TECHNISCHE UNIVERSITÄT
CHEMNITZ

Fakultät für Informatik

CSR-24-05

Development of a Frontend for Agents in a Virtual Tutoring System

Sopuluchukwu Divine Obi · Ummay Ubaida Shegupta ·
Wolfram Hardt

August 2024

Chemnitzer Informatik-Berichte

Development of a Frontend for Agents in a Virtual Tutoring System

Master Thesis

Submitted in Fulfilment of the
Requirements for the Academic Degree
M.Sc.

Dept. of Computer Science
Chair of Computer Engineering

Submitted by: Sopuluchukwu Divine Obi
Student ID: 561419
Date: 02.10.2023

Supervising tutors: Prof. Dr. W. Hardt
Ummay Ubaida Shegupta

Abstract

This thesis presents the development of a tutoring agent for an existing virtual tutoring system. The goal of this project is to design and implement an agent that can enhance the performance and efficacy of the tutoring system by providing personalized and adaptive support to the learners. The system architecture and components, as well as the agent architecture and behavior, were designed and developed using the latest technologies and methodologies. The evaluation of the system and the agent was performed through user studies and experiments, which showed significant improvements in the learners' performance and engagement. The contributions of this work include the development of a novel tutoring agent and the integration of it into an existing virtual tutoring system, as well as the demonstration of the effectiveness of the agent through user studies and experiments. The limitations and future directions for this work are also discussed.

Keywords: Artificial Intelligence, Virtual agent, Natural Language Processing, Avatar, GUI

Content

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
MOOCs	Massive Open Online Courses
API	Application Programming Interface
ITS	Intelligent Tutoring Systems
AI	Artificial Intelligence
ML	Machine Learning
NLP	Natural Language Processing
EQ	Emotional Intelligence
APA	Animated Pedagogical Agent
PA	Pedagogical Agents
SRL	Self-Regulated Learning
CE	Content Evaluations
MPTG	Monitoring Progress Toward Goals
JOL	Judgments Of Learning
FOK	Feelings Of Knowing
HTML	HyperText Markup Language
JS	JavaScript
CSS	Cascading Style Sheets
AJAX	Asynchronous JavaScript and XML
GUI	Graphical User Interface
API	Application Programming Interface
PALD	Pedagogical Agents-Levels of Design
SVG	Scalable Vector Graphics
UI	User Interface
IDE	Integrated Development Environment
URL	Uniform Resource Locator
WebP	Web Picture format
UX	User Experience

1 Introduction

1.1 Motivation

In recent years, the number of students in a single lecture has drastically increased, making it increasingly difficult for lecturers to provide individualized support for each student. This lack of individualized attention and guidance has been shown to lead to weak learning outcomes, high dropout rates and overall dissatisfaction among students [1]. As much as having one teacher per student would be the ideal solution, it is simply not feasible due to financial and organizational restraints. Despite these restrictions, there are several methods that can be employed to mitigate the effects of large-scale lectures. For example, using technology-enabled learning tools [2] such as MOOCs can help to create a more interactive learning environment. Additionally, utilizing student feedback and surveys can allow lecturers to get better understanding of the learning experience and to tailor lectures accordingly [3]. Furthermore, implementing small-group activities and student-led discussion sessions can enable students to engage in meaningful discussions and build up their confidence in the subject [4].

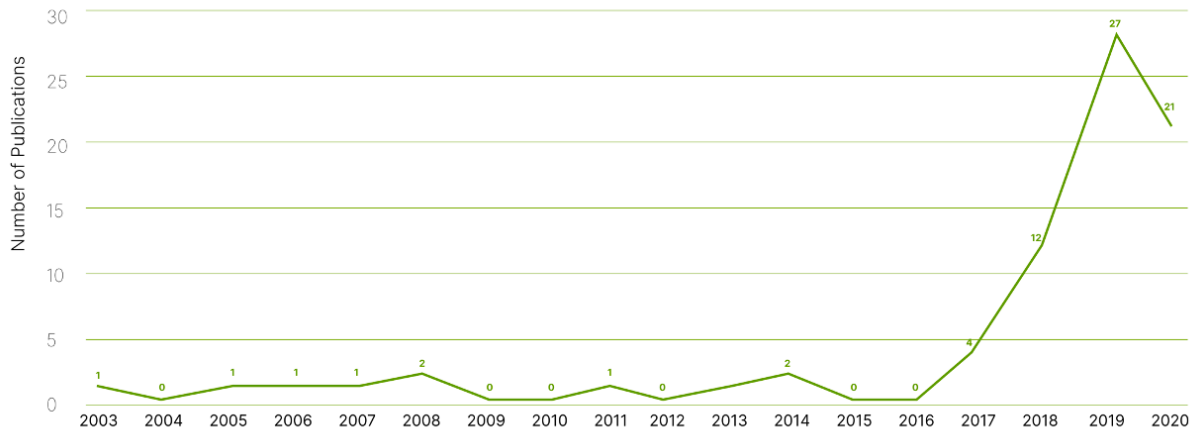

Figure 1: Publications on educational chatbots per year [5].

According to a systematic literature review by Wollny et al. [5], chatbots are increasingly popular in education and have potential as tutoring agents, learning companions, and assessment tools. The number of publications on educational chatbots has significantly increased in recent years, indicating a growing interest in exploring their potential as an educational technology. The study identified articles on chatbots in education published between 2008 and 2020, with a steep increase in the number of publications in recent years, suggesting that chatbots are gaining momentum as a promising technology in education.

As advances in technology continues, so does its application and adaptation. One of those adaptations is in educational sector which is using tutoring applications [6]. This chapter covers the introduction of tutoring agents and virtual tutoring systems.

1.2 Problem Statement

In recent times, virtual tutoring systems have witnessed a surge in popularity due to their ability to provide a streamlined and cost-efficient means of delivering personalized educational content to learners. Despite their widespread adoption, a significant drawback is often noted – the absence of immediate, tailored feedback and guidance for learners, which can potentially impede their overall effectiveness. A viable solution to address this challenge lies in the integration of tutoring agents. These intelligent software entities possess the capacity to furnish personalized support and guidance, meticulously aligned with the distinctive requirements and preferences of each learner [7].

The focal point of this study is an existing virtual tutoring system, which, regrettably, mirrors the common trend. While it offers a diverse range of educational resources and activities, it falls short in providing an engaging tutoring agent that users find compelling and readily interact with. Consequently, there exists a notable opportunity for improvement through the development and integration of a purpose-built tutoring agent. Such an addition holds the promise of significantly amplifying the system's effectiveness and subsequently elevating the overall learning experience for its users.

In light of this pressing issue, the core objective of this study emerges: to delineate the process of designing and implementing a sophisticated frontend for agents within a Virtual Tutoring System. Such a system endeavors not only to be effective but also seeks to be finely attuned to the idiosyncratic learning styles and preferences of individual learners. By doing so, the potential for enhanced learning outcomes and heightened user satisfaction amongst learners is not only promising but highly probable [8].

This study thus stands at the intersection of advancing educational technology and personalized learning experiences, offering a pivotal contribution towards more effective and engaging virtual tutoring systems. Through a meticulous exploration of the design and development process, this research endeavors to not only rectify an existing shortcoming but also chart a course towards a more fruitful and rewarding educational journey for learners using this virtual tutoring system.

1.3 Objectives

The primary goal of this thesis is to enhance an established virtual tutoring system by introducing a tutoring agent. This addition is intended to elevate both the effectiveness and overall user experience of the system. To achieve this overarching objective, we will focus on pursuing the following sub-goals:

Literature Review: The primary goal of this study is to conduct a thorough examination of the existing body of literature surrounding virtual tutoring systems, intelligent tutoring systems, and tutoring agents. This encompasses an extensive exploration of established technologies, methodologies, as well as an analysis of pertinent pedagogical theories and models that underpin computer-based instruction.

Development of the Tutoring Agent: The second aim is to conceive and implement a sophisticated tutoring agent with the capacity to engage students in a meaningful and personalized manner. This agent will possess the ability to offer tailored guidance, constructive feedback, and targeted support to individual learners, considering their unique learning preferences, requirements, and performance metrics [9]. The ultimate objective is to craft an interactive and individualized learning experience.

Integration and Impact Assessment: The third objective involves the seamless integration of the newly developed tutoring agent into the existing virtual tutoring system. Subsequent to integration, a comprehensive evaluation will be conducted to appraise the impact of the agent on various facets of student learning. This encompasses an analysis of learning outcomes, assessment of levels of student engagement, measurement of satisfaction levels, and monitoring of retention rates.

Comparative Study: The fourth objective is to perform a comparative analysis to determine the effectiveness of the tutoring agent. This will be accomplished by juxtaposing the performance of the system with the integrated agent against the performance of the existing system in the absence of the agent. Additionally, comparisons may be drawn with other tutoring agents or instructional approaches delineated in the existing literature. The intention is to provide valuable insights into the relative strengths and advantages of the developed agent.

Strengths, Weaknesses, and Recommendations: The concluding objective centers around a rigorous evaluation of the developed tutoring agent. This encompasses the identification of its strengths, delineating areas where it excels in supporting student learning. Moreover, weaknesses and potential areas for enhancement will be spotlighted. Grounded in this assessment, the study will culminate with a set of

recommendations for future refinements and expansions of the tutoring agent, with the goal of refining its functionality and maximizing its impact on educational outcomes.

By systematically addressing these objectives, this study aspires to furnish valuable insights and innovations to the domain of virtual tutoring systems and intelligent tutoring agents, ultimately elevating the learning experiences of students in virtual environments.

This thesis focuses on creating a tutoring agent for a pre-existing virtual tutoring system that is currently operational within a school setting. The system has already undergone the design and implementation phases and is actively utilized in the educational environment. The purpose of the tutoring agent is to assist secondary school students specifically in domains where it can be most beneficial, particularly in the area of self-tests. It's important to note that this thesis does not seek to extend its findings beyond the specified domains or user groups. Instead, its primary aim is to demonstrate the potential and effectiveness of the developed tutoring agent within this specific educational context.

The primary objective of this master's thesis is to create a tutoring agent integrated into an existing virtual tutoring system. To accomplish this objective, a structured methodology and research approach will be employed to steer the conception, construction, and assessment of the tutoring agent.

This methodology combines elements of design science research and empirical evaluation. Design science research focuses on devising inventive solutions, including systems, models, and methodologies, to tackle real-world issues within a specific field. It emphasizes practical problem-solving through the development of tangible artifacts. On the other hand, empirical evaluation employs data collection and analysis to affirm the efficacy and user-friendliness of a system or artifact. This research method is crucial in gauging how well the tutoring agent functions in practice and how it impacts the learning experience. By integrating these two methodologies, this thesis aims to create an effective and practical tutoring agent within the virtual tutoring system.

The research methodology will consist of three distinct phases: the initial design phase, followed by the subsequent development phase, and culminating in the evaluation phase.

a. Design Phase

In the design phase, the existing virtual tutoring system will be analyzed to identify how the tutorial agent can fit in naturally. Based on this analysis, the requirements and

specifications for the tutoring agent will be defined. Additionally, a conceptual design of the agent will be created to guide the development phase.

b. Development Phase

During the development phase, we will bring the tutoring agent to life and seamlessly incorporate it into the established virtual tutoring system. This agent will be meticulously crafted to offer tailored and responsive tutoring experiences, finely attuned to each learner's unique requirements and preferences. This developmental stage encompasses the careful selection and seamless integration of cutting-edge artificial intelligence methodologies, including machine learning, natural language processing, and sophisticated decision-making algorithms. These techniques will empower the agent to dynamically adapt and cater to the specific learning journey of each individual.

c. Evaluation Phase

During the assessment phase, we will employ empirical techniques to gauge the proficiency and user-friendliness of the tutoring agent. This evaluation will encompass user studies and experiments aimed at appraising the agent's influence on learning achievements, user contentment, and involvement. The benchmarks and criteria for this assessment will draw from well-established pedagogical theories and models, as well as the stipulated system requisites and specifications.

During the course of this research, we are committed to upholding the highest standards of ethical research conduct. We will ensure that all participants provide informed consent, and their privacy and confidentiality will be treated with utmost care. Additionally, our research will strictly adhere to pertinent legal and regulatory frameworks, including but not limited to data protection laws.

This thesis makes several significant contributions across various fields: first and foremost, it advances the realm of virtual tutoring systems by introducing a novel approach to amplify the impact of tutoring. Through the creation of a specialized tutoring agent, this work elevates the quality of the tutoring experience. This agent not only enhances personalized guidance for students but also surpasses the capabilities of conventional tutoring systems.

Furthermore, in the domain of intelligent tutoring systems, this thesis introduces a fresh perspective on crafting an intelligent agent dedicated to aiding the tutoring process. This involves a comprehensive integration of pedagogical theories and models into the

agent's behavior and decision-making processes. This infusion of intelligence substantially augments the agent's capacity to offer superior guidance to students.

In the realm of artificial intelligence, this thesis presents a pioneering method for developing an intelligent tutoring agent. Leveraging machine learning techniques, the agent scrutinizes student performance data, utilizing this information to furnish tailored feedback. Additionally, the agent incorporates natural language processing and dialogue management, enabling seamless and natural interactions between the agent and the student.

Lastly, in the field of educational technology, this thesis introduces a groundbreaking approach to heighten the efficacy of virtual tutoring systems. By introducing the developed tutoring agent, this work tackles the limitations that are inherent in traditional virtual tutoring systems. The result is an enhanced tutoring experience for students, potentially leading to heightened engagement and improved learning outcomes. This collective effort represents a significant stride towards the advancement of educational technology and its positive impact on the learning journey of students.

1.4 Structure of this Thesis

The implementation of this project is divided into the following parts: theoretical background, state of the art of existing solutions, proposed design and implementation, and discussion of results.

The theoretical background section delves into the fundamental principles, theories, and concepts pertinent to the research topic, offering a thorough comprehension of the underlying ideas driving the development of the agents. It establishes a knowledge framework that acquaints readers with the pivotal concepts and principles that underpin the agent's creation.

Moving on to the state-of-the-art segment, it conducts a comprehensive review and assessment of existing solutions, technologies, or methodologies within the realm of the research topic. This section provides an overview of the current landscape of virtual tutoring agents, highlighting their strengths, weaknesses, and any existing gaps. It also elucidates the necessity for further research or enhancements, thereby setting the stage for the proposed solution.

The methodology and implementation section unveils the envisaged design and execution. It outlines the inventive or refined approach, system architecture, algorithms, or frameworks conceived to address the identified gaps. By elucidating the

design rationale, technical intricacies, and potential advantages of the proposed solution, this section showcases the innovative essence of the research.

Shifting to the results and evaluation section, it interprets and scrutinizes the outcomes of the research. This entails an examination of how the proposed solution fared in comparison to the objectives outlined in the research questions or hypotheses. The section delves into the import of the findings, potential ramifications for the field, and any unforeseen results or hurdles encountered during the implementation. It also provides a contextual backdrop for potential avenues of future research.

2 Theoretical Background

The Theoretical Background section is dedicated to examining the fundamental principles that form the basis of our project. Within this chapter, we venture into the realm of tutoring systems, encompassing both intelligent tutoring systems and tutoring agents. Moreover, we closely examine the pedagogical theories that serve as the cornerstone for our initiative. Through a thorough exploration of these theoretical frameworks, our aim is to establish a robust groundwork upon which our innovative educational technology is constructed. This in-depth comprehension will guide the process of designing and implementing tutorial agents within our virtual tutoring system, guaranteeing its effectiveness and applicability in various educational settings.

2.1 Virtual Tutoring Systems

Virtual tutoring systems represent a modern educational approach harnessing the power of computer-based technology to augment and personalize the learning process. This surge in popularity owes itself to the rapid advancements in technology and the surging demand for flexible, remote learning solutions. These systems not only facilitate access to education from virtually anywhere but also provide learners with dynamic, interactive experiences. Through personalized interactions and engaging content, virtual tutoring systems have demonstrated their potential to significantly elevate the overall quality of learning outcomes. This transformative shift in education leverages the strengths of technology to create a more inclusive, interactive, and effective learning environment for students of all backgrounds and levels [10].

Virtual tutoring systems can take many forms, ranging from simple chatbots to sophisticated intelligent tutoring systems (ITS) [11]. Chatbots serve as elementary versions of virtual tutoring systems, offering fundamental information and guidance to learners. These systems operate within the confines of predetermined rules and responses, lacking the capability to dynamically adjust to the unique needs and preferences of individual learners. While proficient in providing foundational support, they fall short in delivering personalized, tailored learning experiences that cater to the diverse requirements of each user [12].

Intelligent tutoring systems (ITS) represent a cutting-edge evolution in virtual tutoring. These sophisticated systems are meticulously crafted to deliver highly personalized instruction, tailoring the learning experience to the unique needs of each individual learner. Employing a wide array of artificial intelligence (AI) techniques including natural language processing (NLP), machine learning (ML), and expert systems, ITS go beyond conventional tutoring systems. They exhibit a remarkable ability to discern

and adapt to diverse learning styles, pacing preferences, and individual inclinations. Through this adaptive approach, ITS offer not only targeted instruction but also furnish invaluable feedback and guidance, ensuring a finely tuned and enriching learning journey for every student [13].

Virtual tutoring systems can be used in various contexts, including middle school education, higher education, corporate training, and professional development. They can support various types of learning, such as self-directed learning, blended learning, and online learning [14].

Virtual tutoring systems revolutionize the landscape of education, ushering in a myriad of benefits that significantly enhance the learning journey. At the forefront of these advantages is their remarkable capacity to provide a tailored and highly individualized educational experience. By adapting to the specific needs and learning pace of each student, these systems ensure that educational material is not only comprehensively understood but also effectively absorbed. This personalized approach cultivates a profound sense of engagement and motivation among learners, forging a deeper connection with the subject matter and a heightened investment in their own educational progress.

In addition to customization, virtual tutoring systems introduce an invaluable element of instant feedback and guidance. This real-time interaction allows learners to promptly address queries or rectify misconceptions, thus propelling the learning process forward with precision and efficacy. This instantaneous responsiveness becomes a cornerstone for accelerated comprehension and mastery of the material, leading to more fruitful educational outcomes.

Furthermore, virtual tutoring systems open the gateway to a treasure trove of diverse learning resources. From engaging multimedia content to interactive exercises, these systems enrich the educational journey by providing a multifaceted learning experience. Learners are presented with a dynamic array of tools and materials that cater to various learning styles, ensuring a well-rounded and comprehensive understanding of the subject matter.

The flexibility and convenience afforded by virtual tutoring systems are equally noteworthy. Learners have the freedom to access educational content at their own convenience, seamlessly integrating learning into their unique schedules and lifestyles. This adaptability empowers learners to take an active role in their education, fostering a sense of ownership and autonomy that invariably leads to a more effective and rewarding learning experience.

In summation, virtual tutoring systems stand as a beacon of educational advancement, offering a holistic and tailored approach to learning. Through personalized instruction, immediate feedback, and a rich tapestry of resources, these systems revolutionize the educational landscape, equipping learners with the tools they need to thrive in their academic pursuits. The flexibility they afford ensures that education becomes a seamless part of a learner's life, ultimately culminating in an educational experience that is not only effective but also deeply fulfilling.

2.2 Intelligent Tutoring Systems

Intelligent Tutoring Systems (ITS) represent a sophisticated class of computer-based learning environments designed to offer tailored instruction and feedback to students. Powered by cutting-edge artificial intelligence and machine learning methodologies, ITS harness the capability to intricately model the extensive knowledge, skills, and proficiencies of a domain expert or educator. This enables them to deliver highly personalized and adaptive guidance to students, finely attuned to their distinct learning requirements and aptitudes. Through the dynamic utilization of advanced technologies, ITS not only enhance the educational experience but also foster a more profound and effective learning journey for each individual learner [15].

Intelligent Tutoring Systems (ITS) are comprehensive educational platforms that encompass four fundamental components, each playing a crucial role in the learning process. At the core lies the domain model, which encapsulates the breadth of knowledge and skills required to proficiently navigate a specific subject area, ranging from foundational subjects like mathematics to more specialized domains like physics or programming. Complementing this, the student model intricately maps out the unique knowledge, aptitudes, and capabilities of each individual learner. This personalized profile serves as the compass guiding instructional adjustments and feedback delivery, ensuring a tailored learning experience [16].

The pedagogical model stands as the strategic backbone of the system, orchestrating the methodologies and approaches employed to effectively impart knowledge. It encompasses a spectrum of techniques, from elucidative worked examples to insightful hints and informative feedback loops. This dynamic array of instructional strategies adapts in real-time, aligning with the student's progress and learning pace. The user interface serves as the gateway for students to engage with the system. Beyond its functional role, it creates an immersive learning environment, facilitating seamless interaction with the material. Additionally, it acts as a conduit for the timely delivery of constructive feedback and personalized instructions, further enhancing the learning journey [16].

Intelligent Tutoring Systems find applicability across a diverse range of educational environments, encompassing secondary schools, higher education institutions, as well as vocational training and professional development programs. Their remarkable utility shines especially in fields demanding a heightened level of personalized instruction, with prime examples being disciplines like mathematics and foreign languages. These systems excel in providing tailored learning experiences, optimizing comprehension and proficiency in these critical subjects [17].

Intelligent Tutoring Systems offer a range of valuable advantages within the realm of education. To begin with, these systems excel at delivering tailored instruction to students, honing in on their unique strengths and areas that require improvement. This personalized approach significantly contributes to enhanced learning outcomes. Moreover, the immediate feedback provided by Intelligent Tutoring Systems plays a pivotal role in the learning process. This timely guidance empowers students to swiftly recognize and rectify any misconceptions or errors in their comprehension, fostering a deeper grasp of the subject matter. Furthermore, these systems equip educators with invaluable insights into their students' learning trajectories and specific needs. Armed with this knowledge, teachers are better positioned to customize their teaching methods, ensuring a more effective and targeted educational experience for each student. In essence, Intelligent Tutoring Systems represent a dynamic tool that not only supports students in their learning journey but also empowers educators to optimize their teaching strategies [18].

While Intelligent Tutoring Systems hold immense promise, they do come with their share of challenges. One of the primary hurdles lies in accurately modeling the extensive knowledge and skills necessary for a specific domain. This undertaking demands substantial domain expertise, often consuming considerable time and resources. Additionally, achieving precise student modeling necessitates the comprehensive gathering and analysis of extensive data on student learning patterns. Lastly, there is a critical requirement for the development of highly effective pedagogical models and instructional strategies capable of not only proficiently imparting knowledge but also seamlessly adapting to the unique learning needs of individual students. These challenges collectively underscore the complexity and depth of considerations involved in the development and utilization of Intelligent Tutoring Systems [19].

2.3 Tutoring Agents

The application of intelligent tutoring systems (ITS) has demonstrated considerable potential in delivering tailored learning experiences to students. Within an ITS framework, a central element that plays a pivotal role is the tutoring agent. This

sophisticated component is instrumental in facilitating personalized guidance and support, contributing significantly to the overall effectiveness and adaptability of the learning environment [20], the intelligent software component, known as a virtual tutor, engages with students through natural language dialogue, offering tailored feedback, personalized guidance, and comprehensive support throughout the learning process [21].

Tutoring agents are generally categorized into two primary types: rule-based and data-driven. Rule-based tutoring agents rely on a predefined set of rules to ascertain the most suitable response to a student's input. These rules may stem from expert knowledge, established pedagogical theories, or a fusion of both. Typically, rule-based agents are tailored for specific domains and their effectiveness is contingent upon the precision of the rules applied in the given context. This approach, while structured and reliable, operates within the framework of established rulesets, potentially limiting adaptability to unforeseen scenarios [22].

Data-driven tutoring agents leverage advanced machine learning algorithms to meticulously examine student data, enabling them to discern and implement the most suitable responses. This distinctive approach endows these agents with heightened adaptability, allowing them to cater to a diverse range of student profiles and accommodate various learning styles. However, it's worth noting that the effectiveness of data-driven agents hinges on the sheer volume and quality of the data they are trained on, underscoring the critical importance of rich, high-quality datasets in optimizing their performance and response accuracy [23].

Tutoring agents can also be designed to incorporate various pedagogical strategies such as scaffolding, mastery learning, and metacognitive support [24]. Scaffolding refers to the gradual reduction of support as the student gains mastery over a concept. Mastery learning involves providing repeated opportunities for students to master a concept before moving on to the next one. Metacognitive support involves helping students develop awareness of their own thinking and learning strategies [25]. One of the key challenges in developing tutoring agents is ensuring that the agents can understand and respond appropriately to student inputs. This requires the use of natural language processing (NLP) techniques to analyze student inputs and generate appropriate responses. Another challenge is ensuring that the agents can provide feedback that is both accurate and understandable to the student.

Despite these challenges, tutoring agents have shown promise in improving student learning outcomes. By providing personalized feedback and guidance, tutoring agents can help students overcome misconceptions, develop deeper understanding of

concepts, and achieve higher levels of mastery [26]. As virtual tutoring systems continue to gain popularity, the development of effective tutoring agents will become increasingly important in providing high-quality personalized learning experiences to students [5].

2.4 PALD Model Theory

The Pedagogical Agents-Levels of Design (PALD) model stands as a substantial milestone in the domain of technology-enhanced learning. It introduces a systematic and comprehensive approach to conceptualizing and integrating pedagogical agents into Virtual Learning Environments (VLEs). This pioneering model, unveiled by Heidig and Clarebout in 2011, has garnered considerable recognition for its meticulously structured framework. It serves as a vital bridge between pedagogical principles and technological applications, guaranteeing that pedagogical agents play a substantive and impactful role in enhancing the overall learning journey of students [27][28].

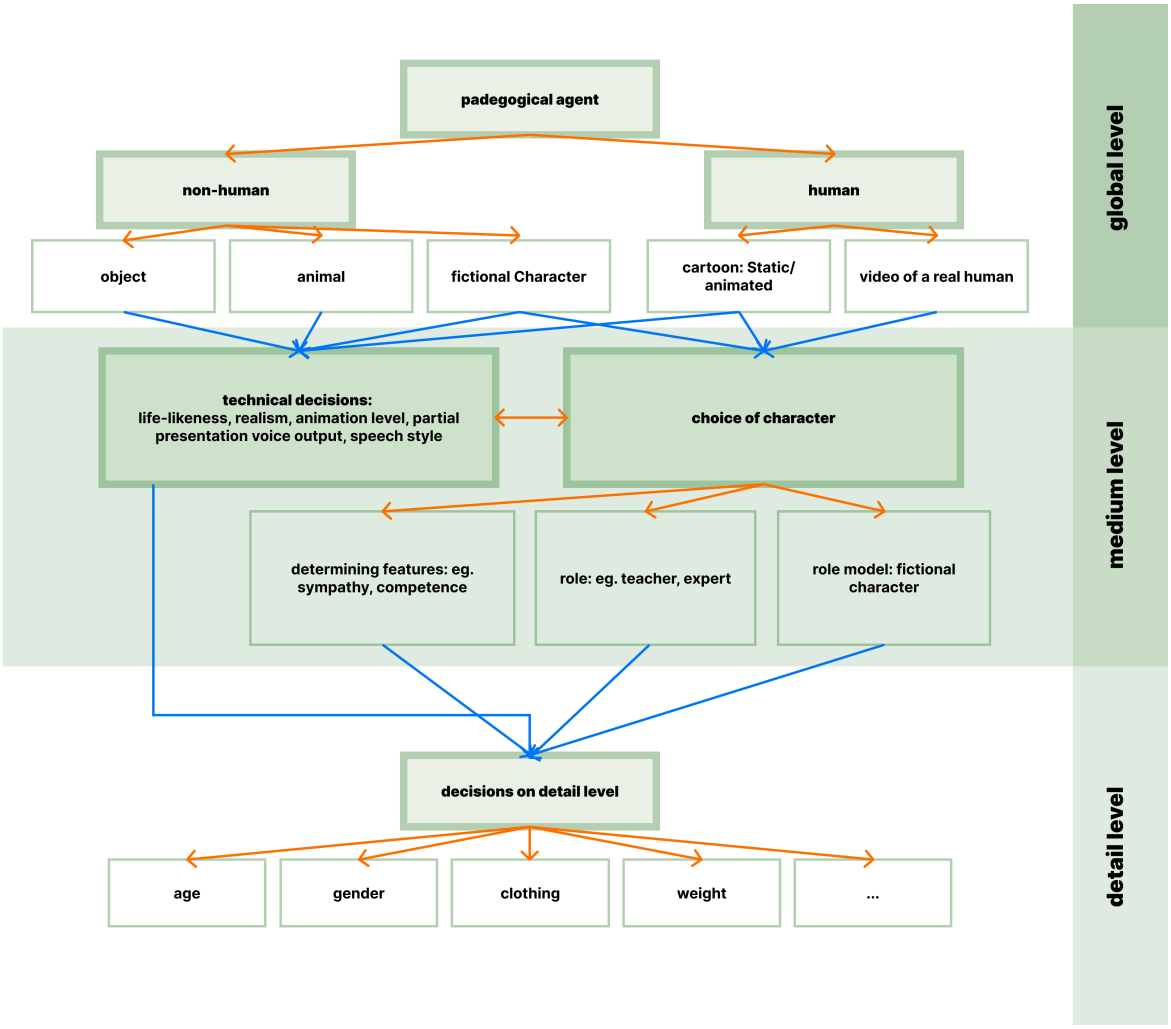

Figure 2: Pedagogical Agents Levels of Design (PALD) Model [30]

At the heart of the PALD model are its three distinct levels of design, each serving a critical role in shaping the visual and functional aspects of pedagogical agents:

Global Level: The global level is the foundational level of the PALD model. It tackles the overarching question of how the pedagogical agent should be perceived within the learning environment. Designers grapple with the fundamental decision of whether the agent should take on a human-like form or exist as a non-human entity. This decision has profound implications for how learners interact with and relate to the agent. By choosing whether the agent is human or non-human, designers are essentially defining the agent's identity within the educational context.

Medium Level: Transitioning to the medium level of design, the emphasis shifts towards the intricate technical facets governing the agent's overall design. Designers meticulously consider elements like animation intensity, degree of lifelikeness, partial presentation, level of realism, speech style nuances, and the nature of voice output. This crucial phase is primarily concerned with defining the agent's behavioral patterns and communication style. It is at this juncture that designers endeavor to strike a harmonious equilibrium between infusing a sense of authenticity while ensuring sustained learner engagement. The choices and determinations made at this medium level exert a pivotal influence on how effectively the agent engages with learners, profoundly augmenting their comprehensive learning encounter.

Detail Level: The detail level delves into the finer points of the agent's visual appearance. Designers consider attributes such as age, gender, clothing, weight, and other specific physical characteristics. Although these may seem subtle, they add depth and nuance to the agent's identity. The detail level allows designers to create a visually appealing and relatable agent, one that learners can connect with on a personal level.

The PALD model recognizes that the design of pedagogical agents extends far beyond mere aesthetics; it encompasses pedagogical and technological considerations that are intrinsically linked to the learning process. An agent's visual representation can influence learner engagement, motivation, and even the perception of the agent's competence [29].

Heidig and Clarebout's groundbreaking model has not only significantly influenced the field but has also paved the way for the integration of other complementary frameworks, such as the Pedagogical Agents – Conditions of Use (PACU) model. This model emphasizes the critical need for personalized design considerations when developing pedagogical agents. This includes a thorough assessment of contextual elements, ranging from the precise role the agent is expected to fulfill, to the unique

attributes of the learners, the characteristics of the learning environment, and the specific content that is being conveyed. This holistic approach ensures that pedagogical agents are finely tuned to cater to the nuanced requirements of each educational setting [30].

The PALD model, distinguished by its multi-tiered methodology, stands as a pivotal landmark in the realm of technology-enhanced learning. It emphatically underscores the profound importance of deliberate and meticulously thought-out design principles in the evolution of pedagogical agents. By adhering to this model and embracing its associated frameworks, researchers and instructional designers are empowered to harness pedagogical agents to their full potential, thereby elevating the educational experience. This integration aligns these agents seamlessly with educational objectives, resulting in a conducive environment for meaningful and captivating interactions within Virtual Learning Environments. This all-encompassing approach to design not only continues to influence the ever-evolving landscape of educational technology but also ensures that pedagogical agents assume a central and indispensable role in guiding learners through their educational journey [31].

3 State of the Art

In recent years, there has been an increasing interest in the development of intelligent tutoring systems and tutoring agents to enhance the learning experience of students. This section presents an overview of the related work and existing systems in the area of virtual tutoring systems and tutoring agents.

3.1 AutoTutor

AutoTutor is a pedagogical agent designed to provide an interactive and personalized learning experience to students. AutoTutor was developed to improve student engagement and learning outcomes in a variety of educational settings. AutoTutor works by utilizing avatars, which are computer-generated characters that can be customized to resemble humans. The avatars are used to interact with the student and provide feedback and guidance. AutoTutor also includes an artificial intelligence (AI) component that allows the pedagogical agent to monitor and adapt its behavior to the needs of the student.

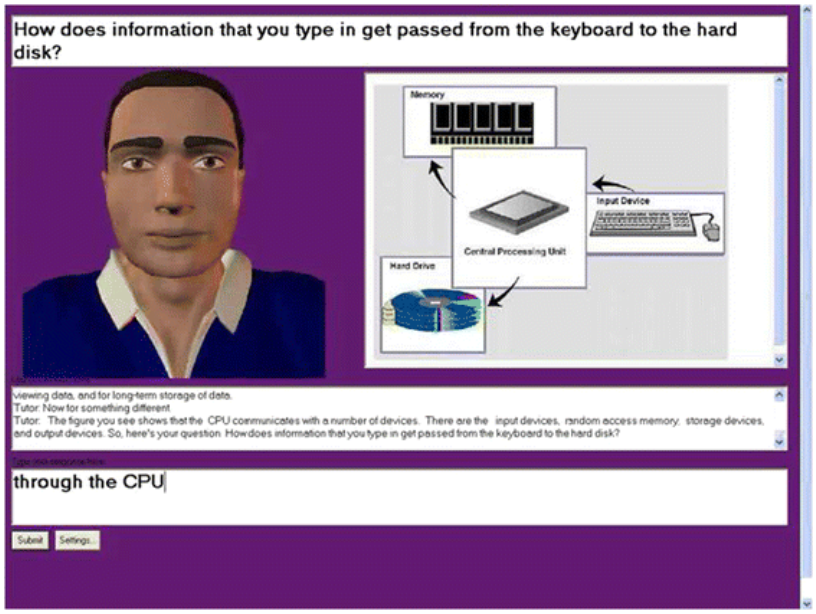

Figure 3: AutoTutor [32]

The AutoTutor system is composed of four components: the avatar, the natural language processing (NLP) module, the AI module, and the pedagogical module. The avatar is the visual representation of the pedagogical agent, and is designed to engage the student and encourage them to interact with the system. The NLP module is responsible for understanding and responding to user input. This module is capable of recognizing and interpreting spoken and written language, which allows it to

communicate with the student in a natural way. The AI module is responsible for the agent's behavior, and is capable of adapting to the student's needs. This module uses machine learning algorithms to improve the pedagogical agent's performance over time. The pedagogical module plays a crucial role in delivering tailored instructional content and constructive feedback to the student, ensuring that the learning experience is finely tuned to individual needs and progress. This component serves as a dynamic guide, adapting its approach to match the student's pace, preferences, and comprehension level, thus fostering a highly effective and personalized learning environment.

The AutoTutor system is designed around the integration of text input from users, complemented by a talking head avatar employing advanced synthesized speech technology. This sophisticated setup enables an exceptionally realistic and engaging conversational experience, as the avatar adeptly addresses user inquiries with a level of naturalness that closely mirrors human interaction.

AutoTutor has garnered widespread utilization across diverse educational environments, showcasing its efficacy in enhancing student engagement and driving positive learning outcomes. Its adaptability and proven track record make it a valuable tool in a range of educational settings, reinforcing its reputation as an influential asset in the realm of technology-enhanced learning [32].

3.2 Guru Biology Tutor

Guru Biology Tutor is an innovative virtual platform meticulously crafted to serve as an indispensable resource for students seeking comprehensive comprehension and practical application of diverse biological concepts. This state-of-the-art virtual tutor is poised to revolutionize the learning experience, providing an interactive and dynamic environment where students can delve deep into the intricacies of biology with ease and confidence. By offering a wealth of curated educational content, Guru Biology Tutor empowers students to not only grasp fundamental biological principles but also apply them effectively in academic pursuits and real-world scenarios. This comprehensive instructional system combines an engaging avatar-driven learning environment with an intelligent agent that can provide personalized instruction and real-time feedback.

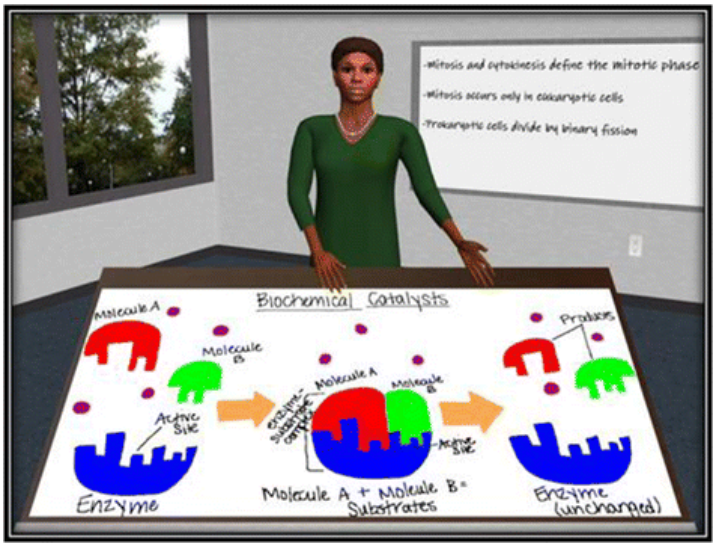

Figure 4: Guru Biology Tutor [32]

The Guru Biology Tutor provides an immersive, interactive learning experience that encourages student engagement and exploration. The system includes a 3D avatar-driven virtual tutor that can be customized to the student's preferences. The avatar provides a conversational interface, allowing students to ask questions and receive immediate responses. The avatar can also provide visual explanations of biological concepts, using 3D models and animations.

The Guru Biology Tutor also incorporates an intelligent agent that provides individualized instruction. The agent monitors the student's progress and suggests appropriate activities. It also provides real-time feedback and guidance, helping the student identify and address areas of difficulty. The agent is also able to diagnose and address misconceptions, providing helpful explanations and further resources.

Alongside the avatar and intelligent agent, the Guru Biology Tutor boasts a diverse array of multimedia resources. Among these, students will find immersive 3D visualizations elucidating complex biological concepts, informative videos, dynamic simulations, and a range of interactive activities. This rich tapestry of resources is meticulously curated to ensure a deeply engaging and interactive learning journey, fostering in students a profound comprehension of the intricate world of biological sciences.

The Guru Biology Tutor is a powerful instructional system that can help students understand and apply biological concepts. The avatar-driven interface and intelligent agent provide an engaging and personalized learning experience, while the multimedia resources help students develop a deeper understanding of biological concepts. This comprehensive system is an effective tool for helping students succeed in their biology studies. [32].

3.3 JEPPY

Engaging in programming is a multifaceted endeavor, demanding proficiency in an array of skills including problem-solving, abstraction, mathematical logic, as well as rigorous testing and debugging techniques. For those new to programming, their understanding often remains in the early stages, potentially leading to challenges in identifying and rectifying syntax errors. These stumbling blocks can pose a hindrance to their advancement and hinder their ability to craft programs that adhere to precise syntactical guidelines. To surmount this hurdle, the advocates of this approach have introduced a comprehensive solution: the conceptualization and subsequent realization of an interactive pedagogical agent, aptly named JEPPY. This innovative tool is poised to serve as an invaluable resource for fledgling programmers, offering invaluable assistance in the formidable task of pinpointing and rectifying syntax errors that may arise during compilation. Through JEPPY's intervention, novice programmers can anticipate a smoother learning curve and enhanced proficiency in crafting syntactically accurate programs.

Figure 5: JEPPY when Instructing to Read Help Carefully [33]

In the meticulous design of agent-based learning environments, a pivotal cognitive aspect to consider is the adept management of control dynamics. This intelligent agent functions as a supportive assistant, focusing on delivering constructive feedback to learners, aiding them in understanding and rectifying their errors. The feedback mechanism is finely attuned, activating when the agent's Emotional Intelligence (EQ) surpasses the threshold of 0.3. The embodiment of this agent takes on an engaging

persona in the form of a whimsical cartoon character, affectionately named JEPPY. JEPPY's interactive repertoire encompasses a diverse range of deictic and affective gestures, deftly combined to form a spectrum of actions that imbue the agent with a remarkably lifelike quality. These expressive gestures, including the encouraging "Thumbs-up," the friendly "Waving," the attentive "Default," the inquisitive "Reading," the affirming "Nodding," and the celebratory "Clapping," collectively contribute to JEPPY's relatability and effectiveness as a learning companion. This thoughtful integration of cognitive processes and interactive behaviors enriches the learning experience within the agent-based environment, fostering a more dynamic and immersive educational journey.

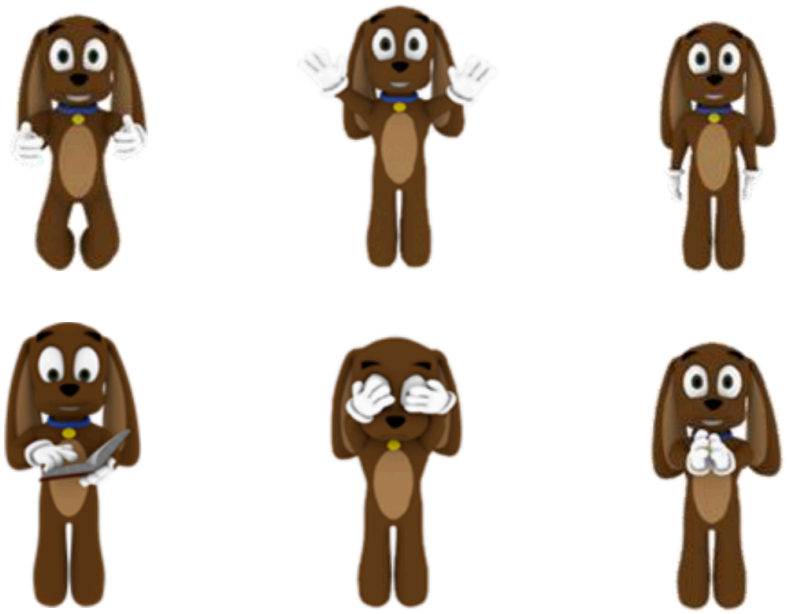

Figure 6: Gestures of JEPPY [33]

The evaluation of JEPPY's components involved a practical test conducted in a controlled laboratory setting with participants. Each participant was provided with a source code containing a series of cascading errors and was tasked with rectifying the issues independently, without seeking assistance from peers or instructors. This approach ensured a comprehensive assessment of JEPPY's functionality under realistic problem-solving conditions.

JEPPY's implementation adhered to the established architecture of a pedagogical agent, meticulously customized to align with its intended purpose. The integration of the pedagogical module into Code::Blocks as a plugin further demonstrated its adaptability to popular development environments. Notably, the compiler-generated errors underwent a preprocessing step to extract only the pertinent information, streamlining the feedback provided to learners. In addition to error handling, an event logger was instrumental in capturing various dimensions of the learning process. It

diligently recorded the preprocessed compiler errors, the code modifications made by the learner, interactions between the learner and the agent, and the resultant error quotient—a metric crucial for assessing the learner's progress and comprehension. This meticulous data collection mechanism enriched the evaluation process, offering valuable insights into the effectiveness and performance of JEPPY's components.

The communication module, implemented in Java, comprises two essential components: the interface and the inference controller. Within this module, the interface serves as the primary point of interaction between the learner and the intelligent agent. Here, elements known as "recalls," which are essentially hints and examples, are presented to the learner in the form of HTML files, enhancing their accessibility and usability within the interface.

On the other hand, the inference controller takes on the pivotal role of retrieving knowledge when intervention is required. It acts as a bridge connecting the pedagogical module with the domain module, facilitating seamless information flow and interaction between these critical components.

The development and implementation of JEPPY, an interactive pedagogical agent designed to assist novice programmers in rectifying syntax errors flagged by the compiler, represent a noteworthy and effective approach. JEPPY's design and functionality have been meticulously crafted to provide learners with efficient and valuable feedback and guidance throughout their programming journey, significantly enhancing the overall learning experience [33].

3.4 Matsuda et al

Learning by teaching is a widely recognized phenomenon across various disciplines, catering to a diverse range of student populations and skill levels. With the advent of advanced learning technologies, researchers now have the tools to delve deeper into the intricate relationship between tutoring activities and their impact on the learning outcomes of tutors.

A significant contribution to this field comes from Matsuda et al., who conducted an insightful investigation into the influence of meta-cognitive support on learning through teaching experiences. One study conducted experiments with students acting as tutors for a teachable agent named Betty's Brain, focusing on river ecosystems. Surprisingly, the presence of a mentor agent showed no discernible effect on the learning outcomes of the tutors.

Another study conducted comparative research on the efficacy of "adaptive" versus "fixed" meta-cognitive assistance in the context of teaching Algebra equations. In this

scenario, student pairs engaged in reciprocal teaching. The "adaptive" form of assistance was finely tuned to the context, while the "fixed" assistance was provided in a more random manner. The results from this classroom-based study provided compelling evidence in favor of the "adaptive" meta-cognitive support, demonstrating its superior effectiveness in enhancing tutor learning compared to the "fixed" approach.

In the current study, the focus is directed towards comprehensively examining the impact of meta-cognitive scaffolding within the dynamic realm of online learning environments, particularly leveraging the capabilities of a platform named APLUS. Central to this investigation is the employment of SimStudent, a highly sophisticated teachable agent that operates as a computational model of learning. This agent incorporates a diverse array of AI techniques, including programming by demonstration. Notably, SimStudent adeptly acquires cognitive skills in the form of production rules, displaying its remarkable capacity for generalizing from both positive and negative examples. This nuanced approach promises to shed further light on the intricate dynamics of learning through teaching, offering valuable insights into optimizing educational experiences.

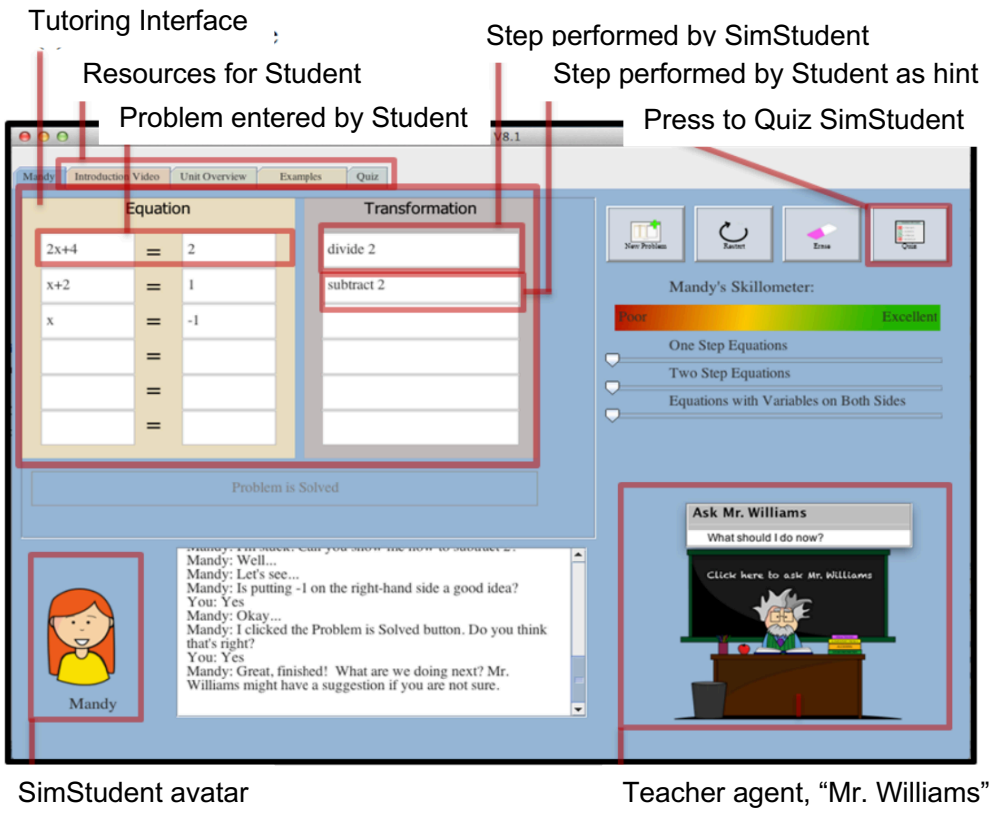

Figure 7: screenshot of APLUS [34]

The meta-tutor version of APLUS offers valuable support in two distinct areas of meta-cognition:

- Quiz assistance, which guides students on when and why they should quiz their SimStudent.
- Problem-solving guidance, which advises students on which problem they should tutor next and provides an explanation for this choice.

It's important to note that the meta-tutor's assistance becomes accessible to students once they have completed a problem or finished a quiz. This design is based on the hypothesis that furnishing students with structured guidance on problem selection and helping them assess their tutee's proficiency can significantly enhance their learning experience as tutors. To empirically assess this hypothesis, the current study employs the online learning platform APLUS to investigate the impact of meta-cognitive support in the context of learning by teaching. The outcomes of this study are expected to shed light on the efficacy of meta-cognitive assistance in tutor learning, thus contributing valuable insights for the future design of learning by teaching environments [34].

3.5 Emma

Van der Meij and colleagues delve into a comprehensive exploration of the design and evaluation process of a motivational animated pedagogical agent (APA) within an inquiry-based learning environment focused on the subject of kinematics. Their objective is to scrutinize the nuanced aspects of the APA's design, paying particular attention to its visual attributes and its interactive dynamics with users. Additionally, the study seeks to gauge the APA's efficacy in heightening students' perceptions of task relevance and self-efficacy in the learning process.

In this inquiry-based learning setting, the chosen APA persona embodies a female character characterized by youthfulness, attractiveness, and an air of 'coolness'. This deliberate choice is aimed at resonating with the typically underrepresented female demographic in science classrooms, creating an environment where they feel more engaged and empowered. The visual representation of the APA is meticulously crafted to align with the age profile of the participants, ensuring relatability, while also exuding an aura of youthful appeal. Furthermore, the design decision to focus solely on presenting Emma's face and a fraction of her upper body aligns with established recommendations, ultimately contributing to an aesthetically pleasing visual experience.

To augment the realism and authenticity of interactions, the APA's movements and facial expressions are intricately programmed within the Elckerlyc software. This level of detail extends to encompass a neutral demeanor in facial expressions, eye-blinks,

and general movements during message delivery. These conscious design choices collectively contribute to a compelling and engaging learning experience, where the APA serves as a relatable peer learner, effectively bridging the gap between technology and student engagement.

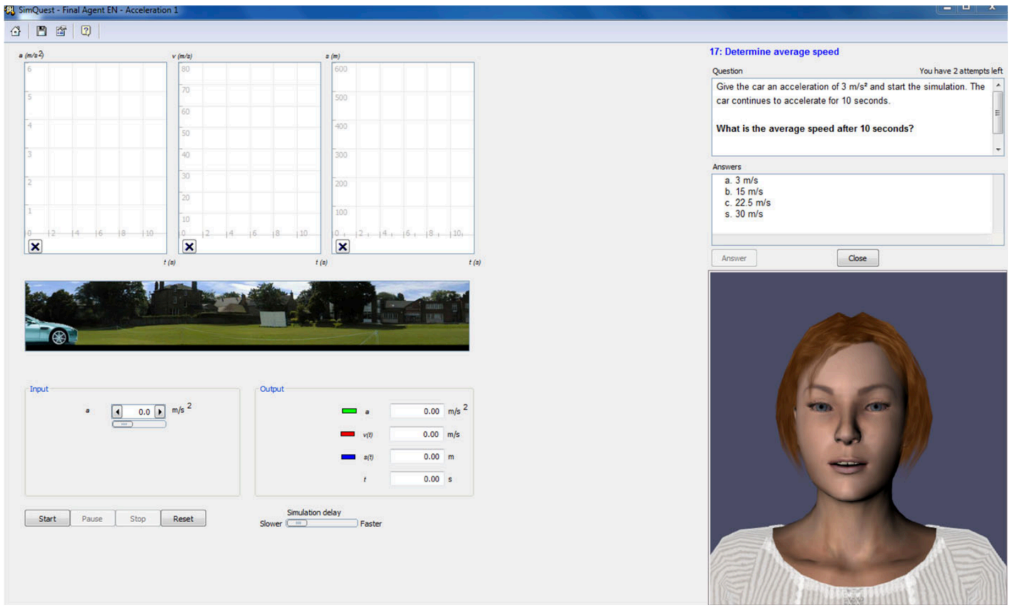

Figure 8: screenshot of the system with Emma [35]

The APA (Animated Pedagogical Agent) engaged with the user through a range of interactive methods. Emma, the APA, was designed to deliver an audible message with precision-timed intervals. For instance, she would provide feedback two seconds after a student accessed an assignment, and one second after the student submitted an answer. These messages were meticulously crafted in alignment with Keller's ARCS model (2010), a framework aimed at enhancing perceptions of task relevance and self-efficacy. Emma's motivational statements, such as ''very annoying'' and ''I am curious,'' were strategically chosen to inspire and engage the students in their learning journey.

Moreover, Emma's responses were tailored based on the student's answers, either acknowledging their correctness or providing feedback in response to the textual input received from SimQuest. The effectiveness of the APA was evaluated through a controlled experiment that encompassed three distinct conditions: agent with both image and voice, voice-only (without an accompanying image), and a control group with neither voice nor image. The central research inquiry sought to understand how students' motivation and knowledge evolved over time within the inquiry-based learning environment, and whether these changes were influenced by condition variations and gender disparities.

The outcomes of the study unveiled compelling insights. Notably, there emerged a noteworthy main effect of time on self-efficacy, with both male and female students experiencing a substantial increase in their self-efficacy beliefs. Furthermore, a significant interaction was observed between time, condition, and gender, especially in relation to self-efficacy. These findings shed light on the nuanced dynamics of motivation and learning outcomes within the context of animated pedagogical agents, offering valuable implications for educational practices [35].

3.6 Azevedo et al

Azevedo et al. conducted a comprehensive study to delve into the efficacy of artificial pedagogical agents (PAs) in bolstering self-regulated learning (SRL) within the context of a hypermedia-based intelligent tutoring system. The report places a specific spotlight on the four distinct PAs integrated into the MetaTutor intelligent tutoring system, each endowed with unique roles and functionalities: Gavin, Pam, Mary, and Sam.

Gavin, in his capacity as the Guide, assumes the pivotal role of providing unwavering support to participants as they navigate through the learning environment. Pam, donning the hat of the Planner, places significant emphasis on the vital aspects of planning, such as activating prior knowledge and crafting pertinent subgoals, thereby enhancing the learning process. Mary, entrusted with the role of the Monitor, assists participants in vigilantly overseeing various metacognitive processes, fostering the ability to make accurate metacognitive judgments during the learning session. Meanwhile, Sam, the Strategizer, takes on the crucial task of championing the adoption of effective cognitive strategies. This encompasses activities like coordinating information sources, drawing inferences, taking notes, and summarizing hypermedia-based science content. All these endeavors are aimed at propelling participants towards the successful attainment of their learning objectives.

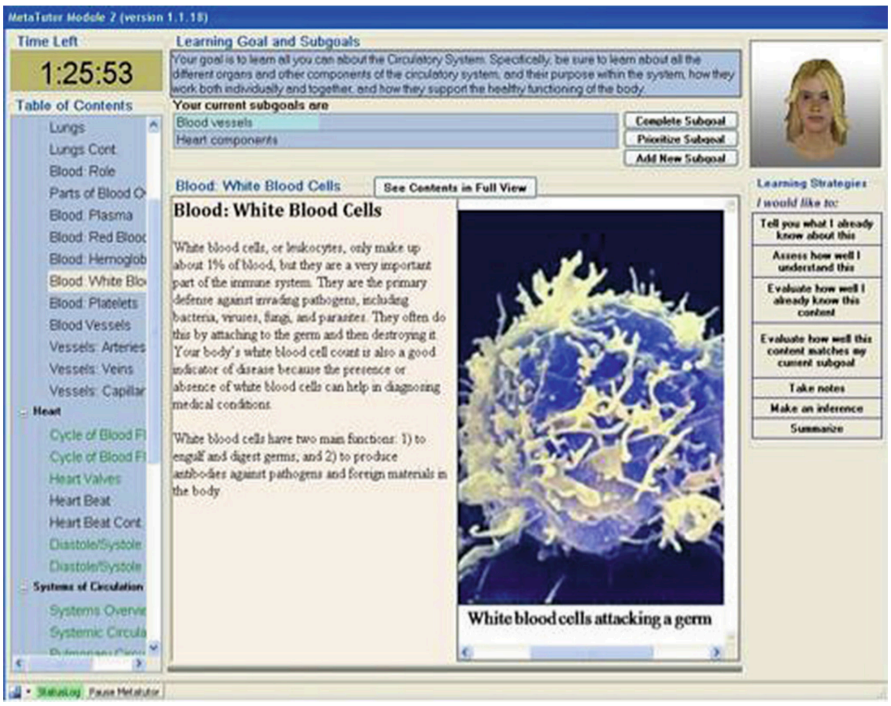

Figure 9: Screenshot of the MetaTutor interface [36]

The design of the MetaTutor interface is meticulously crafted to not only facilitate learning but also to serve as a model for self-regulated learning (SRL). At the core of this interface lies the learning content, comprising text and diagrams, which form the foundational material for achieving both the minor milestones and the broader objective of comprehending the intricacies of the circulatory system. Positioned conveniently on the right side of the interface is the SRL palette, strategically placed to empower participants with a range of SRL strategies.

Adjacent to this palette, in the top right corner of the interface, resides one of the four Pedagogical Agents (PAs). The specific agent displayed is contingent on the session's dynamics, taking into consideration the type of scaffolding being provided by the system and the nature of the instruction being dispensed. These PAs function as external regulators, delivering timely scaffolding and constructive feedback.

Throughout the learning process, the PAs extend essential scaffolding to each participant, especially during pivotal SRL processes like content evaluations (CE), feelings of knowing (FOK), judgments of learning (JOL), and monitoring progress toward goals (MPTG). Furthermore, they offer insightful feedback, aiding participants in assessing their performance and enabling them to maintain a vigilant eye on their progress and growth. This multifaceted approach ensures that participants not only learn about the circulatory system but also develop invaluable skills in self-regulated learning [36].

3.7 Kim et al

Kim et al. conducted a comprehensive study on an innovative anxiety treatment program centered around an embodied agent. This program was thoughtfully designed to specifically target and alleviate mathematics-related anxiety among 9th-grade students. The instructional content was delivered through a series of four online, self-paced modules, each thoughtfully crafted to span an entire class hour. These modules focused on fundamental algebraic concepts, covering a diverse range of topics such as signed number arithmetic, combining like terms and distribution, factoring, and the graphical representation of linear equations through the use of slope and y-intercept. The embodiment of the agent was meticulously designed to emulate the persona of a proficient female educator, ensuring that complex concepts were elucidated with precision and clarity. This multifaceted approach not only addresses anxiety but also enhances the learning experience by providing tailored, interactive instruction.

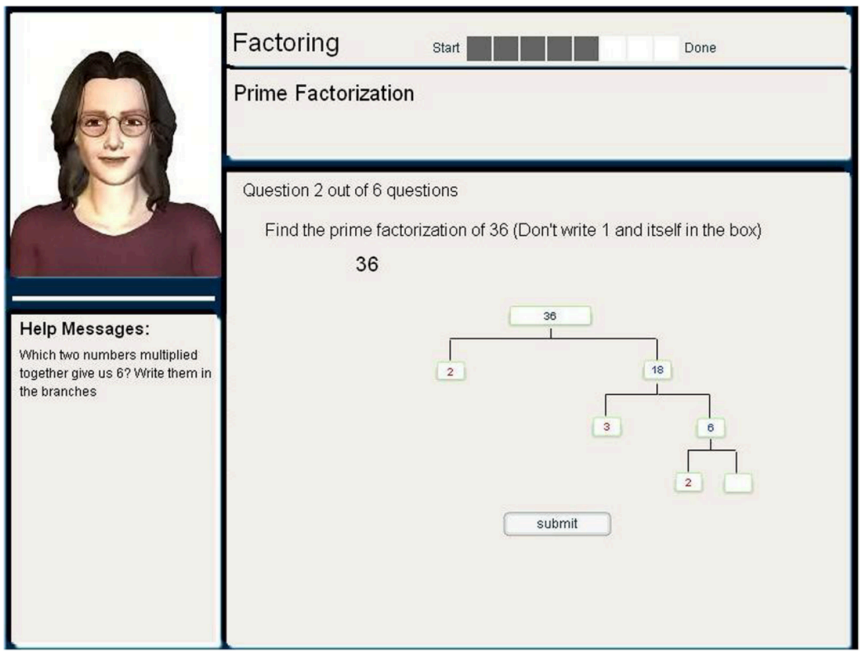

Figure 10: A screen of an example of an agent-based lesson [37]

The pedagogical agent was purposefully crafted as a female teacher, complete with a recorded human teacher voice to ensure a relatable and engaging learning experience. This virtual instructor was equipped with a dual instructional approach, strategically tailored to support apprehensive students. The first facet comprised meticulously structured instruction, adeptly guiding learners through a hierarchical learning trajectory, starting from fundamental concepts and gradually advancing towards more complex material. Moreover, the agent seamlessly administered immediate,

constructive feedback, fostering a dynamic and responsive learning environment. In addition to these facets, the agent included anxiety-relief messages thoughtfully designed to instill self-awareness regarding math-related anxiety, empower students to navigate uncertainty, and embolden them to tackle challenges with confidence.

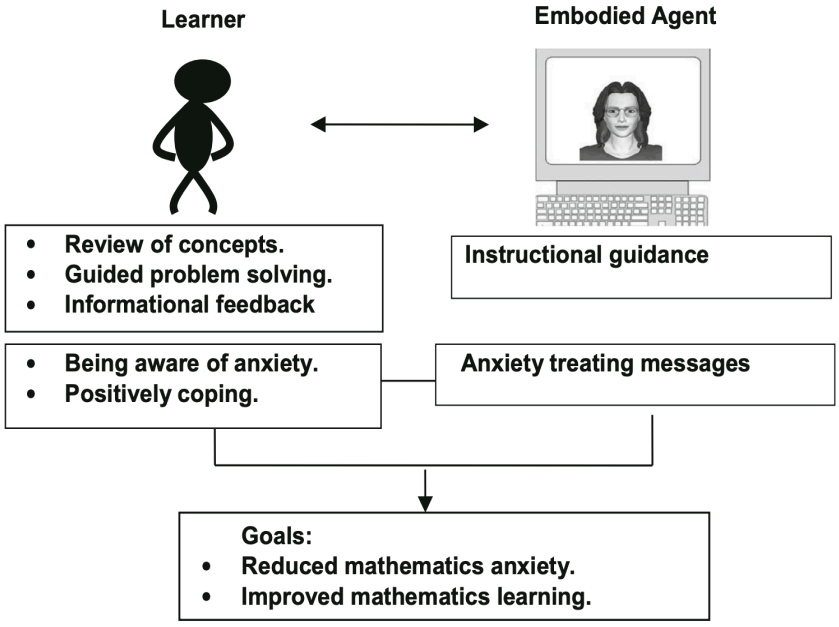

Figure 11: Agent/learner interaction diagram [37]

The virtual agent was meticulously crafted to serve as an animated, computer-based learning companion, dedicated to aiding and engaging learners. This digital character was thoughtfully designed to exude a strikingly human-like appearance, complete with intricately detailed facial expressions and fluid movements, fostering a relatable and immersive learning experience. Through seamless conversational interactions, the agent skillfully provided constructive feedback on students' progress and offered invaluable instructional guidance, ensuring a tailored and supportive learning journey. Notably, the agent's repertoire extended to delivering messages aimed at alleviating mathematics-related anxiety, emphasizing the importance of self-awareness, resilience in the face of uncertainty, and the ability to confront academic challenges head-on.

Upon implementation in the classroom setting, the agent's impact was resoundingly positive. A discernible reduction in mathematics anxiety coupled with a notable upswing in mathematics proficiency among students underscored the effectiveness of this innovative approach. Interestingly, it was observed that the presence of the agent's motivational messages held particular significance for high-anxiety students, signifying the agent's potential to offer targeted emotional support, particularly for learners with unique educational needs. This outcome highlights the promising role that embodied

agents can play in creating an inclusive and supportive learning environment, with the potential to make a transformative difference in the educational experiences of students [37].

Agent	Design				Independent Variable		
	Text	Voice	2D	3D	Overlay	Appearance	Role
AutoTutor [32]	x	x	-	x	-	Emotional Gender Facial Expression	Motivator Assistance Expert/Mentor
Guru [32]	x	-	-	x	-	Emotional Gender Facial Expression	Motivator
JEPPY [33]	x	-	-	x	x	Emotional Gender Facial Expression	Motivator Assistance Hint/Feedback/Prompt
Matsuda et al. [34]	x	-	x	-	-	-	Motivator Assistance
Duffy and Azevedo [38]	x	-	-	x	-	-	Motivator Hint/Feedback/Prompt
Shiban et al. [39]	x	-	-	x	-	Presence Age Gesture	Hint/Feedback/Prompt
Sjödén and Gulz [40]	x	-	x	-	-	-	Assistance Hint/Feedback/Prompt
Van der Meij et al. [35]		x		x	-	-	Motivator
Yung and Pass [41]	x	-	x	-	-	-	Hint/Feedback/Prompt
Azevedo et al. [36]	x	-	-	x	-	Emotional	Instructor/Guidance Self-Report Measures Hint/Feedback/Prompt
Lalle et al. [42]	x	-	-	x	-	Emotional	Hint/Feedback/Prompt
Terzidou et al. [43]	x	-	-	x	-	Instructor/Guidance Assistant Motivator	-
Kim et al. [37]		x	x	-	-	Instructor/Guidance	Instructor/Guidance Hint/Feedback/Prompt
Liew, Tan et al. [44]	X	X		x	-	-	-

Thompson et al. [45]	-	X	-	x	-	Emotional Gender	Hint/Feedback/Prompt

Table 1: Comparisons of the related works

These virtual tutoring systems offer a wide range of features and roles to enhance learning. AutoTutor employs text, voice, and 3D graphics, acting as motivator, assistant, and mentor. Guru focuses on motivation. JEPPY supports text, 3D, and overlays, serving as motivator and assistant. Matsuda et al. use text and 2D graphics as motivator and assistant. Duffy and Azevedo prioritize text and 3D graphics with motivation in mind. Shiban et al. integrate presence, age, and gesture, providing hints and feedback. Sjödén and Gulz assist through text, 2D graphics, and voice interactions. Van der Meij et al. excel as motivators with text, 3D, and voice support. Yung and Pass offer text and 2D graphics with a focus on hints and feedback. Azevedo et al. introduce emotional elements, acting as instructors, self-report measures, and offering hints and feedback. Lalle et al. mirror these attributes, emphasizing hints and feedback. Terzidou et al. excel with text, 3D, and roles as instructors, assistants, and motivators. Kim et al. emphasize text, 2D, and voice, primarily acting as instructors with hints and feedback. Liew, Tan et al. provide text and voice support with 3D graphics for a well-rounded educational experience. Finally, Thompson et al. prioritize voice and 3D graphics, incorporating emotions for effective hints and feedback, showcasing a diverse landscape of virtual tutoring systems.

4 Methodology

For the development or implementation of any project or task, there are always many ways or paths to follow, in this chapter, we look at the systematic approach taken to implement our virtual tutoring system.

4.1 Research Design and Approach

The research underscores the substantial influence of educational agents' appearance within virtual learning environments. It accentuates that visual attributes such as gender, ethnicity, and attire hold a pivotal sway over students' preferences when selecting an agent [46]. This highlights the profound implications of these design choices on learners, signifying the importance of thoughtful consideration in agent development.

Participant Count	Ethnic Background of Participants	Gender of Participants	Prefer Same Ethnicity	Prefer Same Gender	Prefer Same Ethnicity and Gender	No Image
1	ASA	F	0	0	0	0
2	AFA	F	1	1	1	0
9	H	F	6	3	2	0
3	H	M	1	2	0	0
4	NA	F	3	3	3	0
1	NA	M	0	1	0	0
13	NA	F	3	5	1	0
5	O	M	0	1	0	0
2	O	F	-	0	-	0
1		M	-	1	-	0
Total: 40			14	17	7	0

Table 2: Preferred Types of Pedagogical Agents Selected by Participants [46]

Table 2 provide insights into the types of appearances preferred by participants and survey results comparing Peer vs. Teacher agents [47]. It highlights that when students interact with educational agents, their initial perceptions are shaped by visual features

such as gender, ethnicity, and facial expressions [48]. These traits activate preconceptions about the agent's effectiveness, trustworthiness, and intellect.

Additionally, the passage discusses the role of animated pedagogical agents, which are lifelike avatars capable of emotive reactions and effective pedagogy [49]. It notes that agents displaying concern for students' development can inspire similar concern in students, and those attuned to students' emotions can maintain engagement. Furthermore, agents with engaging characters make learning more captivating.

A study conducted by candidates from the National Taipei University, involving a survey with 284 students [50]. This research delves into how the visual appeal of an agent affects the learning process, investigating how students' perceptions of trustworthiness and expertise are influenced by the appearance of the agent. In the study, six agents with varying levels of outlook (demure, seductive, and overtly sensual) were used. These agents were of the same gender and differentiated only by their clothing and accessories. The study also involved using prerecorded voices that matched the gender of the agents. These agents were tasked with explaining the harmful effects of smoking using a computerized video. Various measures were considered, including trustworthiness, attractiveness, expertise, intention, and attitude.

The results of the survey showed interesting gender-based perceptions. Male students perceived overtly sensual female agents as more attractive, trustworthy, and expert. However, female students viewed overtly sensual agents as less trustworthy compared to demure agents with similar expertise. For male agents, both male and female students found demure agents to be more trustworthy and attractive, while they considered seductive agents to be more expert. This suggests that an agent's appearance significantly influences the learning process.

A separate research endeavor [51] delved into how the visual appearance of a pedagogical agent influences learner motivation and results. This study conducted a comparison between groups that interacted with a pedagogical agent and those who relied on on-screen text for learning. This investigation underscored the critical role of elements such as attractiveness, credibility, and expertise in the creation of impactful avatars.

Drawing from the comprehensive body of research and established theories, our selection of criteria for the envisioned agent is firmly grounded. By opting for a Human agent within the age range of 20 to 30 years, we strive to create a persona that learners can readily relate to and find approachable. Extensive research underscores the pivotal role visual representation plays in shaping learner engagement and their perception of the agent's competence. This aligns seamlessly with established

principles in the realm of technology-enhanced learning. Additionally, incorporating both male and female agents acknowledges the importance of diverse representation, ensuring inclusivity in the learning experience. Selecting Caucasian male and female agents recognizes the visual cues that participants in studies have shown to influence initial perceptions. These choices are pivotal in shaping learners' interactions and establishing trust. As a tutoring agent with advanced features like facial expressions and gestures, the proposed agent leverages the effectiveness of animated pedagogical agents, enhancing the potential for emotive reactions and impactful pedagogy.

4.2 System Architecture Overview

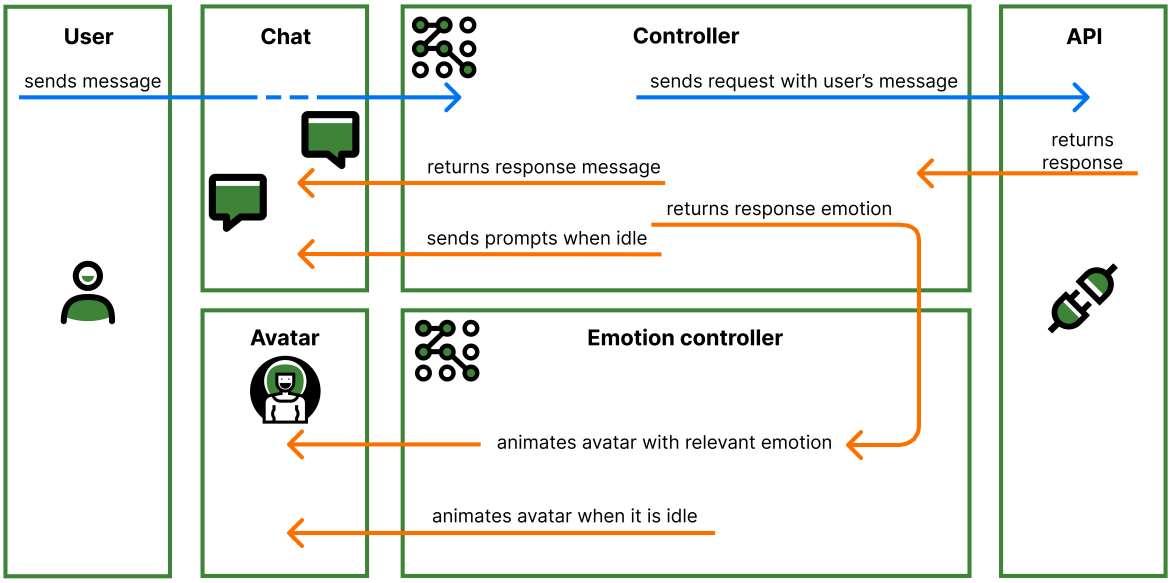

Figure 12: System architecture

The system architecture for this thesis compromises of six (6) components

4.2.1 *User*

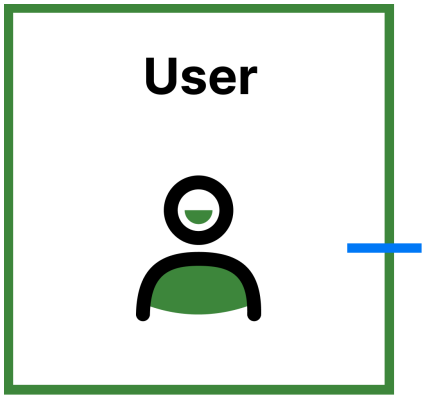

Figure 13: User

The successful deployment and utilization of the tutoring agent within the existing virtual tutoring system hinges on a clear understanding of the intended user base. The users and stakeholders who interact with and benefit from the tutoring agent are as follows:

Students: Students are the central beneficiaries of the tutoring agent. They seek personalized guidance and support, making the agent pivotal in enhancing their learning experiences. Understanding their diverse learning needs and preferences is crucial for tailoring the agent's interactions effectively.

Instructors and Tutors: Instructors and tutors rely on the tutoring agent to facilitate their teaching efforts. They use the agent to gain insights into student performance and adjust their teaching strategies accordingly. The agent acts as a valuable tool for continuous improvement in the teaching process.

Administrators and Developers: Administrators oversee the virtual tutoring system, while developers are responsible for its technical maintenance and enhancements. Administrators use the agent's data to make decisions about system improvements and scalability. Developers rely on user feedback to ensure that the agent remains effective and up-to-date.

Researchers: Researchers in education and artificial intelligence utilize the tutoring agent as a subject of study or as a data source for research. By analyzing the agent's interactions and outcomes, they contribute valuable insights to the field, advancing our understanding of intelligent tutoring systems and learning technologies.

Understanding the diverse user base is pivotal in tailoring the tutoring agent's features, interactions, and performance to meet the specific needs and expectations of each category of user. Throughout the development and evaluation process, we consider

the perspectives and requirements of these stakeholders to ensure that the tutoring agent becomes a valuable asset in the virtual tutoring ecosystem. The user interacts directly with the system via the chat.

4.2.2 *Chat*

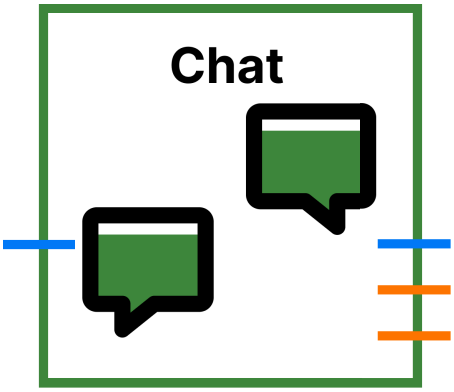

Figure 14: Chat

The system relies on text-based communication as the primary mode of interaction. Users input their questions, comments, or requests using natural language, making the interaction process familiar and accessible. What sets this apart is the visual alignment employed in the chat interface. User-generated queries are positioned to the left of the conversation window, while system-generated responses are aligned to the right. This spatial separation instantly conveys the conversation's structure, making it easy for users to distinguish between their input and the system's output. The chat sends the user input to the controller and displays the controller response to the user.

4.2.3 *Controller*

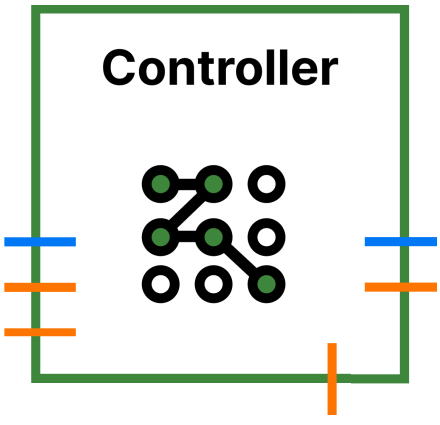

Figure 15: Controller

The controller is the central component that serves as the brains behind the virtual tutoring system. It functions as the command center, orchestrating various critical

aspects of the system's operation. This component plays a pivotal role in ensuring a seamless and interactive user experience. Here are the key responsibilities and functions of the controller:

System Management: The controller is responsible for the fundamental operations of the virtual tutoring system. It handles the opening and minimizing of the system, allowing users to start and pause their tutoring sessions effortlessly. This management function ensures that users have control over their interaction with the system, enhancing usability.

Avatar Selection: One of the user's preferences in the virtual tutoring system is selecting their preferred avatar. The controller facilitates this choice, providing users with options and seamlessly implementing their selections. This personalized avatar choice contributes to a more engaging and relatable learning experience.

Animation Preloading: To create a fluid and dynamic interaction, avatars within the system often employ animations to convey emotions and expressions. The controller is responsible for preloading these animations, ensuring that they are readily available for use during the conversation. This optimization minimizes lag and enhances the overall user experience.

Chat Conversation Management: Effective communication is at the heart of any virtual tutoring system. The controller manages the chat conversation, handling the flow of messages between the user and the tutor agent. It maintains the context of the conversation, processes user queries, and facilitates the delivery of relevant responses from the tutor agent.

API Querying: The controller component efficiently processes the API response, serving as the intermediary for data distribution to two distinct subsystems within the virtual tutoring system: the chat interface and the emotion controller.

Upon receiving the API response, the controller accurately dissects the data, extracting the necessary information for each specific subsystem. For the chat interface, it provides relevant data to construct user-friendly and contextually appropriate responses. Simultaneously, it directs the emotion-related data to the emotion controller, which is crucial for animating the tutor avatar with accurate emotional responses.

4.2.4 *API (Application Programming Interface)*

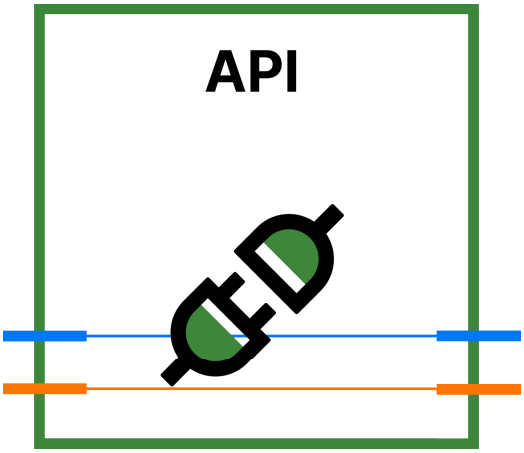

Figure 16: API

The API serves as the essential bridge connecting the backend and the frontend of the tutorial agent, enabling seamless communication and data exchange between these two crucial components. The API facilitates the efficient transfer of data and information between the backend and frontend of the tutorial agent. This encompasses various data types, such as user preferences, instructional content, progress tracking, and real-time updates. By serving as an intermediary, the API ensures that both sides can exchange information effectively. On the backend side, the API establishes connections with the backend services to retrieve and provide data. This integration ensures that the system can offer comprehensive and up-to-date educational materials. On the frontend, the API allows for the presentation and utilization of data received from the backend. This includes rendering dynamic content, responding to user inputs, and updating the user interface in real time. Users can interact with the system, ask questions, and receive responses—all facilitated by the API's ability to relay information between components. The API's architecture is designed to be extensible, accommodating future enhancements and integrations. As the virtual tutoring system evolves, new features or external services can be seamlessly integrated by expanding the API's capabilities. This extensibility ensures that the system can adapt to changing educational requirements and technological advancements.

4.2.5 *Emotion Controller*

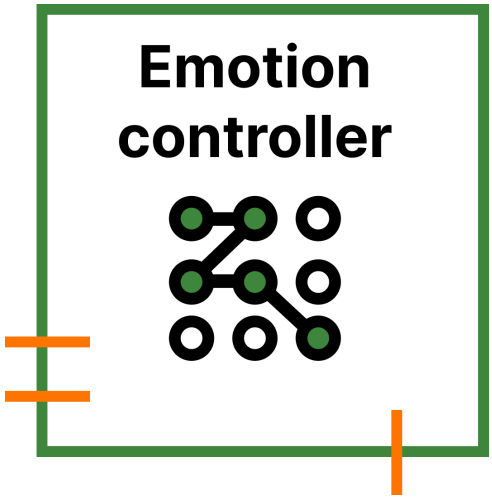

Figure 17: Emotion Controller

The emotion controller, functioning as a pivotal component within the virtual tutoring system's architecture, takes receipt of the API response conveyed by the controller. Subsequently, it engages in a nuanced decision-making process by evaluating the content and context of this response. Within its repository, an extensive library of preloaded animations stands at the ready, each crafted to convey a spectrum of emotional states and expressions.

This decision-making process undertaken by the emotion controller is imbued with technical sophistication. It involves a analysis of various factors within the API response, such as the tone, sentiment, and semantic content of the message, to discern the most apt emotional response. Leveraging advanced algorithms and data-driven modeling, the emotion controller intelligently identifies which animation from its repository aligns most harmoniously with the semantic nuances embedded within the chat response generated by the controller.

Once the selection process concludes, the emotion controller seamlessly integrates the chosen animation into the tutor avatar's repertoire, ensuring that the emotional response is both contextually appropriate and visually engaging. Through this intricate interplay of data analysis and animation management, the emotion controller enriches the user experience by bestowing the tutor avatar with a nuanced and emotionally resonant response, thereby enhancing the overall interaction within the virtual tutoring system.

4.2.6 *Avatar*

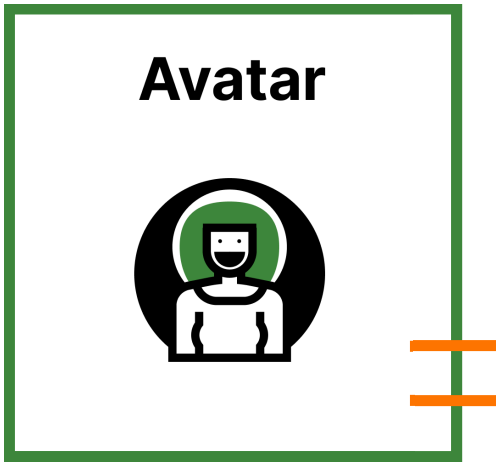

Figure 18: Avatar

Avatar Component

The avatar component represents the visual embodiment of the virtual tutor within the system. It serves as the primary interface through which users engage in the educational content and interact with the virtual tutor. This component encompasses various facets, each contributing to a rich and immersive user experience.

1. **Visual Representation**: The avatar is designed to offer a visually compelling representation of the virtual tutor. It incorporates graphical elements, animations, and visual cues to convey expressions, gestures, and emotions, emulating a lifelike interaction. The avatar component aims to foster a sense of relatability and engagement for users.

2. **Emotional Expression**: One of the key features of the avatar is its ability to convey a range of emotions. Through a carefully curated library of animations and expressions, the avatar component dynamically responds to the user's interactions and the content of the conversation. This emotional responsiveness adds depth and nuance to the user's experience, enhancing the sense of connection and rapport with the virtual tutor.

3. **Customization Options**: The avatar component is customizable vis the controller, to choose from the available avatars. This customization features empower users to tailor their interaction with the virtual tutor to suit their preferences and learning style.

4. **Interaction Capabilities**: Beyond its visual representation, the avatar component is equipped with interactive capabilities. This include gestures, facial expressions, and animations that respond dynamically to user input. These

interactive elements serve to enhance the sense of engagement and interactivity, creating a more dynamic and immersive learning experience.

5. **Performance Optimization**: The avatar component is engineered to be resource-efficient, ensuring smooth and responsive performance across a range of devices and network conditions. This optimization is crucial in providing a seamless user experience, particularly for users with varying levels of technological infrastructure.

4.3 Agent Design and Integration

Agent design and integration describes the architectural blueprint and integration process of the tutoring agent within our virtual tutoring system. The agent serves as the dynamic core, facilitating interactions and delivering educational content. Here, we discuss the intricate design decisions, ranging from the underlying logic to the seamless amalgamation of the agent with the broader system. This section lays the groundwork for a comprehensive understanding of how the agent contributes to the system's overall functionality and effectiveness in providing a rich learning experience.

1 Design

The virtual tutoring system has been designed with two distinct states: open and closed. Each state serves a specific purpose, optimizing the user experience and screen real estate.

Closed State: In this mode, the system conserves space by minimizing itself to a floating icon positioned at the lower right corner of the screen. The icon features a transparent background, ensuring it remains unobtrusive to the user. A simple click on this icon transitions the system to the open state.

Open State: When in the open state, the system maximizes its functionality for user interaction. The layout is thoughtfully organized to enhance usability.

- **Chat Interface**: The left side of the screen is dedicated to the chat interface. Here, users can engage in conversations with the virtual tutor. A textbox is positioned at the bottom left, allowing users to input messages, while the send message button is conveniently located to the right of the textbox.

- **Conversation Display**: Positioned at the top of the textbox and send button is the conversation display. This section provides a clear view of previous messages and their corresponding responses, aiding in the continuity and coherence of the conversation.

- **Avatar Selection**: At the top right corner, two buttons are situated. The button on the right enables students to select a different avatar, adding a personalized touch to their learning experience. On the left, a button allows users to switch the system back to the closed state when needed.

- **Avatar Window**: The avatar window is strategically placed on the right side of the system layout. It has been designed to remain discreet, situated behind all other interface elements. This placement maximizes the available space for interaction while ensuring that the avatar remains prominently visible without obstructing any interactive elements.

This thoughtfully structured design aims to provide a seamless and user-friendly experience, allowing users to effortlessly transition between the open and closed states while making efficient use of screen real estate.

2 Integration

To seamlessly integrate the virtual tutoring system into any webpage, an iframe element is employed. This iframe serves as a container that encapsulates the system's functionality within the desired page. The iframe which is implemented in a line of code is to be put with the body tag of the desired page.

The “src” property of the iframe is configured to link directly to the index file of the system. This establishes a direct connection between the webpage and the core functionality of the virtual tutoring system, ensuring a cohesive user experience.

Additionally, specific styles are applied to the iframe to guarantee the system's proper display within the webpage. These styles have been tailored to harmonize the system's appearance with the overall aesthetics of the page, providing a visually unified experience for users.

.

4.4 Agent Requirements and Specifications

The agent, a critical component of our virtual tutoring system, has been designed with specific requirements and technical specifications to guarantee optimal performance and compatibility. Firstly, the system is crafted to be fully compatible with all modern web browsers, ensuring a seamless user experience regardless of the chosen browsing platform.

In terms of project size, the entirety of the project, which encompasses previous animation versions, totals approximately 400 megabytes, which is required. This comprehensive size accounts for all essential elements that contribute to the system's

functionality and visual elements, providing a comprehensive and feature-rich learning experience.

For the Python backend, specific server specifications are imperative for ensuring optimal performance. This includes compatibility with modern operating systems such as Windows 7 or 10, and Mac OS X 10.11 or higher. Furthermore, the backend necessitates a 64-bit architecture, with support for Linux in the form of RHEL 6/7, 64-bit (with compatibility in Ubuntu). Additionally, an x86 64-bit CPU (Intel / AMD architecture), a minimum of 4 GB of RAM, and at least 5 GB of free disk space are fundamental prerequisites to facilitate seamless operation.

5 Implementation

In this chapter, we transition from conceptualization to realization, bringing the virtual tutoring system to life through implementation. Here, we delve into the technical part and coding methodologies employed to transform ideas into functional components. This chapter serves as the blueprint for turning theory into practice, demonstrating how each facet of the system is translated into tangible, interactive elements. Through a comprehensive exploration of the implementation process, we unveil the inner workings of our virtual tutoring platform, showcasing the fusion of innovative design with robust functionality.

5.1 Development Tools and Technologies

Here we delve into the arsenal of tools and technologies harnessed in the creation of our virtual tutoring system. From programming languages to integrated development environments, each component played a crucial role in shaping the system's functionality and user experience. Here, we provide a concise overview of the key tools and technologies employed, highlighting their significance in realizing our project's objectives.

1. **Visual Studio Code:** Visual Studio Code, an open-source code editor created by Microsoft, offers a lightweight yet powerful platform for developers. It boasts support for a wide array of programming languages and comes packed with essential features such as syntax highlighting, debugging capabilities, and efficient version control. In the development process of our virtual tutoring system, Visual Studio Code played a central role as the primary code editor. Its user-friendly interface and high degree of customization made it an ideal environment for crafting and refining code in HTML, CSS, JavaScript, and Python. Moreover, its seamless integration with popular version control systems like GitHub greatly facilitated collaborative efforts among the development team. This ensured a smooth and effective workflow throughout the project.

2. **Blender 3D:** Blender 3D is a powerful, open-source 3D creation suite that encompasses modeling, animation, rendering, and more. It is widely used for creating 3D models and animations. Blender 3D played a crucial role in the creation and customization of the avatar component. It was used to model and animate the virtual tutor, allowing for the integration of expressive gestures and animations that enhanced the user experience during interactions.

3. Imagemagick: ImageMagick is a command-line tool and software suite used for image editing, manipulation, and conversion. It supports a wide range of image formats and provides extensive functionality. ImageMagick was employed to create sprites which is used for the animation of the avatar.

4. HTML (Hypertext Markup Language): HTML stands for HyperText Markup Language, and it's the fundamental language used to build web pages. It provides the essential structure and elements necessary to define what appears on a web page. This includes things like headings, paragraphs, lists, links, images, and more. In the context of this thesis, HTML played a crucial role as the foundational language for organizing the user interface of the virtual tutoring system. It was used to craft the layout of the system, specify different elements, and set up the overall visual arrangement. HTML ensured that the content was arranged in a way that makes sense both logically and semantically.

5. CSS: CSS, or Cascading Style Sheets, serves as a powerful language for dictating how web pages are presented and organized. It grants designers and developers the capability to imbue HTML elements with styles, encompassing attributes like colors, fonts, spacing, and positioning. In the development of the virtual tutoring system, CSS emerged as a pivotal tool in crafting its visual identity. By leveraging CSS, we could bestow a cohesive and aesthetically pleasing interface upon the HTML structure. This encompassed defining crucial elements like color palettes, font selections, layouts, and other stylistic components, culminating in a seamless and immersive user experience. Furthermore, CSS was instrumental in specifying the animation keyframes for the avatars, ensuring their motions were rendered smoothly and engagingly. Additionally, CSS played a role in preloading animations, further enhancing the system's performance and interactivity.

6. JavaScript: JavaScript, recognized as a flexible and advanced programming language, is chiefly acknowledged for its proficiency in introducing interactivity and lively functionalities to web pages. This powerful language wields the capability to govern how a webpage behaves, react to user inputs, and alter the Document Object Model (DOM). Within the framework of the virtual tutoring system, JavaScript played a pivotal role in elevating user interaction. It facilitated immediate responses to user actions, like submitting inquiries, promptly delivering feedback, and dynamically refreshing content sans the need for a complete page reload. Moreover, JavaScript took charge of overseeing user sessions, managing API requests, thereby ensuring a

seamless and captivating user engagement. This multifaceted language was integral in fostering an interactive and dynamic learning environment.

7. Figma: Figma is a powerful collaborative design and prototyping tool that enables teams to work together on creating, sharing, and refining design mockups and prototypes in real-time. It played a pivotal role in crafting design mockups and wireframes for the user interface. By using Figma, we were able to visualize and refine the layout of the system, making iterative design enhancements based on feedback and specific project requirements.

8. AJAX (Asynchronous JavaScript and XML): AJAX, short for Asynchronous JavaScript and XML, is a powerful technique employed in web development. It enables the sending and retrieval of data from a server without the need to refresh the entire web page. In the context of our virtual tutoring system, AJAX played a pivotal role in establishing real-time communication between the system and the backend server. This resulted in smooth and seamless data exchange, enabling dynamic updates without requiring the page to reload. This enhancement significantly elevated the overall user experience, providing a more interactive and engaging platform for learners.

9. Python: Python stands out as a versatile, high-level programming language recognized for its clear and user-friendly syntax. Its flexibility allows for the adoption of various programming approaches, making it a popular choice across diverse applications. In our project, Python took on a central role in the backend development. It was responsible for orchestrating server-side operations, processing data, and seamlessly integrating with external services. Python's robust capabilities were instrumental in efficiently managing data flow, skillfully handling API requests, and executing vital tasks crucial to the system's overall functionality.

10. GitHub: GitHub is an online platform designed for version control and collaborative software development. It offers a suite of tools that streamline code management, foster collaboration, and facilitate project tracking. In our project, GitHub played a pivotal role as the chosen version control system. It enabled seamless teamwork, allowing team members to collaboratively develop code. This platform not only ensured efficient code management and accurate version tracking but also facilitated the smooth integration of code modifications into the central project repository.

5.2 Agent implementation details

The creation of the tutoring agent was carried out in different phases: modeling, animation, and programming. Each phase played a crucial role in bringing the agent to life within our virtual tutoring system. This section provides a detailed account of how these phases were carefully orchestrated to craft an interactive and responsive agent that enhances the learning experience. Let's delve into the specifics of each phase to gain a comprehensive understanding of the agent's implementation.

5.2.1 Modeling

The modeling phase, executed using the powerful Blender 3D software, constitutes a pivotal stage in the implementation of our tutoring agent. Here, intricate attention was dedicated to the creation of two distinct avatars, one male and one female, with a particular emphasis on the upper body and facial features.

Female 3D Avatar Modeling: Creating the female 3D avatar in Blender 3D involved a detailed process focused on capturing the nuances of the upper body and facial features. First, the foundational structure was sculpted, paying careful attention to anatomical proportions. Emphasis was placed on crafting a lifelike representation, with defined collarbones, shoulders, and torso. The facial features were then sculpted, ensuring a balance between realism and stylization. Particular care was taken in shaping the eyes, nose, lips, and eyebrows to convey a natural and expressive look. Subtle variations in skin texture and tone were added to lend authenticity, while the hair was modeled with intricate strands and layers to achieve a natural flow and volume. The result was a female avatar exuding both grace and personality, ready to engage users in a relatable and interactive manner.

Male 3D Avatar Modeling: Constructing the male 3D avatar in Blender 3D involved a similar process, tailored to convey masculinity and character. The upper body was sculpted with attention to muscular definition and proportions that reflect a male physique. Details like pectoral muscles, shoulder width, and arm structure were carefully shaped to create a robust and dynamic appearance. The facial features were chiseled to exude strength and individuality, with a focus on a well-defined jawline, eyes, and brows. Special care was taken to incorporate facial hair and hairstyles that complemented the overall look, adding an extra layer of realism. Through this process, the male avatar emerged as a commanding and relatable figure, poised to engage users with confidence and authority.

5.2.2 Animation

In the animation phase, the avatars created in Blender 3D are imbued with movement and expression, elevating their interactivity within our virtual tutoring system. Blender 3D, our chosen modeling software, proves instrumental in this process, offering a robust platform for dynamic character animation. Additionally, the sprite creation tool ImageMagick plays a pivotal role in crafting seamless and engaging animations. This subsection provides an in-depth exploration of the techniques and tools employed to animate our avatars, ensuring they communicate with users in a natural and expressive manner. Through a systematic approach to movement, gestures, and facial expressions, the avatars evolve from static models to dynamic, responsive entities, enhancing the overall educational experience.

1 Blender 3D

The animation conducted in Blender 3D is an intricate process, crucial for infusing our avatars with dynamic movements and expressions. This phase is defined by a series of well-defined steps, each contributing to the creation of fluid, natural animations.

a. Rigging and Skeletal Animation:

Rigging is the initial step, involving the creation of an internal skeleton, or armature, within the 3D model. This armature functions as a framework that enables the avatar to move realistically. Bones are strategically placed to correspond with joints and points of movement. They are assigned specific roles, such as controlling limbs, facial features, or body posture.

b. Weight Painting:

Weight painting is a critical aspect of rigging that determines how much influence each bone exerts on the surrounding mesh. This process ensures that when a bone moves, it deforms the mesh naturally. It involves assigning weights to vertices based on their proximity to specific bones.

c. Keyframe Animation:

Keyframe animation is pivotal for defining specific poses or positions of the avatar at different points in time. By setting keyframes at key moments, Blender 3D smoothly interpolates the movements between these frames, resulting in seamless and continuous motion. This technique is applied to all aspects of the avatar, encompassing body movements, facial expressions, and more.

d. Facial Shape Keys:

Facial shape keys, also known as blend shapes or morph targets, enable manipulation of facial features to convey different expressions. This involves creating a series of

predefined facial poses and smoothly transitioning between them. For instance, a smile is achieved by blending the mouth and cheek vertices into an upward curve.

e. **Inverse Kinematics (IK) and Forward Kinematics (FK)**:

Blender 3D offers both IK and FK systems for controlling limb movement. IK provides precise control over the end of a limb (like a hand or foot), while FK controls each individual bone in a limb. Skillful utilization of both techniques allows for a natural range of motion in the avatars' limbs.

f. **Layered Animation**:

Layered animation involves simultaneous manipulation of multiple aspects of the avatar, including body movement, facial expression, and eye tracking. This technique facilitates complex and nuanced interactions, as different layers of animation work together to convey a cohesive performance.

g. **Rendering and Previewing**:

Regular rendering and previewing of frames are essential for assessing the fluidity and coherence of the motion throughout the animation process. This ensures movements appear natural and allows for real-time adjustments or refinements.

h. **Fine-tuning and Iteration**:

The animation phase often necessitates multiple iterations for refining poses, adjusting timing, and perfecting transitions. Feedback and testing play a pivotal role in honing the animation to achieve a high level of realism and expressiveness.

By executing these steps within Blender 3D, the animation phase transforms static models into dynamic, responsive avatars, prepared to engage users in an enriching and immersive learning experience. This process is foundational to the avatar's ability to convey information and interact effectively with users.

2 ImageMagick

Creating sprites in ImageMagick is a crucial step in the animation phase. This process involves generating a sequence of images that represent the avatar's various states and movements. Each image, or frame, contributes to the illusion of motion when played in rapid succession.

To initiate sprite creation, the rendered frames from Blender 3D serve as the source material. These frames are crafted to capture the avatar's different poses and expressions. They are then exported as individual image files, like WebP to preserve image quality and transparency.

Once the source frames are prepared, ImageMagick steps in to compile them into a cohesive sprite sheet. ImageMagick, as a versatile command-line tool, excels at batch processing and manipulation of images. Through a series of commands, it arranges the frames in a grid-like structure, with each frame occupying a distinct cell. This grid format ensures that the frames can be easily accessed and displayed in a synchronized manner during animation playback.

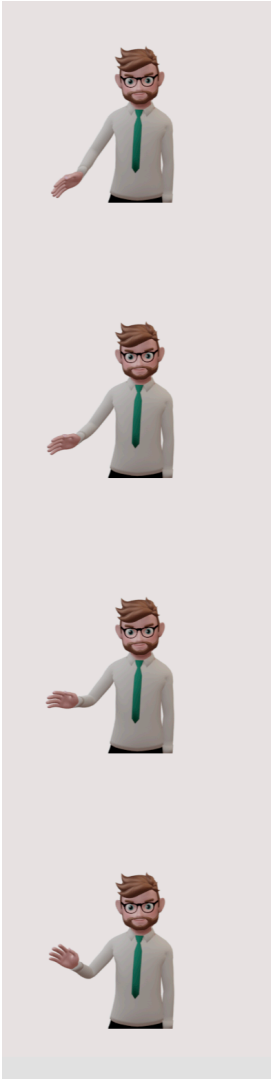

Figure 19: A section from the sprite generated for Obi's bye animation.

Throughout the sprite creation process, attention to detail is paramount. Each frame must align precisely within its designated cell to guarantee seamless animation. Any discrepancies in positioning can result in jittery or disjointed motion during playback.

5.2.3 Development

In this phase, we transition from design to implementation, focusing on the creation of the system design and logic. This critical step involves translating the conceptualized

agents into functional entities within the virtual tutoring system. The development process will be carried out using a combination of programming languages, including Python for backend operations, HTML, CSS, JavaScript for frontend functionality, and AJAX for API calls. Visual Studio Code will serve as our primary integrated development environment (IDE), providing a robust platform for code composition and management.

The code logics to be implemented in Visual Studio Code encompass several key functionalities. These include avatar initialization and behavior control, chat management, and API handling for client and server-side communication.

1. **Design:** This section provides an in-depth exploration of the fundamental design building blocks of our virtual tutoring system - HTML elements. Each element's purpose, structure, and functionality will be dissected. Additionally, we will delve into the corresponding CSS styling, elucidating how it shapes the visual presentation and user experience. By comprehensively examining the synergy between HTML structure and CSS design, we aim to establish a robust foundation for an engaging and intuitive learning platform. This section serves as a detailed guide to the interplay of these crucial web technologies within our virtual tutoring environment.

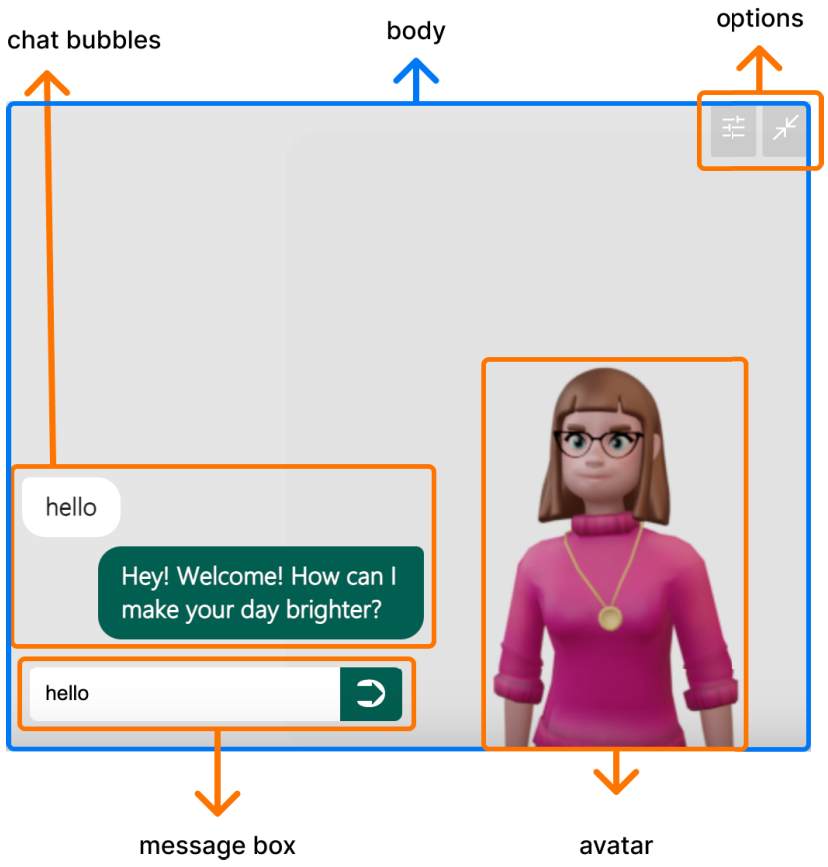

Figure 20: the system design I.

a. **Body:** this is the parent element of the system; every other element is built inside it. Designed to float at the bottom right corner pf the screen in order not to interfere with the user's screen real estate.

Built in an iframe, It's like a small screen within the main screen. This window has some specific settings, like no borders around it and it can float over other things on the webpage. It's also a fixed size, about 520 pixels wide and 420 pixels tall, this size changes based on the system state. When the system is minimized, the size is reduced to 70 pixels wide and 65 pixels tall.it has soft edges and a transparent background.

The body is basically designed like a mini-website inside the main website, and it's designed to be discovered easily and stay in its fixed position.

b. **Chat Bubbles:** this the element serves as the chat interface on a system. It's like the framework for a conversation between a person and the system. There are different layers;

The Chat Scroller: is like a container that holds the chat messages. It helps to manage how the messages appear and allows for scrolling.

The **Chat Bubbles:** is where the actual conversation bubbles go. This is where the messages from the user and the system will be shown. Each message is inside a **chatbubble**. Inside these, there are two types of messages:

- **Agent chat** represents a message from the system.
- **User chat** represents a message from the user.

For each type of message (agent and user), the chat bubble design is different, in color and position. This helps distinguish between messages from the user and the agent.

This is the visual layout for the chat conversation, making it clear who is sending each message and how they're displayed on the webpage.

c. **Message box**: this element holds the chat input field and message send button in a stylized form. Located at the bottom left side of the system layout. Designed formed-like, so users can send their message by hitting the enter key. The user uses the text field to enter the message and the button to send the message.

d. **Avatar**: by default, when the webpage loads, the female agent (Ada) will be displayed on the bottom-right corner, while the agent with the class "obi" will be hidden. That is the location of the element that display the sprites of the avatars.

Inside it, there are two elements, each with a class of "agent". These represent the two individual agents. They have a transparent background, are positioned in the bottom-right corner of the webpage, and have a specific height and width, giving them a distinct shape.

e. Options: This is element that houses the options buttons. It is a menu with two clickable options: one for toggling the options menu and another for hiding or minimizing the system. The menu is positioned at the top-right corner of the webpage and has a visually pleasing design with rounded corners. When clicked, the options trigger specific functions or actions on the webpage.

Figure 21: the system design II.

f. Menu: this element holds the options to select agents (Ada or Obi) as defaults and control their visibility. It also provides buttons for additional menu options and a way to hide the agent. When interacted with, these elements trigger corresponding functions, enabling dynamic behavior on the webpage. Designed to be big to aid the user experience, the left button is for Ada and the right button is for Obi.

2. Logic: This section provides an in-depth exploration of the fundamental building blocks of our virtual tutoring system - HTML elements. Each element's purpose, structure, and functionality will be dissected. Additionally, we will delve into

the corresponding CSS styling, elucidating how it shapes the visual presentation and user experience. By comprehensively examining the synergy between HTML structure and CSS design, we aim to establish a robust foundation for an engaging and intuitive learning platform. This section serves as a detailed guide to the interplay of these crucial web technologies within our virtual tutoring environment.
To carry out any activity in the system a logic is required these activities vary from opening the system to choosing a preferred avatar to chatting with the agent and minimizing the system. To achieve these logics, we have used JavaScript to code series of logic code blocks. The logics and their code blocks are listed and explained below.

a. **onLoad:** this activity happens when the user visits a webpage with the system embedded in it. this code snippet sets up the initial state of the webpage by hiding a specific element, ensures that there is always a default avatar profile selected, and retrieves the selected avatar profile for further use in the application.

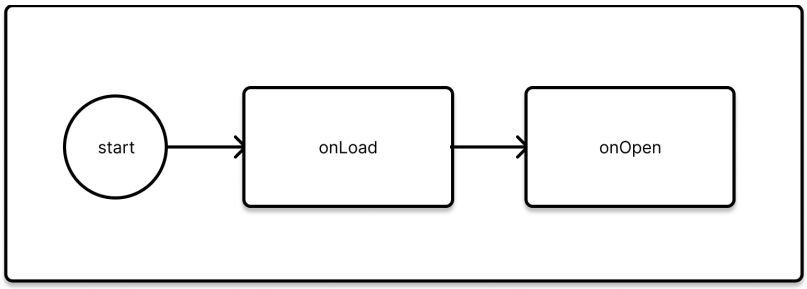

Figure 22: A section from the system logic BPMN diagram I

These are basically code blocks that run to prepare the system for the user. The onload logic has the following functionalities:

Hide Loaded Sprites: When a user accesses the webpage, a crucial step is the loading of sprites required for animations. However, to ensure a seamless and visually appealing user experience, these loaded sprites are cleverly hidden upon the complete loading of the webpage. This strategy guarantees that the animations are not initially visible on the page, preventing any abrupt or disjointed visual elements during the page's loading process. This thoughtful implementation ensures that users are presented with a polished and engaging interface right from the start.

Retrieve Avatar Profile: In the realm of user customization, the system takes a proactive step to enhance user engagement. The underlying logic diligently checks whether the user or client is making their maiden voyage into the system or if they are a returning visitor. In this process, it retrieves the last avatar option stored in the local storage. This retrieved value represents the user's previously selected avatar profile,

ensuring a consistent and personalized experience across visits. By remembering the user's choice, the system fosters a sense of continuity and familiarity, making the user feel right at home within the virtual environment.

Set Default Avatar Profile: To cater to all users, even those who might not explicitly select their avatar profile, the system employs a logical mechanism. It investigates the user's browser local storage under the key "avatarProfile." In cases where it discovers an absence of any value (essentially an empty storage), a default avatar profile is automatically set as Ada. This strategic move guarantees that there's always a default avatar profile in place. Thus, users who may not have made an explicit choice are not left with a generic or incomplete experience. Instead, they are greeted with Ada, ensuring a complete and engaging user journey from the very beginning, whether or not they choose to customize their avatar later on.

b. **onOpen/onMinimise:** this activity happens when the user maximizes or minimizes the system. It controls the toggling of a virtual tutoring agent's visibility and dimensions within a web-based environment. It functions based on a specified 'key' parameter, which can be either 'on' or 'off.'

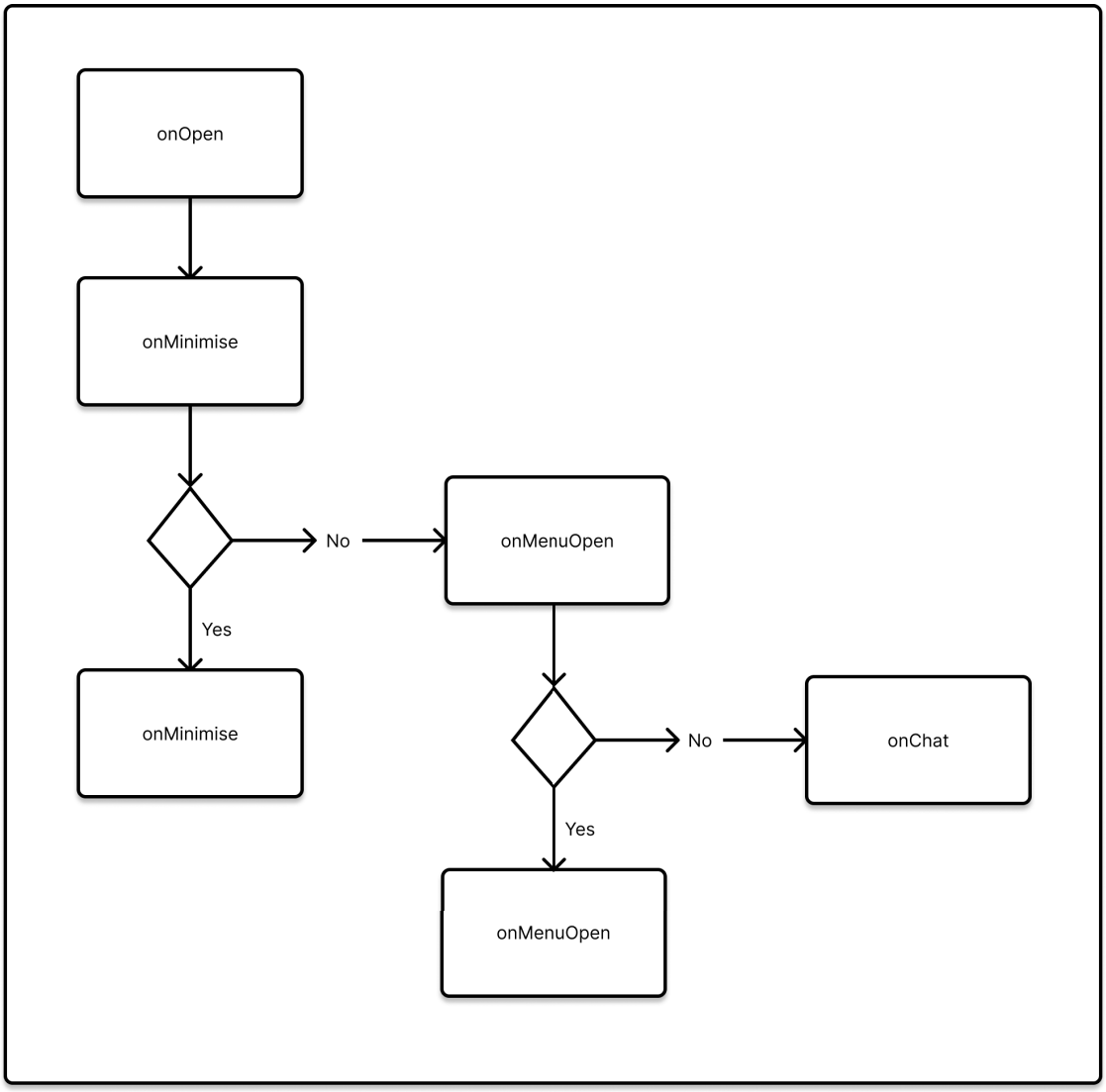

Figure 23: A section from the system logic BPMN diagram II

onOpen: It checks the user's selected avatar profile (either 'ada' or 'obi'). Depending on the profile, it triggers a welcome animation for the selected avatar. It hides a closed tutoring agent element and displays an open tutoring agent element, making the agent visible to the user. It attempts to adjust the dimensions (width and height) of an iframe element in the parent document, potentially resizing it to accommodate the visible agent.

onMinimise: It stops any ongoing keyframe animations. It shows a closed tutoring agent element and hides the open tutoring agent element, effectively making the agent invisible to the user. It attempts to adjust the dimensions (width and height) of the iframe element in the parent document, potentially resizing it to a smaller size or original dimensions.

c. **onMenuOpen/onMenuClose:** this activity happens when the user opens or closes the menu. this logic snippet manages the visibility of an options menu, allows the selection of an avatar profile, and ensures that the menu is hidden when clicking outside of it, providing a user-friendly interface for interacting with the application's settings and avatars.

onMenuOpen/onMenuClose: The logic is triggered when the open icon or the close button is pressed. It checks if the menu is currently displayed on the screen. If the menu is currently displayed, it hides. If it's not displayed, it shows the menu. This logic is also responsible for setting a local storage item named "avatarProfile" based on the argument provided.

Avatar Profile Configuration: After the user has chosen their preferred avatar in local storage, the logic checks its value. If it's set to Ada, it triggers the **Ada's Welcome** animation, and if it's set to anything else, it triggers the **Obi's Welcome** animation. This logic are responsible for configuring and displaying a welcome message or animation associated with the selected avatar profile.

d. **onChat/onSendMessage:** this logic is called when the user enters a message into the textbox and clicks on the send button. This logic manages the communication with the server and updates the chat interface, and also analyzes the content of the received message and determines an appropriate response, taking into account various intents and associated actions.

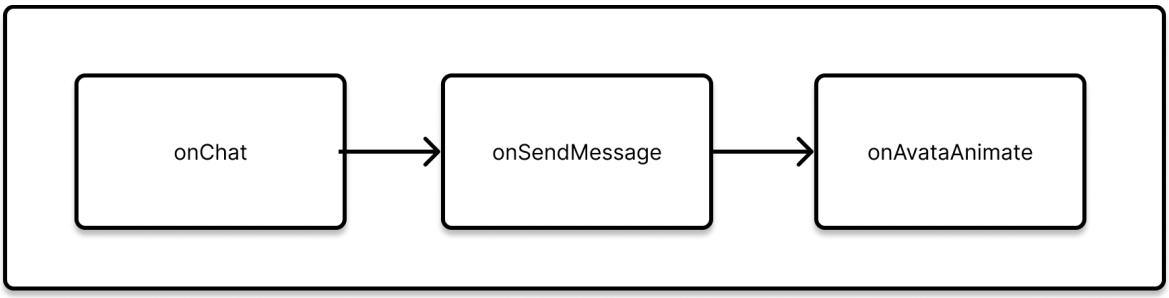

Figure 24: A section from the system logic BPMN diagram III

This logic encompasses two functions: **API calling** and **API response processing**. Here's an abstract description of their functionality:

API calling: This logic is responsible for handling communication between the client (user) and the server. It sends a request to the backend with the user's message. The logic expects a response from the server, which contains a message (**theMessage**) and an associated emotion (**theEmotion**). It updates the chat interface with two chat

bubbles: one representing the user's message and the other representing the agent's response.

API response processing: This processes the response message received from the server to determine the appropriate response. Depending on the content of the input, it selects a random response from the corresponding intent category. Additionally, based on the selected response, it animates the avatar (like displaying a surprised expression).

e. **onAnimate:** this activity happens when the user opens the system, changes the avatar, or sends a message. It is the subsystem of logic behind the animation of the avatar. This logic helps control how the avatar looks and acts on a webpage. It makes the avatar do different things, like showing a welcoming animation or a continuous idle motion. The logic also manages a timer, like a countdown clock, which triggers these actions.

The implementation of the animation was done with sprite animation. The sprite generated from the blender keyframe renders. The approach here was to create a custom timeline that is powered by JavaScript and CSS. A timeline that works for each animation considering they have different durations and similar frame rates. This code manages the animations and timers for the virtual agent, providing dynamic and engaging interactions with the user in the tutoring system.

6 Results and Evaluation

This section provides a comprehensive assessment of the developed virtual tutoring system. It presents the system's effectiveness in meeting defined objectives through both quantitative and qualitative analysis. User testing and feedback are analyzed to evaluate its real-world performance and identify potential areas for enhancement. This section offers valuable insights into the system's practical viability and its user experience.

6.1 Result

Figure 25: the system when minimized.

The finished system is a virtual tutoring platform frontend that offers a rich and engaging learning experience. With avatars capable of dynamic movements, expressions, and interactions, creating a sense of presence for users. The use of innovative design tools and technologies, coupled with the implementation process, ensures that the avatars are not only visually appealing but also highly functional.

The avatars, both male (Obi) and female (Ada), are crafted with attention to detail, capturing the nuances of the upper body and facial features. This model style enhances the relatability of the avatars, enabling them to effectively convey information

and engage users in a meaningful manner. Additionally, the seamless integration of animation brings an extra layer of interactivity, making the learning experience dynamic and immersive.

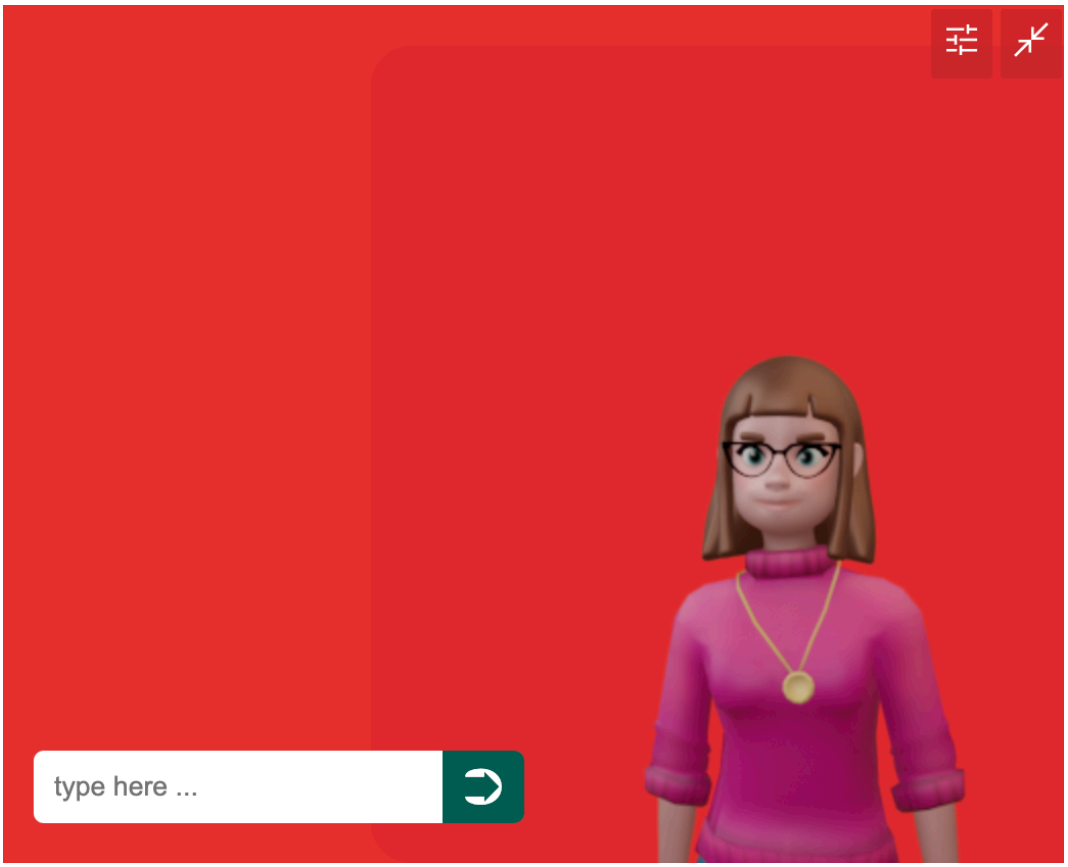

Figure 26: the system when opened.

The frontend, powered by HTML, CSS, and JavaScript, provides an intuitive and visually appealing user interface. AJAX enables real-time communication, enhancing the responsiveness of the system.

6.2 Comparison of Agent Implemented and Criteria

The implementation of the virtual tutoring system is anticipated to yield several key outcomes, aligning with the specified design and functionality criteria:

- **The proposed agent will be Human agent:** The avatar was designed to resemble a human figure, incorporating facial features, body language, and attire that conveyed a relatable and approachable appearance.
- **The age of agent, which is between 20 and 30 years:** The age range of the agent, set between 20 and 30 years, was accurately represented in the avatar's

appearance. The design ensured that the agent's features and attire were consistent with this age range, aligning with the intended user demographic and fostering a sense of relatability.

- **The system will have two agents, both male and female agent:** The system successfully implemented the presence of two agents, one male and one female, as per the proposed criteria. This allowed users to choose an agent that they felt most comfortable interacting with, accommodating diverse user preferences and needs.
- **The agent will be Caucasian male and Caucasian female:** The ethnicity and race of the agents were appropriately represented as Caucasian male and Caucasian female.
- **Agent with features such as facial expressions and gestures:** The proposed agent was designed as a tutoring agent with advanced features including facial expressions and gestures. These dynamic elements were integrated using a combination of 3D modeling and animation techniques, allowing the agent to convey emotions and engage users in a more interactive and immersive manner. This significantly enhanced the agent's effectiveness in conveying educational content and maintaining user interest.
- **Backend integration:** Backend integration was successfully achieved through the development of a functional API that was compatible with mainstream backend technologies. Enabling real-time processing of user inputs and generation of appropriate responses from the agent. This integration played a crucial role in ensuring the overall functionality and responsiveness of the system.
- **Integration with ONYX/Opal Overlay:** The virtual tutor and chat functionality will seamlessly integrate into the existing ONYX/Opal platform. This integration will be designed to ensure a non-disruptive user experience, allowing users to access the virtual tutoring system without the need for extensive platform migration or adaptation. The overlay will be intuitively designed, maintaining the familiar ONYX/Opal interface while introducing the additional functionality of the virtual tutor.

In summary, the system effectively met all the specified criteria, demonstrating a high level of adherence to the design goals and user preferences. The combination of accurately modeled avatars, dynamic features, and robust backend integration contributed to the creation of a virtual tutoring frontend that is engaging, relatable, and highly effective in supporting the learning process.

6.3 Evaluation and User Feedback

A public survey was conducted for the evaluation of the developed system. This was done by hosting the project on a public domain and integrating it to a page on the school website. The survey which was created using Google Forms contained 9

questions which cut across the profession of the evaluator, the look and animation of the agents, and general remarks, suggestions, and remarks for future development on the project.

The survey contained the following question:

- What is your current professional role?

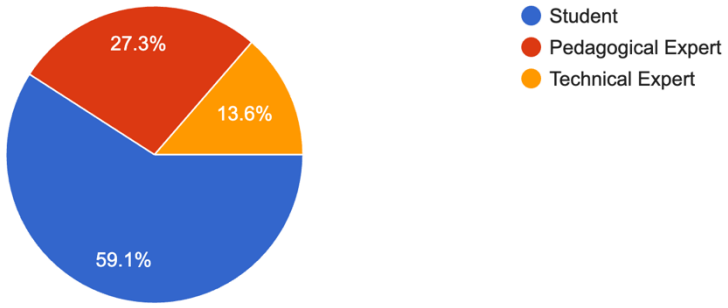

Figure 27: A pie chart representing the professions of evaluators

The survey was done by a total of 22 people which included 13 students, 6 pedagogical experts, and 3 technical experts as shown in Figure above.

- How did you feel about the Avatar as humans? Please give your opinion based on the following points.

Questions		Student	Pedagogical Expert	Technical Expert	Overall
Evaluators		59.1% (13)	27.3% (6)	13.6% (3)	**22**
Review on Avatars as humans	Appearance	92.3%	33.3%	66.6%	**64%**
	Life-Likeness	76.9%	16.6%	66.6%	**42.2%**
	Realism	69.2%	16.6%	33.3%	**39.7%**
	Animation	100%	33.3%	66.6%	**66.6%**
	Half-body Presentation	84.6%	50%	33.3%	**55.9%**
	Role as Tutor (Foruniversity students)	53.8%	0%	33.3%	**29%**
	total	**79.4%**	**27.7%**	**50%**	**52.4%**

Table 3: positive votes from "avatars as human" question

Both technical experts and students value avatars that closely resemble humans. Technical experts place a strong emphasis (66.6%) on appearance, life-likeness, and animation. They also find potential in avatars acting as tutors for university-level students (33.3%). However, there is less consensus (33.3%) on the preference for half-body presentation. In contrast, students highly favor avatars with human-like appearances (92.3%), appreciating life-likeness (76.9%) and animation (100%) as well. They also see potential in half-body presentations (84.6%), but only about half of them (53.8%) consider the avatar's role as a tutor. Pedagogical experts have a more varied perspective, attributing 33.3% of their consideration to appearance and animation, while half-body presentation is important to 50% of them. Life-likeness and realism account for 16.6% each, and the avatar's role as a tutor for university students is not explicitly mentioned. These factors collectively contribute to 27.7% of the experts' evaluation of human-like avatars.

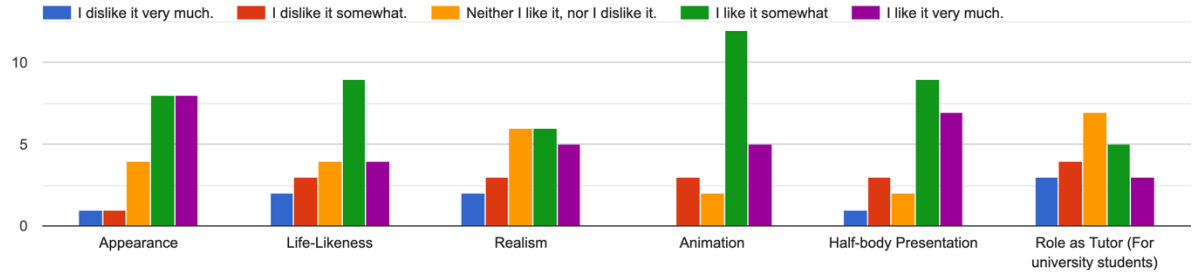

Figure 28: A bar chart representing feedback from "avatars as human"

- What do you think about the outlook of the male avatar "Obi"?

Questions		Student	Pedagogical Expert	Technical Expert	Overall
What do you think about the outlook of the male avatar "Obi"?	Age	92.3%	100%	100%	**97.4%**
	Ethnicity	100%	66.6%	100%	**88.6%**
	Clothing	92.3%	66.6%	66.6%	**75.1%**
	Hair style and beard	100%	83.3%	100%	**94.4%**
	total	**96.1%**	**79.1%**	**91.6%**	**88,9%**

Table 4: positive votes from "outlook of Obi" question

The male avatar "Obi" receives overwhelmingly positive feedback from both students and experts. Students express exceptionally high satisfaction, with 92.3% contentment regarding attributes like age and clothing, and a perfect 100% agreement on ethnicity and hairstyle/beard. Pedagogical experts also commend "Obi", particularly appreciating attributes such as age, hair style, and beard, with approval ratings ranging from 66.6% to 100%. Clothing and ethnicity receive positive feedback as well, each with a 66.6% approval rating. Among technical experts, there is unanimous appreciation for considerations given to age, ethnicity, and hair style/beard, all garnering 100% consensus. While 66.6% of experts approve of the clothing choices, there is a suggestion for potential refinement in this area. Overall, these high approval percentages indicate a favorable perception of "Obi" among all three groups, highlighting the importance of thoughtful design in educational avatars.

Figure 29: A bar chart representing feedback from "outlook of Obi"

- What do you think about the outlook of the female avatar "Ada"?

Questions		Student	Pedagogical Expert	Technical Expert	Overall
What do you think about the outlook of the female avatar "Ada"?	Age	92.3%	100%	66.6%	**86.3%**
	Ethnicity	84.6%	83.3%	100%	**89.3%**
	Clothing	92.3%	83.3%	66.6%	**80.6%**
	Hair style and beard	100%	83.3%	100%	**94.4%**

	total	**92.3%**	**87.4%**	**83.3%**	**87,6%**

Table 5: positive votes from "outlook of Ada" question

Students highly appreciate the female avatar "Ada," with strong satisfaction expressed across various aspects. Specifically, 92.3% of respondents find the avatar's age representation fitting, and 84.6% are pleased with the portrayal of ethnicity, resonating with their backgrounds. Clothing choice receives a substantial approval rating of 92.3%, and hair style and beard are unanimously well-received at 100%. Pedagogical experts also hold a favorable view, with an overall approval rate of 87.4%. They particularly appreciate age, ethnicity, clothing, and hairstyle considerations, each receiving positive ratings ranging from 83.3% to 100%. Technical experts provide a predominantly positive view, with a total approval rating of 83.3%. They unanimously endorse ethnicity and hair style at 100%, and clothing and age receive a commendable 66.6% approval. This feedback collectively indicates a strong positive sentiment towards the design and representation of the female avatar "Ada" across a diverse range of experts.

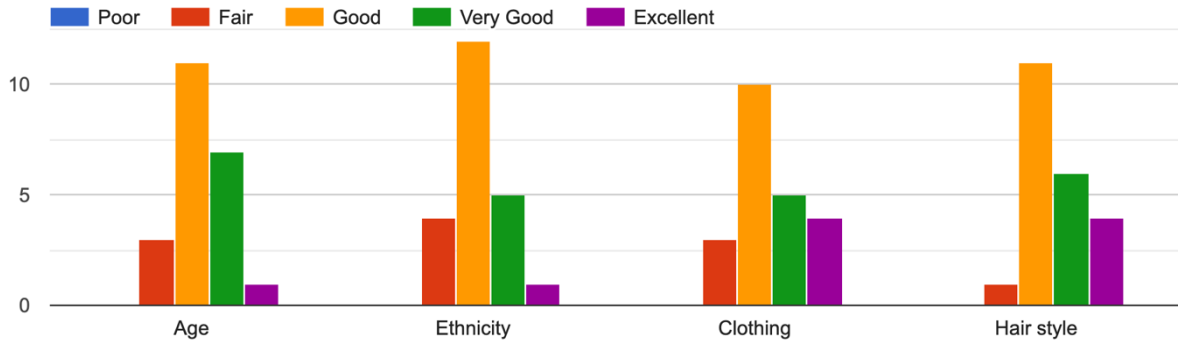

Figure 30: A bar chart representing feedback from "outlook of Ada"

- How would you rate the level of animation of the male avatar "Obi"?

Questions		Student	Pedagogical Expert	Technical Expert	Overall
How would you rate the level of animation of the male avatar "Obi"?	Hello	92.3%	83.3%	100%	**91.8%**
	Idle	92.3%	33.3%	33.3%	**52.9%**
	Happy	100%	66.6%	33.3%	**66.6%**
	Sad	92.3%	83.3%	66.6%	**80.7%**

	Surprise	92.3%	16.6%	33.3%	**47.4%**
	Bye	92.3%	50%	33.3%	**58.5%**
	total	**93.5%**	**55.5%**	**49.9%**	**66.3%**

Table 6: positive votes from "animation of Obi" question

The male avatar "Obi" receives high ratings for its animation across different emotional states from students. Impressively, animations like "Hello," "Idle," "Sad," "Surprise," and "Bye" all score 92.3%, with "Happy" achieving a perfect 100%. The overall average animation rating for "Obi" is an impressive 93.5%, indicating a strong positive reception from students. Pedagogical experts also rate Obi's animation positively, with the highest scores for "Hello," "Sad," and "Happy" ranging from 66.6% to 83.3%. However, "Surprise" received a lower rating of 16.6%, suggesting room for improvement. While "Idle" scored 33.3%, there's potential for enhancement. "Bye" animations were rated at 50%. In total, the average animation rating is 55.5%, showing an overall positive perception, with specific areas identified for potential improvement. Technical experts gave "Hello" a perfect score of 100%, while other animations like "Idle," "Happy," "Surprise," and "Bye" received ratings of 33.3%. The "Sad" animation received the highest rating at 66.6%. The average animation level for "Obi" stands at approximately 49.9%, indicating room for potential improvements in certain animations for better performance and engagement.

Figure 31: A bar chart representing feedback from "animation of Obi"

- How would you rate the level of animation of the female avatar "Ada"?

Questions		Student	Pedagogical Expert	Technical Expert	Overall
How would you rate the level of animation of the male avatar "Ada"?	Hello	84.6%	83.3%	100%	**91.8%**
	Idle	76.9%	16.6%	33.3%	**52.9%**

	Happy	100%	50%	66.6%	**66.6%**
	Sad	84.6%	33.3%	66.6%	**80.7%**
	Surprise	100%	16.6%	33.3%	**47.4%**
	Bye	100%	50%	33.3%	**58.5%**
	total	**91%**	**41.6%**	**55.5%**	**62.7%**

Table 7: positive votes from "animation of Obi" question

Students highly appreciate the animation level of the female avatar "Ada," giving it an impressive overall approval rating of 91%. Specifically, animations for emotions like "Hello," "Happy," "Sad," "Surprise," and "Bye" received very positive feedback, ranging from 84.6% to a perfect 100%. Pedagogical experts also responded favorably, with 83.3% approving of the "Hello" animation, and 50% finding "Happy" and "Bye" animations effective. The "Sad" animation garnered a 33.3% approval. Technical experts rated the animations positively, with the "Hello" animation receiving a perfect 100% approval. "Happy" and "Sad" animations both scored 66.6%, while "Idle," "Surprise," and "Bye" animations received around 33.3% approval. Overall, technical experts gave an average approval of about 55.5%. These results collectively demonstrate a strong positive reception for the animation levels of the female avatar "Ada" across different expert groups.

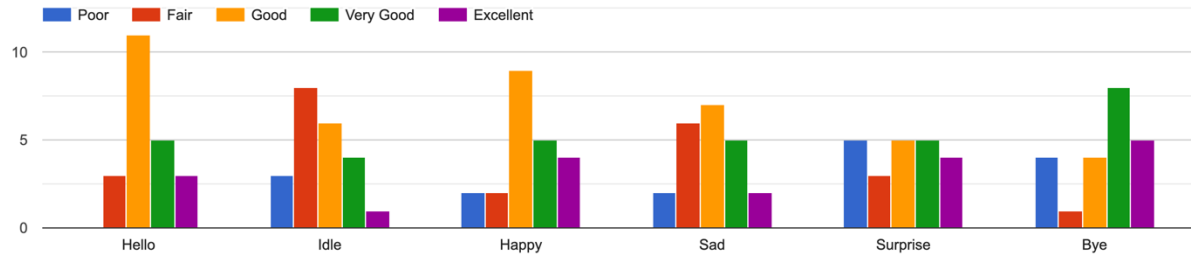

Figure 32: A bar chart representing feedback from "animation of Ada."

- Feedback from avatars as human

This feedback was diverse. Students generally appreciated the avatar's blinking feature, finding it engaging and realistic. Some students expressed interest in seeing a broader range of expressions for a more dynamic interaction. A few students felt that the use of avatars for virtual tutoring at the master's level might require further consideration. They emphasized the importance of genuine human interactions in

educational settings. Among pedagogical experts, there was a mixed response. Some felt that the avatar might be perceived as too playful or childlike for a university context. Others expressed a preference for human facilitators, citing the need for understanding students' emotions and behavior. Technical experts noted that the avatar's responsiveness needed improvement, though they generally found its appearance to be impressive. Overall, the feedback reflects a range of perspectives, indicating areas for potential refinement and highlighting the importance of considering both technical and pedagogical aspects in avatar design for educational settings.

- Feedback from the outlook of the avatars

Students generally liked the avatars, but some suggested making Ada look more lifelike. One student mentioned being unfamiliar with the Outlook avatar. Pedagogical experts wanted questions about teaching aspects, and they stressed the need for diverse avatars, allowing students to choose. Technical experts had mixed opinions, with one noting a problem with responsiveness. Others thought the avatars looked good. Overall, the avatars are liked, but there's room to improve customization, diversity, and responsiveness.

- Feedback from the outlook of the avatars

Overall, the feedback on the level of animation for the avatars was positive. Students appreciated the subtle, continuous movements of the avatars, finding them engaging. However, there were suggestions for improvement, particularly concerning the chat box functionality. Some students noted that sent messages should automatically clear from the text box for a smoother user experience. Pedagogical experts provided valuable insights on specific expressions, suggesting adjustments for a more realistic portrayal of emotions. Technical experts emphasized the need for further refinement, especially in reactions like 'surprise' and 'bye', and recommended enhancements to the GUI. Despite these suggestions for improvement, the general consensus was that the animations were promising, with potential for further enhancement and optimization.

6.4 Testing Methodologies and Scenarios

6.4.1 Speed Test

The speed of our virtual tutoring system is a critical aspect that directly impacts user experience. We conducted a series of tests to evaluate the system's performance in terms of startup time and resource loading.

Initial Startup Time: For a first-time user, the system takes approximately 32 seconds to initialize. During this period, the system loads essential resources such as animation sprites, images, and system files to the client device for the first time. This initial load ensures that the necessary components are readily available for seamless operation.

Subsequent Load Times: On subsequent loads, the system demonstrates significantly improved performance, requiring only 11 seconds to start up. This reduced load time is attributed to the caching of resources on the client device after the initial use. As a result, the system can retrieve stored data, minimizing the time required for initialization.

Animation Load Times: Upon first-time opening, the system efficiently loads the welcome animation, taking less than 2 seconds to display this engaging introductory sequence. This swift loading time ensures that users are promptly greeted by the avatar, setting a positive tone for their learning experience. Messages take 60 milliseconds while Subsequent animations take maximum of 70 milliseconds.

These speed tests demonstrate that our virtual tutoring system strikes a balance between resource-intensive initial loading and subsequent, optimized performance. The careful management of resource loading ensures that users can swiftly access the platform and engage with the avatars, enhancing the overall usability and interactivity of the system.

6.4.2 Unit Testing

Unit testing focused on examining individual components and functions within the system. Each module was isolated and tested for correctness, input validation, and edge cases. For example, the startTimer(seconds) {…} was tested to verify that it correctly processed user inputs, generated appropriate API requests, and handled responses from the backend. With the console logs always returns desired logs.

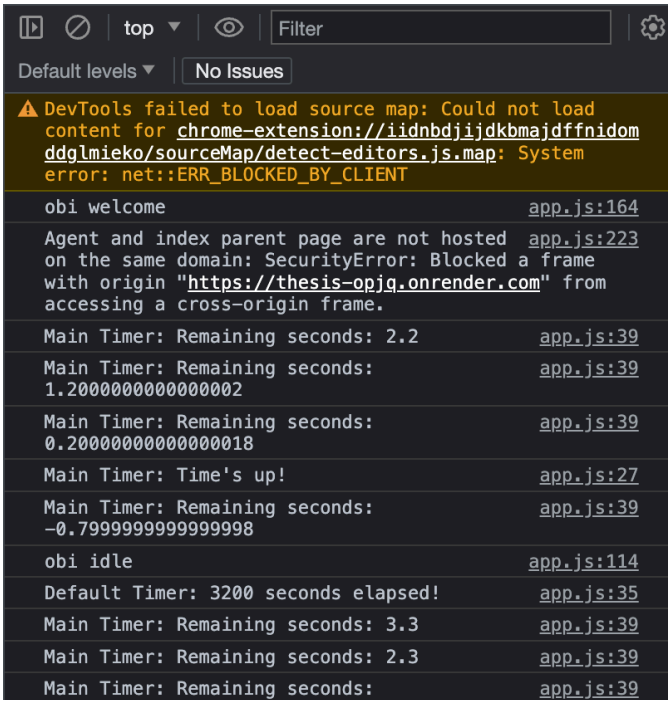

Figure 33: cross-section of the console log

6.4.3 Integration Testing

Integration testing was conducted to evaluate the interactions between different system components. This included testing the seamless integration between the user interface, chat controller, API, emotion controller, and avatar. For instance, it was verified that the chat controller effectively communicated with the API to retrieve responses and emotions.

6.4.4 Compatibility Testing

Compatibility testing was conducted to ensure the system functioned seamlessly across various devices and browsers. This included testing on different operating systems (Windows, macOS, Linux) and browsers (Chrome, Firefox, Safari, Edge) to verify consistent performance.

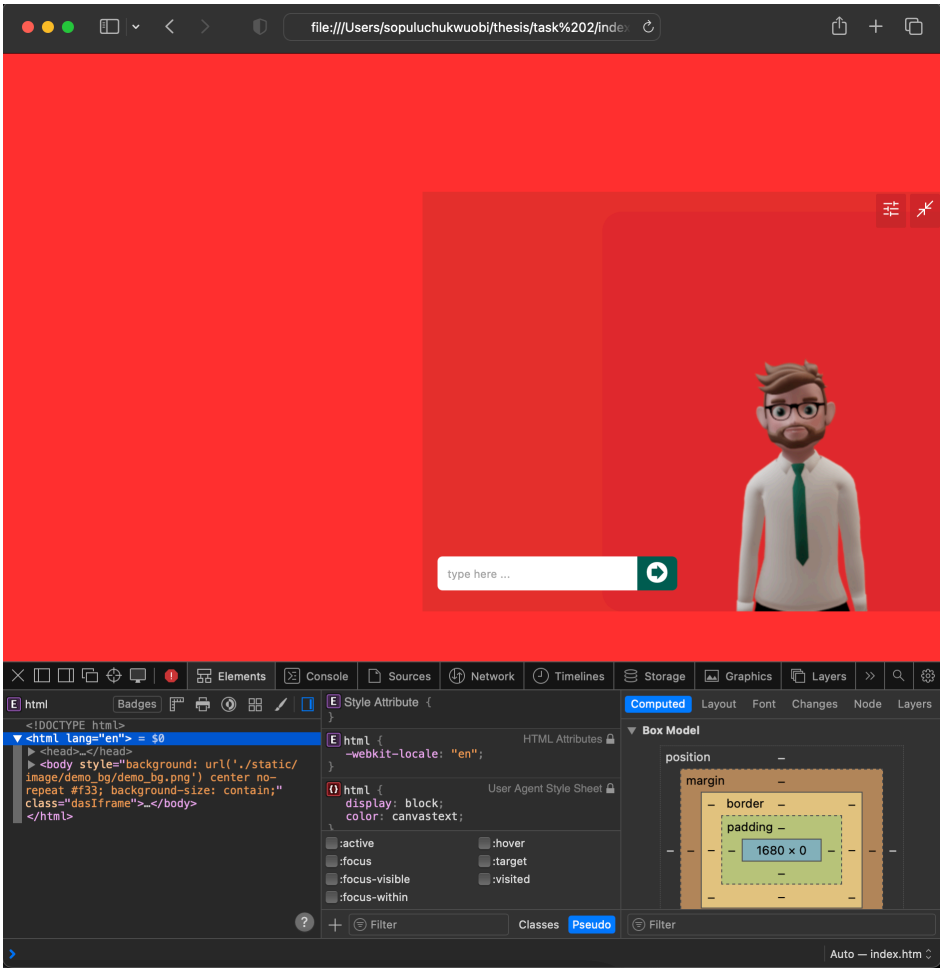

Figure 34: the system running on Safari browser.

The combination of these testing phases ensured that the system met high standards of functionality, performance, security, and user experience, ultimately resulting in a robust and reliable virtual tutoring platform.

7 Conclusion and Future Work

For a virtual tutoring system, the development of the frontend featuring human-like agents with emotion capabilities is a step in the right direction. This integration holds the promise of revolutionizing the learning experience, rendering it more engaging and effective for users. By embodying avatars with human-like traits and emotions, the system taps into a realm of interactive learning that has the potential to resonate with a diverse user base. The successful implementation of pivotal criteria, such as achieving a human-like appearance and specifying an appropriate age range for the agents, underscores the system's adaptability and potential to cater to a wide spectrum of learners.

However, it's imperative to acknowledge the limitations encountered during the implementation phase, with time emerging as a prominent constraint. Given the expansive scope of the project, an extended timeline would have facilitated finer animation refinement, enhanced optimization, and the incorporation of additional features. This temporal restriction, though a challenge, underscores the potential for even greater strides in future iterations of the system.

Looking ahead, there exist myriad opportunities for further enhancement and refinement. Leveraging the capabilities of the Three.js JavaScript framework, for instance, presents an avenue for transitioning from 2D sprites to 3D models. This transition not only holds the potential to elevate the system's overall performance but also stands to improve loading times, providing a seamless user experience. Moreover, the precision of emotion animations represents a critical area of focus, with the aim of achieving even greater accuracy in emotional representation, thus further enhancing the agent's user interaction.

In the pursuit of broader accessibility and appeal on a global scale, delving into multi-language support and cultural adaptability remains a pivotal future endeavor. This expansion would not only cater to diverse linguistic backgrounds but also ensure that the system resonates with users from varying cultural contexts. Additionally, the incorporation of advanced machine learning techniques stands as an exciting frontier for further development. Personalized learning paths and content recommendations, driven by sophisticated algorithms, hold the potential to significantly elevate the system's efficacy in catering to individual learner needs.

Furthermore, diversifying customization options is another avenue for enriching user engagement. The provision of an extended array of avatars or even enabling users to craft their own avatars within the system would empower users to create a learning

environment that aligns perfectly with their preferences and learning style. These customizable elements, when integrated, have the potential to substantially enhance user immersion and overall satisfaction.

In summation, the strides made in developing the frontend agents within the virtual tutoring system represent a pivotal milestone in the educational technology. The integration of human-like avatars, coupled with emotion capabilities, holds the promise of transforming the learning landscape. While the current implementation lays a robust foundation, the identified areas for future development pave the way for an even more dynamic, personalized, and globally accessible learning experience. Through continuous refinement and the integration of cutting-edge technologies, the virtual tutoring system is poised to evolve into an indispensable tool for learners worldwide.

Bibliography

[1] C. G. Brinton, R. Rill, S. Ha, M. Chiang, R. Smith and W. Ju, "Individualization for Education at Scale: MIIC Design and Preliminary Evaluation," in IEEE Transactions on Learning Technologies, vol. 8, no. 1, pp. 136-148, 2015.

[2] T. Köhler, E. Schoop, N. Kahnwald, R. Sonntag, "*Digitality and Diversity Overcoming Barriers with Digital Transformation"* in *Communities in New Media. Proceedings of 25th Conference GeNeMe* (TUDpress Verlag der Wissenschaften GmbH), pp. XXXV–XL, 2022.

[3] U. U. Shegupta, R, Schmidt and W. Hardt, "Analysis of Assessment Methodologies in Learning Management System for Scalable Mentoring Processes" IBS International Symposium on Computer Science, Computer Engineering and Educational Technology, pp. 3-8, 2019.

[4] S. Oeste, K. Lehmann, A. Janson, M. Söllner and J. M. Leimeister, "Redesigning University Large Scale Lectures: How To Activate The Learner. Academy of Management Proceedings" In Academy of Management Annual Meeting (AOM), Vancouver, BC, Canada, 2015.

[5] S. Wollny, J. Schneider, D. Di Mitri, J. Weidlich, M. Rittberger, and H. Drachsler, "Are We There Yet? - A Systematic Literature Review on Chatbots in Education. *Frontiers in Artificial Intelligence*. Frontiers Media S.A., 2021.

[6] R. Jain, T. Saha, S. Chakraborty and S. Saha, "Domain Infused Conversational Response Generation for Tutoring based Virtual Agent," *2022 International Joint Conference on Neural Networks (IJCNN)*, Padua, Italy, pp. 1-8, 2022.

[7] U. U. Shegupta, G. V. Gowthaman Malarvizhi, M. Jose, T. S. Kumar, W. Hardt, "INTERACTIVE TUTORING AGENT GENERATION FOR STUDENT ASSISTANCE IN HIGHER EDUCATION" in *EDULEARN22 Proceedings* (IATED), vol. 1, pp. 9861–9869, 2022.

[8] A. C. Graesser, X. Hu and R. Sottilare, "Intelligent tutoring systems" in *International Handbook of the Learning Sciences*, Taylor and Francis, pp. 246–255, 2018.

[9] T. Schmidt, C. Wolff, "The influence of user interface attributes on aesthetics" in I-Com. Walter de Gruyter GmbH, pp. 41–55, 2018.

[10] Bonner, Desmond, S. Gilbert, M. C. Dorneich, E. Winer, A. M. Sinatra, A. Slavina, A. Macallister, and J. Holub, "The challenges of building intelligent tutoring systems for teams" In *Proceedings of the Human Factors and Ergonomics Society*, Human Factors an Ergonomics Society Inc, pp. 1974–1978, 2016.

[11] M. Malekzadeh, M. B. Mustafa, A. Lahsasna, "A review of emotion regulation in intelligent tutoring systems" *Educational Technology and Society*. International

Forum of Educational Technology,National Taiwan Normal Universityand Society 18, pp. 435–445, 2015.

[12] J. Yan and R. Agada, "Intelligent Agent Embodied Online Tutoring System," *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, USA, pp. 613-617, 2018.

[13] J. A. Kulik, and J. D. Fletcher, “Effectiveness of intelligent tutoring systems: A meta-analytic review”, Review of Educational Research, 85, 171–204, 2015.

[14] V. Aleven, B. M. McLaren, I. Roll, and K. R. Koedinger, “Help helps, but only so much: Research on help seeking with intelligent tutoring systems”, International Journal of Artificial Intelligence in Education, 26(1), 205–223, 2016.

[15] Calvo, Rafael, S. D’Mello, J. Gratch, A. Kappas, R. S.J.d. Baker, and J. Ocumpaugh (2014) “Interaction-based affect detection in educational software. In The Oxford Handbook of Affective Computing” Oxford University Press, Oxford, UK, pp. 233–245, 2014.

[16] M. Chassignol, A. Khoroshavin, A. Klimova, & A. Bilyatdinova, “Artificial Intelligence trends in education: A narrative overview” In *Procedia Computer Science* Vol. 136, Elsevier B.V., pp. 16–24, 2018.

[17] G. Westerfield, A. Mitrovic and M. Billinghurst, “Intelligent augmented reality training for motherboard assembly” *International Journal of Artificial Intelligence in Education* 25, pp. 157–172, 2015.

[18] J. R. Anderson, C. Franklin Boyle, & B. J. Reiser, “Intelligent tutoring systems“, *Science*, *228*(4698), pp. 456–462, 1985.

[19] K. R. Koedinger, E. Brunskill, R. S. J. D. Baker, E. A. McLaughlin, and J. Stamper, “New potentials for data-driven intelligent tutoring system development and optimization” *AI Magazine*, *34*(3), pp. 37–41, 2013.

[20] A. Gulz and M. Haake, “Virtual Pedagogical Agents – Design Guidelines Regarding Visual Appearance and Pedagogical Roles.” *The IV International Conference on Multimedia and Information and Communication Technologies in Education (MICTE 2006)* 5: 1848–52, 2006.

[21] N. Ruiz, H. Yu, D. A. Allessio, M. Jalal, A. Joshi, T. Murray, J. J. Magee, J. R. Whitehill, V. Ablavsky, I. Arroyo, B. P. Woolf, S. Sclaroff, M. Betke, "Leveraging Affect Transfer Learning for Behavior Prediction in an Intelligent Tutoring System," *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, Jodhpur, India, pp. 1-8, 2021.

[22] X. Zhang, T. Miyaki, J. Rekimoto, JustSpeak: Automated, User-Configurable, Interactive Agents for Speech Tutoring. *Proceedings of the ACM on Human-Computer Interaction*, *5*(EICS). 2021.

[23] O. Kwon, K. Lee, Y. Roh, J. Huang, S. Choi, Y. Kim, H. Jeon, Y. Oh, Y. Lee, B. Kang, E. Chung, Park J and Y. Lee, “GenieTutor: A Computer-Assisted Second-Language Learning System Based on Spoken Language Understanding. In Natural language dialog systems and intelligent assistants”, Springer International Publishing, pp. 257–262, 2015.

[24] Y. Shiban, I. Schelhorn, V. Jobst, A. H¨ornlein, F. Puppe, P. Pauli, and A. M¨uhlberger, “The appearance effect: Influences of virtual agent features on performance and motivation,” Computers in Human Behavior, vol. 49, pp. 5–11, 2015

[25] V. Waterschoot, Jelte, D. Reidsma, M. Bruijnes, D. Davison, D. Heylen, J. Flokstra, and M. Theune, “Flipper 2.0: a pragmatic dialogue engine for embodied conversational agents” In Proceedings of the 18th International Conference on Intelligent Virtual Agents, pp. 43–50, 2018.

[26] V. Rus, D. Stefanescu, W. Baggett, N. Niraula, D. Franceschetti, and A. C. Graesser, “Macro-adaptation in conversational intelligent tutoring matters.” In Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), (Springer Verlag), pp. 242–247, 2014.

[27] R.J.S. Sloan, “*Virtual character design for games and interactive media”,* CRC Press, pp. 1–222, 2015.

[28] A. Z. B. M. Ali, “Different realism designs of 2d virtual agents and its’ arousal effect on students emotions in learning,” 2020.

[29] K. Raman, A. N. Othman, M. Z. Idris, and V. Muniady, “Kawaii-style pedagogical agents designs in virtual learning environment: A research conceptual framework,” 2021.

[30] T.-H. Peng and T. H.Wang, “Developing an Analysis Framework for Studies on Pedagogical Agent in an e-Learning Environment.” *Journal of Educational Computing Research*, *60*(3), 547–578, 2022.

[31] L. Xu, X. Li, N. Tang, H. Wang, M. Liao, N. Liu, “Learning with Pedagogical Agent: Effects of Transfer of Information Function.” In: Gao, Q., Zhou, J. (eds) Human Aspects of IT for the Aged Population. HCII 2023. Lecture Notes in Computer Science, vol 14042. Springer, Cham. 2023.

[32] B. D. Nye, A. C. Graesser and X. Hu, “AutoTutor and family: A review of 17 years of natural language tutoring.” International Journal of Artificial Intelligence in Education, Springer Science and Business Media, LLC. 2014.

[33] J. E. Perez, D. D. Dinawanao and E. S. Tabanao, “JEPPY: An interactive pedagogical agent to aid novice programmers in correcting syntax errors.” International Journal of Advanced Computer Science and Applications, (2), pp. 48–53, 2020

[34] N. Matsuda, C. L. Griger, N. Barbalios, G. J. Stylianides, W. W. Cohen, K.R. Koedinger, “Investigating the Effect of Meta-cognitive Scaffolding for Learning by Teaching.” In: Trausan-Matu, S., Boyer, K.E., Crosby, M., Panourgia, K. (eds) Intelligent Tutoring Systems. ITS 2014. Lecture Notes in Computer Science, vol 8474. Springer, Cham, 2014.

[35] H. Van der Meij, J. van der Meij, and R. Harmsen, “Animated pedagogical agents effects on enhancing student motivation and learning in a science inquiry learning environment.” Education Tech Research Dev 63, pp. 381–403, 2015.

[36] R. Azevedo, S. A. Martin, M. Taub, N. V. Mudrick, G. C. Millar and J. F. Grafsgaard, “Are pedagogical agents’ external regulation effective in fostering learning with intelligent tutoring systems?” In A. Micarelli, J. Stamper, & K. Panourgia (Eds.), Intelligent Tutoring Systems, ITS Vol. 9684, pp. 197–207, 2016.

[37] Y. Kim, J. Thayne, and Q. Wei, “An embodied agent helps anxious students in mathematics learning.” Educational Technology Research and Development, 65(1), pp. 219–235, 2017.

[38] Duffy, M. C., & Azevedo, R., “Motivation matters: Interactions between achievement goals and agent scaffolding for self-regulated learning within an intelligent tutoring system.” Computers in Human Behavior, 52, pp. 338–348, 2015.

[39] Y. Shiban, I. Schelhorn, V. Jobst, A. Hörnlein, F. Puppe, P. Pauli and A. Mühlberger, “The appearance effect: Influences of virtual agent features on performance and motivation.” Computers in Human Behavior, 49, pp. 5–11. 2015.

[40] B. Sjödén, and A. Gulz, “From learning companions to testing companions.” In C. Conati, N. Heffernan, A. Mitrovic and M. F. Verdejo (Eds.), Artificial Intelligence in Education: 17th International Conference, AIED 2015, Madrid, Spain, pp. 459–469, 2015.

[41] H. I. Yung, and F. Paas, “Effects of cueing by a pedagogical agent in an instructional animation: A cognitive load approach.” Educational Technology & Society, 18(3), pp. 153–160, 2015.

[42] S. Lalle, N. V. Mudrick, M. Taub, J. F. Grafsgaard, C. Conati, and R. Azevedo, “Impact of individual differences on affective reactions to pedagogical agents scaffolding.” In D. Traum, W. Swartout, P. Khooshabeh, S. Kopp, S. Scherer, and A. Leuski (Eds.), Intelligent Virtual Agents, IVA 2016, vol. JOURNAL OF EDUCATORS ONLINE 10011, pp. 269–282, 2016.

[43] T. Terzidou, T. Tsiatsos, C. Miliou and A. Sourvinou, "Agent Supported Serious Game Environment," in IEEE Transactions on Learning Technologies, vol. 9, no. 3, pp. 217-230, 1 July-Sept. 2016.

[44] T. W. Liew, S. M. Tan and H. Ismail, “Exploring the effects of a non-interactive talking avatar on social presence, credibility, trust, and patronage intention in an e-commerce website.” Human-Centric Computing and Information Sciences, 7(42), 2017.

[45] N. Thompson, T. J. McGill, “Genetics with Jean: the design, development and evaluation of an affective tutoring system.” Education Tech Research Dev 65, pp. 279–299, 2017.

[46] R. Moreno and T. Flowerday, “Students’ choice of animated pedagogical agents in science learning: A test of the similarity-attraction hypothesis on gender and ethnicity,” Contemporary educational psychology, vol. 31, no. 2, pp. 186–207, 2006.

[47] F. Abbattista, G. Catucci, G. Semeraro, and F. Zambetta, “Samir: A smart 3d assistant on the web.” PsychNology J., vol. 2, no. 1, pp. 43–60, 2004.

[48] S. Branham, “Creating physical personalities for agents with faces: Modeling trait impressions of the face,” in Proceedings of the UM2001 Workshop on Attitudes, Personality and Emotions in User-Adapted Interactions. Sonthofen Germany, 2001.

[49] N. K. Person and A. C. Graesser, “Instructional design - pedagogical agents and tutors,” 2021. [Online]. Available: https://education.stateuniversity.com/pages/2095/Instructional-Design-PEDAGOGICAL-AGENTS-TUTORS.html

[50] C.-C. Wang and W.-J. Yeh, “Avatars with sex appeal as pedagogical agents: Attractiveness, trustworthiness, expertise, and gender differences,” Journal of Educational Computing Research, vol. 48, no. 4, pp. 403–429, 2013.

[51] H. V. Diez, S. Garc´ıa, J. R. S´anchez, and M. del Puy Carretero, “3d animated agent for tutoring based on webgl,” in Proceedings of the 18th International Conference on 3D Web Technology, pp. 129–134, 2013.

References from Computer Engineering Professorship

[2] T. Köhler, E. Schoop, N. Kahnwald, R. Sonntag, "*Digitality and Diversity Overcoming Barriers with Digital Transformation"* in *Communities in New Media. Proceedings of 25th Conference GeNeMe* (TUDpress Verlag der Wissenschaften GmbH), pp. XXXV–XL, 2022.

[3] U. U. Shegupta, R, Schmidt and W. Hardt, "Analysis of Assessment Methodologies in Learning Management System for Scalable Mentoring Processes" IBS International Symposium on Computer Science, Computer Engineering and Educational Technology, pp. 3-8, 2019.

[7] U. U. Shegupta, G. V. Gowthaman Malarvizhi, M. Jose, T. S. Kumar, W. Hardt, "INTERACTIVE TUTORING AGENT GENERATION FOR STUDENT ASSISTANCE IN HIGHER EDUCATION" in *EDULEARN22 Proceedings* (IATED), vol. 1, pp. 9861–9869, 2022.

[9] T. Schmidt, C. Wolff, "The influence of user interface attributes on aesthetics" in I-Com. Walter de Gruyter GmbH, pp. 41–55, 2018.

Contents in CD:

Master Thesis Documents

- Master_Thesis_Report_Obi_Sopuluchukwu_Divine.pdf
- Master_Thesis_Technical_Report.pdf
- Master_Thesis_Task_Description.pdf
- Master_Thesis_Poster.pdf

Implementation

- Frontend for Agents Folder
- Blender 3D folder

TECHNISCHE UNIVERSITÄT CHEMNITZ

Zentrales Prüfungsamt
Selbstständigkeitserklärung

Name: Obi Vorname: Sopuluchukwu Divine geb. am: 23.12.1992 Matr.-Nr.: 561419	Bitte beachten: 1. Bitte binden Sie dieses Blatt am Ende Ihrer Arbeit ein.

Selbstständigkeitserklärung*

Ich erkläre gegenüber der Technischen Universität Chemnitz, dass ich die vorliegende Masterarbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe.

Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch nicht als Prüfungsleistung eingereicht und ist auch noch nicht veröffentlicht.

Datum: 02.10.2023 Unterschrift: .....

* Statement of Authorship

I hereby certify to the Technische Universität Chemnitz that this thesis is all my own work and uses no external material other than that acknowledged in the text.

This work contains no plagiarism and all sentences or passages directly quoted from other people's work or including content derived from such work have been specifically credited to the authors and sources.

This paper has neither been submitted in the same or a similar form to any other examiner nor for the award of any other degree, nor has it previously been published.

Chemnitzer Informatik-Berichte

In der Reihe der Chemnitzer Informatik-Berichte sind folgende Berichte erschienen:

CSR-20-01 Danny Kowerko, Chemnitzer Linux-Tage 2019 - LocalizeIT Workshop, Januar 2020, Chemnitz

CSR-20-02 Robert Manthey, Tom Kretzschmar, Falk Schmidsberger, Hussein Hussein, René Erler, Tobias Schlosser, Frederik Beuth, Marcel Heinz, Thomas Kronfeld, Maximilian Eibl, Marc Ritter, Danny Kowerko, Schlussbericht zum InnoProfile-Transfer Begleitprojekt localizeI, Januar 2020, Chemnitz

CSR-20-03 Jörn Roth, Reda Harradi und Wolfram Hardt, Indoor Lokalisierung auf Basis von Ultra Wideband Modulen zur Emulation von GPS Positionen, Februar 2020, Chemnitz

CSR-20-04 Christian Graf, Reda Harradi, René Schmidt, Wolfram Hardt, Automatisierte Kameraausrichtung für Micro Air Vehicle basierte Inspektion, März 2020, Chemnitz

CSR-20-05 Julius Lochbaum, René Bergelt, Time Pech, Wolfram Hardt, Erzeugung von Testdaten für automatisiertes Fahren auf Basis eines Open Source Fahrsimulators, März 2020, Chemnitz

CSR-20-06 Narankhuu Natsagdorj, Uranchimeg Tudevdagva, Jiantao Zhou, Logical Structure of Structure Oriented Evaluation for E-Learning, April 2020, Chemnitz

CSR-20-07 Batbayar Battseren, Reda Harradi, Fatih Kilic, Wolfram Hardt, Automated Power Line Inspection, September 2020, Chemnitz

CSR-21-01 Marco Stephan, Batbayar Battseren, Wolfram Hardt, UAV Flight using a Monocular Camera, März 2021, Chemnitz

CSR-21-02 Hasan Aljzaere, Owes Khan, Wolfram Hardt, Adaptive User Interface for Automotive Demonstrator, Juli 2021, Chemnitz

CSR-21-03 Chibundu Ogbonnia, René Bergelt, Wolfram Hardt, Embedded System Optimization of Radar Post-processing in an ARM CPU Core, Dezember 2021, Chemnitz

CSR-21-04 Julius Lochbaum, René Bergelt, Wolfram Hardt, Entwicklung und Bewertung von Algorithmen zur Umfeldmodellierung mithilfe von Radarsensoren im Automotive Umfeld, Dezember 2021, Chemnitz

Chemnitzer Informatik-Berichte

CSR-22-01 Henrik Zant, Reda Harradi, Wolfram Hardt, Expert System-based Embedded Software Module and Ruleset for Adaptive Flight Missions, September 2022, Chemnitz

CSR-23-01 Stephan Lede, René Schmidt, Wolfram Hardt, Analyse des Ressourcenverbrauchs von Deep Learning Methoden zur Einschlagslokalisierung auf eingebetteten Systemen, Januar 2023, Chemnitz

CSR-23-02 André Böhle, René Schmidt, Wolfram Hardt, Schnittstelle zur Datenakquise von Daten des Lernmanagementsystems unter Berücksichtigung bestehender Datenschutzrichtlinien, Januar 2023, Chemnitz

CSR-23-03 Falk Zaumseil, Sabrina, Bräuer, Thomas L. Milani, Guido Brunnett, Gender Dissimilarities in Body Gait Kinematics at Different Speeds, März 2023, Chemnitz

CSR-23-04 Tom Uhlmann, Sabrina Bräuer, Falk Zaumseil, Guido Brunnett, A Novel Inexpensive Camera-based Photoelectric Barrier System for Accurate Flying Sprint Time Measurement, März 2023, Chemnitz

CSR-23-05 Samer Salamah, Guido Brunnett, Sabrina Bräuer, Tom Uhlmann, Oliver Rehren, Katharina Jahn, Thomas L. Milani, Güunter Daniel Rey, NaturalWalk: An Anatomy-based Synthesizer for Human Walking Motions, März 2023, Chemnitz

CSR-24-01 Seyhmus Akaslan, Ariane Heller, Wolfram Hardt, Hardware-Supported Test Environment Analysis for CAN Message Communication, Juni 2024, Chemnitz

CSR-24-02 S. M. Rizwanur Rahman, Wolfram Hardt, Image Classification for Drone Propeller Inspection using Deep Learning, August 2024, Chemnitz

CSR-24-03 Sebastian Pettke, Wolfram Hardt, Ariane Heller, Comparison of maximum weight clique algorithms, August 2024, Chemnitz

CSR-24-04 Md Shoriful Islam, Ummay Ubaida Shegupta, Wolfram Hardt, Design and Development of a Predictive Learning Analytics System, August 2024, Chemnitz

CSR-24-05 Sopuluchukwu Divine Obi, Ummay Ubaida Shegupta, Wolfram Hardt, Development of a Frontend for Agents in a Virtual Tutoring System, August 2024, Chemnitz

Chemnitzer Informatik-Berichte

ISSN 0947-5125

Herausgeber: Fakultät für Informatik, TU Chemnitz
Straße der Nationen 62, D-09111 Chemnitz