

Use of a Haptic Glove-Gaming Interface to Stimulate Kinaesthetic and Auditory Coordination in Blind Children

Authors: Abhishek Pillai¹; Mohan Kshirsagar²; Reetu Jain³; Shekhar Jain⁴

PACE Junior Science College, Andheri¹; Singhad College of Technology, Lonavala²; On My Own Technology Private Limited, Mumbai^{3,4}

Abstract – Gaming develops coordination, problem-solving skills, quick movement and manipulation making the youth more alert, swift in reaction, for real-time decision making skills. These are skills to be honed by the visually impaired as they need them for survival. With 1.4 million blind children in the world, accounting for 70 million ‘blind-years’, this platform aims at improving their Hand-Sound coordination along with all other tactile decision-making reflexes. This platform, “Beyond Vision”, is a fast-paced brain training environment that demands coordination of motor skills, through a haptic-feedback glove, with the sound dynamics in the virtual environment, to enable coordinated decision making, use of strategies, thus effectively increasing their sensitivity to sound and touch. The blind are extremely sensitive to interaural differences and this has been utilised to create a computer-simulated environment that helps them to orient themselves to their surroundings. The primary aspects of this application are the gaming interface with a 3D sound system (for skill development) , the training-analysis system (that lays focus on the user’s sensory ability) and the haptic glove, that communicates with the game software, thus creating an intuitive experience through tactile feedback. To further serve the purpose of seamless interaction between the digital and real world, the game implements a combination of Reinforcement and Imitation Learning, along with a simple data analyser that tracks the individual’s progress, guiding him through the ‘levels’. Once implemented on a larger scale, even real-life locations can be virtually modelled to emulate specific situations.

Keywords – Gaming Interface, Blind, Haptic-Feedback Glove, Deep Reinforcement Learning, 3D-Sound System, Data analysis.

I. INTRODUCTION

Gaming improves memory, spatial orientation & coordination, information organizations and fine motor skills. Like exercise, playing games for as little as 30 minutes a day, can improve one’s life skills, like making real-time quick decisions, using strategies and anticipating consequences.

Visual impairment limits people’s ability to perform everyday tasks thus reducing their ability to interact with the surrounding world. This impacts the person's quality of life ,independence, and mobility and has been linked to falls, injury, and worsened status in domains spanning mental health, cognition and social function. Individuals with visual impairments must be taught compensatory skills and adaptive techniques in order to be able to acquire knowledge from methods other than sight. Reduced vision often results in a low motivation to explore the environment, initiate social interaction, and manipulate objects. The limited ability to explore the environment may also affect motor development in children.

According to the World Health Organization (WHO), blindness and vision impairment affect at least 2.2 billion people around the world[1]. Childhood blindness affects approximately 1.4 million children.75% of the world’s blind children live in Africa and Asia. Figure(A) shows the distribution of blind people across the globe.

Figure (B) depicts the prevalence of vision loss in the world.



Fig [A] :Number of people affected by all-Vision loss (2020)

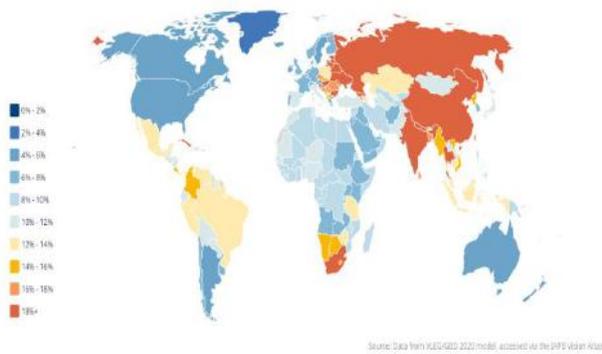


Fig [B]: Prevalence of Vision loss (2020)



Figure [C] :Age-Standardized prevalence of Vision loss

Figure (C) shows the age-standardized prevalence of Vision loss (2020)[2]

Table [I]: Distribution of Blindness across different countries (2020)

| Sr. No | Country | Total with Vision Loss | Blindness |
|--------|------------|------------------------|-------------|
| 1 | India | 275 million | 9.2 million |
| 2 | China | 274.3 million | 8.9 million |
| 3 | Indonesia | 34.9 million | 3.7 million |
| 4 | Russia | 28.6 million | 0.6 million |
| 5 | Brazil | 26.6 million | 1.8 million |
| 6 | Bangladesh | 26.3 million | 0.9 million |
| 7 | Pakistan | 26.3 million | 1.8 million |
| 8 | Nigeria | 24.3 million | 1.3 million |
| 9 | U.S.A | 16.3 million | 0.6 million |
| 10 | Mexico | 16 million | 0.5 million |

As indicated by Table [I] , blindness and vision impairment vary widely from region to region, being largely determined by socioeconomic development.

Due to the loss of the primary sensory modality, visually-impaired individuals learn through experiences that make use of their remaining senses, particularly hearing and touch. Although such experiences are provided through Orientation and Mobility (O&M) skill training, the lack of refinement of the curriculum for children and the potential risk involved, render it ineffective.

The project aims to create a system to train visually impaired children in a virtual environment, eliminating the risk to life involved, whilst developing essential navigation skills. This game aims at:

1. Continuous and Progressive training for the visually impaired, on linking the proximity of the SOUND to ACTION
2. Multiple varying sound systems that can be incorporated to make it closer to real-life experiences
3. Creating an automated, cost-effective, dynamic training system with a high standard of accessibility
4. Most importantly the game, consistently works on improving their decision making reflex, based on two sensory skills that they utilize the most in their daily lives

II. WORKING PRINCIPLE

This project has been designed on the fact that the Blind are extremely sensitive to interaural differences. Thus, this project incorporates audio points with 3D audio for spatial orientation in the virtual surroundings. To use the application, once installed, all the user has to do is calibrate the Haptic-Glove by resetting it. The entire application can be accessed simply through speech and Audio Protocol. It also provides a User Interface for sighted people to manually access. Thus, the user can navigate and access each element of the software easily. For every action taken, the user is provided with audio-feedback, so that the user has an idea of the interactions and the consequences of users actions in the virtual world. For instance, if the user makes a move that causes failure in the

level, the user is made aware of the action's consequences, so that their performance can be improved in the next attempt.

An important aspect of this system is the Assist Mode that integrates an artificially intelligent agent (based on Deep Reinforcement Learning) with the existing environment, to train the player, without any additional human guidance. Thus, the entire learning process is automated. The user, on starting the application, must create an account, that stores the details of the user's past performance and experience; this account in the external database is constantly Automatic updated as the user attempts a level. The audio in the level, is 3D; certain areas in levels rely on a combination of 3D and 2D audio(called Spatial Blend)[3]. This audio, from audio points that are strategically spawned in specific areas of the level, is controlled by a logarithmic rolloff.

The Haptic-Glove, that serves as an essential component of this project, links the two worlds. Haptic technology, also referred to as kinaesthetic communication or 3D touch, essentially means any technology that can create the experience of the sense of touch, through vibrations, forces, and motions, to the user. The Glove communicates with the user through coordinated tactile feedback.[4] To use the glove, the user can connect it, through a USB (A to B) cable to a computer that has the game software installed. The data obtained as a result of the user's actions, is not only implemented in the virtual environment, but is also sent through the serial channel to the Arduino, to generate output in the form of vibrations. The data obtained from the MPU6050 and the push-button on the Arduino is also sent to the software as input, for the actions to take place.

The system comprises two principal components:

1. The Virtual Training Environment
2. The Haptic-Feedback Glove

Figure (D) shows the high level block diagram of the entire system.

The user can develop essential skills and learn whilst enjoying, thus creating an unique, easily accessible system that provides important skills to the individual.

2. High-Level Block Diagram

The Scene Manager serves as the very backbone of each scene. It governs every single element in the scene, including UI elements. It enables communication between the Script and Animation Engines, and implements every user-defined action (along with its consequences) in the scene. It also plays a pivotal role in space partitioning and interaction between objects.

The Scene Manager, based on the inputs from the Script Engine, relays information to both, the Animation Engine as well as the Audio Manager, thus handling the primary aspects of the working of the scene. The most important part of the interface, however, or the 'brain' of the operation is the Script Engine. It incorporates all scripts and libraries they depend on, functioning based on the requirements of the level. The Script Engine is responsible for determining, defining and conveying the required action to the Scene Manager. It is closely associated with the Timer, to carry out the execution in real-time. The Script Engine is also linked to the Animation Engine, providing all the input it requires to function. The engine processes all the input (from both, the microphone (voice-commands) and the haptic-feedback glove) and relays the output to the appropriate part of the system, with minimum latency, aiding the smooth working of the application.

As the primary form of output is audio, the Audio Manager forms a crucial aspect, and constantly receives data from the Script Engine, relaying it from the collection of predefined sounds to the sound renderer. The Script Engine ensures that the Animation Engine functions simultaneously, so that the output as a whole is in sync. It also relies on the Physics Engine, to determine the possibility and degree of movement along each axis, as well as relative to the player's position. Additionally, the Physics Engine enables interaction between various components in the scene.

The Script Engine communicates with two key features of the application, the database and the Neural Network model (Inference only, when the Assist Mode is enabled), processing the data from each to produce the desired result. The user's data is modified and saved frequently for further analysis, to determine the training path to follow. The course of training can also be manually altered, making the system potentially useful to all. Messages can be sent and received from the Input-Output subsystem through the Haptic glove as well as speech.

Figure (F) displays the high-level block diagram of the system.

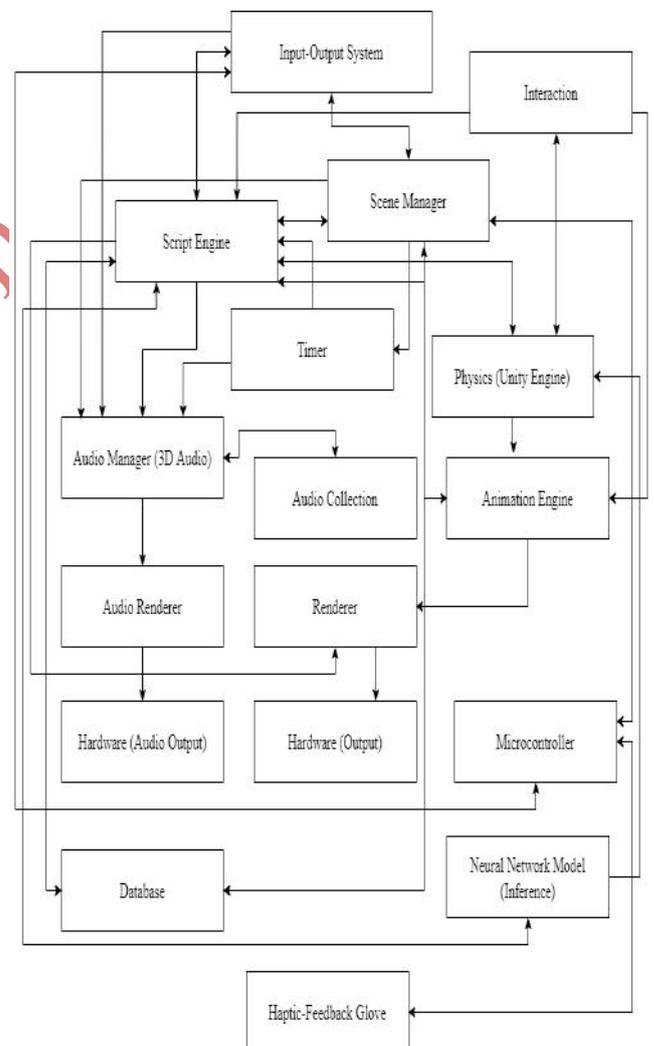


Fig [F]: High Level Block Diagram

3. Detailed Working

A) Virtual Training Environment:

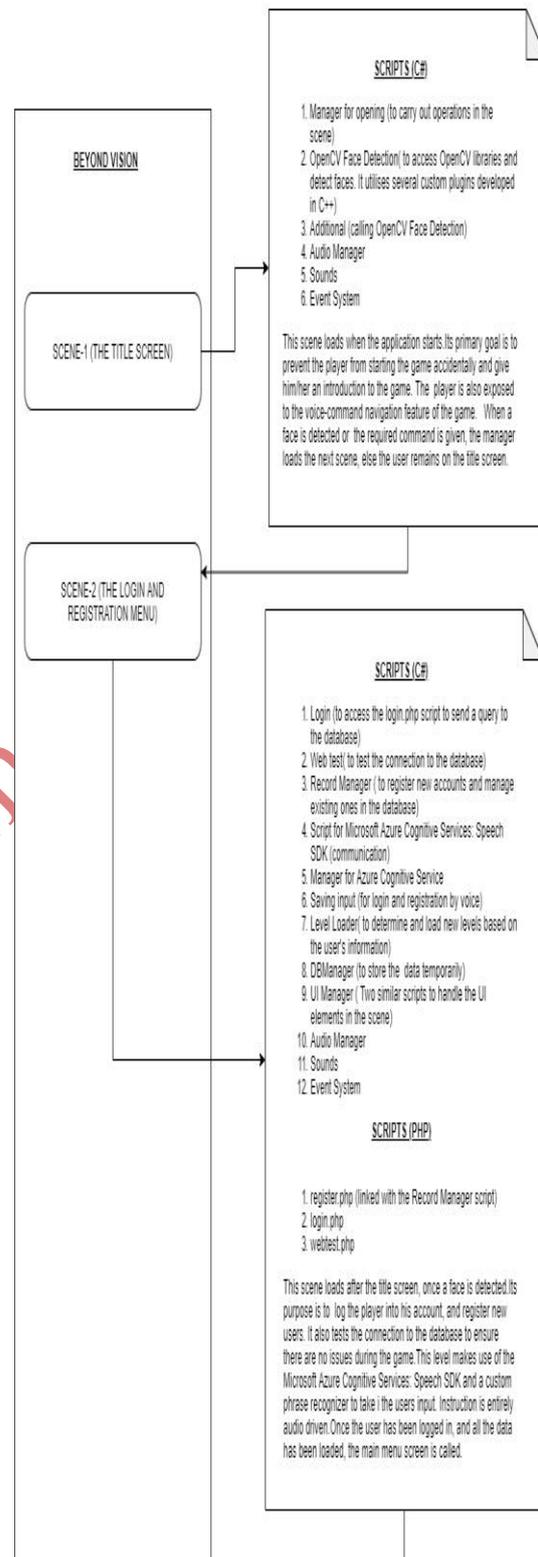
The system comprises two main components- the Haptic Feedback Glove and the Virtual Training Environment. The training environment and the glove work in sync, to serve the ultimate purpose of training the individual; each part plays a pivotal role.

The Virtual Training Environment, built in the Unity Engine[10], consists of the following primary elements:

1. The Scene Manager (for each scene)
2. The Script Engine (that determines and defines every action)
3. The Physics Engine (that defines all interactions)
4. The Animation Engine
5. The Timer (closely associated with the Script Engine)
6. The Input-Output subsystem (key part of the Script Engine)
7. The Audio Management subsystem
8. The Audio Collection for each Scene (linked with the management subsystem)
9. The Database (MySQL, that stores the user’s data and relays information to and from the Script Engine, through PHP Scripts associated with the engine)
10. The Assist Mode (the Neural Network Model that is enabled when selected, Inference only)
11. The Rendering Engine (for both audio and the graphical output)

A.1 Detailed Game-Context Diagram

Figure (G) shows the Game Context diagram, with all the scripts in use.



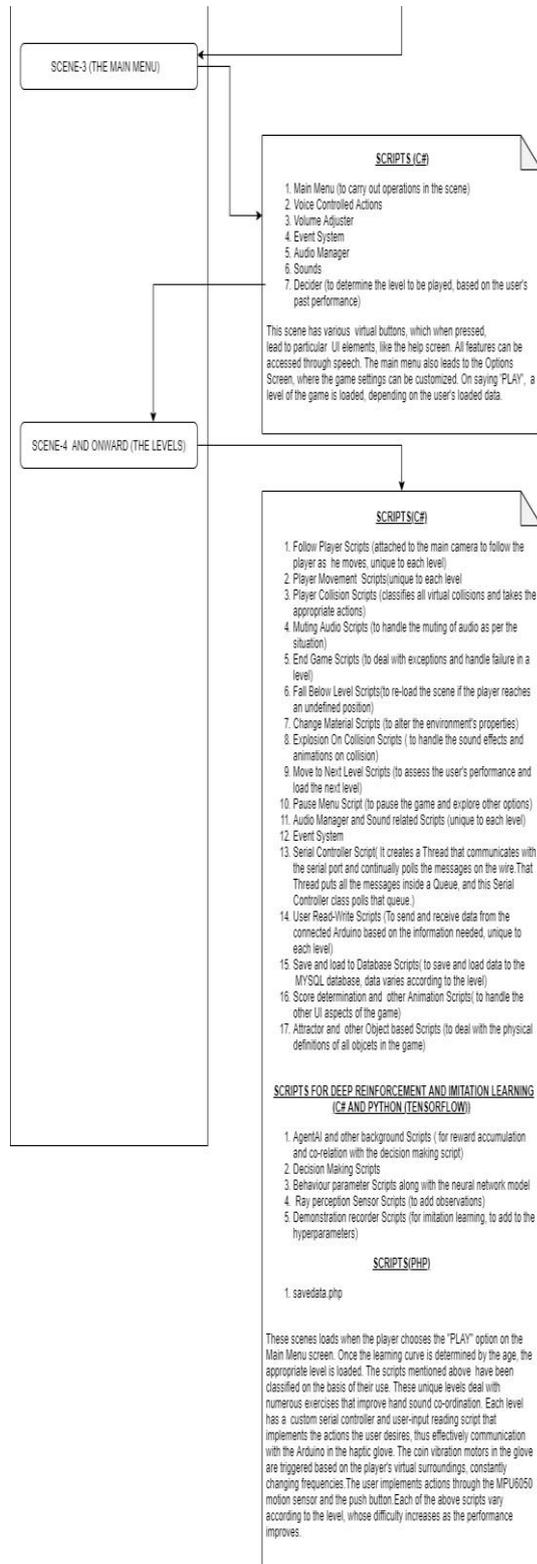


Fig [G]: Game-Context Diagram

In addition to the primary scripts mentioned above, several background scripts aid the functioning of the application. They are:

1. Additional Camera Data Scripts
 2. trainer_configuration.yaml (to train the Neural Network)
 3. Graphic Raycaster Script
- This application also uses custom plugins developed in Visual Studio (C++) for use in the Unity Engine.

The above mentioned scripts were written solely for the purpose of this project. The standard Unity packages utilized were:

1. Com.unity.barracuda
2. Com.unity.burst
3. Com.unity.cinemachine
4. Com.unity.render-pipelines.core
5. Com.unity.ide.rider
6. Com.unity.ext.nunit
7. Com.unity.mathematics
8. Com.unity.postprocessing
9. Com.unity.searcher
10. Com.unity.collab-proxy
11. Com.unity.render-pipelines.lightweight
12. Com.unity.render-pipelines.universal
13. Com.unity.test-framework
14. Com.unity.timeline
15. Com.unity.ugui
16. Com.unity.ide.vscode
17. Com.unity.ide.visualstudio
18. Com.unity.ml-agents

A.2] Explanation

Each scene in the application is entirely governed by the Scene Manager present in the scene. It coordinates the working of each element present in the scene to produce the desired result. When the application starts, the first scene to be loaded is the Title screen. The primary goal of this scene is to check whether the user has consciously activated the application or if it was an accidental start. This scene utilises a custom plugin, designed for this

very purpose, built in Visual Studio in C++ with CMake and existing OpenCV libraries. Imported into Unity as DLL (Dynamic Link Library) File, it performs face detection to determine the presence of a player. The camera stream and pixel processing is done within OpenCV, and only the properties of the detected faces are sent to Unity in real-time. The associated C# scripts receive and process the data to take the appropriate action.

The Audio Manager in this scene is responsible for the audio output as well as handling speech input. It achieves the latter through the combination of custom phrase recognition code and Windows Speech Services (when offline). When access to the Internet is enabled, the application heavily relies on the Microsoft Azure Cognitive Services: Speech SDK for Unity. If a face is detected, the management subsystem loads the next scene, else the application shuts down after 2 minutes

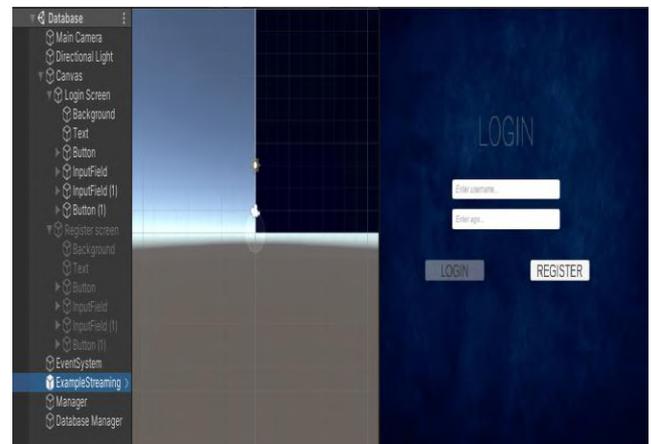


Fig[H]: Title Screen

The next scene is the Login and Registration Scene, where the user must either log into an existing account or create a new one. This scene, like the others, can be navigated entirely through speech. This scene can be broadly classified into two UI elements- the Login screen and the Registration screen.

The Registration screen employs the Microsoft Azure Cognitive Services: Speech SDK for accurate speech-recognition, which then communicates with the associated C# scripts, to record the data. After the connection to the database is tested, the data is relayed via a PHP

script and the Unity Web Request System to the database, that creates and initialises the account.



Fig[I]: Layout of the Login Screen in the editor

Once the account has been successfully created, the user is led back to the Login screen to log into the account with the two key parameters- the username and age of the user. This data is verified, and then based on it, the user's score and past performance is loaded. All the data being sent to and received by the database is processed by the Database Manager. The loaded information is assessed and analysed by the C# scripts written for this purpose and the appropriate decision is taken (the next level that suits the needs of the user is determined). The other elements in the scene include the Audio Manager, the collection of audio, the Azure: Speech to Text Manager and the Script that enables it's communication with the server

The Main Menu scene, which is displayed next, has multiple features that enable the user to manage the application. The 'Play' button, accesses the level determined previously. The 'Options' button presents the Options Menu, where the user can alter the basic settings of the game, to customize the experience. The 'Help' button leads the player to a scene, explaining the purpose and the controls of the game, with the Haptic Glove as well as speech, through a voice-over. This provides a detailed explanation on the working of the game and

exposes the player to the voice-control navigation feature available in the game. The last UI element present in the scene is the 'Quit' button that shuts down the application. This scene primarily serves as a 2D interface for the user to understand the functioning of the game and launch it. This scene includes scripts that deal with the speech-based actions, the Audio-Manager, and other trivial ones that alter specific settings.

As specified earlier, the 'Play' button loads the level previously determined, a 3D world that poses certain challenges that vary according to the difficulty and the time spent by the player in the virtual environment. The levels present sound-based cues to help the player navigate the world, thus heavily relying on the 3D audio to solve the level. Certain levels also implement Microsoft Azure Cognitive Services: LUIS (Language Understanding). The levels have been designed to engage the user, with constant audio feedback being provided based on the performance of the user. As the user plays the game, a score is recorded and constantly fed to the database to track the user's performance. The moment the user enters the world, he/she is provided a brief overview of the level. The user can then proceed to complete the challenges presented in a fixed time period. The user can also pause the game at any point to change the settings, quit or return to the Main Menu. The C# scripts associated with the levels deal with the following aspects:

1. Scripts associated with the functioning of the cameras in the scene- determine which camera to render from, and position the cameras relative to the movement and position of the player. The Audio Listener Script is attached to the Camera, thus even the slightest change in position can alter the audio output. The cameras utilise the Com.unity.cinemachine package to aid the transition of cameras as well as any change in direction.
2. Scripts associated with the player's movement and collisions with virtual obstacles and audio points- these scripts determine and implement player movement, and classify all collisions, relaying this information to the Scene Manager. They are closely linked to the camera scripts to coordinate the working of the camera with the player. These scripts also handle movement in undefined locations.

3. Scripts associated with the Audio Manager- to mute and play audio based on the events that occur
4. Scripts to handle failure or completion of a level- they determine the cause of failure and send this information to the audio manager. They also control the Event System in the case of completion of the level. The 'End Game' script in particular deals with the communicating the score and time to the Database Manager and reloading the level
5. Scripts to change materials and alter the environment's properties, when certain events are triggered
6. Scripts to handle the animation and physics related to the destruction and fragmentation of objects
7. Script that spawns and defines all the objects present in the scene
8. Script to determine the score and time taken- the score is constantly updated and sent to the Database Manager
9. Script to pause and resume the game- it also assists in displaying UI elements to explore other in-game options
10. Scripts to connect and relay information to and from the Haptic- Feedback Glove- to enable real-time, latency-free communication with the glove's Arduino Uno R3. These scripts have been built, utilizing the Serial Port library (Ardity) by Daniel Wilches for bi-directional communication with the Arduino[11]. These primarily include the following two scripts:

- 10.1. **The Serial Controller Script** – that creates a thread to communicate with the Serial Port and continually polls messages on the wire. The thread puts the messages in a queue and the Serial Controller polls the queue. It also makes use of the Ports library to access the COM Port.
- 10.2. **The User Read-Write Script**- This enables the user to send custom input and receive output, through the Arduino, based on the specifications of the level. It is closely linked to the Serial Controller for communication with the Arduino with the USB cable.

Apart from the scene-related Scripts mentioned above, certain Scripts also handle communication with the MYSQL database, that stores the

$$L^{CLIP}(\theta) = \hat{E}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t)]$$

information of all current users. The C# scripts that handle this communication include:

1. **Database Manager Script**- this script handles all the data sent and received from the

database, creating a temporary location to store the active user's data while the application is functioning

2. **Save and Load to Database Script-** this script loads the data to the MYSQL database, creating a permanent location for it. This script is closely associated with the Database Manager and the PHP script to send the Web Request. This C# script makes use of Unity Engine's Web Request System to access the database via the PHP script .

The PHP scripts linked with the entire application, enable access to the database, thus playing a crucial role in the functioning of the System:

1. Savedata.php
2. Register.php
3. Login.php
4. Webtest.php(to test the connection to the database)

If a level in the process of training proves to be too difficult, the user has the option to enable the 'Assist Mode'. In this mode, the standard movement script is replaced by a decision making script that makes use of the trained Neural Network model[12] (only for inference at this stage) to take the appropriate action .The Demonstration recorder Script was used to train the model, by recording manual actions for certain levels to increase the accuracy of the decisions made. The decision-making script also has a heuristic method that accommodates manual player input. The Ray-Perception Sensor script makes use of raycasts[13] to determine the relative distance between objects in the 3D environment, to keep track of the player's surroundings, adding all observations to an array that is ultimately passed to the decision requester , that takes the appropriate decision. The Timer, times the actions taken; these scripts supporting the neural network also keep track of the other elements in the scene, thus preventing any errors during the functioning of the mode. The Neural Network model was built and deployed using the specialized Unity package 'Com.unity.ml-agents', along with Tensorflow, for training the model, based on the observations passed and the actions taken by the decision-making script. The hyperparameters were tuned in the trainer_config.yaml file, before training, and the reward system was established. The accumulated reward was stored locally as plain-text file,

accessed during training.The final Neural Network model was deployed as a behaviour parameter for inference only. The script handling the working of Assist Mode, resets the level after the trained agent navigates through it, so that the player can attempt it, based on the hints given and the audio output received.

The Neural Network Model was trained using the Proximal Policy Optimization Algorithm (PPO)[14]. The PPO Algorithm developed by OpenAI, to replace the Deep -Q learning method. It involves collecting a small batch of experiences interacting with the environment and using that batch to update its decision-making policy. Once the policy is updated with this batch, the experiences are thrown away and a newer batch is collected with the newly updated policy.

Hence, it is called an "on-policy learning" approach. The new variant implements the following objective function:

- θ denotes the policy parameter
- $E^{\pi} r_t$ denotes the empirical expectation over timesteps
- r_t is the ratio of the probability under the new and old policies, respectively
- $A^{\pi} r_t$ is the estimated advantage at time t
- ϵ is a hypermatameter

Due to the fact that it has no experience buffer, it trains the model faster, for both continuous and discrete data. The Python API communicates between the external Python process and the game software.

A.3J Results

Approximately 97% of the children in the United States play games for an average of an hour a day[15]. Video games have now become a ubiquitous part of children's lives around the world. A vast majority of research on the effects of games has only been on its negative effects, leading to depression, violence and depression. However, a more balanced perspective has only recently begun to emerge, highlighting the possible benefits of playing video games in the areas of cognitive and social development. Video games have the potential to educate, in a manner, completely

different from the traditional approach. When strategic or educational games are played for a small, fixed period of time, they can have several positive effects on the individual[16]. Moreover, video games have the ability to offer extensive virtual environments to train and build on essential real-world skills.”Beyond Vision” relies on a similar approach and aims to develop navigation and spatial orientation skills, by building on the visually impaired user’s primary senses- hearing and touch. Figure(J) shows the visible improvement in individuals as a result of playing video games associated with learning and memory as a function of group.

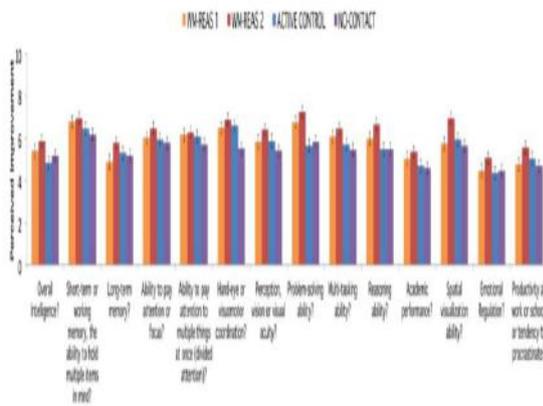


Fig.[J]:Improvement in cognitive skills as a result of playing strategic video games

The game is partitioned into numerous ‘levels’ that simulate real life situations.All the levels contain audio cues and incorporate tasks that must be successfully completed to move on to the next level. Certain levels also incorporate trained Reinforcement Learning agents, to provide an additional challenge. Each level is unique and trains the primary sensory skills of the visually impaired individual. The difficulty increases with each level; the learning curve is determined based on the age of the individual as well as the score attained in each attempt of the previous levels. The use of levels helps identify the key areas of improvement and also enables the user to explore various environments and situations during the process of training. Each time the user plays a level, the score and time taken for that attempt is recorded and stored in the MYSQL database, and this information is used to determine if the user

should move on to the next level. It also identifies the levels that should ideally be played next. This use of ‘levels’ in the application helps determine the existing skills of the user and build on the key skills required in the desired environment.

The difficulty of the next level to be played is determined by the system based on the user’s past performance. If any level in the course of training proves to be too difficult for the user, the game offers an Assist Mode. This mode implements an artificially intelligent agent that has been trained in the environment, based on its specific requirements. The user is provided with the audio output in real-time to determine the possible paths that can be taken. Constant audio feedback is provided, so that the user can successfully complete the level.As soon as the agent completes the level, it is reset, so that the user can ultimately implement the audio to improve his/her score. The user can use the Assist Mode only twice in each level. If the user fails after that, he is forced to play a different level of the same difficulty as the previous one. The use of Reinforcement Learning (the PPO algorithm) allows the use of the same network for variable situations, thus capturing all outcomes of a level. The neural network was trained using the PPO algorithm, to implement both discrete as well as continuous values. The results of the first training round are displayed below (from Tensorboard). Figure (K) shows the cumulative reward gathered by the agent in the environment.



Fig.[K]:Cumulative reward

Figure(L) shows the episode length.

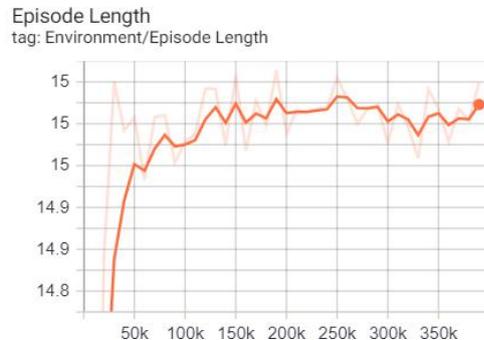


Fig.[L]:Episode Length

The above data pertains to a particular level of the application that requires the user to avoid randomly spawned moving obstacles. They are required to avoid these incoming obstacles to increase their score. Thus, the agents trained for each level have the potential to effectively navigate every possible situation and provide excellent examples on how the level should be attempted.

B] Haptic-Feedback Glove:

The Haptic-Feedback Glove serves as the principal form of input in this system. The only current alternative to O&M training is acoustic Virtual Reality, however, the price of such systems is extremely high, making them inaccessible to the masses. The areas in the world with the highest concentration of blind people fall in the poorest regions of Africa and Asia, thus, only a cost-efficient solution would prove to be useful. The Haptic-Feedback glove was designed to serve this very purpose.

The glove incorporates an Arduino Microcontroller, that serves as the principal component. The Arduino has been programmed with protothreads, to reduce latency and for efficient serial communication with the application in real time. This was truly a crucial aspect as any delay in communication can lead to a plethora of problems. The board is connected to multiple coin

vibration motors, along with a push button, and a 6-axis motion sensor. The MPU 6050 tracks the motion of the user's hand and relays the information to the Arduino, which sends a message through a USB cable to the application, to register the movement. This movement is implemented in the virtual world in real time, and the consequences of this action are determined along with the output that is relayed to the Arduino board. The vibration motors vibrate at changing frequencies, depending on the player's virtual surroundings and the current situation.

The user has two primary modes of input, that alter the message based on each level- the push button and the MPU6050 motion sensor. The role of the motion sensor is to send the precise change in the coordinates so that it can be scaled and implemented in the virtual environment. Once the direction is changed, the motion in that direction can be triggered using the push button. The use of these two, varies according to the necessities and difficulty of the level. The number of buttons and controls have been minimised to prevent any confusion, that may ultimately lead to wrong inputs from the user. The combination of these two mechanics, successfully capture the inputs required, successfully developing the skills required by the individual.

The vibration motors serve as an important form of tactile feedback, as in its absence, the only form of output is the audio. Tactile feedback plays an essential role in developing the user's skills, as it is what plays an extremely crucial role in daily movement. The main objective of this application is to train the Blind, so that they can navigate their physical environment with ease. The main mode by which they determine obstacles in their path is the 'White Cane'. The vibration motors exist to emulate that effect, by vibrating at changing frequencies based on the contact with virtual objects. Thus, the application not only provides the location of the obstacle, by exposing the audio point, but also provides tactile feedback to aid them in their motion in the virtual world. The vibration motors are four in number and are position on the ends of the sheaths that cover each finger,

excluding the thumb. They are connected to the Arduino board individually, to trace each endpoint. The Push button is placed on a small fragment of a Zero PCB board, that connects to the side of the index finger, so that the thumb can easily access the push button, like on a standard controller. The MPU6050 motion sensor has been positioned next to the Arduino, placed on the upper surface of the glove.

B.1] Schematic Diagram

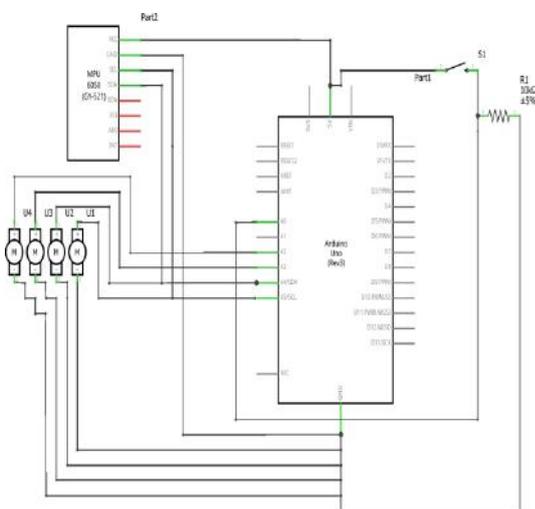


Fig.[M]:Schematic Diagram

Figure (M) shows the schematic diagram of the Haptic-Glove[17]. The Arduino connects to the game software via a USB A/B cable connected to the COM port.

B.2]Explanation

The Haptic-Glove can be connected to the COM port once the application starts. The MPU6050 only has to be calibrated for accurate information to be transferred. The Glove can then be used with the application. The user can access and start the level through speech, and the glove starts functioning as soon as the level loads. For any level in the application, the user can change directions with the MPU6050, and move forward in that direction,

relative to the position of the player, by pressing the push button down, which moves the player by a fixed number of units. The user is compelled to rely on audio to choose the direction as well as other actions based on the audio output given as a result of earlier decisions, resulting in a continuous cycle of processes.

To make the interactions in the virtual world feel as real as possible, every interaction is accompanied by a set of vibrations that are experienced through the coin vibrations motors present in the glove. The frequency of their vibrations are altered individually to emulate the experience of touch, as these motors are connected separately to the Arduino pins. The above mentioned inputs vary according to the level, thus offering a complete training experience to the user. The system has the ability to function through speech, in the absence of the glove but the haptic feedback glove can help emulate even real life experiences, providing an intuitive experience through tactile feedback. The glove would have no real impact if it weren't cost-effective, so the cost has been minimised by reducing the number of components without altering the true working of the glove. With just three components, excluding the Arduino, it serves as the perfect low-cost alternative to an acoustic VR system. Figure (N) shows the arrangement of the components on a working prototype of the Haptic-Feedback Glove.



Fig.[N]:Haptic-Feedback Glove

B.3] Integration of the Haptic-Feedback Glove with the Software

The most crucial aspect of the working of this system, is the communication between the Arduino and the game software installed on the computer. As the timing of the user’s performance assumes utmost significance in this application, any form of latency can cause a plethora of problems, and result in an undesirable final outcome. As true multithreading is impossible on an Arduino, the system uses a technique called proto-threading to make each component run as a separate process. It utilises a sophisticated proto-threading library for Arduino, thus effectively eliminating any form of latency. For communication with the software, as mentioned above, it incorporates bi-directional communication through a serial channel. This partly utilises Daniel Wilches’ networking library for Unity to process the incoming and outgoing data.e.

B.4]Results

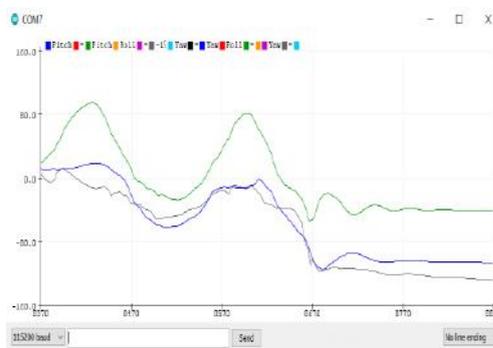
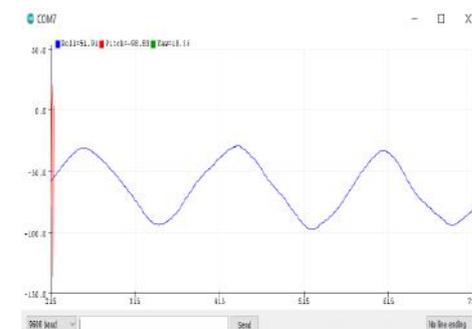


Fig.[O]: Change in roll,pitch and yaw with movement



Fig[P]: Change in pitch with uniform movement

Figure (P) shows the change in pitch as obtained by change in the direction of movement of the glove. This information is tracked by the MPU6050 on the glove. It also tracks the roll and yaw when these values are important in certain levels (as indicated by Figure(O)). The pitch plays the most important role as it determines change in the primary directions-Left and Right, which is required by a majority of the levels. When the glove is turned right, a trough is obtained and when it is turned left, a crest. The roll and yaw is also obtained in a similar manner, to be scaled. This information is crucial in determining the exact direction in which the player moves by scaling these values appropriately. The Arduino returns the direction as a single character as well as the exact pitch values obtained as a result of the Arduino code, so that the game software can process the information easily and apply it, without the slightest trace of latency.

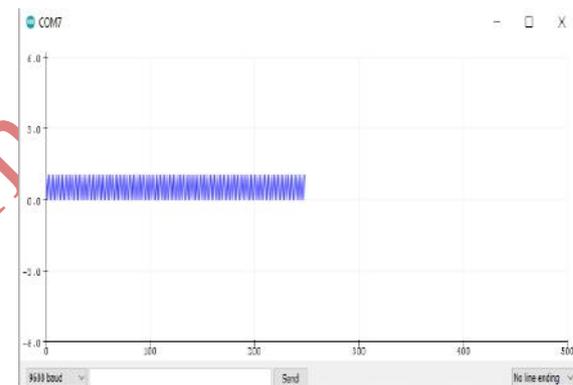


Fig.[Q]: Variation in input from push-button

Figure (Q) shows the variation in input from the push-button. The primary purpose of the push button is to relay a discrete value for an action to be undertaken. Its primary purpose is to either confirm movement or take discrete actions. In the figure above, the variable holding the value provided by the push button, varies between 1 and 0 to relay input for a fixed amount of time, as the button is pressed. This can be explained with an example of movement in a level. The MPU6050 changes the relative direction, but the input from the push button causes actual movement, by a fixed number of units based on the number of crests, or positive inputs(1).

Due to visual impairment, the other senses of the Blind are heightened. They are extremely sensitive to touch and change in external physical factors[18]. The vibration motors are attached to the gloves at the point right above the fingertips of the user. The vibrations vary according to the situation. For example, in a certain level, when the user approaches an obstacle, the frequency of vibrations increase to provide a realistic effect. The relative direction is determined by raycasts to send the signal to the appropriate vibration motor. According to Dr. Lotfi Merabet, the director of the Laboratory for Visual Neuroplasticity at Schepens Eye Research Institute of Massachusetts Eye and Ear, in blind people the brain rewires itself in a manner to use the information at its disposal so that it can interact with the environment in a more effective manner. Dr. Corinna Bauer, a scientist at the same institution, stated that as a result of visual impairment, the connection between other areas of the brain are strengthened, to compensate for loss of vision. Thus the tactile feedback provided serves as an excellent mechanism for them to develop their sense of touch[19].

4. Future Scope

Although the current version of the system works extremely well to suit the needs of the blind, it does not model real world locations. It only emulates the possible experiences in locations. Therefore, the further development of this application would undoubtedly involve mapping real world locations and environments into the virtual world. This would enable the user to train according to his/her needs and trace actual paths in a controlled environment. The existing glove can also be worked on to improve its cost-efficiency as well as design. By lowering the cost of production, the accessibility of the system will surely increase, benefitting the vast majority of the visually-impaired population. A potential solution to the real world mapping issue can be the use of the Google Maps SDK, that enables the development of vast open worlds. This can surely be implemented in the later stages of development. It can also be made wireless; the existing prototype uses a wired connection (USB- A to B cable) The primary goal of this system was to create a safe, controlled environment to train, primarily blind children,

which it achieves. It provides an easily accessible digital interface and a glove with simple controls, for tactile feedback. Other individuals who desire mobility training can also use this application to develop particular skills for use in daily life.

5. Conclusion

This system aims to provide a training environment for visually-impaired individuals to enhance their existing sensory abilities, as well as develop essential navigation and spatial orientation skills. It essentially comprises a gaming interface with 3D audio for spatial orientation, a database management and analysis system to assess and track the user's performance, and a tactile feedback mechanism, in the form of the Haptic-Feedback glove. The system takes the user's age to determine the learning curve, and stores the scores and time taken for each attempt alongside the username, to track the user's progress. It emulates real world situations, and provides challenging situations for the user to learn and grow. Constant audio feedback is also provided to guide and motivate the user. This application serves as an excellent alternative to traditional Orientation and Mobility training, as the training takes place in a controlled virtual environment, eliminating the risk to life. Moreover, it incorporates the element of 'fun' so that the user can enjoy while building essential skills. The Haptic-Feedback glove provides simple controls for the user to interact with the virtual environment, and solves the problem of cost-efficiency. It provides tactile feedback, based on the outcome of the situation, that serves as the most important form of output, for the blind, as they are forced to rely on their cognitive skills to excel. Constant audio feedback is provided along with 3D audio from audio points in the environment to determine relative position. The system also implements a combination of Deep Reinforcement Learning and Imitation learning to train multiple neural network models, that serve as the brain of agents in the system. These agents not only serve as varying factors that increase the difficulty level of the environment, but also assist the player. Agents that assist the player can be deployed through the Assist Mode that can be easily accessed by the player, if the level proves to be too tough. These agents have been trained to

perform discrete as well as continuous actions, based on the specific needs of the level. These agents train the player within the system itself, removing the need for human guidance. The levels can be curated by therapists and mobility specialists, to develop a training pattern for the user.

The levels offered by the game software develop essential skills through tasks or challenges, recording a score and time, that is eventually analysed to identify the key areas of improvement. The entire software can be accessed through speech due to the implementation of the Microsoft Azure Cognitive Services :Speech to Text SDK and the custom phrase recognizer. Thus, these elements successfully create a cohesive and highly accessible environment for the user to learn and grow.

This system also compensates for the shortage of mobility experts, thus negating the effects that this has on the visually-impaired population.

In conclusion, this system fulfills its original purpose. "Beyond Vision" has the potential to have a major impact on the lives of children and other individuals coping with blindness, who struggle with even daily tasks. It defines a low-cost, accessible solution that can improve the lives of countless untreatable blind children in the world.

6. References

1. World Health Organization (2020, 8 October). Blindness and Vision Impairment <https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment>
2. Bourne R, Steinmetz J, Flaxman S, et al., Trends in prevalence of blindness and distance and near vision impairment over 30 years: an analysis for the Global Burden of Disease Study. *Lancet Glob Health*. 2020. <https://www.iapb.org/learn/vision-atlas/#:~:text=1.1%20billion%20people%20live%20with,%2D%20and%20middle%2Dincome%20countries.>
3. Zhang, Wen. (2017). Surround by Sound: A Review of Spatial Audio Recording and Reproduction. *Applied Sciences*. 7. 10.3390/app7050532. https://www.researchgate.net/publication/317036523_Surround_by_Sound_A_Review_of_Spatial_Audio_Recording_and_Reproduction
4. Kammoun, Slim & Jouffrais, Christophe & Guerreiro, Tiago & Nicolau, Hugo & Jorge, Joaquim. (2012). Guiding Blind People with Haptic Feedback. https://www.researchgate.net/publication/235005436_Guiding_Blind_People_with_Haptic_Feedback
5. National Federation of the Blind (Minnesota)(2017)Regarding the Alarming Shortage of Qualified Teachers of Blind Students and Orientation and Mobility Professionals. <https://www.nfbmn.org/about/resolutions/A2017-04>
6. An Introduction to Deep Reinforcement Learning (2018, 3 December). Authors: Vincent Francois-Lavet, Peter Henderson, Riashat Islam, Marc G. Bellemare, Joelle Pineau <https://arxiv.org/abs/1811.12560>
7. Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. 2017. Imitation Learning: A Survey of Learning Methods. *ACM Comput. Surv.* 50, 2, Article 21 (June 2017), 35 pages. DOI: <https://doi.org/10.1145/3054912>
8. Arduino(2020) <https://www.arduino.cc/>
9. Jafari N, Adams KD, Tavakoli M. Haptics to improve task performance in people with disabilities: A review of previous studies and a guide to future research with children with disabilities. *Journal of Rehabilitation and Assistive Technologies Engineering*. January 2016. doi:10.1177/2055668316668147
10. Unity Technologies(2020) <https://unity.com/>
11. Ardity(Daniel Wilches) <https://ardity.dwilches.com/>
12. Mikulcik, Simon D., "Application of Neural Networks for Intelligent Video Game Character Artificial Intelligences" (2016). Honors Theses. 367. https://encompass.eku.edu/honors_theses/367
13. Unity Documentation(2020) Scripting API: Physics.Raycast <https://docs.unity3d.com/2020.2/Documentation/ScriptReference/Physics.Raycast.html>
14. Proximal Policy Optimization (2017, 20 July) Authors:John Schulman, Oleg Klimov, Filip Wolski, Prafulla Dhariwal, Alec Radford <https://openai.com/blog/openai-baselines-ppo/>
15. The Benefits of Playing Video Games(2014). Authors: Isabela Granic, Adam

Lobel, and Rutger C. M. E. Engels-Radboud University Nijmegen

<https://www.apa.org/pubs/journals/release/s/amp-a0034857.pdf>

16. Virtual Environmental Enrichment through Video Games Improves Hippocampal-Associated Memory. Authors: Gregory D. Clemenson, Craig E.L. Stark. Journal of Neuroscience 9 December 2015, 35 (49) 16116-16125; DOI: 10.1523/JNEUROSCI.2580-15.2015

<https://www.jneurosci.org/content/35/49/16116/tab-article-info>

17. Fritzing(v 0.94)(2019, 1 November) <https://fritzing.org/>

18. Comparative Study of Touch Perception in Normal and Blind People Dr.Aishwaryah Ravisankar., Dr.M.P.Brundha (2016)

<https://www.jpsr.pharmainfo.in/Documents/Volumes/vol8Issue11/jpsr08111608.pdf>

19. Brain Rewires Itself to Enhance Other Senses in Blind People(2017, 22 March)

<https://neurosciencenews.com/brain-rewiring-senses-blind-6276/>Source: *Massachusetts Eye and Ear Infirmary.*

7. Acknowledgements

I would like to express my gratitude towards my parents, who helped me finalize this project within the limited time frame, and my mentor, Mrs.Reetu Jain, who guided me throughout the development of this project. I would also like to thank On My Own Technology (OMOTEC) for providing me with the opportunity to work on this project. It allowed me to research numerous topics, through which I gained immense knowledge; I am extremely grateful to them.

8. Authors

First Author:



Abhishek Pillai

Cleared the Indian Certificate Of Secondary Education Examination (2020).Currently pursuing the Higher Secondary School Certificate from PACE

Junior Science College, Andheri.

Email ID:abhishekpillai94@gmail.com

Second Author:



Mohan Kshirsagar

Completed Bachelor's Degree in Electronics and Telecommunication Engineering from Sinhgad Institute of Technology Lonavala, 3 Years of

Experience in the field of Robotics and STEM Education. Currently working with On My Own Technology Private Limited as a Senior Branch Manager. Email ID: k94mak@gmail.com

Third Author:

Reetu jain



Chief-Mentor and Founder of On My Own Technology Private Limited Mumbai, reetu.jain@onmyowntechnology.com

Fourth Author:

Shekhar jain



Chief Executive Officer and Co-founder of On My Own Technology Private Limited Mumbai, shekhar.jain@onmyowntechnology.com

[om](http://www.onmyowntechnology.com)