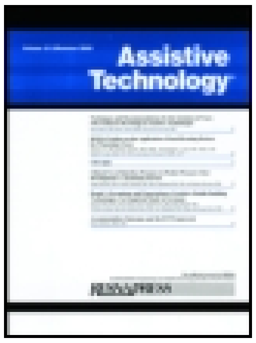


Assistive Technology
The Official Journal of RESNA

ISSN: 1040-0435 (Print) 1949-3614 (Online) Journal homepage: http://www.tandfonline.com/loi/uaty20

MoKey: A versatile exergame creator for everyday usage

Martina Eckert, Marcos López, Carlos Lázaro & Juan Meneses

To cite this article: Martina Eckert, Marcos López, Carlos Lázaro & Juan Meneses (2017): MoKey: A versatile exergame creator for everyday usage, Assistive Technology, DOI: 10.1080/10400435.2017.1406016

To link to this article: https://doi.org/10.1080/10400435.2017.1406016

Accepted author version posted online: 27 Nov 2017.
Published online: 22 Dec 2017.

ASSISTIVE TECHNOLOGY
https://doi.org/10.1080/10400435.2017.1406016

MoKey: A versatile exergame creator for everyday usage

Martina Eckert, PhD, Marcos López, BSc, Carlos Lázaro, BSc, and Juan Meneses, PhD

Research Center on Software Technologies and Multimedia Systems for Sustainability (CITSEM), Universidad Politécnica de Madrid, Madrid, Spain

ABSTRACT

Currently, virtual applications for physical exercises are highly appreciated as rehabilitation instruments. This article presents a middleware called "MoKey" (Motion Keyboard), which converts standard off-the-shelf software into exergames (exercise games). A configurable set of gestures, captured by a motion capture camera, is translated into the key strokes required by the chosen software. The present study assesses the tool regarding usability and viability on a heterogeneous group of 11 participants, aged 5 to 51, with moderate to severe disabilities, and mostly bound to a wheelchair. In comparison with FAAST (The Flexible Action and Articulated Skeleton Toolkit), MoKey achieved better results in terms of ease of use and computational load. The viability as an exergame creator tool was proven with help of four applications (PowerPoint®, e-book reader, Skype®, and Tetris). Success rates of up to 91% have been achieved, subjective perception was rated with 4.5 points (from 0–5). The middleware provides increased motivation due to the use of favorite software and the advantage of exploiting it for exercise. Used together with communication software or online games, social inclusion can be stimulated. The therapists can employ the tool to monitor the correctness and progress of the exercises.

ARTICLE HISTORY
Accepted 6 November 2017

KEYWORDS
activities of daily living; computer access; electronic aids to daily living; neuromuscular impairment; older adults; rare degenerative diseases; universal design; usability

Introduction

According to the World Report on Disability (World Health Organization, 2011), the *Global Burden of Disease* study found that in 2004, around 12% of the population aged 15 to 59 years from high-income-countries had a moderate or severe disability. Including the elderly over 60, this rises to 18%. In spite of living in high-income, technologically-advanced countries, this group of the population still faces numerous disadvantages: On one hand, due to reduced mobility, physical activity significantly decreased. Only 25% of handicapped people participate in weekly exercises, compared to 43% of adults without physical impairments (Boslaugh & Andresen, 2006). A side-effect of reduced mobility is the risk of reduced social activity, as many elderly and disabled become tied to their homes, which increases isolation (Skjæret et al., 2016). The most efficient way to cope with reduced mobility is rehabilitation, and technology could help to enhance the possibilities for doing it at home with help of exergames (exercise games). Bonnechère, Jansen, Omelinab, and Van Sint Jan (2016) showed that video games, not especially designed for clinical purposes, are as efficient as conventional therapies or lead to even better therapeutic outcomes. They also have numerous advantages, such as preventing monotony and boredom, increasing motivation, providing direct feedback, and allowing double-task training (Bonnechère et al., 2016).

Webster and Celik (2014) found that most exergames are aimed at solving individual problems or frequent diseases particularly common in the elderly, such as arm rehabilitation for stroke patients or balance training for Parkinson. Nevertheless, there is a distinct lack of investigation into applications that could be used by everybody, with any kind of problem, necessity, or capacity. They stated that the Kinect would not be suitable to capture gross or weak movements and are generally felt to be unsuitable for severely disabled patients. They also argued that a technology initially intended for a younger and healthier audience has to be used focusing on both the physiological and psychological requirements of aging and injured users. A further and extremely important shortcoming detected by Webster and Celik (2014) is an insufficiently robust and easy-to-manipulate user interface (Webster & Celik, 2014). Skjæret and colleagues (2016) highlighted a lack of long-term enjoyment and an under-exploited potential regarding social inclusion as the typical deficiencies of exergames. They argued that specifically for older adults, appropriate games have to be tailored individually because they do not adapt easily to new environments (Skjæret et al., 2016).

These findings encouraged us to create an exercise software that could be adapted to everybody's needs and capacities. It would be attractive for young people and elderly, suitable especially for the disabled, provide easy configurability for non-experts, and give possibilities to improve social inclusion. This lead to "MoKey" (Motion Keyboard), a versatile middleware that enables the usage of commercial video games and other standard applications (off-the-shelf software) with corporal gestures (e.g., arm, hand, or trunk movements), such that any type of application could be converted into an "exer-application" (not exclusively games). The middleware works

CONTACT Martina Eckert, PhD martina.eckert@upm.es Research Center on Software Technologies and Multimedia Systems for Sustainability (CITSEM), Universidad Politécnica de Madrid, Alan Turing St., Madrid 28031, Spain.

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/UATY.

as an interface that translates human gestures, recorded by a motion capture camera (Kinect), into the keyboard events necessary to control the application. The gestures are predefined and configurable according to the user's needs, and can be assigned to any key.

Few similar proposals have been found in the literature. Kamel Boulos and colleagues (2011) provided an early attempt to use software, not initially developed for Kinect, with the help of an interface (Kamel Boulos et al., Skjæret). They showed the control of Google Earth with gestures, recognized by the so-called Kinoogle GUI (graphical user interface). The gestures they selected are adequate and intuitive but not user adaptive; a handicapped user could not benefit from this system. Suma, Lange, Rizzo, Krum, and Bolas (2011) published a configurable and versatile interface called FAAST (The Flexible Action and Articulated Skeleton Toolkit; Suma et al., 2011). The complete toolkit was published in Suma and colleagues (2013) and Koenig, Ardanza, and Cortes (2014). It is a highly flexible middleware for integrating body control with custom virtual reality applications and video games. Human actions and input bindings are configurable at run-time, allowing the user to customize the controls and sensitivity to adjust for individual body types and preferences. Sevick and colleagues (2016) applied the FAAST toolkit and presented a study with four children with cerebral palsy (CP), about using a free Internet videogame in conjunction with the Kinect motion sensor. Results indicated the feasibility of delivering a videogame motor training intervention and also indicated a high level of motivation among the small number of participants (Sevick et al., 2016). Pool and colleagues (2016) presented an application that allows for navigation through a virtual environment. It was created to test viable movements for people affected by stroke or CP (Pool et al., 2016).

The only versatile middleware found is FAAST, but we consider the configuration procedure to be difficult and tedious, and aimed at presenting similar or better performance with MoKey. The following main benefits are addressed in this work:

- Give access to certain applications for people who have difficulties using a regular keyboard;
- Convert favorite and frequently used standard applications into exercise software to increase motivation; and
- Promote social inclusion by providing easy usage of communication software when keyboard usage is difficult.

A preliminary version of our approach was already presented in Eckert et al. (2015). In this article, we want to present the technological details and to show the results of the first user tests performed on a group of 11 volunteers with different motor function disabilities. Eight of them presented an elevated dependency in daily life activities (e.g., need assistance on the toilet, to get dressed, fed, or cleaned) and showed strong limitations in mobility (electrical wheelchair dependent) and limb movements. Those eight were considered as being severely disabled. Our results should prove the viability of the tool for the performance of physical exercises, potentially also for rehabilitation, and applicability to very different user groups. We will not show the effectiveness of the exercises for specific diseases.

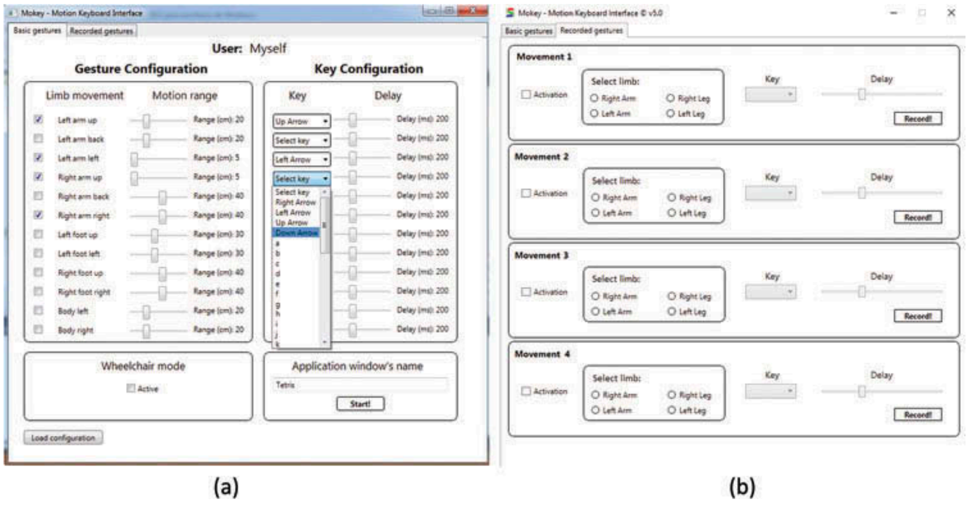
Methods

System description

Unlike FAAST, the present system does not allow users to configure any possible movement as this would be too complicated and probably not fully explored in everyday use. Instead, the tool offers the possibility to select in between 12 simple movements that correspond to the capabilities of each patient and to adapt them easily to the user's needs. The movements were selected according to the rehabilitative necessities of the target group, which includes the elderly as well as a cross-range of subjects with severe motor disabilities. Unlike most studies, those suffering from rare degenerative diseases (RDD) have been specifically taken into account (The group of 11 volunteers included 7 with RDD). RDD sufferers have diminished lower limb movements, and therefore require frequent exertion to prevent the complete loss of use, deformations, and other related complications. The principle aim of rehabilitation is to delay as long as possible the loss of basic functionalities, such as fine and gross motor skills, trunk, and head control. Therefore, lateral trunk movements were included to exercise the strength of the back, arm gestures to train shoulder flexion, and frontal and lateral feet elevations were added to movement options to provoke hip flexions. Additionally, the heuristics proposed in Jiang, Duerstock, and Wachs (2012) have been taken into account. They proposed gesture patterns specific to patients with upper extremity impairments, such as: "Select gestures that do not strain the muscles"; "Select gestures that avoid outer positions"; and "Select dynamic gestures instead of static gestures," etc (p. 535) Any of the gestures can be assigned to one of the keys available on the computer keyboard by marking the corresponding checkbox and selecting a key from the drop-down list. Figure 1a shows the main window, which provides the basic movements. In case none of the 12 gestures fulfills the user's needs, four customized arm or leg movements could be recorded (Figure 1b). By recording, a specific point is captured in the 3D space that has to be reached during the exercise; whereas with the predefined gestures, there is only one threshold in one direction (sideways, for/back, or up/down) to be overcome to execute the configured key stroke. As an example, this mode could be applied if a person should rise the arm up to a certain point while reaching a certain distance from the body at the same time.

The 12 basic gestures are: three movements for each arm (abduction, raise, and pulling backward), two movements for each foot (abduction and raising), as well as the inclination of the trunk to each side. For any of these gestures, two parameters can be adjusted, as described in the following.

The "motion range" is the amplitude needed to trigger the corresponding keyboard event: The larger the value, the wider the gesture has to be, to be detected. The range can be adjusted from 1 cm (0.4 in) to a maximum of 70 cm (27.6 in) with the help of a slider. Figure 1a shows the default settings of the values.


Mokey - Motion Keyboard Interface

Basic gestures | Recorded gestures

User: Myself

Gesture Configuration | Key Configuration

	Limb movement	Motion range	Key	Delay
✓	Left arm up	Range (cm): 20	Up Arrow	Delay (ms): 200
	Left arm back	Range (cm): 20	Select key	Delay (ms): 200
✓	Left arm left	Range (cm): 5	Left Arrow	Delay (ms): 200
✓	Right arm up	Range (cm): 5	Select key	Delay (ms): 200
	Right arm back	Range (cm): 40		Delay (ms): 200
✓	Right arm right	Range (cm): 40		Delay (ms): 200
	Left foot up	Range (cm): 30		Delay (ms): 200
	Left foot left	Range (cm): 30		Delay (ms): 200
	Right foot up	Range (cm): 40		Delay (ms): 200
	Right foot right	Range (cm): 40		Delay (ms): 200
	Body left	Range (cm): 20		Delay (ms): 200
	Body right	Range (cm): 20		Delay (ms): 200

Figure 1. (a) Main window for motion configuration with 12 pre-defined gestures, (b) Window for recording 4 additional gestures

The "delay" defines the time lapse needed to detect two individual consecutive keystrokes. When longer gestures should simulate a continuous keystroke, a very short delay should be selected. The adjustment provides values between 1 and 600 ms, and allows for the control of both: user movements and software requirements (e.g., if a user cannot return a limb quickly enough to its normal position, the delay can be augmented to prevent the pose being recognized as multiple keystrokes). If a software requires a repeated stroke, at a certain speed, or continuous pressure of one key to perform a particular action, and the user is not able to do this, the delay can be augmented to ensure the smooth performance of the application.

The configuration window as shown in Figure 1a also includes a special "wheelchair mode," where the detection of the foot movements is resolved in a different way than for standing people to avoid complications with the chair.

To use MoKey, the user (or an assistant, if necessary) first opens the software application, and afterward starts MoKey. After the optional load of a configuration file, the main window opens. On the lower right side of the screen, the name of the application is entered, and the "Start!" button is pressed. From that moment, MoKey will translate detected movements to keystrokes and send them to the application so that it will work with gestures for the defined keys.

Implementation details

MoKey was developed in C# for Windows 8 for the Kinect v1 (Xbox 360) with the use of the Software Development Kit v1.8 (SDK) provided by Microsoft. As shown in the diagram of the architecture (Figure 2), the Kinect sensor obtains an RGB (red green blue) and a depth image with the help of an infrared emitter and sensor. Those images are used to calculate 20 joints and 121 facial feature points from a maximum of two user skeletons posed in a distance of 1.2 to 3.5 m (3.9 to 11.5 ft). With the help of these skeletons, the user's movements can be captured and evaluated. A very detailed description of the Kinect and the possibilities of the provided API

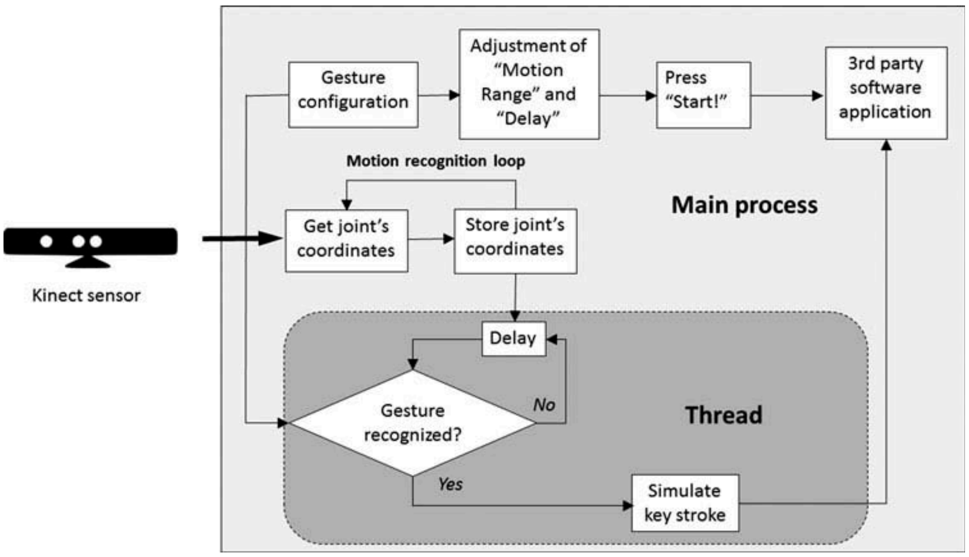


Figure 2. Flow chart of the motion recognition process realised in MoKey.

(application programming interface) can be found in Borenstein (2012).

In this article, the default skeleton provided by the API was used to obtain the coordinates of the joints. As most applications require multiple keyboard events at a time, the recognition of each gesture was implemented in an individual thread. Thread programming also allows optimal management of CPU (central processing unit) usage by applying keyboard event delays to restrict the number of operations per second. Due to this implementation, our approach achieves a better performance in speed than FAAST, as shown in the results section.

Each joint is defined by a set of four values, where three represent the position in space in Cartesian coordinates (*x*, *y*, *z*) with the origin placed in the sensor, *z* pointing to the user, *y* up, and *x* to the right from the user's point of view; the fourth value indicates the self-rotation of a joint. For the 12 predefined gestures, MoKey calculates the distance between one joint of a limb (e.g., hand for arm movements) and the spine in one direction. The coordinate depends on the gesture: *x* for horizontal movements, *y* for up/down, and *z* for forward/backward. If the value obtained is at least as large as the amplitude adjusted in motion range, a keyboard event is sent to the software application. In case an additional customized movement is recorded, MoKey checks that the corresponding limb reaches the limits in all 3D coordinates at the same time such that the gesture has to be more precise. When the wheelchair mode is activated, foot movements are calculated as the difference between foot and knee instead of spine, as that joint cannot give reliable results due to the seated position. This wheelchair mode is different to the seated mode offered by the Kinect API, which does not take into account the lower limbs at all.

Data collection

Eleven participants (8 male), with different disabilities (ages 5–51 years) were invited to participate in the tests (Table 1). All participants were recruited in Madrid, Spain, at the Neurological Muscular Diseases Association[1] and the Sports Integration Foundation.[2] Approval was obtained from the Review Board at Universidad Polit&cnica de Madrid. Table 1 also gives an overview of the grade of mobility and physical impairments of each subject. Values were obtained from the participants, who were asked to rate their perception of mobility for each limb according to a 6-level Likert Scale, ranging from none (0) to wide and strong (5). To get an idea of the severity of their restrictions, we calculated a degree of limitation based on that scale, as follows:

$$\text{Degree of limitation} = \left(1 - \frac{\sum s}{20}\right) 100\% \qquad (1)$$

where s = score given to a limb; 20 is the maximum possible scoring. The resulting values are given in the rightmost column of Table 1. All values equal or greater than 40% are highlighted, as those volunteers were classified by the authors as severely affected. This method is custom made and clinically not verified. It was created due to the lack of a grading system valid for different types of disabilities, which would also be easy to assess without having a medical background. Nevertheless, our system is very similar to the widely used Daniels index (Daniels & Worthingham, 1986) used to measure muscle strength on a scale of 0 to 5.

To find suitable third party software for testing the interface, 10 frequently used programs of different genres were checked for applicability with MoKey. The main requirement was the possibility to be controlled exclusively with key shortcuts (no mouse); secondly, it should make sense to be used with only few keys (two to four) to not exaggerate the number of movements to be made. The pre-selected programs were one web browser, one anti-virus software, one communication software, two office tools, one e-book reader, and four games as shown in Table 2. The results of those tests are presented in the results section. Four of the 10 applications were selected for the user tests: PowerPoint®, an e-book reader (Icecream®), and Skype® for testing two movements; and the Tetris game for testing four gestures. The reason was to find well-known software to save time explaining, and to limit the number of movements to find enough that are convenient for the users. Tetris was selected for testing the viability of more than two movements; unfortunately, there was no other similar software for comparison.

Table 1. Demographical data of the volunteers (target group).

Subject ID	Gender	Age (years)	Disease	Mobility	Arms	Hands	Trunk control	Legs/ feet	Degree of lim.
1*	Male	5	SMA-2	Wheelchair	3	4	2	1	**50%**
2	Male	13	SMA-2	Wheelchair	1	3	2	0–1	**67.5%**
3	Male	10	CP	Wheelchair	3–4	3	3	2	**42.5%**
4	Male	12	CP	Wheelchair	4	4	4	4	20%
5	Male	12	HYP	Wheelchair	2	2	2	4	**50%**
6	Male	13	BMD	Free walking, plays seated	5	3	4	3–4	22.5%
7	Male	15	DMD	Wheelchair	1	5	4	0	**50%**
8	Male	16	DMD	Walking some meters, scooter for long distances	3	4	2	1–2	**47.5%**
9	Female	43	FSH	Wheelchair	1	5	1	0	**65%**
10	Female	49	FSH	Free walking, plays standing	3–4	5	4	4–5	15%
11	Female	51	PPS	Wheelchair and walking with crutches, plays seated	4	4	3	1	**40%**

Notes. The scores range from no movement (0) to wide and strong movement (5). The last column gives an idea of the degree of limitation of each person. SMA-2: SMA type 2; HYP: Hypertonia. Bold indicates severe disability according to the author's definition by eq. (1). *Subject 1 was excluded from the tests, because the Kinect could not be calibrated due to confusion with the wheelchair.

[1]ASEM = Asociación de Enfermedades Neuromusculares Madrid: http://www.asem-madrid.org.
[2]Fundación TAMBIEN: http://www.tambien.org.

Table 2. Applications tested with MoKey.

Application	Minimum number of keys required	Possibilities and limitations
Web browser (Google Chrome)	4 (F6, left–right arrows, enter)	Allows navigation through previously stored markers, no introduction of new links.
Anti-virus (Avast)	3 (enter, left–right arrows)	Basic navigation through the pages, no change of configuration as this needs the introduction of text.
Skype® (Microsoft)	2 (Alt-PgUp, Alt-PgDn)	Allows desired keys to be assigned to each function; allows answering and hanging up of voice or video calls, but not sending text messages.
Word (Microsoft)	4 (Ctrl-O, up/down arrows, enter)	Allows opening a new document and browsing through pages, but no editing.
PowerPoint® (Microsoft)	2 (PgUp, PgDn)	Allows forward and backward flipping of the pages; the presentation mode has to be set by an assistant.
e-book reader (Icecream®)	2 (PgUp, PgDn)	Allows forward and backward flipping of the pages; the book has to be loaded by an assistant.
Tetris (Crystal Office)	4 (left–right, up–down arrows)	An additional key (F2) would be needed to enable the user to restart the game.
Minecraft (Mojang AB)	4–6 different keys	Allows assigning each functionality to the desired key. Four are sufficient to move forward and backward and to perform two actions. Six would enhance the possibility of displacement. No possibility to control camera rotation by key-shortcuts.
Pinball (Microsoft)	3 (free to select)	This game was *not playable*, as it needs continuous keystrokes to hold the flippers. This is not possible to realize with the current version.
League of Legends (Riot games)	N/A	This video game needs a very high amount of computational power, which makes it inadequate for usage with MoKey. It also requires a mouse for character movement.

During the test sessions, the interface, the possible gestures, and the software to be used were firstly explained to the volunteers. Then, they were asked to choose their preferred gestures with the opportunity to test them. In this moment, the motion range and delay were configured and the possibility was offered to record a customized gesture. During the play, the positions of the joints tracked for each movement were automatically recorded at adjustable time intervals and stored into Excel files. The data was later analyzed regarding endurance time (summing up the playing times of all attempts on each application), the frequency of movements for each limb (dividing the sum of movements by the total time played each application), and the success rate (number of successful attempts divided by total number of attempts multiplied by 100). The analysis of the data is presented in the results section. During the play, we also observed if the control worked well, and re-adjusted the default parameters if required. Some applications were started multiple times—in between different applications, the users had a short rest during the explanations. In the case of fatigue, the tests were paused or stopped. On the average, the volunteers were occupied during approximately 30 minutes with a pure playing time of around 18 minutes. Afterward, all participants were asked to complete a questionnaire about their impressions.

Results

Two different aspects were tested to check the viability of the proposed middleware: technically, the computational performance was analyzed in comparison with FAAST and the applicability for different types of programs was checked. From the human point of view, a usability test was made regarding the ease of configuration as well as the viability for disabled people.

Technical evaluation

Computational performance

All tests were driven on an Intel® Core i7-4790/3.6 GHz computer. The CPU load and memory usage were analyzed and compared with FAAST. The CPU usage changes significantly with the activity in front of the Kinect. When MoKey is running on the computer but the Kinect is not detecting the user, the CPU usage is low, about 3.3%. In the moment a user is detected, this value increases up to 10%, independently from the number of configured movements. The CPU load of FAAST is more than twice as high as MoKey, and above 85% when there is a user in front of the camera. In a case that the main application needs much CPU power, the system would produce an overload. The average RAM (random access memory) usage is similar in both middleware solutions (about 70 MB). As current computers are generally equipped with 8 GB or more, the tools would use less than 1% of their capacity.

Applicability to different types of software

As formerly mentioned, 10 frequently used programs of different genres that offer the possibility to be controlled with few key shortcuts were checked for applicability with MoKey. Table 2 lists the key-shortcuts we tried out and provides information about the possibilities and limitations inherent in the use with MoKey. Applicability means that at least one basic task could be performed with less than four gestures, as a higher number would require a complicated corporal coordination. Unexpectedly, two of the games could not be used with MoKey. One was “Pinball,” because it needs a continuous key stroke, which is not possible to simulate with MoKey. “League of Legends” needs a very high amount of computational power that effects the smooth working of MoKey and made no sense without using the mouse. All other applications could be controlled without problems.

Usability tests

Configuration

As one of the requirements is to provide simple and robust interfaces (Webster & Celik, 2014), the user friendliness of the proposed work has also been compared to FAAST as a reference software. Therefore, an expert (a person who knew both programs very well) and a non-expert (who saw both programs for the first time) were requested to set up both tools with two movements: “Right arm to the right” to press the A-key and “Left arm to the left” to press the B-key. The expert achieved the configuration of MoKey in 22 seconds, while the non-expert needed 72 seconds. For FAAST, the expert needed exactly two minutes (six times as much), and the non-expert gave up after three minutes without achieving the desired configuration. The time to execute the applications after the user pressed “Start!” is also different. While MoKey shows a latency time below one second, FAAST needs more than 20 seconds to start working properly (Eckert et al., 2015).

User tests

Issues. All but one (subject 1) of the invited volunteers (Table 1) could actually use the tool; as for subject 1, a small five-year-old child, the application did not work properly. The Kinect seemed to confuse the wheels of his wheelchair with his arms, so no arm movements could be configured. Due to instability, it was also not possible to sit him in a normal chair.

- Data for lateral trunk movements was only recorded for subject 7 after detecting a bug during the tests. Therefore, we cannot present results about the viability of this movement. However, being a frequently selected one, we considered it still worthwhile to mention.
- Besides the missing data for trunk movements, the results obtained for five subjects are incomplete because of some corrupted data sets.
- The computer we had available for the tests (Intel® Core i7-4790/3.6 GHz) did not allow a higher data recording rate than 2 samples/s. To assure that this was not too low to capture very quick movements properly, a special test was performed by a person without any physical limitation who was asked to do the movements as quick as possible. It was found that only the quickest possible movements, which last around 1.5 seconds, were not reliably detected. Instead, medium-speed movements take around 2.5 seconds, and the system detected them perfectly.

Neglecting subject 1, the remaining 10 participants covered seven types of diseases and an age range from 10 to 51 years. Table 3 shows the combinations of gestures selected by the volunteers, those for two key strokes on the left side and those for four key strokes on the right side. For 2-key applications, the predefined parameters of motion range and delay were valid for all users. In case of Tetris, subjects 7 (Duchenne muscular dystrophy; DMD), 9, 10 (facioscapulohumeral muscular dystrophy; FSH), and 11 (post-polio syndrome; PPS) needed slight adjustments. Most volunteers preferred to use their arms and trunk—only subject 10 (FSH) tried the feet. Nobody wanted or needed a special movement to be recorded.

We considered the following indices to test viability of the middleware as an exercise instrument: the *time* users endure with an application, the *frequency* of key strokes, and the *success rate* (number of successful attempts divided by total number of attempts multiplied by 100). Viability would be given if all three indices were high, in the sense that: the user did not give up before 5 minutes, playing times were long enough to test all applications and get sufficient measurements, the frequency of key strokes was sufficient to achieve a reasonable play in Tetris (the other applications do not require quick strokes), and the success rate was at least 2 of 3 attempts (67%) to avoid frustration.

- The playing times were obtained by summing up the durations of each trial of an application, which could be read out of the Excel files stored by MoKey (see time columns in Table 4). Times were quite different for each volunteer, on the average 10.3 ± 6.7 minutes with 2-key applications and 7.3 ± 5.1 minutes with Tetris, totally around 20 minutes playing time. Subject 9 (FSH) claimed tiredness after 20 minutes; subject 7 (DMD) was the only one who gave up (after 7 minutes).
- The average number of strokes per minute (k*n* in Table 4) is generally higher for the right arm (k1) than for the left arm (k2) in cases of the 2-key applications. As all users were right-handed, the right hand was used to turn pages or slides forward and the left hand to move backward, which was done only occasionally. In the case of Tetris, both arms were used with a similar frequency. Here, the most comfortable gestures were selected to move the bricks with the left–right arrow keys, as this is the prior task in the game. The other two movements were used to turn the bricks or move

Table 3. Combinations of movements selected by the volunteers to control the different applications.

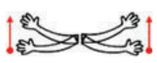
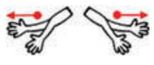
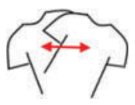
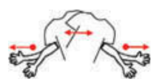
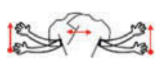
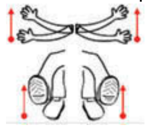
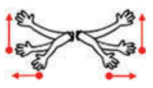
		2-key movements (PowerPoint®, e-book, Skype®)			4-key movements (Tetris)			
		Arms up	Arms sideways	Trunk sideways	Arms & trunk sideways	Arms up, trunk sideways	Arms and feet up	Arms up & sideways
ID	Disease							
2	SMA-2		✓		✓			
3	CP	✓						✓
4	CP		✓		✓			
5	Hyp			✓	✓			
6	BMD	✓			✓			
7	DMD			✓		✓		
8	DMD	✓				✓		
9	FSH	✓				✓		
10	FSH	✓					✓	
11	PPS	✓				✓		

Table 4. Time of usage and frequency of movements per minute during testing the 2-key (PowerPoint®, e-book, Skype®) and 4-key applications (Tetris).

		PPT®		e-book		Skype®				Tetris				
ID	Movements used (2 key apps)	k1	k2	k1	k2	k1	k2	t [min]	Movements used (4 key apps)	k1	k2	k3	k4	t [min]
2	Arms sideways							**17**	Arms & trunk sw.	5.2	4.0			**9**
3	Arms up	1.5	1.7	1.0	2.9	1.8	1.8	**16**	Arms up & sw.	2.6	4.7	3.6	3.1	**4**
4	Arms sideways	2.5	0.8			2.0	2.0	**7**	Arms & trunk sw.	4.9	2.1			**15**
5	Trunk sideways								Arms & trunk sw.	7.3	4.3			**17**
6	Arms up	4.0	1.9	2.0		0.8	0.8	**8**	Arms & trunk sw.	6.6	4.6			**2**
7	Arms sideways	11.8	6.4					**3**	Trunk sw., arms up	9.2	5.0	2.0	1.7	**7**
8	Arms up	4.7	3.6	4.9		0.6	1.2	**15**	Arms up, trunk sw.	4.2	6.1			**5**
9	Arms up	1.8	1.1	6.0	6.5	1.8	1.8	**19**	Arms up, trunk sw.	4.8	5.2			**4**
10	Arms up	3.6	1.3	6.7	2.2	1.3	1.9	**15**	Arms and feet up	5.2	6.9	5.9	2.1	**3**
11	Arms up	6.2	4.8	3.6		3.5	1.2	**5**	Arms up, trunk sw.	6.5	6.5			**4**
Arms up mean:		3.6	2.4	4.0	3.9	1.6	1.5	**10.3**	Arms up mean:	4.2	5.2			**7.3**
Arms up *SD:*		1.8	1.5	2.2	2.3	1.0	0.4	**6.7**	Arms up *SD:*	1.7	1.9			**5.1**
Arms sideways mean:		7.1	3.6			2.0	2.0		Arms sideways mean:	5.5	3.6			
Arms sideways *SD*:		6.6	3.9			-	-		Arms sideways *SD*:	1.5	1.0			

Notes. *kn* indicates the assigned key, grey shading marks arm-up movement, dark grey shading marks trunk movement, bold face is for times in minutes. SD = standard deviation; sw = sideways.

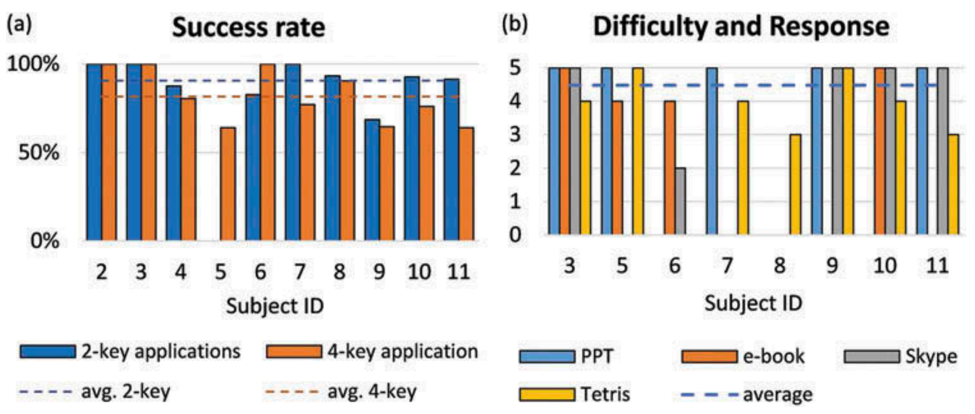


Figure 3. (a) Illustration of the success rates in case of 2-key and 4-key applications, (b) Subjective perception of difficulty averaged with the application&s response (0=difficult/bad to 5=easy/good)

them down quickly, so the game dynamics requires them less. For subject 7 (DMD), an interesting result can be observed: being very weak in arm movements, the trunk movements resulted to be adequate for him such that he achieved quite high frequencies.

- The most meaningful measure is the success rate. The chart in Figure 3a illustrates the average rates obtained by each subject for the 2-key applications (averaged) and the 4-key application. Subject 5 has no data for the 2-key applications due to the formerly mentioned failure of body movement capturing. In general, the success rate for the 2-key applications is higher than for the 4-key applications, with no score lower than 50%. The 4-key application (Tetris) generally involved a longer play and required faster movements due to the game dynamics. This induces more fatigue and incorrect movements, while the 2-key applications are slower, thus requiring fewer movements per minute as shown in Table 4. Overall, success rates are high: 91% on average for 2-key applications and 82% for Tetris. The individuals performed similarly high in both, or better for the 2-key applications. Only subject 6 (Becker muscular dystrophy; BMD) did better with Tetris, although he played only 2 minutes.

After the tests, users were asked to complete a survey to determine: (a) ease-of-use (whether they found the software easy or difficult to use), and (b) accuracy (how well the application responded to their movements). Both questions were rated from very difficult/bad response (0) to very easy/good response (5). The answers for difficulty and response are very similar, such that they were averaged for the chart—values are presented in Figure 3b. The application was perceived as best performing with the PowerPoint® presentation, which was the first software tested. The e-book reader and Skype® were perceived similarly, while Tetris received the worst rating, probably due to the much more complex usage, as here four gestures had to be coordinated and made more quickly. The total average of perceived easiness was 4.5. For some subjects, bars are missing because they did not fill out the questionnaire completely.

Figures 4 and 5 show two different graphical representations from two Tetris plays: Both participants were wheelchair users. The user in Figure 4 (subject 8, DMD) selected arm raising to move the bricks horizontally and trunk movement to rotate and dump the brick down. Figure 4a shows the arm movements graphically, while Figure 4b presents the test person during play. The threshold amplitude of 20 cm (8 in) was easily reached and often surpassed by up to 10 cm (4 in).

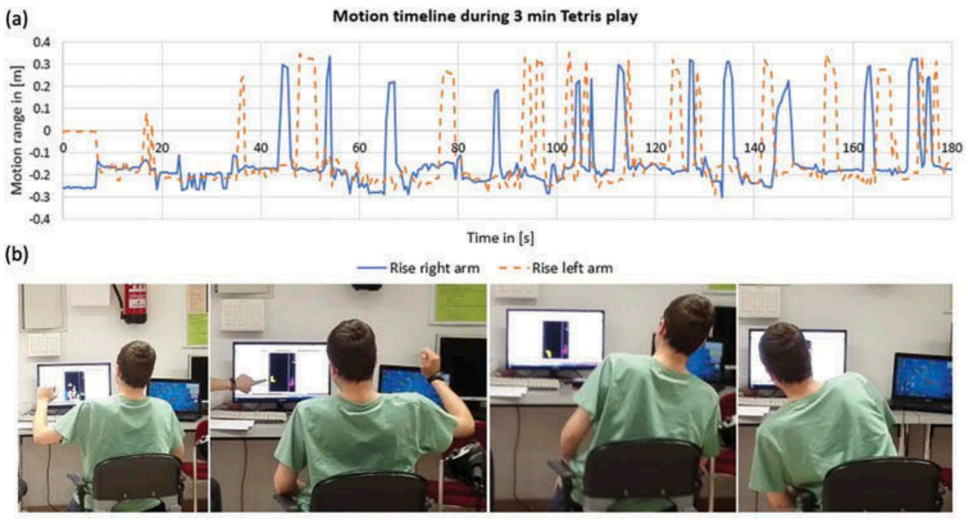


Figure 4. (a) The 3 mins capture of two movements configured for a Tetris session. The threshold was set to 20 cm (8 in) and was surpassed about ca. 10 cm (4 in), (b) The test person while performing the captured arm movements (first and second photo) and left-right trunk movement (third and fourth photo).

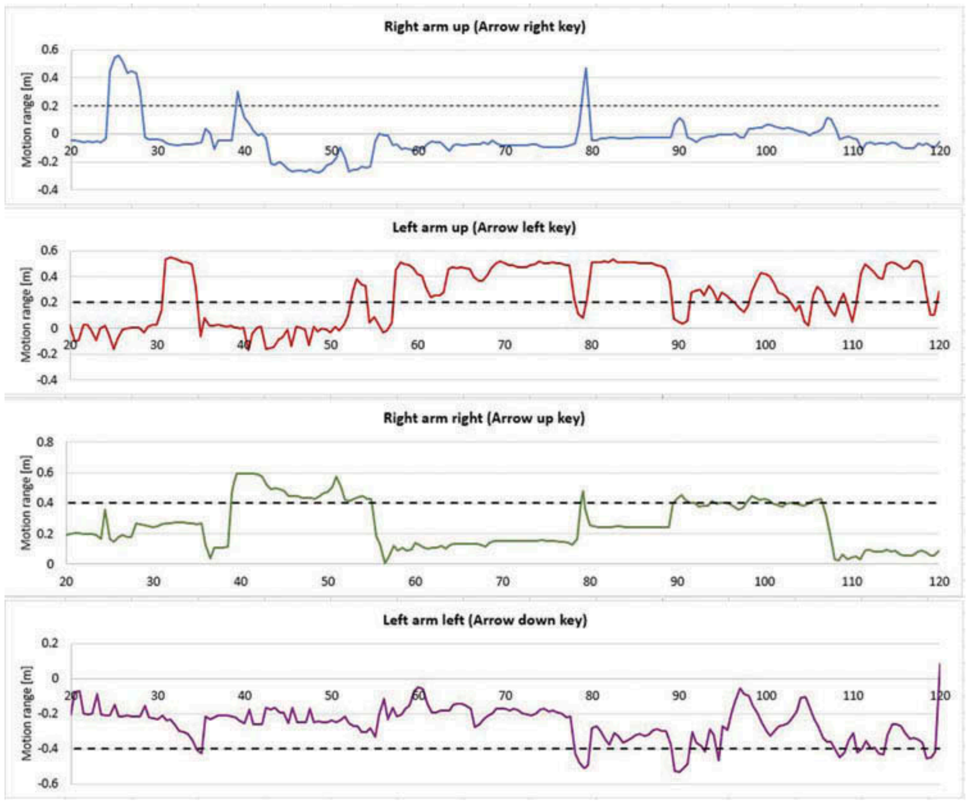


Figure 5. Illustration of two-minute Tetris-play for subject 3. The motion ranges to reach were 20 cm (8 in) vertically and 40 cm (16 in) horizontally and are marked with a dashed line.

The other participant (subject 3, CP, Figure 5) selected only arm movements. Comparing both figures, the different frequencies of movements are noticeable, with subject 8 being the more agile. The two different types of graphs are shown to highlight the medical possibilities of motion analysis: at one glance, it would be possible to see multiple types of performances (e.g., velocity and quality of movements), as well as temporal variations.

Discussion

The main objective of this research is to prove that the tool could be used as an instrument to program physical exercises by using different types of standard software and that disabled persons of different kind have no problem using. Furthermore, the gaps mentioned in literature inherent to typical rehabilitation software wanted to be filled.

By "exercise," we refer to physical activity with the aim to develop and maintain physical fitness or to improve a capability or skill. In this sense, MoKey enables a person to perform repetitive movements using a software application (i.e., passing pages of an e-book by waving the right arm that could be programmed as rehabilitative exercises or be useful to increase physical activity in general).

From a technical point of view, the presented middleware shown works well with several types of off-the-shelf software: games, web browsers, anti-virus programs, office software, and communication software. Eight of 10 tested programs could be executed with gestures through MoKey doing some simple but useful tasks used in daily life, and as such, could be performed doing exercise.

The viability for a wide range of people has been successfully tested with an inhomogeneous group of volunteers (ample age range and severely affected from different rare diseases). One of the major difficulties in rehabilitation is that the reduced motion range of the severely disabled does not allow training in real situations, so it is limited to the physiotherapy sessions. With the help of our middleware, the user could manipulate regularly used applications while working out motor functions that have passed to a secondary level. As the MoKey interface allows to configure the gestures according to each person's requirements, the exercise program can be customized and adapted to changes in user needs (e.g., an improvement or a deterioration in the underlying condition). With our results, we refute the findings of Webster and Celik (2014) that the Kinect would not be suitable to capture gross or weak movements, and are generally felt to be unsuitable for severely disabled patients (Webster & Celik, 2014). Even for users unable to control their movements precisely, as in some cases of CP, the simplicity of motion recognition led to very positive results. Regarding the applicability for users with difficulties using a keyboard, unfortunately it was not possible to find any volunteer who had that problem, so this test was postponed.

The volunteers suffering from DMD, BMD, and spinal muscular atrophy (SMA) have a strong tendency to get tired. When reaching a certain level of muscular fatigue, the activity has to be stopped until the muscle completely recovers. During the tests, two persons complained about tiredness; in general, the volunteers were captivated by the game and did not notice their effort consciously. This shows that activity monitoring is of vital importance to avoid overexertion. MoKey allows the acquisition of motion data, which could be stored for later analysis or transmitted in real-time to a supervisor.

A further gap detected in the literature was the difficulty of configuration for non-experts. MoKey proved to provide an easy handling when it was compared with FAAST. It is less versatile regarding the possibilities of configuration, but the 12 predefined gestures completely covered the needs of the tested volunteers. If any additional gesture were necessary for some user, it could be recorded.

The data obtained during the tests reveal very good responses from the volunteers with respect to physical conditions as well as motivation. All of them stated the experience as positive and confirmed that they would use the tool at home for their exercises. This shows that the middleware would provide long-term enjoyment, because the users can choose the application they like. A further advantage is the possibility to combine everyday tasks with exercise, which gives the feeling of not losing time.

With respect to the goal of combining physical activity with social inclusion, the video conferencing software Skype® was tested and positively accepted. The volunteers confirmed its utility, as it provides the opportunity to communicate with others without any kind of help. The use of an online game would open a further possibility to improve social contacts.

Limitations

System configuration was generally straightforward, as default-values were applicable in most cases. However, the electric wheelchairs presented detection problems in two cases. One was due to a remote control fixed on the armrest that could be avoided by moving the control. The other case was a boy who was very small compared to his chair. Here, no viable detection was achieved as his arms were confused with the wheels.

Users unable to perform wide movements could only perform tiny gestures in front of their body. This would disturb the detection process due to an occlusion of the spine by the hands, such that the distance between limb and body center could not be well measured by the camera. The use of wheelchair adaptations or supporting devices like orthosis may cause further detection problems. Additionally, corporal deformations may create difficulties for the camera when the skeleton does not coincide with standard measures on which library functions are based.

Finally, the very severely disabled, in an advanced state of their disease, will not benefit from this system, as a minimum detectable movement is required. Examples would be advanced states of DMD, limb girdle muscular dystrophy type E2, and congenital or metabolic dystrophies. Others who could benefit from the system would be SMA-3 (SMA type 3), myotonic dystrophy, polyneuropathies, myotonia, or CMT (Charcot-Marie-Tooth), among others.

The execution of complicated tasks in the chosen software is not possible, as this would require the combination of too many different gestures. A further drawback is that an assistant is needed to start a program and occasionally enter some text, if the user is unable to do so.

Conclusions and future work

This article presented a novel, versatile middleware called MoKey that aims at providing new possibilities for disabled people to perform physical exercises, thereby improving their quality of life, and in turn, long-term health benefits. As it can be used at home together with almost any desired off-the-shelf application, the advantages are manifold: Users are more motivated to exercise, as they use their favorite software; the type of software can increase social inclusion (online games, communication software); therapists can program and personalize exercises easily; gestures could be selected or newly defined that are not typically required or avoided in daily life; the quality of movements can be monitored at real time or afterward by analyzing system data, and overexertion can be supervised.

Although the results are encouraging, longer-duration studies are needed to show the long-term benefits for disabled people. Tool usage should be integrated into the regular time schedule of a large group of volunteers and they should be supervised by a therapist. If those tests reveal an improvement or stabilization of muscle functionalities, this type of application would be worthwhile promoting as a valuable aid to enhance physical activity, combined with social inclusion, which impact the everyday lives of the disabled.

To assure a robust performance of the middleware, the system should be improved in terms of precision for small and weak movements for hands and wrist, if possible in front of the body or with the arms rested on the armrest. To achieve this, a solution has to be found to distinguish wheelchairs. Furthermore, it would be interesting to enhance the set of movements to include facial gestures, and thus target people with severe impairments of motor functions, such as those in advanced stages of their disease. To augment the range of programs that can be used with MoKey, mouse control could be integrated. In addition, to overcome the inability to insert text, voice recognition could be added. Finally, it would be interesting for users to receive feedback on the quality of their movements and performance.

Funding

This work was sponsored by Spanish National Plan for Scientific and Technical Research and Innovation: TEC2013-48453-C2-2-R.

ORCID

Martina Eckert, PhD http://orcid.org/0000-0002-4531-9918

References

Bonnechère, B., Jansen, B., Omelinab, L., & Van Sint Jan, S. (2016). The use of commercial video games in rehabilitation: A systematic review. *International Journal of Rehabilitation Research*, *39*(4), 277–290. doi:10.1097/MRR.0000000000000190

Borenstein, G. (2012). *Making things see*. In A. Odewahn & B. Jepson (Eds.), Sebastopol, CA: O'Reilly.

Boslaugh, S. E., & Andresen, E. M. (2006). Correlaties of physical activity for adults with disability. *Preventing Chronic Disease*, *3*(3), 1–14.

Daniels, K., & Worthingham, C. (1986). *Muscle testing techniques of manual examination* (5th ed.). Philadelphia, PA: WB Saunders.

Eckert, M., Lopez, M., Lazaro, C., Meneses, J., & Martinez Ortega, J. F. (2015, August). MoKey—A motion based keyboard interpreter. In *Proceedings of the International Symposium on Consumer Electronics, ISCE. Madrid, Spain: IEEE Press.* (pp. 4–5) doi:10.1109/ISCE.2015.7177799.

Jiang, H., Duerstock, B. S., & Wachs, J. P. (2012). Integrated gesture recognition-based interface for people with upper extremity mobility impairments. In V.G. Duffy (Ed.), *Advances in human aspects of healthcare* (pp. 546–555). Boca Raton, FL: CRC Press.

Kamel Boulos, M. N., Blanchard, B. J., Walker, C., Montero, J., Tripathy, A., & Gutierrez-Osuna, R. (2011). Web GIS in practice X: A Microsoft Kinect natural user interface for Google Earth navigation. *International Journal of Health Geographics*, *10*(1), 45. doi:10.1186/1476-072X-10-45

Koenig, S., Ardanza, A., & Cortes, C. (2014). Introduction to low-cost motion-tracking for virtual rehabilitation. In J. L. Pons & D. Torricelli (Eds.), *Emerging therapies in neurorehabilitation* (Vol. 4, pp. 287–303). Berlin, Berlin, Germany: Springer. doi:10.1007/978-3-642-38556-8

Pool, S. M., Hoyle, J. M., Malone, L. A., Cooper, L., Bickel, C. S., McGwin, G., … Eberhardt, A. W. (2016). Navigation of a virtual exercise environment with Microsoft Kinect by people post-stroke or with cerebral palsy. *Assistive Technology*, *28*(4), 225–232. doi:10.1080/10400435.2016.1167789

Sevick, M., Eklund, E., Mensch, A., Foreman, M., Standeven, J., & Engsberg, J. (2016). Using free internet videogames in upper extremity motor training for children with cerebral palsy. *Behavioural Sciences*, *6*(2), 1–14. doi:10.3390/bs6020010

Skjæret, N., Nawaz, A., Morat, T., Schoene, D., Lægdheim-Helbostad, J., & Vereijken, B. (2016). Exercise and rehabilitation delivered through exergames in older adults: An integrative review of technologies, safety and efficacy. *International Journal of Medical Informatics*, *85*, 1–16. doi:10.3233/978-1-60750-883-0-34

Suma, E. A., Krum, D. M., Lange, B., Koenig, S., Rizzo, A., & Bolas, M. (2013). Adapting user interfaces for gestural interaction with the flexible action and articulated skeleton toolkit. *Computers & Graphics*, *37*(3), 193–201. doi:10.1016/j.cag.2012.11.004

Suma, E. A., Lange, B., Rizzo, A. S., Krum, D. M., & Bolas, M. (2011). FAAST: The flexible action and articulated skeleton toolkit. In *2011 IEEE Virtual Reality Conference. Singapore, Singapore: IEEE Press*. (pp. 247–248). doi:10.1109/VR.2011.5759491

Webster, D., & Celik, O. (2014). Systematic review of Kinect applications in elderly care and stroke rehabilitation. *Journal of Neuroengineering and Rehabilitation*, *11*, 108. doi:10.1186/1743-0003-11-108

World Health Organization. (2011). *World report on disability*. Retrieved from http://whqlibdoc.who.int/publications/2011/9789240685215_eng.pdf?ua=1