EEG: The Missing Gap between Controllers and Gestures

Joshua Carl Hendricks and Sudhanshu Kumar Semwal

Abstract—**Gestures provide ways to easily navigate a system, as seen with the Microsoft's Kinect™. Problems start to occur in gesture heavy systems making it difficult to use gestures to control the system. This paper explores using Neural Impulse Actuator (NIA), an EEG headset, to replace gestures while having the control of a controller. Through the use of face recognition and the Microsoft Kinect, two systems are implemented to demonstrate the NIA as an alternative to gestures. Results are presented showing the success of the NIA.**

Key words— **Computer Games, Entertainment, brain computer interface, NIA implementation.**

I. Introduction

ADVANCES in Electroencephalography (EEG) headsets have made them easier to use. EEG headsets are used in video games, apps, and programs for the handicapped. In future, it is expected that EEG headsets will be able to understand simple thoughts and so it will have potential to revolutionize industries. Neural Impulse Actuator (NIA) is a simple EEG headset for registering alpha and beta waves along with muscle tension in the face and eye movement. NIA captures these signals from the brain using three plastic carbon-fiber injected Electrodes. These electrodes capture the brains EEG or electrical activity from the skin and convert it into biopotentials [1, 11]. NIA allows for the mapping of the biopotentials to keyboard or mouse inputs, allowing the capabilities to be endless.

The main research areas of EEG headsets tend to be accompanied with other technologies. The focus of this paper is test the NIA as an alternative to gestures and acting as a controller. The inputs of a controller can be simplified to moving and pushing actions. Moving action is similar to how a mouse moves the cursor or a joystick moves an object. Pushing action is similar to a mouse click or a button getting pushed to perform an action. The NIA, being a simple EEG headset, would best be used as the push action while the system does the move action. Through two methods, the use of face detection and Microsoft Kinect's body tracking, the NIA is a controller that can be precise and accurate. We have tested NIA as a replacement to gestures that work like controllers in two different application which are described in this paper. First the NIA is tested with a hands-free system, implemented in Java, along with face detection to order something, for example, a pizza. Second, NIA is used to implement a system in Unity 3D using Microsoft's Kinect. Using the NIA, Unity, Kinect (NUK) system simple zombie games are created for the blind and visually impaired (b/vi) and the sighted community. NIA is used for push, or click, while the system implements the move side of the control. Results indicate NIA's success in replacing gestures and providing accurate control.

Manuscript received July 6, 2014; revised August 2, 2014.

Joshua Carl Hendricks was an MS Student at the Department of Computer Science, University of Colorado at Colorado Springs, C) 80918, USA; e-mail: hendricksjc@gmail.com).

Sudhanshu Kumar Semwal is with the Department of Computer Science, University of Colorado, Colorado Springs, CO 80918 USA (e-mail: ssemwal@uccs.edu).

II. Previous Work

A. Neural Headset

Neural signals are constantly on the skin of a person's head even when the person is unaware that they are happening. Thanks to the lowering cost of electroencephalography (EEG) headsets, developers are bringing head control technology to everyday interfaces. Like the NIA, new EEG headsets are as easy to put on and can be as easily used. Many EEG headsets are typically used to allow those with handicaps to perform new tasks, and can also be useful to users with no handicap [1].

Through the users EEG headset, signals can be used to control a quadrotor that could help with mobility [31]. Signals from the user are decoded and converted into movements for the quadrotor allowing it to move in 3D space.

Saulnier used an EEG headset to influence a Roomba, a room cleaning robot, and its behavior [29]. Using the Neural Impulse Actuator (NIA) headset developed by the company OCZ [14], muscle tension from the user was mapped to the robot's speed. Muscle tension was used as a form of estimating the users stress level. If the user was more stressed or had higher tension the robot would move away from the user.

Delving into EEG headsets used within mobile apps, Campbell uses an EEG headset and an iPhone to create a brain-controlled address-book dialing app [5]. The phone app goes through the user's address book showing a list of photos of people in the address book. When the photo matches the person of interest, the user can use the EEG headset through think-triggered or winking to select and call that person of interest. Interested in the possibility to link EEG headsets to mobile devices, Coulton explored this link not only to provide new interactions but also the ability to record brain activity to reveal interactions [6]. A fully functioning mobile game was created which also used a low cost EEG headset to create a new approach for mobile interactions. A marble labyrinth game was created that allowed the users to tilt their mobile device to move the ball.

In the EEG headset, attentive or meditative states could be controlled, and therefore the game was designed around these to unlock gates which provided new areas to explore in the game [6].

B. Kinect

With the release of the Microsoft Kinect, many new innovations were created through the use of its segmenting and body tracking. This is done through the Kinect's RGB camera and depth sensor allowing the Kinect to capture video data in 3D under any ambient light [26]. With the Kinect's cost being very reasonable, it allows anyone who was interested to get one and develop new systems and ideas.

Anjo and Feuerstack [3] introduced a system called Gesture User Interface (GestureUI). The GestureUI system recognized gestures of the Brazilian Sign Language in real-time from a continuous video stream. Using the Kinect, developers took advantage of its power of segmenting and tracking of the body to break down the body into manageable parts. Similarly, Xiaoyu [29] developed a robust vision-based hand recognition system through the use of the Kinect. Color, depth, and contrast are extracted from the Kinect to improve the recognition performance with predefined static and dynamic hand gestures. These gestures would be processed and then mapped to control a computer. By using the depth information from the Kinect the system would locate the hands with the assumption that they would be in front of the body. Once the hands are located they are processed by the system as gestures. Adaboost allowed training methods to be created from hand gestures that were used to train the models [29]. These were all stored in a database. A gesture algorithm mapped different gestures to computer inputs. Different gestures used in succession allowed control of a computer. Kinect allows the user to have a cord free experience still allowing full control of the overall system. The Kinect system [20] allows users to control a program with gestures from the bed lying down. The program would allow the user to check weather, stock, and many other types of information the user might want to see. The system incorporated features such as auto turn, when there was motion, and auto turn off, when there was none, for example, during sleeping [20].

As gestures provide body movement, this can be seen as providing exercise to the user depending on how the gesture is performed. Ganesan and Anthony [9] took this idea and used Kinect to help the elderly exercise regularly [12]. Using the body tracking of the Kinect, a simple game was created with the goal to touch circles and avoid squares. A stick figure representation of the user was displayed along with circles and squares. The user would raise their hands to touch the circles and avoid the squares. When a circle was touched the score would increase while touching a square would lower the score. The use of a score system encouraged movement [9]. Shrewsbury introduced a project that aimed to create a new way to supplement the loss of depth perception in the visually impaired populace [25].
Games for the B/VI Community

Games that have been specialized and created for the b/vi community continue to push the boundaries blurring the line between games for blind and games for sighted. Adapting current games and making them available for the b/vi community is an emerging trend. Morelli and Folmer created a system to enable a b/vi player to play existing Kinect games through the use of cues and vibrotactile feedback [18]. A USB video capture unit captured the game play video stream for processing while a Nintendo Wii remote provided the vibrotactile feedback. The captured video would be analyzed for any visual cues indicating user gestures. Once a visual cue was identified, the system would provide an audio cue or a vibrotactile cue using the Wii remote. This indicated to the user they must preform a desired gesture such as jumping or throwing. Morelli and Folmer's system allowed games on a console to become available to the b/vi community. Folmer and Yuan converted visual cues of a game like Guitar Hero and mapped it to a haptic glove [32]. A glove was created that contained motors at the fingers allowing the user to know which note was needed to be pressed on a Guitar Hero guitar. Modifying Frets on Fire, a Guitar Hero like game, finger motors on a glove would vibrate indicating the note that needed to be pushed.

Commercially available games are not generally played by visually impaired users. Often it is easier to create a game from scratch with visual impairment in mind from the beginning. Friberg and Gardenfors discuss the design of audio games using TiM project (Tactile Interactive Multimedia) and three sound-based games developed by SITREC (Stockholm International Toy Research Center, at KTH, Royal Institute of Technology), Mudsplat, X-tune, and Tim's Journey [8]. Instead of three-dimensional visual environments, sonic landscapes or "soundscapes" are used to build an environment that can be just as exciting as the visual counterpart. Three listening modes were focused on while designing and creating an auditory interface which were casual listening, semantic listening, and reduced listening [8]. Casual listening is listening for the source of a sound and trying to figure out what had caused it. Semantic listening is listening that deals with speech or language. Reduced listening, which is less common, is listening to sounds without needing to consider the source [8]. Within visually impaired games the different audio layers should give information to the user and build a sonic landscape for the user to understand. The sounds of the machine can reveal activity such as if the machine is broken, dangerous, or even its location [8]. SITREC develops and uses these ideas in the development of their games. SITREC's game Mudsplat gives the player control of an avatar to defeat monsters which are throwing mud. Sounds try to give a realistic impression of movement and distance to allow the player to estimate where monsters are located [8]. SITREC's second game X-tune is more of a music simulator than a game. It includes the making of musical compositions, recording and manipulating sounds or voices, creating sound collections, or simply playing with sounds [8]. In the game Tim's Journey the player explores a complex world made up of sounds in a three-dimensional soundscape. The game is divided into different scenes to allow the player to easily identify where they are [8].

Audio alone can allow for very complex games even giving a feeling of a three-dimensional environment. Wired magazine contained an article that showed that even visual games with audio can still be won blind, where a visually impaired person learned to play the visual game Oddworld: Abe's Oddysee from the games audio cues [24]. Through lots of hard work and the help from others, the person can listen carefully to the games sounds and win the game on his own as explained in the article [24]. This underlines the importance of good audio in games for both the blind and sighted.

Recent efforts have come under the general umbrella of crossover applications [30], which are interfaces and games designed specifically for a handicap, for example, the Blind and the Visually Impaired [2, 4, 13, 19, 21], yet these applications crossover to general population.

III. Gestures and NIA

A gesture is defined as "a movement usually of the body or limbs that expresses or emphasizes an idea, sentiment, or attitude" in [16]. For many systems that are using the Microsoft Kinect gestures are ultimately used to control the system.

Gestures are useful in controlling simple systems when they can be discriminated. However, when many gestures are used in a system their movements can overlap if they are too similar. Gestures may also overlap if the user moves too quickly and the system is still trying to catch up. This lag issue creates decoupling of gestures that need to be immediately understood. As seen in the Kinect-Ultra demonstration available online [27], a user, through the use of the Kinect, turns himself into a super hero, Ultra Seven. The user, through gestures, can shoot lasers from his head, from his arm, and even throw a boomerang from his head. Sometime in the demo, the Kinect executed the first gesture but also executed another gesture that user wasn't intending. Some uses of the Kinect and gestures can result in difficult controls. Ryan McCaffrey from IGN reviewed Steel Battalion, a Kinect Xbox 360 game [17]. According to him "as we've unfortunately come to expect form most Kinect titles, the motion controls are terrible". Ryan went on to say "in the heat of battle we had constant issues both at home and in the office with accidentally closing the viewport hatch when we were trying to simply look out the view pane" and "Kinect again frustrates … we were frequently hitting other buttons on that panel or, worse, unable to pull out the panel all the way at all" [17].

Gestures can be difficult to use for the visually impaired. Gesture-based video games are inaccessible to players who are blind as they require the player to perceive visual cues that indicate what gesture to provide and when they should be provided [18]. Controllers or keyboards are better to use than gestures. If up is pushed by the user he/she expects something to move up. This same button control tries to get applied to gestures but ultimately the user doesn't have the same ability to perform the gesture command and get fast and predictable results like a button click. For example if a user wants to perform a gesture that moves up, the gesture could map to other similar gestures which could be the case more often than not. Also, similarity of gestures make the user pass through multiple gestures on their way to a specific gesture. How long does the user need to hold the gesture before the command responds? What if the user wants to perform the gesture again, do they stay where they are or do they move then perform the gesture again? Replacing a button click by a gesture makes the system more unreliable. In the Kinect game, Kinectimals, the menu system requires you to hold your hand on a menu item for a few seconds before it is selected. This approach works fine and is precise. However, this gesturing is far slower than its counterpart key-based controllers.

In our research, we found NIA coupled with gestures could allow gestures to be meaningful and intentional, or even replace gestures altogether. NIA could make something happen when users want and as many times as they want without waiting. We describe a system that replaces gestures and gives controller like functionality by using NIA. This system fills in the missing gap between gestures and controller buttons.

IV. Designing using NIA

A. EEG Face Detection

The EEG headset, Neural Impulse Actuator (NIA) from OCZ, was selected due to its low price in our implementation. While investigating NIA, it was determined that NIA must be used in a simplified way. NIA extracts the EEG signal from the muscles, brain, and eyes, respectively [33]. We found brain and eye movements in the NIA to be difficult to incorporate for control. So they were not used in our implementation, as we could not rely on them. By not using a mouse and keyboard for navigation, we wanted to test NIA's precision. We use face tracking on FaceTrackNoIR along with a PS3 eye camera to allow the user to move the mouse cursor by moving his face. This allowed hands-free control of the system. To test feasibility, a program was created to order pizza. Boxes were displayed on the screen with text indicating audio that would be spoken as seen in Figure 1. When a user clicked a box using a white mouse-like indicator the audio would be spoken through speakers. A user would call for pizza leaving the phone on speaker mode and then select the appropriate boxes to order a pizza. The NIA was successful as a controller as it always pushed the box correctly and quickly, similar to a user pushing a button. NIA worked well for button clicks in a system.

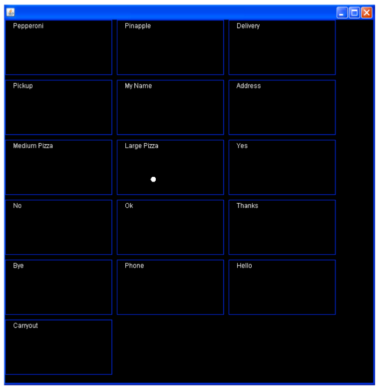

Figure 1: Ordering Pizza using NIA

B. OpenNI, Microsoft Kinect SDK, and Unity3D

OpenNI, an open-sourced software framework, was used with the Kinect. The NITE middleware library works seamlessly with Kinect and provides a wide range of

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

functionalities [28]. OpenNI is also available for Windows, Mac, and Linux which makes our implementation portable across multiple platforms. OpenNI seemed to respond very well to user movements. However, OpenNI could be difficult to setup and upgrade from one version to another. Wrappers for OpenNI were created allowing multiple programming languages to create programs using the Kinect. We tested multiple graphic engines. Java Monkey Engine (JME) was used along with the Java wrappers in OpenNI. Unity, a cross platform game engine, allowed for easy development of games which can be released on multiple gaming devices. A wrapper was used for Unity that allowed use of the Kinect SDK distributed by Microsoft. The Kinect wrapper included scripts for scripting in Unity that controlled every aspect of the Kinect. Due to the ease of use of Unity with the Kinect and the strong results, it was the platform of choice for our implementation.

C. Rail Shooter

A rail-shooter, or light-gun shooter game acts like a first-person shooter but the movement of the player is controlled by the game. Similar games that use this approach are games like Time Cop, Time Crisis, or even Duck Hunt, to some extent. By using a rail-shooter approach, the need to navigate through an environment is eliminated which simplifies the overall system but also allows the user to concentrate more on playing the game instead of trying to move. This approach also works well for the Kinect as it simplifies its usage and wouldn't require the user to move to navigate in the game. Due to the popularity of zombies and its simplicity, we decided that this game would be about zombies. Our intention was that this would allow visually impaired and sighted to both enjoy a zombie game together. The user would stand still in front of the Kinect and only need to move his arms. When the user moves his arms the guns also move. The game tries to mimic the user's movement accurately. It would be possible for the user to shoot the guns through the NIA's facial muscle movement recognition and extraction. By moving an eyebrow, jaw, or even winking, the guns are activated to fire providing superb control. Audio is provided by speakers placed in front of the user and is not intrusive as we continue to allow a hands-free system. Two games and a test demonstration were created using this approach.

D. Shoot Blocks

Shoot Blocks were created to test the environment and create a demonstration for NUK system. A simple 3D environment was created with a model representing the player and with cubes in front of the model as seen in Figure 2. The user would then use the Kinect and NIA to shoot the blocks that were in front of the player on the screen. As blocks were hit they would randomly change colors to help indicate that the block was successfully hit. While testing Shoot Blocks, some system problems related to precise-shooting, became apparent which made the system very difficult to use. It was clear in Shoot Blocks that precisely shooting at the cubes was not easy. Because of the setup and some approximation inherent in using Kinect, there was a lot of guesswork with where a cube could be in relation to the players model in 3D space. A visual indicator would be helpful in knowing where the player was pointing the guns. We created a laser pointer starting from the guns by using a ray that shined in the direction the guns were pointing. This also indicated where the bullets would be firing. Accuracy improved tremendously with the laser pointer allowing for better control in firing at the cubes. However, there were still times that the laser looked to be on a cube, but due to the 3D depth, sometimes it really wasn't on the cube but near it. In addition, the laser pointer is a visual representation which would be impossible for the visually impaired to use this mode. We solved these two problems by developing an audio laser. The laser pointer that was on the guns was altered into an audio-laser-pointer which would emit a sound when on a cube. The sounds would be different for each gun allowing for easier judgement as to which hand was pointing at a cube and which wasn't. These audio-laser-pointers produced a higher pitch for the right and a lower pitch for the left. Ultimately, this allowed the user to move his hands around to “feel” the cubes through the audio. By feeling the cubes the user would know where the cubes were and also know he had a precise shot.

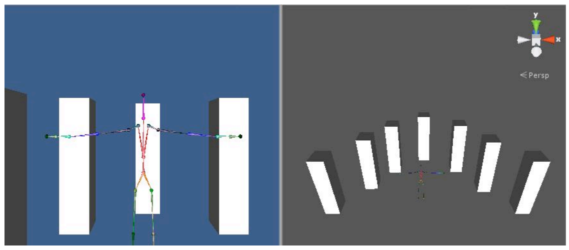

Figure 2. Shoot Blocks Demonstration

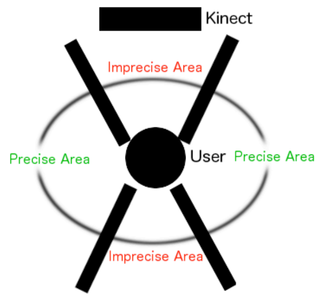

Figure 3. Kinect's Precise And Imprecise Areas

There was a problem in the Kinect SDK that didn't allow the user to put his arms directly in front of him without the arms becoming unstable on the model. This unstableness made the arms jump around without the ability to control them. This was approached in two ways. First, a virtual box was placed in front of the user. Whenever the user had his arms anywhere inside that box it would allow the user to shoot the blocks in front of him. When the user did put his hands forward it was thought the arms would fall into the box regardless of the unstableness of the Kinect model's arms (Figure 3). This, however, wasn't the case and the unstableness of the Kinect model would randomly fall outside of this box. This made any enemies in front of the user very difficult to hit even with the virtual box and usually required multiple shoots before hitting them. The second approach was to always make sure the enemies appeared in the "view" of the Kinect model's arms. The view was determined from inspection of the Kinect model's stable areas showing there was an acceptable area the user could shoot enemies. In Figure 3 it is shown from a 45 degree angle from the front of the user to their back composed of the precise areas that the Kinect could shoot. This area includes areas behind the player. Due to the adding of the audio cues that were previously created, enemies could come from places the user couldn't see visually but could feel. Shoot Blocks were integrated with the above changes to improve the demonstration. By integrating the above changes, the user was better able to shoot blocks with precision.

E. Zombie Village

Zombie Village was created using NUK. The game would take place in a village where the user would need to protect from zombies. A simple story was created through use of the characters talking. Zombie Village would start off with an initial practice session before moving onto the game. This would help the user get familiar using the Kinect and NIA together. Once the practice session was complete the game would start and the user would start to defend the village. As a rail-shooter, the game would move the character around the village stopping at different areas of the village to destroy the zombies. As each area is visited, the zombies were increased by one or two. The game ends outside of the village in a big zombie showdown with several zombies. The game is completed after all zombies are destroyed from the village. Figure 4 shows Zombie Village with zombies. To allow the user to know if they shot a zombie the zombie would make a howl or moan indicating they've been shot. The zombies also made different moans and howls to indicate they were on screen and where they were. Sounds in Unity were built to be 3D to better allow finding the zombies and in which direction they are. Indicating the user is moving through the village, footsteps from the player can be heard as they run around.

Figure 4. Zombies [15] approach in Zombie Town [10].

F. Zombie Wave

Zombie Wave is much more simple then Zombie Village. Zombie Wave has no graphics other than zombies and a white floor. Zombie Wave gets its idea from the popular Call Of Duty Nazi Zombie. Zombies come in waves and the player defeats them. The game is over when the player is touched by a zombie and the number of the wave would become their score. Zombies can spawn around a player in any of the precise areas the player can target as explained earlier. Once a zombie reaches the player the game is over. The player doesn't move as they do in Zombie Village, they stay still and have the zombies come to them. Figure 5 shows Zombie Wave as zombies start approaching from wave 4.

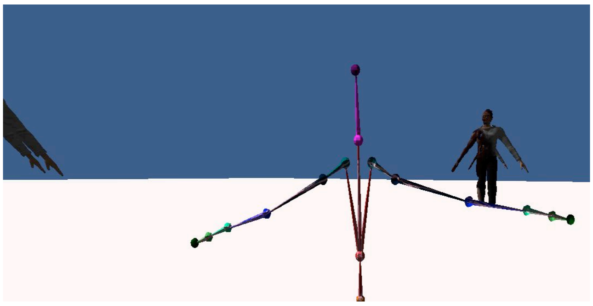

Figure 5. Zombie Wave Zombies Creeping

II. RESULTS

The NUK system worked well for shooting cubes and zombies. The ability for a user to use their own body and facial muscles proved to be easy and consistently reproducible between different users. The Shoot Blocks and Zombie Wave demonstrations had a normal lag, between user and Kinect, which has been seen in many other Kinect games created. The Zombie Town demonstration's lag would sometimes become noticeable as zombies are being created initially, but became better as the demonstration went on. The system was tested on a Windows 7, 2.90 GHz PC with 6GB ram and could be faster with better hardware. It was possible to create a simple story while using the rail shooter approach. The Kinect and NIA provided a hands-free approach that was simple to use. NIA proved to be a useful alternative to gestures used as a controller. While testing Zombie Town the NIA was replaced by a wireless mouse to better compare the two inputs. The mouse was noticeably heavy and extra work was required to ensure so that it was not dropped. Due to this, we noticed that movements toward zombies tended to be slower while the arm was holding the mouse. Whenever a cube or zombie were supposed to be shot, both the click of the mouse or usage of the NIA resulted in it being shot when and where the user was aiming. The NIA allowed a better hands free and intuitive experience according to feedback provided when the game was demonstrated in open house for our college to potential students.

Zombie Wave is very similar to Zombie Town however it had one big difference that made it unique. It made the user scan places with their hands and that made it exciting and interesting. Through the use of the audio-laser it made users "feel" around their entire body to find zombies. This created a new theatrical experience, that was unexpected fun to play with and was something that was unique and novel to Zombie Wave implementation. In conclusion, the NUK system worked well together allowing the user to play the game. NIA was shown to work well as a controller during the demonstrations. Kinect was easy to use and allowed pointing to zombies to be intuitive and easy. Pointing to zombies with the audio-laser was the best iteration of all.

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

V. Results and Conclusions

This paper has focused on testing the NIA as a replacement for gestures and its usage as a controller. Our order pizza implementation using the hands-free NIA controller showed that it was possible to have precise control. The paper also showed the NUK system could be used to replace gestures. The demonstrations show the feasibility of the NIA as an intuitive, robust, and effective controller in Kinect and Unity systems. In our implementation, audio-laser provided a means to indicate where a zombie was without needing to look. This feature was unique and allowed b/vi community and sighted to have a new tool to play games. It does add noise to the array of existing noises that may create overload for some users. Haptic feedback of some kind might help eliminate the sounds in the game and allow the player to feel where the zombies are literally, but this is a future investigation. Moving to a more complex EEG headset such as Emotiv EPOC neuroheadset could allow for better game control and new interactions while keeping handsfree. We believe that we have laid the foundation to co-develop games which can be used by both b/vi and sighted population together in a hands-free manner.

References

[1] Allison, B., Graimann, B., & Gräser, A. (2007, June). Why use a BCI if you are healthy. In *ACE Workshop-Brain-Computer Interfaces and Games* (pp. 7-11).

[2] Almurayh, Abdullah and Sudhanshu Semwal, Cultural Considerations for Designing Crossover Applications for the Visually Impaired, IEEE IRI 2013 – The 14th International Conference for Information Reuse and Integration, DIM Program San Francisco, USA, August 14-16, (2013).

[3] Anjo, M. D. S., Pizzolato, E. B., & Feuerstack, S. (2012, November). A real-time system to recognize static gestures of Brazilian sign language (libras) alphabet using Kinect. In *Proceedings of the 11th Brazilian Symposium on Human Factors in Computing Systems* (pp. 259-268). Brazilian Computer Society.

[4] Brunner, Andrea and Sudhanshu Kumar Semwal, Musical Canvas: A Drawing Tool for the Visually Impaired, MHCI 2013 International Conference on Multimedia and Human Computer Interaction conference, Ryerson University, Toronto, Canada.

[5] Campbell, A., Choudhury, T., Hu, S., Lu, H., Mukerjee, M. K., Rabbi, M., & Raizada, R. D. (2010, August). NeuroPhone: brain-mobile phone interface using a wireless EEG headset. In *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds* (pp. 3-8). ACM.

[6] Coulton, P., Wylie, C. G., & Bamford, W. (2011, September). Brain interaction for mobile games. In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments* (pp. 37-44). ACM.

[7] Folmer, E., Yuan, B., Carr, D., & Sapre, M. (2009, October). TextSL: a command-based virtual world interface for the visually impaired. In *Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility* (pp. 59-66). ACM.

[8] Friberg, J., Gärdenfors, D. "Audio games: new perspectives on game audio." *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*. ACM, 2004.

[9] Ganesan, S., & Anthony, L. (2012, May). Using the kinect to encourage older adults to exercise: a prototype. In *Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts* (pp. 2297-2302). ACM.

[10] GearTech Games. (Model Designer) (2012). *Medieval Village*[Model]. Retrieved from Unity Asset Store.

[11] Hagedoorn, H. (2008, December 11). *OCZ nia review- neural impulseactuator*http://www.guru3d.com/articles_pages/ocz_nia_review_neural_impulse_actuator,1.html , December 2008

[12] Ijsselsteijn, W., Nap, H. H., de Kort, Y., & Poels, K. (2007, November). Digital game design for elderly users. In *Proceedings of the 2007 conference on Future Play* (pp. 17-22). ACM.

[13] Keith Johnson and Sudhanshu K Semwal, Shapes: A Multi-Sensory Environment for the B/VI and Hearing Impaired Community, 2nd International Workshop on Virtual and Augmented Assistive Technology (VAAT) at IEEE Virtual Reality 2014, 29 March - 2 April, Minneapolis, MN, USA, pp.1-6 (2014).

[14] Kowaliski, C. (2008, June 27). *Ocz's neural impulse actuator*. http://techreport.com/review/14957/ocz-neural-impulse-actuator

[15] Mixamo. (Model Designer) (2011). *Zombie Character Pack* [Model]. Retrieved from Unity Asset Store.

[16] "Gesture." *Merriam-Webster.com*. Merriam-Webster, 2013. Web. 26 July 2013.

[17] McCaffrey, R. (2012, June). Steel Battalion: Heavy Armor Review. http://www.ign.com/articles/2012/06/19/steel-battalion-heavy-armor-review

[18] Morelli, T., & Folmer, E. (2011, June). Real-time sensory substitution to enable players who are blind to play video games using whole body gestures. In*Proceedings of the 6th International Conference on Foundations of Digital Games* (pp. 147-153). ACM.

[19] Namburu, Mounika and Sudhanshu Kumar Semwal, Voice Morphing for the B/VI Community, MHCI 2013 International Conference on Multimedia and Human Computer Interaction conference, Ryerson University, Toronto, Canada.

[20] Oh, J., Jung, Y., Cho, Y., Hahm, C., Sin, H., & Lee, J. (2012, May). Hands-up: motion recognition using kinect and a ceiling to improve the convenience of human life. In *Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts* (pp. 1655-1660). ACM.

[21] Ping, Li and Sudhanshu Kumar Semwal, Interactive Math Game for Elementary Aged School Children, MHCI 2013 International Conference on Multimedia and Human Computer Interaction conference, Ryerson University, Toronto, Canada.

[22] Sall, A., & Grinter, R. E. (2007). Let's get physical! In, out and around the gaming circle of physical gaming at home. *Computer Supported Cooperative Work (CSCW)*, *16*(1-2), 199-229.

[23] Saulnier, P., Sharlin, E., & Greenberg, S. (2009, March). Using bio-electrical signals to influence the social behaviours of domesticated robots. In*Proceedings of the 4th ACM/IEEE international conference on Human robot interaction* (pp. 263-264). ACM.

[24] Schreier, J. (2011, April 07). *How a blind gamer plays zelda by ear*. Retrieved from http://www.wired.com/gamelife/2011/04/blind-gamer-plays-zelda-by-ear/

[25] Shrewsbury, B. T. (2011, October). Providing haptic feedback using the kinect. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility* (pp. 321-322). ACM.

[26] Smisek, J., Jancosek, M., & Pajdla, T. (2013). 3D with Kinect. In *Consumer Depth Cameras for Computer Vision* (pp. 3-25). Springer London.

[27] Tomoto S. Washio. kinect-ultra-transform yourself to superhero by kinect. http://code.google.com/p/kinect-ultra/, July 2011.

[28] Villaroman, N., Rowe, D., & Swan, B. (2011, October). Teaching natural user interaction using OpenNI and the Microsoft Kinect sensor. In *Proceedings of the 2011 conference on Information technology education* (pp. 227-232). ACM.

[29] Xiaoyu, W., Cheng, Y., Youwen, W., Hui, L., & Shengmiao, X. (2012). *An intelligent interactive system based on hand gesture recognition algorithm and kinect*. In *2012 Fifth International Symposium on Computational Intelligence and Design* (pp. 294-298). doi: 10.1109/ISCID.2012.225

[30] Wilke, B., Jonathan Metzgar, Keith Johnson, SK Semwal, Bonnie Snyder, KaChun Yu, Dan Neafus. Crossover Applications. IEEE VR 2009 Conference, pp. 305-306 March 14-18 (2009).

[31] Yu, Y., He, D., Hua, W., Li, S., Qi, Y., Wang, Y., & Pan, G. (2012, September). FlyingBuddy2: a brain-controlled assistant for the handicapped. In*Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (pp. 669-670). ACM.

[32] Yuan, B., & Folmer, E. (2008, October). Blind hero: enabling guitar hero for the visually impaired. In *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility* (pp. 169-176). ACM.

[33] Zhang, B., Wang, J., & Fuhlbrigge, T. (2010, August). A review of the commercial brain-computer interface technology from perspective of industrial robotics. In Automation and Logistics (ICAL), 2010 IEEE International Conference on (pp. 379-384). IEEE.