LAB University of Applied Sciences

AI-Enhanced Serious Game for Children with Special Needs

LAB University of Applied Sciences
Bachelor’s degree in information and communications technology
2025
Pilar Romero Cutillas

Abstract

Author(s)	Publication type	Completion year
Pilar Romero Cutillas	Thesis, UAS	2025
	Number of pages	
	54	

Title of the thesis

AI-Enhanced Serious Game for Children with Special Needs

Degree, Field of Study

Bachelor's degree in Information and Communication Technology

Name, title and organisation of the client

e.g., Clint Client, Design Engineer M.Sc. (Tech.), Design Agency Ltd.

Abstract

This thesis explores the application of educational games for kids with special needs and how artificial intelligence (AI) can be effectively integrated into these games to create a more personalized, adaptive, and engaging user experience. The application features a variety of mini-games tailored to foster cognitive and linguistic development. It utilizes AI-driven image captioning to give personalized feedback based on the user's facial expressions and speech recognition for voice-based interaction in certain games. By combining entertainment with targeted educational strategies, this study examines how serious games can support the development of cognitive and linguistic skills in children with diverse learning needs, and how AI technologies can facilitate individualized experiences, elevate student interest, and ultimately enhance learning outcomes.

Keywords

Artificial Intelligence, Education, Educational Games, Special Needs, Speech Recognition, Interactive, Machine Learning, Natural Language Processing, Unity

Contents

1 Introduction

In recent years, technological innovation has become a key driver in the pursuit of more inclusive and effective education. Digital tools are now widely integrated into classrooms, not only to modernize teaching methods but also to better respond to the diverse needs of learners. This is particularly important for children with special educational needs, who often face challenges in areas such as speech, attention, or social interaction. In this context, the combination of serious games and artificial intelligence presents new opportunities to design adaptive and motivating learning environments.

The theoretical foundation will delve into how serious games can serve as powerful tools to support the development of children with special needs. It will first highlight how these games can address specific learning challenges by providing engaging, interactive experiences tailored to the needs of these children. Then, the discussion will continue exploring the evolution of artificial intelligence and its role in enabling more adaptive and responsive educational interactions. Additionally, the text will provide a comprehensive overview of both game engines and backend development, offering insight into the technical processes involved in creating an engaging, accessible, and user-friendly learning interface.

The educational experience developed in this project revolves around a mobile application structured as a world-travel journey, where children progress through different mini-games designed to stimulate key cognitive and linguistic skills. Each activity has been carefully designed to support learning through repetition, interaction, and positive reinforcement. The integration of AI enriches this experience, transforming the application into an intelligent companion capable of responding to the user's voice and emotional cues in a supportive way. Using speech recognition, the app enables interactive voice-based gameplay, helping children learn how to pronounce vowels and simple words. In addition, image recognition analyzes the child's emotions and provides personalized feedback through speech, making the experience more engaging and emotion-aware.

This application aspires to offer children with special needs a novel, interactive way to develop their cognitive and linguistic skills. By adapting to each child's pace and providing personalized, real-time feedback, the app aims to foster a more inclusive and dynamic learning environment where each child can progress at their own rhythm while feeling supported and motivated throughout their educational journey.

2 Educational Games and Inclusive Learning

2.1 Introduction to Educational Games

Educational games are a subset of Serious Games, video games or interactive applications designed to go beyond mere entertainment by incorporating learning elements. These games are used in different fields, such as healthcare, marketing and education, to enhance knowledge retention, improve cognitive abilities, and provide practical experience in an engaging and interactive way. (Peña-Miguel & Sedano 2014, 230-231.)

Figure 1 illustrates an example of an educational game that fosters learning languages by immersing players in dynamic challenges.

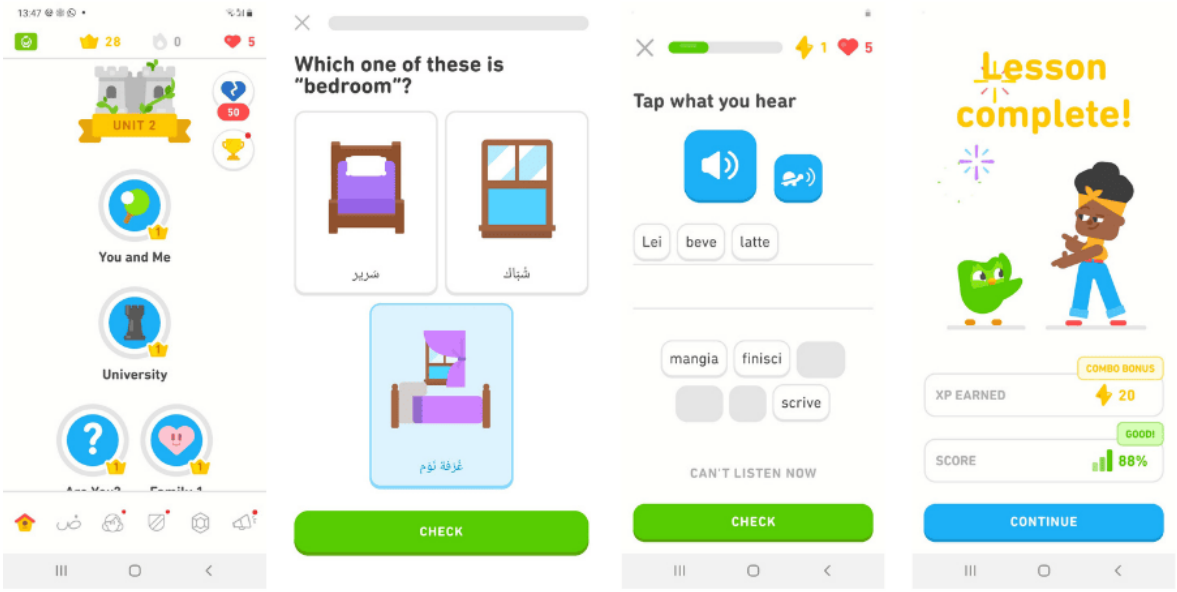

Figure 1. Duolingo allows you to learn languages through play (Marie-Astrid 2022)

A key aspect of these games is their ability to maintain user engagement and motivation by adapting the gameplay experience to individual skill levels (Peña-Miguel & Sedano 2014, 230-231). One effective strategy to achieve this is through gamification, which involves integrating familiar game mechanics from traditional video games. By leveraging elements such as rewards, challenges, and progression systems, these games become more immersive and stimulating, encouraging users to stay focused and actively participate in the learning process. (Lane.)

Additionally, educational games feature a unique feedback system that distinguishes them from traditional learning environments. Unlike conventional methods, where feedback may be delayed or threaten the student, these games offer real-time, immediate responses to

user's decisions. This instant feedback helps learners understand their mistakes and reinforces a growth mindset, encouraging them to see errors as opportunities for learning rather than failures. As a result, these games create a low-pressure environment where players feel more comfortable experimenting and improving without fear of negative consequences. (Arterburn et al. 2022, 1-4.)

Furthermore, serious video games naturally incorporate challenges, making them an ideal platform for fostering perseverance and problem-solving skills. When games reward effort, strategy, and incremental progress, players become more persistent and develop more effective problem-solving strategies. By promoting a mindset that values effort over immediate success, these games help learners build resilience and confidence in tackling difficult tasks, skills that can extend beyond the game into real-world learning scenarios. (Arterburn et al. 2022, 1-4.)

Numerous studies have examined the impact of gamification in education, highlighting its effectiveness in enhancing motivation, engagement, and learning outcomes. For instance, a research conducted at the National Technical University of Athens evaluated a gamified learning application for statistics and reported that students who used the application outperformed their peers in traditional lecture-based classes by 89.45%. Furthermore, they improved their own performance by 34.75% compared to their previous results, demonstrating the potential of challenge-based gamification in academic settings. (Verma 2023.)

As the digital generation finds educational games familiar and practical, gamification will continue to play a crucial role in the future of education, providing innovative and interactive methods to make learning more attractive and enjoyable (Verma 2023).

2.2 Learning Challenges in Special Education

Neurodevelopmental disorders are a group of conditions that originate during the early stages of brain development and can impact various cognitive, behavioural, and social functions. These disorders often manifest in childhood and can affect essential skills such as attention, memory, language, perception, and problem-solving. They may also influence social interactions and adaptive behaviours, making everyday tasks more challenging. Affecting between 5% and 10% of the population, they typically emerge before adolescence. (Chappotin 2022; Brian 2024.)

The Diagnostic and Statistical Manual of Mental Disorders (DSM-5), classifies several neurodevelopmental disorders that impact cognitive, social, and motor functions from early childhood (Asociación Americana de Psiquiatría 2013). These include:

- Autism Spectrum Disorder (ASD): Involves persistent deficits in social communication and interaction, difficulty in understanding social-emotional cues, restricted interests, repetitive behaviours, and sensory sensitivities. Individuals often struggle with adapting behaviours to different contexts (Asociación Americana de Psiquiatría 2013).
- Attention-Deficit/Hyperactivity Disorder (ADHD): Marked by persistent inattention, hyperactivity, and impulsivity. It affects the ability to sustain attention, organize tasks, and regulate impulses, often leading to academic and social challenges (Asociación Americana de Psiquiatría 2013).
- Communication Disorders: Includes difficulties in language development, speech production, fluency (such as stuttering), and social communication, affecting both verbal and nonverbal interactions (Asociación Americana de Psiquiatría 2013).
- Motor Disorders: Encompasses coordination disorders, stereotypic movement disorders characterized by repetitive and purposeless movements, and tic disorders such as Tourette syndrome, which is characterized by both motor and vocal tics (Asociación Americana de Psiquiatría 2013).
- Specific Learning Disorders: Includes dyslexia, which involves difficulty with word recognition and reading comprehension; dysgraphia, characterized by challenges with writing and spelling; and dyscalculia, which refers to difficulty in understanding mathematical concepts and operations (Asociación Americana de Psiquiatría 2013).
- Intellectual Disability: Characterized by limitations in intellectual functioning and adaptive behaviour, affecting reasoning, problem-solving, and abstract thinking. Severity ranges from mild to profound, with more severe cases impacting personal autonomy and social responsibility (Asociación Americana de Psiquiatría 2013).

2.3 Serious games in Special Education

Digital game-based learning has proven effective for individuals with intellectual and sensory disabilities, improving skills such as memory, decision-making, and spatial awareness. According to a survey conducted by Futurelab and reported by Williamson in 2009, involving over 1,600 classroom teachers in English state primary and secondary schools, a significant majority believe that computer games support children's motor and cognitive development with 83%, enhance ICT skills with 73%, and foster higher-order thinking skills like logical thinking, planning, and strategizing with 65%. This highlights the broad potential of digital

games in promoting various cognitive and developmental abilities. (Papanastasiou et al. 2022, 41-42.)

An example of this innovative approach is Paths (Figure 2), a digital game designed created by Ombretta et al. specifically for children with dyslexia. In the game, the child is tasked with drawing a path between two objects, an activity that encourages the development of spatial awareness, memory, and problem-solving skills. By engaging children in a fun and interactive way, Paths supports their cognitive growth while addressing specific learning challenges associated with dyslexia. (Ombretta et al. 2017.)

Figure 2. Paths game (Ombretta et al. 2017)

This innovative approach offers a promising educational resource specifically tailored for individuals with intellectual disabilities, providing them with opportunities to learn in a supportive and enjoyable environment. By focusing on their strengths and interests, this method fosters a sense of autonomy and encourages personal growth, making it a valuable addition to educational strategies aimed at enhancing the development of individuals with diverse learning needs. (Papanastasiou et al. 2022, 49.)

2.4 Enhanced AI Educational Games

Enhanced AI gamification is revolutionizing learning by adapting to users' needs and learning styles and creating a more personalized experience. Through machine learning and dynamic game environments, users are no longer passive participants but active learners

engaged in interactive experiences. This innovative approach is making a significant impact across various sectors, from education and corporate training to industries like healthcare and technology. (Tipping Point Media 2024.)

In Figure 3, results of a survey conducted by Forbes Advisor in October 2023 can be seen, showing how 500 educators across the U.S. perceive the impact of AI in the classroom.

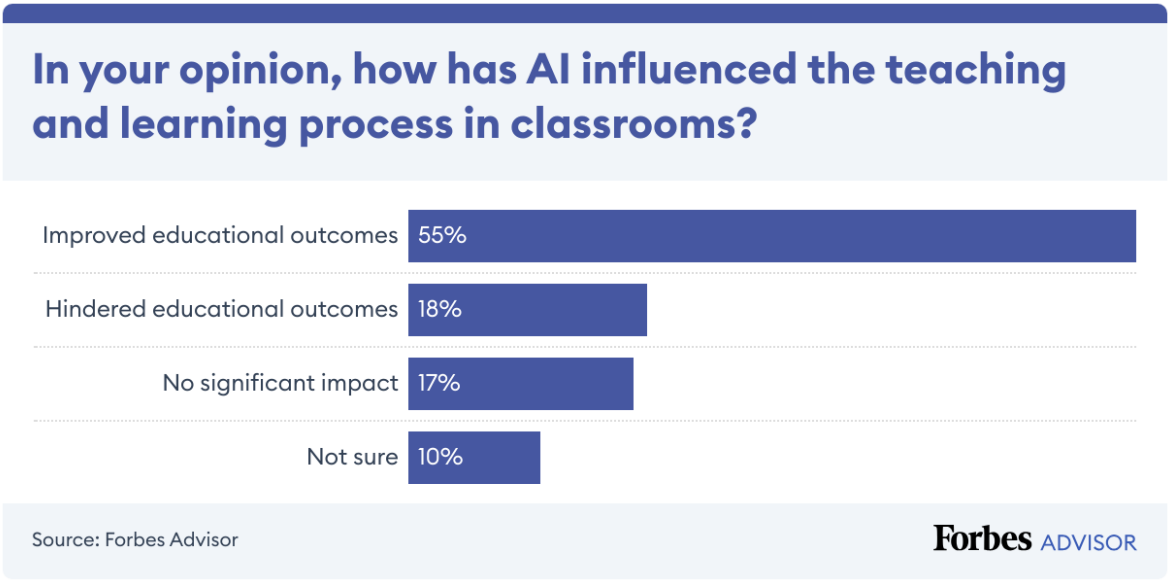

Figure 3. How has AI influenced the teaching and learning process in classrooms? (Forbes 2024)

AI can be integrated into education in a lot of different ways, potentially enhancing learning by personalizing experiences, adapting content to individual needs, and providing real-time feedback. It could support students through virtual assistants, help streamline administrative tasks like grading, and optimize tutoring by adjusting to each learner's progress. (Hamilton 2024.)

With Natural Language Processing (NLP), AI is transforming education, enabling more interactive, efficient, and personalized learning experiences. One of its main applications is interactive conversational agents, which act as virtual mentors that provide real-time explanations, answer student queries, and simulate real-world conversations. (Medewar 2023.)

NLP also plays a crucial role in assessment and feedback by automating grading and generating constructive responses. By analyzing syntax, semantics, and content, these systems ensure fairness and uniformity in evaluation. (Medewar 2023.)

Additionally, NLP enhances multilingual learning through advanced translation tools, offering accurate translations that make education more inclusive for different cultures. Integrating speech recognition and interactive language practice also supports language learning, helping students improve pronunciation and fluency through personalized feedback. (Medewar 2023.)

3 Artificial Intelligence

3.1 Overview

Artificial Intelligence is the science and engineering of making intelligent machines, especially intelligent computer programs (McCarthy 2007, 2-7). However, a comprehensive understanding of this concept requires first establishing a clear definition of intelligence.

Intelligence is the capacity to acquire and apply knowledge. Additionally, it includes the ability to benefit from past experience, act purposefully to solve problems and adjust to new situations. (Gupta & Mangla 2020, 3.)

Human intelligence, according to the theory of multiple intelligences, can be categorized into seven distinct types: linguistic, logical-mathematical, musical, spatial, bodily-kinesthetic, interpersonal, and intrapersonal. The objective of artificial intelligence (AI) is to achieve a comprehensive understanding of human intelligence and develop technological systems that simulate, enhance, and advance its capabilities. (Huawei Technologies Co. 2022, 1-3.)

Artificial intelligence is closely integrated with a number of related fields, such as machine learning, deep learning, data science, big data, and data analytics. These disciplines complement and extend AI's capabilities, enabling more advanced and effective intelligent systems. (Huawei Technologies Co. 2022, 1-3.)

A detailed discussion of several of these fields will follow in subsequent sections of this document. An overview of these fields is shown in Figure 4.

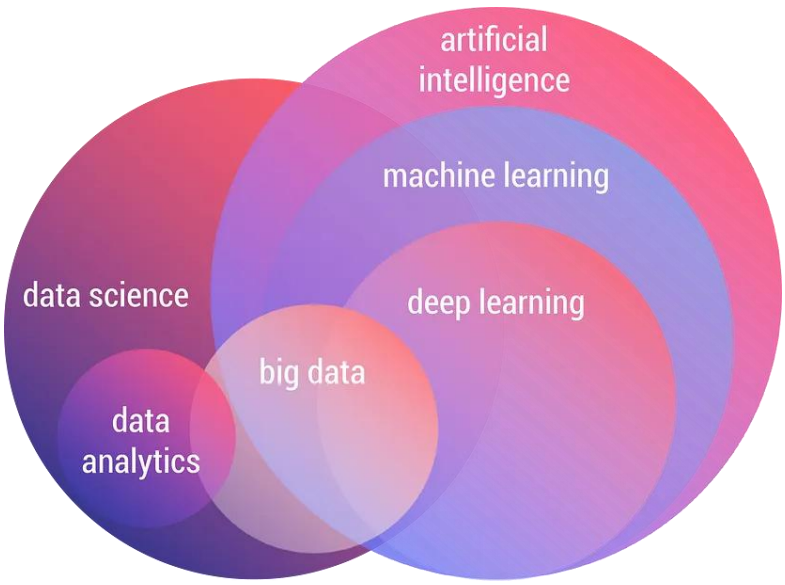

Figure 4. The Taxonomy of Artificial Intelligence and Data Science (Beam 2022)

3.2 History

From its beginning, the development of AI has been influenced by ideas from different disciplines, such as engineering, biology, physiology, communication theory, game theory, mathematics, and statistics. The first major turning point in the history of AI is Turing's seminal 1950 paper in the philosophy journal Mind. (Buchanan 2005, 4-8.)

In 1950, Alan M. Turing, often regarded as "the Father of Artificial Intelligence," proposed a fundamental question: "Can machines think?" In his paper, he introduced what is now known as "The Imitation Game" or "Turing Test", illustrated and explained in Figure 5 (Turing 1950; Oppy & Dowe 2003).

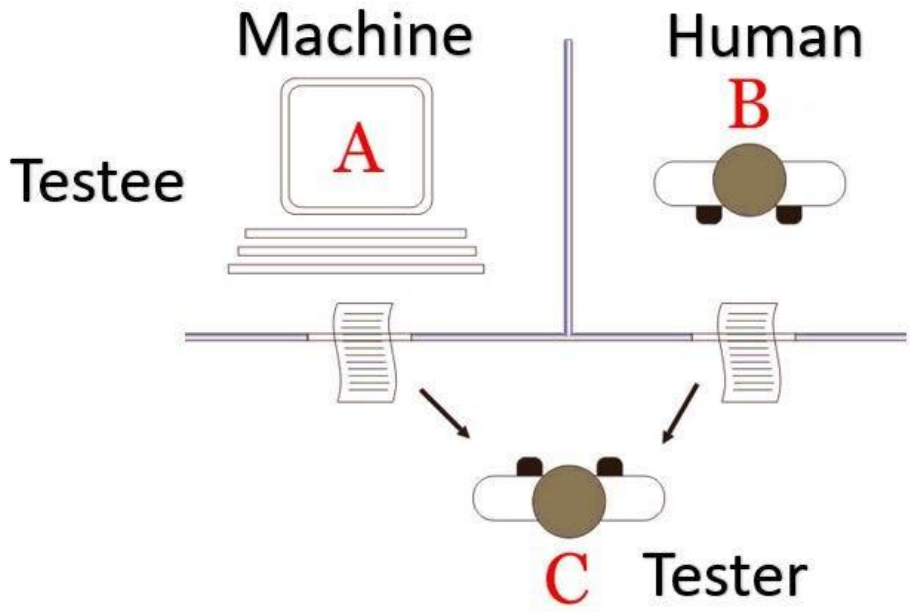

Figure 5. The setup of the Turing test: both Machine and Human try to convince the Tester that they are the "real" intelligence, by exchanging messages (Stefan 2023)

The test works as follows: there are 3 participants: a machine, a test subject, and a tester. The tester's role is to determine which of the other two participants is the test subject and which is the machine (Oppy & Dowe 2003). Turing believed that within 50 years, advancements in computer technology would enable machines to perform "The Imitation Game" so convincingly that an average human tester would have, at most, a 70% chance of correctly distinguishing between a human and a machine after five minutes of questioning (Turing 1950). His prediction was nearly accurate, as the first successful completion of the test occurred in 2014, when a Russian computer program, *Eugene Goostman*, passed the test at the Royal Society in London (University of Reading 2014).

Following this proposal, the first major boom in artificial intelligence occurred during the 1960s and 1970s. It was then the time when AI was finally introduced as a discipline by John McCarthy in the 2-month long Dartmouth Conference in 1956. This encouraged researchers like Allen Newell, Marvin Lee Minsky, and Herbert Alexander Simon to form AI-specialised research groups. (Huawei Technologies Co. 2022, 7.)

Over these two decades, AI developed rapidly across a range of fields. This period saw the birth of “Machine Learning” with Arthur Samuel’s (Figure 6) checkers-playing program, which employed “alpha-beta pruning” to evaluate which positions were likely to win and make decisions to win. Additionally, early “Pattern matching” techniques emerged, remarkably with ELIZA, the first conversation program which was able to respond with pre-written answers to user’s questions. (Sutton & Barto 2014, 270-272; Huawei Technologies Co. 2022, 7.)

Figure 6. The principles Samuel developed laid the groundwork for a series of breakthroughs in artificial intelligence at IBM during the 1990s (IBM)

Despite the optimistic predictions about the future development of AI, the technology was not sufficiently advanced yet, so this led to a disillusioned period between the 1970s and 1980s known as “the first AI winter” (Huawei Technologies Co. 2022, 8-9). Although the problems faced by machines at that time were relatively simple, emulating human behaviour requires considerable background knowledge, much like we do. Consequently, substantial

processing power and storage capacity were necessary, capabilities that the technology of the era could not provide. (Warwick 2012, 4-6.)

As if the technological challenges were not enough, many philosophers got interested in the topic and began criticizing the limitations of AI development and future capabilities. McCarthy continued to develop and inspire other researchers to persevere despite both the technological and conceptual obstacles. (Warwick 2012, 5-6; Huawei Technologies Co. 2022, 9.)

From 1980 onwards, technology began to improve, and the interest in AI began to rise again. In 1997, IBM's Deep Blue program made history as the first computer to defeat Garry Kasparov, the reigning world chess champion. The potential of AI was rediscovered, and this encouraged developers and researchers to explore its capabilities and integrate them into industrial applications. (Warwick 2012, 6-8.)

The 2000s brought lots of advancements in the field of AI. One notable milestone was Cynthia Breazeal's Kismet, a social robot capable of interacting with humans through emotional and social cues using facial recognition. Another breakthrough came with the development of AlexNet, a convolutional neural network (CNN) that could achieve a remarkable image classification error rate of less than 16%. Later on, Sophia (Figure 7), a highly advanced humanoid robot, demonstrated the ability to recognize faces, maintain eye contact, and participate in conversations through a combination of image recognition and natural language processing, surpassing Kismet's capabilities. (Mucci 2024.)

Figure 7. Sophia was made by Hanson Robotics (Balibouse 2017)

Since the 2010s, AI has seen substantial growth, and the development of deep learning architectures has enabled models like OpenAI's GPT-4 and DALL-E to revolutionize text

and image creation. Meanwhile, virtual assistants like Apple's Siri have transformed how humans interact with technology through natural language processing. (Mucci 2024.)

3.3 Types of AI

Artificial Intelligence is constantly evolving, but generally, it can be classified based on its capabilities or functionality.

Based on capabilities:

- Artificial Narrow AI: It is commonly known as Weak AI, and it is the only type of AI that currently exists. Any other types remain purely theoretical. This type of AI can be trained to perform a narrow task, and may even do it better than humans. However, it cannot perform outside of what it has been trained for. Instead, it focuses on a specific subset of cognitive abilities and develops within that domain. (IBM 2023a.)
- General AI: It is commonly known as Strong AI, and it remains a theoretical concept. This form of AI has the ability to use previous learnings and skills to perform new tasks in a different context without requiring human involvement in retraining the underlying models. They will have the ability to think abstractly, strategize, and access previous knowledge to make informed decisions or generate creative ideas, just as humans can. (Jajal 2018; IBM 2023a.)
- Super AI: It is commonly referred to as superintelligence and is also strictly theoretical. It embraces the idea of machines possessing cognitive abilities that surpass human intelligence. Therefore, being able to go beyond simply understanding human emotions and experiences, to the extent of experiencing emotions, having needs, and forming their own beliefs and desires. (IBM 2023a.)

Classification is shown in Figure 8.

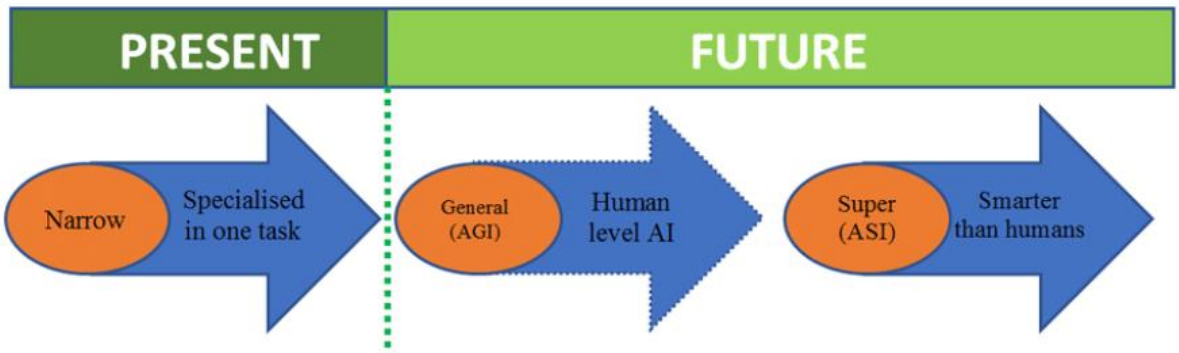

Figure 8. Types of Artificial Intelligence (AI) and their progression (Hardwood et al. 2019)

Based on functionalities:

- Reactive Machine AI: Such systems are trained to perform simple and specific tasks. They do not retain information from past experiences, so they operate solely based on the data available at the moment. Additionally, they rely on statistical mathematics to analyze large volumes of data, generating outputs that simulate intelligent answers. An example of this is the Netflix Recommendation System, which, based on users' viewing history, suggests content they are most likely to enjoy. (IBM 2023a.)

- Limited Memory AI: Unlike the previous type, this form of AI can retain past experiences data or outcomes for a specific amount of time and use it in order to make more informed decisions. As it learns and adapts, its performance improves over time. Generative AI's such as ChatGPT and Bard are examples of this type of technology. (IBM 2023a.)

- Theory of Mind AI: This theory is a significantly more advanced form of AI which remains non-functional today. It is able to comprehend emotions and thoughts, enabling it to give personalized user experiences through unique and adaptive interactions. Additionally, unlike today's generative AI tools, it also possesses capability to interpret and contextualize artwork and essays. Emotion AI, a subset of Theory of Mind AI, is currently under development, and its goal is to respond to humans on an emotional level by analyzing voices, images, and other types of data. However, it has yet to fully achieve this capability. (IBM 2023a.)

- Self-Aware AI: Surpassing Theory of Mind AI, this form of AI would not only comprehend human emotions and thoughts but also possess the ability to understand its own internal conditions and feelings. Furthermore, it would develop its own unique set of emotions, needs, and beliefs. This type of AI is anticipated to surpass human intelligence. (IBM 2023.)

3.4 Machine Learning

Machine learning is a subset of artificial intelligence dedicated to developing systems that can learn and enhance their performance through data analysis. Unlike traditional statistical methods, machine learning uses powerful algorithms to identify complex relationships within large datasets, allowing for more adaptive and efficient decision-making. (Chen 2024.)

Traditional programming requires explicitly writing instructions for a computer to follow. However, for certain complex tasks, this approach becomes impractical or time-consuming. Instead of being manually programmed, machine learning enables computers to recognize patterns in data and generate their own models, making it possible to automate tasks that would otherwise be difficult to program explicitly. (Chen 2024.)

The difference between traditional methods and machine learning is illustrated in Figure 9.

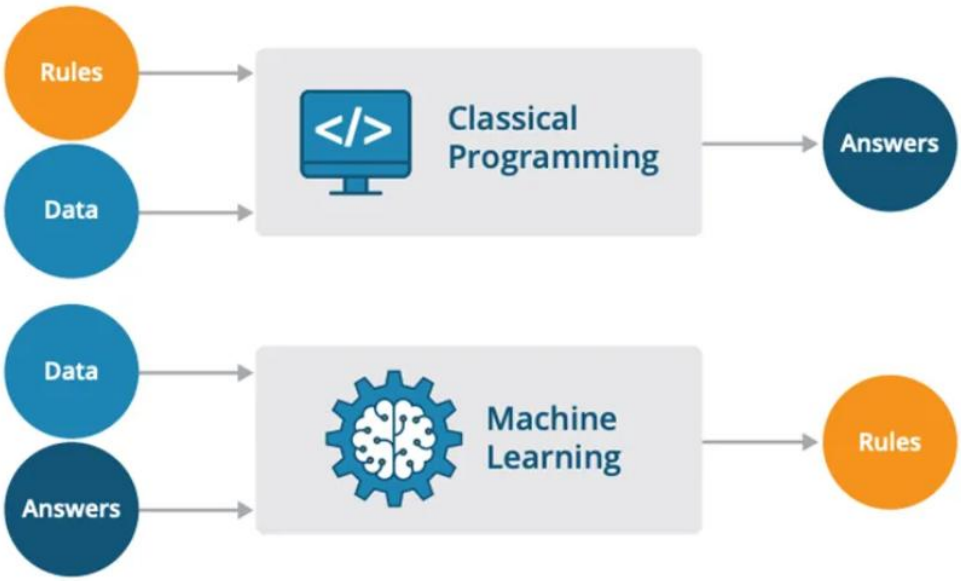

Figure 9. Traditional Programming vs Machine Learning (Reddysetty 2021)

3.4.1 Approaches

Machine Learning has different approaches depending on how the models are trained. Each approach has its own strengths and limitations.

Supervised machine learning is a technique that uses labeled data to train an algorithm to predict outcomes or classify data. The system is provided with data associated with known outcomes, or labels. By analyzing these examples, the algorithm has to work on understanding the relationship between inputs and outcomes, enabling it to make predictions on new data. Finally, the model's accuracy is evaluated using test data, with a loss function that measures the discrepancy between its predictions and the actual outcomes. This approach is the most common one, but it requires human supervision in order to provide the correct answers by labeling the data. (Chen 2024; Belcic & Stryker 2024.)

In contrast, unsupervised machine learning is a more independent approach and it does not need human intervention. The system learns to identify patterns within unlabeled data on its own. It usually works by detecting groupings of similar data, forming clusters with them,

and once trained, it is able to recognize patterns and assign new unseen data to the appropriate group. This technique is used in cases like analyzing customer purchases to identify different types of clients and make new recommendations for products to them. (Brown 2021; Chen 2024.)

In situations where a vast amount of training data is available but resources for labeling are limited, a combination of supervised and unsupervised machine learning can be utilized to achieve a fully trained model. This technique is known as semi-supervised machine learning. The training begins similarly to supervised learning, utilizing the labeled data to generate the initial predictions and establish guidelines for the algorithm. Then, when the labeled data is used, the model is given the unlabeled dataset. The model analyzes the unlabeled data and attempts to assign labels; if the sample is classified with sufficient certainty, it is added to the labeled data. The process repeats iteratively, expanding the labeled data with what are commonly referred to as pseudo-labels. Through continuous refinement, the model improves its accuracy and generalization capabilities. (Chen 2024.)

Reinforcement machine learning uses unlabeled datasets, but unlike the previous approaches, it does not explore the data to find patterns. Instead, it works toward a specific set goal, and it applies a trial-and-error process where the algorithm receives positive or negative feedback based on its decisions. This feedback helps refine the algorithm’s decision-making over time. Reinforcement machine learning is mainly used for complex and dynamic situations because it allows the context of the project goal to influence decision-making, even if it involves short-term negative consequences. By continuously adjusting its strategy based on feedback, the algorithm works toward long-term success, making it particularly effective in scenarios like teaching a computer how to play chess, where sacrificing pieces may be necessary to win the game. (Chen 2024.)

3.4.2 Learning process

As explained earlier, machine learning creates statistical models by analyzing and refining training data. These models improve through exposure to data, adjusting their accuracy using a series of established variables known as hyperparameters and algorithmically adjusted variables, known as learning parameters. As more data is processed, the algorithm continuously learns, becoming more effective in making predictions. (Chen 2024.)

Most machine learning projects follow a series of essential steps, which can vary in complexity depending on the project's scope and requirements.

- Gathering and compiling data: The foundation of any ML project lies in acquiring high-quality data (Chen 2024).

- Selecting an appropriate algorithm: A suitable algorithm is chosen depending on the project's nature. Various machine learning algorithms are available, including neural networks, linear and logistic regression, clustering methods, random forests, and decision trees. Selecting the right one ensures optimal model performance and accuracy. (Chen 2024.)
- Refining and preparing data: Incoming data typically requires cleaning and transformation to standardize formats and ensure quality. This process includes tasks such as handling duplicates and outliers to optimize data for training. (Chen 2024.)
- Educating the model through training: Training involves feeding curated datasets into the chosen algorithm and refining performance iteratively (Chen 2024).
- Assessing model performance: After the training, the model undergoes testing with unseen data to measure accuracy and performance metrics (Chen 2024).
- Fine-tuning model parameters: Once the tests are done, optimization continues through parameter adjustments and additional training with specific datasets, as necessary (Chen 2024).
- Launching the model: Once optimized, the model is deployed for real-world use, where ongoing monitoring and performance audits ensure its effectiveness and alignment with desired outcomes (Chen 2024).

A similar Machine Learning workflow can be seen in Figure 10.

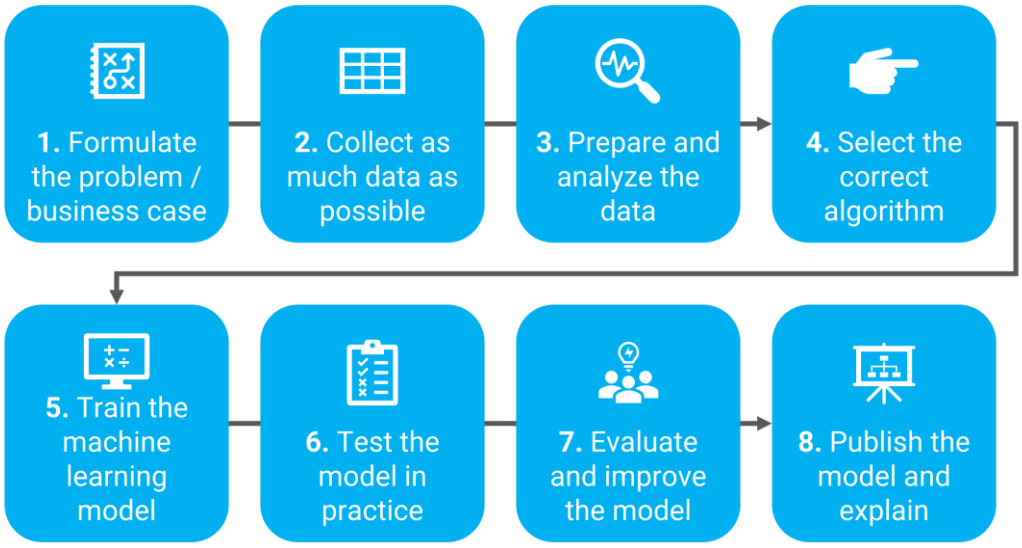

Figure 10. A roadmap for developing machine learning models and putting them into production (Van Beek 2025)

3.4.3 Training issues

Training a model usually requires significant computational power, including high-performance processors and extensive memory. Since not all organizations possess such capabilities, cloud platforms often provide the necessary infrastructure for efficiently storing and processing large volumes of data. (Kolena 2024.)

Another common issue while training is data bias. When the training data does not accurately represent the broader population, the result of the training can lead to biased predictions, affecting the model's overall reliability (Saxena 2024). For example, if a model has been trained with data that mainly reflects a specific demographic, its predictions may be skewed in favor of that group. Well-balanced datasets and preprocessing techniques such as resampling strategies and feature selection are crucial to address data bias and lead to more reliable and impartial model predictions. (Kolena 2024.)

In addition, AI models often face challenges such as overfitting and underfitting, both of which can significantly impact their performance. Overfitting occurs when a model becomes too specialized in recognizing patterns from its training data, making it less effective at handling unseen information, generating erroneous results and decreasing accuracy. To prevent this issue, techniques such as regularization, early stopping and cross validation in training are usually used. (Chen 2023; Kolena 2024.)

On the other hand, underfitting happens when a model is not sufficiently trained, causing it to produce accurate results only in highly specific scenarios. This issue often arises when the model is too simplistic or when the training data is not adequately prepared accuracy. (Chen 2023.)

A graphic representation of these two last issues can be seen in Figure 11.

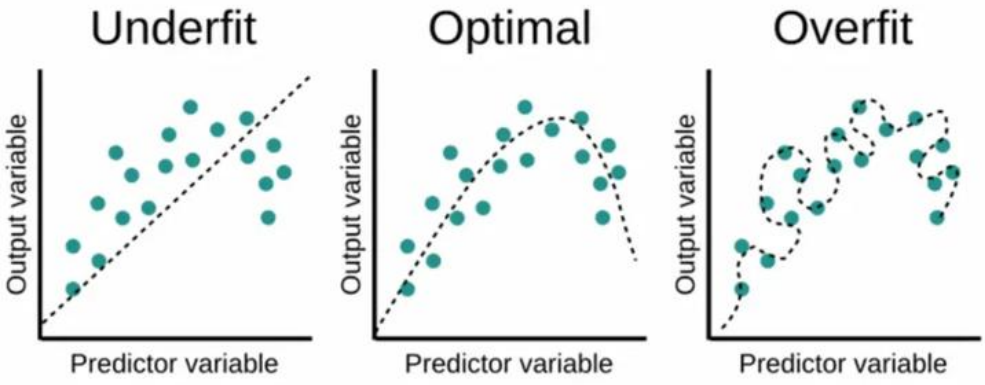

Figure 11. Overfitting and Underfitting (Bhandari 2022)

3.5 Deep Learning

To understand Deep Learning, we must first define what neurons and neural networks are in the context of mathematics and machine learning.

An artificial neuron (Figure 12) is a mathematical function that processes one or more input values by multiplying them with specific weights and summing the results. The sum is then passed to a non-linear function, known as the activation function, which determines the neuron's final output. (Ronaghan 2018.)

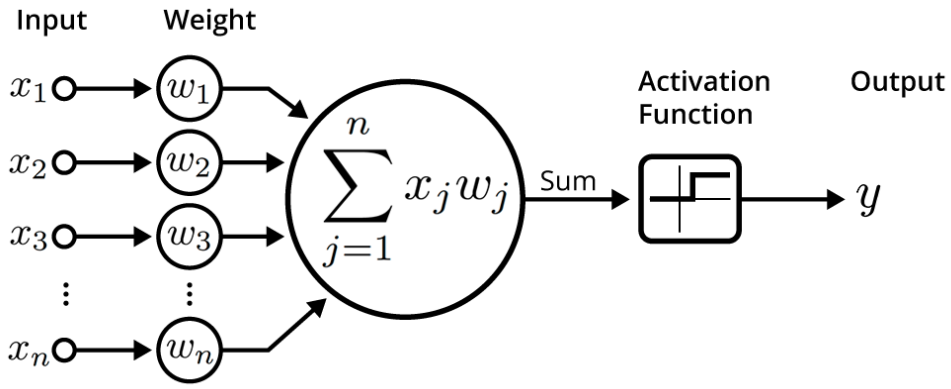

Figure 12. An illustration of an artificial neuron (McCullum 2020)

Neural networks consist of layers of artificial neurons connected in a structured way, including an input layer, one or more hidden layers, and an output layer. Figure 13 shows this structure where each neuron is linked to others and has an associated weight and threshold. When a node's output exceeds its threshold, it becomes activated and passes data to the next layer; otherwise, no data is transmitted forward. (IBM 2021.)

By processing large amounts of training data, neural networks continuously learn and improve their accuracy. This ability makes them particularly effective for tasks such as image and speech recognition, where they can quickly and efficiently analyze complex patterns. (IBM 2021.)

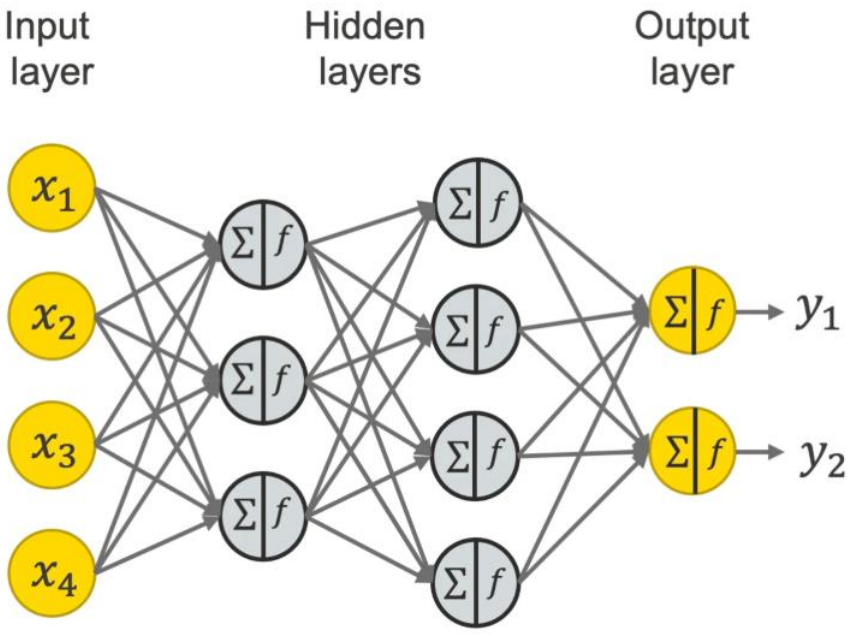

Figure 13. Example of a fully connected, feedforward neural network (Melcher 2021)

Once the fundamental concepts are understood, deep learning can be defined. It is a subset of machine learning that utilizes multilayered neural networks, known as deep neural networks, to replicate the complex decision-making capabilities of the human brain (Holdsworth & Scapicchio 2024).

Deep neural networks are composed of multiple layers of interconnected neurons, where each layer builds upon the previous one to refine and optimize predictions or classifications. This process is known as forward propagation, and it enables the network to transform raw input data into meaningful predictions. (Holdsworth & Scapicchio 2024.)

To enhance accuracy, deep neural networks undergo a training process called backpropagation. This method adjusts the connections between neurons by identifying errors in predictions and fine-tuning the network's internal parameters. Through iterative learning, the model continuously improves its performance. (Holdsworth & Scapicchio 2024.)

3.6 Transformers

Transformers were first described in Google's 2017 paper "Attention is All You Need", and they are one of the most powerful types of nowadays models. They apply an evolving set of mathematical techniques, known as attention or self-attention, to identify complex relationships between data elements, even when they are far apart within a sequence. (Merritt 2022.)

Sequence-to-Sequence is a type of neural network that transforms a given sequence of elements into another. These models follow an Encoder-Decoder architecture, where the Encoder processes the input sequence and converts it into a higher dimensional representation. This encoded information is then passed to de Decoder, which generates the corresponding output sequence. (Maxime 2019.)

Attention mechanisms, inspired by the way humans focus on important details while ignoring less relevant ones, enhance efficiency in processing large amounts of data without losing critical information. They assign attention weights to different parts of a sequence, adjusting their influence based on relevance to the task. (Bergmann & Stryker 2024.)

Transformers build upon the Encoder-Decoder architecture commonly found in sequence-to-sequence models and incorporate attention mechanisms to improve efficiency. Unlike previous neural networks, it does not require sequential dependency, enabling greater parallelization and faster training. By leveraging attention mechanisms, Transformers can handle complex sequence tasks more effectively, making them a powerful alternative to traditional RNN-based approaches. (Kalra 2024.)

3.7 Natural Language Processing

A language can be understood as a collection of sentences, each composed of a limited set of symbols or characters that come from a finite alphabet. Since both the set of symbols and the sentence length are finite, the total number of possible sentences within a language is also finite. In many theoretical studies, however, we treat languages as infinite, allowing for sentences of any size. But when focusing on languages used for machines, we typically limit the language to a finite set of symbols and sentence lengths. This limitation arises because real-world machines, with finite memory and processing capabilities, can only handle a limited amount of data at a time. (Chowdhary 2020, 603-604.)

Natural Language Processing is a branch of Artificial Intelligence aimed at enabling computers to understand and interpret human languages. As shown in Figure 14, NLP can be divided into two main components: Natural Language Understanding, which includes phonology, morphology, pragmatics, syntax and semantics, and Natural Language Generation, which focuses on the creation of text. (Khurana et al. 2023, 3714-3715.)

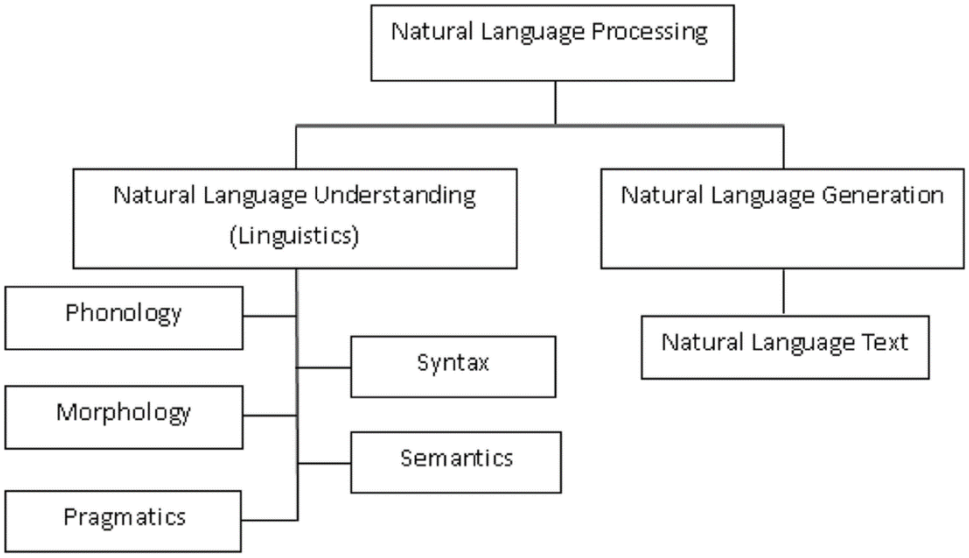

Figure 14. Broad classification of NLP (Khurana et al. 2023, 3715)

3.7.1 Workflow and techniques

In a complete NLP system, the process begins with text pre-processing, where raw text is cleaned and standardized to ensure that machines can effectively analyze it. This involves applying various techniques to structure the data for better interpretation. (Stryker & Holdsworth 2024.)

One of the fundamental techniques is tokenization, which consists of breaking the text into smaller units such as words, sentences, or phrases, making it easier to manage and interpret (Stryker & Holdsworth 2024). Various approaches can be used depending on the task's specific requirements. Some methods include splitting text based on whitespace or punctuation. In contrast, others incorporate context and semantics for a more nuanced understanding (Wisdom 2023).

Another essential process is stemming and lemmatization, which refine words into their base forms. While stemming reduces words to their root structure, lemmatization maps them to their dictionary forms. Additionally, stop word removal is often performed to eliminate common but non-essential words like "the," "is," and "and," allowing the model to focus on more meaningful content. These processes transform the raw text into a structured format. (Wisdom 2023.)

To further enhance understanding, part-of-speech (POS) assigns grammatical labels to each token based on its role in a sentence. This technique relies on a combination of linguistic rules and statistical algorithms to determine the correct tag for each word. Linguistic

rules analyze contextual cues, such as surrounding words, to enhance accuracy. By identifying the relationships between words and their functions within a given context, POS tagging helps machines better understand sentence structure and meaning. (Wisdom 2023.)

Finally, feature extraction techniques play a crucial role in transforming raw text into numerical formats that machines can process effectively. One of the most advanced approaches is word embeddings, which map words into a multidimensional space, capturing their relationships (Stryker & Holdsworth 2024). This method enables models to identify intricate patterns in language far beyond human perception. Instead of treating words as independent entities, word embeddings allow AI systems like ChatGPT to recognize connections between terms and categories, improving their ability to understand context and meaning. (Barnard 2023.)

Embedded models transform objects into embeddings, which are numerical representations typically in the form of vectors. A vector consists of an array of numbers, where each number corresponds to the position of an object along specific dimensions within a high-dimensional space. The number of dimensions can vary but often extends to a thousand or more, depending on the complexity of the input data. (Barnard 2023.)

An example of a representation of words classification using word embeddings is illustrated in Figure 15.

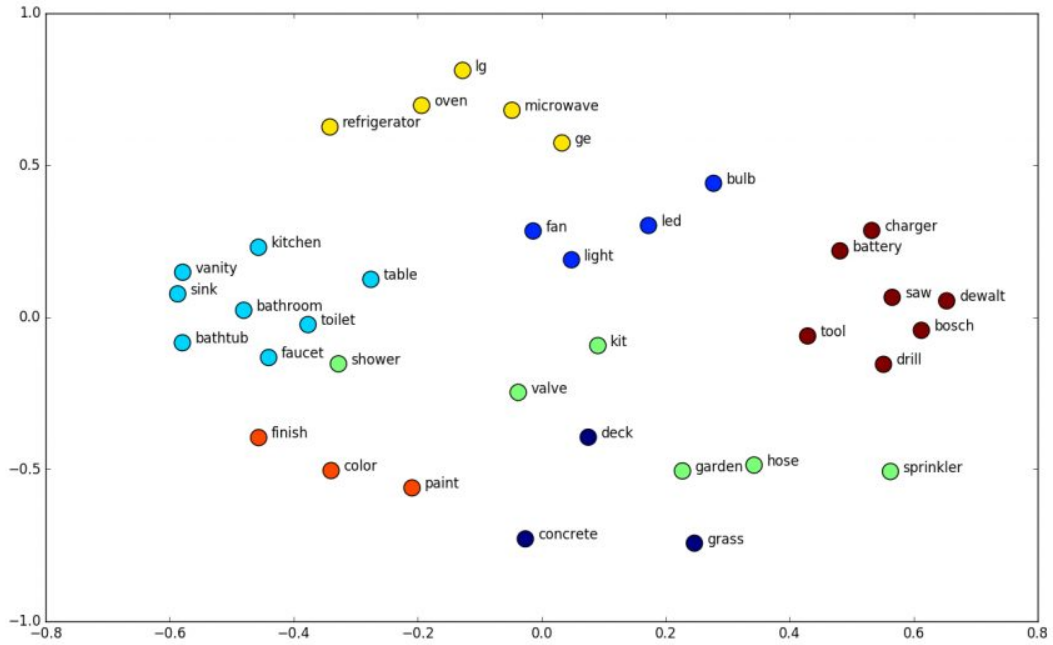

Figure 15. Example 2D word embedding space, where similar words are found in similar locations (Lynn)

The similarity between embeddings is gauged by their proximity in this n-dimensional space. The closer two embeddings are spatially, the more similar their respective objects are considered to be. Measures like Euclidean distance, cosine similarity, or other metrics quantify this proximity, reflecting how closely related or alike the embedded objects are in distribution. (Barnard 2023.)

After all these techniques have processed the data, machine learning models are trained on it, learning from patterns to enhance their ability to classify, predict, and generate text. These models undergo continuous refinement through evaluation and fine-tuning, making natural language processing (NLP) a powerful tool for applications such as chatbots, translation, sentiment analysis, and text summarization. (Stryker & Holdsworth 2024.)

As the volume of text data continues to expand every day, NLP presents a valuable opportunity to effectively interpret this information and enhance a wide range of applications. It has become essential in daily life, enabling multilingual communication through tools like Google Translate and powering virtual assistants such as Alexa and Google Assistant for accurate language processing. It is transforming industries by helping digital marketers analyze customer sentiments for personalized engagement and revolutionizing customer service in finance with AI chatbots that provide quick, human-like responses. (Lee 2019.)

3.7.2 Large Language Models (LLMs)

Large language models (LLMs) are advanced neural networks that process and understand human language. Trained on massive datasets, they can perform a wide range of natural language processing (NLP) tasks, such as translating languages, generating text, and summarizing information. The large number of parameters in language models enables them to recognize complex patterns in text. This capability allows them to manage extensive vocabulary and generate coherent, high-quality responses. (Linares et al. 2023.)

In recent years, large language models (LLMs) have transformed natural language processing (NLP), making sophisticated language processing tools more accessible to the general public. Several prominent models have emerged, as can be seen in Figure 16, including OpenAI’s ChatGPT-3 and GPT-4, which have gained widespread popularity with support from Microsoft. Similarly, Meta’s Llama and Google’s transformer-based models, such as BERT, RoBERTa, and PaLM, have been key in advancing AI-driven language applications. (IBM 2023b.)

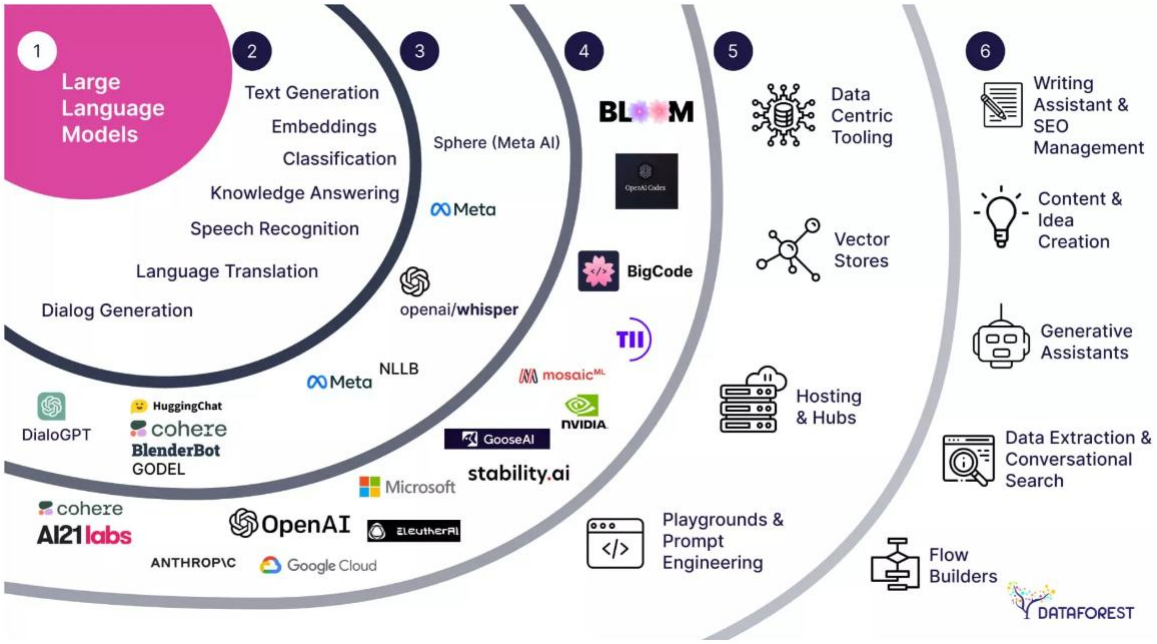

Figure 16. Large Language Model Landscape (Sheremeta 2024)

One intriguing feature of large language models (LLMs) is their capacity to develop unexpected skills. When they reach a certain scale, these models start to display emergent behaviors, allowing them to address new problems without explicit programming. For example, they can perform reasoning tasks, solve arithmetic problems, and adapt to new challenges with few examples. This phenomenon was initially surprising because these capabilities were not intentionally built into the models. However, they highlight the remarkable adaptability of LLMs in understanding and generating human-like text. (Linares et al. 2023.)

3.8 State of art

Artificial Intelligence is advancing rapidly, reshaping industries, automating tasks, and enhancing human capabilities. As it becomes more powerful and widespread, it presents both exciting opportunities and significant challenges, making it essential to understand its current state and future impact.

ChatGPT-4.5 is OpenAI's most advanced generative AI model at launch, surpassing previous models like GPT-4o and GPT-4. Like its predecessors, GPT-4.5 is a large language model (LLM) built on transformer architecture (Michael 2025).

While earlier versions, such as GPT-4o, focused on multimodal capabilities, GPT-4.5 prioritizes deeper comprehension and more natural interactions. It significantly improves re-

sponse accuracy, minimizing hallucinations, which are incorrect or misleading outputs, compared to previous GPT models. These advancements enhance AI's ability to generate reliable, context-aware responses, resulting in smoother and more precise interactions. (Michael 2025.)

Figure 17 illustrates the evaluation of GPT-4.5 compared to its previous version.

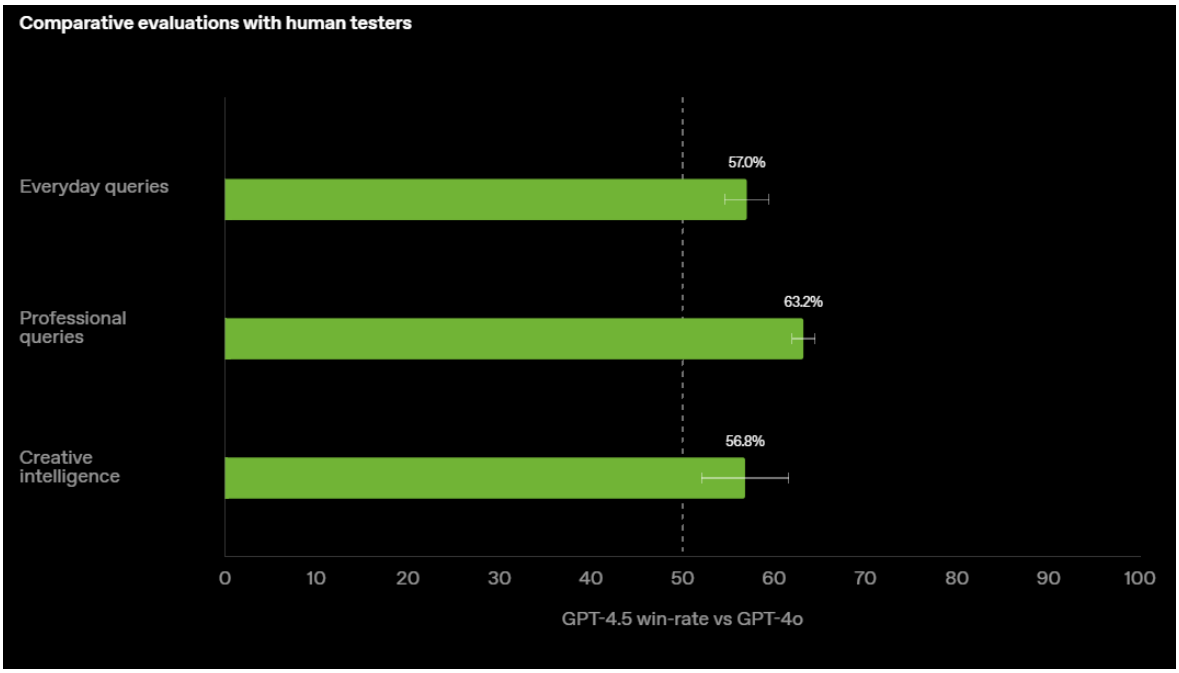

Figure 17. GPT-4.5 win rate vs GPT-4o (OpenAI 2025)

Another significant milestone in AI is video generation. Sora is OpenAI's video generation model. It allows users to create videos from text, images, or existing footage. It is built on insights from DALL·E and GPT models and enhances creative expression by refining video generation capabilities. Using a diffusion process, Sora starts with static noise and gradually refines it, ensuring subject consistency even when temporarily out of view. It also adopts DALL·E 3's recaptioning technique for better text-to-video accuracy and leverages transformer architecture for scalable performance. (OpenAI 2024a.)

In addition to generating videos from text, Sora can animate still images, extend existing footage, and fill in missing frames. This makes it a powerful tool for video creation and simulation, pushing AI closer to understanding the real world. (OpenAI 2024a.)

4 Development Tools

4.1 Game engines

A game engine is a software development environment that allows the creation of video games across various platforms, including PC, mobile and consoles. It provides essential tools and systems that manage different aspects of games, such as rendering graphics, handling physics, managing audio, coordinating animations, implementing artificial intelligence and facilitating networking. (Huston 2019.)

The game engine acts as the game's core, playing a crucial role in its functionality and interaction with players. A well-suited engine can significantly enhance the game experience and performance, while an incompatible or inefficient engine can contribute to a game's failure. (Huston 2019.)

There are two main types of game engines, each with its own benefits and limitations: third-party game engines and proprietary game engines (University of Silicon Valley 2020).

Third-party game engines are developed by companies that license them to other studios, allowing developers to create games without having to build an engine from scratch. These engines support a wide variety of game genres and styles, making them accessible to many developers. However, studios using a third-party engine must pay licensing fees. Two of the most popular engines of this type are Unreal Engine, known for its high-fidelity graphics and use in both AAA and indie games, and Unity, which offers an easier learning curve and is widely used for mobile and indie game development. (University of Silicon Valley 2020.)

A comparison of these engines features is shown in Figure 18.

	Unity	Unreal
Graphics	Physically-Based Rendering, Global Illumination, Volumetric lights after a plugin installed, Post Processing	Physically-Based Rendering, Global Illumination, Volumetric lights out of the box, Post Processing, Material Editor
Unique Features	Rich 2D support	AI, Network Support
Target Audience	Mostly indies, coders	AAA-game studios, indies, artists
Coding	C#, Prefab, Bolt	C++, Blueprints
Community	More than 200k members on the official subreddit	About 100k members on the official subreddit
Performance	Does not scale well, unlike Unreal Engine	Has support for distributed execution (Incredibuild)

Figure 18. Comparison Table – Unity vs Unreal (iRender 2022)

Proprietary game engines, on the other hand, are developed internally by specific studios or publishers. Building a custom engine requires substantial resources, but it enables the creation of specialized features tailored to the unique needs of a game. An example of this type of engine is EA Ignite, which is specifically designed for EA Sports titles and focuses on delivering realistic animations and detailed visuals in sports simulations. (University of Silicon Valley 2020.)

4.2 Back End

Backend development focuses on the server-side operations that power applications, ensuring efficient communication between the user interface and the database. It is responsible for processing data, enforcing security measures, and maintaining overall system performance. By creating server-side logic, handling data management, and processing user requests, backend development enables seamless interaction between different components of an application, ensuring dynamic and responsive digital experiences. (Thanh 2023.)

4.2.1 Server

A server is a computer or software system designed to provide services, data, or resources to other devices, known as clients, within a network. It operates using a client-server architecture, allowing it to process and respond to client requests. This enables functions such as web hosting, file storage, email communication, and application management. Servers facilitate data exchange and accessibility across networks, making them essential for online services and digital infrastructure. (Monteclaro 2023.)

There are various server types, each performing distinct functions within network infrastructures. Web servers deliver content to users, while proxy servers act as intermediaries that connect host servers with client servers. Virtual machines enable the creation of virtualized environments, allowing multiple users to operate on a single physical hardware system. Application servers provide essential business logic for various applications, while file servers manage and store files for network users. Additionally, mail servers facilitate the sending, receiving, and storage of email communications. These are just a few examples of the many server types, each fulfilling a specific role and contributing to the efficient operation of network services. (Curtis 2024.)

4.2.2 API

An Application Programming Interface (API) is a structured set of rules that allows different software applications to interact and share data, features, or functionalities. They serve as a bridge between systems, which enables developers to integrate external services and capabilities without having to build them from scratch. APIs also provide a secure and efficient way for organizations to manage and share application data internally or with external partners, streamlining processes and fostering interoperability between different platforms. (Goodwin 2024.)

There are different types of APIs categorized by their architectural styles, each designed to meet specific needs and technical constraints. Among the most widely used are REST, SOAP, and GraphQL. (Acar 2024.)

REST is particularly popular for building modern web applications due to its simplicity and scalability. It operates on a stateless, client-server model, illustrated in Figure 19, where communication is carried out through standard HTTP methods such as GET, POST, PUT, and DELETE. REST APIs typically use JSON data exchange, which is both lightweight and easy for humans to read and machines to parse, making development faster and integration smoother. (Acar 2024.)

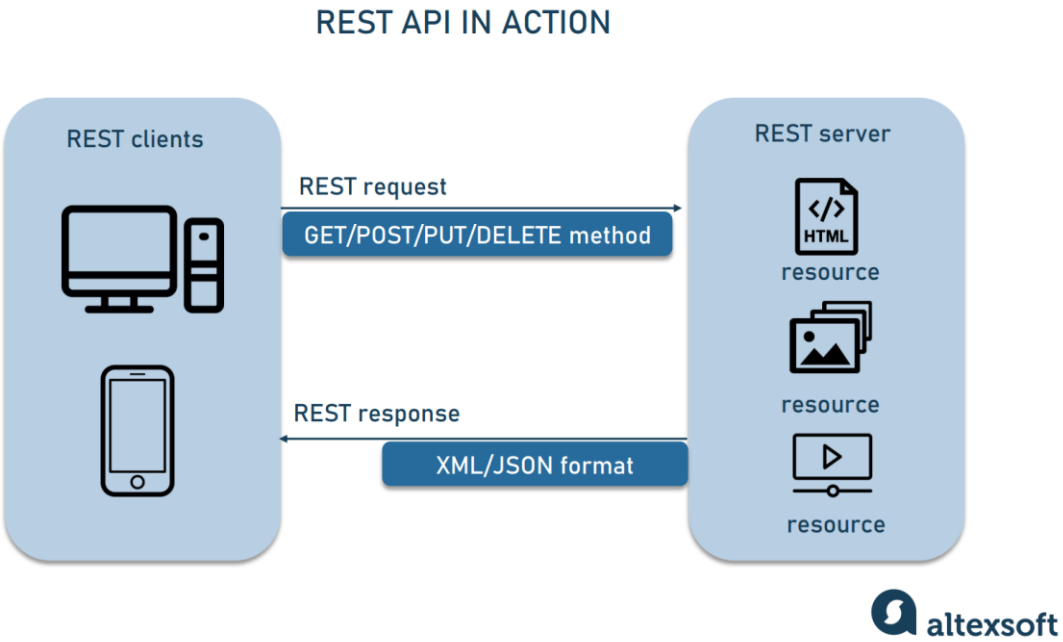

Figure 19. REST API in action (Altexsoft 2022)

REST APIs are essential in modern web development due to their ability to facilitate seamless communication between diverse software components. By standardizing interactions, they enable developers to integrate functionalities efficiently, promoting innovation and enhancing user experiences. Their scalability and flexibility support the growing demands of web platforms, data sources, and user interactions, making them a cornerstone in building robust, interconnected digital ecosystems. (Acodez 2023.)

5 Practical case

5.1 Introduction

Children with special needs often face unique challenges when learning to read and write. Traditional teaching methods may not always be effective or engaging for them, which is why technology can play a crucial role in making learning more accessible and enjoyable. This project focuses on developing an interactive mini-games application designed specifically to help these children improve their literacy skills in a fun and supportive way.

The project features a variety of mini-games that encourage children to practice essential reading and writing skills through interactive activities. To enhance the experience, artificial intelligence (AI) has been integrated into the system. One of the key AI-powered features is speech-to-text technology, allowing children to interact with the games using their voice. This not only makes learning more engaging but also helps them develop their language skills naturally through speaking and listening.

This application also includes a virtual puppet, which serves as both a companion and a guide throughout the learning process. This puppet provides positive feedback and encouragements by analysing their emotional state using facial recognition and sentiment analysis. By adapting its responses based on the child's mood, the virtual puppet creates a supportive environment that keeps the child motivated and eager to continue learning.

The inspiration for this project comes from APSA, a non-governmental organization (NGO) dedicated to improving the quality of life for individuals with different abilities throughout their lifetime.

5.2 Project Structure

The application is designed around a world travel theme, where children embark on an educational journey across different continents. It is divided into two thematic blocks, each represented by a continent: Europe and America.

Each block contains several levels, and within each level, children engage in interactive mini-games specifically designed to target the skills being developed. The progressive structure of the game ensures that children build a solid foundation before moving on to more complex literacy concepts.

5.2.1 Block 1 – Europe: Pre-reading and Pre-writing Skills

Before learning to read and write, children need to develop various foundational skills that will enhance their literacy learning process. This block is divided into different levels that focus on different aspects of the pre-reading and pre-writing skills.

Level 1: Visual Discrimination and Memory. At this level, children focus on recognizing and differentiating letters, symbols, and patterns. Engaging in these exercises enriches their visual memory and attention to detail, both of which are critical for accurate letter recognition and word formation.

Level 2: Auditory Discrimination and Phonological Awareness. This level emphasizes the development of listening skills and phonemic awareness, enabling children to distinguish between various sounds and syllables. These activities lay the groundwork for effective word decoding as they progress to reading.

Level 3: Laterality and Graphomotor Skills. This level concentrates on motor coordination and hand dominance, vital for improving handwriting skills. The exercises include tracing and drawing, alongside activities designed to enhance fine motor control and the formation of letters.

The main level selection screen of this block is illustrated in Figure 20.

Figure 20. Main View Europe Block Levels selection screen

5.2.2 Block 2 – America: Learning Vowels through the Syllabic Method

Following the acquisition of pre-reading skills, children transition to learning vowels. This block is structured around the syllabic method, which introduces letters in a logical and progressive way, facilitating early reading.

Level 1: Learning the Vowels. The initial focus at this level is on the recognition, writing, and pronunciation of vowels. To reinforce this understanding, interactive mini-games integrate graphomotor exercises, visual discrimination activities, and phonological awareness tasks. These tools assist children in identifying vowel sounds, associating them with their written forms, and practicing their writing skills.

The mini-games selection screen of level 1 can be seen in Figure 21.

Figure 21. Level 1 America Block Mini-Games selection screen

5.3 AI Selection

The application is mainly directed to help children with special needs learn the basic concepts of reading and writing, so in order to offer a smooth and engaging experience for them, selecting the right AI models is a crucial step.

These models must combine accuracy and adaptability, especially in tasks like speech recognition and emotional analysis, to ensure that the game responds accurately and empathetically to each child. This not only supports better learning outcomes but also helps build trust and emotional connection with the game, which is particularly important for children with special needs.

5.3.1 Speech-To-Text

One of the most important functions of artificial intelligence in the game will be speech recognition, allowing players to interact with some mini-games using spoken words or letters.

Since the users are children with special needs, many of whom may have diverse speech patterns, accents, or developmental speech delays, the model must demonstrate both high precision and robustness to accommodate varying pronunciations and vocal characteristics. Additionally, to ensure a smooth and engaging experience, it is essential to choose a fast transcription model that can process speech quickly, ideally within a few seconds. This capability will enable children to receive immediate and reliable feedback on their responses during gameplay, ensuring that learners remain actively involved and motivated throughout the process.

For this reason, Whisper model from OpenAI was chosen. Whisper is a powerful, open-source automatic speech recognition system capable of converting spoken language into text. Trained on 680,000 hours of multilingual supervised data, this model can handle diverse speech patterns, accents, and background noise (OpenAI 2022). These features make Whisper particularly suitable for this application since children with special needs, as said before, may exhibit varied speech characteristics. Furthermore, Whisper's robustness to noisy environments enhances its reliability, making it an ideal choice for use in educational settings where background noise is often present.

As explained, Whisper will successfully fulfill the application's needs, and children will be able to interact through voice with the mini-games and get immediate, accurate feedback.

5.3.2 Image-To-Text

To create a more immersive and emotionally supportive experience, the application will provide dynamic feedback tailored to the child’s emotional state. When the child appears frustrated or sad, the system will generate a supportive message to uplift and motivate them. On the contrary, if the child seems to be enjoying the activity, it will respond with congratulatory messages to reinforce positive engagement.

In order to achieve this functionality, the application must first assess the user's emotional state. For that purpose, it will require an intelligent and fast model that can accurately interpret facial expressions from captured images.

For this purpose, two OpenAI models were tested in the application GPT-4o-mini and GPT-4o. Both are advanced autoregressive multimodal models capable of processing and generating text, audio, images, and video. They can analyze visual content and respond to. related questions by receiving the image through a URL or as a base64-encoded input (OpenAI 2024b; OpenAI 2024c). As shown in Figure 22, GPT-4o and GPT-4o-mini are very similar in their core capabilities. The comparison highlights minor differences, such as the intelligence or the speed that were noticed in the testing.

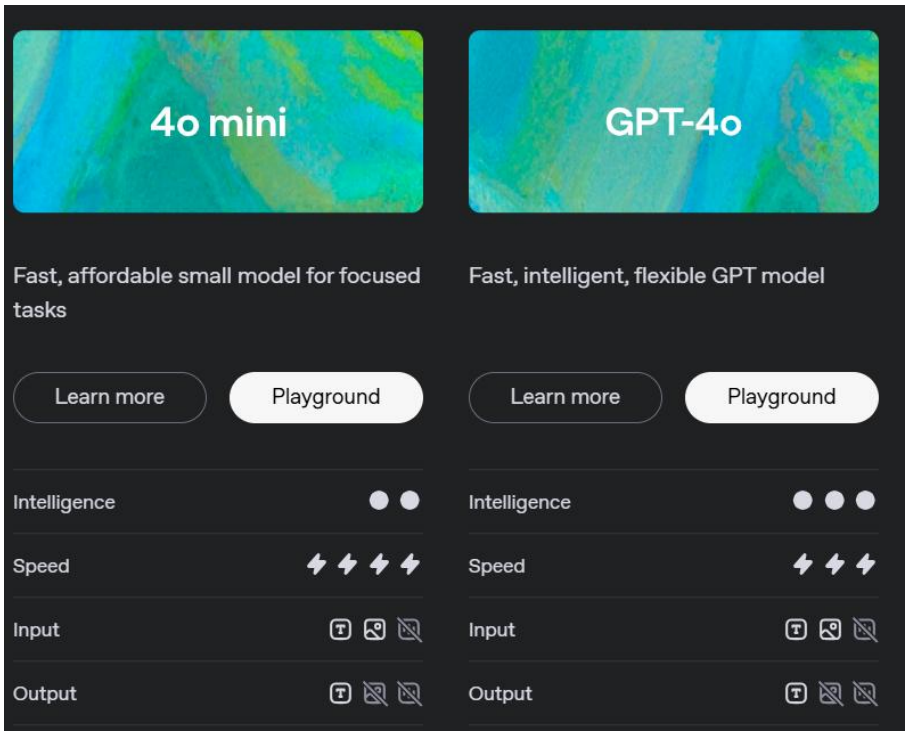

Figure 22. Comparison GPT-4o and 4o mini

During the model testing, the 4o mini model provided more concise feedback and demonstrated a slightly faster performance compared to GPT-4o, making it the preferred choice for application development.

5.3.3 Text-To-Speech

Once GPT-4o-mini generates appropriate feedback based on the child's facial expressions, the next step is to convert that text into audio, allowing the game's puppet to deliver the message in a more personal and engaging way. For this matter, a Text-To-Speech (TTS)

model is required, capable of generating a natural yet slightly cartoonish voice that children can interpret more as a friend than a supervisor or a teacher. The tone and style of the voice play a key role in shaping the user's emotional response, so it is essential that the voice feels warm, playful, and approachable. This not only helps maintain the child's engagement but also fosters a sense of comfort and trust, which is especially important for children with special needs.

There are many Text-to-Speech (TTS) solutions available, but two of the most notable are ElevenLabs and Hume AI. Both models introduce a new AI-driven approach that not only lets creators fine-tune a wide range of technical settings to craft a unique voice style but also enables voice generation to be guided through prompts and detailed descriptions. (Girdhar 2025.)

ElevenLabs is a leading force in the AI voice space and is known for setting the standard in ultra-realistic text-to-speech. Its voices are clear, lifelike and rich in emotional depth, capturing tone, pacing, and intent with remarkable accuracy. The platform supports over 30 languages and offers powerful customization tools like Voice Cloning and Voice Design. (Girdhar 2025.)

On the other hand, Hume AI has gained strength since it shows a deep understanding of both text and emotions. Its system, Octave, does not just read words, but it gets to interpret them within the context of the sentence. This enables it to automatically adjust tone, emotion, and even delivery style based on the content. Additionally, it provides a really easy way to create an entirely new voice from a text description, as illustrated in Figure 23 (Girdhar 2025.)

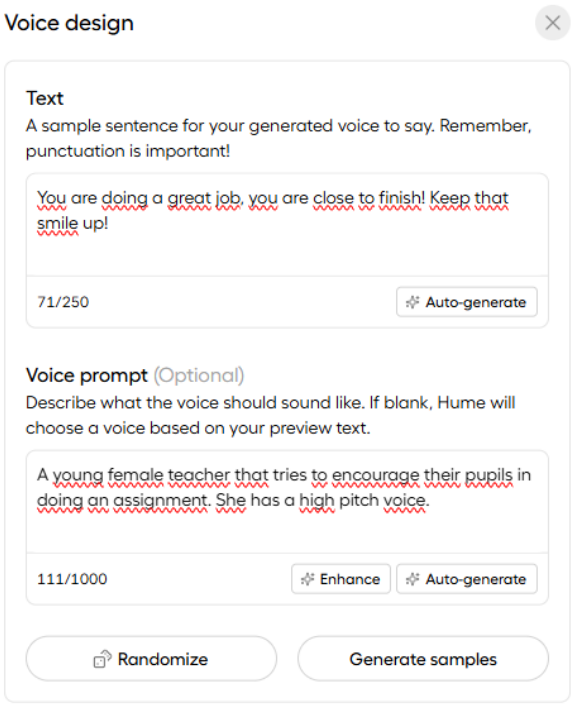

Figure 23. Voice design option in HumeAI

For the final selection of the TTS model, both ElevenLabs and Hume AI were tested by generating custom voices giving the following description text: “A young female teacher that tries to encourage their pupils in doing an assignment. She has a high pitch voice.”. Even though both platforms provided high-quality results, HumeAI was notable for sounding more natural, emotionally expressive, and better aligned with the intended meaning of the text. Consequently, HumeAI was ultimately selected as the final text-to-speech (TTS) solution for the application.

5.4 Backend

Once the necessary models for the application are selected, the next step is to design and implement a robust backend that can efficiently manage data, handle requests, and facilitate communication with the external APIs of the chosen models. The backend plays a crucial role in making sure all the data coming from the user is handled carefully. It processes the information efficiently, stores it securely, and sends it back to the app in a way that keeps everything running smoothly. Consequently, it ensures that users have a fast, reliable, and safe experience, even when the app is dealing with complex tasks behind the

scenes. Since this application involves image and audio data, the backend becomes especially valuable in handling format conversion and data preparation, allowing the front end to remain lightweight and focused on user interaction.

To achieve this goal, a RESTful API is an excellent choice, as the application will require a backend capable of efficiently handling numerous user interactions simultaneously. This architecture allows the backend to easily process requests using standard HTTP methods, ensuring smooth communication between server and client and quick responses. Its stateless system makes the development process more flexible and allows easy updates or integration with external services. This flexibility is crucial for dynamic features, such as the AI-powered tools that will be included, like speech-to-text, text-to-speech and image-to-speech, which need to work seamlessly with other systems. REST ensures that these features are reliable and can function across different devices and platforms, providing users with a seamless experience.

5.4.1 Backend Tools

In order to meet the requirements explained, a programming language and a framework to integrate the RESTful API must be chosen. Regarding the first one, Python was selected due to its clear syntax, ease of use, and extensive support for libraries that facilitate fast development. These libraries simplify multimedia processing, making the overall process of image and audio capturing more efficient and accessible. As for the second, FastAPI was selected because it is one of the fastest Python frameworks available. Its simplicity and intuitive syntax make the development process significantly faster and more efficient, especially when integrating multiple AI services. Additionally, FastAPI automatically generates interactive API documentation, which greatly simplifies testing and integration.

FastAPI can run locally on a computer using the simple command “uvicorn main:app –host 0.0.0.0 –port 8000”. This command launches the FastAPI application defined in main.py, on port 8000 making it reachable from any device that has network access to the host machine. During the development phase, the backend will operate on a local computer, allowing other devices within the local network to interact with it seamlessly.

5.4.2 Backend Development

Once the tools were selected, the next step was to design the API calls that would facilitate data exchange with the OpenAI and HumeAI AI services, as well as communication between the backend and the Unity frontend. Since the application integrates image-to-text,

speech-to-text, and text-to-speech, dedicated endpoints were created to manage interactions with each of these services. Additionally, an image-to-speech endpoint was implemented to streamline communication with the frontend, allowing it to request audio feedback directly from an image. This approach ensures that the frontend does not need to handle the underlying logic for image recognition or audio generation. The created endpoints are shown in Figure 24.

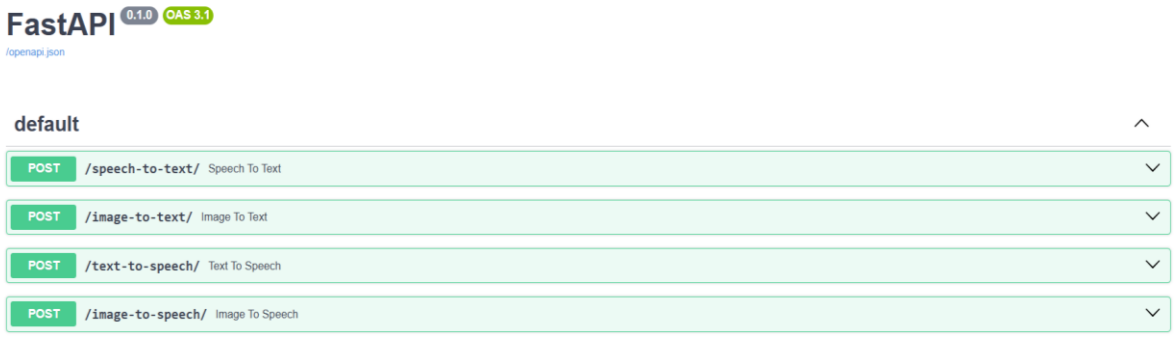

Figure 24. Endpoints for the application in FastAPI documentation

To integrate OpenAI services into the backend using FastAPI, a connection is established with the OpenAI client, allowing POST requests to be sent to the API to process data such as images and audio. In the case of the image-to-text endpoint shown in Figure 25, an image is received and converted into Base64 format, which provides a textual representation of the image instead of transmitting the binary file. This image is sent in the body of the request as part of a structured JSON message.

The structure and parameters of the POST request are always defined in the API documentation. In this case, it is required to specify the model used, which is gpt-4o-mini as said before, the format in which the image is sent, and a message that provides context about the image's content, which in this case is the following: "This child or person is playing an educational game. Based on their facial expression in the image, give them one short motivational sentence (no more than 10 words). If they look sad or upset, try to cheer them up. If they look happy or excited, acknowledge it and encourage them to keep going. If you're not sure about their emotion, give a friendly, general motivational sentence without mentioning any uncertainty.". Additional configuration parameters can also be specified but were not necessary in this case.

```python
@app.post("/image-to-text/")
async def image_to_text(file: UploadFile = File(...)):
    try:
        client = OpenAI(api_key=openai_api_key)
        file_content = await file.read()
        base64_image = base64.b64encode(file_content).decode("utf-8") #Convierte la imagen a formato Base64, que es un formato

        completion = client.chat.completions.create(
            model="gpt-4o-mini",
            messages=[
                {
                    "role": "user",
                    "content": [
                        { "type": "text",
                        "text": "This child or person is playing an educational game. Based on their facial expression in the
                        {
                            "type": "image_url",
                            "image_url": {
                                "url": f"{base64_image}",
                            },
                        },
                    ],
                }
            ],
        )
        print(completion.choices[0].message.content)
        return completion.choices[0].message.content
    except Exception as e:
        return {"error": f"No se pudo procesar la imagen: {str(e)}"}
```

Figure 25. Image-to-text API call in FastAPI

Once the request is sent, the OpenAI API processes the data and returns a response with the generated result, which, in this case, is the text with the feedback message based on the facial expression of the user in the image. This text is extracted from the API's response and sent back to the frontend for use in the application.

The speech-to-text endpoint works similarly, where an audio file is sent to the OpenAI client and a text transcription is returned. In this case, the POST request includes key parameters such as the model, Whisper, a custom prompt to guide the transcription process, and the language setting, Spanish. In the application, speech recognition is used in the second block to identify simple words and vowel pronunciations. To support this functionality, the prompt provided to Whisper was: "This audio contains an object, animal, or vowel in Spanish", which, after testing various options, proved to return the best and most accurate transcriptions.

The text-to-speech endpoint, as illustrated in Figure 26, establishes the connection between the backend and HumeAI, and works in a similar way to the endpoints created for OpenAI. This endpoint receives a text as an input, which will be the feedback returned from the image-to-text call and will generate a synthesized audio file in response.

In the POST request body, both the text and the voice used for interpretation are required as parameters. Many voices are available in Hume’s library nevertheless, the voice named

"Puppet" which is the one that was created previously in HumeAI's playground to test the model is used.

```python
@app.post("/text-to-speech/")
async def text_to_speech(text: str):
    try:
        hume = AsyncHumeClient(api_key=hume_api_key)
        speech1 = await hume.tts.synthesize_json(
            utterances=[
                PostedUtterance(
                    voice=PostedUtteranceVoiceWithName(name="Puppet"),
                    text=text,
                )
            ]
        )
        audio_path = await write_result_to_file(speech1.generations[0].audio, "speech1_0")
        #print("Audio file path: ", audio_path)
        print("Text: ", text)
        return {"audio_file_path": audio_path}
    except Exception as e:
        return {"error": f"No se pudo obtener el audio: {str(e)}"}
```

Figure 26. Text-to-speech API call in FastAPI

Once the request is processed by HumeAI, the API responds with an audio generation object, which includes a reference to the generated audio. This audio data is then saved locally on the server, and the path to the audio file is returned in the response to the frontend. This allows the frontend to access and play the synthesized speech as feedback to the user.

Now, with this system, to generate a complete feedback response, the frontend must first call the image-to-text endpoint to retrieve the message and then send that message to the text-to-speech endpoint to receive the corresponding audio. However, this process involves two separate interactions with the backend, which may introduce delays and complexity to the frontend.

To improve efficiency and reduce response time, a dedicated image-to-speech endpoint, shown in Figure 27, was developed. This endpoint handles both internal API calls, image-to-text and text-to-speech, on the backend, allowing the frontend to simply send an image and receive the generated audio in return without needing to manage multiple requests.

```python
@app.post("/image-to-speech/")
async def image_to_speech(file: UploadFile = File(...), request: Request = None):
    try:
        # Step 1: Get motivational feedback text from image
        feedback_text = await image_to_text(file)

        # Step 2: Use the text-to-speech endpoint with the result
        audio_response = await text_to_speech(text=feedback_text)

        print("motivational_text: ", feedback_text)
        print("audio_file_path: ", audio_response["audio_file_path"])

        # Step 3: Move the audio file to a public folder called static/
        # (inside the project directory), so that it is accessible via a URL
        original_path = audio_response["audio_file_path"]

        # Step 4: Convert to PCM 16-bit mono since the original file is 32-bit and is not compatible with Unity
        audio = AudioSegment.from_file(original_path)
        audio = audio.set_frame_rate(22050).set_channels(1).set_sample_width(2)  # 2 bytes = 16 bits

        # Step 5: Save the converted audio to the static directory
        filename = f"{uuid.uuid4()}.wav"
        static_dir = Path("static")
        static_dir.mkdir(parents=True, exist_ok=True)
        new_path = static_dir / filename
        audio.export(new_path, format="wav")

        # Step 6: Return the URL of the audio file
        base_url = str(request.base_url).rstrip("/")
        audio_url = f"{base_url}/static/{filename}"

        return JSONResponse(content={"audio_url": audio_url})

    except Exception as e:
        return {"error": f"Error in image-to-speech pipeline: {str(e)}"}
```

Figure 27. Image-to-speech API call in FastAPI

When an image is uploaded via a POST request, the backend first processes it using the image-to-text function to generate a motivational sentence based on the facial expression of the user. The resulting text is then passed directly to the text-to-speech function, which uses HumeAI to synthesize the audio.

Once the audio is generated, it is further processed to ensure compatibility with Unity. Specifically, the audio is converted to PCM 16-bit mono format using the pydub library, which internally relies on ffmpeg for audio conversion, as Unity does not support the default 32-bit format returned by HumeAI. To make the generated audio accessible from the frontend, the file is saved in a public directory called static/, which is mounted in FastAPI. This configuration allows the server to serve any file stored in that directory through a direct URL. After the audio file is saved and assigned a unique name, its full path is returned as a response. The frontend can then access this path to play the audio within the application.

With all the endpoints implemented and the frontend development still underway, it was possible to test them directly through the FastAPI documentation by accessing http://127.0.0.1:8000/docs, as shown in Figure 28.

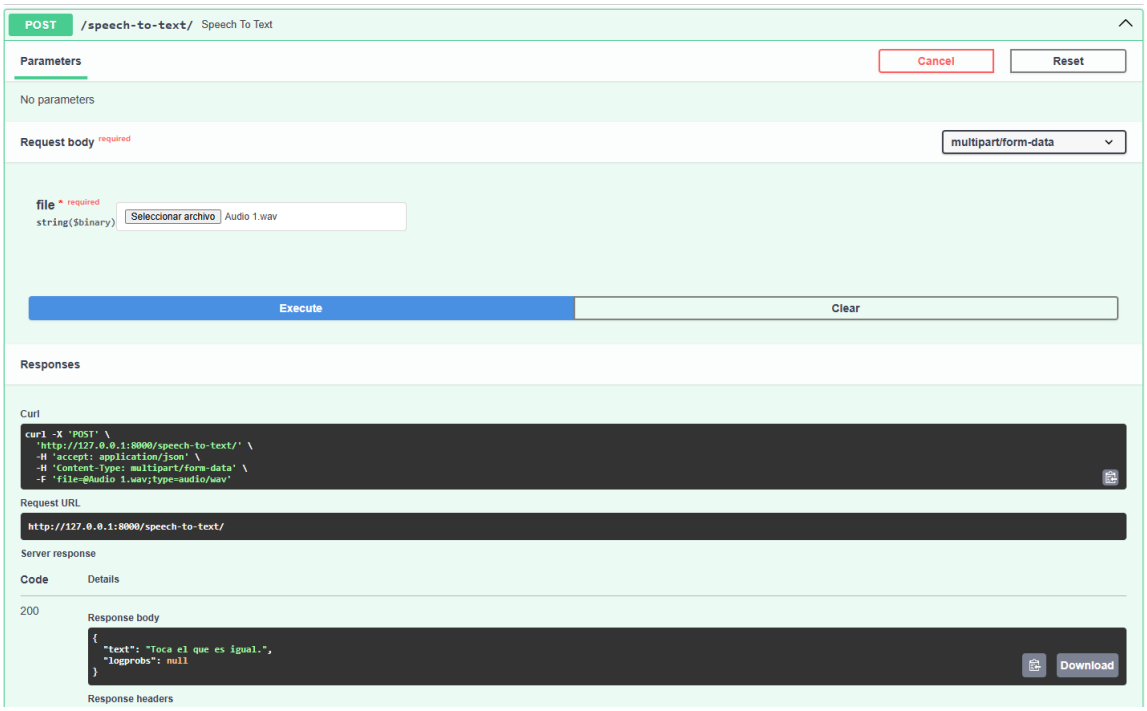

Figure 28. Testing API's endpoints from FastAPI's documentation

5.5 Frontend

While the backend handles the connection with AI services and processes the data, the frontend is the most crucial part of the application from the user's perspective. It is through this interface that children interact with the system, engage in the educational mini-games, and receive real-time feedback. The design and functionality of the frontend are essential, as it must be intuitive, visually appealing, and accessible for children with special needs. This section presents the tools and technologies used to develop the frontend and explains how the game mechanics were implemented.

5.5.1 Frontend Tools

For the development of the frontend, Unity and C# were chosen as the primary tools due to their versatility, performance, and suitability for interactive and educational applications.

Unity is well-suited for 2D development and UI-centric applications, offering a robust canvas system, flexible UI components, and animation tools that make it ideal for building interactive interfaces. It allows for the creation of smooth transitions, responsive layouts, and engaging visual elements, which are essential for keeping children engaged while learning. Additionally, Unity's cross-platform capabilities ensure that the application can run on various devices, such as tablets and computers, providing accessibility for different users.

C# is the primary programming language used in Unity development. It is known for its clean syntax, robust support for object-oriented programming, and ability to manage UI interactions efficiently. With C#, efficient event-driven systems can be created, user interactions can be handled seamlessly, and UI elements can be ensured to respond accurately to inputs. Its strong integration with Unity's UI system allows menus, buttons, and other interface components to be implemented easily while maintaining high performance and ease of maintenance, making it an ideal choice for this project.

5.5.2 Frontend Development

As mentioned earlier, the application is divided into two thematic blocks, each focusing on a different stage of the learning-to-read-and-write process. The mechanics of the mini-games in each block are tailored to the specific skills children are expected to develop at that stage.

5.5.2.1 Block 1

Block 1 focuses on pre-reading and pre-writing skills, which are essential for preparing children to begin the formal process of learning to read and write. Each level within this block is designed to target a specific foundational ability, such as visual discrimination, shape recognition, spatial orientation, auditory processing, laterality, and graphomotor skills.

The mini-games in Level 1, illustrated in Figure 29, are centered around visual discrimination and memory. These games are designed with simple yet engaging mechanics that encourage children to recognize and differentiate between various objects. The interaction is intuitive, involving actions such as dragging and dropping elements or simply tapping on the correct image.

There are four different games in this level, with difficulty increasing progressively to adapt to the user's development. Each game introduces new visual challenges while maintaining familiar interaction patterns to ensure a smooth learning curve.

In Unity, the interactive elements are implemented using image objects, and user input is managed through the IPointerDownHandler interface, allowing for efficient and responsive touch-based interaction. To make the experience visually stimulating, the game elements use visual cues such as blinking when they require attention, or changing color or size when the child makes a correct selection. These visual responses are carefully designed to maintain the child's focus and provide positive reinforcement, which is essential in keeping young users engaged throughout the learning process.

Figure 29. Minigames Block 1 Level 1

The mini-games in Level 2, illustrated in Figure 30, are focused on auditory discrimination and phonological awareness. These games are designed to develop essential listening skills, helping children recognize, differentiate, and remember sounds and syllables, which are key abilities for building strong reading foundations.

This level includes six mini-games, with difficulty increasing progressively to challenge and support the child's growth. Each game presents a different auditory task, such as identifying matching sounds, recalling sound sequences, distinguishing syllables within words, or recognizing how many syllables a word contains. These activities are crafted to reinforce the user's capacity to decode words effectively.

From a technical perspective, the games continue to rely on image objects for visual interaction, using Unity's IPointerDownHandler interface. However, Level 2 introduces a strong auditory component, requiring the integration of audio sources to play animal sounds and words. Audio feedback plays a central role in the game logic, and the interaction is designed so children respond to what they hear rather than what they see, shifting the focus from visual to auditory learning while keeping the mechanics intuitive and accessible.

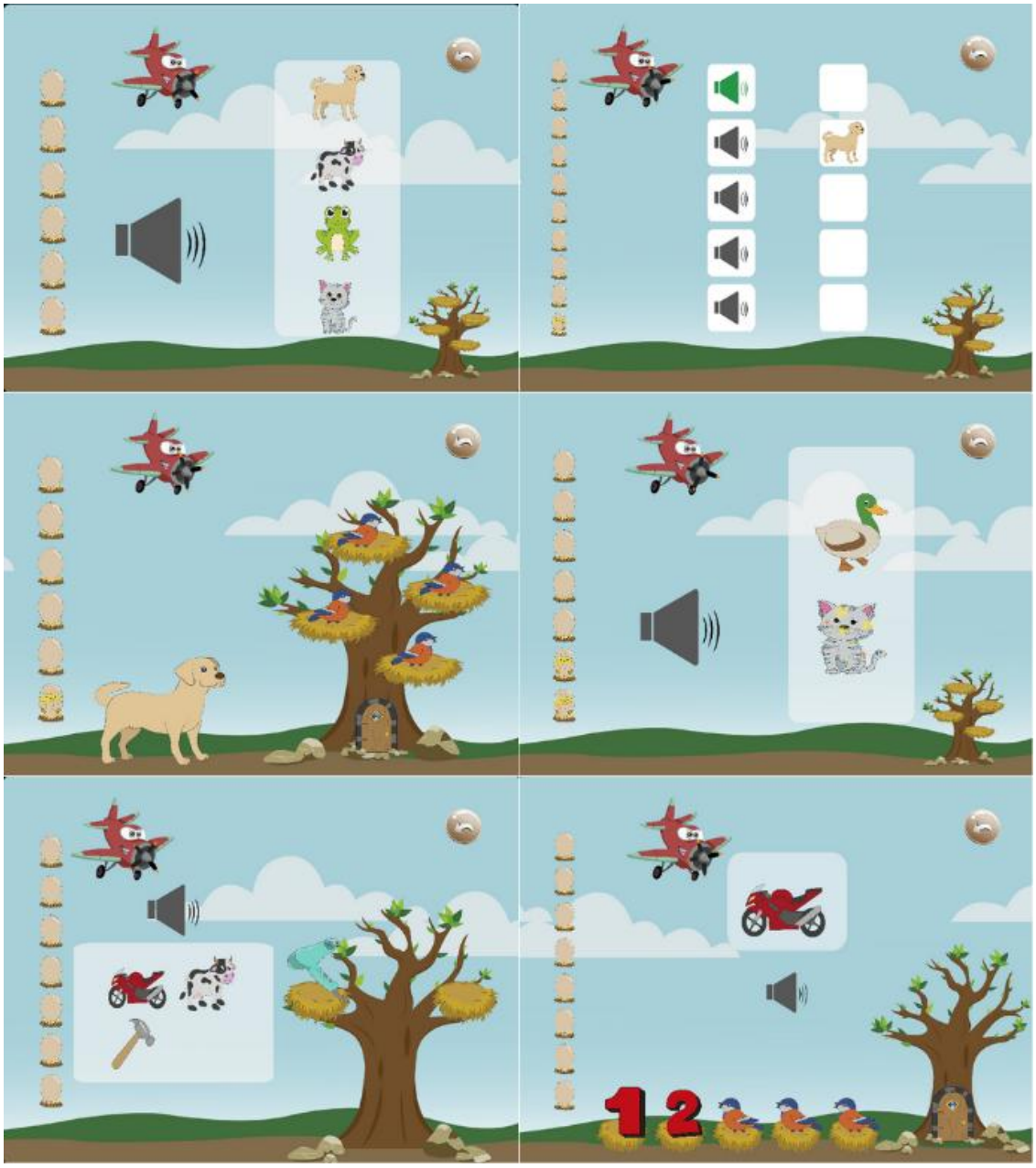

Figure 30. Mini-games Block 1 Level 2

The mini-games in Level 3, illustrated in Figure 31, are centered around laterality and graph-omotor skills, which are fundamental for the development of handwriting and fine motor coordination. These games are designed to help children improve their hand-eye coordination, recognize spatial orientation, and strengthen their ability to perform precise, controlled movements.

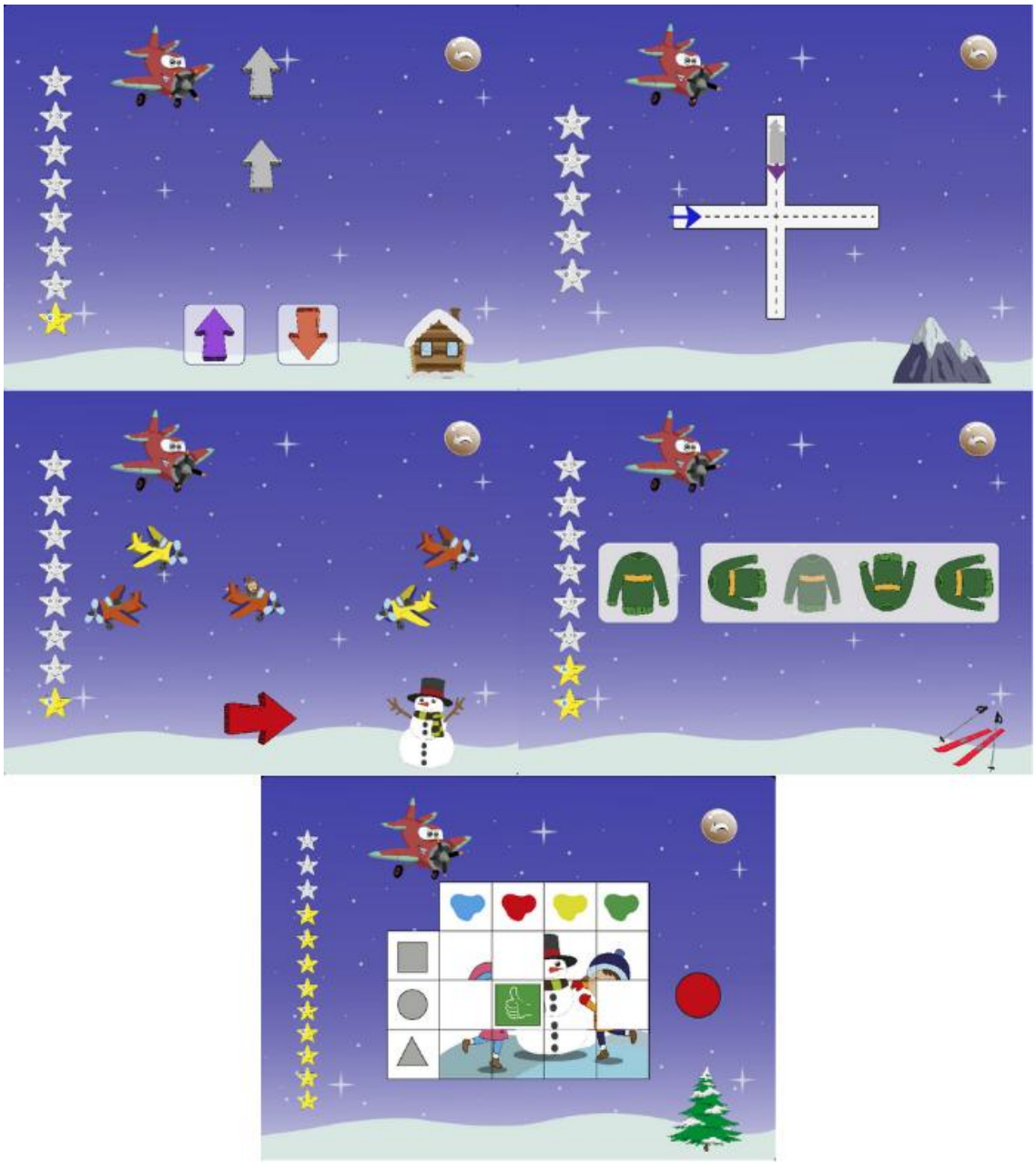

Figure 31. Mini-games Block 1 Level 3

This level includes five mini-games in which challenges involve tasks such as identifying the orientation of elements, following directional cues, associating objects with colors, and especially tracing shapes and lines to reinforce motor control.

From a technical standpoint, the most relevant thing is that Unity's IBeginDragHandler, IDragHandler, and IEndDragHandler interfaces are used to implement the tracing and drawing mechanics. The most prominent system developed for this purpose is the tracing system, where users drag an object along a predefined path to underline a shape, leaving a visual trail using Unity's TrailRenderer component. This creates an effect similar to drawing on paper.

The system tracks the trail's interaction, shown in Figure 32, with designated points, and only when all required targets are correctly reached, the game confirms that the shape was traced correctly. Visual cues such as trail color, object feedback, and error correction, like resetting the position and clearing the trail when the task is incomplete, help guide the child through the activity while reinforcing correct motion patterns.

```
public class TrazoUI : MonoBehaviour, IBeginDragHandler, IDragHandler, IEndDragHandler
    public void OnDrag(PointerEventData eventData)
    {
        Vector2 localPoint;
        foreach (Transform formaObjetivo in formasObjetivo)
        {
            RectTransform formaObjetivoRectTransform = formaObjetivo as RectTransform;
            if (RectTransformUtility.ScreenPointToLocalPointInRectangle(formaObjetivoRectTransform, eventData.position,
                eventData.pressEventCamera, out localPoint))
            {
                if (formaObjetivoRectTransform.rect.Contains(localPoint))
                {
                    Vector3 worldPoint = formaObjetivoRectTransform.TransformPoint(localPoint);
                    worldPoint.z = 0;
                    transform.position = worldPoint;

                    foreach (GameObject baseObjetivo in basesObjetivo) //Check if the point is inside the baseObjetivo
                    {
                        foreach (Transform punto in baseObjetivo.transform)
                        {
                            if (!puntosTocados.Contains(punto.gameObject) && Vector2.Distance(worldPoint, punto.position) < 0.3f)
                            {
                                puntosTocados.Add(punto.gameObject);
                                //sDebug.Log("punto tocado");
                            }
                        }
                    }
                }
            }
        }
    }
```

Figure 32. Trail’s interaction function

In addition to the core mini-games, this block features a special game for each level, designed to reward and motivate children once they complete all the standard activities. These games serve as interactive rewards that enhance engagement and provide a sense of achievement, reinforcing the learning process in a fun and entertaining way.

Each special game, as illustrated in Figure 33, is thematically aligned with the skills of its corresponding level but introduces a fresh and enjoyable twist to keep the experience exciting. The first game challenges children to find a hidden object, which encourages attention to detail and visual exploration. The second is inspired by the classic electronic game Simon, where players must memorize and repeat sequences, helping to strengthen auditory memory and concentration. Finally, the third game involves guiding a ball along a path by tilting the device, utilizing its motion sensors to promote motor coordination and spatial awareness in an engaging, physical way.

These activities are designed to be entertaining while reinforcing educational objectives in a relaxed, pressure-free environment, making learning feel like play and helping maintain the child's motivation throughout the experience.

Figure 33. Extra games Block 1

Most of the mechanics used in these special games are based on those developed for previous mini-games. However, the third special game introduces a unique and engaging interaction. In this game, the child guides a ball into a goal by physically tilting the device.

This interaction utilizes the device’s built-in accelerometer which is implemented by capturing input through the Input.acceleration function in Unity. The game continuously reads the accelerometer data and converts the device’s movements into directional forces that are applied to the ball’s Rigidbody2D component. As the child moves the device, the ball responds accordingly, moving across the screen toward the target.

5.5.2.2 Block 2

Block 2 focuses on learning vowels through the syllabic method, following the development of essential pre-reading skills. This method introduces each vowel progressively and logically, helping children build a strong phonetic foundation. The activities in this block are designed to reinforce vowel recognition and pronunciation, supporting the initial steps of reading. Unlike Block 1, this block contains only one level dedicated explicitly to practicing vowel sounds.

To encourage children to practice vowel pronunciation and simple words, this block incorporates speech-to-text AI. This system allows the application to analyze and assess the child's pronunciation, offering immediate feedback and turning speech practice into an active part of the learning process.

Level 1 of Block 2, as shown in Figure 34, is dedicated to helping children master the recognition, writing, and pronunciation of the main vowels. The mini-games in this level focus on reinforcing the basic phonetic components needed for reading and writing. The activities progressively introduce the vowels and their associated sounds.

Figure 34. Mini-games Level 1 Block 2

One of the key games consists of underlining vowels, where the user has to trace over the different vowels using their finger or stylus. Another important type of exercise involves recognizing vowels among other letters, such as consonants, either in isolated letters or within full words. Finally, there is an activity where children identify words that start with specific vowels.

The AI features in this section are smoothly integrated into the mini-games, encouraging children to pronounce vowels in the first activity and full words in the third, as can be seen in Figure 35. By utilizing speech recognition technology, the system actively listens to the child's pronunciation and provides real-time feedback. This approach allows children to practice and improve their pronunciation skills in a dynamic and interactive environment.

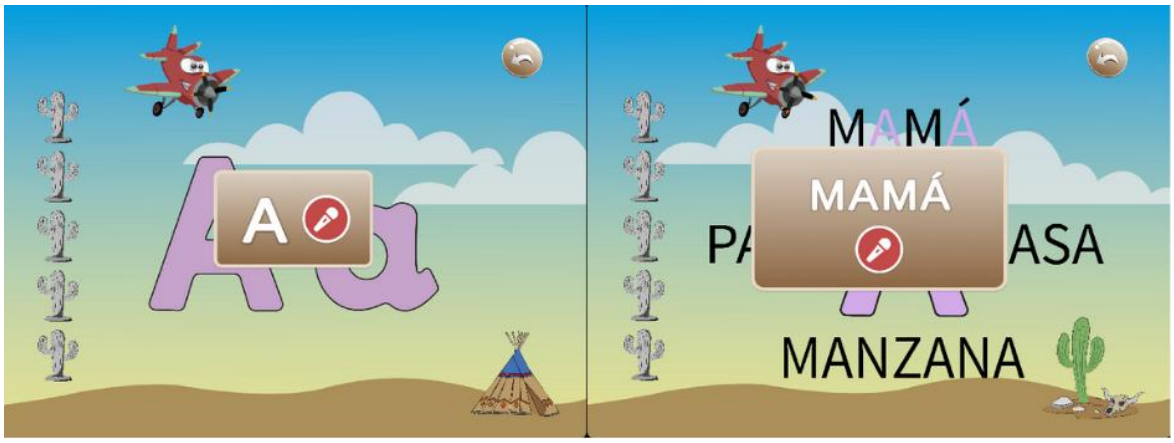

Figure 35. AI enhancement in Level 1 Block 2 activities

The primary function of speech recognition is to encourage children to repeat either a vowel or a word after completing the main activities, such as underlining or selecting vowels within words. This allows them to practice and improve their pronunciation actively.

For this matter, the backend includes a preconfigured API call to Whisper AI, that was programmed before, which handles the speech recognition process by uploading an audio file.

Before making the connection with the backend, the audio has to be recorded and stored temporarily. Audio recording in the application is handled through Unity's built-in microphone capabilities. When the child starts a voice task, the microphone captures audio for a fixed duration, using a sample rate of 16 kHz for optimal compatibility with the speech recognition system. During this time, visual feedback is provided to indicate that the recording is active by doing an ease-in-ease-out effect on the recording button. Once the input is captured, the clip is trimmed to the actual speaking duration to avoid unnecessary silence and then saved locally in WAV format.

Once the audio is stored, it is sent to the backend using a UnityWebRequest with an HTTP POST method, as can be seen in Figure 36. This Unity-specific networking tool allows efficient file uploads from within the app, ensuring compatibility across platforms like Android. The backend, already configured with the Whisper AI speech-to-text API, receives the audio and returns the recognized transcription.

```
private IEnumerator SendAudioToAPI(string filePath)
{
    Debug.Log("Enviando audio a la API...");

    byte[] audioData = File.ReadAllBytes(filePath);
    WWWForm form = new WWWForm();
    form.AddBinaryData("file", audioData, filePath, "audio/wav");

    using (UnityWebRequest request = UnityWebRequest.Post(apiUrl, form))
    {
        Debug.Log("Esperando respuesta de la petición a la API...");
        yield return request.SendWebRequest();
        Debug.Log("Respuesta recibida.");

        if (request.result == UnityWebRequest.Result.Success)
        {
            Debug.Log("Respuesta de la API: " + request.downloadHandler.text);
            // Comprobar respuesta
            CheckAnswer(request.downloadHandler.text);
            // Devolvemos el alpha del microUI a 1
            microButton.SetNotRecordingSprite();
        }
        else
        {
            Debug.LogError("Error al enviar el audio: " + request.error);
            microButton.SetNotRecordingSprite();
        }
        System.GC.Collect();

    }
}
```

Figure 36. Speech-to-text call to the backend from Unity

Even though the request is sent immediately, the response is not received right away. The audio must first be processed by the backend and then sent to the external Whisper AI API for transcription, which introduces a brief delay. During this waiting period, the user interface visually indicates that the system is processing by changing the recording button to grey. This signals to the child that a response is in progress, ensuring that the interaction remains intuitive.

In addition to the main activities, this block features a special game designed to further reinforce vowel recognition through speech: the classic “I Spy” game, as can be seen in Figure 37. In this activity, the child is encouraged to identify objects whose names begin with a specific vowel and then pronounce those words aloud. This adds a playful, exploratory element to the learning experience. Like in the activities mentioned before, the game uses the same speech recognition system using the backend API call and Unity’s built-in audio recording and handling methods to process the child’s spoken input.

Figure 37. I spy game in Block 2

5.5.2.3 Pupped Feedback

Throughout the entire experience, a friendly puppet companion will accompany the child during every game. Acting as both a guide and supportive friend, the puppet will observe how the child interacts with the activities and offer immediate feedback based on their responses. Going a step further, this character will not just provide encouragement or corrections through sound cues but also interpret the child’s emotional state by analyzing their facial expressions. With this information, the puppet will give personalized, empathetic feedback tailored to how the child is feeling, making the learning process more supportive, adaptive, and emotionally engaging.

To achieve this, the front end is set up to capture a photo of the user that will be provided to the backend, which includes a preconfigured API call to handle the feedback generation that was programmed before, which handles both the image-to-text and the text-to-speech.

WebCamDevice is used to access the device's front camera to capture the image from Unity. The camera activates periodically and captures one frame that will be encoded into JPG format to prepare it for transmission.

The encoded image is then sent to a backend server using UnityWebRequest through an HTTP POST request for image-to-speech processing, as can be seen in Figure 38. As previously described, the backend returns a JSON response containing the URL of a generated audio file. Unity extracts this URL, downloads the audio clip, and plays it back through the in-game character. This process is executed automatically at regular intervals, allowing the virtual assistant to deliver real-time, emotionally aware feedback throughout the gameplay experience.

```
public IEnumerator SendImageToAPI(byte[] imageBytes)
{
    Debug.Log("Enviando imagen a la API");
    WWWForm form = new WWWForm();
    form.AddBinaryData("file", imageBytes, "photo.jpg", "image/jpeg");

    using (UnityWebRequest request = UnityWebRequest.Post(apiUrl, form))
    {
        Debug.Log("Esperando respuesta de la petición a la API...");
        yield return request.SendWebRequest();
        Debug.Log("Respuesta recibida.");

        if (request.result == UnityWebRequest.Result.Success)
        {
            string jsonResponse = request.downloadHandler.text;
            Debug.Log(jsonResponse);
            string audioUrl = ExtractAudioUrlFromJson(jsonResponse);

            // obtener el audio desde la URL
            StartCoroutine(DownloadAndPlayAudio(audioUrl));
        }
        else
        {
            Debug.LogError("Error en la solicitud: " + request.error);
        }
    }
}
```

Figure 38. Image-to-speech call to the backend from Unity

This process repeats throughout each game, giving the child ongoing, dynamic interactions with the puppet. To manage this, a camera call controller is integrated into each game's

logic, triggering photo captures after a configurable number of user interactions. However, since the API responses are not instantaneous, there is a risk of overlapping requests if a new image is sent before the previous one has been processed. To prevent this, a control mechanism ensures that additional API calls are only made once a response from the previous request has been received, avoiding interference between interactions and maintaining the flow of real-time, emotionally responsive feedback, as illustrated in Figure 39.

```
public void CameraCall()
{
    if (webCameraController == null || webCameraController.IsProcessing()) return;

    cameraCalls_counter++;
    if (cameraCalls_counter >= cameraCalls_interval)
    {
        StartCoroutine(webCameraController.CaptureAndSendImage());
        cameraCalls_counter = 0;
    }
}
```

Figure 39. Camera call controller from the games

5.5.2.4 General features of the games

The game includes several features designed to make the experience more enjoyable and educational for children. One of the key elements is the score system, which uses playful visuals that change depending on the level and block. For instance, in Block 1 Level 1, scores are represented as suns, while in Block 1 Level 2, they appear as eggs. These elements not only track progress but also add a fun and thematic touch to each stage.

To maintain children's attention and make the gameplay more interactive, many of the clickable or highlighted elements include animations. For example, in Block 2 Level 1 Game 1, the vowels perform different animations after being underlined, making the activity more engaging and visually stimulating. Another important feature is the presence of audio instructions in all games, helping children follow the activities more independently. Additionally, some games offer extra audio guidance during the initial interactions to further support understanding.

The game also features a settings menu that allows switching between teacher and student modes, as Figure 40 shows. In teacher mode, all games are unlocked and can be accessed freely, which is helpful for guided learning. On the other hand, student mode unlocks content gradually as the child progresses through the game.

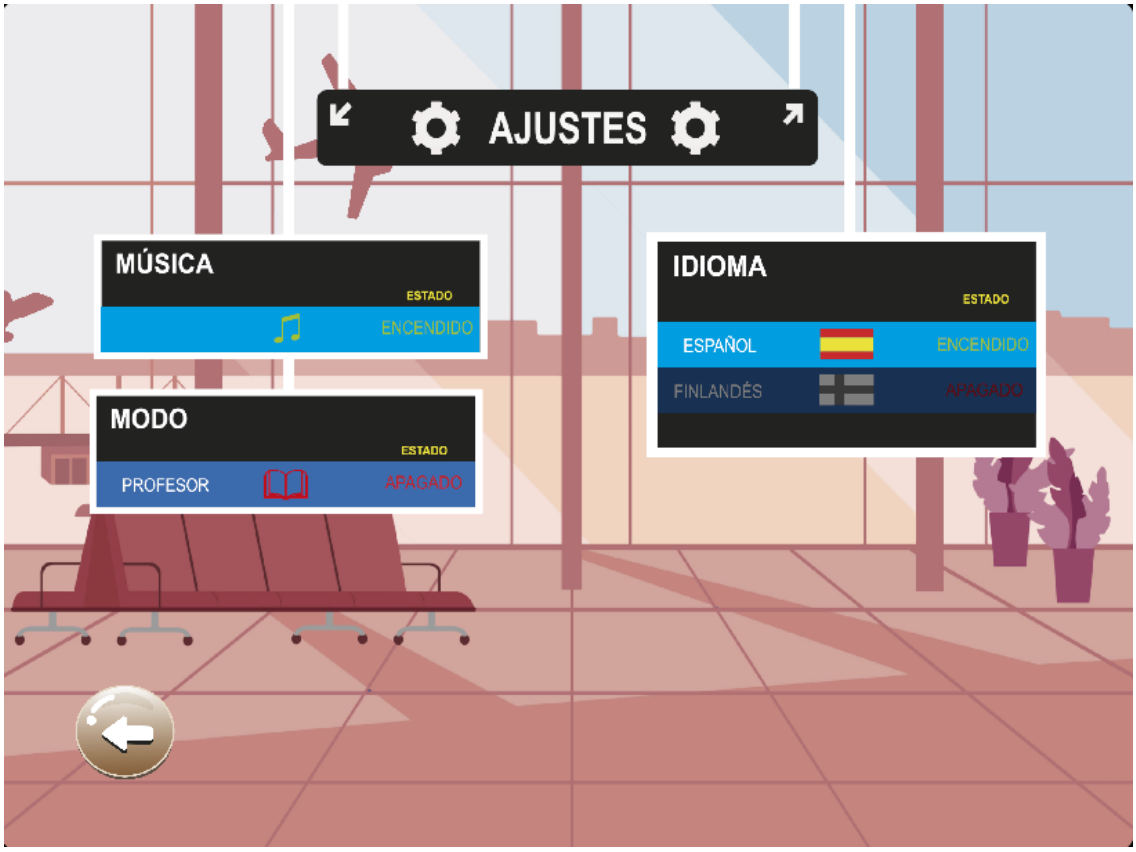

Figure 40. Settings menu

6 Summary

This thesis has explored the intersection of artificial intelligence, educational technology, and inclusive learning through the design and development of a mobile application aimed at supporting children with special educational needs. The application, structured as a world-travel adventure with a variety of mini-games, seeks to create an engaging and adaptive learning environment where cognitive and linguistic development are encouraged through interaction, repetition, and positive reinforcement. At the core of this educational experience lies the integration of AI technologies, particularly speech and image recognition, which enable the app to become an intelligent, responsive companion. This capability allows it to adapt to each child's pace and provide real-time, personalized feedback based on both verbal input and emotional cues, making the learning process more personalized and unique.

The theoretical component of the thesis provided a foundation for understanding how serious games and artificial intelligence can transform traditional learning methodologies. The thesis also covered the technological landscape that makes this possible, addressing both game engines and backend development practices and the integration of external AI services. These technical discussions were essential not only to build the final application but also to demonstrate the complexity and potential of combining different AI tools in a coherent and pedagogically meaningful system.

Throughout the development process, several challenges emerged, such as crafting effective prompts to guide the AI image recognition for accurate sentiment analysis, or selecting the most appropriate text-to-speech model to generate a voice that could function as a supportive companion for users. Although the final prototype demonstrates a functional and promising educational tool, there is still room for improvement. For instance, refining the emotional analysis models, expanding the variety of voice-based activities, and deploying the backend infrastructure on a cloud platform, rather than relying on a local machine, would significantly enhance performance and user experience.

In essence, this thesis demonstrates that by combining pedagogical insight with AI-driven interaction, it is possible to offer children with special needs a novel and effective way of learning that goes beyond traditional methods. More than just a technical project, the app aspires to become a real educational tool that listens, understands, and supports children with special needs on their learning journeys. The work presented here sets the groundwork for future research and development in this area, opening the door to more inclusive, adaptive, and intelligent educational technologies.

References

Acar, Y. 2024 What's API ? API Types, Most Popular API Services, REST vs SOAP : What's the Difference. Medium. Retrieved on 1 April 2025. Available at https://medium.com/@yusufacarr18/whats-api-api-types-most-popular-api-services-rest-vs-soap-what-s-the-difference-1bd6a685afa1

Acodez. 2023. The Role of APIs in Modern Web Development. Acodez. Retrieved on 1 April 2025. Available at: https://acodez.in/apis-in-modern-web-development

Altexsoft. 2022. REST API in action. Retrieved on 1 April 2025. Available at https://www.altexsoft.com/blog/rest-api-design/

Arterburn, E. A., Posteher, K. A., Hansom, A. M., Wilson, S. N., Cecena, F. E., Thompson, W. M., Ralston, R. L. & Thomas, D. M. 2022. Serious Games and Growth Mindsets: An experimental Investigation of a Serious Gaming Intervention. International Journal of Game Based Learning. Retrieved on 20 March. Available at https://eric.ed.gov/?q=Games+For+Growth%3b+Educational+Games+in+the+Classroom.&ft=on&id=EJ1384692

Asociación Americana de Psiquiatría. 2023. Guía de consulta de los criterios diagnósticos del DSM 5. Arlington, VA, Asociación Americana de Psiquiatría. Retrieved on 24 March 2025.

Balibouse, D. 2017. Sophia was made by Hanson Robotics, based in Hong Kong and had previously said she wanted to kill all humans. Retrieved on 27 November 2024. Available at https://www.weforum.org/stories/2017/11/an-interview-with-the-artificially-intelligent-robot-sophia/

Barnard, J. 2023. What is embedding? IBM. Retrieved on 1 March 2025. Available at https://www.ibm.com/think/topics/embedding

Beam, C. 2022. The Taxonomy of Artificial Intelligence and Data Science. Retrieved on 28 October 2024. Available at https://www.ursahealth.com/new-insights/making-sense-of-advanced-analytics-terminology

Belcic, I. & Stryker, C. 2024. What is supervised learning?. IBM. Retrieved on 5 February 2025. Available at https://www.ibm.com/think/topics/supervised-learning#:~:text=Supervised%20learning%2C%20also%20%20known%20as,data%20or%20predict%20outcomes%20accurately

Bergmann, D. & Stryker, C. 2024. What is an attention mechanism? IBM. Retrieved on 25 February 2025. Available at https://www.ibm.com/think/topics/attention-mechanism

Bhandari, S. 2022. Overfitting and Underfitting. Retrieved on 23 February 2025. Available at https://thecorrelation.in/overfitting-and-underfitting/

Brian, S. S. 2024. Definición de los trastornos del Desarrollo. MSD. Retrieved on 24 March 2025. Available at https://www.msdmanuals.com/es/hogar/salud-infantil/trastornos-del-aprendizaje-y-del-desarrollo/definici%C3%B3n-de-los-trastornos-del-desarrollo

Brown, S. 2021. Machine learning, explained. MIT. Retrieved on 3 February 2025. Available at https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained

Buchanan, B. G. 2005. A (Very) Brief History of Artificial Intelligence. AI Magazine, 26(4), 53. Retrieved on 28 October 2024. Available at https://doi.org/10.1609/aimag.v26i4.1848

Chappotin, D. 2022. Trastornos del neurodesarrollo: concepto, tipos y tratamiento. Neuron. Retrieved on 24 March 2025. Available at https://neuronup.com/estimulacion-y-rehabilitacion-cognitiva/trastornos-del-neurodesarrollo/trastornos-del-neurodesarrollo-concepto-tipos-y-tratamiento/

Chen, M. 2023. 6 Common AI Model Challenges. Oracle. Retrieved on 23 February 2025. Available at https://www.oracle.com/ca-en/artificial-intelligence/ai-model-training-challenges/

Chen, M. 2024. What Is Machine Learning? Oracle. Retrieved on 3 February 2025. Available at https://www.oracle.com/ng/artificial-intelligence/machine-learning/what-is-machine-learning/

Chowdhary, K.R. 2020. Natural Language Processing. In: Fundamentals of Artificial Intelligence. New Delhi: Springer. Retrieved on 27 February 2025. Limited availablility at https://doi.org/10.1007/978-81-322-3972-7_19

Curtis, A. 2024. What is a Computer Server?. Splunk Blogs. Retrieved on 30 March 2025. Available at https://www.splunk.com/en_us/blog/learn/computer-servers.html

Forbes 2024. How has AI influenced the teaching and learning process in classrooms? Retrieved on 28 March 2025. Available at https://www.forbes.com/advisor/education/it-and-tech/artificial-intelligence-in-school/

Girdhar, A. 2025. Hume AI vs ElevenLabs: Comparing Two Expressive Text-to-Speech Models. Appypie design. Retrieved on 10 April 2025. Available at: https://www.appypiedesign.ai/blog/hume-ai-vs-elevenlabs#key-comparison

Goodwin, M. 2024. What is an API (application programming interface)?. IBM. Retrieved on 1 April 2025. Available at https://www.ibm.com/think/topics/api

Gupta, N., & Mangla, R. 2020. Artificial Intelligence Basics: A Self-Teaching Introduction. Dulles: David Pallai.

Hamilton, I. 2024. Artificial Intelligence In Education: Teachers' Opinions On AI In The Classroom. Forbes. Retrieved on 28 March 2025. Available at https://www.forbes.com/advisor/education/it-and-tech/artificial-intelligence-in-school/

Hardwood, T. & Maltby, J. & Mukaetova-Ladinska, E. 2019. Types of Artificial Intelligence (AI) and their progression. Retrieved on 1 January 2025. Available at https://www.researchgate.net/figure/Types-of-Artificial-Intelligence-AI-and-their-progression-A-narrow-AI-is-used-as_fig1_334598603

Holdsworth, J. & Scapicchio, M. 2024. What is deep learning? IBM. Retrieved on 24 February 2025. Available at https://www.ibm.com/think/topics/deep-learning

Huawei Technologies Co., Ltd. 2023. Artificial Intelligence Technology. Springer Singapore. Retrieved on 27 October 2024. Available at https://doi.org/10.1007/978-981-19-2879-6

Huston, C. 2019. Game Engines. Medium. Retrieved on 30 March 2025. Available at https://medium.com/@chusto2/game-engines-8763c6782eef

IBM. The principles Samuel developed laid the groundwork for a series of breakthroughs in artificial intelligence at IBM during the 1990s. Retrieved on 29 October 2024. Available at https://www.ibm.com/content/dam/connectedassets-adobe-cms/worldwide-content/arc/cf/ul/g/c5/7d/5322_as_playingcheckers.component.item-horizontal-with-media-right-nocrop-xl.ts=1702909368058.jpg/content/adobe-cms/us/en/history/early-games/jcr:content/root/table_of_contents/body/content_section_styled/content-section-body/item_horizontal_grou_392733260/items/item_horizontal_with/image

IBM 2021 What is a neural network? Retrieved on 24 February. Available at https://www.ibm.com/think/topics/neural-networks

IBM 2023a. Understanding the different types of artificial intelligence. Retrieved on 14th January 2025. Available at: https://www.ibm.com/think/topics/artificial-intelligence-types

IBM 2023b. What are large language models (LLMs)? Retrieved on 5 March 2025. Available at https://www.ibm.com/think/topics/large-language-models

iRender. 2022 Comparison Table – Unity vs Unreal. Retrieved on 30 March 2025. Available at https://irendering.net/unreal-engine-vs-unity-3d-which-engine-is-best-choice-for-your-game/

Jajal, T. 2018. Distinguishing between Narrow AI, General AI and Super AI. Medium. Retrieved on 1 January 2025. Available at https://medium.com/mapping-out-2050/distinguishing-between-narrow-ai-general-ai-and-super-ai-a4bc44172e22

Kalra, R. 2024. Introduction to Transformers and Attention Mechanisms. Medium. Retrieved on 26 February 2025. Available at https://medium.com/@kalra.rakshit/introduction-to-transformers-and-attention-mechanisms-c29d252ea2c5

Khurana, D., Koli, A., Khatter, K. & Singh, S. 2023. Natural language processing: state of the art, current trends and challenges. Multimed Tools Appl. Retrieved on 27 February 2025. Limited availability at https://link.springer.com/article/10.1007/s11042-022-13428-4#citeas

Kolena. 2024. Model Training in AI/ML: Process, Challenges, and Best Practices. Retrieved on 23 February 2025. Available at https://www.kolena.com/guides/model-training-in-ai-ml-process-challenges-and-best-practices/

Lane, J. From Duolingo to Wordle: how educational games are changing the way we learn. NCFE. Retrieved on 19 March 2025. Available at https://www.ncfe.org.uk/all-articles/how-educational-games-are-changing-the-way-we-learn/

Lee, A. 2019. Why NLP is important and it'll be the future – our future. Towards data science. Retrieved on 8 March 2025. Available at https://towardsdatascience.com/why-nlp-is-important-and-itll-be-the-future-our-future-59d7b1600dda/

Linares-Pellicer, J., Izquierdo-Doménech, J., & Ferri-Molla, I. 2023. The new revolution of Generative AI models. GitBook. Retrieved on 5 March 2025.

Lynn, S. Example 2D word embedding space, where similar words are found in similar locations. Retrieved on 3 March 2025. Available at https://www.shanelynn.ie/get-busy-with-word-embeddings-introduction/

Marie-Astrid. 2022. Duolingo allows you to learn languages through play. Retrieved on 22 March 2025. Available at https://oneworldbeyondborders.com/en/duolingo-review-free-application-language/

Maxime. 2019. What is a Transformer? Medium. Retrieved on 25 February 2025. Available at https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04

McCarthy, J. 2007. What is artificial Intelligence?. Standford: Computer Science Department Stanford University, 2-7.

McCullum, N. 2020. An illustration of an artificial neuron. Retrieved on 24 February. Available at https://www.freecodecamp.org/news/deep-learning-neural-networks-explained-in-plain-english/

Medewar, S. 2023. The Role Of Natural Language Processing In eLearning. eLearning Industry. Retrieved on 30 March 2025. Available at https://elearningindustry.com/the-role-of-natural-language-processing-in-elearning

Melcher, K. 2021. Example of a fully connected, feedforward neural network. Retrieved on 24 February 2025. Available at https://www.knime.com/blog/a-friendly-introduction-to-deep-neural-networks

Merritt, R. 2022. What Is a Transformer Model? NVIDIA. Retrieved on 25 February 2025. Available at https://blogs.nvidia.com/blog/what-is-a-transformer-model/

Michael, S. 2025. GPT-4.5 explained: Everything you need to know. TechTarget. Retrieved on 18 March 2025. Available at https://www.techtarget.com/whatis/feature/GPT-45-explained-Everything-you-need-to-know#:~:text=GPT%2D4.5%2C%20OpenAI's%20latest%20model,generation%2C%20multilingual%20proficiency%20and%20more.

Monteclaro, A. J. 2023. What Is a Server? | Definition, Types, and Features. ServerWatch. Retrieved on 30 March 2025. Available at https://www.serverwatch.com/guides/what-is-a-server/

Mucci, T. 2024. The history of artificial intelligence. IBM. Retrieved on 27 November 2024. Available at https://www.ibm.com/think/topics/history-of-artificial-intelligence.

Ombretta, G., Palazzi, C. E., Ciman, M., Galiazzo, G., Franceschini, S., Ruffino, M., Gori, S. & Facoetti, A. 2017. Serious Games for Early Identification of Developmental Dyslexia. Computers in Entertainment. 15, 2, Article 4. Retrieved on 24 March 2025. Limited availability at http://dx.doi.org/10.1145/2629558

OpenAI 2022. Introducing Whisper. Retrieved on 8 April 2025. Available at https://openai.com/index/whisper/?utm_source=chatgpt.com

OpenAI 2024a. Sora System Card. Retrieved on 18 March 2025. Available at https://openai.com/index/sora-system-card/

OpenAI 2024b. GPT-4o System Card. Retrieved on 8 April 2025. Available at https://openai.com/index/gpt-4o-system-card/

OpenAI 2024c. GPT-4o mini: advancing cost-efficient intelligence. Retrieved on 8 April 2025. Available at https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

OpenAI 2025. GPT-4.5 win rate vs GPT-4o. Retrieved on 18 March 2025. Available at https://openai.com/index/introducing-gpt-4-5/

Oppy, G. & Dowe, D. 2021. The Turing Test. The Stanford Encyclopedia of Philosophy. Winter 2021 Edition. Retrieved on 29 October 2024. Available at https://plato.stanford.edu/archives/win2021/entries/turing-test/

Papanastasiou, G. P., Drigas, A. & Skianis, C. 2022. Serious Games: How do they impact special education needs children. Technium Education and Humanities. 2. 41-58.

Peña-Miguel, N. & Sedano, M. 2014. Educational Games for Learning. University of the Basque Country. Retrieved on 19 March 2025. Available at https://eric.ed.gov/?id=EJ1053979

Reddysetty, S. 2021. Traditional Programming vs Machine Learning. Retrieved on 3 February 2025. Available at https://sravya-tech-usage.medium.com/traditional-programming-vs-machine-learning-e9bbed5e491c

Ronaghan, S. 2018. Deep Learning: Overview of Neurons and Activation Functions. Medium. Retrieved on 24 February 2025. Available at https://srnghn.medium.com/deep-learning-overview-of-neurons-and-activation-functions-1d98286cf1e4

Saxena, A. 2024. Issues in Machine Learning. Applied Roots. Retrieved on 23 February 2025. Available at https://www.appliedaicourse.com/blog/issues-in-machine-learning/

Sheremeta, A. 2024. Large Language Models. Retrieved on 5 March 2025. Available at https://dataforest.ai/blog/large-language-models-advanced-communication

Ștefan, V. 2023. The setup of the Turing test: both Machine and Human try to convince the Tester that they are the “real” intelligence, by exchanging messages. Retrieved on 1 February 2025. Available at https://tudorvladstefan.medium.com/back-in-the-50s-when-artificial-intelligence-was-merely-fiction-turing-introduced-his-eponymous-af64db45baf

Stryker, C. & Holdsworth, J. 2024. What is NLP (natural language processing)?. IBM. Retrieved on 28 February 2025. Available at https://www.ibm.com/think/topics/natural-language-processing

Sutton, R. S. & Barto, A. G. 2014. Reinforcement Learning: An Introduction. A Bradford Book. Cambridge, MA: MIT Press.

Thanh, N. 2023. What Is Backend Development? Medium. Retrieved on 30 March 2025. Available at https://namtheartist95.medium.com/what-is-backend-development-bcc6a15f8472

Tipping Point Media. 2024. AI Gamification: The Future of Interactive Learning Experience. Retrieved on 27 March 2025. Available at https://www.linkedin.com/pulse/ai-gamification-future-interactive-learning-v00ke/

Turing, A. M. 1950. Computing Machinery and Intelligence. Mind 49: 433-460.

University of Reading 2014. Turing Test success marks milestone in computing history. Retrieved on 29 October 2024. Available at https://archive.reading.ac.uk/news-events/2014/June/pr583836.html

University of Silicon Valley. 2020. What is a game engine?. Retrieved on 30 March 2025. Available at https://usv.edu/blog/what-is-a-game-engine/

Van Beek, D. 2025 A roadmap for developing machine learning models and putting them into production. Retrieved on 22 February 2025. Available at https://www.passionned.com/machine-learning/

Verma, N. 2023. How Effective is Gamification in Education? 10 Studies and Examples. Axonpark. Retrieved on 22 March 2025. Available at https://axonpark.com/how-effective-is-gamification-in-education-10-case-studies-and-examples/

Warwick, K. 2012. Artificial Intelligence: the Basics. London: Routledge.

Wisdom, D. 2023. What is natural language processing and how does it work?. Datalinknetworks. Retrieved on 28 February 2025. Available at https://www.datalinknetworks.net/dln_blog/what-is-natural-language-processing-and-how-does-it-work