually create my own graphs with help of a quad which severed as a background for th graph and I placed equally spaced points and arrange them in the right order so that the graphs looks alike and the values match with the result. LineRender was used to connect these set of points together so that its looks like a graph and as the test is conducted in the simulator it plots the values in the graph.
- Reading data from CSV file format these data format isn't supported by unity so I had to create a similar looking DLL file which can be read and used by the unity and had to read data from these files.
- Generating a curve path needed to thought about because various waypoints needed to be set up and the pegs had to be set so that they follow the same route when we concentrate about these waypoints I had to recalculate the velocities and had to be moved along the path.
- Reading the values from these file because I was given a huge chuck of data not in a right format so I had to arrange these data first and then take the values.
- Creating it to be user-friendly and having all the options in one particular screen had to be worked out very careful and needed to careful placed so that the

user can use the simulator without any user manual or notes.

I. Benefits

The Benefits of the Pegboard simulator are

- The basic tool for the analysis has been created and various other analysis can be performed using this tool and also it can be upgraded to perform different other analysis.
- Now you don't have look in the data or manually check the data and calculate the values for it you can just load in the file into the particular directory and then sit base watch it recreating the simulation and generating the results for you.
- If you need to perform the test again with the same result on the same patient it's not possible in the real time but using this simulator we can create any data any number of times just by passing in the file and analyze what went wrong with the patient and can diagnose it well.
- We can find the patient who finish the test faster and who had the most dexterity just by visually analyzing the graph.
- Different View modes and Pause/Play, Fast Forward, Stop and also Rotation has been implemented so that the user can look into any view and can analyze the data.
- By virtually simulating the test we reduced the stress on the patients to do the same test again and again.
- Static and Dynamic maps are drawn so that the user can select whichever he feels appropriate in conducting the test.
- The current velocity is also depicted so that can find the velocity at that particular moment of time.

V. FUTURE IMPLEMENTATION

The basic tools for analysing the data has been created so it can be used as a benchmark to create new analysis based on this and can also be upgraded to by adding in new data and perform different analysis on this data.

The current version is running on the windows version it can be released on multiplatform.

The doctors can lend the pegboard to the patients and the patients can perform this test at their home by this doctors can have the upgraded version of this to so that it can virtually simulate the test as the patient is performed and the analysis can be made with no stress and not relying on anyone to pick them up to the hospital and drop them back home which is similar to the related work as discussed above in the paper.

It can also be used in the stereoscopic 3D environment where the patient can perform this test using the stereoscopic devices available in market which is also similar to the related work as discussed above.

VI. RESULTS

The result to this simulation is provided by the velocity-time graph which depicts the values in the form of graph.

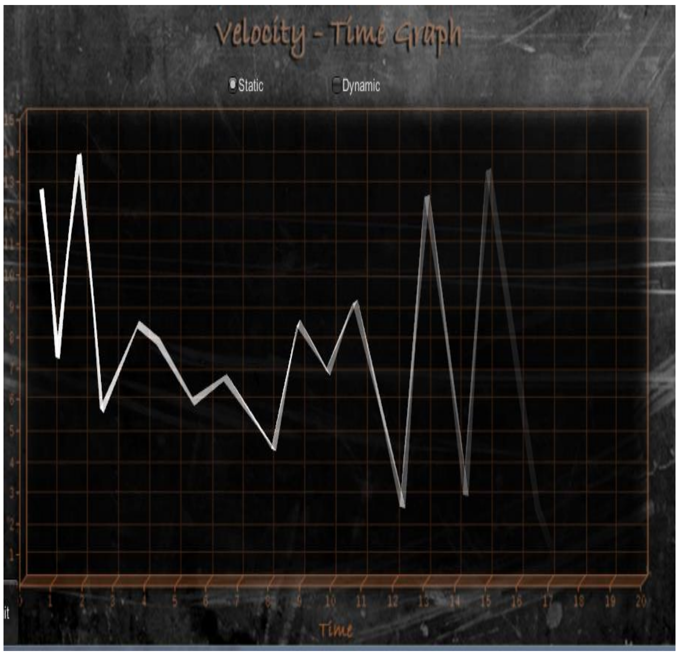


Figure 12:Non-Action games playing patient V-T graph

As you can see the Figure 12 the picked and moved with a velocity of 12.9cm/sec and then it drops while the velocity drops and then when he picks up the second ball he is moved at a higher velocity than before and then he tries to maintain a slightly stable speed and the drops. when these test are compared with action games playing patient vs non action games playing patients the graphs can vary and also the time taken to complete the test also will be lesser.

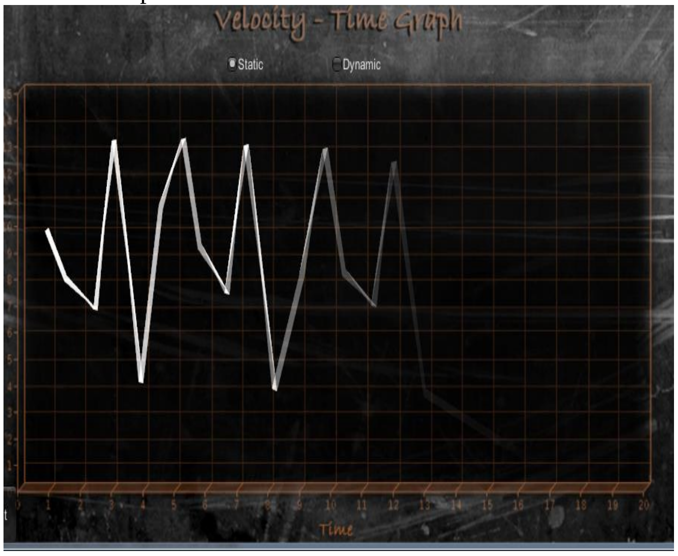


Figure 13:Action games playing patient V-T graph

As we can look at the Figure 13 he has as stable velocity and also has performed the test fast which depicts that the action games playing patients are performing the test faster than the non-action games playing patients also they have a stable which do not drop down and won't have much variations as the graph which the non-action games playing patient has performed.

When we compare these two graphs we tell that action games playing patient performed the test faster with stable velocity thought the test and Non-action games playing patient performed test slower and their velocity was unstable thought the test

.

Further analysis has to be done to find the strategy the action games playing patient took to perform the test fast. But with this help of this basic tool we can tell they had good dexterity so that the velocity increased in a considerable manner so they were able to finish the test faster compared to Non action games playing patient as they had bad dexterity there velocity never remained stable they were always fluctuating so the velocity decreased as the velocity decreased they couldn't perform the test faster.

VII. Conclusion

The paper presented here describes the simulation of the pegboard and the results of the Action games playing patients vs the Non action games playing patients are compared with the help of the virtual simulator and with the graphs we can tell the action games playing patients were good with fingers and hands so they had a stable velocity because they played action games because in action games you have to be quick in order to successfully completed the mission or the game so this has enabled these action games playing patient to perform the test faster over the Non-action games playing patient as they are not used to playing action as they are not good with hand movements and finger they lack with unstable velocity and perform the test slower.
Although this alone is not enough to derive the strategy the action games playing patient used but if further analysis is done with the basic tool provided we can find out the strategy they have used and if we know what kind of strategy they have used we can use this strategy to minimize the hemiplegia and a save lot of patients that are suffering from it.

References

[1] Basu AP, Pearse J, Kirkpatrick EV, Ling ITC, Tan GSL, Eyre JA(2011,October). Raising awareness of the myth of the "unaffected hand" in childhood hemiplegia. Pediatric Research: 52nd Annual Meeting of the European Society for Paediatric Research. http://www.nature.com/pr/journal/v70/n5s/pdf/pr2011377a.pdf

[2] Hemiplegia,"what is hemiplegia",viewed on June 2014 http://www.hemihelp.org.uk/hemiplegia/what_is_hemiplegia

[3] Nine Hole Peg Test,"purpose of the measure",viewed on June 2014 ,http://strokengine.ca/assess/module_nhpt_indepth-en.html

[4] Dexterity,"Dexterity",viewed on June 2014 http://www.oxforddictionaries.com/definition/english/dexterity

[5] Game Genre,"Action Games",viewed on June 2014 http://www.allgame.com/genres.php

[6] Grigore Burdea, Viorel Popescu, Vincent Hentz, and Kerri Colbert(2000,September). Virtual Reality-Based Orthopedic Telerehabilitation.,http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=867886

[7] Patrick A. Kenney, Matthew F. Wszolek, Justin J. Gould, John A. Libertino, Alireza Moinzadeh(2009,June). Face, Content, and Construct Validity of dV-Trainer, a Novel Virtual Reality Simulator for Robotic Surgery.http://www.sciencedirect.com/science/article/pii/S0090429509000351

[8] Unity,"Scripting::Time"http://docs.unity3d.com/ScriptReference/Time.html

[9] Loading Data,"Read from Excel 2003",viewed on June 2014 http://wiki.unity3d.com/index.php?title=Load_Data_from_Excel_2003

[10] DLL file,"What is a dll file??",viewed on June 2014http://support.microsoft.com/kb/815065

[11] CSV file format, "The CSV File Format" viewed on August 2014,http://creativyst.com/Doc/Articles/CSV/CSV01.htm

[12] Line Renderer,"Line Rendere",viewed on August 2014,http://docs.unity3d.com/ScriptReference/LineRenderer.html