

Integrating Motion Tracking Sensors to Human-Computer Interaction with Respect to Specific User Needs

Michal Vinkler*

Supervised by: Jiří Sochor[†]

Faculty of Informatics
Masaryk University
Brno / Czech Republic

Abstract

This paper presents a novel framework for combining Microsoft Kinect device (Kinect) and Leap Motion sensor with Java Monkey Engine and utilizing it for natural human computer interaction. The framework supports the standard input devices (keyboard, mouse) as well as selected motion tracking sensors Microsoft Kinect, Leap Motion). To demonstrate its applicability, the framework was subsequently used for creating the demonstration application called Labyrinth. This application was designed according to the current requirements of psychologists from the Department of Psychology at the Faculty of Arts, Masaryk University, with respect to children with specific disorders, such as ADHD (Attention Deficit Hyperactivity Disorder). Similar kinds of applications are widely utilized by psychologists for observing and evaluating their impact on various groups of patients.

To make the application more realistic, our framework integrates the stereo projection and the jBullet physics library for simulation of the gravitational force applied to the scene. The resulting application forms a comprehensible basis for future applications based on this framework.

Keywords: virtual environment, human-computer interaction, jBullet, physical simulation, Microsoft Kinect, Leap Motion, motion tracking

1 Introduction

The expansion of modern technologies in the last decades introduced the virtual environment and motion tracking to the broader community of users. Moreover, these users currently come from diverse fields, such as flight industry, robotics, art or medicine. In the latter case, many applications utilizing interactive environments were created to help both doctors during the medical intervention [2] [16] and patients during the recovery process. Another very interesting and important area is the usage of motion tracking in the process of the treatment and therapy for children

with various disorders, such as Attention Deficit Hyperactivity Disorder or Oppositional Defiant Disorder.

Attention Deficit Hyperactivity Disorder (ADHD) is a psychiatric disorder of the neurodevelopmental type in which there are significant problems of attention, hyperactivity, or acting impulsively that are not appropriate for the person's age [3]. Oppositional Defiant Disorder (ODD) is described as an ongoing pattern of anger-guided disobedience, hostility, and defiant behavior toward authority figures that goes beyond the bounds of normal childhood behavior. Children suffering from this disorder may appear stubborn and often angry [10].

Studies concentrating on the influence of playing video games to different groups of users consistently show that this activity improves the hand-eye coordination and increases humans visuospatial skills. This increase of brain activity is tied not only to playing games but also to several other real-world scenarios. When the brain encounters new visual and auditory stimulation, or new and different way of processing information, it can influence the brain in the most remarkable ways [1]. This improvement may naturally mark in the person's behavior and then be incorporated into the daily life [6].

The goal of this paper was to create the environment which is straightforward to use and allows designing interactive games for children with specific needs in easy way. The resulting framework uses modern features and devices in virtual environments, enables full body interaction and can display the output using the stereoscopic projection. Moreover, the new system integrates the simulation of physical forces which supports the plausibility and perception of real environment of the implemented games. Although the Microsoft Kinect used for motion tracking is primarily designed for Microsoft Windows operating system, our framework is platform independent as it uses Java programming language and open source libraries for communication with the Kinect device. For user convenience, the detection of gestures for both hands was added.

The idea behind this project is to provide children with smaller tasks in a form of a mini-games. Children are then asked to keep playing and repeat the game to be able to achieve mastery. The system supports basic logging thus

*xvinkle1@fi.muni.cz

[†]sochor@fi.muni.cz

the psychologists can subsequently evaluate if there is any observable impact or improvement, for example that the child can stay focused on one activity much longer than before, etc.

For demonstration purposes we implemented a small game called Labyrinth (see Figure 5). It basically aims to strengthen the hand-eye coordination. In terms of practical application, this can help to improve real world skills, such as handwriting or any other activity requiring the fine motor activity. In the past, it was proved by various studies that children with ADHD have had difficulties with various hand-eye coordination skills [9]. When practicing, the brain can learn how to focus on what the hand is doing. With the repeated play, the brain can become more accurate and more aware of the focus required for complete control of the body.

The Labyrinth playing field is formed by the rectangular-shaped maze (see Figure 1). Player can tilt the board and therefore indirectly control the movement of the metal-like ball located inside. The maze board contains holes (so the ball can fall through) and obstacles defining the inner layout of the maze. The goal is to guide the ball through the maze to the finish hole in the fastest possible time. There are multiple levels pre-designed with difficulty ranging from very easy to hard. For evaluation purposes, selected values are logged. For example, the runtime of the application, the duration of each game, the idle time at the beginning (when the user sees the maze for the first time and has to make the decision about the route) and idle time at crossroads.



Figure 1: Example of the playing field in Labyrinth application.

2 Related Work

Many studies have been done in last years to support the hypothesis that the daily use of brain games can help to strengthen the focusing ability and the executive functioning in adolescents with ADHD. Wegrzyn et al. [18] confirmed the positive impact on participants playing Nintendo DS game Brain Age twenty minutes per day for five

weeks. Hashemian and Gotsis [7] created series of mini-games, each one focused on a specific strength or weakness prevalent in children with ADHD. The project uses Microsoft Kinect for a full body motion tracking.

Virtual reality (VR) is an emerging technology with a variety of potential benefits for many aspects of rehabilitation assessment, treatment, and research. Schultheis and Rizzo [15] focused on examining the specific benefits VR offers to consumers and providers of rehabilitation services. They also discuss the potential areas of application and important considerations in applying the VR technology. Strickland [17] concentrated in her study on using VR as a learning aid with an immersive headset system. Finally, Parsons et al. [14] made a controlled clinical comparison of attention performance of children with ADHD in a VR classroom. These children exhibited more omission errors, commission errors, and overall body movement than normal control children in the VR classroom.

Virtual environment enables to utilize new interaction techniques mainly via spatial input tracking and stereoscopic rendering. From the technical point of view, various virtual reality systems were created which can be exploited by serious games. The VRECKO framework for virtual reality [5] supports several kinds of stereoscopic projection and is able to process data from various input devices. Except for the traditional devices, such as mouse and keyboard, the OptiTrack system can be used, enabling to capture the position of objects in space, 3D mouse, Nintendo Wii Remote and Microsoft Kinect.

3 System Overview

In this section we briefly describe the utilization of Java Monkey Engine for our purpose. Then the motion tracking devices used in our solution are introduced. Finally we will concentrate on the technical solution of devices integration.

3.1 Java Monkey Engine

In order to utilize the most of already implemented and tested features, our new framework was integrated into the Java Monkey Engine 3.0 (JME) [8]. JME is free, open source game engine, programmed entirely in Java, intended for wide accessibility and quick deployment. It is the high level programming engine using the underlying toolkit and graphics library for low level tasks, such as rendering 3D primitives, texturing, scene culling, etc. JME itself does not integrate any motion tracking device. We have chosen to implement the support for Microsoft Kinect and Leap Motion sensors.

3.2 Devices

Microsoft Kinect is a full body motion tracking device. By using a RGB camera and a four-element, linear micro-

phone array it can record both video and audio streams. The depth sensor transmits invisible near-infrared light and measures its “time of flight” after it reflects off the objects. Knowing how long the light takes to return, Kinect can calculate how far away an object is. As a result, an accurate depth map of the scanned area is created.

Leap Motion allows accurate movement tracking of hands, fingers or other rigid objects (for example a pen) working on a very similar principle as Microsoft Kinect. Using two monochromatic IR cameras and three infrared LEDs, the device observes a roughly hemispherical area, to the distance of about 1 meter. The LEDs generate a 3D pattern of dots of IR light which is reflected from the objects placed in the scanned area and then captured by IR cameras.

3.3 Kinect – OpenNI Framework

The OpenNI organization [13] is an industry-led, non-profit organization, formed to certify and promote the compatibility and interoperability of Natural Interaction devices, applications and middleware. The organization has made an open source multi-language, cross-platform framework available – the OpenNI framework – which provides an application programming interface (API) for writing applications utilizing natural interaction.

The main purpose of OpenNI is to form a standard API that enables communication with both physical devices (audio, video and depth sensors) and middleware components, which further process the input data. In this project, NiTE 1.3.1 Middleware was used (see Figure 2).

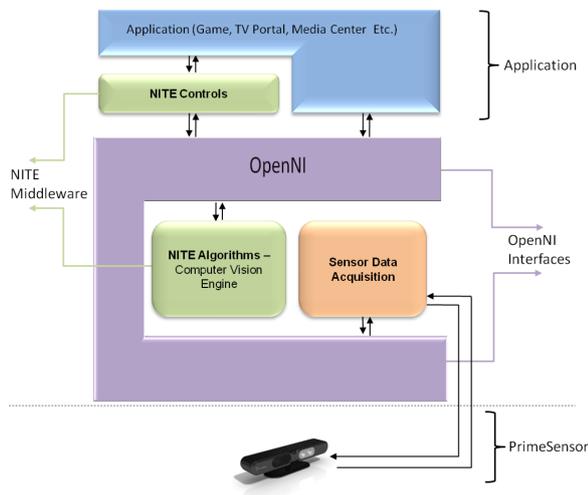


Figure 2: A three-layered view of the OpenNI concept (taken from [13]).

3.4 Kinect – NiTE Middleware

NiTE Middleware [11] was based on the concept including two base paradigms: *Point Control* and *Full Body Control*.

When *Point Control* is used, only one point (typically a hand) is being tracked and NiTE gestures for this point are recognized. There are three possible states for *Point Control*:

- **Not in Session:** In this state, there is no active session, hence the system is in a mode of scanning the scene to detect a “focus gesture” (for example, waving). Once this gesture is recognized, the state changes to *In Session*.
- **In Session:** In this state, there is a hand that is currently in control and being tracked by the system.
- **Quick Refocus:** This is an intermediate state, in which, while in session, the hand point is lost. We don’t want to stop the session yet, but rather give a grace time period in which the session can be resumed with a different (perhaps shorter or easier) hand gesture (referred as *Quick Refocus Gesture*). While in *Quick Refocus* state, the session can also be resumed using the regular focus gesture. Once the grace period has timed out, the state changes to *Not in Session*. The *Quick Refocus* state is optional.

When *Full Body Control* is active, the whole body is being tracked. The player’s skeleton data is extracted from the depth map and handed over for further processing. Therefore, the application receives information about position and rotation of up to 16 joints, together forming user’s full-body skeleton.

In our framework, both controls were utilized. Based on the user experience, the focus gesture was set to waving user’s hand, whereas the quick refocus gesture is defined as raising the hand. Unlike *Point Control*, *Full Body Control* does not determine any states and the flow between them. Therefore, it was necessary to define, how will the application react on the newly detected user in the scene. In order to start skeleton tracking, it is often required to strike a *Psi* pose (standing in front of the camera and holding arms up). NiTE Middleware can also be requested to perform a skeleton calibration right after new user is detected in the scene. If the calibration is successful, the tracking is started (see Figure 3).

Also, initializing Kinect and accessing the events from OpenNI framework is rather complicated. To make this process easier and to achieve full integration with the JME, the Kinect input was programmed to be handled by JME as a standard input (the similar way as for example a joystick). The initialization process is done automatically when the application is started. If not successful (the device is missing or the drivers are not installed), the application can still be controlled by standard input devices or by Leap Motion.

In order to make the tracking-related information easily accessible, new structures representing tracked joints and the whole skeleton were created. The application can ask any time for the tracked skeletons and position of their

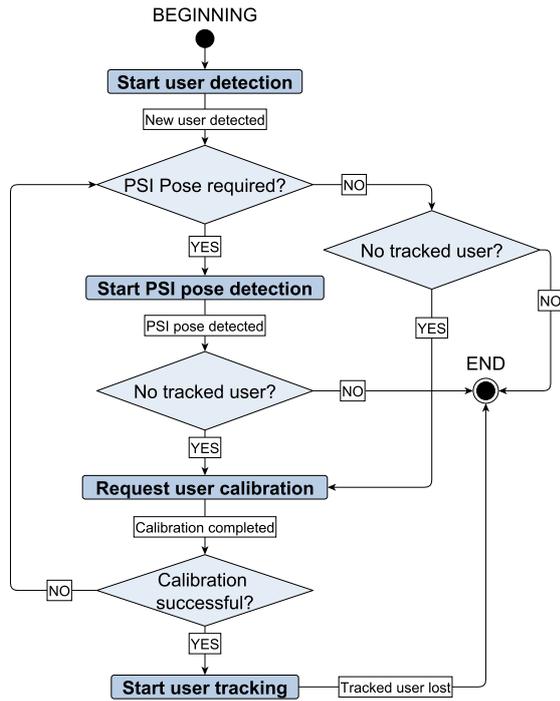


Figure 3: The procedure for new user detection in the scene.

joints. The framework also provides an interface for access to all the important OpenNI and NiTE components (for example VGA and depth image generators), and also to all NiTE middleware events.

3.5 Kinect – Gestures Detection

NiTE defines a set of basic gestures recognized for active hand in Point Control tracking. If we want to use the point tracking for controlling the selector (i.e., mouse cursor), the same hand would have to manage all the gestures as well. In our framework, we wanted to have the possibility to use both hands for controlling the application. Daria Nitescu [12] mentions several strategies to accomplish the select and click action. Based on our experience, selecting the item with one hand and clicking with the other hand is convenient and user friendly. This action however cannot be accomplished using the NiTE gesture recognition ability.

In order to provide as much freedom as possible when controlling the application, we defined a new set of custom gestures. The input data is taken from Full Body Control, which allows us to define our gestures on any limb we choose, overcoming the limitation of NiTE. Gesture manager handles all registered detectors. Whenever the input data is updated, the arm gestures detector checks if any of the defined gestures was performed and fires respective events accordingly. Let us explain how the gesture *Arm Forwards* (analogue of NiTE's *Push* gesture) works. The arm gesture detector retrieves the current position of

the joints representing the user's shoulder and hand and checks if the position is valid (i.e., NiTE can recognize the joints location in the depth map). The distance between the two joints must exceed a predefined value in depth in order to activate the gesture (see Figure 4). Once the user pushed in depth sufficiently enough, he or she has to release the hand. The depth value can be set manually or estimated based on the user's height.

If necessary, we could register for example leg gestures detector (with a set of relevant gestures defined) in a similar way.

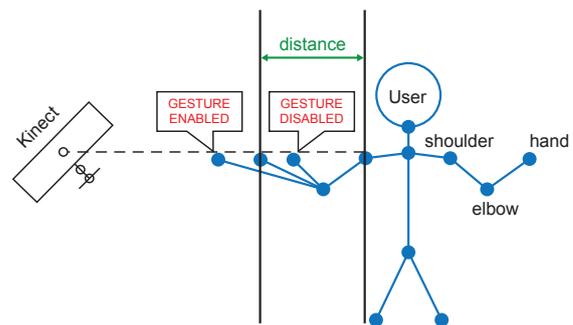


Figure 4: Schematic representation of *Arm Forwards* gesture detection.

3.6 Leap Motion

For Leap Motion integration, the official SDK was used. The whole process is rather straightforward compared to the Kinect integration. When the Controller object is created, it connects to the Leap Motion software running on the computer and makes hand tracking data available through the Frame objects. Each computed frame contains information about recognized hands and fingers of a particular hand, finger tips position, recognized gestures, hand's sphere radius, normal vector and direction and palm absolute position. The pitch, roll, and yaw can be computed from the hand's direction vector. All the information is available to the application through the Controller interface.

3.7 Physical Simulation

Our framework utilizes the jBullet, Java port of Bullet Physics Engine [4]. The jBullet is implemented into the JME as one of the application states. Therefore, it can be paused (resulting in freezing physical simulations) or resumed as needed. The main features of jBullet include: discrete and continuous collision detection, swept collision queries, ray casting with custom collision filtering, support for generic convex and non-convex triangle meshes.

3.8 Stereo Projection

JME was primarily designed for displaying monocular image. By modifying some core classes we were able to create side-by-side stereoscopic projection. It was necessary to position two views with separate cameras into the main window. Hardware cursor provided by the operating system cannot be duplicated, therefore it was replaced with two software cursors displayed in both views. The position of software cursors is calculated from the position of the hardware cursor to match the stereo projection accordingly. Also the GUI rendering was modified to appear in both views.

The two images representing the GUI projection are not mutually shifted in x-axis so they do not create the spatiality effect. However, the shift of the scene images is customizable, which directly determines the parallax type and the depth of the stereoscopic 3D effect.

4 Labyrinth Application

The objective of the sample mini-game in this project is to demand focused attention while players use various hand moves and body gestures to control the application. This process aims to teach children to pay attention for longer periods of time. The game can be fully controlled by implemented motion tracking devices, either Kinect or Leap Motion. When Kinect is used, the cursor movement and tilting of the maze is controlled by user's right hand while the custom-defined gestures (Subsection 3.5) are performed by left hand (see Figure 5). In the case of Leap Motion being used, only one hand is necessary for performing the actions mentioned above. To achieve the realistic behavior, jBullet was used for controlling the ball movement. The side-by-side stereo projection is also supported.

4.1 Events Logging

One of the main requirements from psychologists was logging of the important events in the application. The logged values are:

- Application runtime: Indicates, how much time the user spent with practicing on the daily basis.
- Duration of each game: Used for comparing of the progress.
- Time spent in the decision areas: Each maze contains the defined "decision areas" – the starting area in the maze and forks where the path splits. For each decision area the idle time spent in it is logged (see Figure 6).
- Ball falling through a hole: In both cases of ball falling either into the finish or any other hole, the time of the event and the ball position is logged.

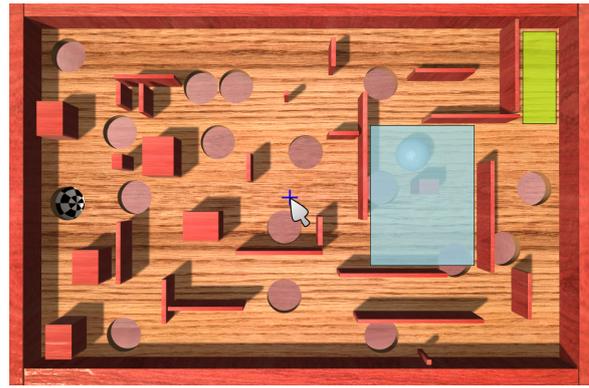


Figure 6: Example of a maze with decision areas highlighted (green - the starting area, blue - the fork).

5 Discussion

We believe that our proposed system along with the logged data will help to gain the novel insights for the psychologists. However, the evaluation of the system as well as the Labyrinth mini-game is currently in an early stage, as this preliminary version is standing at the very beginning of a long-term research project. The definitions of events for logging were provided by psychologists and correspond to the events which are normally observed during personal sessions with psychologists. In comparison with this traditional approach, our automated logging system has one considerable advantage. It removes the possibility that the conclusions made from observations are influenced by the subjective view and experience of the psychologist.

6 Conclusion

In this paper we described the novel framework enabling the integration of Kinect and Leap Motion sensors with Java Monkey Engine. The main aim of the framework was to provide the users with an easy way to develop various applications. The applicability was demonstrated on the Labyrinth mini-game which was designed with respect to children with specific needs. As this mini-game should also serve for further evaluation of the hand-eye coordination and the ability of the player to focus, the event logging system was implemented and integrated. This helps the psychologists to measure and easily compare the improvements.

In the future, the framework could also support the recording of audio and video stream captured by Microsoft Kinect and possibly analyzing these streams to reveal specific patterns in user behavior specified by the psychologists.

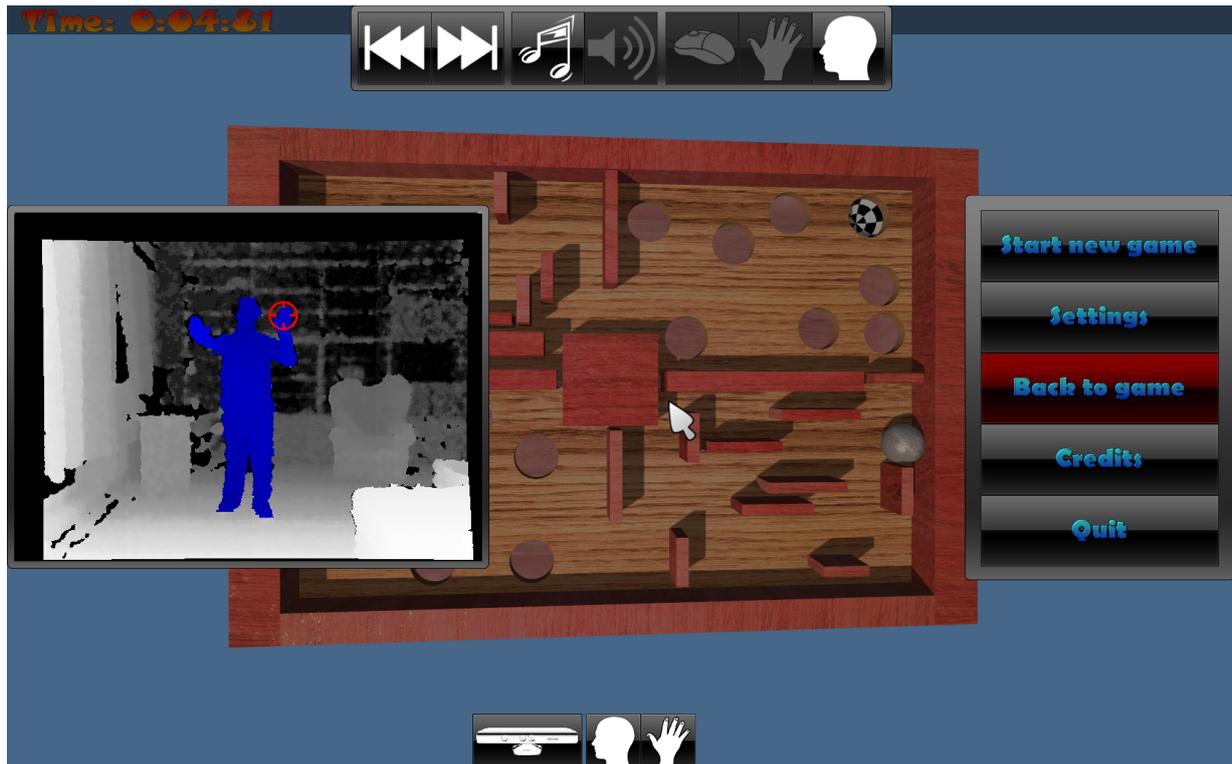


Figure 5: Application controlled by Microsoft Kinect. The depth map and GUI are displayed. The red cross represents the tracked point (right hand), the tracked person is highlighted. The left hand performs the *Arm Forwards* gesture.

References

- [1] Daphne Bavelier, C. Shawn Green, Doug Hyun Han, Perry F. Renshaw, Michael M. Merzenich, and Douglas A. Gentile. Brains on video games. *Nature Reviews Neuroscience*, 12(12):763–768, 2011-11-18.
- [2] Mark W. Bowyer, Kevin A. Streete, Gilbert M. Muniz, and Alan V. Liu. Immersive virtual environments for medical training. *Seminars in Colon and Rectal Surgery*, 19(2):90–97, 2008.
- [3] Ann C. Childress and Sally A. Berry. Pharmacotherapy of attention-deficit hyperactivity disorder in adolescents. *Drugs*, 72(3):309–325, 2012.
- [4] Erwin Coumans, et al. Bullet Engine. <http://code.google.com/p/bullet/>. [cit. 2014-03-12].
- [5] Jan Flasar, Luděk Pokluda, Radek Ošlejšek, Pavel Kolčárek, and Jiří Sochor. Vrecko: Virtual reality framework. In *Theory and Practice of Computer Graphics 2005*, pages 203–208, Canterbury, 2005. Eurographics Association.
- [6] Jeffrey M. Halperin, Anne-Claude V. Bédard, and Jocelyn T. Curchack-Lichtin. Preventive interventions for adhd. *Neurotherapeutics*, 9(3):531–541, 2012.
- [7] Yasaman Hashemian and Marientina Gotsis. Adventurous dreaming highflying dragon. *Proceedings of the 4th Conference on Wireless Health - WH '13*, pages 1–2, 2013.
- [8] Java Monkey Engine 3.0 – Java OpenGL Game Engine. [online]. <http://jmonkeyengine.org>. [cit. 2014-03-08].
- [9] Dubi Lufi and Elisheva Gai. The effect of methylphenidate and placebo on eyehand coordination functioning and handwriting of children with attention deficit hyperactivity disorder. *Neurocase*, 13(5-6):334–341, 2008-05-15.
- [10] Walter Matthys, Louk J. M. J. Vanderschuren, Dennis J. L. G. Schutter, and John E. Lochman. Impaired neurocognitive functions affect social learning processes in oppositional defiant disorder and conduct disorder. *Clinical Child and Family Psychology Review*, 15(3):234–246, 2012.
- [11] PrimeSense: NiTE Middleware – the Natural Interaction Engine. [online]. <http://www.primesense.com/wp-content/uploads/2012/10/PrimeSense-NiTE-Middleware-A4-Lo.pdf>. [cit. 2014-03-08].

- [12] Daria Nitescu. Evaluation of pointing strategies for microsoft kinect sensor device. Diploma thesis, University of Fribourg, Department of Informatics, 2012.
- [13] OpenNI organization: OpenNI Programmer's guide. [online]. <http://www.openni.org/openni-programmers-guide/>. [cit. 2014-03-08].
- [14] Thomas D. Parsons, Todd Bowerly, J. Galen Buckwalter, and Albert A. Rizzo. A controlled clinical comparison of attention performance in children with adhd in a virtual reality classroom compared to standard neuropsychological methods. *Child Neuropsychology*, vol. 13(issue 4):363–381, 2007-06-11.
- [15] Maria T. Schultheis and Albert A. Rizzo. The application of virtual reality technology in rehabilitation. *Rehabilitation Psychology*, vol. 46(issue 3):296–311, 2001.
- [16] Robert J. Stone. The (human) science of medical virtual learning environments. *Philosophical Transactions of The Royal Society B*, 366(1562):276–285, 2011.
- [17] Dorothy Strickland. Virtual reality for the treatment of autism. In *In G. Riva (Ed.), Virtual reality in neuro-psycho-physiology (pp. 8186*. IOS Press, 1997.
- [18] Stacy C. Wegrzyn, Doug Herrington, Tim Martin, and Adriane B. Randolph. Brain games as a potential nonpharmaceutical alternative for the treatment of adhd. *Journal of Research on Technology in Education*, 45(2):107–130, 2012.