Preface

The De Vinci Innovation Center (DVIC) is a community of makers that develops technologies within philosophical and critical frameworks to shape our societies' futures. The objective is to implement real-world solutions as well as design projects to enhance public engagement, improve education, and overall provide scientific knowledge. Our researchers contribute actively to top-level international research in multiple fields, including artificial intelligence, human-computer interactions, education, and ecology. We believe that these objectives require a transdisciplinary approach, that bridges the gap between sciences, techniques, sociology, and philosophy. This is performed by collaborating with other scientists and industrial and startup sharing our values, to form strong research partnerships...

The Artificial Lives group, led by Dr. Clement Duhart, aims to develop the next generation of machines and Human-Machine Interfaces. The group members strongly believe that through the combination of Design and Engineering, human-centered technologies can blend into our environments to become invisible, vastly improving daily lives. To achieve this vision, the members contribute to human-computer interactions, cognitive enhancement through new forms of extended intelligence, learning platforms, and cobotic. Our bio-inspired, multidisciplinary approach couples AI and virtual reality with intelligent materials, robotics and the Internet of Things.

For the past two years, De Vinci Innovation Center (DVIC) students following the Creative Technologies curriculum had the opportunity to develop their vision on technology, innovation, and society. This proceeding is a composition of six master's theses, ranging from Machine Learning, Human-Computer-Interaction to Robotics. The authors strongly believe that developing alternative futures requires new types of engineering that take into consideration both the people's needs and the environment. These documents have been written to reflect this vision and refined over several months with an iterative reviewing supervised by the Principal Investigators.

The Authors, the Principal Investigators and the whole DVIC community is proud of releasing this first proceeding. We dedicate this first edition to Pascal Brouaye and Nelly Rouyres, without whom nothing would have been possible.

DE VINCI INNOVATION CENTER

Artificial Lives

DE VINCI HIGHER EDUCATION

List of Theses

Adrien LEFEVRE -

Adrien: Gamification of Interfaces for Learning and Performance Enhancement

Adrien LEFEVRE

Contents

Introduction 1

1.1 Context

The learning-performance distinction is a concept in behaviorism that stresses the difference between the learning of a behavior and actual performance of the behavior. Learning is a change in the ability and potential to do when the performance consists of an execution of the learned behavior. [1]

This distinction is significant in subjects involving physical movements attached to something else, an idea, image, or sound, for example, in music or language learning.

The relative persistence of learning is sometimes referred to as an enhanced capacity for motor skill performance. [1]

These subjects are traditionally challenging to learn because integration with performance takes much time. Acquiring expertise in the practice of a discipline through repetition can be laborious. People get bored. With training, the learner can perfect their ability to link an idea, a wished sign, a sound, a mental image with a movement, or a "physical" sound. Performance psychology is the scientific field describing the human ability to translate mental concepts into physical or musical practice.

1.2 Field of research

The domain of gamification looks at how to make people less bored when learning subjects by introducing playful mechanisms [2]. In recent years gamification has revolutionized the field of professional training. It has accompanied the digital transformation of training and modernized existing modules. This principle of gamification has brought real advantages to learning

mechanisms by combining pleasure and skill acquisition.

The gamification of training corresponds to a set of playful mechanisms designed to "gamify" learning content to personalize the relationship with training. This technique improves learner engagement by arousing their interest. Being immersed in an enjoyable educational experience enrich the memorization process thanks to the emotional trigger provided by the game. The playfulness of training will encourage positive emotions that lead to the improvement of learning. The effect is engagement and motivation improvement.

Another way to improve engagement is through activating different senses. In the HCI field, Augmented Reality and Tangible Interfaces make digital information more immersive by projecting it into the real world, engaging sight, sound, and kinesthesia [3].

The Augmented Reality (AR) experience is thriving as a significant trend. Around 2.4 billion people use AR on their mobile worldwide in 2023. AR can augment computer-generated graphics into the natural environment on screen. This augmentation can serve gamification, improving the education system efficacity and making students' attitudes more positive. It makes learning interesting, fun, and effortless, improving collaboration and capabilities.

These paradigms make the link between abstract information and the body more legible. HCI optimizes the symbiosis between user and technology (Human-Computer Confluence) or how the elements of the human ecosystem cooperate to optimize their interaction with humans. Communication Technology (ICT) can be based on radically new forms of sensing, perception, interaction, and understanding [4]. The particularity of digital learning environments lies in the fact that they can accommodate diverse users' needs [5].

According to Howard Gardner, there are 8 forms of intelligence, each of which has certain preferential strengths. These intelligences are: logical-mathematical, verbal-

linguistic, musical-rhythmic, bodily-kinesthetic, visual-spatial, interpersonal, intrapersonal and naturalist-ecological. Multiple intelligences are often used to identify the profiles and intelligences of students, to offer them appropriate support. The use of multiple intelligences in learning or practice allows for the stimulation of different areas of the brain and thus promotes and optimizes the retention of conscious and unconscious information in many ways.

This thesis takes inspiration from work in gamification, performance psychology, and theory of multiple intelligences linked to AR/Tangibles. This document reflects on how to learn and perform by exploiting these different domains.

1.3 Approach

We have thus carried out several projects within the framework of studying a user's ability to learn and perform with the help of an augmented or tangible interface.

These projects aim to exploit specific human ways of interacting to promote using different intelligences, thus the retention of information and the ability to practice efficiently.

The devices that users interact with use augmented reality, sound, visual feedback, haptics, and enhanced features through electronics or software to help motivate them.

1.4 Contributions

The areas studied are: learning the basics of music theory using an interactive electronic score, practicing music through augmented reality learning applications, and learning and practicing sign language through a video game and an AR training application.

Interactive Musical Score 2

2.1 Introduction

Reading music from the score is essential to Western classical music training. Traditionally, children learn the different musical notes by singing or playing notes on an instrument guided by a teacher. We envision a way for children to learn the correspondence between notation and sound by directly touching the score. The Interactive Score is effortless and allows children to make discoveries independently. The correspondence between the visual, the tactile, and the sound can aid in learning.

In this work, we introduce the Interactive Score, a novel instrumental device for children's solfege learning. Paper scores lay onto a staff drawn with conductive ink and connected to an Adafruit musical box. Pressing a note in the score triggers its sound, and running fingers over the notes play a melody.

Motivation

2.2 Related work

2.2.1 Context

A review of music theory pedagogy over the past decade reveals many criticisms about music theory courses. Other concerns include taking a harmonic, melodic, or compositional approach to teaching theory. The early involvement of students in creative thinking about harmony, melody, and rhythm partially determines their success in the academic program [6]. Developing a feeling for the mastery of the preliminaries to music and the presentation of the basics are the prerequisites for a successful future theoretical education.

However, out of the large number of students who embark on learning music, only a few retain the courage and motivation to continue their studies to the end. For example, out of 100 French people aged 15 and over, a study recorded that only 30% of the musicians who have learned music continue practicing their instrument during their lives [7].

It is pointless and frustrating for students to be pushed too quickly into advanced and sophisticated theory without musically visualizing what they are studying.

2.2.2 Approaches

Music theory courses can take many forms. There are many approaches to developing musicianship skills. Teachers often choose between traditional or more contemporary approaches.

Traditional approach Traditional music lessons prepare students to read, write and perform music from the work of great academic composers. It produces excellent results in terms of sight-reading skills. However, many students leave these music studies because they need more enjoyment of the method.

Contemporary approach Contemporary music lessons are paired with instrumental lessons (usually piano or guitar). They do not require music reading or theory. These lessons focus on skill, musicality, and more intuitive learning methods. They require the student to have a good ear for music and a sense of rhythm. The contemporary approach benefits students with difficulty visualizing with theory classes. They can play real pieces only a few months after starting their lessons.

The primary advice if a student has any problem understanding music theory is to study with a piano. Intervals and harmonies are easier to understand on a piano first, as is all the rest of the theory.

2.2.3 Musical Mental Projection

Musical imagery, or the ability to create an image of sound in our minds, is an essential skill for all musicians. For example, brass, winds, strings, and singers imagine the pitch of an upcoming note to make it easier to play it and determine the distance from the previous note [8]. Composers and arrangers also use musical imagery when creating a new piece. Musical imagery training improves the ability to follow the upward and downward movements of the tonal contour of a musical phrase or imagined tune [9].

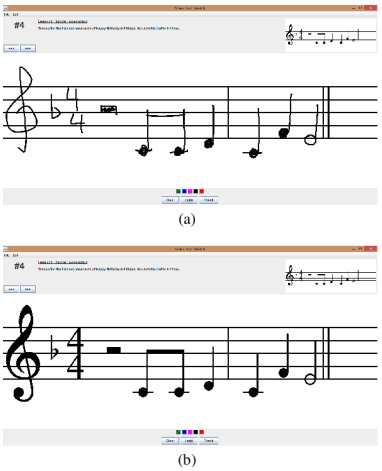
Ear training" (or solfege) has traditionally been part of the curriculum of most music schools. An essential part of solfege is the ability to read music notation and imagine how it is supposed to sound. We are interested in teaching this skill to children.

2.2.4 Tangible Interactive Media for Music Practise

Many projects aim at getting a child involved in the world of music. However, only some consist of tangible interactive media for music discovery.

For example, Zigelbaum et al. investigated how electronic instruments can engage young learners in learning to make music. Their project was the development of different tools involving movement, linking it to a sound. They created a trampoline, an interactive matrix, or musical bracelets [10].

Xiao Xiao et Al. propose an understanding of the essential workings of music without going into the details of music theory [11]. The authors expose a new technique for visualizing musical motion on a piano keyboard. The technique, called Andante, uses walking figures that move along the keyboard to represent the movement of musical phrases. The results of their user test showed that the participants found the Andante animation to be significantly more informative and engaging than the video without animation.



(a) (b)

Figure 2.1: Maestoso Educational Sketching Tool for Learning Music Theory

The work of Taele et al. [12] 2.1 describes the practical and cognitive benefits of learning music theory for both musicians and non-musicians. The paper proposes an intelligent educational tool to help students learn music theory. The tool, called Maestoso, utilizes sketch-based interaction and machine-learning techniques to provide personalized feedback to the user. The paper first introduces the importance of music education and the challenges students face when learning music theory, such as the abstract nature of music concepts and the difficulty of translating musical ideas into notation. Maestoso is a sketching interface that allows users to draw musical notes, chords, and melodies using a stylus or finger. The program uses machine learning algorithms to recognize the user's sketches and provide feedback on their accuracy and completeness.

Amico et al. discuss the development of a new type of MIDI controller designed for music education [13]. The device, called Kibo, is a tangible user interface that allows students to interact with music physically and intuitively. The authors introduce the concept of tangible user interfaces and explain how they can be used to create more immersive and interactive learning experiences. The device includes a set of modular blocks that the user can rearrange and customize to create different musical experiences.

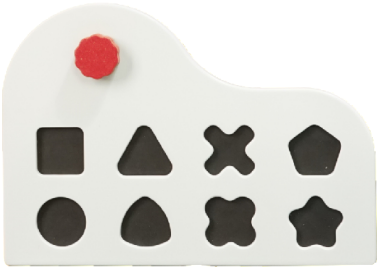


Figure 1: The interface of Kibo.

Figure 2.2: The interface of Kibo

Implementing such systems is often fully digital and interactive through a screen. Some projects aim to teach children music theory or interact with notes to compose, learn and experiment. Most of these projects are applications that users can download on smartphones or tablets. The relationship with the tangible paper score is gradually lost, and digital interaction replaces it more and more.

One solution is to use conductive ink to keep a tool in paper form without losing its interactive aspect. Conductive inks, paints, and varnishes are liquids containing metal particles, conductive polymers, or graphite. They have the specific ability to conduct electricity.

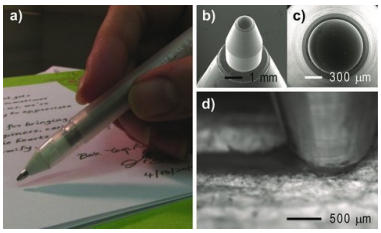


Figure 2.3: Pen-on-Paper Flexible Electronics. a) Optical image of a rollerball pen loaded with conductive silver ink. b) and c) side and top views of the rollerball pen. d) Optical image of the rollerball pen tip writing a conductive silver track

Inkjet printing allows to pattern of organic semiconductors [14], metal contacts on organic semiconductors [15] [16], and metallic structures that require minimal further processing. Researchers such as Ahn et al. used conductive ink printing to realize metallic connections between functional components of flexible devices.

Another project like that of Russo et al. [17] has resulted in an optical image of a flexible paper display containing a LED array **??**. The prototype is a multi-color 25 × 16 LED array connected to the printed silver electrodes by depositing a drop of concentrated silver ink.

2.3 General Architecture

2.3.1 Overview

Our design augments a traditional paper score in many digital music learning applications on screen-based devices. Children already spend a considerable amount of time in front of screens, which can harm their eyes from a young age. Paper is flexible, lightweight, and easily transportable, and incorporating electronic circuits in the paper has shown its attractiveness to children [18].

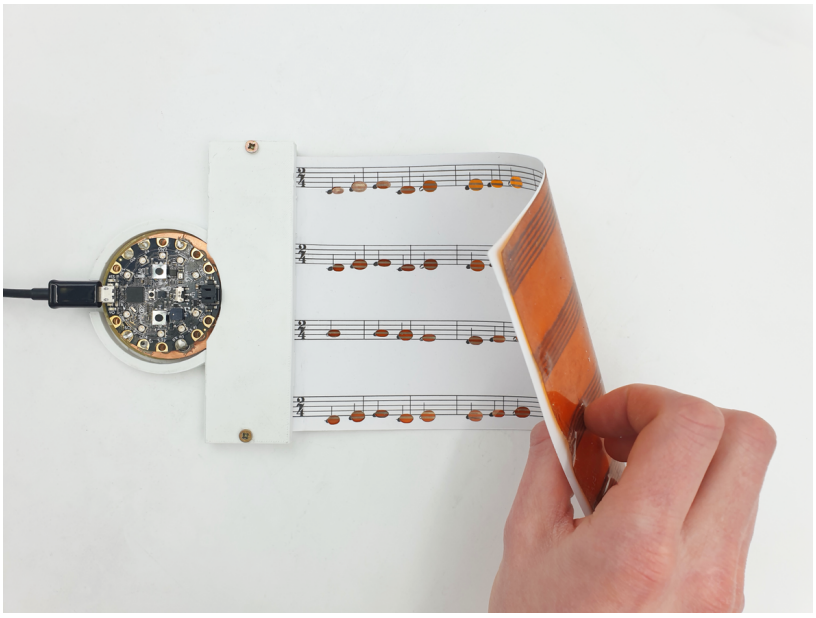


Figure 2.4: Interactive Score Prototype

The project does not aim to teach a user how to play

music. It seeks to link music theory directly with music practice without requiring knowledge.

The user has to supply the electronic part (substrate and PCB), then he places a partition (a cardboard stencil) on top of the substrate (where the conductive lines are located) 2.4. He can play the music and change it to another one.

2.3.2 System Design

The Interactive score consists of two thin layers. The first layer is the traditional sheet music, printed on cardstock paper, with holes punched for each note. Under this sheet is a polyimide substrate with conductive lines printed on it 2.5.

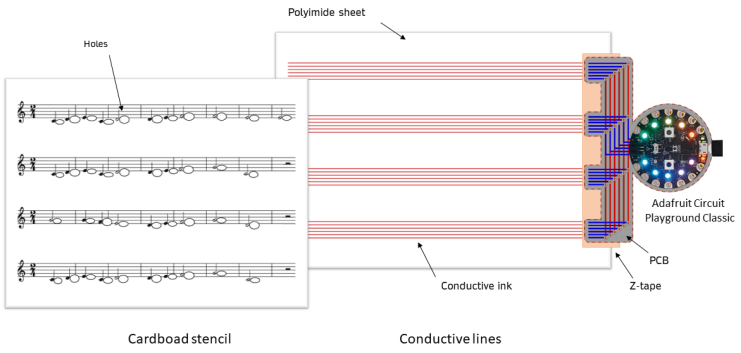


Figure 2.5: Interactive Musical Score architecture.

The conductive lines are connected to an Adafruit Circuit Playground printed circuit board (PCB) using a double-sided "z-tape".

When the user touches a note on the top layer, the finger makes contact with the conductive lines through the holes in the cardstock. The signal travels through the ink paths and the z-tape to the PCB, which detects a potential difference using capacitive touch and plays the relevant note. Detecting several simultaneous signals on multiple pins allows playing eleven different notes with only six lines 2.5.

The system is kept in a 3D printed case, maintaining contact between the polyimide, the z tape, and the PCB.

The user can easily open and close the case to change the music.

2.3.3 Electronic Music Box

The signal is recovered and used in capacitive touch with an Adafruit Circuit Playground Classic 2.6. An Arduino program allows for generating a vast number of different notes. The code is retrievable on GitHub [19].

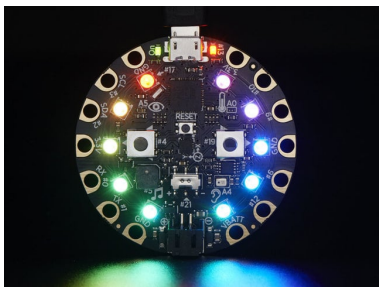


Figure 2.6: Adafruit Circuit Playground Classic

About the sound, the Adafruit Circuit Playground embarks a built-in buzzer. The implemented Mini Speaker is the SQMS5002S4036A. This device is a miniature magnetic speaker not adapted for playing detailed audio but for beeping, buzzing, and simple bleepy tunes. It is also possible to change the tone by changing the name of a variable in the code on the microcontroller.

The Adafruit circuit playground has ten mini NeoPixels of all colors, which can be animated with light when a user plays a note. The controller includes motion, temperature, light, sound sensors, a switch, and a mini speaker. This device is ideally suited for use on the score and integrates many features.

2.3.4 Paper Score Manufacturing

Notes, staves, lines, hyphenation, and cutouts were all placed on the same project for perfect dimensional compatibility between the different elements of the score. It allows the elements to fit together perfectly and export the cutout locations at the correct size relative to the partition's rest.

An inkjet printer draws all cardboard lines, hyphens, notes, and numbers.

A score of "J'ai du bon tabac" (on the left) lies on the substrate. Then, the notes were isolated in another PNG file, selected on Cricut design, and cut directly into the cardboard with Cricut maker 3.

2.3.5 Conductive Sheet Manufacturing

The conductive ink paths are 1mm thick and 5cm long, with a resistance of 0.07 Ohms. A simple inkjet printer equipped to print with silver nanoparticles conductive ink drew the lines directly on the substrate. A hoven sintered the printed patterns at 180°C for 73 minutes. This process allows the quick production of flexible circuits [15].

Figure 2.7: Interactive Score Conductive Sheet

This prototype comprises four staffs of 6 lines made of conductive ink on a Kapton polyimide sheet substrate. This polymer does not melt at high temperatures and has excellent mechanical strength, and is very dimensionally stable and creep resistant at temperatures above 260°C. The lines are 1mm thick, and at a distance of 15cm in length, it gave a resistance of 0.07 Ohms.

An Epson WF-2010 printer printed the lines on the substrate [19].

Conductive ink allows the score to be flexible, like an actual musical score. The use of ink rather than flexible brass strips has the advantage of quickly printing interactive scores on an industrial scale. The material needed is only simple printers and ink and cleaning products for the printers. This process facilitates and accelerates production possibilities.

2.3.6 Integration and Usability

Ability to change the score

Its ease of interchangeability and production characterizes the paper score. The user can easily remove the paper sheet from the box and place another to play a different melody. All the electronic parts (substrate, PCB, microcontroller) are independent of the paper score. The sheet music has the exact dimensions of the substrate. Therefore, placing the two precisely on each other to align the holes with the ink lines is simple.

Ability to improvise

The user can improvise by not playing the notes in the same order. The project allows a great deal of modularity in its use. Just by touching specific notes at certain times, users can experiment with different rhythms and melodies and reconstruct a piece from a few notes. With a simple score, including an ascending scale, he can try his hand at composition. It is impossible to create dissonance as the Circuit Playground plays melodies in a fixed tonality set in the code. Unfortunately, this fact also limits the improvisation capacity.

2.4 Applications and Evaluation

2.4.1 Set up

Four users interact with the prototype already prepared for use. They then answer several questions. The project is plugged in and set up as a base for further use. Therefore, measuring the openness and visualization of music theory given by the project to the user is interesting. The questions concern understanding use, practicality, interest, innovation, attractiveness, and playfulness.

2.4.2 Results

The users did not encounter any particular problems during the test. The prototype worked well. The testers gave interesting feedback. The majority of the comments were positive, as the project considered most of their comments since a previous test.

The users proposed many ideas to make the project evolve. They advised adding indications, potentially with LEDs, to indicate actions to be carried out. One idea was to create a binder with different stencils inside. They would have appreciated a correction of the electronics, which would warn them if they made a mistake. The testers liked that the person playing has to press the right notes, unlike some music toys that automatically correct the sound to play the right notes wherever the user presses. The users were also very interested in the fact that the project is portable. They would have liked to be able to take it with them to play music everywhere.

2.5 Discussion

2.5.1 Attractivity

The project is attractive for children as a tangible object. Its gamified aspect makes it close to a toy. This aspect contributes to its attractiveness, especially to young children. Its musical property favors the user's creativity in addition to being portable, tactile, fun, and easy to use.

2.5.2 Musical Imagery learning

An essential aspect of musical practice is the ability of the musician to decipher a score. When this one practices with visual support, his brain must take the exercise to connect the note, which he links visually with the position of his fingers on the instrument. The transcription of the note read is an intermediate the brain uses, deducing a position and calculating an interval

from it. The ability to translate a visual note into a sound is a reflex to be trained and very relevant in the practice of music. The basics of this ability must be acquired very quickly when learning the basics of music.

2.5.3 Mobilising multiple intelligences

This kind of device has a genuine interest in terms of learning. The user practices bodily intelligence with touch, visual intelligence since he has a support in front of him, and musical-rhythmic intelligence.

2.6 Conclusion

In conclusion, the interactive score project has a bright future, as it implements a very recent technology to popularize it. This process could be industrialized and even replicable for individuals with some improvements and optimizations. In the future, our goal is to make the score more accessible. Features such as flexibility and a simple "plug and play" aspect make it attractive even to children.

2.6.1 Limitations

The project has some limitations being a prototype. The device is limited in terms of the number of playable notes. The conductive sheet has six lines of conductive ink, which allows for playing 13 different notes between C5 and F6. It is not possible to generate alterations while playing. This fact means that if the playground circuit is set in a specific key, it is impossible to play a note with an alteration not present in the score's framework. The key must also be changed manually at each score change (paper sheet).

The major problem of the prototype is in its transmission of electrical signals. The most crucial problem of the project is its sensitivity to the electric field due to the

use of capacitive touch. Near electric fields, the microcontroller can detect false contacts. The device must also be connected to a power outlet so that the potential difference calculated between the user's finger and the ground is remarkable. Otherwise, the playground circuit may not detect the touch of a line or start playing by itself because of the detection of surrounding electric fields. A user should therefore keep the device at a distance from electronic devices and metal surfaces.

Z-tape transmits the signal from the conductive sheet to the PCB (inside the box). This conductive tape can be damaged and cause false contacts after many uses and transport. The tape can not transmit the current to the lines, and some notes stop working.

Another problem is the conductive ink which can crumble, preventing the current from crossing certain lines. The ink traces are indeed quite fragile and easily damaged. The ink is only dried on the substrate, so its adhesion can weaken. As the user has to run his finger over the ink, he can also remove thin layers, thus causing some lines to break.

The prototype also presents a difficulty in overlaying the sheet paper on top of the polyimide sheet to align the notes with the ink lines while keeping a flexible system. The case was designed for this purpose but could be more effective.

The device also needs to be improved in terms of sound quality. The speaker driver circuitry is an on/off transistor, so the device can only play square waves. The device's loudest frequency is around 4 KHz. The sound generated is shrill, very electronic, and of poor quality. It can thus slow down the desire to practice and does not resemble the sound of an instrument.

2.6.2 Future works

Adding the play of an alteration with three "buttons" usable thanks to the capacitive touch: flat, sharp, and natural, which the user can trigger with the index finger

of the left hand, is an idea to answer the problem of changing the key. These buttons will allow the user to discover the notion of these three tools and their meaning. They would be an interesting tool that would add an improvisation capacity to the system.

A "musical tutorial" should be added so the user can listen to the score's music before playing it. The tutorial would allow the user to assimilate the musical rhythm (the time between playing each note) with the physical rhythm (the time between pressing notes).

The electronic PCB/microcontroller part should also be redesigned to no longer integrate an Adafruit Circuit Playground but a much smaller, handmade circuit. It would be possible to improve the speaker's quality and connect the device to Bluetooth or Wi-Fi to play music at a distance. The next steps are also about looking for partnerships in children's education to research experimentations on the impact of this interactive score on music assimilation. Several parameters would be evaluated, such as concentration level, playing time, and knowledge retention. It is necessary to consider different strategies to transcript musical-rhythmic on this interactive score.

Sign Language Learning Game in AR

3

3.1 Introduction

Sign Language (SL) is the primary communication tool for deaf or hard-of-hearing people. It is a visual language that uses movements to provide people communication with the world.

Everyone does not understand SL because its practice requires a significant learning investment, forming a communication gap between the mute and the able people. According to the World Health Organization (WHO) report, the number of people affected by hearing disability in 2020 was 466 million, of whose 34 million are children [20]. Over 900 million people will have this disability by 2050.

In France in 2020, there were approximately 4 and 5 million deaf or hard-of-hearing people who have difficulties or cannot communicate through speech. Concerning the deaf speakers of French Sign Language (LSF), the figures are uncertain: they oscillate between 80,000 and 120,000, depending on the sources.

This work introduces a short video game to teach sign language. The application is intended to work on the DVIC Interactive Mirror but is fully functional on a simple PC.

This report deals with implementing an AI for sign language recognition (SLR) using the Mediapipe framework to extract the user's coordinates and analyze them through a model in Pytorch. The outputted model is then implemented in a video game engine to create a visual novel video game. The application is adapted to an augmented mirror to allow the user to play it in augmented reality. The game stimulates the player's motivation to favor his learning.

3.1.1 GOSAI for Augmented Mirroir

GOSAI

GOSAI is a new framework to help the development of augmented interfaces on the computer with a display. This framework targets all developers, from beginners to experts. GOSAI offers basic and often used augmented reality functionalities. Thomas Juldo is a DVIC alumnus who developed the project, the augmented mirror system, and some applications.

The system's structure allows it to reuse components between its applications and thus build a general catalog of tools that grows over time.

In addition, the framework uses mainstream programming languages to allow a wide range of developers to use it. The framework is written in Python and Javascript. Python is used for the framework's core components, while Javascript is used for display.

JavaScript is a flexible programming language. It is one of the core technologies of web development, and everyone can use it on both the front-end and the back-end. It is a versatile and robust language for video games. The developers can use JavaScript to make games using a variety of platforms and tools. They can use 2d and 3d libraries combined with JavaScript to create fully-fledged games in the browser or external game engine platforms [21].

The interactive mirror

The following projects presented in this thesis are implemented on an augmented mirror running on the GOSAI software system.

The augmented mirror is a platform that provides extensive interaction between the real and the virtual world. The objective is to create a recreational, medical, and educational platform.

A one-way mirror is placed against a screen. The mirror reflects perfectly where the screen is black and can

display information when the pixels emit light. A camera is placed at the top of the mirror, facing slightly down. A laptop is placed at the back of the screen (see figure 3.1).

Figure 3.1: The Augmented Mirror

The mirror can scan the room and detect and position objects the user interacts with. The augmented mirror uses the Intel D435 camera to estimate the position of a user in front of the mirror with Mediapipe. Thus, it can add a new dimension of interaction by exploiting kinesthesia. This dimension is an advantage over traditional interfaces using touch or a mouse.

This interface is ideal for the development of applications requiring movement. It is relevant for the implementation of AI-assisted sign language learning modules.

3.2 Related work

3.2.1 Sign Language Recognition

A wide range of domains uses SLR for different purposes. It can be found in robotics, human services, games, virtual reality applications, direct or remote communication, or HCI projects [22] (see figure 3.2).

Figure 3.2: Polhemus

Many early SLR systems used data gloves and accelerometers to acquire specifics of the hands. The devices measure x,y,z, orientation, and velocity directly using a sensor such as the Polhemus tracker [23] [24] or Data-

Glove [25] [26] including accelerometers, gyroscopes, and electromyography sensors (see figure 3.3). Those techniques did not allow entire natural movement and constricted the mobility of the signer, altering the signs performed and being restrictive because of the need for supplementary material.

Most techniques based on data gloves convert the position of fingers and hands according to their angles into electrical signals to obtain the desired sign.

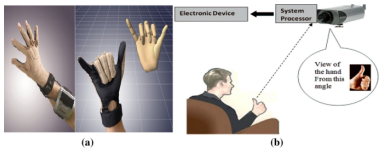


Figure 3.3: Human–computer interaction using: a CyberGlove-II [27], b vision-based system

In 2010, the ImageNet files appeared [28]. They provided a basis for the current CNNs and deep learning models, which was the beginning of computer vision. In 2012, AlexNet appeared and dramatically reduced the error rate for image recognition [29]. After the appearance of these models, the use of data gloves is gradually abandoned to focus on the implementation of modules using computer vision.

Computer vision-based techniques use pose estimation on the face, body, hands, and fingers to detect their position. This method uses images or videos of the signs through a camera and calculations on the images assisted by artificial intelligence [22].

The identification of signs must take into account many different parameters. Facial expressions and body posture are vital in determining the meaning of sentences; e.g., eyebrow position can determine the question type. Some signs are distinguishable only by lip shape, as they share a common manual sign [30].

The speed of the sign realization can change the speed induced by the performed sign. A sign can also depend on its position on the body. All limbs must therefore be taken into account during the analysis. These challenges include sensor placement, data collection and preprocessing, and model training and evaluation [31] (see figure 3.4).

Figure 3.4: Various custom gloves constructed by researchers in the sign language recognition field.

Sign language recognition systems based on computer vision and wearable sensors have been proposed by several researchers [32] [33] [34] [35] [36] [37] [37] [38],

[39] [40].

Most recent SLR techniques use various image or vision-based SLR systems comprising feature extraction and classification [41] (see figure 3.5).

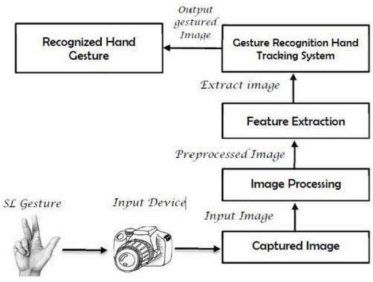


Figure 3.5: Typical Vision Based Sign Language Recognition architecture.

Many projects using computer vision assisted SLR exist [42] [43] [44] [45] [46] [47] [48].

Many of these projects use a Convolutional Neural Network (CNN) model for predicting the American Sign Language (ASL) alphabet [49]. Previously, classifiers like support vector machine [50], random forest, multilayer perceptron, transfer learning, and fine-tuning [51] were introduced for sign language recognition on simple images. Recently, shallow CNN and Capsule Networks have obtained better results [52].

Skeleton coordinate-based action recognition with coordinates has recently been attracting more and more attention to compute sign language videos because of its invariance to the subject or background. In contrast, skeleton coordinate-based SLR only takes the crucial data for its learning.

Figure 3.6: Top: Multi-person pose estimation. Body parts of the same person are linked, including foot key points (big toes, small toes, and heels). Bottom left: Part Affinity Fields (PAFs) corresponding to the limb connecting the right elbow and wrist. The color encodes orientation. Bottom right: A 2D vector in each pixel of every PAF encodes the position and orientation of the limbs.

The most commonly used pose estimation frameworks that extract coordinates from a person using pose estimation are, for example, OpenPose [53] (see figure 3.6), MoveNet [54], PoseNet [55] and MediaPipe [56].

Techniques using computer vision or data gloves recover and process the coordinates with training methods. Various machine learning algorithms are used for sign language recognition, including neural networks, support vector machines, and hidden Markov models [31].

Adeyanju et al. comprehensively reviews the state-of-the-art techniques used in sign language recognition using machine learning [57] 3.7. The paper highlights the significance of sign language recognition and its potential to revolutionize communication between the deaf and hearing communities. The authors then review the different machine-learning techniques used for sign language recognition, such as Hidden Markov Models (HMMs), Support Vector Machines (SVMs), and Deep

Neural Networks (DNNs). The authors also highlight the importance of datasets in sign language recognition and review some of the commonly used datasets for sign language recognition. They emphasize the need for large, diverse datasets to train machine learning models effectively.

However, sign language is more than a collection of well-specified gestures.

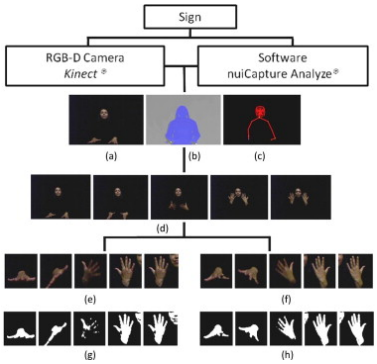


Figure 3.7: Feature extraction in Brazilian Sign Language Recognition based on phonological structure and using RGB-D sensors

3.2.2 Sign Language Learning Video Games

The video game is a dynamic audiovisual entertainment platform, accessible and stimulating the imagination of players. Using them to strengthen skills and abilities within society is possible. Video games are becoming increasingly valuable for children's development and youth culture. They enhance the function of the attentional system, stimulate visual attention, reduce reaction time, and improve the ability to discriminate shape and color, plus efficiency when following multiple objects [58].

They are an efficient didactic way to promote interest and motivation by linking playfulness and pedagogical functions [59].

The augmentation of interfaces thanks to technology, causes a better attractiveness of the learning method and thus an increase of the time voluntarily dedicated to self-learning and the motivation to concentrate on the method [**baker1994**].

Very few games use sign language as a primary element in the gameplay. We can cite Moss [60] (see figure 3.8), a video game on PlayStation VR in which the hero communicates with the player with ASL.

Figure 3.8: Moss hero communicating with the player through American sign language

Zahoor Zafrulla et al. present Copycat, a game designed to improve the American Sign Language (ASL) skills of deaf children [61]. A team of researchers developed the game at the University of California in collaboration with members of the deaf community. The game, called

CopyCat, is a digital game that uses machine learning to provide feedback to the players. The game consists of a series of mini-games that focus on different aspects of ASL, such as finger spelling, vocabulary, and grammar (see figure 3.9). In each mini-game, the player watches a video clip of someone signing a word or phrase in ASL. The player is then asked to copy the sign or phrase using their signing.

CopyCat uses machine learning to analyze the player's signing and provide feedback on their performance. The authors designed the game to be adaptive, meaning it adjusts the difficulty level based on the player's skill level. The game also tracks the player's progress and provides feedback on improvement areas. CopyCat developers then enhanced their SLR system with the Kinect depth-mapping camera, which uses colored gloves and embedded accelerometers to track children's hand movements [62].

Figure 3.9: Screenshot of ASL Game Interface and the input devices for user

Bouzid et al. explore using a learning game for SignWriting, a system for writing sign languages, to enhance sign language learning for students [63]. The authors designed the game to be played on a computer or tablet. It includes various activities such as matching signs to their written symbols and translating written symbols into signs. The authors used a 3D human character to interpret the SignWriting notation. An avatar-based system called tuniSigner [64] (see figure 3.10).

Figure 3.10: The interpretation of the sign "house" via tuniSigner

Lesmes et al. discuss the development of educational video games for deaf children in order to facilitate their inclusion in mainstream educational settings [65]. The paper outlines the design and development process of educational video games, including using a participatory design approach that involves deaf children and educators throughout the design process. The authors designed the games to incorporate sign language, visual cues, and other features that would make them accessible to deaf children. The user can see an objective written on the left part of the interface, a character in the center, and an interpreter on the right (see figure **??**).

Most mobile applications for sign language learning are simple quizzes with a lesson and a questionnaire. PocketSign is an application for learning American Sign Language through interactive activities (see figure 3.12). The project offers learning lessons, a dictionary for translating words into sign language videos, and a "finger-spelling" mode in which the user watches video tutorials of words or letters of the alphabet and then has to repeat them.

The application uses the phone's camera to film the user and verify that he or she is doing a word correctly before validating it and moving on to the next word.

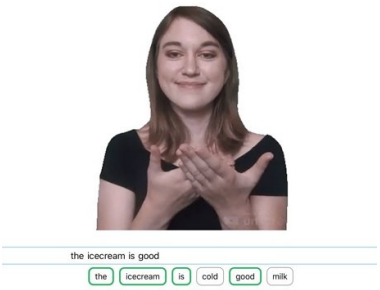


Figure 3.12: example of exercise with a tutorial in Pocket Sign

3.2.3 Visual Novel Engines

P5VN is an open-source Interactive Design and Media Major at NYU Tandon School of Engineering student project, allowing the creation of a visual novel using P5.js. It allows the display of dialogues, sprites, and backgrounds and using menus by clicking (see figure 3.13). The game scenario is easily editable through a script that the engine parses. The project initially aimed to implement a prototype engine based on p5.js with a custom scripting language.

Figure 3.13: Presentation of the visual novel engine P5VN

Another visual novel engines, Tuesday JS [66] or Monogatari [67] are simple web-based visual novel editors that can be used in a web browser. They are written in JavaScript, allowing the use of vector graphics svg, gif animations, and CSS styles.

Tuesay.js is an easy-to-use visual novel editor, free and open source. It runs on any web browser. The engine is written in JavaScript, does not use third-party libraries, and does not require additional software installation. It uses a drag-and-drop interface for editing scenes and creating interfaces. The script is displayed as a flowchart with all the elements and branches of the plot. The navigation is easy, and the editor helps to create stories with many plot options.

Monogatari.io is similar to Tuesday.js. The platform sup-

ports different media (images, videos, music) and multiple languages. It is highly customizable, open source, and multi-platform.

JavaScript is a language adapted for the creation of projects like this one. The language is used extensively in games that only require a few resources. A 2D interactive game displays only a few details and elements.

Frameworks like Phaser JS library or p5.js are suitable for coding a simple game. p5.js is a JavaScript library for creative coding, focusing on making coding accessible and inclusive for artists, designers, educators, beginners, and anyone else. It can display graphics and process many different elements.

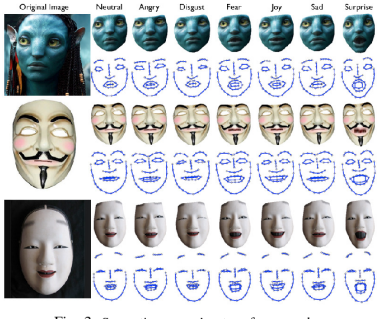


Figure 3.14: example of a real-time avatar animation from a single image thanks to semantic expression transfer.

3.2.4 Animated 3D Avatar

Most systems implement the display of an animated model from animation software (Blender, 3DS Max, Maya, Unity, Houdini) to create an animated avatar. The animations are worked directly in the software and then imported into a program for display in a game. Some projects allow the animation of a 3D model directly from the motion capture. In the project "Real-time Avatar Animation from a Single Image" [68] Saragih et Al. realize the modeling of 3D models from a simple photograph (see figure 3.14).

Figure 3.15: Animating full body character using FaceMesh and PoseNet with TensorFlow.js.

A tool such as Unity or Maya can make its animation from MediaPipe coordinates. Pose Animator is a tool that gives rendering for 2D animation. A demonstration works with FaceMesh and PoseNet (from MediaPipe) online [69] (see figure 3.15). Unfortunately, this one does not understand the finger movements necessary for precise sign language tutorials.

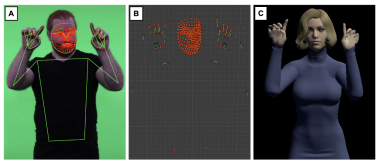


Figure 3.16: Holistic tracking applied to a video frame. A) is the annotated original footage where the red dots are the tracked landmarks, while the green lines connect the joints. B) is the same frame in Blender with landmarks plotted as orange spheres and cyan-colored bones. C) shows the motion capture data applied to an avatar. [70]

Another way is to adapt the coordinates in real-time to animate a character on Blender using Daz Studio as Nguyen et Al. [70] (see 3.16). Most interactive avatars with pose estimation use MediaPipe and TensorFlow.js (namely FaceMesh, BlazePose, and HandPose) [71] (see 3.17).

Posenet_demo [72] is a project allowing moving a 3D model with Three.js and Tensorflow's Pose Estimation model (PoseNet). The model uses Tensorflow and PoseNet to detect the critical points of the joints in each frame and then send those points over to the Model file.

The project uses an Adobe Mixamo [73] 3D model in FBX format and Blender to import the FBX format and export a GLB format [74]. The avatar follows the head and shoulder's inclination (see 3.18).

A last interesting example is that of Kalidokit [75]. Kalidokit is a tool that uses MediapipeTensorflow.js models for tracking face, eyes, pose, and hand movements. It is compatible with various models such as Facemesh, Blazepose, Handpose, and Holistic. The tool calculates simple Euler rotations and blendshape face values based on the predicted 3D landmarks.

Kalidokit is the core component for Vtuber web apps, such as Kalidoface and Kalidoface 3D. Its purpose is to rig 3D VRM models and Live2D avatars. The project is a JS library intended for developers using Mediapipe pretrained models rather than a complete app. The library still has to be adapted to run on different platforms. The project is based on Three.js, a powerful library for creating three-dimensional models and games. With few lines of JavaScript, Kalidokit allows the creation of simple 3D patterns for photorealistic, real-time scenes. The library can create complex 3D geometrics and animate and move objects.

Three.js is a JavaScript library for creating 3D scenes in a web browser. It can be used with the HTML5 canvas tag without needing a plugin. The library enables the application of textures and materials. It also provides various light sources to illuminate scenes, advanced postprocessing effects, custom shaders, and load objects from 3D modeling software. It is easy to use, intuitive, and a well-documented library.

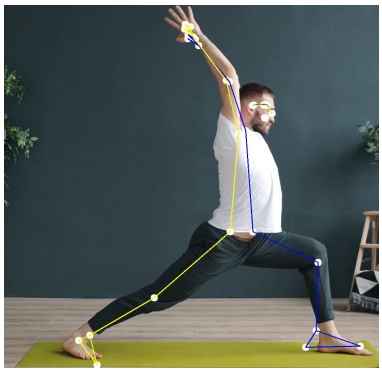


Figure 3.17: BlazePose results on yoga use-cases

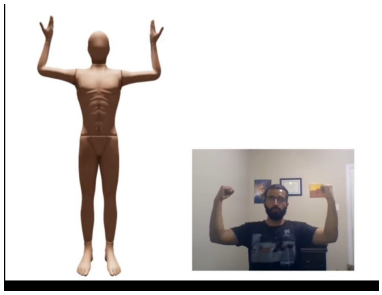


Figure 3.18: React project that will allow us to move a 3D model with Three.js (React Three Fiber) and TensorFlow's Pose Estimation model (PoseNet).

Figure 3.19: KalidoKit can move 3D avatars by tracking face and body movement with a simple webcam.

3.3 Gameplay

The sign language game is an augmented reality game, a visual novel on the augmented mirror. We follow a character during a short adventure where the user can make choices by making American Sign Language signs in front of the mirror. He can answer the characters, interact with objects, choose actions, and choose a path. Sometimes the player sees the tutorial of one, two, or three signs simultaneously. The user must then make the sign of the choice he takes. For example, he can turn right or left by making the appropriate sign (see 3.20).

Figure 3.20: Choice to go left or right in the sign language learning game.

3D avatars are the models for all the characters. Their limbs (including fingers) are animated. Only the character Aria (the hero's friend) is performing the sign tutorials. The camera captures the signs, and AI in the back end guesses them. Dialogue lines appear during the adventure to guide and discuss with the player. The user must make the "ok" sign to scroll the text with his hand.

3.4 General Architecture

3.4.1 The Sign Language Video Game in GOSAI

The application is connected to the get_sign and SLR drivers of GOSAI to estimate the sign of a person in front of the mirror. The socket.io then transmits the data to the engine.js, which runs the game, considering the user's movements (see 3.21).

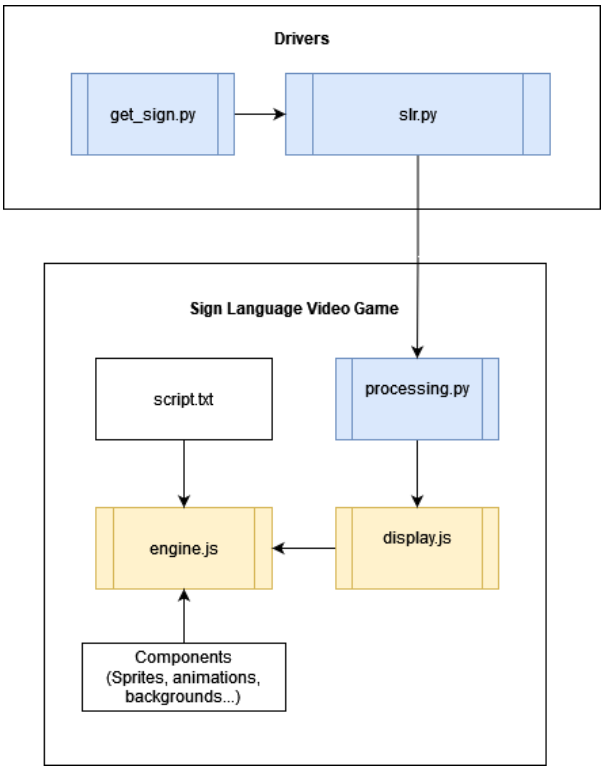


Figure 3.21: The Sign Language Video Game architecture in GOSAI.

The engine.js accesses the script containing the whole game and the components contained locally (sprites, animations, or backgrounds).

3.4.2 Visual Novel Engine

Engine implementation

The best solution to make such a game was implementing a visual novel engine on the mirror to develop everything else in the game. The P5VN engine is precisely adaptable for the mirror because the engine works with p5.js.

Previously, p5VN could load and display background, sprites of characters, some text interactive with the mouse click, and menus with multiple buttons. The engine was running synchronously in a single thread. Thanks to an important adaptation, the module is now running asynchronously to be compatible with GOSAI. The engine loads video animations automatically at launch and can then play them. The user can now interact and select menus thanks to sign language and added many other features.

Menu

In a visual novel, the player does not interact directly with the keyboard but must click on menu buttons to make choices. As the user interacts with the mirror only by the estimated pose, the menu system had to be adapted to enable debugging by clicking and making choices with movements.

Each time a menu appears, it takes the form of two or three words aligned and distributed horizontally on the screen. An avatar of Aria (the principal character) appears behind each word to demonstrate the sign related to this word (see 3.20). The location of the words and the 3D tutorials are spread over the screen's width according to the number of buttons in the displayed menu.

Aria's avatar animation plays in a loop until the player makes a sign detected by the SLR module and included in the menu.

Script

A script.txt file contains the game script in the application components. It contains a set of commands starting with the $ symbol telling the game to do a particular action. These commands can be :

- $tag indicating a specific point in the script
- $jump to jump to a tag

- $defineC to define a character (name, sprite address, text color)
- $defineImg to define an image
- $bg to display a background
- $show to display a character
- $addAnimation to play an animation to a character
- $setSprite to display the sprite of a character
- $menu to create a new menu
- $hide to hide a character
- $setvar to set a variable to a value
- $if to manage if then else commands

All the loading of sprites, animations, playback, and display are now managed automatically at startup in asynchronous instead of being set manually in the script or the code as before. The engine was running in synchronous mode at the beginning. The game starts on a loading time at startup corresponding to the loading of sprites and animation videos in the memory of GOSAI. This loading allows the game to play each animation instantly. The game directs the chosen path toward the continuity of the chosen branch. The player can choose different paths to finish the game.

3.4.3 Animations

The most efficient way to create accurate custom 3D animations for the animated sign language tutorials was to implement a module allowing one to control an avatar in 3D through pose estimation and then record the movements produced or a video of the animation.

An application to control an avatar remotely has been developed on the augmented mirror thanks to the estimation pose. The avatar can track and copy the position of the user's limbs accurately. The created application is now a free demonstration on the mirror. It also allowed the creation of all the 3D character animations on the sign language learning game.

Esla

The Educative Sign Language Avatar (ESLA) is a 3-dimensional avatar controllable in a GOSAI application following the estimated MediaPipe pose.

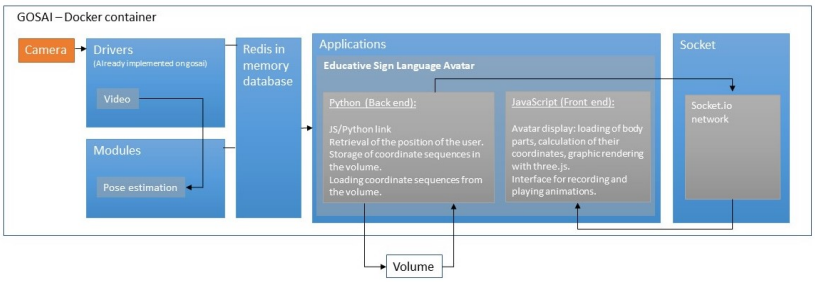


Figure 3.22: ESLA architecture on GOSAI.

The program uses the driver of the mirror camera, processed by an estimation module (see 3.22). The Redis database then transmits the data to other modules of the back end in Python. The front end, written in Javascript, manages the display, particularly the calculations related to three.js. Its movements copy live those of the user in front of the mirror [76]. This version was the first version of the interactive avatar on the augmented mirror.

Figure 3.23: The Educative Sign Language Avatar (ESLA).

An animated 3D character appears on top of the scenery at the launch of the application (see 3.23). Its animation is made possible by playing features using MediaPipe and OpenCV. The program can retrieve the coordinates of a user's position in real-time. These coordinates are processed, and the avatar copies the movements thanks to Animation Retargeting, a video game animation technique to map movements on 3D objects.

The program retrieves the links between each point of the pose provided by MediaPipe, and Three.js calculates and displays the 3D part.

Three.js entirely manages the loading of the character, its display, its animation, the rendering, the light, and the camera.

Three.js uses a pivot system with matrix4 and quaternions. However, some functions to transfer rotations from one base to another still need to be included (even though they are very much in demand by the community). Therefore, the entire limb rotation system was implemented by hand. MediaPipe first retrieves the (real) coordinates of the user. Transformations convert these coordinates into the world base (x-axis to the right, y-axis to the top, z-axis to the camera) of the three.js scene. The program retrieves the coordinates of the top of the spine, the shoulder, and the elbow (see 3.24). It creates two vectors spine/shoulder and shoulder/elbow. The coordinates of these two vectors are then read from the base of the parent of the bone to be rotated (here, the left shoulder). An algorithm then calculates a quaternion containing the rotation data from the first vector to the second.

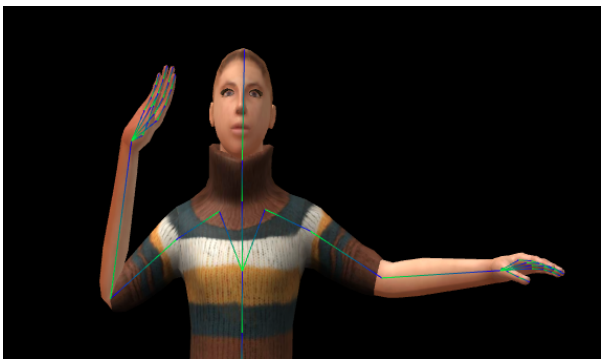


Figure 3.24: ESLA's squeletton.

Three.js then applies this quaternion to the limb, which rotates the same as the user's arm with his shoulder in its local base. This rotation allows precise rotation transfers from one base to another, thus animating all parts of the avatar.

Unfortunately, the project lacks much precision in the movements, especially at the level of the fingers, which

is very inconvenient for creating accurate tutorials in sign language. Another more efficient solution for creating 3D animations with the estimated pose had to be implemented.

Kalidokit module implementation

The creation of an avatar controllable with Mediapipe on the mirror has been by the implementation of the Kalidokit solver. This JS library allows the animation of arms, hands, fingers, face, mouth, eyelids, and pupils of VRM models (Virtual Reality Models). The VRM is based on the standard 3D format glTF2.0 to manage humanoid models. It aims to be particularly expressive. This model has many articulations and can blink and animate its mouth. It is often used in VR games (VR chat) or by Vtubers (entertainment broadcasters who use a virtual avatar).

The models used in the game come from the site Vroid Hub cite[77]. They must respect specific conditions of use: use by a third party, downloadable, use as an avatar, and commercial use by a company. The avatar must also be sober, as it should be a model taking part in a demonstration on the augmented mirror deployed at the De Vinci Innovation Center. The one selected in the demo application allowing to control the avatar in augmented reality is called "papa_de_him_chan".

The model is easily changeable locally in the application, allowing the user to manipulate different characters. This application makes creating sign language tutorial animations for the learning game possible.

The implementation of Kalidokit on the mirror has been reworked to suit the operation of GOSAI. The joints are set to greater freedom to allow more flexible animations (see 3.25). Such an application allows recording qualitative signs and poses for the characters and tutorials of the game.

The models used for the characters of Aria, the grandmother, and the salesman were recovered on Vroid Hub

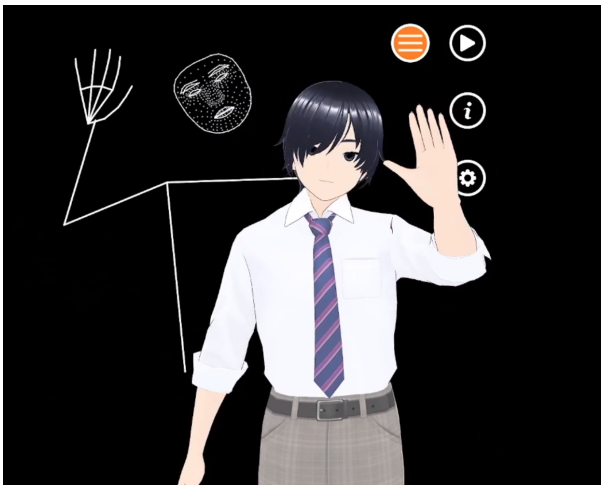


Figure 3.25: The first version of the Augmented Reality Interactive Avatar (ARIA).

and are free to use. The VRM models are mainly adapted for Vtubers, explaining why the characters look childish and look like Japanese anime characters.

Figure 3.26: Aria's character performing the "I love you" sign.

All the avatars' sprites and 3D video animations were recorded from the "Aria" application on the mirror. These animations include 22 video animations and 14 sprites for the character of Aria in the sign language game (see 3.26).

3.5 Sign Language AI

3.5.1 Overview

Sign language involves using the upper part of the body, such as hand gestures, facial expressions, lip-reading, head nodding, and body postures to disseminate infor-

mation [22].

The final goal of this work is to implement an SLR AI inside the sign language learning game. This AI allows the recognition of choices in the game thanks to the estimation of the user's pose and the control of the game using commands linked to signs.

Many models with sign language recognition already exist. The work done here proposes a model and an easy system of creation, dataset management, training, and visualization of the data.

3.5.2 Integration in GOSAI

The project of creating the SLR AI is apart from GOSAI [**slr_mirror**]. GOSAI integrates only the weights in a module, making the comparison between a sign made in front of the camera and the values of the weights recorded locally.

The video game retrieves the user's coordinates and places them in tensors containing the data of a video of 30 frames.

The program passes the model by passing the data to it, retrieving a table of probabilities concerning each action.

The guessed sign is then transmitted from the SLR module through GOSAI's Redis database to the game engine. When the probability of a detected sign overfits a threshold, the game engine considers the sign as validated.

3.5.3 Structure

The project contains six local modules and a main file that initializes the parameters and launches the different processes. The processes are an optional phase of dataset creation, a tutorial phase that displays the skeleton for data visualization, a preprocessing phase that retrieves the data from the dataset and formats them, a data

augmentation module, and a weight calculation and exportation in different file types.

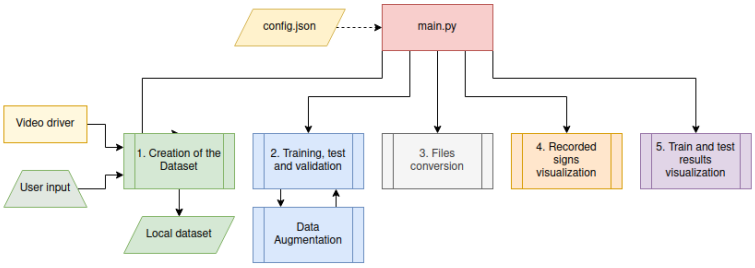


Figure 3.27: Structure of the AI for sign language recognition.

The developed AI is a full-fledged project that does not require a goal. It takes five optional steps (see 3.27). These steps are: the creation of a dataset, the training, test, and validation (with data augmentation or not), the conversion of the created files, the visualization of the recorded signs, and the visualization of the training and test results.

Dataset If a movement is included in the known actions but is not present in the files, the creation of the dataset of this movement launches automatically.

The user creates the dataset, which is stored in a folder "MP data" created and separated into three folders: train, valid, and test containing the actions (name of the movement). The recorded sequences are automatically separated between the three folders train (80% of them), test (10%), and valid (10%). The program automatically completes the existing dataset by default according to the number of sequences entered in the configuration file.

Each sequence is 30 frames each by default, including coordinates, that is to say, and now 116 data per frame (that is to say 58 points) after removing 431 points of the face and just Keeping 4. The four kept points are one for the chin, forehead, right, and left cheek. It enables signs using the head to be better detected.

The program retrieves the coordinates from Mediapipe during the creation of the dataset. If the Intel camera is

undetected, the device's webcam record, an Intel real sense, or the default pc webcam. When implemented on another device, such as the interactive mirror, this driver must be modified to match the camera used.

At the start of the dataset creation, the user alternates one second of video recording, then two seconds to reposition in a loop for each sign. He has video feedback on himself to check if he is well-positioned in the plan.

The recorded signs visualization The recorded signs visualization module (data_vis) shows a visualization of every recorded sign (stored in the dataset). This process verifies that the signs have been made and are stored correctly on site. The program retrieves the coordinates of a person stored in the dataset (for each video), sorts the coordinates of the points of each body part, and finds the different links between each point frame by frame.

The visualization displays all the links between the points stored in the dataset. The program also displays the action with which the visualization of the sign is associated above the drawn character.

Preprocess The preprocessing phase retrieves all the data from the dataset and places them in tensors to provide them to the model. After creating the preprocess instances (according to their type: train, test, valid), we pass them to the data loader.

These pass an index to them (concerning a sequence among all the sequences attributed to the type of the preprocess). Their program calculates which action is concerned by the desired sequence and recovers all the coordinates of the frames of this sequence. Some papers express that sign language movement is only possible by indicating with lips or facial expressions citecooper2011sign. Therefore, the dataset should keep some lip coordinates integrated during the preprocessing. For other papers, keeping facial and lips expressions is unnecessary to keep these points, which takes much

storage for nothing [78]. Here, the preprocess takes into consideration only four coordinates of the face. This selection allows for much faster train loops because all these points are unnecessary.

Data augmentation The preprocessing phase optionally calls the data augmentation just after retrieving the data of the concerned sequence. The functionality reviews the data and applies a random horizontal and vertical shift and scale to it. This process artificially increases the number of positions the user can stand.

Model The model comprises a bidirectional LTSM, a linear, a dropout, a batchnorm1d, a relu, and another Linear. The AI trained on 16 different signs or 1600 sequences. The model reaches an accuracy of 87% on the test set and 96% on the training (see 3.28).

Figure 3.28: Visualization of testing and training accuracy over 200 epochs. In yellow is the accuracy without data augmentation, and in green with data augmentation.

Recurrent Neural Networks (RNN) allow the processing of temporal sequences (language, videos, numerical data). They keep the memory of past data to predict data sequences shortly. LSTM (Long-Short-Term-Memory) and Bidirectional layers are RNNs used in the current model to consider the different coordinates and frames

already passed for better performance. The LSTMs are composed of several gates (respectively 3 and 2), which allow one to forget or to selectively memorize the information of the previous temporal sequence in dynamic memory. This RNN is particularly adapted to multiple input dimensions. For example, here, the data is composed of sequences containing frames and coordinates (3 dimensions).

The training uses the optimizer AdamW, a stochastic gradient descent method based on adaptive estimation of first-order and second-order moments with an added method to decay weights [79].

The program exports the output models in pth format. Pth is a standard PyTorch convention to save models. They can also be exported to onnx (Open Neural Network Exchange) [80], an open format designed to represent machine learning models.

Onnx extension enables greater interoperability in the AI tool community, allowing many AI frameworks to be more compatible by allowing them to share models. This interoperability allows users to deploy trained models in different software platforms like GOSAI quickly.

Here, the onnx models retrieved from the project output are directly implemented in the SLR module of GOSAI.

3.5.4 Configuration

All the processes mentioned above are optional. The user can enable or disable many project features in the config.json file.

All the options are: "actions" (all the signs recognized by the AI, "adapt_for_mirror" (to transform the coordinates specifically for the augmented mirror), "convert_files" (to convert the files from pth to onnx), "erase_dataset" (erasing the dataset before recording a new one), "erase_runs" (erase the local TensorBoard files), "make_data_augmentation", "make_dataset", "make_train", "make_-

tuto" (visualization of the recorded movements), "nb_-epochs", "nb_sequences" (total video number the user wants to record for a sign), "sequence_length" (one video frame number).

The default actions are "nothing", "empty", "ok", "yes", "no", "left", "right", "house", "store", "hello", "goodbye", "television", "leave", "eat", "apple", and "peach". Those are all the signs used inside the SL video game.

The project aims to make it as easy as possible for the user to create weights according to his desired characteristics.

3.5.5 Visualization of the results

TensorBoard is a tool providing visualization solutions to machine learning tests. It allows for tracking and visualizing metrics such as loss and correctness. TensorBoard allows displaying the exported model's graph, histograms of weights, biases or other Tensors, and many other functions.

Just by entering the "make set-up_tensorboard" function in the terminal (or by doing a "make first_boot"), the user launches the creation of a local server running on port 6006, retrieving the weights as the model trains.

A browser page then opens automatically to access the local server address. The user can study a graph of the training and test phase (percentage of success according to the number of epochs) on this page.

The logs of these results are stored locally. They are accessible on a webpage, and the user can easily compare them. This visualization makes experimentation easier.

3.5.6 Limitations

After many experiments, several biases have appeared that directly affect the AI results. For example, the size of a user differing too much from the person who created

the dataset is sometimes problematic, resulting in a consequent difference in recognition performance between users with the same or different morphology.

The results of the SLR also depend on Mediapipe parameters since the AI bases the training on the data obtained from the pose estimation. The training results show that Mediapipe was better at recognizing a person's limbs with a strong contrast with the background. The background choice resulted in a significant difference of about 15% in performance on the training and test set. Mediapipe better detects the contrast between white skin and a black shirt and better recognizes afterward when a person creates the dataset or tests the SLR wearing one.

This contrast causes the SLR to be more effective on signs made on the belly (being a plain background without detail) than those made on other places with more details in the background.

3.6 Usage scenario

The augmented mirror primarily aims to be placed in a living room. Its extensive use allows one to have fun and learn by interacting. The mirror requires no additional material, so its use remains very simple. The user directly launches the learning game of sign language from the menu of the mirror. The user has to follow the instructions given by Aria to follow the course of the adventure. The game is primarily aimed at young children. Through repetition and distraction, they intuitively learn certain gestures.

3.7 User study

3.7.1 Set Up

As the sign language learning game is intended for young users, this test focuses on a sample of 12 people between 18 and 25 years old. Six people watch a number

of video sign language tutorials on the "pocket sign" application. The signs viewed correspond to those encountered in the game when choosing the path through the house ("ok", "yes", "left", "right", "house", "store", "hello", "goodbye", "television", "leave", "eat") or the path through the store ("ok", "yes", "no", "left", "right", "house", "store", "goodbye", "apple", "peach"). 4 people view the signs of the passage through the house, and two view the signs of the passage through the store. When they think they have understood the sign viewed, they skip to the next sign.

Six other people use the sign language learning game on the mirror. They are told the principle of the game before using it. Two of them go through the house, and four of them go through the store.

Both groups are told at the beginning that they will be evaluated on certain signs at the end of the study. For both groups, the session is timed from start to finish.

3.7.2 Results

Sign Language Video Game Users took an average of 4.44 minutes from game launch to completion in one session. Some users have experienced problems related to poor AI recognition. Notably, the sign "peach" and the sign "eat" were difficult for two users.

One user took 8:25 to complete the game. He had difficulty making the "go" sign in the house. The fact that a tall person (1,84m) created the dataset and this user had a much smaller height (1,60m) explains the bias. Users were tested after their session on some random signs. Their average score was 2.33 signs learned out of 4. The errors made are related to several elements. The AI sometimes finds a sign too quickly when the user still needs to place his hands correctly.

The user then needs more time to look at the signs and choose. Another error is related to the fact that when there are three tutorials simultaneously (menus with

three choices), the user needs help concentrating on each one, and memorizing the signs is less efficient.

Increasing the detection threshold value would be an efficient method to correct the problem related to the too-high detection speed of the AI.

To conclude, people had to note their opinion about diverse questions (on a score of 10). Their average motivation to learn sign language this way (with an augmented mirror and the video game) was 7,91/10. The average fun in interacting notation was 8,25/10. People's ease of understanding what needs to be done had a notation of 7,58/10. The practicality of using this learning method (augmented mirror) had a notation of 8,25/10. The effectiveness of using this learning method received a notation of 6,41/10. Moreover, they noted the capacity of the AI to guess the correct sign 6,11/10.

The global feedback is that the avatar sometimes moves his fingers too much and cannot be precise. The fact that the display is in 2D (on the mirror screen) and not in 3D makes it challenging to understand the signs. The fact that the characters look like anime characters and not more realistic ones blocked a user. His aspect can put off the desire to practice.

Some users forgot the "ok" sign to skip dialogues during the game, although the sign was shown twice at the beginning of the game. Their forgetfulness prevented them from continuing the practice without external intervention to remind them of the sign. One user had learning difficulties due to the avatar's movements being considered unclear. However, interacting with a real person allowed him to learn more effectively. The lack of accuracy of the AI was also strongly criticized (rated 6.11/10 by users).

The 3D animated tutorials were generally liked and considered more attractive than simple video tutorials. Users enjoyed using the game on an augmented mirror. They considered the mirror's interface mainstream and easy to integrate into a living room.

Implementing the game on an augmented mirror

Videos *POTENTIAL USER STUDY ABOUT LEARNING WITH VIDEO TUTORIALS*

3.8 Discussions

The SL Game stimulates visual, verbal, and kinesthetic intelligence. Visual intelligence is the ability to create mental images and perceive the visible world precisely in its three dimensions. Viewing 3D sign language tutorials and having to "decode" the different movements and positions of the avatars' limbs stimulate the user's visual intelligence. The user needs to perceive, to make a mental projection before re-performing the identified signs.

Verbal intelligence is the ability to read, speak quickly and connect mental ideas into words. Connecting a picture to a word in the project leads to the ability to express oneself fluently and connect a word's mental idea into a gesture.

Kinesthetic intelligence is the ability to use one's body finely and elaborately, express oneself through movement, and control one's body movements well. This type of intelligence is stimulated through play in this project because the brain must translate the mental image of body positions and movements in space into real positions and movements.

The stimulation of three different intelligences, added to the motivating gamification aspect, favors information retention. The user can remember a particular gesture by having seen it and projected it mentally, by the memory of giving it a linguistic meaning, and by having performed it physically in space.

3.9 Conclusion

3.9.1 Playful Learning and ASL

Playful learning and gamification are learning methods that use playful elements to facilitate the assimilation of knowledge.

Video games are a popular form of these approaches and are often used to teach practical skills such as sign language. Using augmented reality, pose estimation, and artificial intelligence in the Sign Language Video Game creates an immersive and interactive experience for the learner to practice reproducing signs correctly in real-time. This approach promotes learning by providing instant feedback and allowing learners to practice as often as they want.

In addition, using cutting-edge technology often generates interest and engagement in the learner, which can help increase motivation and perseverance in learning sign language.

3.9.2 Limitations

The results of the user studies make it possible to extract a certain number of limitations from the project.

The size difference between the person who created the dataset and a potentially taller or shorter user is problematic. The accuracy is high for the dataset creator but low for other users with a different morphology. This problem can be solved by randomly compressing the coordinates to the horizontal and vertical axis in the data augmentation or by people of different sizes registering the dataset.

The dataset produced and used is small. It contains little noise, slight variation, and little mixing. The model is also moderately efficient, even if he trains on several different signs. In the case of an enlargement of the dataset, the accuracy would decrease even more. The model should therefore be reworked in the future. Implement-

ing transformers in the model could be relevant and improve performance. Transformers are used in areas such as image processing, biological sequence analysis, and video understanding.

Another area for improvement is the sometimes tricky clarity of avatars in sign language tutorials. Each avatar's bones' pivot angles are limited to avoid glitches and ragdolls. It is sometimes hard to record clear signs, especially at the finger level. The user must then imitate and learn a model that is not accurate. The avatar can be slightly reworked to widen the possible angles and thus gain precision.

3.9.3 Future Works

The current game is suitable for direct practice. However, the learning system is too simple because users can afford to copy the sign without actually learning it. It would be necessary to add an evaluation dimension with signs that come up in the potential choices without giving the tutorials every time. Thus the user will have to force himself to remember and learn better by repetition.

(Current work) This training dimension is already present in another application on the mirror, similar to sign language training. The application consists of successive viewings of sign language tutorial videos, practice, and correction.

The correction phase consists of frozen Mediapipe bones and points (forming a skeleton) appearing on the mirror and performing the beginning of a sign.

The user also has a Mediapipe skeleton displayed on the mirror and following his position. When he superimposes his skeleton on that of the model, the model's skeleton moves slightly to display the next frame of the recorded sign movement. The user must then superimpose his skeleton on the second frame until he validates the 30 frames that make up a sign. The person must superimpose some points of his body, especially his

hands, and fingers. These corrections allow him to learn the sign he has just made precisely. The training then moves to a tutorial of another sign.

Implementing the correction phase of this training in the sign language game would have a genuine interest. The corrections allow the user to better remember the signs by having to perform them several times by focusing on the position of each finger. The clear demonstrations of the signs allow him to visualize the correct position he has to take in the space. This implementation would thus answer some of the current problems of the game. The correction phase is not implemented for now because it broke the game's rhythm during its progress, but its implementation at the end of the game as a reminder of the learned signs would be an effective learning method.

4 Music learning applications in AR

Conclusion 5

5.1 Contribution

5.2 Future Works

5.3 Acknoledgements

References

[1] Shailesh S Kantak and Carolee J Winstein. 'Learning–performance distinction and memory processes for motor skills: A focused review and perspective'. In: *Behavioural brain research* 228.1 (2012), pp. 219–231 (cited on page 5).

[2] Awaz Naaman Saleem, Narmin Mohammed Noori, and Fezile Ozdamli. 'Gamification applications in E-learning: A literature review'. In: *Technology, Knowledge and Learning* 27.1 (2022), pp. 139–159 (cited on page 5).

[3] Hartmut Seichter. 'Augmented reality and tangible interfaces in collaborative urban design'. In: *Computer-Aided Architectural Design Futures (CAADFutures) 2007: Proceedings of the 12th International CAADFutures Conference*. Springer. 2007, pp. 3–16 (cited on page 6).

[4] Alois Ferscha, Stefan Resmerita, and Clemens Holzmann. 'Human computer confluence'. In: *Universal Access in Ambient Intelligence Environments: 9th ERCIM Workshop on User Interfaces for All, Königswinter, Germany, September 27-28, 2006. Revised Papers*. Springer. 2007, pp. 14–27 (cited on page 6).

[5] Constantine Stephanidis et al. 'Seven HCI grand challenges'. In: *International Journal of Human–Computer Interaction* 35.14 (2019), pp. 1229–1269 (cited on page 6).

[6] Leland D Bland. 'The college music theory curriculum: the synthesis of traditional and comprehensive musicianship approaches'. In: *College Music Symposium*. Vol. 17. 2. JSTOR. 1977, pp. 167–174 (cited on page 8).

[7] Olivier Donnat. 'Les amateurs, enquête sur les activités artistiques des Français'. In: *Les amateurs, enquête sur les activités artistiques des Français*. 1996, p. 198 (cited on page 9).

[8] Robert J Zatorre and Andrea R Halpern. 'Mental concerts: musical imagery and auditory cortex'. In: *Neuron* 47.1 (2005), pp. 9–12 (cited on page 10).

[9] Robert J Weber and Suellen Brown. 'Musical imagery'. In: *Music Perception* 3.4 (1986), pp. 411–426 (cited on page 10).

[10] Jamie Zigelbaum et al. 'BodyBeats: whole-body, musical interfaces for children'. In: *CHI'06 extended abstracts on human factors in computing systems*. 2006, pp. 1595–1600 (cited on page 10).

[11] Xiao Xiao, Basheer Tome, and Hiroshi Ishii. 'Andante: Walking Figures on the Piano Keyboard to Visualize Musical Motion.' In: *NIME*. Cambridge, MA. 2014, pp. 629–632 (cited on page 10).

[12] Paul Taele, Laura Barreto, and Tracy Hammond. 'Maestoso: An intelligent educational sketching tool for learning music theory'. In: *Proceedings of the*

AAAI Conference on Artificial Intelligence. Vol. 29. 2. 2015, pp. 3999–4005 (cited on page 11).

[13] Mattia Davide Amico, Luca Andrea Ludovico, et al. 'Kibo: A MIDI controller with a tangible user interface for music education'. In: *Proceedings of the 12th International Conference on Computer Supported Education. 1: CSME*. SCITEPRESS. 2020, pp. 613–619 (cited on page 11).

[14] Dongjo Kim et al. 'Heterogeneous interfacial properties of ink-jet-printed silver nanoparticulate electrode and organic semiconductor'. In: *Advanced materials* 20.16 (2008), pp. 3084–3089 (cited on page 12).

[15] Arshad Khan et al. 'Soft inkjet circuits: rapid multi-material fabrication of soft circuits using a commodity inkjet printer'. In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 2019, pp. 341–354 (cited on pages 12, 15).

[16] Michael Wessely et al. 'Sprayable user interfaces: Prototyping large-scale interactive surfaces with sensors and displays'. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020, pp. 1–12 (cited on page 12).

[17] Analisa Russo et al. 'Pen-on-paper flexible electronics'. In: *Advanced materials* 23.30 (2011), pp. 3426–3430 (cited on page 12).

[18] Anneli Hershman et al. 'Light it up: Using paper circuitry to enhance low-fidelity paper prototypes for children'. In: *Proceedings of the 17th ACM conference on interaction design and children*. 2018, pp. 365–372 (cited on page 12).

[19] Adrien Lefevre. *capacitive$_t o_n otes$*. https://github.com/AdrLfv/capacitive_to_notes. 2022 (cited on pages 14, 15).

[20] Salma A Essam El-Din and Mohamed A Abd El-Ghany. 'Sign Language Interpreter System: An alternative system for machine learning'. In: *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*. IEEE. 2020, pp. 332–337 (cited on page 21).

[21] *javascriptgaming*. https://medium.com/nerd-for-tech/javascript-and-the-gaming-industry-3d2ff7f102e9 (cited on page 22).

[22] IA Adeyanju, OO Bello, and MA Adegboye. 'Machine learning methods for sign language recognition: A critical review and analysis'. In: *Intelligent Systems with Applications* 12 (2021), p. 200056 (cited on pages 23, 24, 39).

[23] M.B. Waldron and Soowon Kim. 'Isolated ASL sign recognition system for deaf persons'. In: *IEEE Transactions on Rehabilitation Engineering* 3.3 (1995), pp. 261–271. DOI: 10.1109/86.413199 (cited on page 23).

[24] B. Lekhashri and A. Arun Pratap. 'Use of motion-print in sign language recognition'. In: *2011 National Conference on Innovations in Emerging Technology*. 2011, pp. 99–102. DOI: 10.1109/NCOIET.2011.5738842 (cited on page 23).

[25] Mohammed Kadous. 'Machine Recognition of Auslan Signs Using PowerGloves: Towards Large-Lexicon Recognition of Sign Language'. In: (Feb. 1970) (cited on page 24).
[26] Dimitris Metaxas. 'Adapting hidden Markov models for ASL recognition by using three-dimensional computer vision methods'. In: (1997) (cited on page 24).
[27] *cyberglovesystems*. https://www.cyberglovesystems.com/products/cyberglove-II/photos-video (cited on page 24).
[28] F-F ImageNet Li. 'Crowdsourcing, benchmarking & other cool things'. In: *CMU VASC Semin* 16 (2010), pp. 18–25 (cited on page 24).
[29] Md Zahangir Alom et al. 'The history began from alexnet: A comprehensive survey on deep learning approaches'. In: *arXiv preprint arXiv:1803.01164* (2018) (cited on page 24).
[30] Helen Cooper, Brian Holt, and Richard Bowden. 'Sign language recognition'. In: *Visual Analysis of Humans: Looking at People* (2011), pp. 539–562 (cited on page 24).
[31] Karly Kudrinko et al. 'Wearable Sensor-Based Sign Language Recognition: A Comprehensive Review'. In: *IEEE Reviews in Biomedical Engineering* 14 (2021), pp. 82–97. DOI: 10.1109/RBME.2020.3019769 (cited on pages 24, 25).
[32] Bogdan Ionescu et al. 'Dynamic hand gesture recognition using the skeleton of the hand'. In: *EURASIP Journal on Advances in Signal Processing* 2005 (2005), pp. 1–9 (cited on page 24).
[33] Chenglong Yu et al. 'Vision-based hand gesture recognition using combinational features'. In: *2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. IEEE. 2010, pp. 543–546 (cited on page 24).
[34] Shao-Zi Li et al. 'Feature learning based on SAE–PCA network for human gesture recognition in RGBD images'. In: *Neurocomputing* 151 (2015), pp. 565–573 (cited on page 24).
[35] Jayesh S Sonkusare et al. 'A review on hand gesture recognition system'. In: *2015 International Conference on Computing Communication Control and Automation*. IEEE. 2015, pp. 790–794 (cited on page 24).
[36] Vladislava Bobić, Predrag Tadić, and Goran Kvaščev. 'Hand gesture recognition using neural network based techniques'. In: *2016 13th Symposium on Neural Networks and Applications (NEUREL)*. IEEE. 2016, pp. 1–4 (cited on page 24).
[37] Md Mohiminul Islam, Sarah Siddiqua, and Jawata Afnan. 'Real time hand gesture recognition using different algorithms based on American sign language'. In: *2017 IEEE international conference on imaging, vision & pattern recognition (icIVPR)*. IEEE. 2017, pp. 1–6 (cited on page 24).
[38] Himadri Nath Saha et al. 'A machine learning based approach for hand gesture recognition using distinctive feature extraction'. In: *2018 IEEE 8th*

annual computing and communication workshop and conference (CCWC). IEEE. 2018, pp. 91–98 (cited on page 24).

[39] Razieh Rastgoo, Kourosh Kiani, and Sergio Escalera. 'Hand pose aware multimodal isolated sign language recognition'. In: *Multimedia Tools and Applications* 80 (2021), pp. 127–163 (cited on page 25).

[40] Biao Xu, Shiliang Huang, and Zhongfu Ye. 'Application of tensor train decomposition in S2VT model for sign language recognition'. In: *IEEE Access* 9 (2021), pp. 35646–35653 (cited on page 25).

[41] KP Nimisha and Agnes Jacob. 'A brief review of the recent trends in sign language recognition'. In: *2020 International Conference on Communication and Signal Processing (ICCSP)*. IEEE. 2020, pp. 186–190 (cited on page 25).

[42] Yonas Fantahun Admasu and Kumudha Raimond. 'Ethiopian sign language recognition using Artificial Neural Network'. In: *2010 10th International Conference on Intelligent Systems Design and Applications*. IEEE. 2010, pp. 995–1000 (cited on page 25).

[43] Mohamed Deriche, Salihu O Aliyu, and Mohamed Mohandes. 'An intelligent arabic sign language recognition system using a pair of LMCs with GMM based classification'. In: *IEEE Sensors Journal* 19.18 (2019), pp. 8067–8078 (cited on page 25).

[44] Tareq Z Ahram, Waldemar Karwowski, and Jay Kalra. *Advances in Artificial Intelligence, Software and Systems Engineering: Proceedings of the AHFE 2021 Virtual Conferences on Human Factors in Software and Systems Engineering, Artificial Intelligence and Social Computing, and Energy, July 25-29, 2021, USA*. Vol. 271. Springer Nature, 2021 (cited on page 25).

[45] Tao Song et al. 'Intelligent human hand gesture recognition by local–global fusing quality-aware features'. In: *Future Generation Computer Systems* 115 (2021), pp. 298–303 (cited on page 25).

[46] Carman KM Lee et al. 'American sign language recognition and training method with recurrent neural network'. In: *Expert Systems with Applications* 167 (2021), p. 114403 (cited on page 25).

[47] Kunyoung Lee et al. 'A comparative analysis on the impact of face tracker and skin segmentation onto improving the performance of real-time remote photoplethysmography'. In: *Intelligent Human Computer Interaction: 12th International Conference, IHCI 2020, Daegu, South Korea, November 24–26, 2020, Proceedings, Part II 12*. Springer. 2021, pp. 27–37 (cited on page 25).

[48] Liqing Gao et al. 'RNN-transducer based Chinese sign language recognition'. In: *Neurocomputing* 434 (2021), pp. 45–54 (cited on page 25).

[49] Lee Yi Bin, Goh Yeh Huann, and Lum Kin Yun. 'Study of convolutional neural network in recognizing static American sign language'. In: *2019 IEEE international conference on signal and image processing applications (ICSIPA)*. IEEE. 2019, pp. 41–45 (cited on page 25).

[50] Celal Savur and Ferat Sahin. 'Real-time american sign language recognition system using surface emg signal'. In: *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2015, pp. 497–502 (cited on page 25).

[51] Yaser Saleh and Ghassan Issa. 'Arabic sign language recognition through deep neural networks fine-tuning'. In: (2020) (cited on page 25).

[52] Md Mehedi Hasan et al. 'Classification of sign language characters by applying a deep convolutional neural network'. In: *2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT)*. IEEE. 2020, pp. 434–438 (cited on page 25).

[53] Zhe Cao et al. 'Realtime multi-person 2d pose estimation using part affinity fields'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7291–7299 (cited on page 25).

[54] Rishabh Bajpai and Deepak Joshi. 'MoveNet: A Deep Neural Network for Joint Profile Prediction Across Variable Walking Speeds and Slopes'. In: *IEEE Transactions on Instrumentation and Measurement* 70 (2021), pp. 1–11. DOI: 10.1109/TIM.2021.3073720 (cited on page 25).

[55] Alex Kendall, Matthew Grimes, and Roberto Cipolla. 'Posenet: A convolutional network for real-time 6-dof camera relocalization'. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2938–2946 (cited on page 25).

[56] Camillo Lugaresi et al. 'Mediapipe: A framework for building perception pipelines'. In: *arXiv preprint arXiv:1906.08172* (2019) (cited on page 25).

[57] Sılvia Grasiella Moreira Almeida, Frederico Gadelha Guimarães, and Jaime Arturo Ramırez. 'Feature extraction in Brazilian Sign Language Recognition based on phonological structure and using RGB-D sensors'. In: *Expert Systems with Applications* 41.16 (2014), pp. 7259–7271 (cited on page 25).

[58] C Shawn Green and Daphne Bavelier. 'Enumeration versus multiple object tracking: The case of action video game players'. In: *Cognition* 101.1 (2006), pp. 217–245 (cited on page 26).

[59] Ricardo Tejeiro Salguero, Manuel Pelegrina del Rıo, and Jorge Luis Gómez Vallecillo. 'Efectos psicosociales de los videojuegos'. In: *Comunicación: revista Internacional de Comunicación Audiovisual, Publicidad y Estudios Culturales, 1 (7), 235-250.* (2009) (cited on page 26).

[60] *Moss*. https://store.steampowered.com/app/846470/Moss/. 2018 (cited on page 26).

[61] Zahoor Zafrulla et al. 'CopyCat: an American sign language game for deaf children'. In: *Face and Gesture 2011*. IEEE Computer Society. 2011, pp. 647–647 (cited on page 26).

[62] Zahoor Zafrulla et al. 'American sign language recognition with the kinect'. In: *Proceedings of the 13th international conference on multimodal interfaces*. 2011, pp. 279–286 (cited on page 27).
[63] Yosra Bouzid et al. 'Using educational games for sign language learning-a signwriting learning game: Case study'. In: *Journal of Educational Technology & Society* 19.1 (2016), pp. 129–141 (cited on page 27).
[64] Yosra Bouzid and Mohamed Jemni. 'An Avatar based approach for automatically interpreting a sign language notation'. In: *2013 IEEE 13th International Conference on Advanced Learning Technologies*. IEEE. 2013, pp. 92–94 (cited on page 27).
[65] Clemencia Zapata Lesmes et al. 'Design and Production of Educational Video Games for the Inclusion of Deaf Children'. In: *Procedia Computer Science* 198 (2022), pp. 626–631 (cited on page 27).
[66] *TuesdayJS*. https://kirilllive.github.io/tuesday-js/ (cited on page 28).
[67] *Monogatari*. https://monogatari.io/ (cited on page 28).
[68] Jason M Saragih, Simon Lucey, and Jeffrey F Cohn. 'Real-time avatar animation from a single image'. In: *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)*. IEEE. 2011, pp. 117–124 (cited on page 29).
[69] $pose_animator$. https://blog.tensorflow.org/2020/05/pose-animator-open-source-tool-to-bring-svg-characters-to-life.html (cited on page 29).
[70] Lan Thao Nguyen et al. 'Automatic generation of a 3D sign language avatar on AR glasses given 2D videos of human signers'. In: *Proceedings of the 1st International Workshop on Automatic Translation for Signed and Spoken Languages (AT4SSL)*. 2021, pp. 71–81 (cited on page 29).
[71] *blazepose*. https://blog.tensorflow.org/2021/08/3d-pose-detection-with-mediapipe-blazepose-ghum-tfjs.html (cited on page 29).
[72] $posenet_demo$. https://github.com/aphrx/posenet_demo (cited on page 30).
[73] *mixamo*. https://www.mixamo.com/#/ (cited on page 30).
[74] *posenet*. https://github.com/aphrx/posenet (cited on page 30).
[75] *kalidokit*. https://github.com/yeemachine/kalidokit (cited on page 30).
[76] *esla*. https://github.com/AdrLfv/esla (cited on page 35).
[77] *vroid*. https://hub.vroid.com/en/ (cited on page 37).
[78] Philippe Dreuw et al. 'Speech recognition techniques for a sign language recognition system'. In: *hand* 60 (2007), p. 80 (cited on page 42).
[79] Ilya Loshchilov and Frank Hutter. 'Decoupled weight decay regularization'. In: *arXiv preprint arXiv:1711.05101* (2017) (cited on page 43).
[80] *onnx*. https://onnx.ai/ (cited on page 43).