SYSTEM DESIGN AND DEVELOPMENT OF A DISTANCE LEARNING PROTOTYPE FOR A VIRTUAL MAKERSPACE

by

Ziyi Liu

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science in Mechanical Engineering

School of Mechanical Engineering

West Lafayette, Indiana

August 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Karthik Ramani, Chair

School of Mechanical Engineering

School of Electrical and Computer Engineering

Dr. Song Zhang

School of Mechanical Engineering

Dr. Stanley H. Chan

School of Electrical and Computer Engineering

Approved by:

Dr. Nicole Key

This work is dedicated to my father Liu Jinhai and my mother Chen Min for their lifelong encouragement, guidance, and support.

ACKNOWLEDGMENTS

Through this acknowledgement, I want to show my sincere gratitude and respect to my advisor and committee chair Dr. Karthik Ramani Donald W. Feddersen Professor of Mechanical Engineering, Purdue University for his precious guidance and supervision for my research work. Without his support, this thesis could not have been accomplished.

I also want to thank Dr. Song Zhang Assistant Head for Experiential Learning, and Professor of Mechanical Engineering, Purdue University and Dr. Stanley H. Chan Elmore Associate Professor of Electrical and Computer Engineering and Associate Professor of Statistics Purdue University for their valuable advice as my examining committees.

I will also use this opportunity to show my appreciation to my collaborators-Ana Villanueva,Zhu Zhengzhe, and Du Xin for their support in the execution and testing of various parts of this work. Special thanks to Wang Tianyi, Qian Xun, He Fengming, Luis Paredes, and Ananya Ipsita– senior Ph.D. students of Convergence Design Lab for technical help and intellectual support.

I would like to thank Liu Min, Lecturer of School of Mechanical Engineering, Purdue University. She introduced me to Convergence Design Lab, which helped me to begin my master's research work.

The love and support from my family always give me strength and push me forward. My girlfriend, Xu Xuejing , whose love and encouragement also give me the power to go after all my pursuits. Without them I would never have made such achievement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

τ torque

r distance measured from the axis of rotation to where the linear force is applied

F linear force

θ the angle between F and r

ABBREVIATIONS

AI	Artificial Intelligence
API	Application Program Interface/Interaction
AR	Augmented Reality
CAD	Computer Aided Design
HMD	Head Mounted Device
HTTP	Hypertext Transfer Protocol
IMU	Inertial Measurement Unit
IoT	Internet of Things
MR	Mixed Reality
NLP	Natural Language Processing
PC	Personal Computer
SLA	Stereolithography
TCP	Transmission Control Protocol
VR	Virtual Reality

ABSTRACT

Distance learning is facing a critical moment finding a balance between high quality education for remote students and engaging them in hands-on learning. This is particularly relevant for project-based classrooms and makerspaces, which typically require extensive trouble-shooting and example demonstrations from instructors. We present RobotAR, a teleconsulting robotics system for creating Augmented Reality (AR) makerspaces. We present the hardware and software for an AR-compatible robot, which behaves as a student's voice assistant and can be embodied by the instructor for teleconsultation. As a desktop-based teleconsulting agent, the instructor has control of the robot's joints and position to better focus on areas of interest inside the workspace. Similarly, the instructor has access to the student's virtual environment and the capability to create AR content to aid the student with problem-solving. We also performed a user study which compares current techniques for distance hands-on learning and an implementation of our system.

1. INTRODUCTION

1.1 Distance Learning

In recent years, interest in remote education has been steadily on the rise and many believe that it will become even more prevalent in the near future. Compared to traditional education, remote education does not require participants(students and instructors) to be physically presented at the same place. It has a number of benefits, which include but are not limited to: (a)lower cost;(b) larger coverage; (c)more flexibility (d)better scalability. Numerous companies and organizations including MOOC[1],edX[2], Futurelearn[3] have dedicated themselves to bringing out remote education to a larger audience. So far, they have all made some initial success. For example, by 2020 MOOC has an impressive 180 million students worldwide registered to choose from the 16.3k courses listed on its website[31]. These online learning platform also offer degrees and credentials for those who successfully completed the corresponding courses, a practice that aids many people's job search endeavor.

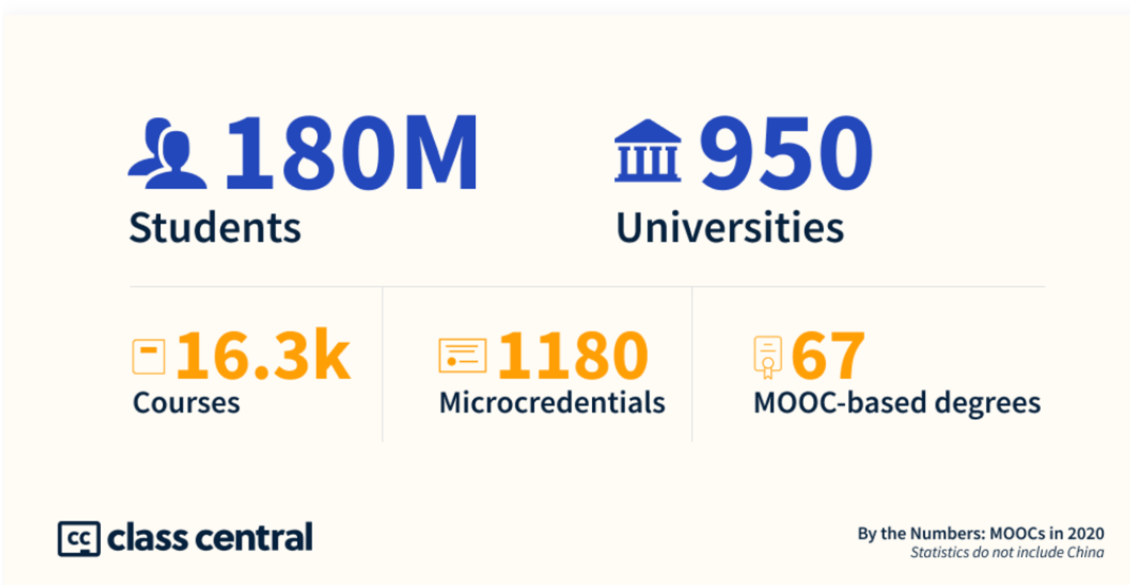

Figure 1.1. Overview of MOOC Distance Learning Platform

[h!]

How to deliver efficient and satisfactory remote learning experience has always been a popular research topic among the education community. In particular, facilitating remote

Table 1.1. Learners, Courses, Microcredentials, Degrees from the Four Most Popular Distance Learning Platforms

	Learners	Courses	Microcredentials	Degrees
Coursera	76 million	4,600[3]	610	25
edX	35 million	3,100	385	13
FutureLearn[2,4]	14 million	1,160	86	28
Swayam[2]	16 million	1,130	0	0

learning in makerspaces is one of the areas that attracted significant attention. Unlike the format of a regular lecture, makerspace and project-based classes are oriented around hands-on learning experience which require constant instructor intervention, inspection, and troubleshooting. Traditional teleconferencing tools which are designed for general purpose remote communication failed to accommodate these needs. Thus, distance education in makerspaces requires specialized tools to facilitate immersive, hands-on learning without the constraints of geographical bounds. In terms of physical embodiment at a distance, social robots as tutoring agents have demonstrated great potential at achieving learning outcomes in education[4], as well as providing students with access to consulting with the instructor in their own home environment. The robot can serve as an agent to represent the instructor during makerspaces activities, which is designed to approximate a human-to-human interaction experience in a remote setup. The robots can carry out tasks dictated by the instructor such as moving to a specific location to inspect certain components, or display guidance provided by the instructor to the students. In present times, the majority of formerly in-person classrooms have made use of online platforms, such as Zoom[5], Webex [6], Google Classroom [7], Skype[8]. These virtual platforms can offer some of the real-time capabilities as robots-for-tutoring without the cost of hardware, the concern for scalability, and the challenge of installation time; thus, the use of a robot for a distant educational setting needs to be clearly justified. The justifications are given in the form of unique benefits brought by a robotic toolkit designed for remote makerspace educaiton.

When comparing a robotics toolkit with an alternate virtual platform or agent, there are three major uses-

a) Used as tools for curricula and for students who require hands-on engagement with the physical world.

b) Used as physical embodiments which prompt students to display social behaviors which are conducive to learning.

c) Used as physical agents which provide interactions that have proven to increase learning gains as compared to virtual agents[9].

Similarly, past work has demonstrated that physical embodied tutoring agents provide an increase in compliance[10]–[12], engagement [13], [14]and conformity [15], which in turn provide an increase in cognitive learning gains[16]. These attributes are essential for bridging the gap between students and instructors engaged in remote makerspace learning.

It goes without saying that a robotic system alone cannot supplement the social in person aspect of a classroom or the advantage of having an instructor standing next to a student and helping them with problem solving. In light of this limitation, we proceed to take advantage of new technologies, such as augmented reality (AR)–which overlays virtual information into the physical world [17]–to make use of the virtual world and superimpose instructions, hints, and visual cues into a student's workspace. The device we employed to achieve AR display is a smartphone with SLAM technology enabled. We refer to makerspaces adopting AR technology as augmented makerspaces. AR allows instructors to embody and immerse themselves into the physical environment while providing engaging and visually rich learning content and guidance to the students at the same time.

Augmented reality as a nascent technology has experienced a rapid adoption in education ever since its creation. In terms of educational material, AR has been empirically implemented in classrooms for five main applications: (a) collaborative and situated learning by students exploring new interaction modalities in the same environment [18](e.g., students simultaneously exploring 3D objects in school grounds); (b) selecting and manipulating 3D objects[19](e.g., look inside the inner-workings of a system); (c) providing students with a social fabric to discuss the learning material and change attitudes towards real-world issues (e.g., students exploring an AR environment of melting glaciers); (4) visualizing abstract or invisible concepts [20](e.g., pressure, temperature, current flow in a circuit); (5) creating a transition between formal and informal learning (e.g., lecture vs. laboratory experiment). Augmented Reality has also proven itself capable of improving learning outcomes through increased engagement and interactivity[21].

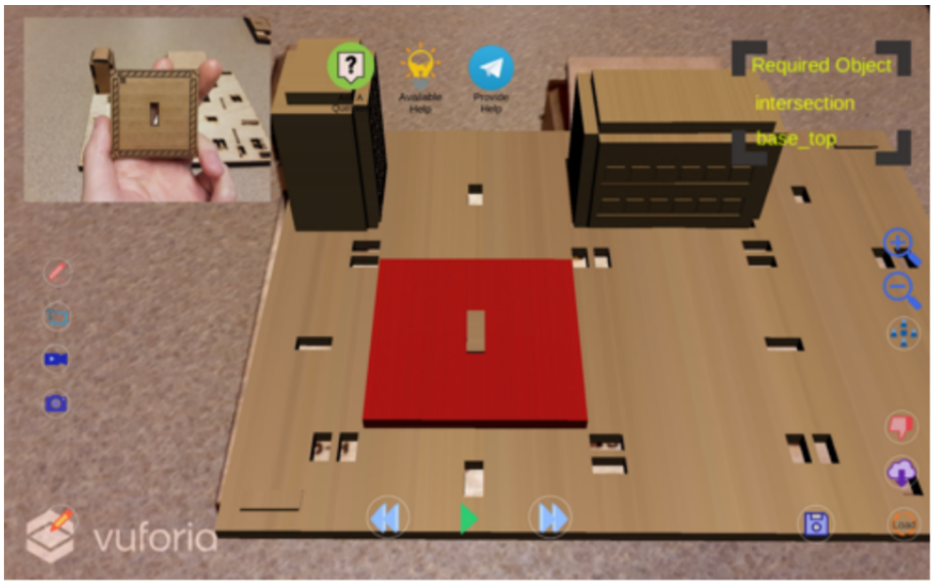

Figure 1.2. Meta-AR-App: An Authoring Platform for Collaborative Augmented Reality in STEM Classrooms

Virtual learning can be difficult to manage for instructors which are responsible for attending to the needs of multiple students at the same time. The predicament exacerbates in a makerspace/hands-on learning activity when instructors are trying to diagnose problems and communicate instructions from a screen. In order to alleviate the burden placed on instructors, we built an initial helper in the form of an AI voice assistant built on top of the state-of-the-art natural language processing framework, which can provide hints to help solve issues with the work. Recent breakthrough in NLP technology (i.e. BERT[22]) has opened the door for development of NLP applications in various fields. Companies like IBM have been seeking to develop a mature chatbot that can correctly recognize intent from a question and provide corresponding answers. IBM Watson Assistant has already been deployed to real world classrooms and has achieved satisfactory result[23].

In our setup, students will first reach out to a virtual assistant for help while teleconsulting instructors will only take place if students are not satisfied with the aid or if they would like check-ins. We adopted the iterative design process when creating the AI voice assistant, which means that it can constantly evolve based on the question-answer pair fed into the system at run time. The ultimate goal is to create a designated assistant to share the burden of the instructors so they can focus on more important matters other than answering similar questions repetitively.

1.2 Motivation

The COVID-19 pandemic brings huge challenges in education across the world. Due to the closure of schools, more than 91% of students-approximately 1.6 billion- were unable to physically attend schools.[24] Most of the schools have to shift the programs to online platforms. As distance learning becomes a lifeline for education, the importance of distance learning has been revisited and lifted to an unprecedented level. While the traditional distance learning platforms did great job on lectures and teleconsulting, many issues and deficiencies emerged and brought our attention. In particular, instructors of project-based classes and makerspaces which now have to be delivered with teleconferencing tools (e.g. Zoom, Teams, Skype,etc) have faced major difficulties in reproducing equivalent learning experience as in the pre-pandemic era.

Unlike attending a regular lecture class, students in a project-based class or a makerspace often seek the instructor's intervention, support, and troubleshooting. Most current distance learning platforms only provide video conference functions with a fixed field of view, which is considered to satisfy general purpose remote learning scenarios such as lecture and Q&A sessions. However, such video conference tools didn't take into account the aforementioned unique features and requirements of project-based class. Due to the lack of effective methods navigating through the space of the virtual makerspace, an instructor often wastes tons of time collaborating with the student to trudge to the right position in order to view the components; then spends another century asking the student to adjust the camera for troubleshooting; finally makes great effort translating the correct spatial operations into words without any spatial hints for the student. The process mentioned above is just a troubleshooting case for one student, considering the normal size of a makerspace lab, it is impossible to make a smooth transition from an in-person makerspace to a virtual makerspace using existing distance learning methods.

Therefore, we believe that a new distance learning method for virtual makerspace is an urgent need for both instructors and students. In this thesis we aimed to solve the challenges from the virtual makerspace and came up with an educational toolkit which proved to be intuitive and effective for both instructors and students to work together in a virtual makerspace. We hope this thesis can be regarded as a valuable reference for the future improvement of educational quality in the field of distance learning.

1.3 Research Goals

Beneath the educational challenges from the COVID-19 pandemic hides the great opportunity of a revolution. The massive deployment of distance learning has made this educational method popular and acceptable among instructors and students, suggesting that distance learning will become one of the most prevalent educational methods in the near future. To overcome the deficiencies in distance makerspaces and project-based classes learning, this thesis has established following goals-

a) Understanding the challenges of holding a virtual makerspace for distance learning.

b) Come up with a distance learning architecture specifically designed for virtual makerspace to help the instructor and the student better focus on areas of interest inside the workspace.

c) Implement the system for creating the new virtual makerspace experiences.

d) Design and perform a user study which compares current remote learning techniques for virtual makerspace vs. an implementation of our system to validate the efficiency of the system.

Aside from our contributions, we will investigate into the effects of our system implementation into a distant makerspace environment. Our work is targeted towards undergraduate students who seek a makerspace-based instruction to mix creativity and technology learning. While we hypothesize that physical embodiment will result in an increase in student engagement [13]; more importantly, we raise another question: **Q1: To what extend does the use of our system lead to an improvement in students' key competencies and user experiences?**. If our robotic system allows learners to meet key competences, we ask another question from the point of view of the instructor: **Q2: To what extend does the use of our system allow the instructor to offer more on-point instruction and at a higher level during problem-solving?**. Finally, if both questions result favorably, we wonder how an improvement in learning can influence in the interactions between instructors and students, in the form of the following question: **Q3: To what the extend does the use of our system increase instructor's credibility and promote students' compliance?**. Our work will explore all these research questions. This paper aims to advance our understanding of hands-on distance learning, which is becoming increasingly important in today's society.

1.4 Scope

For the purpose of system development, we have limited the scope in the following respects-

a) Scale of the virtual makerspace: Although the scale of a makerspace can be varied from table scale to building scale. In this thesis the scale of the makerspace was limited to table scale, which is considered to be both the most common makerspace scale in school and the only possible virtual makerspace scale at home.

b) Size of the classroom: Under current pandemic, it is difficult to gather enough users to simulate a normal size makerspace classroom-approximately 20 students- setup. Also the research budget would be overwhelmed by the cost of building 20 sets of the toolkit. Therefore the size of the classroom was limited to 3 students and 1 instructor, which can still provide significant data to validate the system.

c) Content of the virtual makerspace: Among various choices of the content for the makerspace, we decided to use a circuitry lab which is one of the most typical makerspace labs to test our system. A circuitry lab often includes both theoretical concepts learning and hands-on experience, which is ideal for evaluating the performance of the system.

1.5 Thesis Overview and Organization

This thesis is organized by first introducing the current related works in the area of educational robots, and applications of mixed reality in the educational field. Then the engineering method we used to develop the system is explained. Next we present the elicitation process and figure out the system requirements as the result of the elicitation study. Based on the system requirements, we went through conceptual synthesis and came up with the system architecture design. The detailed design and implementation of each module of the system is demonstrated in the following chapter. Additionally, the process and result of the validation test and further discussion are presented. Finally, the contributions, limitations, and future direction of the presented work of this thesis are listed and discussed. A chapter-wise organization is provided below-

Chapter 1: Introduction

Chapter 2: Related Work

Chapter 3: System Engineering Methods

Chapter 4: System Requirements

Chapter 5: Conceptual Synthesis and System Architecture Design

Chapter 6: Detailed Design and System Integration

Chapter 7: System Validation and Results

Chapter 8: Contributions, Limitations, and Future of Work

2. RELATED WORK

2.1 Natural Language Processing and Its applications

Natural language processing refers to the branch of computer science—and more specifically, the branch of artificial intelligence or AI—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can[25]. Being a very active area of research and development[26], it has applications in numerous fields[27].

In late 2018, a significant step forward in natural language processing was taken with the introduction of the BERT[9], a Transformer deep learning architecture that has transformed the landscape of NLP. When BERT was published, it achieved state-of-the-art performance on a number of natural language understanding tasks. Google Search has started adopting BERT models for search queries on December 9, 2019 [28], and in October 2020, almost every single English-based query was processed by BERT[29].

Many companies have provided publicly accessible frameworks for training and deploying NLP models including 'wit.ai'[30] from facebook, 'LUIS'[31]from Azure, and 'Jarvis'[32] from Nvidia. All of these frameworks provide an easy interface and quick learning APIs which significantly lower the setup effort and learning curve for end users. With the help of these frameworks, people with little prior knowledge of NLP can apply it in their respective fields.

In the field of education, natural language processing is primarily used to build a Question Answering (QA) system. It works by retrieving the intent from a student's question and automatically provides answers either from a pre-structured database or collection of online search results. IBM has been seeking to develop a mature chatbot that can correctly recognize intent from a question and provide corresponding answers. IBM Watson Assistant has already been deployed to real world classrooms and has achieved satisfactory results[23]. Additionally language assessment technology has been utilized to build an automated scoring tool in writing classes and proves to have shown remarkable promise[33]. Nature language dialogue generation tool has been developed to create voice-interfaces that facilitate storytelling activity with children[34].

2.2 Robot in Education

2.2.1 Social Robot

Social robots are physical agents that interact with humans by following social roles and behaviors attached to those roles [35]. Social robots for education are intended for delivery of learning experiences through social interactions with the students. In this context, robots for education have been mainly used in three areas: (a) language acquisition and development, (b) science and mathematics education, and (c) technology and computer programming [36]. Past work has demonstrated the benefits of using a robot in the classroom. Perhaps the most common use of an educational robot has been robot tutoring[15], [37] for teaching a second language [38]. Robot tutoring for second language acquisition, has shown cognitive gains among children, through storytelling and adaptation of the robot to the child's knowledge level [39]–[42]. Robotics for science and mathematics have included gaming using adaptive exercises [43] and teaching equations with the robot addressing an entire group of learners [44]. Technical education with robots typically uses the robot as the learning tool, instead of tutoring [45], [46]. These lesson plans involve introduction to programming the robot and hands-on activities that lead to tinkering and making the robot work [45], [47], [48]. Some of the most commonly used commercial robots adapted for educational interventions have been: NAO [49], RoboThespian [50], Bioloid [51], BAXTER [52], Darwin [53], TIRO [54], Keepon [55], LEGO Mindstorms NXT [56]. The robot as a tutor can provide learning support through multiple hints, visual cues, tutorials, and help with troubleshooting problems. In some cases, the robot is used as the medium to deliver the lesson to the class. Thus, the interactions between the robot and the students are limited and meant to capture the students' attention and encourage engagement with the subject [57]. The robot typically delivers the lesson from one to many students [58], [59]. However, the most frequently used tutoring robots for education allow teaching students individually, in which learning outcomes are highly dependent on the interactions between the robot and the student [60]. The problems with using a robot as an individual tutor in the previously mentioned work, include the lack of scalability, portability, and cost. In our work, our toolkit provides a minimalist design that keeps the cost low and allows for easy installation. Similarly, the autonomous aspect of the robot will solve the scalability issue by allowing an individual experience with the robot, open to improvement.

Figure 2.1. Nao Robot

2.2.2 Teleconsulting Robot

While social robots are used for physical interactions and communication, telepresence robots are embodied agents that enable the user to videoconference while on a moving platform from a distant location [35]. The user has remote control of the mobility and behavior of the robot, and communicates by using the robot as a delivery medium. Telepresence has been used to promote engagement and provide immersion to participants regardless of distance [61]. While the use of telepresence robots has been mainly used in the context of bringing distance students into a physical classroom, teleconsulting robots can be used to bring the instructor into the student's workspace [62]. New technologies (e.g., robotics, AR) can expand the consultation experience for students and make it easier for instructors to diagnose the problem. The benefits of using teleconsultation range from an increase in support and mentoring from the consultant to the consultee [63], an increase in access to rural youth [64], and an increase in frequency and quality of the interactions [65]. School-based teleconsultation has been successful in disruptive behavior consultation through videoconferencing. Further, teleconsultation was rated by the teachers as being just as an acceptable delivery medium as traditional face-to-face consultation [62], [64]. While teleconsultation has been an effective medium for instructors, studies have used them in static platforms (e.g., Kubi [66]) that do not mimic real-world interaction, in which students and teachers move frequently in their environment [11]. This is a significant limitation, because the quality of teleconsultation can be hindered if the consultant is unable to follow along and view the student's work. Thus, telepresence robots (e.g., [67], [68]) may be the best solution to the static nature of typical teleconsultation. In our work, we will be using a desktop-based teleconsulting robot to evaluate the quality of teaching in the context of an augmented makerspace.

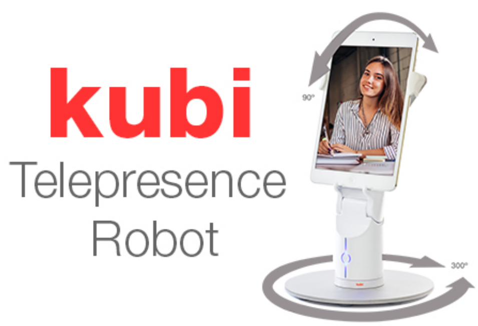

Figure 2.2. Kubi Teleconsult Robot

2.3 Mixed Reality in Education

Mixed reality (MR) is the merging of real and virtual worlds to produce new environments and visualizations, where physical and digital objects co-exist and interact in real time. An MR experience is one where the user is placed in an interactive setting that is either real with virtual asset augmentation (augmented reality), or virtual with real-world augmentation (augmented virtuality). Figure depicts these discrete points in the MR continuum.Thanks to development of mixed reality hardware (Oculus[69], HoloLens[70] and etc) as well as the user-friendly mixed reality content creation platform(Vuforia[71], Unity[72] and etc) the the quick the education community has witnessed a wide adoption of mixed reality applications for learning purpose. VR(Virtual Reality) and AR(Augmented Reality) are the two most common forms of mixed reality that have been applied in an educational environment. The former immerses users into an artificial world, while the later allows virtual objects to be superimposed on the real world, each of them of its unique benefit and suits specific scenarios.

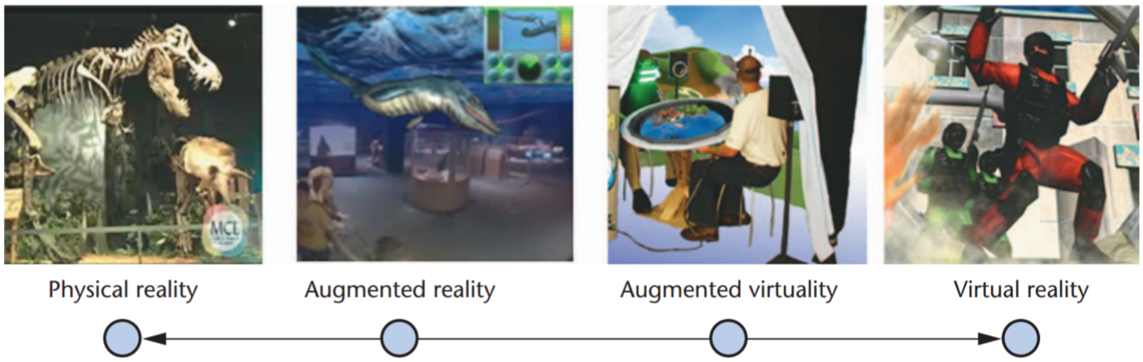

Figure 2.3. Example of Physical Reality, Augmented Reality, Augmented Virtuality, and Virtual Reality

The phone-based augmented reality is the most prevalent form of augmented reality in a classroom[73]. Students can simply point the phone onto a certain object with marker attached and augmented contents will be displayed on top of it. In this paradigm, instructors can deliver extra learning material into different media forms(video/text/image/3D animation) and superimpose them on the targeted physical objects[74]. In this way students can easily associate what they learn with what they see in the real world. Previous studies have found out many advantages of AR compared to traditional learning media[73]. The advantages of AR include positive learning outcomes, increased engagement and enjoyment, and most importantly collaboration between students and teachers. AR-facilitated collaboration in a classroom ranges from synchronous collaboration where students share the physical object[75] to asynchronous collaboration where AR contents are used to communicate ideas[21]. AR also has potential benefits for students' spatial ability, the relationship between application design and user experience[76]. However, there are also challenges of using AR in education including difficulty learning to use AR technology, and students experience cognitive overload[77]. However, those issues can be mitigated through careful system design such as implementing scaffolding mechanisms[78].

Thanks to the development of more affordable VR devices such as Oculus Quest[69], the adoption of VR in the field of education has been steadily growing. In contrast to augmented reality(AR), virtual reality offers an entirely immersive virtual experience that is free of the constraints of the physical environment. The unique affordances of immersive technology, VR has provided new opportunities for visualizing and interacting with abstract learning materials[79]. It is also useful in recreating simulations for situations that would otherwise be hazardous to access in real life[80]. Furthermore, VR is used for teaching numerous types of knowledge. including but not limited to procedural practical, and declarative knowledge[81]. The rising interest in seeking application of VR ineducation has inspired new research initiatives that aims to investigate the value of educational immersive technology. Studies have found out that when compared to traditional media, VR-based instruction is more effective in achieving multiple learning outcomes including self-efficacy, perceived learning and motivation[82]. On the other hand, some researchers found out that students learning via VR may score lower in terms of retention of information which implies VR may tax cognitive resources of learners heavily. Thus careful rethinking of the instructional design of educational VR is necessary before introducing it to an actual classroom.

3. ENGINEERING METHOD

A system is defined as a combination of elements that function together to produce the capability to meet a purpose. The elements stand for every hardware, software, equipment, personnel, facilities, processes, and procedures needed for such a purpose.[83] Thus it can be seen that the development of a system is usually more complicated than the development of a single product. It is essential that a comprehensive system engineering method should be followed to engineer a complex system.

System engineering is regarded as a robust method in the process of system design, creation, and operation. This method involves identification of the goals, generation of various design concepts, evaluation of different alternatives, implementation of the system, validation of the functionality of the implementation, and the assessment of the performance of the system.[84]

In this Thesis, we followed the system engineering method in the process of goal identification, system design, system implementation and system assessment. An overview of the detailed system engineering method is presented in this chapter.

3.1 Five Stage of System Engineering

The process of system engineering usually follows a top-down approach. In this thesis the whole process was divided into the following five stages which were applied sequentially and iteratively[85]-

a) Task Definition

b) Concept Generation

c) Design

d) Implementation

e) Validation

These five stages are presented in the flow chart in fig 3.1 Each Stage will be briefly illustrated.

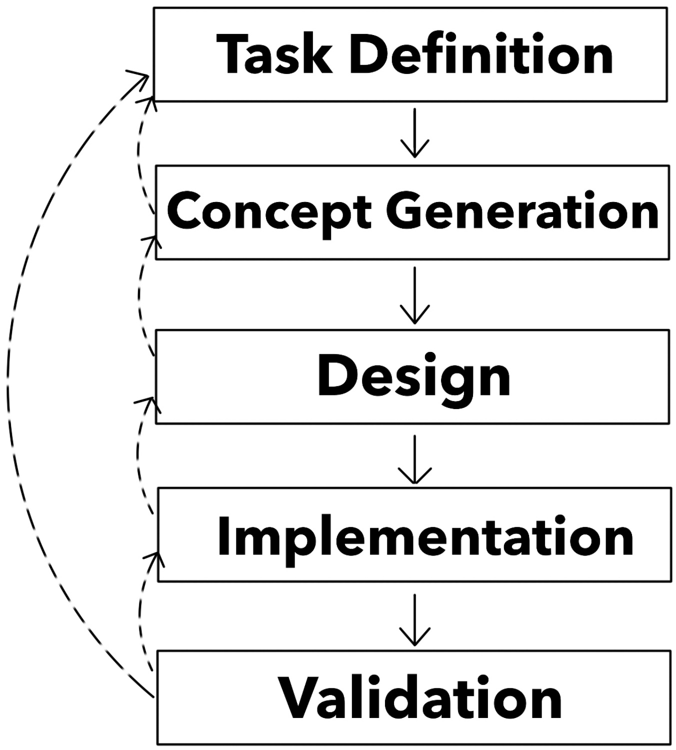

Figure 3.1. Five-Stage Flow of System Design

Task Definition stage involves problem definition and requirements specification. At this stage the problem that needs to be solved is identified. And the requirements of the system are filtered and specified by conducting elicitation study.

Conceptual stage is the next stage after identifying the general goals and requirements of the system. The requirements acquired from the task definition stage are compiled and translated into system functions at this stage. The essential functions of the system and the overall structure are first established and then splitted into functional individual modules-or subsystem-. Functions with more specific detail are then generated for each module. Possible solutions will be explored, assessed, and selected and then form the system architecture .

Design stage is the critical stage of system development. The translation from the function requirements to the actual hardware and software design take place at the design stage. The design of the hardware includes the design of the mechanism, selection of the components, and the approach of the fabrication. The design of the software includes the design of the software architecture, the design of the control hierarchy, the selection of software develop platforms.

Implementation stage relates to development of the system fabrication. Each subsystem will be developed following the solutions generated from the design stage. The subsystem will then be brought together forming the complete system. A functional prototype will be presented at the end of the implementation stage.

Validation stage is the final stage of system engineering. After building a functional system, it is critical to test the validity of the system. Although the system may perform perfectly in the aspect of meeting the functional requirements, it could still have undetected errors or deficiencies which may lead to the failure in the aspect of meeting the system requirements-or the design goal-. Therefore, a series of tests are needed to be carried out to gather essential data for analysis in order to evaluate the validity of the system.

3.2 The V-Model of System Engineering

The V-Model is one of the most widely used models in system engineering. The V-Model separates the five stages of the system engineering process into more detailed activities that are hierarchically structured. Figure 3.2 shows the architecture of the V-Model.

There are two streams of the V-Model. The left steam includes the activities of system definition and functional decomposition. It follows a top-down approach, which starts from the higher level architecture to the subsystem detailed design. After going through the process of system implementation, the right stream, which contains the evaluations related to system verification and system validation, changes the direction to bottom-up, and tests from the subsystem's functionality to the whole system's validity. The V-Model not only navigates the whole engineering process clearly from the very beginning definition and decomposition stage to the final verification and validation stage of the system but also establishes the relation between stages at the same system level. The two-way arrows demonstrate the connection between each activity of the definition & decomposition stream and each evaluation from the verification & validation stream. For example, the subsystem test is connected with

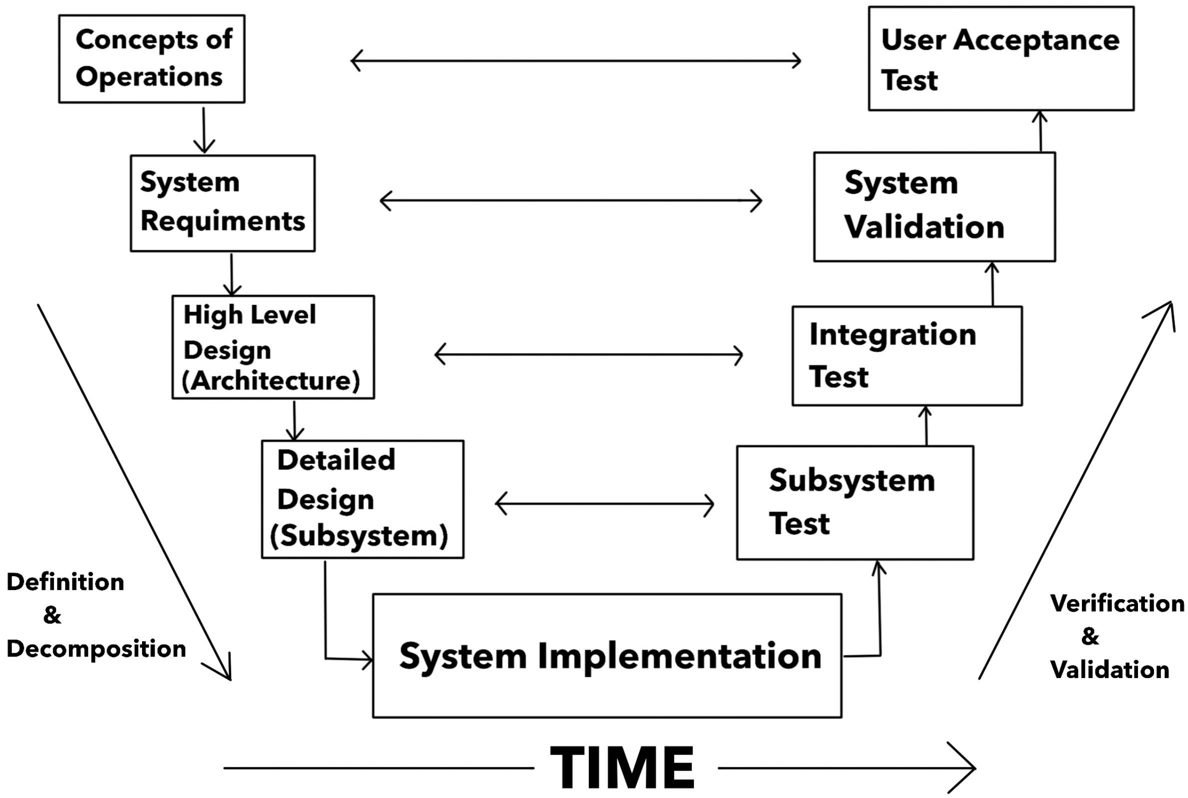

Figure 3.2. V-Model of System Engineering

the detailed design-or subsystem design-, which means any problem found in the subsystem test can be traced back to the detailed design stage. Most of the deficiency found in the right stream of the V-Model can be improved by such iteration.

The structure of this thesis matches the work flow of the V-Model. Chapter 4 is related to System Requirement, Chapter 5 includes High Level & Low Level Design, Chapter 6 illustrates System implementation, and Chapter 7 presents the Verification and Validation Test.

4. SYSTEM REQUIREMENTS

4.1 Process of Elicitation

Since STEM distance learning in virtual makerspaces presents its unique set of challenges, we wanted to find out how an AR compatible robotics toolkit would be an appropriate solution to this context. In order to understand the need of students and instructors in remote makerspaces activities, we proceeded to conduct a elicitation study in order to derive some design guidelines.

We interviewed 4 instructors and 10 students who had participated in previous full-day sessions of an online makerspace over a 3-day period, in which the participants took part in engineering activities and learned basic electrical circuits. Two of the instructors had more than 2 years of experience with physical makerspaces and workshops, and two had volunteered for their first virtual makerspace. Instructors were encouraged to reflect on their experiences by responding to the semi-structured interview. We conducted separate interviews with each instructor over an 1-hour period. Interviews with students were surveys completed voluntarily.

4.2 Findings

During the interview, students have expressed appreciation and contentment for their instructors and their quick adaptation to the new format of distance learning. Overall, students and instructors showed a positive attitude towards virtual makerspaces; however, this enthusiasm was mostly related to the opportunity of realizing the activity at all, instead of getting cancelled, and of using technology in a meaningful way. Also, they recognized several issues and shortcomings associated with these new interactions. We have listed a summary of the feedback from the participants of our interview as below-

(R1) Need for teleconsultation for proximal demonstration. Students reported missing aspects of physical makerspaces when taught in a remote setup. More specifically, they felt a lack of demos "on-the-fly" throughout the process. Face-to-face sessions meant that the instructor has the ability to walk toward their workspace and sometimes, quickly shows a student a short example of something they did not understand or instructions from which they fell behind. This provided encouragement and support for students to continue working on the material. Similarly, instructors reported that the ability to diagnose a problem depended on them being able to approach students and analyze what was wrong with their work.

(R2) Need for reshaping the landscape. An instructor pointed out that screens can be limiting and lack 3D perception of what the instructions look like. When the 3D content is projected onto a 2D screen, some level of visual information inevitably is lost during the process. There was a consensus among instructors that they see the future of distance makerspaces to provide learners with a more immersive interface, such as mixed or virtual reality. With virtual reality, they can perceive the remote environment in a realistic 3D space.

(R3) Need for reshaping the hardware. Instructors and students all reported issues with videoconferencing when instructors wanted to hold components or demos towards the camera, and when students needed to show their progress and request help with problem-solving. Our technology needs to solve the aforementioned issues in terms of facilitating zooming, centering, and adjusting the camera angles. More importantly, the hardware has to tilt, zoom, and move so that instructors and students can capture any area of interest within the workspace.

(R4) Need to relieve the instructor. Virtual learning can be difficult, especially when trying to diagnose problems and communicate instructions from a screen. Instructors reported that about half the time of the session was allotted for debugging and troubleshooting of students' errors. In most of the cases, questions asked by the students are similar or stem from a common misunderstanding of certain concepts. In this case it would be a waste of time to ask instructors to perform the repetitive work and some AI enabled bot that can help students with commonly occurring problems will definitely help with the situation. In order to alleviate the burden placed on instructors, we should have an initial helper in the form of an AI voice assistant, which can provide hints to help solve issues with the work; thus, teleconsulting instructors take place if students are not satisfied with the aid or if they would like check-ins.

(R5) Need for a scalable architecture. Remote learning platforms which are hosted on the internet are inherently scalable and our platform using the robotic toolkit should not be an exception. It needs to support cloud capabilities to enable multiple students to simultaneously participate in an augmented makerspace. Students reported that much of the vibrancy of makerspaces is due to the community of makers to showcase and demo their work between makers of different skill levels. In other words, the more participants joining the platform, the better it would be.

4.3 System Requirements

From the results of the elicitation study, we conclude the following high-level requirements for our new system of distance learning for virtual makerspace-

a) **Efficiency**: The system should increase the efficiency of the learning of a virtual makerspace significantly.

b) **Presence**: The system should have the ability to emphasize the presence of the Instructor in order to optimize the engagement of the students.

c) **Intuitivity**: The system should have intuitive interfaces and smooth learning curves for both instructor and student.

d) **Economical**: The extra cost for the system deployment should be minimized.

e) **Scalability**: The system should be able to deploy in classroom scale and adapt to various types of makerspace with little modification.

5. CONCEPTUAL SYNTHESIS AND SYSTEM ARCHITECTURE DESIGN

5.1 Functional Analysis

The high level system requirements from chapter 4 are mostly abstract characteristics of the desired system, which, according to the system engineering process mentioned in chapter 3, need to be translated into high level functions of the system. We came up with six essential functions to satisfied the high level system requirements-

a) **Voice Communication**: The system should first support voice communication between the instructor and the student since voice communication function is the basis of a distance learning environment.

b) **Spatial Navigation**: The system should give the instructor the ability to navigate in the makerspace to improve the efficiency of troubleshooting.

c) **Spatial Instruction**: The system should be able to display spatial content which guarantees a smoother information transportation between the instructor and the student. For example notations or drawings in 3D space which may highlight the position of specific components.

d) **Broadcasting**: The system should allow the instructor to perform self-demonstration to the student.

e) **Workload Sharing**: The system should offer a method to provide hints to general repetitive questions autonomously in order to share the instructor's workload.

f) **Adaptivity to Various Makerspace**: The system should be able to import customized learning material in order to adapt different makerspace.

Based on these essential functions, conceptual system designs were generated. In the next section the best three concepts will be illustrated and the subsystem concepts with more detailed functions will be generated for each system concept. The decision will be made after evaluating each concepts' feasibility.

5.2 Concept Generation, Functional Decomposition and Decision

We applied general concept generation methods such as brainstorming, benchmarking, literature reviewing, etc, to generate various conceptual designs of the system. In this thesis we only present three best concepts and the conceptual subsystems of these concepts. The advantages and disadvantages are evaluated and the rationale for the final decision will be provided. All the conceptual systems share three same subsystems, which are 1) **Instructor Side Interface**, 2) **Student Side Interface**, 3) **AI Assistant**. The instructor side interface and the student side interface are the two fundamental subsystems that form a distance learning system. The introduction of the AI as the third subsystem aims at the workload sharing function. The main differences among each concept are mostly focused on the modules of the Instructor Side Interface and the Student Side Interface. We explored possible hardware combinations which may achieve the essential functions and the system requirements.

5.2.1 Virtual Makerspace in VR

The first concept is to translate the whole makerspace from the physical world into the virtual reality world. The rough architecture of the system is shown in figure 5.1, which provides a high level overview of the subsystems and their relation.

The most significant advantage of VR is that it can eliminate the space boundary between the instructor and the student by bringing them into the VR world. In the VR makerspace setup, the physical components are converted to their digital twins, and the whole makerspace is simulated in VR, the student and the instructor will manipulate the components with the controller to achieve operations such as connection, assembly etc.

In this concept, the instructor can have access to the student's makerspace and troubleshooting together with the student in real time. The instructor can tell the student the correct operation, add spatial instructions onto the virtual makerspace, and demonstrate by manipulating the student's component in real time, which mimics the real makerspace to a considerable degree. The essential functions are decomposed to the following detail functions for each module-

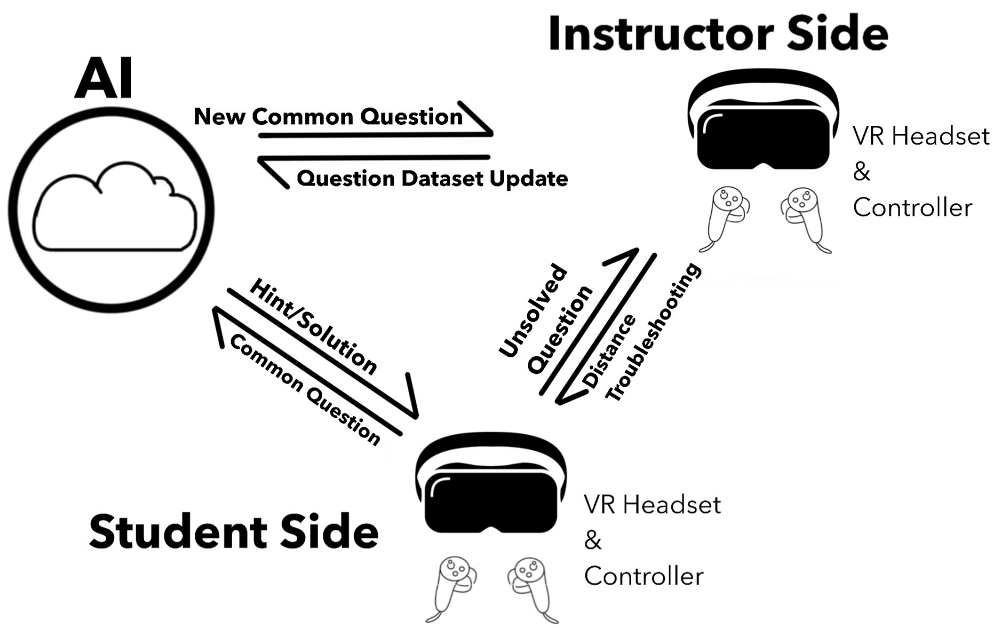

Figure 5.1. Concept of VR Setup

a) AI Assistant-

   i) Allow Instructor to customize the dataset of the common questions

   ii) Transform the student's voice input to question input and provide available solutions as output

   iii) Record the new questions from the student and feedback to the instructor

b) VR Headset(Student Side Interface)-

   i) Import the digital twin of the makerspace's components into the Interface

   ii) Allow the student to manipulate the digital twin components through the controller

   iii) Share the virtual reality makerspace with the instructor

   iv) Voice chat with the instructor

   v) Receive and display spatial instructions from the instructor

vi) Allow the instructor to manipulate the digital twin components for live demonstration

vii) Send the student's question to the AI assistant

viii) Receive solution from the AI assistant

c) VR Headset(Instructor Side Interface)-

i) Allow the instructor customize and import makerspace content

ii) View the student's virtual reality makerspace

iii) Voice chat with the student

iv) Allow the instructor to manipulate the digital twin components for live demonstration

v) Allow the instructor to customize spatial instructions

vi) Send spatial instructions to the student side interface

vii) Allow the instructor to customize the common question dataset and update the AI assistant

5.2.2 Augmented Makerspace with HoloLens

The second concept is to use the HoloLens to create an augmented makerspace. The rough architecture of the system is shown in figure 5.2, which provides a high level overview of the subsystems and their relation.

The idea is similar to the idea of the makerspace in VR-to give the instructor access to the digital twin of the student's makerspace- while the difference is the makerspace setup. In this setup, students learn through hands-on experience with physical objects such as electronic components. The HoloLens is capable of gathering spatial information of the makerspace and reconstructing the 3D space in the form of point cloud as the digital twin.

In this concept, the makerspace at the student side will be translated into point cloud and shared with the instructor in real time. And the instructor can directly generate spatial instructions which are visible to the student in real time.

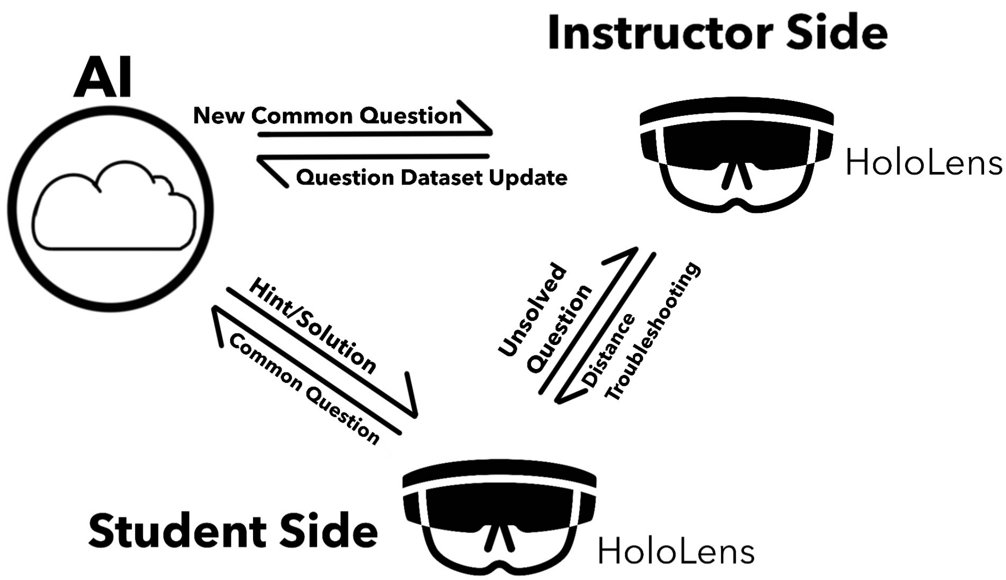

Figure 5.2. Concept of HoloLens Setup

The essential functions are decomposed to the following detail functions for each module-

a) AI Assistant-

  i) Allow Instructor to customize the dataset of the common questions

  ii) Transform the student's voice input to question input and provide available solutions as output

  iii) Record the new questions from the student and feedback to the instructor

b) HoloLens(Student Side Interface)-

  i) Send the digital twin of the makerspace to the instructor side device

  ii) Receive spatial instructions from the instructor and display as AR content on top of the makerspace

  iii) Receive the digital twin of the instructor's live demonstration and display in 3D space

  iv) Send the student's question to the AI assistant

  v) Receive solution from the AI assistant

c) HoloLens(Instructor Side Interface)-

   i) Receive the digital twin of the student's makerspace and display in 3D space

   ii) Voice chat with the student

   iii) Send digital twin of the instructor's makerspace to the student side device for live demonstration

   iv) Allow the instructor to customize spatial instructions

   v) Send spatial instructions to the student side interface

   vi) Allow the instructor to customize the common question dataset and update the AI assistant

5.2.3 Phone-Based Minibot

The third concept is to combine the concept of the educational robot and the concept of phone-based augmented reality. The system is designed as a teleconsulting robotic toolkit for creating AR makerspaces. The rough architecture of the system is shown in figure 5.3, which provides a high level overview of the subsystems and their relation.

Instead of choosing HMD for the Instructor side subsystem and student side system implementation, this concept uses conventional devices which are commonly available to instructors and students. The makerspace setup is the same as the setup of the HoloLens concept-students learn through hands-on experience with physical objects-. A minibot module is introduced into the student side subsystem. The combination of the minibot and the AR-capable smartphone provides the subsystem with the possibility to help the instructor navigate through the physical space and deliver spatial instructions similarly as the HMD device subsystem does. During the teleconsulting process, the student's smartphone will feed live video of the makerspace to the instructor side PC. The instructor can control the robot which carries the smartphone to move in the makerspace, observe the setup and diagnose the problem through the PC. In the troubleshooting process the instructor will provide spatial instruction by the built-in ARtoolkit of the smartphone.

The essential functions are decomposed to the following detail functions for each module-

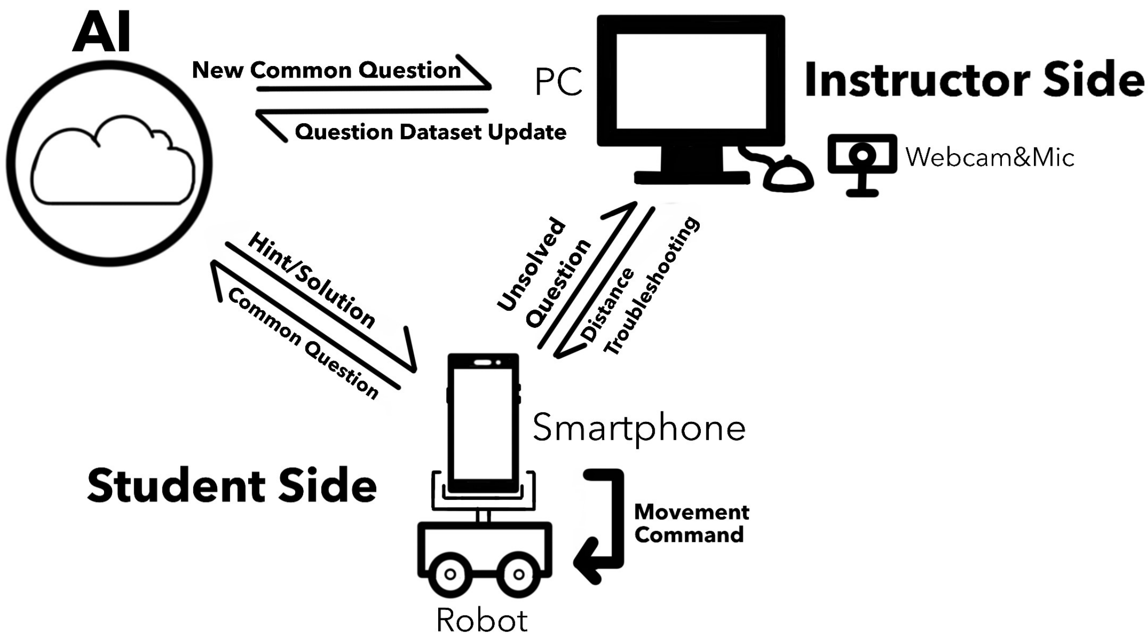

Figure 5.3. Concept of Phone-Based Minibot Setup

a) AI Assistant-

   i) Allow Instructor to customize the dataset of the common questions

   ii) Transform the student's voice input to question input and provide available solutions as output

   iii) Record the new questions from the student and feedback to the instructor

b) Robot(Student Side Interface)-

   i) A compact size which should not disturb the makerspace

   ii) High mobility to travel smoothly in the makerspace

   iii) Ability to change the view angle of the smartphone

   iv) Receive command from the smartphone for movement

   v) Durable battery life for a 1.5 hour makerspace learning

   vi) Boundary detection mechanism to prevent falling from the table

vii) Some level of autonomous navigation

c) Smartphone(Student Side Interface)-

i) Augmented Reality integration with 3D coordinate construction

ii) Send command to the robot for robot's movement

iii) Voice chat with the instructor

iv) Send live video from the camera to the instructor side interface

v) Receive the spatial instructions from the instructor and display on screen

vi) Receive command from the instructor and send to the robot

vii) Receive live demonstration video from the instructor side interface

viii) Send the student's question to the AI assistant

ix) Receive solution from the AI assistant

d) PC(Instructor Side Interface)-

i) Receive video feed from the student side interface

ii) Send command to the student side interface to operate the robot

iii) Voice chat with the student

iv) Send live video from the camera to the student side interface for live demonstration

v) Allow Instructor to customize spatial instructions

vi) Send spatial instructions to the student side interface

5.2.4 Concept Evaluation and Decision

To systematically decide the best concept, we performed the decision matrix evaluation. The five essential system requirements listed at the end of chapter 4 are set as the criteria in the decision matrix shown below in table 5.1, where the weight of each criteria is decided by the importance of the requirement. The most important requirement is the efficiency

Table 5.1. **Decision Matrix**

		Alternatives		
Criteria	weight	Virtual Reality	HoloLens	Phone-Based Minibot
Efficiency	0.25	5	4	4
Presence	0.15	5	5	4
Intuitivity	0.2	3	3	4
Economical	0.2	3	1	5
Scalability	0.2	2	4	4
Total Score		3.6	3.35	4.2

of the system, since our main goal is to develop an efficient system of distance learning for makerspace. The importance of intuitivity, economical, and scalability are considered to be the same in distance learning. The least important requirement is the presence enhancement of the instructor, while it is still a nontrivial factor of the design. The rationale of how the scores are gave are presented below-

Efficiency: We believe that virtual reality is the most efficient approach among these three concepts in the aspect of teleconsulting. Since in VR the instructor and the student are put together in a virtual makerspace which is the most similar to the physical makerspace, the efficiency of this concept is thought to be the closest to the real world makerspace efficiency. The other two concepts are both using AR technique to deliver instructions, which is not as efficient as the face-to-face approach from VR.

Presence: Both the VR and the HoloLens concepts provide a strong presence of the instructor since the instructor will be shown as a real time avatar in both concepts, while the minibot concept is not as humanoid as the other two concepts.

Intuitivity: In the aspect of intuitivity, the interaction methods draw the VR and HoloLens concepts back. The interaction in the VR makerspace is limited to the controller of the VR system, which makes the experience of the makerspace unrealistic compared to the physical world and leaves a big question mark to the actual learning outcome. The HoloLens concept does not have such an issue since it is based on physical makerspace. However, through our test we found out that the hand detection of the HoloLens, which is the main input method to the system, is lacking accuracy to generate desk-level detailed instructions such as 3D drawings etc. The input method of the HoloLens is not friendly to the instructor. The minibot concept, on the contrary, uses a conventional touch screen as input method for the student side, and mouse & keyboard for the instructor side, which are the input methods most people already get used to. This makes the minibot concept the most intuitive method among the three concepts.

Economical: The cost is one of the most critical factors for the deployment of the system. Since the budget of an educational institution is usually quite limited. The cost of an Oculus Quest,which is the most popular VR device today, is around $299. And the cost of a HoloLens is unbelievably high,which is $3500. The estimated cost of building the minibot is less than $150, which is half of the price of an Oculus and only five percent of the price of a HoloLens.

Scalability: Compared to the other two concepts, the VR concept is the hardest to scale. The reason is that every single component of each makerspace has to be transformed into a digital twin in VR, which requires a lot of effort in modeling and programming. While the other two approaches only require the common question & solution dataset of the makerspace for the AI assistant.

As the result shown in the decision matrix in Fig 5.4 , the minibot concept earned the highest score and was selected as the final design concept. The rationale of the decision is well explained as the highest score in the decision matrix means this concept fits the essential system requirements best among the alternatives. We give the name "RobotAR" to our system.

5.3 Subsystem Conceptual Design

In the previous section, we adopt the RobotAR concept as the basis of the system architecture design, which has been decomposed into three subsystems- the student side interface, the instructor side interface, and the AI assistant. The student side interface will be built up with two modules-a mini robot and a smartphone-. The instructor interface will be based on a PC. And the AI assistant will be developed by using one of the available open and free AI platforms. In this section, the detailed concepts of each module will be presented.

5.3.1 AI Assistant

The AI assistant subsystem will play a role as a helper-a TA to be more specifically speaking- of the instructor. The goal is to implement an AI assistant that can provide answers to the student for most of the common questions of the makerspace. The instructor should take care of the collection of the question and answer sets. Then the instructor will update and train the AI with the data set. In a physical makerspace, the Q&A session happens through the communication between the student and the TA. Therefore voice is the most natural medium for a student who wants to ask a question. We design the AI assistant in the form of a voice assistant which will be integrated in the student side interface. We chose 'Wit.ai' as the platform of our AI voice assistant. 'Wit.ai' is an open and free platform which provides developers with a powerful yet simple natural language processing API for building customized NLP applications.The AI will listen to the student's question, and display the answer if it is available. Furthermore, the AI should be able to record new questions plus the questions that are not properly answered as feedback to the instructor for future improvement.

5.3.2 Student Side Interface

The two modules that make up the student side interface are the smartphone and the robot. It needs to be clarified that the robot module we defined is more like an extension of the smartphone which offers mobility. And the smartphone plays multiple roles such as the vision sensor, the display, and the main processor of the robot. In fact, these two components are bundled together to function as a complete robot in this subsystem. Figure 5.4 shows the overall layout of the subsystem. As mentioned in chapter 2, the AR capable smartphone has the ability to detect the surface and retrieve its position and orientation data in 3D space, which is exactly the data we need to navigate the robot in the makerspace. Besides, the computational power of a smartphone today is quite advanced. Therefore, neither extra vision sensor nor extra high end processor are needed for the robot, which significantly narrows the overall cost of the system. In the workflow, the smartphone will first gather the space information and then send commands to the robot to travel to the destination.

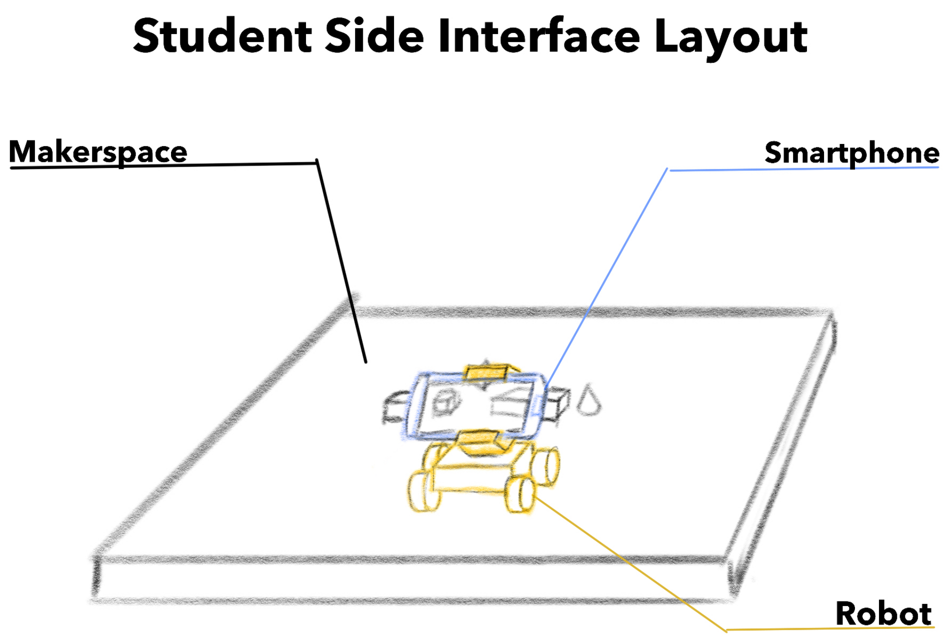

Figure 5.4. Concept Sketch of the Student Side Interface

Student Side Interface Setup and Overall Workflow

Based on our observation on a makerspace, the working area of the makerspace is usually the area in front of the student. Figure 5.5 shows the top view of a table-scale makerspace. The working area is highlighted in the figure. In a makerspace, students spend most of the time interacting with the components inside the working area. So it is undesirable to have the robot stay in any location inside the working area or the area between the working area and the student when the robot is not summoned by the student. In RobotAR we came up with the idea of **Standby Point and Engaging Point** to satisfy such a requirement. The idea is to develop an autonomous navigation mechanism so that the robot would not disturb the student from working in the working area. The student will need to set a standby point and an engaging point at the very beginning of the system setup. The typical setup of the standby point and the engaging point is shown in figure 5.5. The standby point is the location where the student feels comfortable to place the robot when he or she does not need teleconsulting. The engaging point is the location right in front of the working area, where the robot would be able to view the components and display the spatial instructions to the student. We defined four modes for this subsystem based on the idea of standby point and engaging point, namely setup mode, standby mode, AR animation mode, and teleconsult mode.

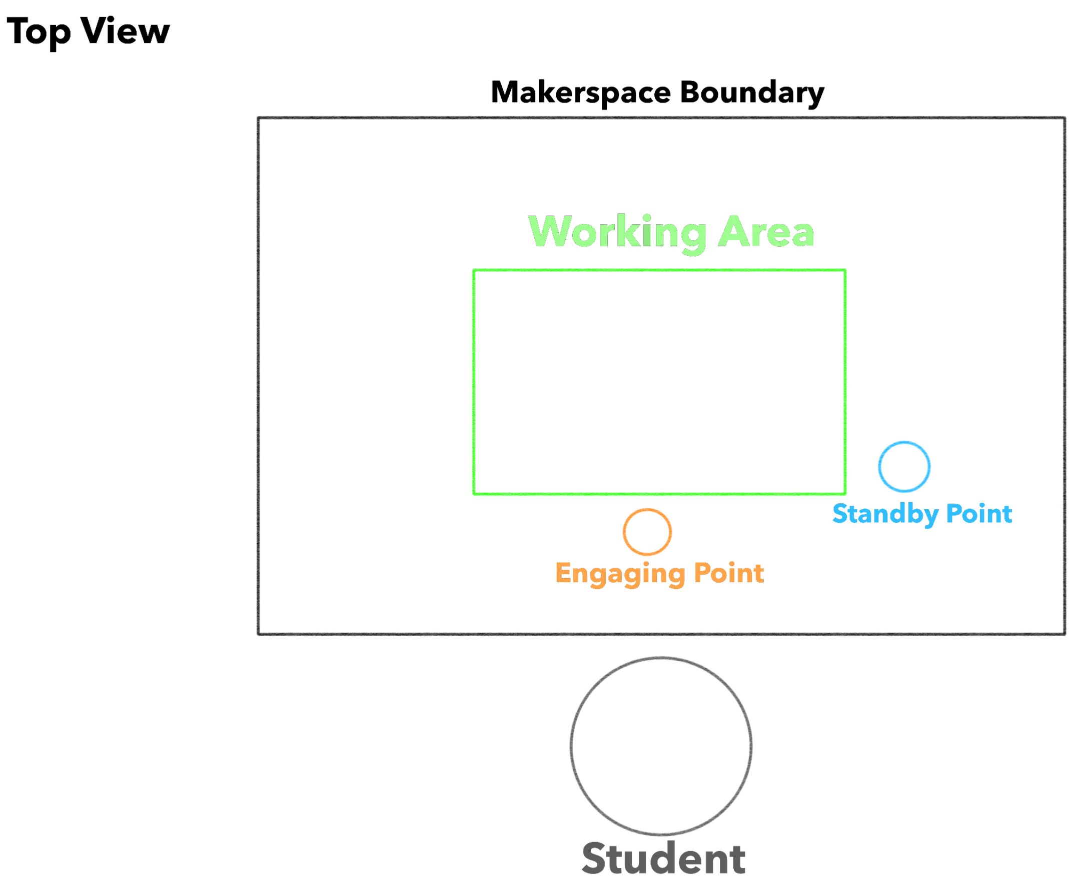

Figure 5.5. Top View of the Layout of a Makerspace

Setup Mode: The setup mode is the first step of the student side interface workflow. In setup mode, the mobility of the robot will be disabled and the student will hold the robot to define the standby point and the engaging point. After finishing the setup, the robot will be activated and travel to standby point by default.

Standby Mode: When the student is not in need of help, the robot moves itself aside to the standby point while remaining in the field of view. If a problem occurs, the student can ask the AI voice assistant directly or enter the teleconsult mode. In the first case, answers in texts with supportive media are displayed in the current scene. In the second, it moves to Engaging Point.

AR Animation Mode: Compared with traditional text-based or video-based tutorials, AR delivers a richer user experience and conveys spatial information which is important to hands-on tasks. The function of importing AR animation as supplemental material will be implemented in the system. The robot will move to the engaging point to play the AR animation and travel back to standby point when finished.

Teleconsult Mode: Once the robot enters teleconsult mode, it moves to the engaging point to assist the student. During this period, the robot behaves as an agent for the instructor. The instructor will manually move the robot, observe the components and add spatial instructions. The robot will return to standby point once the teleconsulting session terminates.

Robot

The process of building a robot from scratch is usually time consuming and accompanied with many unpredictable problems. So the best choice is to find a robot kit available in the market that meets our demands. After the process of benchmarking, we found the Yahboom Omniduino Smart Robot[86] to be the best candidate for our system. A Yahboom Omniduino Smart Robot is shown in figure 5.6 and its specifications are shown in table 5.2.

It is clear that the Yahboom Omniduino Smart Robot offers enough features our system needs. It has incredibly small dimensions of 11cm by 11cm, which is shorter than most of the smartphones. And the mecanum wheels provide significant agility which allows movements such as panning and yawing. And the battery life of the robot also satisfies the length of a normal makespace. And the cost of the unit is just $123[]. Based on the factors mentioned above, we chose the Yahboom Omniduino Smart Robot as the base of the robot module in the student side subsystem. As we can see from Fig 5.6 , the upper platform of the Yahboom

Figure 5.6. Yahboom Omniduino Smart Robot

Table 5.2. Specifications of Yahboom Omniduino

Microprocessor:ATmega328P	Programming language:C language
Battery:2*18650 battery (8.4V)	Working time:Around 1 hour and a half
Platform Rotation:Rotating up and down controlled by servo	Obstacle avoidance method: 5-way coverage obstacle avoidance sensor
Posture calibration;Gyroscope posture calibration	RGB Light: 4pcs full color RGB Light
Motor Drive: Single continuous output current 0.8A (Max:1.8A)	Remote control method: Mobile APP (WIFI Camera), PS2 handle
Motor model:N20 geared motor	Safety protection:Over current protection,low voltage protection, anti-reverse protection
Car tire:Mecanum wheel	Communication method:Serial communication
Product size:114mm*114mm*121mm	Product Weight:422g

Omniduino Smart Robot is equipped with a camera. In our system the smartphone should be mounted on top of the robot. Therefore the modification to the upper platform of the robot is necessary. Fig 5.7 shows the concept sketch of the robot after modification. The original camera will be replaced by a phone holder. The phone holder should be suitable for most of the smartphones in the market and be able to adjust the viewing angle by a servomotor. A connection between the smartphone and the robot will be established wirelessly.

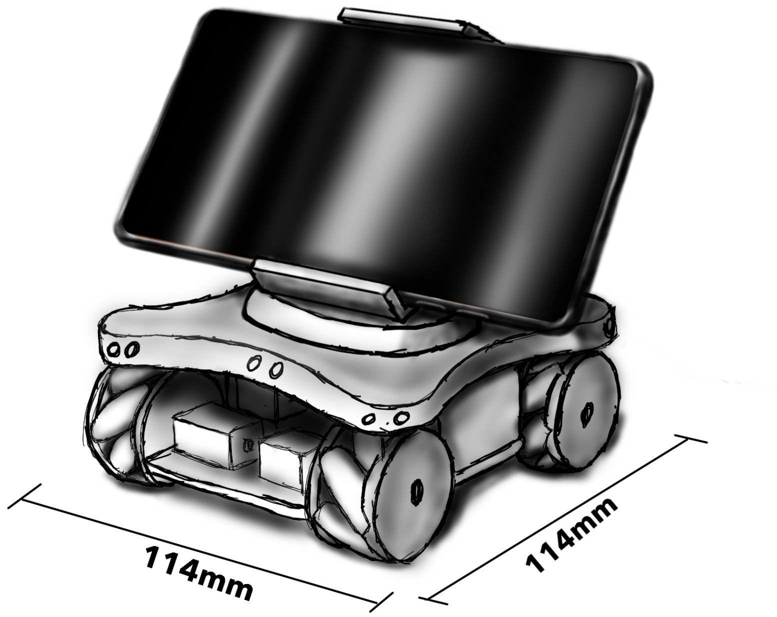

Figure 5.7. Concept Sketch of the Modified Robot

Smartphone Interface

The smartphone interface is the main approach by which the student interacts with the student side interface. Also the smartphone serves as the main processor of the robot. Figure 5.8 and figure 5.9 show two parts of the smartphone interface design.

The first part is the interface for setting the engaging point and the standby point of the robot. The smartphone will first scan the surface of the makerspace and generate a 3D coordinate system. The 3d surface of the makerspace is highlighted as a grid shown in figure 5.8. The student can then define the engaging point and the standby point by touching the desired location on the screen.

When the makerspace takes place, the smartphone will enter the main interface, which is shown in figure 5.9. The main interface contains three sections-a) Standby Mode Interface,

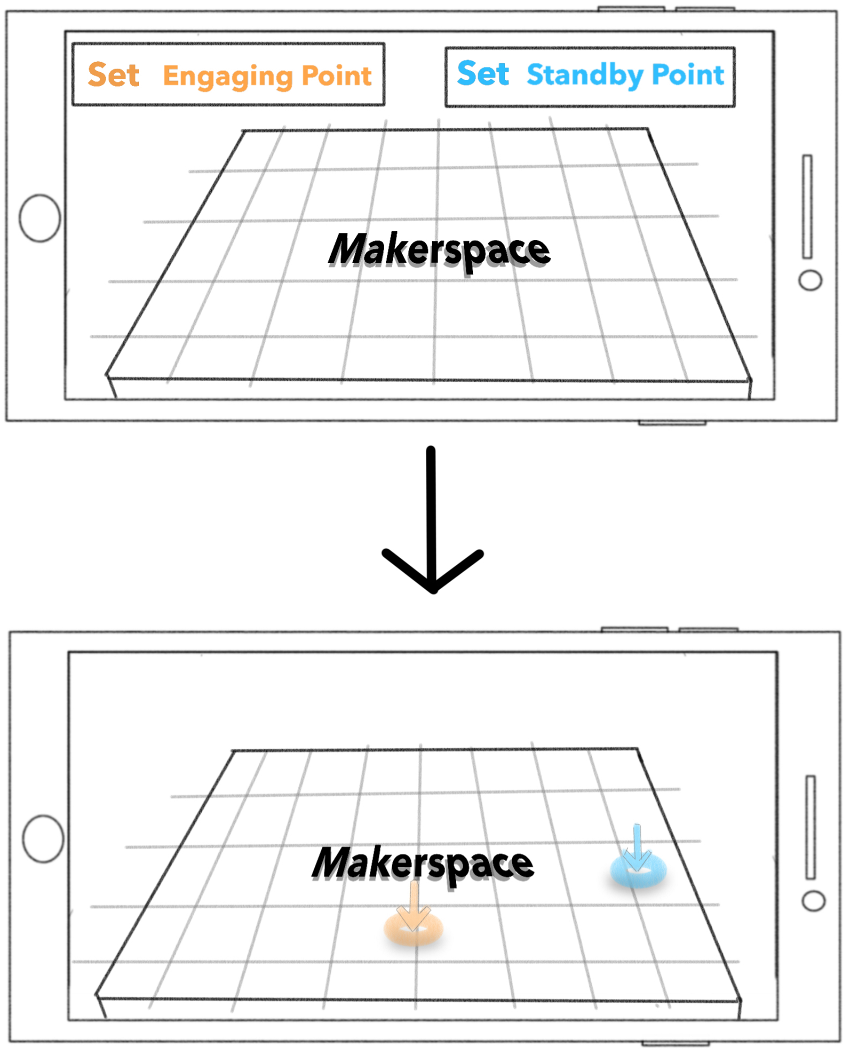

Figure 5.8. Concept Sketch of the Smartphone Interface pt.1

b)AI Assistant Answering Interface, c)AR Interface. The detail design of each section will be discussed below-

a) **Standby Mode Interface**: As mentioned previously, the robot will remain idle in standby mode. During this mode the interface will provide two buttons to the student- AI Voice Assistant Button and Teleconsulting Button. By holding the AI voice as-

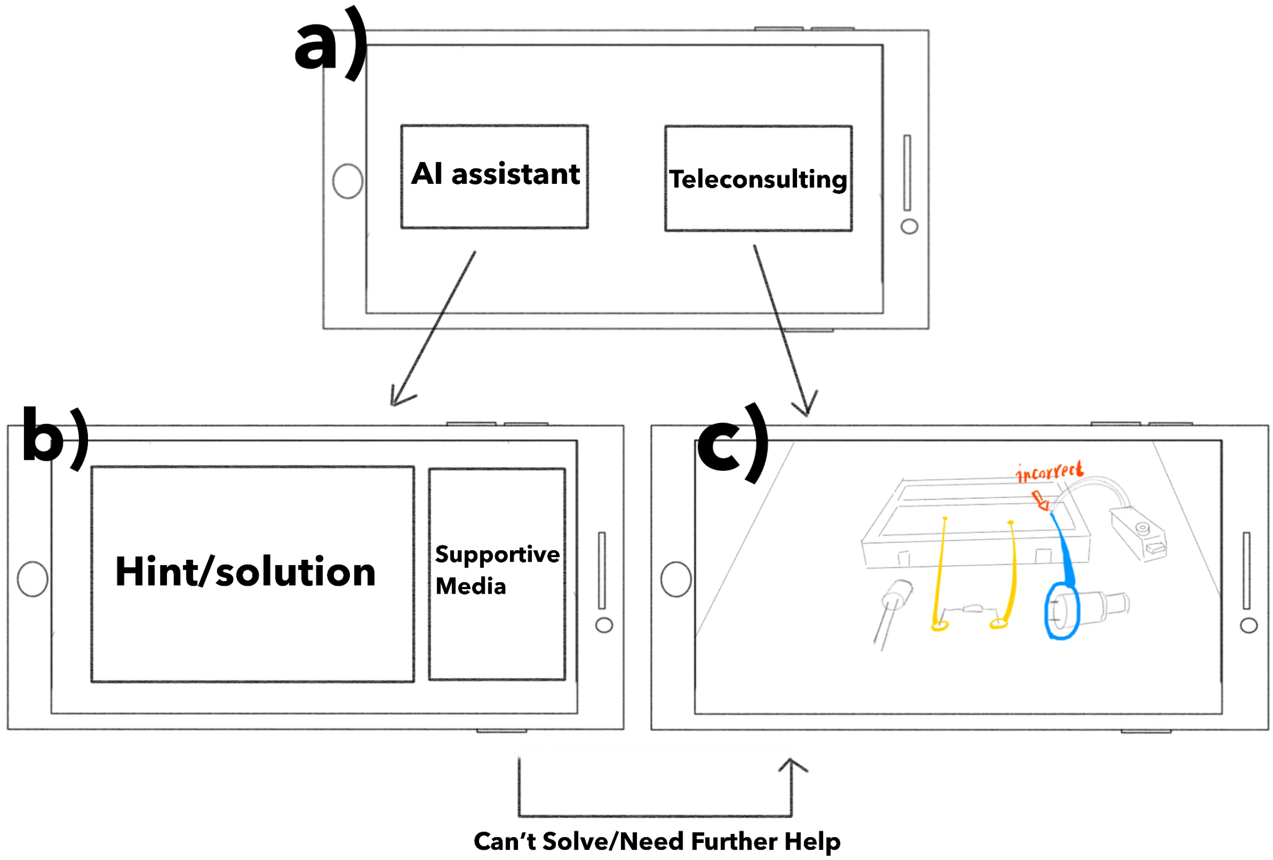

Figure 5.9. Concept Sketch of the Smartphone Interface pt.2

sistant button, the student can ask the question to the robot and if the question is recorded in the dataset, the smartphone will enter the AI Assistant Answering Interface. By touching the teleconsulting button, a teleconsulting request will be sent to the instructor through the network. Once the instructor accepts the request, the connection will be established between the instructor and the student. The robot will enter Teleconsult Mode and the smartphone will display the AR Interface.

b) **AI Assistant Answering Interface**: If the AI voice assistant can answer the question from the student, the smartphone will enter the AI Assistant Answering Interface. In this interface, there are two sections of the answer. As shown in b) of Fig 5.11, the left section is the text content of the answer, while the right section is for the display of extra supportive media such as video, figure or AR animation etc. If the student needs further help, a teleconsulting request will be sent to the instructor.

c) **AR Interface**: AR Interface activates once the robot arrives at the engaging point. At this interface, the front camera view of the makerspace will be displayed on the screen. At Teleconsult Mode, the view of the camera will be shared with the instructor in real time. The instructor has full control on the robot to navigate and observe inside the makerspace. The spatial instructions added by the instructor will be displayed on top of the components in the form of AR content on the screen. At AR Animation Mode, the pre-imported AR animation can be viewed by the student.

Instructor Side Interface

The instructor side interface is designed for the purpose of teleconsulting. Figure 5.10 and figure 5.11 show two panels of the interface. The first panel is the query list of the students. In this panel, all the participants of the maker space are listed with their student ID or name. The corresponding ID will be highlighted as shown in figure 5.10 when the student is requesting for teleconsult. The Instructor will connect to the student's robot and have access to the student's makerspace by clicking on the highlighted ID.

Once the connection has been made between the PC of the instructor side and the robot of the student side. The Instructor will have access to the teleconsult panel. There are 4 sections of the teleconsult panel as shown in figure 5.11, namely a) Display Window, b) Robot Control, c) Function Group, and d) Customization group. The detail of each section will be illustrate below-

a) **Display Window**: The Display Window will display the camera view of the robot. The instructor can interact with the Display Window through the mouse in order to place spatial instructions. The spatial instructions will be displayed on the Display Window as well as on the student side screen. The Display Window will switch to the webcam view from the instruction side when the instructor activates the live demonstration.

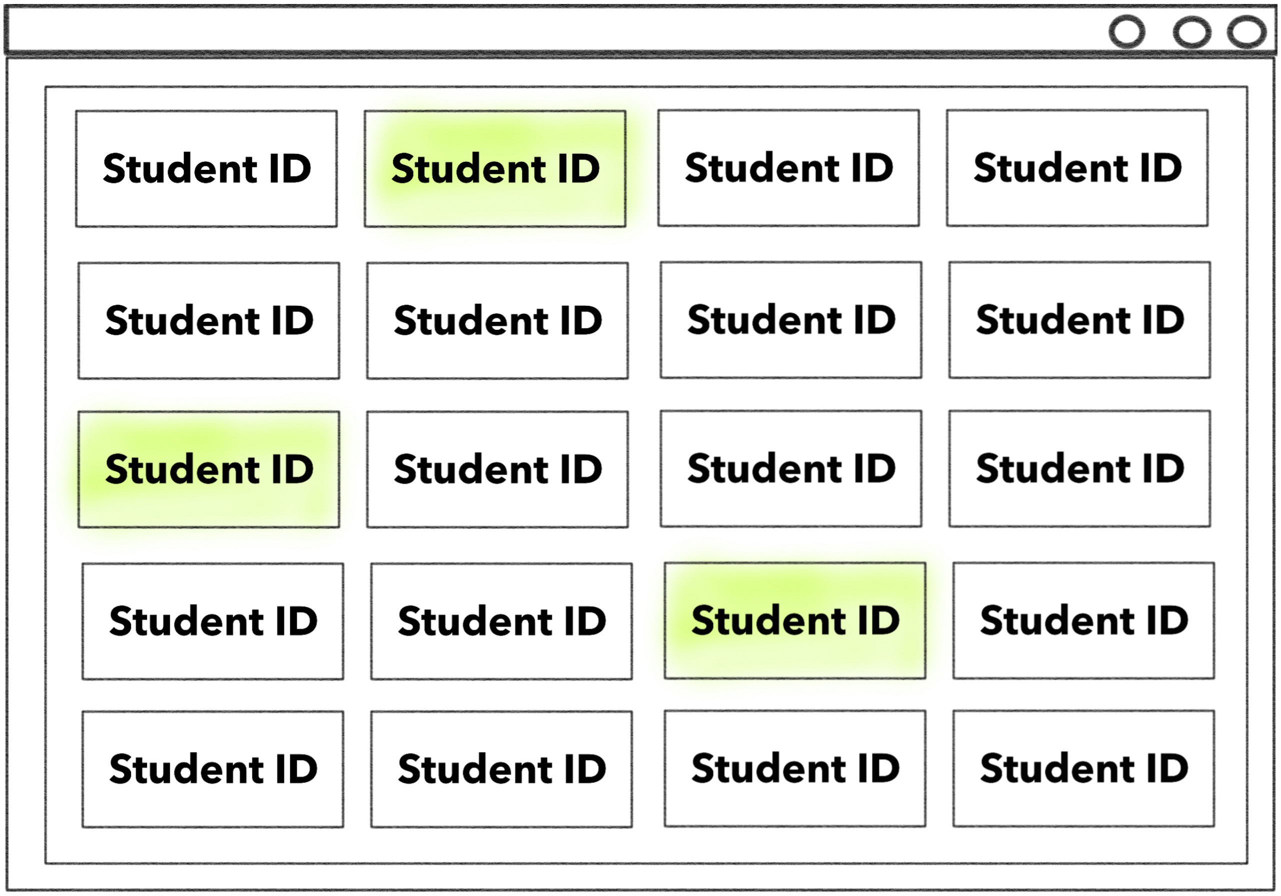

Figure 5.10. Concept Sketch of the Instructor Side Interface pt.1

b) **Robot Control**: The instructor can control the movement of the robot by clicking the buttons in the Robot Control section. The button group includes functions such as yawing clockwise and counterclockwise, panning horizontally, forward, backward, and adjusting the view angle.

c) **Function Group**: The Function Group provides different approaches to instructions, which are listed below-

   i) **3D Drawing**: the instructor can use this function to create freestyle 3D drawings in 3D space. This function can be used when the instructor wants to highlight specific connections or circle specific components.

   ii) **3D Text**: The instructor can type and place 3D text right next to the target in the makerspace, which lowers the possibility of misunderstanding by the student.

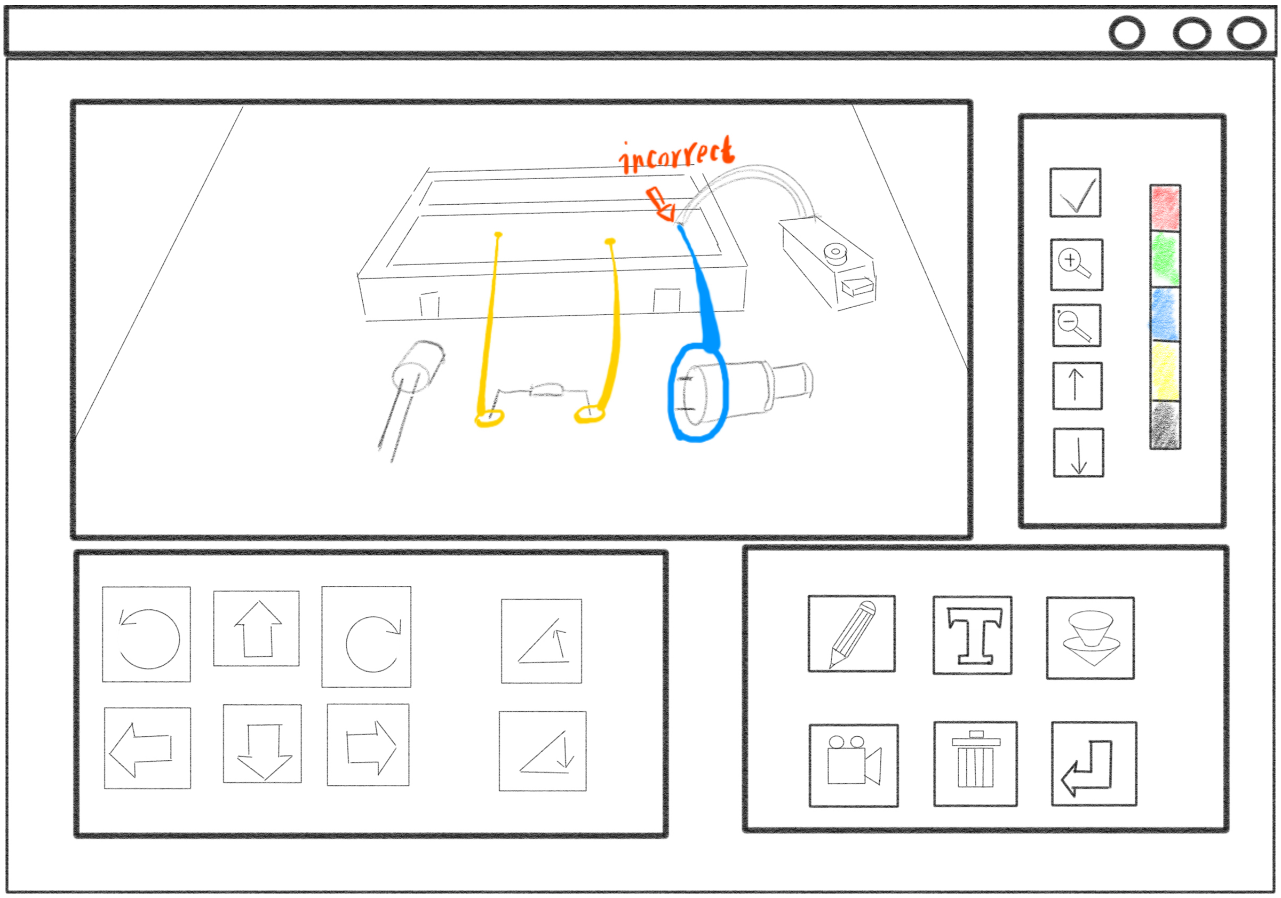

Figure 5.11. Concept Sketch of the Instructor Side Interface pt.2

iii) **3D Indicator**: 3D Indicator contains different predefined indicators such as arrow, star, and circle etc. The instructor can use these indicators to emphasize a location, a component, or a mistake.

iv) **Live Demonstration**: When live demonstration is necessary, the instructor can use this function to share the live demo with the student. During the live demonstration, the instructor can watch the webcam view through the Display Window. And the webcam view of the instructor side will be sent to the robot on the student side. The student can watch the live demonstration through the smartphone screen.

v) **Delete**: The instructor can delete the unwanted AR contend through this function.

vi) **End Teleconsult**: The instructor can click the button to terminate the teleconsult session and return to the student query list.

d) **Customization Group**: After the instructor places a spatial instruction, he or she can use the Customization Group to make modifications for the instruction. The possible customization options include but not limited to-scale change, location change, rotation adjustment, and color change etc. The instructor can click the check button to finish the customization.

Based on the designs of each subsystem, we moved forward into the system implementation stage. In the next chapter, the details of how we implemented each function that are discussed in this chapter will be presented. It should be noted that in the process of system engineering, there are always new designs that come out from the iteration loop between the design stage and implementation stage. The new designs and their implementation will also be present in the next chapter.

6. DETAILED DESIGN AND SYSTEM IMPLEMENTATION

6.1 Development Platforms

In the last chapter we went through the overall system architecture, and we explored the conceptual designs of the three subsystems which are the AI assistant, the student side interface, and the instructor side interface. In this chapter, we will discuss the detailed design and the implementation of each subsystem. To begin with, the choices of development platform for each module will be presented below.

For the AI assistant, we decided to use the 'Wit.ai' platform.'Wit.ai' is an open and free web based platform which provides developers with a powerful yet simple natural language processing API for building customized NLP applications. It has a straightforward interface for training as shown in Fig where developers only need to input sample utterances and label it with corresponding intent.

For the student side smartphone interface and the instructor side interface, we chose Unity as the development platform. Unity is a real-time development platform which supports cross-platform software development. It has a user-friendly GUI for the developers and uses C# as the programming language. It also supports AR toolkit integration such as Vuforia, Google ARCore, Apple ARKit. The aforementioned features of Unity perfectly suit our requirements for the student side and instructor side interface development.

For the robot base of the studentside interface. Arduino IDE is chosen as the development platform, since the robot kit comes with an Arduino Uno board as the processor and a ready-to-use arduino source code for the robot control. Arduino is an open-source electronics platform based on easy-to-use hardware and software, which provides an easy-to-use programming environment for electronics or IoT prototyping.

6.2 Network Architecture

The core of the whole teleconsulting robotic toolkit is the network architecture that connects instructor AR-enabled robots and 'Wit.ai' platform together. The overall hierarchy of the network is depicted in figure 6.2. The robot interface as well as the instructor's console

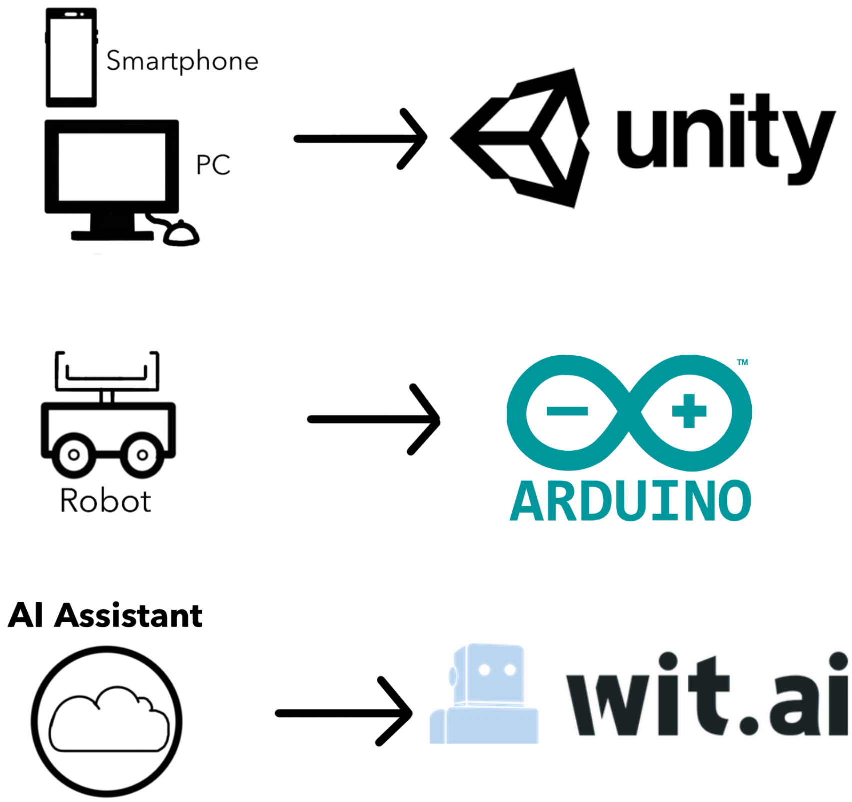

Figure 6.1. Platform Choice for Each Module

application are all developed in Unity 3D, which is a game engine. During the remote instruction process, the phone on top of the robot first transmits the live video feed to the instructor's console application. The video input from the phone is first converted into a byte array per frame using the widely adopted Base64[87] video encoding mechanism.

Meanwhile, the instructor has to send back commands to the robot to control its movements as well as virtual instructions-in the form of video/text/spatial sketch-to the phone. Except for the videos which are handled in the earlier mentioned way, other discrete messages are all encoded straightforwardly in json format to avoid extra overhead.

The aforementioned bilateral data transfer is implemented using the ZeroMQ[88], an open-source universal network library that supports multiple languages and platforms. It has inbuilt high-speed asynchronous I/O engines which makes it ideal for our video/audio transmission application. The connection is established with the TCP/IP protocol, which has an error-detection and retransmission mechanism that ensures reliability.

Since in our design, the phone is mounted on top of the robot for better flexibility, there has to be another layer of transmission to ensure that the command signals received by the phone are routed to the robot. We choose the bluetooth protocol for this task by adding an extra bluetooth module to the robot which connects to the bluetooth channel of the phone. Transferring data using Bluetooth brings appealing advantages such as lower cost and lower power consumption, which is exactly what we are looking for in our use case. Its limitation in terms of transmission range and throughput doesn't concern us since we are only using it to transfer remote command data which is in low volume and the phone will be closely on top of the robot throughout the time.

Last but not least, we need to establish the data transfer pipeline between the phone which takes in student's questions and displays recognized intent on the Wit.ai platform which runs the natural language processing application. Wit.ai as a cloud-hosted framework offers the HTTP API for developers to access its core services. The Hypertext Transfer Protocol (HTTP) is an application layer protocol for distributed, collaborative, hypermedia information systems. It has many advantages including better flexibility, less latency and more security. For maintaining security, Wit.ai uses OAuth2 as an authorization layer. As such,every API request must contain an Authorize HTTP header with an access token. Every students' utterance is sent to the Wit.ai server as an HTTP request encoded in a JSON format. The calssified intent which is then sent back to the phone as an HTTP response is also encoded using the JSON format if the recognition process runs successfully.A failed request would yield a plain text error. In short the bilateral data exchange between the phone and the Wit.ai server is implemented using the straightforward HTTP post/request mechanism.

By working together, those different network protocols and their implementations adopted in different layers collectively construct a unified network architecture. It works seamlessly throughout the process to ensure that the instructor can always get access to students' workspace, remote instruction can always reach the students, and the natural language processing service on the cloud is always accessible.

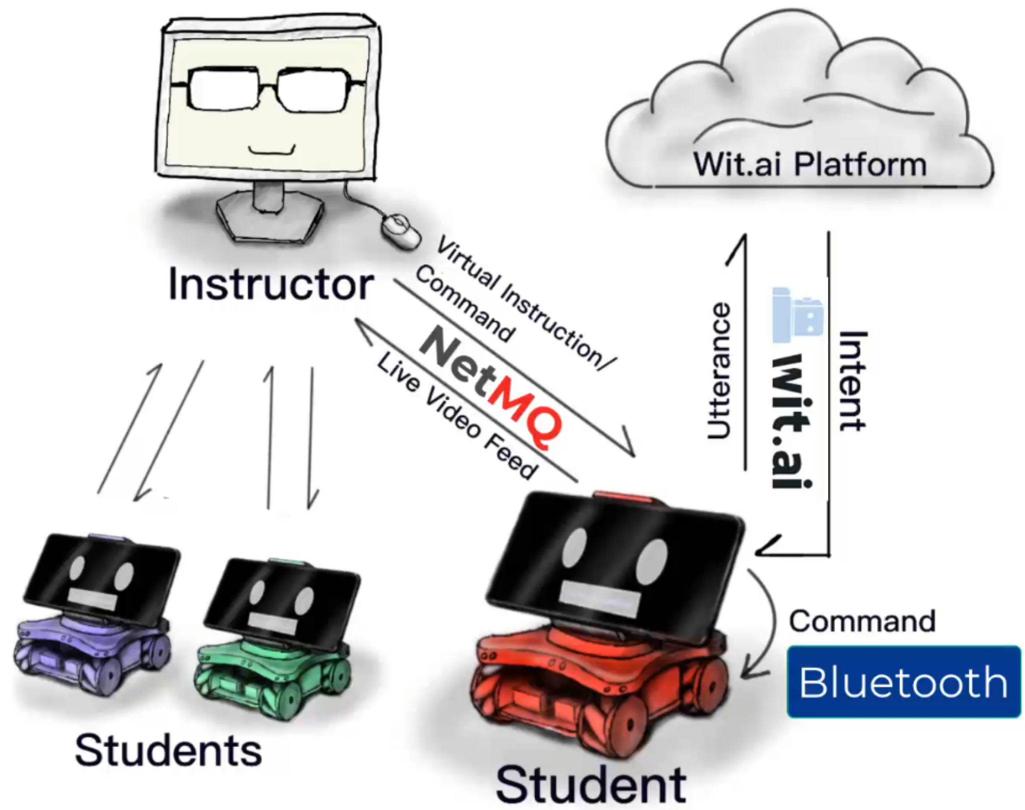

Figure 6.2. Network Architecture

6.3 Implementation of the AI Assiatant

From our elicitation study we find out that there is a need to relieve instructors from answering similar questions repeatedly throughout the session. To tackle this problem and let instructors focus on important matters, we trained an AI voice assistant responsible for providing hints or direct answers–to a common set of questions we trained for the makerspace session–using the 'Wit.ai' framework[30] which has advanced natural language processing capability. Once the NLP model is properly trained, the developer can then connect his/her own applications to wit.ai backend with an HTTP request as shown in. Both voice language input and intent recognition results are sent back and forth using the HTTP protocol.

We trained our own chatbot for answering questions from students following the aforementioned standard guideline with an additional interactive design process. Our goal is to make the AI assistant more competent-which can recognize questions comprehensively and

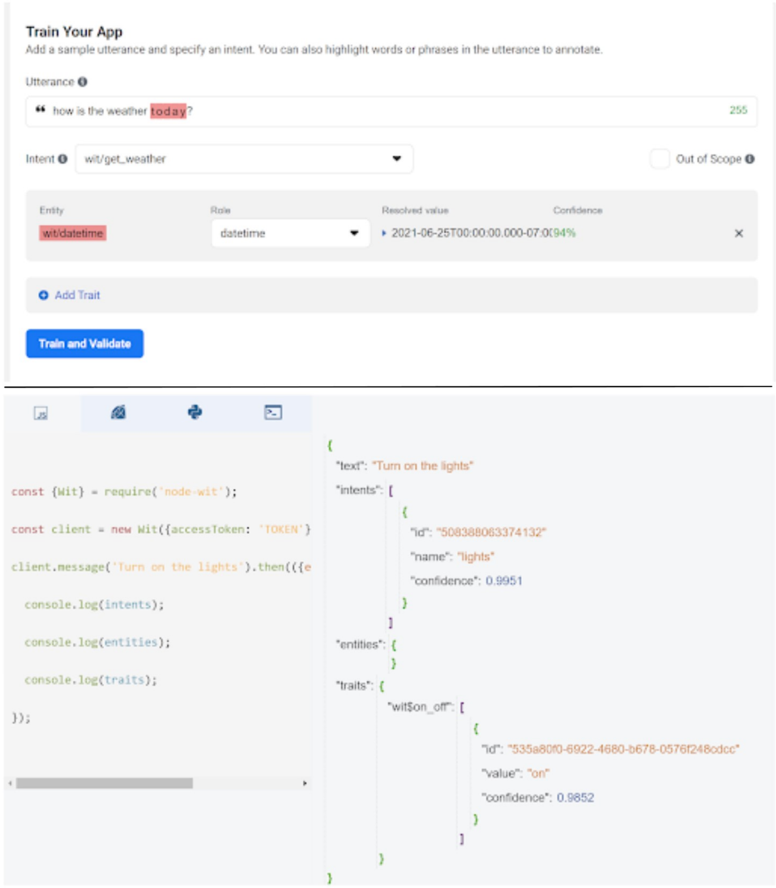

Figure 6.3. Wit.ai Interface

provide the most accurate answer-a sufficient number of questions and answers are needed for training. In our workflow, if the AI assistant understands the questions asked, it instantly displays pre-logged answers on the screen. We designed an iterative scheme to progressively train the assistant as seen in the schematic of figure 6.4. Whenever a student finishes a Q&A session, he or she is prompted with a question asking if the AI assistant provided the appropriate answer so that he/she no longer needs additional guidance from the instructor personally. If not, the system automatically logs the question that was asked into an aggregated database, for later reference by the instructor after the makerspace session is over. We assume that an unsuccessful Q&A experience could be caused by two possible reasons: either the question is not properly recognized, or the answer is not satisfactory. In the first case, the instructor adds the new question-answer pair into the training queue. In the latter case, the instructor can choose to modify the preexisting answer should he or she deem it necessary. We leave it to the instructor's discretion to distinguish between these two cases and take appropriate actions accordingly. With these feedback processes happening periodically, the accuracy of the robot improves over time. This whole workflow essentially becomes a crowdsourcing experience where different instructors contribute their own input to make the NLP functionality better.

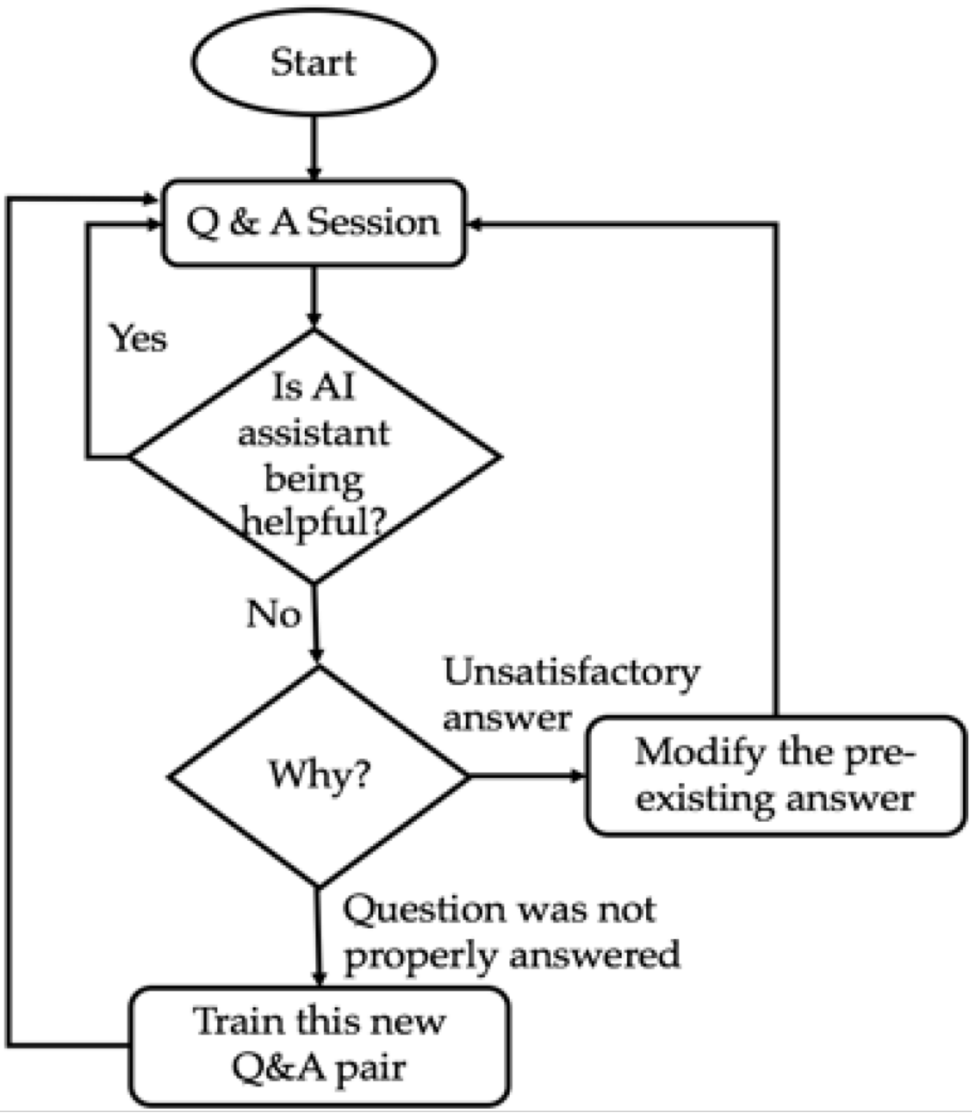

Figure 6.4. Workflow of the AI Assistant

6.4 Implementation of the Student Side Interface

The student side interface contains two modules: a)the robot, and b) the smartphone. The robot provides mobility to the smartphone, which allows the instructor to send the smartphone to areas of interest inside the workspace and view the layout through the camera. The smartphone controls the movement of the robot, plays the role as an AI teaching assistant, and displays the spatial instructions created by the instructor to the student on the screen.

6.4.1 Implementation of the Robot

Hardware Architecture

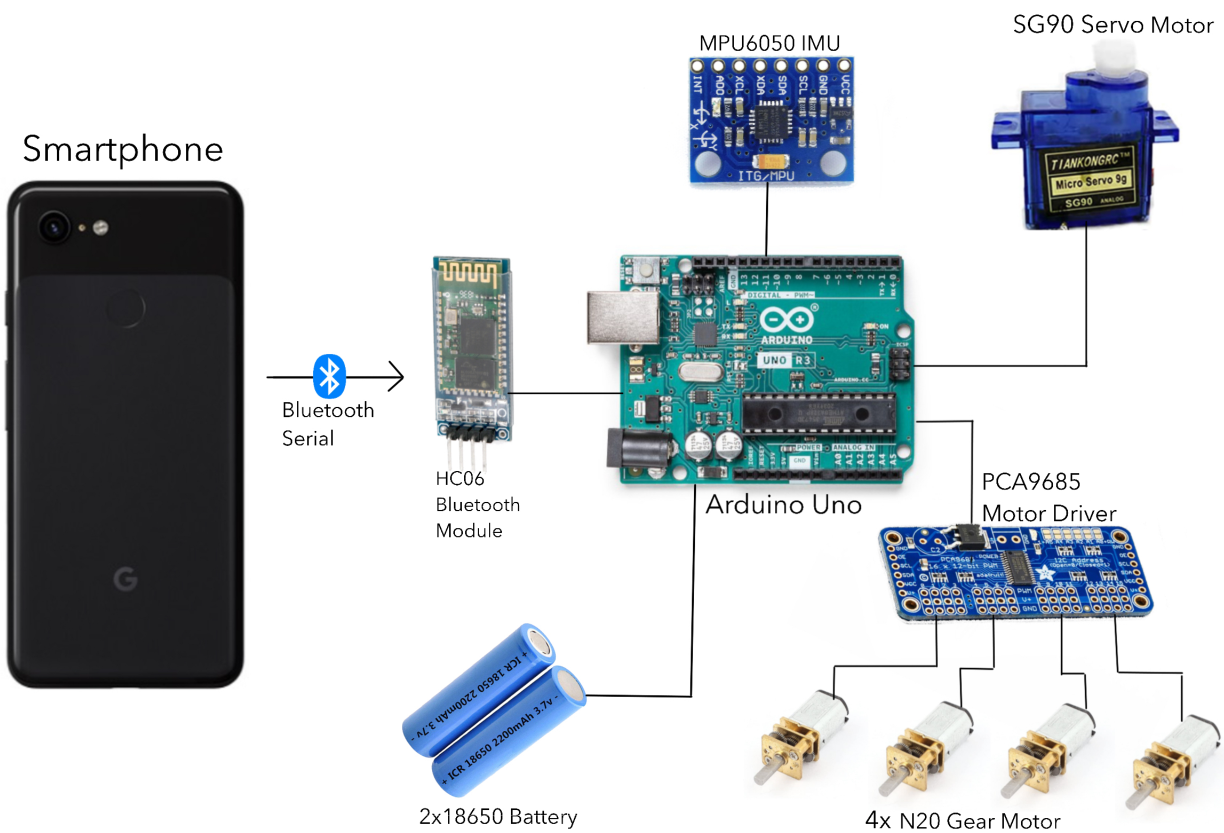

Figure 6.5. Hardware Architecture of the Robot

Figure 6.5 shows the hardware architecture of the robot. It should be noticed since in our design, the phone is mounted on top of the robot for better flexibility, there has to be another layer of transmission to ensure that the command signals received by the phone are routed to the robot. The original robot base has a WIFI module for wireless data transmission. We instead choose the bluetooth protocol for this task by adding an extra bluetooth module HC06 to the robot which connects to the bluetooth channel of the phone. Transferring data using Bluetooth brings appealing advantages such as lower cost and lower power consumption, which is exactly what we are looking for in our use case. Its limitation in terms of transmission range and throughput doesn't concern us since we are only using it to transfer remote command data which is in low volume and the phone will be closely on top of the robot throughout the time. The robot is powered by two 18650 batteries, each has 2500mAh capacity and provides 3.6V voltage. These batteries guarantee a 1.5 hour operation lifetime of the robot, which satisfies most of the makerspace learning. The processor of the robot is an Arduino Uno board, it can receive data through the bluetooth serial port from the smartphone and output signals to the attached components. The components include a PCA9685 16-channel motor driver which connects to the 4 N20 gear motor to drive the manucam wheels, a SG90 servo motor which can generate a torque of 1.8kg.nm that can lift every smartphone easily, and a MPU6050 IMU which provide the orientation data to the processor when the smartphone is offline.

Mechanical Design and Fabrication

As we mention in chapter 5, there are modifications that need to be applied to the original Yahaboom robot. The main modification is to replace the camera platform with a phone holder that can adjust the view angle of the smartphone. Figure 6.6 shows the first design of the phone holder. This design attaches the phone holder directly to the drive crank of the servo motor. However there was a significant deficiency in this design. As shown in figure 6.6, around ¼ field of view was blocked by the robotbase when the phone holder reached the lowest position. This deficiency cannot be solved by moving the phone holder's position forward or by increase the length of the phone holder, since the first approach will unbalance the center of gravity of the robot cause rollover, and the second approach will significantly increase the required torque base on the equation

$$\tau = r * F * sin(\theta)$$

. In such a case a new mechanism is needed.

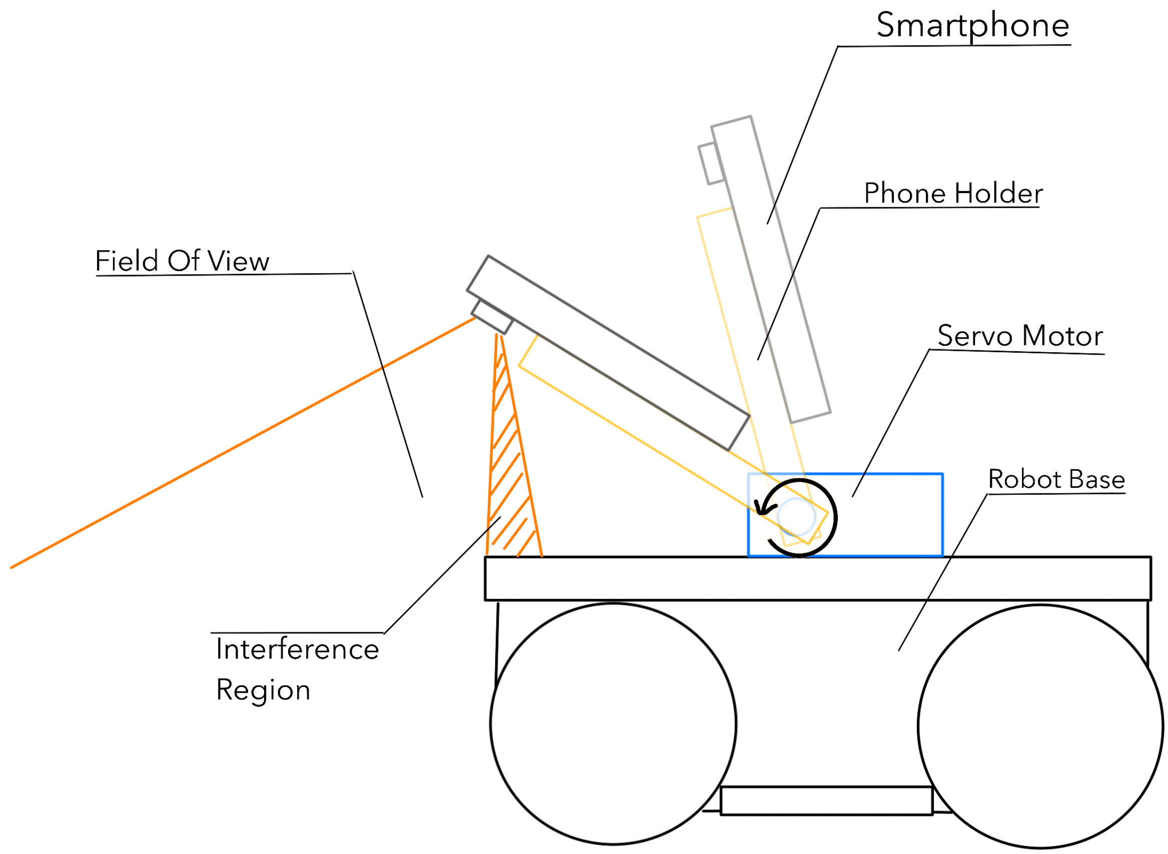

Figure 6.6. Deficiency of Direct Mount Mechanism of the Phone Holder

In the second iteration, an inverted slider crank mechanism was introduced. As shown in fig 6.7, the new phone holder mechanism has three components- a crank attached to the servo, a slider, and a rocker. The end of the crank is connected to the pin joint of the slider, and the slider will carry the smartphone and move inside the slot of the rocker. Figure 6.7 demonstrates the mechanism by showing the two limit positions of the phone holder. As the phone holder moves to the lower position, the slider will be driven by the crank and push the smartphone forward. As a result shown in figure 6.7, there will be no more interference for the field of view.

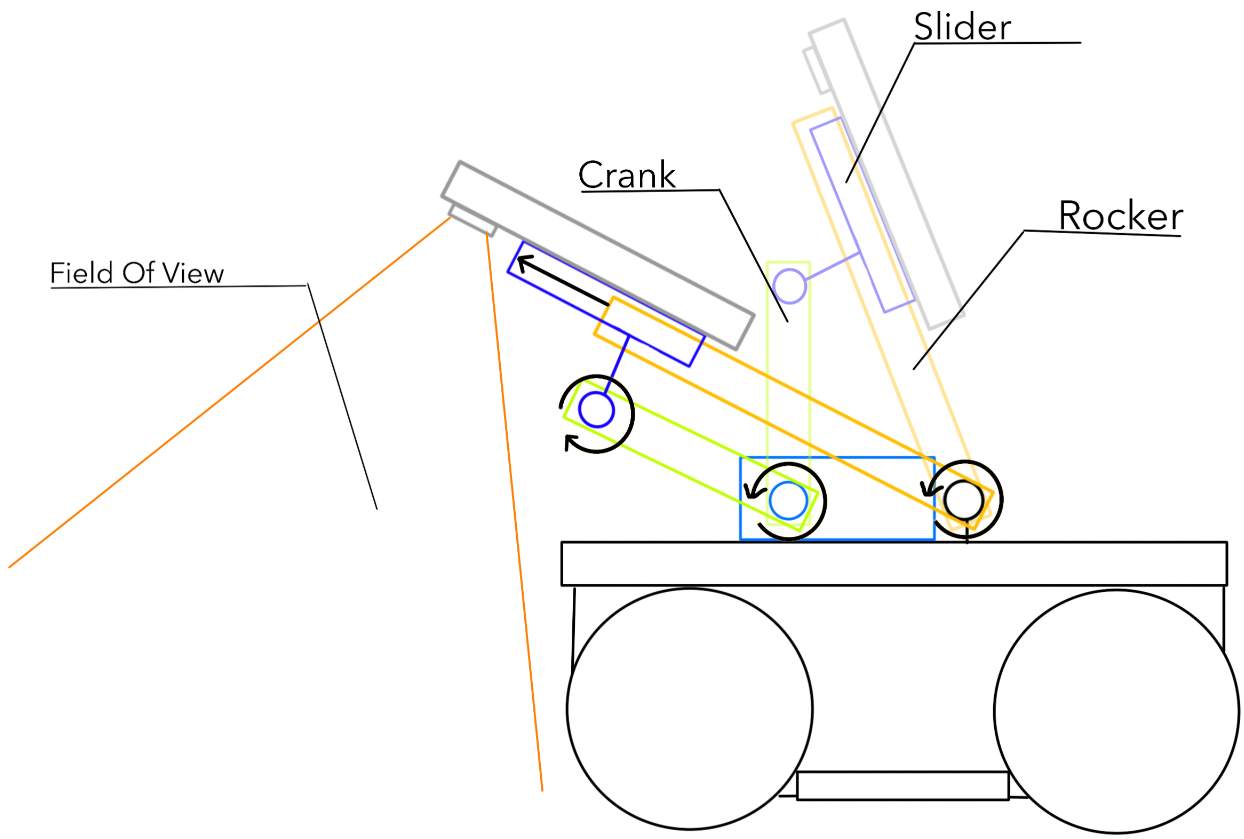

Figure 6.7. The Inverted Slider Crank Mechanism Design

After we settle the phone holder design, we use CAD to generate our first virtual prototype. The prototype has following parts as shown in the left side of figure 6.8-

a) The base of the phone holder

b) The right stand for the crank and the rocker

c) The left stand for the crank and the rocker

d) The crank

e) The rocker

f) The right part of the slider

g) The left part of the slider

h) The center part of the phone holder clip

i) The upper part of the phone holder clip

j) The lower part of the phone holder clip

By assembling these parts together, the phone holder mechanism was complete with following components-

A) The ground

B) The crank

C) Rocker

D) The slider

E) The phone holder clip

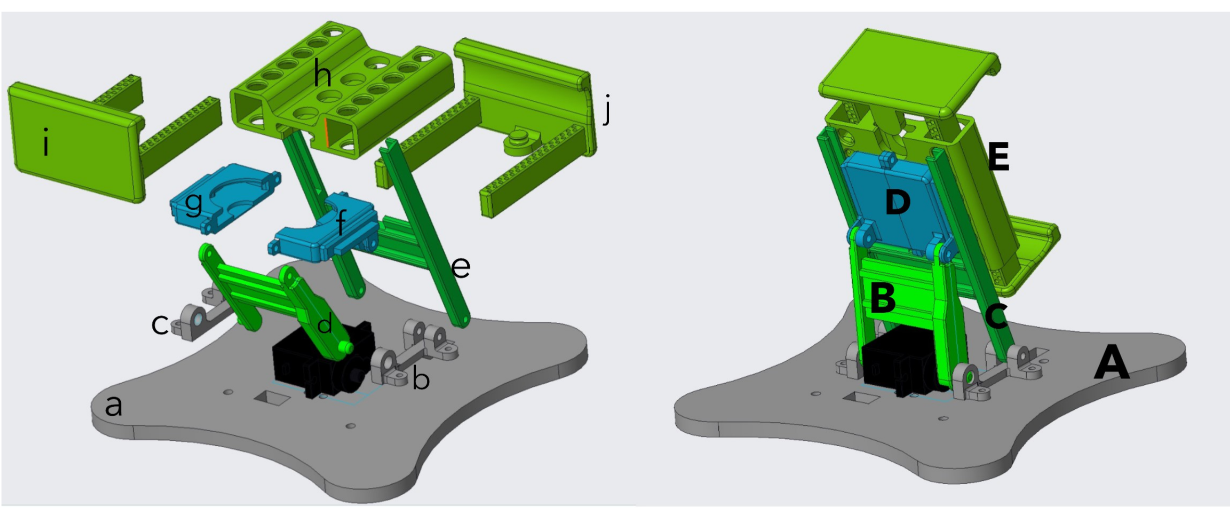

Figure 6.8. CAD Model of the Phone Holder

Figure 6.9 provides a demonstration of the phone holder mechanism. The slider can move the phone up to 15mm further from its original position and provides a 20 degree viewing angle at lower limit position and a 70 degree viewing angle at upper limit position, which have been tested as the appropriate viewing angles to inspect a makerspace.

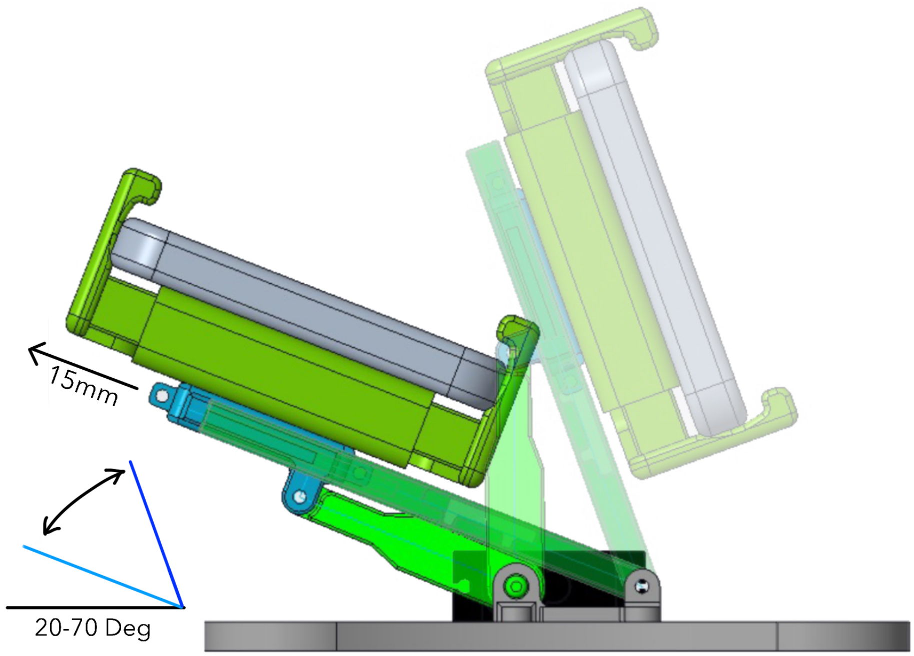

Figure 6.9. The Inverted Slider Crank Mechanism Demonstration

In order to satisfy different preferences of viewing the phone. We gave the phone holder a rotating mechanism. As shown in figure 6.10, the phone holder clip can be rotated by the user to switch the phone between portrait mode and landscape mode. There are limit slots at both positions so that the phone will not rotate freely.

After we generated the CAD model of the prototype, we used a SLA 3D printer to print all the parts. A SLA printer can print a part with relatively high accuracy, so that it can save us a lot of time from polishing the 3D printed parts. All the parts were assembled together and the final integration of the robot is shown below in figure 6.11. All the functions were tested and verified working.

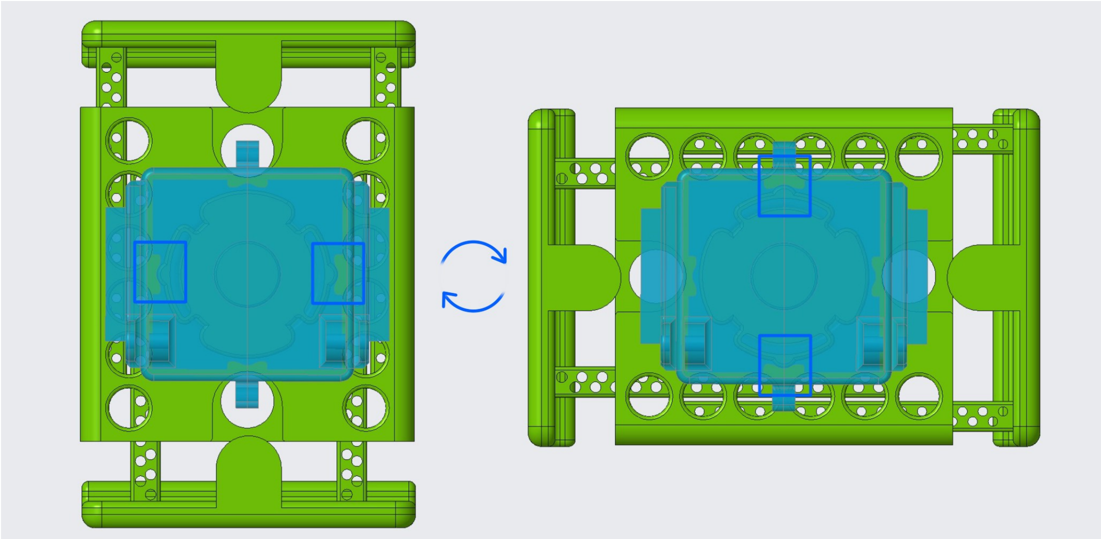

Figure 6.10. Phone Holder Switch Between Portrait and Landscape Mode

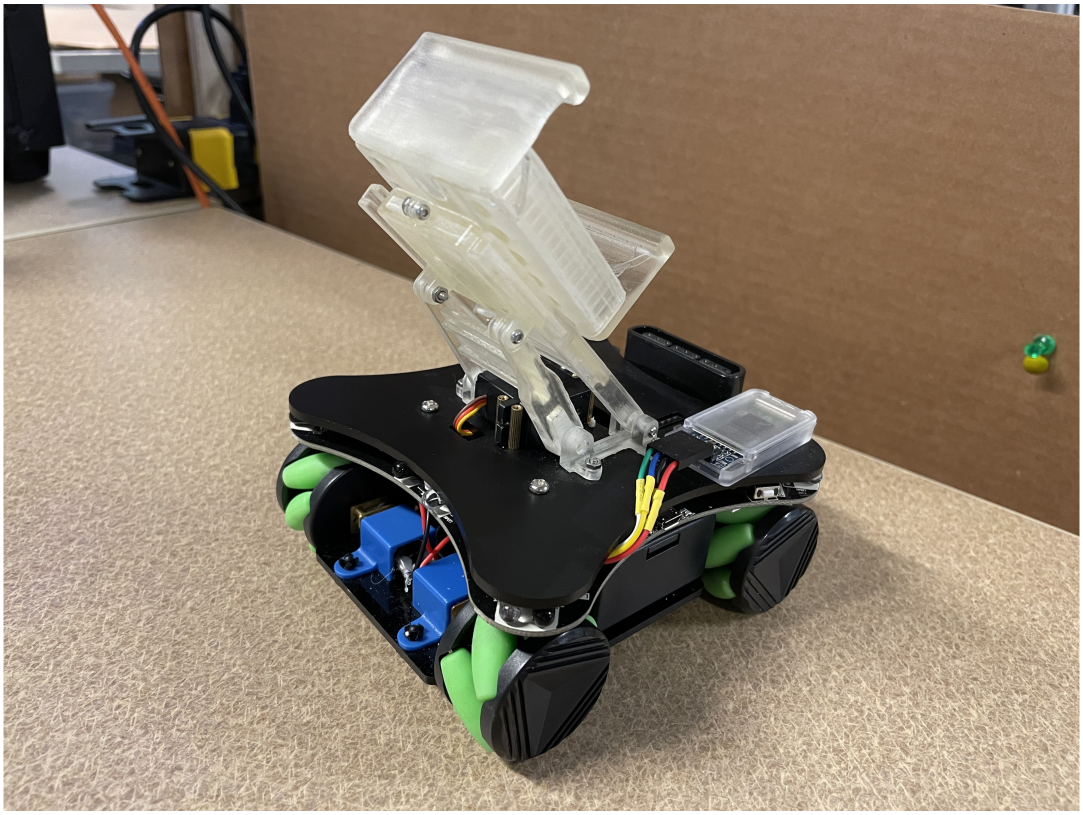

Figure 6.11. Final Integration of the Robot

6.4.2 Implementation of Smartphone Interface

We developed the smartphone interface using unity engine. The smartphone interface serves the purposes for collecting the 3D space information, controlling the robot, establishing the connection to the instructor and displaying AR instructions. In this section we will demonstrate the whole workflow of the smartphone interface, which includes Initialization, The Four-Step Setup, and the three modes i.e. Standby Mode, AR Animation Mode, and Teleconsult Mode.

Initialization

During the initialization phase, the smartphone will automatically connect to the robot-base via Bluetooth as previously mentioned. The user first needs to attach the smartphone to the phone holder, then the user can move the robot around the makerspace to obtain the 3D information and generate a 3D coordinate system. By integrating Google ARCore into the smartphone interface, the system can use the smartphone's main camera to detect the surface of the makerspace,generate a 3D coordinate system and gather 3D information of the smartphone's position and orientation in the 3D coordinate system. Figure 6.12 shows the moment when the user is scanning the makerspace. A 3D grid indicator will be displayed once the system successfully generates the 3D coordinate system. The user can step to the four-step setup phase.

The Four-Step Setup

Compared to the setup phase from chapter 5, we add two more steps in the final implementation. The new setup phase follows a four-step sequence as shown in figure 6.13, namely-Boundary Setup, Engaging Point Setup, Standby Point Setup, Face Location Setup.

Figure 6.14 shows the detailed operation of each step of the setup phase-

a) **Boundary Setup**: We found there is a possibility that the robot could fall off from the makerspace, which could cause hazard as well as money loss. So we design a virtual boundary mechanism to prevent the robot from falling. As shown in figure 6.14, the

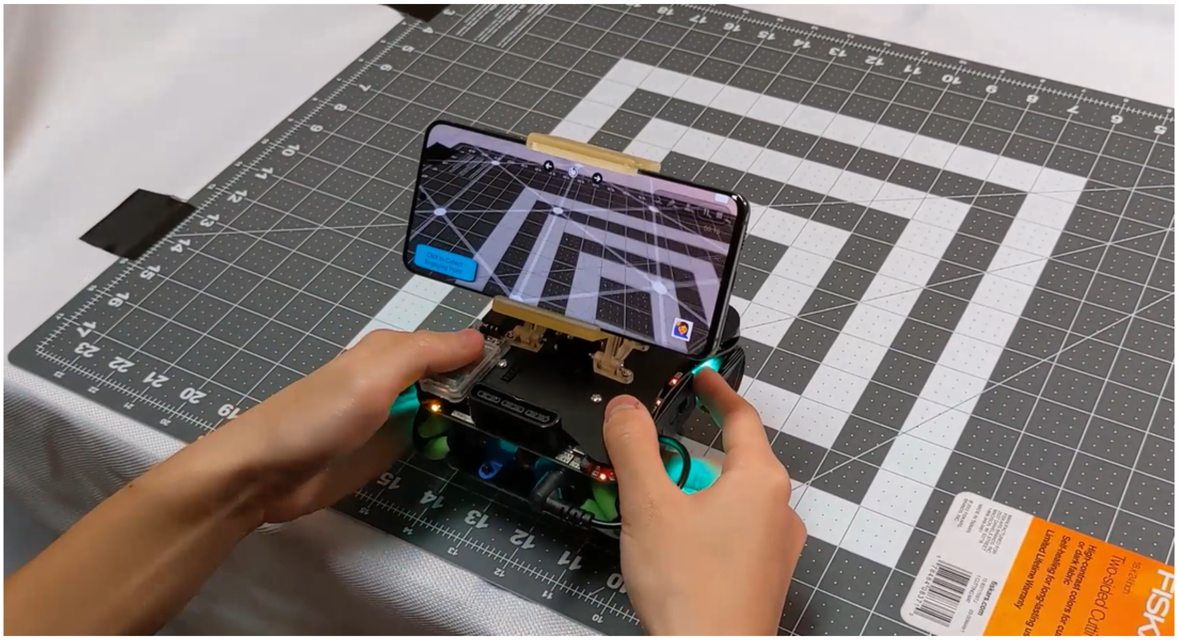

Figure 6.12. Process of 3D Coordinate System Generation

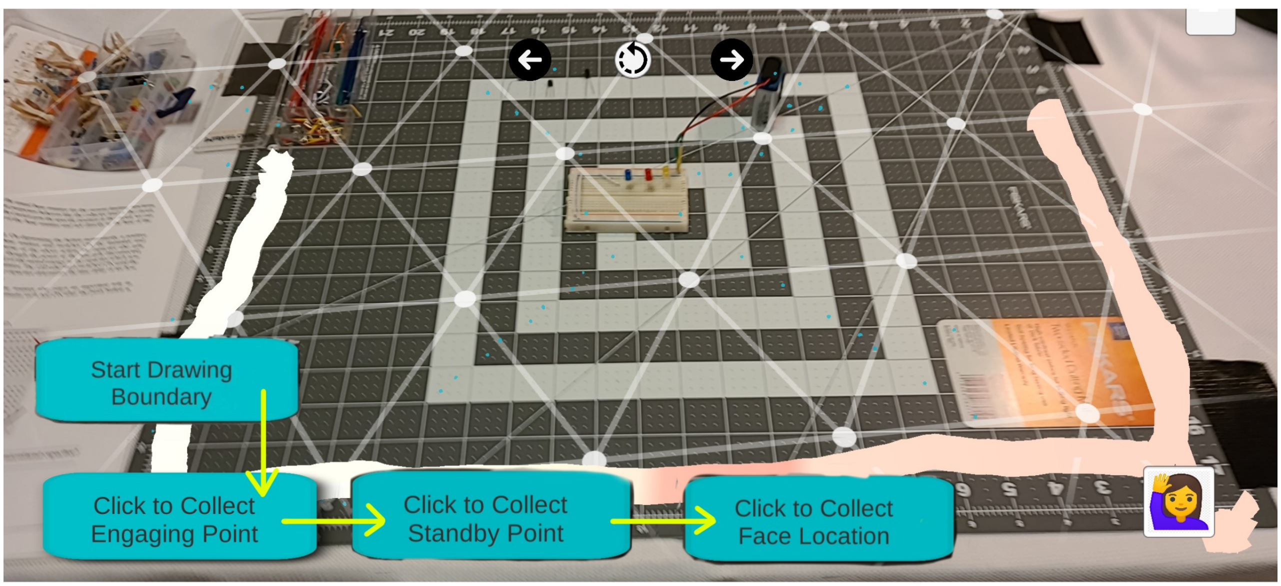

Figure 6.13. Four-Step Setup Overview

user can draw the virtual boundary on the screen using their finger. The system will then memorize the virtual boundary. Once the robot collides with the boundary during operation, the smartphone will send a stop signal to the robot base to terminate the current movement.

b) **Engaging Point Setup**: The engaging point is the location right in front of the working area, where the robot would be able to view the components and display the spatial instructions to the student. Instead of selecting the engaging point on the screen, we found that it is more natural to set the engaging point by placing the robot directly in the makerspace. To set the engaging point the student needs to place the robot at the position in front of his or her working area and then click the button on the screen to finish setup.

c) **Standby Point Setup**: The standby point is the location where the student feels comfortable to place the robot when he or she does not need to view spatial content through the robot. The setup process is similar to the process of engaging point setup. The student needs to place the robot at the position where he or she feels uninterrupted and then click the button on the screen to finish setup.

d) **Face Location Setup**: We found there is a need to improve the presence of the robot to the student. So we added a new feature to the robot i.e. the robot will face to the user's face location during standby mode. This feature not only increases the overall engagement, but also allows a more natural interaction between the student and the AI voice assistant during standby mode. Since during a makerspace the student barely moves the head, we only need to collect the face location once at the setup stage. The student needs to bring the robot close to his or her face and click the button to finish face location setup.

Once the student finished the four-step setup and placed the robot back to the makerspace, the robot will move to the standby point automatically and be ready for the makerspace session.

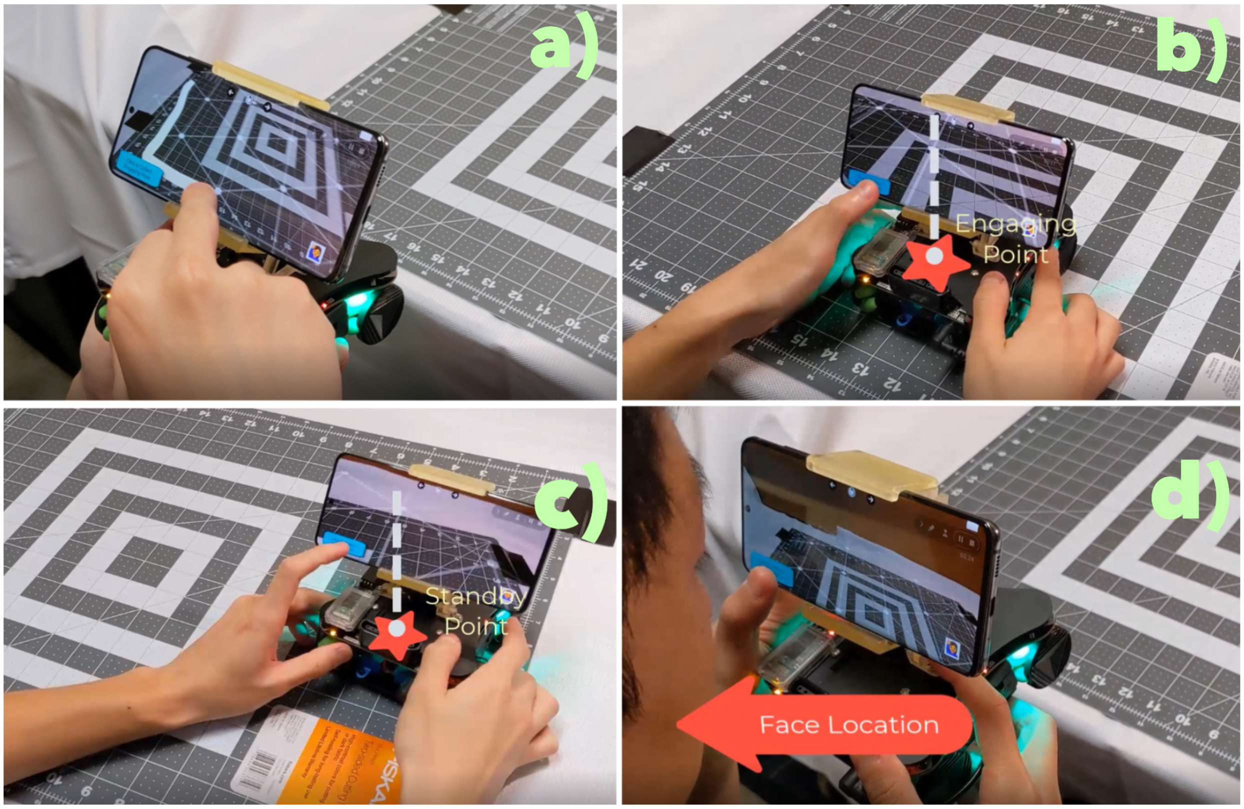

Figure 6.14. Four-Step Setup Workflow

Standby Mode

At standby mode the robot will stay at the standby point and face the student. The AI voice assistant is ready to use in this mode. As shown in figure 6.15, we design a cartoon face for the robot to make the robot feel more alive. To improve the liveliness of the robot we defined a set of facial expression animations.

When the student meets a question, he or she can interact with the AI voice assistant by touching the mouth of the robot and vocally raise the question. The AI voice assistant will send the question to the wit.ai platform as mentioned previously in the AI voice assistant implementation section. If there is an answer that matches the question in the dataset, the robot will display the answer with supportive material as shown in figure 6.16.

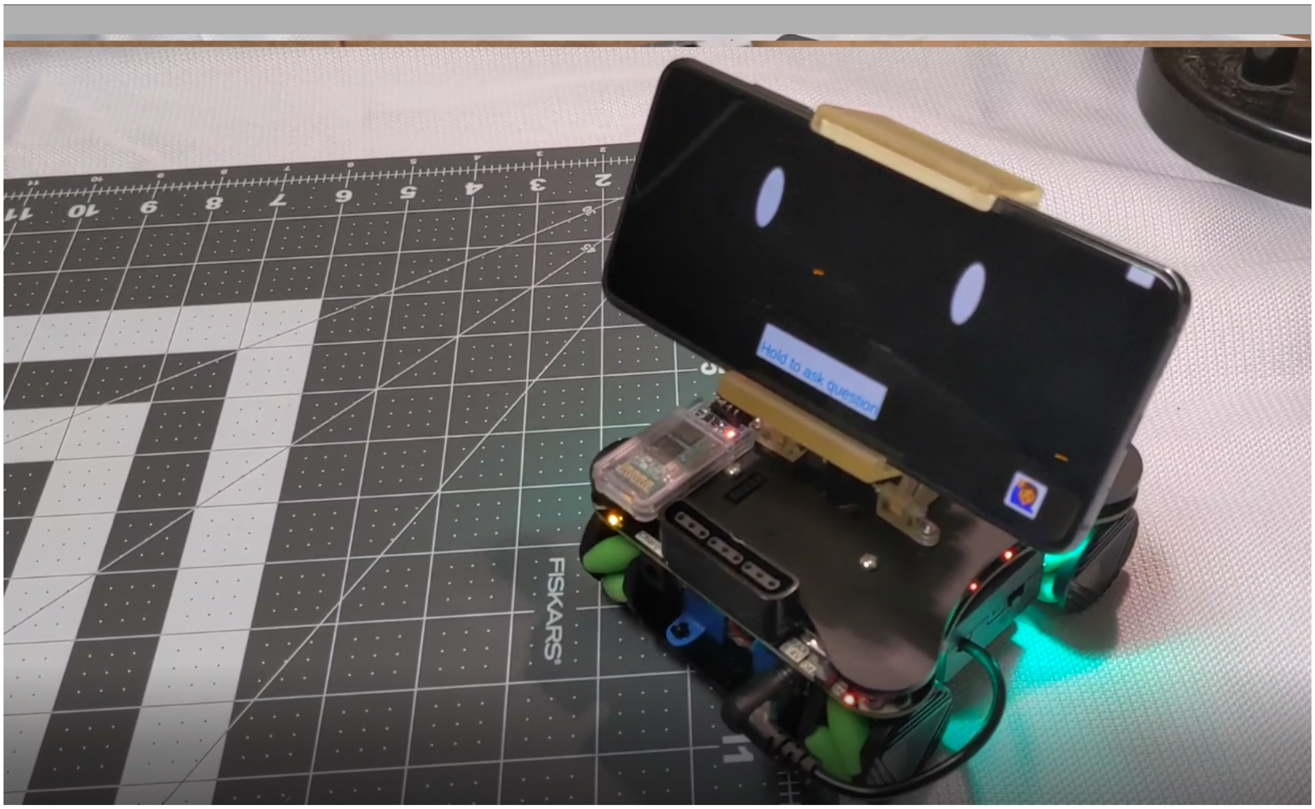

Figure 6.15. Standby Mode of the Robot

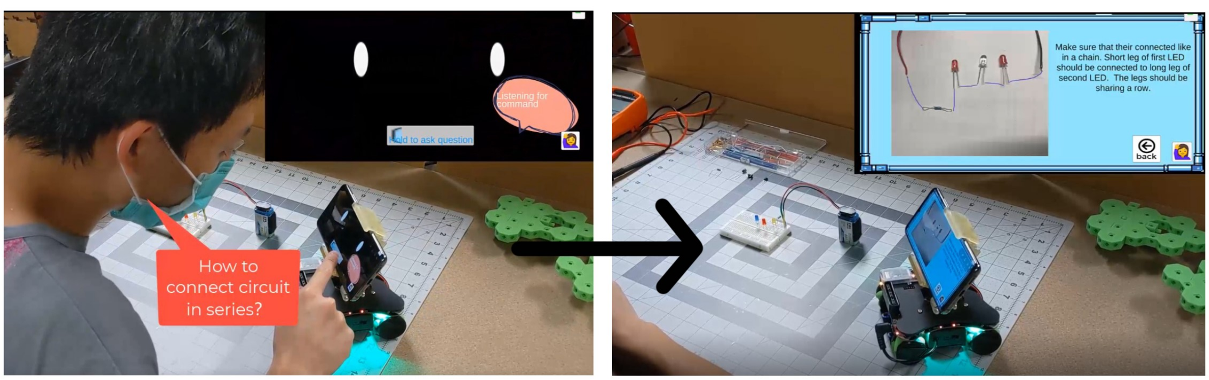

Figure 6.16. AI Voice Assistant Interface

AR Content Display Mode

The robot can move to the engaging point and display predefined AR animations. Compared with traditional text-based or video-based tutorials, AR delivers a richer user experience and conveys spatial information which is important to hands-on tasks. The instructor can import several AR animations in the form of Unity package to the system as supplemental material to introduce abstract concepts to students. These animations–which are initially displayed at the beginning of the session–can always be reviewed by students when scrolling back to this scene.

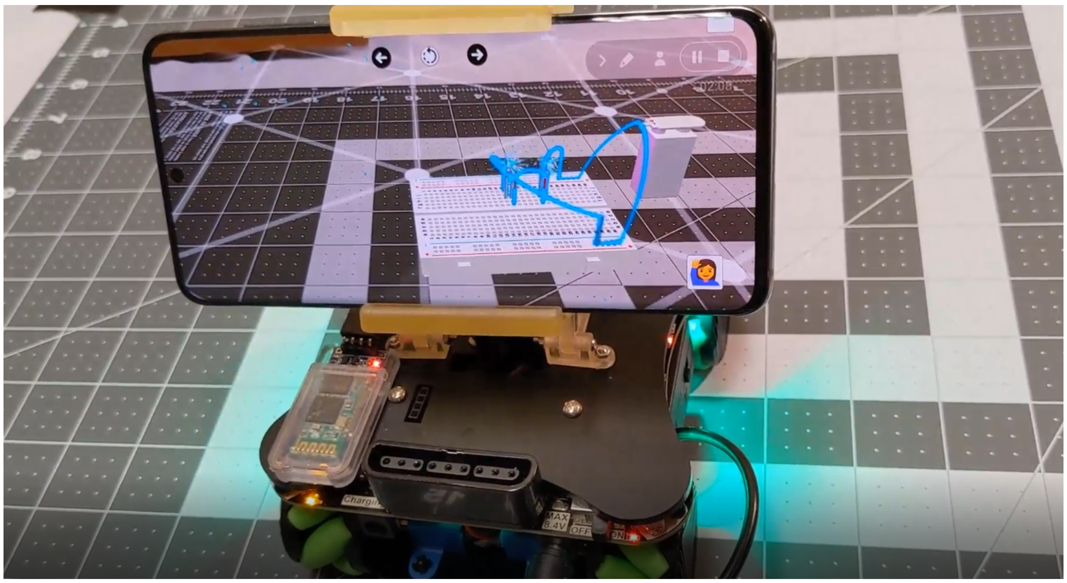

Figure 6.17. The Robot is playing an AR Animation about Circuitry

Teleconsult Mode

If the student needs further help, he or she can click the teleconsult button on the screen to request a teleconsult session from the instructor. Once the connection between the student and the instructor is established, the robot will automatically travel to the engaging point and share the camera view with the instructor. The instructor can manipulate the robot inside the makerspace and create spatial instructions as AR content on top of the components. All the Instructions will be synchronized and visible to the student. The details of the spatial instruction creation will be explained in the next section.

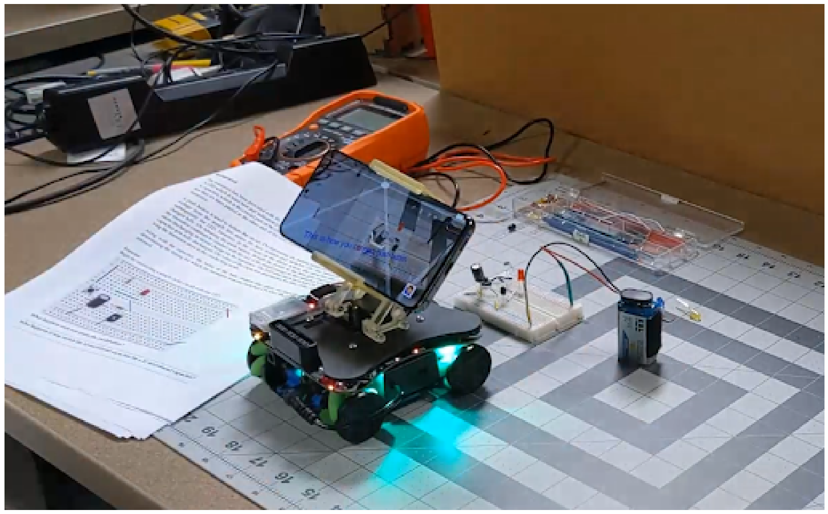

Figure 6.18. The Robot is at Teleconsult Mode and Displaying AR Content

6.5 Implementation of the Instructor Side Interface

The instructor side interface is also developed in Unity. The interface is running on the instructor's PC and requires external microphone and webcam attachment. The instructor side interface has two panels, which are the **Student Query List** and the **Teleconsult Panel**. The following sections will discuss the details about each panel.

6.5.1 Student Query List

Figure 6.19 shows the interface of the student query list. In this panel, all the student IDs are showing on the screen. Since we only prototyped three robot bases, we created three student IDs into the system. The student ID will be shown in grey color when the robot from the student side is in standby mode. As shown in figure 6.19, once student 3 requests a teleconsult session, the student's ID will be highlighted and a message will pop up to let the instructor accept or reject the request. By default the instructor will stay in the student query list. By clicking the accept button, the instructor will enter the teleconsult panel to assist the student.

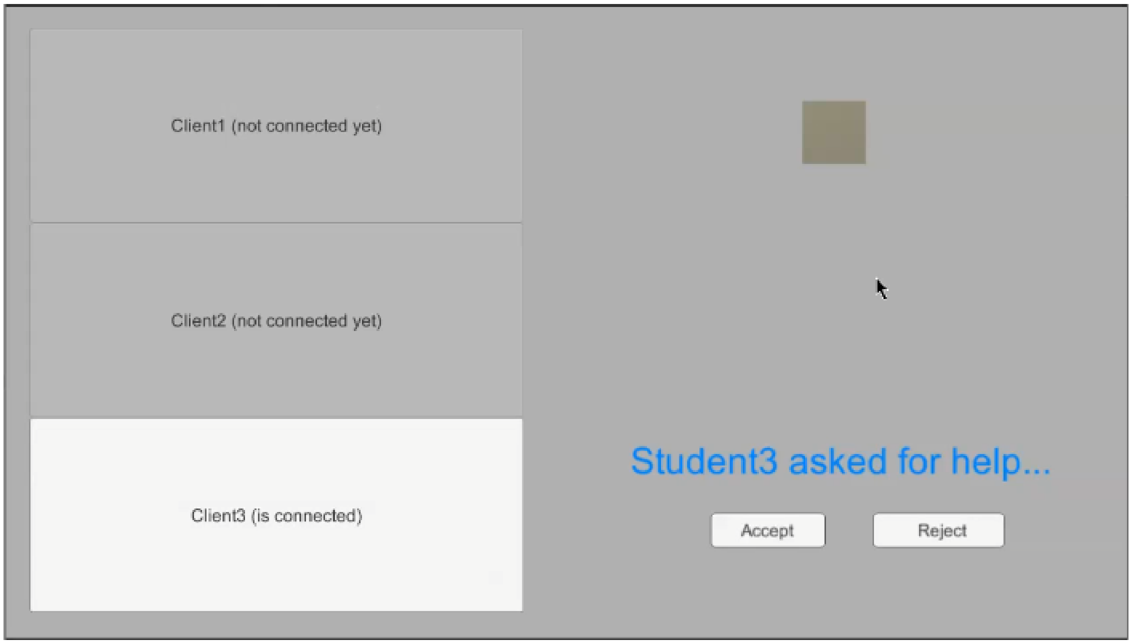

Figure 6.19. Student Query List Panel

6.5.2 Teleconsult Panel

Once the connection is established between the student side robot and the instructor side PC. The teleconsult panel can be separated into four regions as shown in figure 6.20. a) The live view window shows the student's realtime workspace. b)The command panel enables the instructor to operate the robot by moving it in any direction and tilting the angle of the phone holder. c)The content creation group provides a variety of options for instructors to deliver real-time AR instructions or send live demo from the webcam. d) The customization group allows the instructor to customize the AR instructions. By default, students and the instructor are able to talk to each other throughout the process. We will focus on explaining the implementation of the c)content creation group and d)customization group since they are most relevant to the creation of spatial instructions.

Content Creation Group and Customization Group

Section c in figure 6.20 includes all the function buttons of the content creation group. They are from left to right, top to bottom, namely-**Spatial Drawing, Spatial Text,**

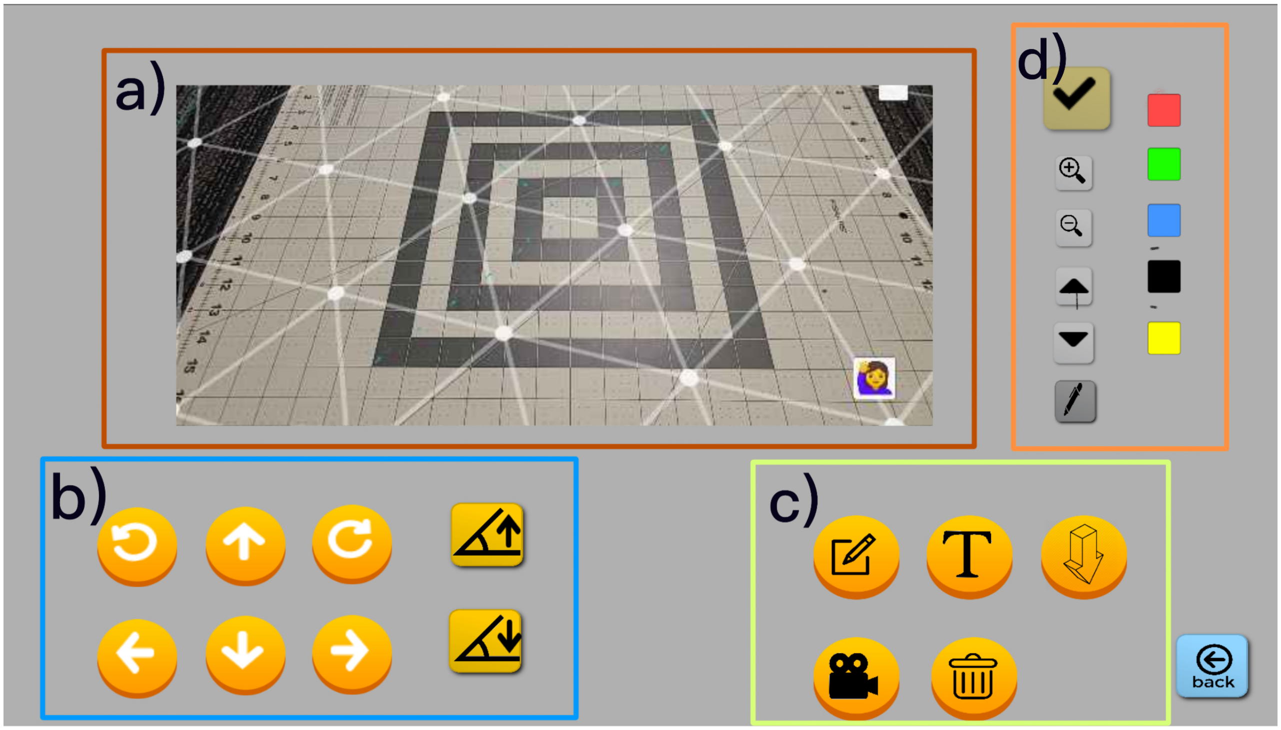

Figure 6.20. Teleconsult Panel

Spatial Indicator, Live Demo, and Delete Content. By default these buttons are highlighted and the d)customization group is disable in grey color. Once the instructor selects one function and enters content creation mode, the other buttons will be disabled and the customization group will be highlighted and activated. The options in the customization group will also change corresponding to the selected function. The instructor can click the check button in the customization group to finish customization and exit content creation mode.

Figure 6.21 shows the content creation mode for the three Spatial Content Creation Functions. To begin with, the instructor needs to use the mouse to click on the desired location in the live view window to place the AR content. The AR content will be instantiated at the mouse-click location with a Z axis value of 0 and an angular orientation of (0,0,0). The instructor can click on the scale buttons to adjust the scale of the AR content. And the instructor can change the height of the AR content by clicking the vertical adjustment buttons. We implemented an intuitive way to adjust the orientation of the AR content. To activate angular adjustment, the user needs to hold the right mouse button during the process. Then the user will move the mouse to change the orientation of the AR content. The system will gather the vector of the mouse's moving direction, and the AR content will rotate around the vector that is orthogonal to the mouse's moving direction. The detail of the AR content creation process of each function will be explained below-

a) **Spatial Drawing**: Spatial drawing gives the instructor the ability to draw in 3D space. In spatial drawing, the instructor first needs to place the drawing pad, then adjust its 3D position and orientation. Once the adjustment is fixed, the instructor can click the drawing icon in the customization group to start drawing. by holding the left mouse button and dragging the mouse on the drawing pad, the instructor can draw 3d lines. The instructor can change the color of the drawing as well as the size of the drawing. Once the instructor is satisfied with the drawing, he or she can click the check button to finish the content creation.

b) **Spatial Text**: Instructor can create 3D text instructions and superimpose the 3D text right next to the corresponding components in the makerspace by the spatial text function. The instructor will first type inside the input field in the customization group. Then the instructor needs to click on the desired location to place the 3D text, and adjust the height, orientation, color, and size of the 3D text. Once the previous steps are done, the instructor can click the check button to finish the content creation.

c) **Spatial Indicator**: When the instructor wants to highlight a component or a location in the makerspace, he or she can use the spatial indicator function. We provide three 3D indicator options, which are star, maker, and arrow. The instructor can choose the indicator, adjust the position, orientation, size and color of the indicator following the same approach mentioned above, and finish the content creation by clicking the check button.

Figure 6.21. The Spatial Content Creation Functions

7. SYSTEM VALIDATION AND RESULTS

7.1 Validation Approach

To validate our system, we performed a user study to test our setup and its effects on an augmented makerspace, involving a hands-on session between instructors and makers as shown in figure 7.1. In this user study, we mimic the methodology being used by instructors in virtual makerspaces and compare it to our robotics toolkit. Thus, we split our experiment into two conditions: (a) Videoconferencing with Zoom, (b) RobotAR, which includes AR delivered instructions, the voice assistant, and the option of teleconsulting. We decided to juxtapose our toolkit capabilities with the technology currently used and available in virtual makerspaces. Then, we will analyze the effects of our toolkit for the instructors, the students, and the interactions between them.

7.2 Participants

We recruited 24 participants (15 male, 9 female) ranging from 20 to 28 years old (M=22.3, SD=2.65), all of which had experience with online classes and virtual laboratories, but little experience with electrical circuitry or virtual makerspaces. Participants were distributed in groups of 3 students per each session. The instructor leading all the sessions for both conditions had more than 2 years of experience teaching robotics classes and giving workshops at physical makerspaces. 15 of our participants had previous experience with voice assistants, 2 had prior experience with robotics, and 10 had experience with AR applications.

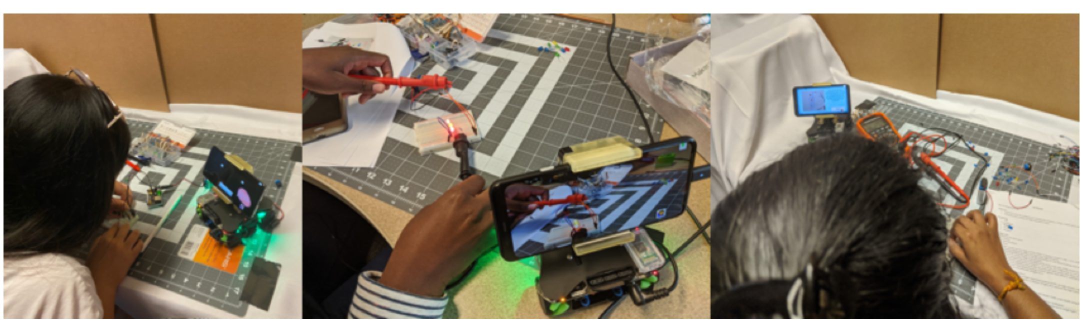

Figure 7.1. the Student Interacts with RobotAR

7.3 User Study Setup

First, the context of the class was a three-part single session–using RobotAR or Zoom, in which each participant was in a separate room. Each part lasted about an hour and there was a short break (5-10 mins) in between each hour. Likewise, the instructor was in another room, but given complete vision of the student's workspace via Zoom or our platform. There was at least one researcher physically present with each participant, while the participant teleconferenced with the instructor as necessary. Due to conflicting schedules and availability of robots, we had the instructor teach each session to 3 students at a time for both conditions. For the RobotAR condition, each student was provided with a robot; while for the Zoom condition, each student was provided with a tablet.

We chose a crash-course introductory lesson on basic electrical circuitry, which is part of an undergraduate class on electrical circuitry and programming. The series included the following parts: Using basic tools, Connections in series and parallel, Transistors and capacitors. We selected this use case due to the following reasons: (a) we had access to a robotics instructor, undergraduate curriculum for the class, and the students' kits from previous classes; (b) circuitry and tools are the most used subjects in makerspaces.

Thus, each session was split as follows: (1) Lecture part, in which students got introduced to the material, received some demos, and discussed the new concepts; (2) Hands-on making, in which students attempted to complete all activities on their own, and requested instructor's aid if necessary. The lecture part lasted about 30 minutes and the rest of the session lasted about two and a half hours. In the Zoom condition, following the lecture part which included some live-demos, students were able to teleconsult the instructor any time they required help. In the RobotAR condition, during the lecture part, students received the demos via AR. During the hands-on making, they were able to use the voice assistant first, then teleconsult with the instructor via the robot if they wanted help, clarification or a check-in.

7.4 User Study Activity

We gave each student a Makeronics (7 in 1) electrical circuitry components kit, so they could participate in the experiment. These are the components from the kit which were used for the session: a breadboard, jumper wires, capacitors, LEDs, buttons, transistors, resistors. We also provided a multimeter for each student to take measurements of current and voltage, and verify connections. Since our audience had little knowledge in circuits, the activities at each of the three parts involved a short lecture on basic tools and components (e.g., LEDs, wires, batteries, multimeter) with instructor-guided circuits (e.g., 2 LEDs in series and 2 in parallel), and a self-guided follow-up circuit (e.g., combined series and parallel circuits, while writing down measurements of voltage and current).

7.5 User Study Results

7.5.1 Pre- and post-test evaluations

Since we are aware that electrical circuitry performance goes beyond whether the circuit is working or not, we decided to establish a coding scheme to evaluate conceptual knowledge and hands-on performance. Past work has shown that important circuitry concepts are pervasively misunderstood well into adulthood [89]; thus, we decided to test participants on these concepts in the pre- and post-assessment (after the 3-hour session) tests. Additionally, we tested on whether students were able to identify the appropriate schematic diagrams of the circuits they were building. For example, the participant may use redundant connections to complete a circuit. Similarly, students may be able to calculate and measure voltage and current, but may not understand them conceptually. Each answer was scored with a 0 if incorrect, +0.5 if answer had some substance, or a +1 point if correct. Then, the total points were normalized to fit into the 1-point scale for each category. Past work on circuitry has proposed similar coding schemes and categories to score circuitry learning [90], [91]. The categories we considered for evaluation were the following:

Knowledge of voltage and current conceptual and applied understanding of voltage and current; *Polarized component orientation:* the positive terminal (+) of polarized components are consistently oriented toward the positive terminal or pin(s) of other components; *Connections in series and parallel:* successfully connect one component to another in series or parallel, as well as knowing its effects on voltage and current; *Knowledge of circuitry components:* functionality , placing, and connecting LEDs, resistors, push buttons, capacitors, transistors, batteries; The next key competencies did not have a pre-test because they included calculations from hands-on performance. *Use of breadboard:* appropriate placing of components to power and ground rails and in respective rows; *Use of multimeter and measurements:* measuring resistance, voltage, current, conducting short tests; *Working circuit:* using appropriate components, wires, and making sure the circuit is closed.

4 Key competencies were analyzed by coding pre- and post-tests, graded on a 1-point scale. While 3 other key competencies were obtained by collecting the answers from lab manual (test). All tests were coded by one primary coder. Inter-rater reliability on both the pre-test, test, and post-test was validated by having a secondary person score over 25% of the data. From our rubric, two researchers in charge of grading had a Cohen's Kappa of 0.714. As for the workshop, we had to wrap it up at the 3-hour mark. From the Zoom condition, only 3 out of the 12 students managed to complete all the exercises available. While, 7 out of the 12 students managed to complete them from the RobotAR condition. The rest of the students oscillated between 25% to 75% completion of the exercises. As for the results of the pre-test, test, and post-tests by condition, these are summarized in Table 7.1.

We analyzed scores with our aforementioned rubric for the key competences assessment. We began with a Shapiro-Wilk normality test to verify whether the normal distribution assumption was not met. Thus, to analyze the significance of our results from RobotAR and Zoom conditions, we conducted the Friedman Test with a post hoc analysis from Wilcoxon signed-rank test. When comparing these conditions, the Wilcoxon sign-rank test showed a statistically significant improvement for RobotAR condition in 3 out of 4 conditions: *knowledge of voltage and current* [$Z=-2.333$, $p<0.05$, $p=0.02$]; *connections in series and parallel* [$Z=-2.084$, $p<0.05$, $p=0.037$]; *knowledge of circuitry components* [$Z=-2.12$, $p<0.05$, $p=0.034$]. Likewise, the learning gains between pre-, post-tests are presented in Table 7.1.

Table 7.1. Pre-test, test, and post-test results of key competencies assessment

		Zoom		RobotAR		
Key Competency	**Time**	**M**	**SD**	**M**	**SD**	**Sig.**
Knowledge of voltage and current	Pre-test	0.246	0.263	0.254	0.235	*Z = -2.333* *p < 0.05*
	Post-test	0.842	0.07	0.813	0.092	
	Gain	**0.790**		**0.749**		
Polarized components orientation	Pre-test	0.153	0.246	0.169	0.262	*Z = -1.095* *p > 0.05*
	Post-test	0.813	0.084	0.788	0.068	
	Gain	**0.779**		**0.745**		
Connections in series and parallel	Pre-test	0.138	0.123	0.163	0.136	*Z = -2.084* *p < 0.05*
	Post-test	0.596	0.214	0.763	0.146	
	Gain	**0.531**		**0.717**		
Knowledge of circuitry and components	Pre-test	0.167	0.155	0.191	0.166	*Z = -2.12* *p < 0.05*
	Post-test	0.525	0.221	0.767	0.259	
	Gain	**0.430**		**0.712**		
Use of Breadboard	Test	**0.646**	0.155	**0.804**	0.144	*Z = -2.771* *p < 0.05*
Use of multimeter and measurements	Test	**0.508**	0.155	**0.717**	0.228	*Z = -2.998* *p < 0.05*
Working circuit	Test	**0.479**	0.13	**0.697**	0.21	*Z = -2.053* *p < 0.05*

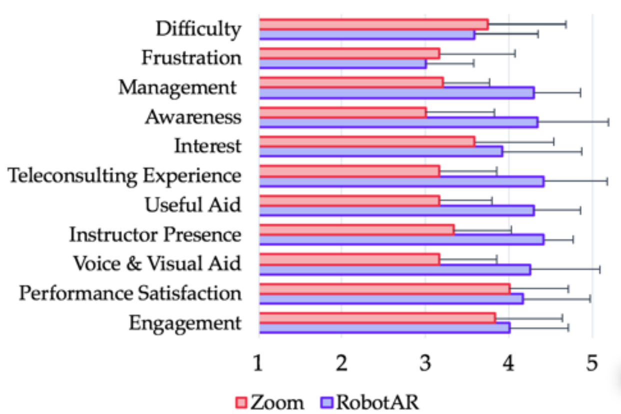

Figure 7.2. Results from average scores on the usability of RobotAR vs. Zoom

For the remaining key competencies, which are the scores obtained from the lab manual students returned, we also performed the Wilcoxon signed-rank test and found for that RobotAR condition showed a statistically significant improvement in all 3 competencies: *use of breadboard* [Z=-2.771, p<0.05, p=0.006]; *use of multimeter and measurements* [Z=-2.998, p<0.05, p=0.003]; *working circuit* [Z=-2.053, p<0.05, p=0.04].

7.5.2 Usability Evaluation

After the 3-hour user study session, we provided participants with a 5-point Likert scale (1-strongly disagree, 5-strongly agree) questionnaire. This survey was meant to assess the usability of RobotAR vs. the traditional teleconferencing media, Zoom. Figure 7.2 shows the average scores reported by participants. These results were representative of the following categories: *Engagement*; *Performance satisfaction*; *Voice and visual aid* from the system; *Instructor presence*; *Useful aid* from the instructor in real-time; *Teleconsulting experience*; *Interest* in the subject; *Awareness* of instructor; *Management* by instructor; *Frustration* with problem-solving; *Difficulty* of the learning material.

We conducted a Mann-Whitney U test on each of the categories. Thus, from the reported responses, we found participants preferred usability of RobotAR condition for the following ($p<0.05$): RobotAR (M=4.25, SD=0.829) provided a higher quality of voice and visual aid with its system than Zoom videoconferencing (M=3.167, SD=0.687), U=110, p=0.007; RobotAR (M=4.417, SD=0.344) improved the overall instructor presence as compared to Zoom videoconferencing (M=3.333, SD=0.687), U=131, p=0.000; RobotAR (M=4.292, SD=0.557) allowed the instructor to provide more useful aid in real-time than Zoom videoconferencing (M=3.167, SD=0.624), U=128, p=0.001; RobotAR (M=4.417, SD=0.759) provided a higher quality of teleconsulting experience than Zoom videoconferencing (M=3.167, SD=0.687), U=135, p=0.000; RobotAR (M=4.333, SD=0.849) provided greater awareness of instructor than Zoom videoconferencing (M=3, SD=0.816), U=126, p=0.001; RobotAR (M=4.292, SD=0.557) instructor's management of student's workspace than Zoom videoconferencing (M=3.208, SD=0.557), U=130, p=0.000. For the remaining categories no statistically significant differences were found ($p>0.05$): Engagement (RobotAR: M=4, SD=0.707; Zoom: M=3.833, SD=0.799, U=0.78, p=0.727); Performance satisfaction (RobotAR: M=4.177, SD=0.799; Zoom: M=4, SD=0.707, U=81, p=0.6); Interest (RobotAR: M=3.917, SD=0.954; Zoom: M=3.583, SD=0.954, U=83, p=0.505); Frustration (RobotAR: M=3, SD=0.577; Zoom: M=3.177, SD=0.897, U=66, p=0.727); Difficulty (RobotAR: M=3.583, SD=0.759; Zoom: M=3.75, SD=0.924, U=61, p=0.506).

7.6 Disscussion

In this section, we discuss the findings of our user study and reflect on how they influence the questions we posed in the introduction.

Q1: To what extent does the use of RobotAR lead to an improvement in students' key competencies and user experiences compared to traditional teleconferencing platforms?

Students were overwhelmingly positive about RobotAR. There was a consensus among students that our robotics toolkit was a viable alternative to provide high-quality teleconsulting in an immersive, focused approach.

"It's fun, it's convenient, it's educative. I feel like I'm in a new age of learning."-P8

Our results showed that RobotAR was conducive to an improvement in assessment of key competences when compared to Zoom teleconferencing for 6 out of 7 categories: *knowledge in voltage and current, connections in series and parallel, knowledge of circuitry and components, use of breadboard, use of multimeter and measurements, working circuit.* Much of the learning that takes place at makerspaces is hands-on and through an exploration process. One common mistake among participants included which points in a working circuit were appropriate for measuring voltage or current. For example, if participants could not map the schematic of the circuit, it typically translated into a lack of knowledge on what it meant to measure voltage across the power source or across an LED. In our case, RobotAR provided students with important tools that accelerated or guided them through the discovery of these questions.

AR content. The AR animations that had been set up on the robot for the session were used in ways we were not expecting. Those animations were meant to be used as the lecture section to provide follow-along, basic information of circuitry; however, we found out that students were using those animations throughout the workshop to internalize or refresh some of the concepts they had not understood.

"In real life you can't re-play the TA."-P10, who reportedly used the AR to differentiate between capacitors and transistors and how to connect them.

There is a discussion to be had as to how much of the learning gains depend on AR, and why it should be used instead of a different technology (e.g., a video which loads on a website). In our setup, the use of AR was presented in two formats: (1) to provide tutorials for the lecture with demos for students; (2) to provide students with real-time notes/drawings from the instructor. (2) was a feature of our toolkit, enabled by the instructors' UI. This was especially useful, since access to the phone's camera and the toolkit, established the 3D coordinate system of the workspace. With (2), AR superimposes content and provides spatial information corresponding to students' specific workspace and requires no extra steps from instructors. Conversely, (1) is an optional process, since we decided to deliver the laboratory with entirely AR-based content. AR content is supported by the toolkit, but needs to be created in Unity 3D, which makerspace instructors can choose to do. However, students emphasized on the usefulness of being able to replay the content, rather than the format (i.e., AR, video), even if they found voice and visual aid to be helpful. Thus, we would recommend makerspace instructors to focus on creating tutorial content to the best of their abilities, whether in AR or typical video.

Voice Assistant. In most cases, the voice assistant was the go-to tool for participants who had a simple, quick question. For example, *"which leg is my positive side in my LED?"*; *"how do I read a resistor?"*; *"what is voltage?"*. Referred to as a *"first-responder"* (P2), students pointed out that the voice assistant helped them not get too complacent, just get a quick fix, but continue trying to solve their circuits by themselves. Similarly, students reported that it took away the anxiety of asking the "wrong question" or overwhelming the instructor.

"At first I use [the voice AI] because I don't want to rely too much on the TA…because I want to learn, so maybe I want help but not too much."–P7

"The AI helped me to not overload the TA with embarrassing questions. Simple things, [the voice AI] helps you fix."–P1

The effectiveness of the voice assistant is an ongoing process. As the database incorporates more utterances, it will become more accurate at responding to students' questions. Although incorporating more questions and answers into the database is a simple procedure, instructors–who are already in charge of all content creation–may consider whether this is a necessary burden. First, the size of the makerspace is an important detail upon which to take decisions. For example, if a makerspace has 5 instructors and 7 students, then maybe a voice AI assistant to answer questions may not be worth the effort. However, if that same makerspace has 5 instructors and 75 students, then the quantity and quality of available aid will be crucial for a positive learning experience. It should be up to instructors' judgement to decide whether a makerspace requires of the AI voice assistant feature.

Another important feature of makerspaces is brainstorming projects and solutions. This process is synergistic in a physical makerspace, because students are in close proximity, but in a virtual makerspace this is more constrained. One possible solution is for instructors to use a platform (e.g., Slack, Discord) in which students can share, brainstorm, and comment on each others' work. If so, this should take place before or after makerspace hours instead of during, so as to not distract students while they work on their projects. However, we consider the voice assistant for RobotAR–which was used during makerspace hours–to be a proxy for these brainstorming in-person sessions. After all, the AI is crowdsourced from previous sessions with students, and while it does not replace human-to-human interaction or brainstorming, it can become a placeholder to keep students engaged and feel like they are getting community support.

Instructor Teleconsulting. As for the teleconsulting, which was the favorite feature of the robot, students found the AR visual cues provided by the instructor (i.e., arrows, drawings) to be useful and engaging.

"I liked that you can contact the instructor, which is super convenient, because they can show you [the correct answer] in your scene and it's like you never left the lab."–P5

"For me, the instructor [teleconsulting] with the AR is best...it helps to accurately locate something into my view. With [the AR] there is no gap, I don't have to map from his view to my end."–P9

To provide context, the AR demos and the voice assistant were the first-stop tools of most participants. However, there was consensus among students that the teleconsulting feature–either by having the instructor make AR annotations in the students' scenes or by sharing his own camera to do a focused live-demo–was important to understand some difficult concepts that would otherwise make them fall behind. RobotAR, as an intermediary agent for teleconsulting, deviates from current makerspace practices (e.g., Zoom sessions), which require students to double as camera-men (e.g., zooming in, focusing) and creators (i.e., working on their circuits). These dual responsibilities–even with only basic phone functions–were too overwhelming and cumbersome for students. Without the robot, students had to change the position and focus of the camera, which kept their hands busy and unable to follow instructions from the teacher in order to receive timely help. Thus, while they worked on solving their problem, the tablets/phones ended up getting dropped and laying down on the table in disuse.

In terms of the documentation that instructors would typically require from their makerspaces, the lack of physicality would severely hinder instructors' ability to keep track of students' progress. In a physical makerspace, instructors walk around the classroom, glance over students' shoulders, and check progress status. However, in a virtual makerspace, these routine check-ups are difficult without interfering with students' concentration, by asking them to stop and cooperate with focusing/zooming into their workspace. RobotAR removes the need for extra work because the camera repositions according to the students' view or follows along. This is a promising step towards a pathway to have more natural interactions with distance technology, which should be the goal of all makerspaces. Also, this greatly reduces the workload of the students.

Q2: To what extend does the use of RobotAR allow the instructor to offer more on-point instruction and at a higher level during problem-solving?

"It's not just the movement of the robot, it's the voice!"–P4, who emphasized that while he liked how the robot could focus on his workspace, it was the instructor's voice–which could be heard as the robot moved along–which made him feel like the instructor was there next to him.

Several students pointed out that the combination of AR annotation plus voice from the instructor made the class content *"more interesting"* (P10).

The robot mobility and focus capabilities certainly facilitated a higher quality of teleconsulting. Instructor had better access to students' problems, could provide visual cues and notes, and no longer had *"to worry about guiding the student to a particular area, I can use [RobotAR] to focus on what I know I'm looking for."* (Instructor). In this case, the instructor is referring to providing trouble-shooting help. The instructor reported that, for the RobotAR session, questions were not necessarily about problem-solving, but rather to ask for a check-up, more along the lines of: *"Am I doing things correctly?"*–P12. The instructor, who had previously referred to the Zoom session as *"chaotic-fun"*, expressed satisfaction at finding that students were somewhat better prepared in RobotAR condition. While this perceived increase in understanding was probably due to the other tools available (i.e., AR demos, voice AI), the instructor reported that *"it's always easier to help when [the students] get what they're doing"*. With all this in mind, the instructor was enthusiastic about the prospect of using RobotAR in future workshops.

Q3: To what the extend does the use of RobotAR increase instructor's management and presence in the workspace and promote students' engagement and interest?

As reported in the results, there was no statistically significant difference in engagement and interest between conditions. However, mean scores for RobotAR (Engagement, M=4, SD=0.707; Interest, M=3.917, SD=0.954) and Zoom (Engagement, M=3.833, SD=0.8; Interest, M=3.583, SD=0.816) were already fairly high to begin with. While we cannot claim that RobotAR provided an improvement in interest or engagement as opposed to Zoom, it did provide a significant improvement in user experience for several categories: *voice and visual aid from the system, instructor presence, useful aid from instructor, quality of teleconsulting, awareness of instructor, management of workspace.*

As we previously mentioned, the robot added to the teleconsulting experience, helped boost awareness and credibility of instructor and made students feel as if the instructor was next to them. P3 remarked that as *"the instructor was controlling the robot, I felt [the instructor] was here, more like his hands were in my [workspace]."*

It follows that if higher level problem-solving takes place over teleconsulting, then the instructor becomes more credible and the students are more satisfied with the level of workspace management and aid. For example, at different points throughout the experiment, students wanted to get assistance, but the instructor was sometimes busy helping out another student. If at this point, students–seeking assistance–had exhausted the resources (i.e., AR, voice AI), then they either continued problem-solving on their own or became distracted.

Since our voice AI was still limited, then the available support was limited. We logged all students' utterances that were mistakenly classified or not recognized. In the future, our voice AI should continue to recognize a larger set of questions from students. Thus, while we had an engaged set of participants, we need to make sure to always have available resources to keep them concentrated in the work and not lose focus[92].

8. CONTRIBUTIONS, LIMITATIONS AND FUTURE WORK

8.1 Contributions

In this thesis we have accomplished the following contributions-

1. A system architecture specifically designed for virtual makerspace, which introduces effective teleconsulting desktop-based robots into the virtual makerspaces by enabling mobility and translational joints from the robot to help the instructor and the student better focus on areas of interest inside the workspace.

2. A system for creating augmented makerspaces experiences using an AR-compatible robot that behaves as a tutor to the students, and as a versatile agent with access to the physical and the virtual world during teleconsultation.

3. A user study which compares current techniques for distance learning vs. an implementation of our system.

8.2 Limitations

While our current network enables multiple users being part of the session at the same time, the problem-solving through teleconsulting is done in a one-on-one basis. This is due to the need for plane mapping so that the AR can be superimposed on the scene. Only one AR SLAM map is shared per problem-solving session so instructors can only perform a single remote guidance through AR. However, in order to achieve more efficient problem-solving and improve the scalability of the system in the future, we will add a broadcasting option that will allow simultaneous teleconsulting for multiple people by synchronizing multiple SLAM maps across the network. In that case, the instructors can simultaneously place AR content and sketch out annotations on multiple students' virtual workspaces. This is particularly helpful when instructors deem that those students share similar issues and would benefit from a collective instruction session.

Also, our system only utilizes only one of the smartphone's cameras throughout the whole process. However, smartphones with multiple rear cameras are becoming more and more prevalent nowadays. Switching between cameras with different focal lengths will offer instructors more flexibility when they are inspecting a student's workspace. It enables further view of the student's workspace to the instructor's benefit. For instance, when the instructor tries to inspect an electronic component which is placed far away from the robot's current location, he/she would have to manually navigate the robot towards that direction in order to perform a proper examination. Such operation if occurs frequently would no doubt consume the time and energy of the instructor. On the other hand, the instructor can simply zoom in/out if granted the capability to dynamically switch between cameras with different focal lengths. Such an operation is more intuitive and takes less effort to perform. Thus we plan to add this functionality to the toolkit.

Currently, our robot does not have automatic object avoidance capability and relies on the instructor's navigation skill to remote robots. This would be an issue if the workspace of the students becomes cluttered, something that usually occurs during a makerspace that deals with multiple components. Failure to properly navigate the robot will discourage the instructor from performing a detailed inspection since he/she no longer feels confident enough in operating the robot. In the future, we will add the object avoidance capability using the combination of computer vision and external sensors on the robot. When stumbling across an object on its current trajectory, the robot will automatically stop and take out a detour without instructor's intervention. If these technologies prove to be applicable, more advanced navigation capabilities of the robot can be implemented. "Tap-Go" for instance, is the functionality which enables the instructor to navigate the robot to a designated location by simply tapping a place on the screen. In that case, instructors are more willing to move the robot around more frequently, which would be beneficial for the troubleshooting process.

Last but not least, the instructor that participated in our user study points out that the sketch functionality-although useful- is not quite convenient and intuitive to use. After detailed analysis, we determined that it is due to the limitation of the 2D screen upon which the instructor has to draw out the spatial sketch. In our set up the instructor has to first create a surface at a location with a certain angle relative to the desktop. And then the user draws the annotation on this created surface. This step- although is cumbersome- remains necessary if the user wants to do spatial drawing on a screen. In light of the limitation posted by the current hardware, we take a step further by envisioning a setup where instructors can use a VR headset to draw out the 3D annotations as well as other tasks that would be performed more intuitively inside a 3D space. Instead of a mouse and keyboard, they can directly use VR controllers to perform sketches and navigate directly.

8.3 Future Work

Our current AR-enabled robot toolkit is designed for the traditional virtual makerspace where the majority of the tasks focus on development of motor skills. In our user study, we chose an electronics assembly task to test our final design. However, as programming skill is becoming increasingly important for nowday's students, lots of makerspaces have adapted towards involving more programming-oriented tasks. With the goal of developing participants' general computational thinking skills, these makerspaces activities taught basic programming concepts and let participants practice through coding smart IOTs.

To meet the growing demand for teaching programming in a virtual makerspace, we plan to develop a VR programming environment for authoring and customizing smart devices. Novice learners can use it to program the behavior of an individual smart device and its interaction with other devices in the same environment. We will fully utilize the social attribute VR to make into a collective learning experience in which students can work together on a complex project. By inviting students to join a shared virtual environment, we bridge the physical gap between them. Through guided exploration of the virtual world populated with smart devices they programmed, students will acquire general knowledge about smart devices interaction behavior as well as computational thinking skills.

REFERENCES

[1] *Welcome to mooc.org*, https://www.edx.org/, 2020.

[2] *Restless learners change the world*, https//www.mooc.org/, 2020.

[3] *This is future learning*, https://www.futurelearn.com/, 2020.

[4] W. A. Bainbridge, J. Hart, E. S. Kim, and B. Scassellati, "The effect of presence on human-robot interaction," in *RO-MAN 2008-The 17th IEEE International Symposium on Robot and Human Interactive Communication*, IEEE, 2008, pp. 701–706.

[5] *Zoom: Video conferencing, web conferencing, webinars*, https://zoom.us/, 2020.

[6] *Webex*, https://www.webex.com/, 2020.

[7] *Google classroom - manage the classroom with ease*, https://classroom.google.com/h, 2020.

[8] *Skype | communication tool for free calls and chat*, https://www.skype.com/en/, 2020.

[9] *Bert now used on almost every english query*, https://searchengineland.com/google-bert-used-on-almost-every-english-query-342193b, 2020.

[10] W. A. Bainbridge, J. W. Hart, E. S. Kim, and B. Scassellati, "The benefits of interactions with physically present robots over video-displayed agents," *International Journal of Social Robotics*, vol. 3, no. 1, pp. 41–52, 2011.

[11] A. J. Fischer, B. S. Bloomfield, R. R. Clark, A. L. McClelland, and W. P. Erchul, "Increasing student compliance with teacher instructions using telepresence robot problem-solving teleconsultation," *International Journal of School & Educational Psychology*, vol. 7, no. sup1, pp. 158–172, 2019.

[12] D. Leyzberg, A. Ramachandran, and B. Scassellati, "The effect of personalization in longer-term robot tutoring," *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 7, no. 3, pp. 1–19, 2018.

[13] H. Köse, P. Uluer, N. Akalın, R. Yorgancı, A. Özkul, and G. Ince, "The effect of embodiment in sign language tutoring with assistive humanoid robots," *International Journal of Social Robotics*, vol. 7, no. 4, pp. 537–548, 2015.

[14] A. Pereira, C. Martinho, I. Leite, and A. Paiva, "Icat, the chess player: The influence of embodiment in the enjoyment of a game," in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3*, 2008, pp. 1253–1256.

[15] J. Kennedy, P. Baxter, and T. Belpaeme, "Comparing robot embodiments in a guided discovery learning interaction with children," *International Journal of Social Robotics*, vol. 7, no. 2, pp. 293–308, 2015.

[16] D. Leyzberg, S. Spaulding, M. Toneva, and B. Scassellati, "The physical presence of a robot tutor increases cognitive learning gains," in *Proceedings of the annual meeting of the cognitive science society*, vol. 34, 2012.

[17] R. T. Azuma, "A survey of augmented reality," *Presence: Teleoperators & Virtual Environments*, vol. 6, no. 4, pp. 355–385, 1997.

[18] T. H. Chiang, S. J. Yang, and G.-J. Hwang, "Students' online interactive patterns in augmented reality-based inquiry activities," *Computers & Education*, vol. 78, pp. 97–108, 2014.

[19] H.-K. Wu, J. S. Krajcik, and E. Soloway, "Promoting understanding of chemical representations: Students' use of a visualization tool in the classroom," *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching*, vol. 38, no. 7, pp. 821–842, 2001.

[20] E. Beheshti, D. Kim, G. Ecanow, and M. S. Horn, "Looking inside the wires: Understanding museum visitor learning with an augmented circuit exhibit," pp. 1583–1594, 2017.

[21] A. Villanueva, Z. Zhu, Z. Liu, K. Peppler, T. Redick, and K. Ramani, "Meta-ar-app: An authoring platform for collaborative augmented reality in stem classrooms," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–14.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[23] *Jill watson*, https://www.cc.gatech.edu/holiday/jill-watson/, 2020.

[24] a. J. M. Jason Miks, "Keeping the world's children learning through covid-19," 2020, https://www.unicef.org/coronavirus/keeping-worlds-children-learning-through-covid-19.

[25] *Natural language processing (nlp)*, https://www.ibm.com/cloud/learn/natural-language-processing, 2020.

[26] E. D. Liddy, "Natural language processing," 2001.

[27] G. G. Chowdhury, "Natural language processing," *Annual review of information science and technology*, vol. 37, no. 1, pp. 51–89, 2003.

[28] *Understanding searches better than ever before*, https://azure.microsoft.com/en-us/services/cognitive-services/language-understanding-intelligent-service/, 2020.

[29] *Understanding searches better than ever before*, https://blog.google/products/search/search-language-understanding-bert/, 2020.

[30] *Wit.ai*, https://wit.ai/, 2020.

[31] *Language understanding (luis)*, https://wit.ai/, 2020.

[32] *Nvidia jarvis*, https://developer.nvidia.com/nvidia-jarvis, 2020.

[33] C. A. Chapelle and Y.-R. Chung, "The promise of nlp and speech processing technologies in language assessment," *Language Testing*, vol. 27, no. 3, pp. 301–315, 2010.

[34] G. Dietz, J. K. Le, N. Tamer, J. Han, H. Gweon, E. L. Murnane, and J. A. Landay, "Storycoder: Teaching computational thinking concepts through storytelling in a voice-guided app for children," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–15.

[35] A. Edwards, C. Edwards, P. R. Spence, C. Harris, and A. Gambino, "Robots in the classroom: Differences in students' perceptions of credibility and learning between "teacher as robot" and "robot as teacher"," *Computers in Human Behavior*, vol. 65, pp. 627–634, 2016.

[36] V. Rosanda and A. I. Starcic, "The robot in the classroom: A review of a robot role," in *International Symposium on Emerging Technologies for Education*, Springer, 2019, pp. 347–357.

[37] T. Belpaeme, J. Kennedy, P. Baxter, P. Vogt, E. E. Krahmer, S. Kopp, K. Bergmann, P. Leseman, A. C. Küntay, T. Göksun, *et al.*, "L2tor-second language tutoring using social robots," in *Proceedings of the ICSR 2015 WONDER Workshop*, 2015.

[38] T. Belpaeme, P. Vogt, R. Van den Berghe, K. Bergmann, T. Göksun, M. De Haas, J. Kanero, J. Kennedy, A. C. Küntay, O. Oudgenoeg-Paz, *et al.*, "Guidelines for designing social robots as second language tutors," *International Journal of Social Robotics*, vol. 10, no. 3, pp. 325–341, 2018.

[39] J. Kory and C. Breazeal, "Storytelling with robots: Learning companions for preschool children's language development," in *The 23rd IEEE international symposium on robot and human interactive communication*, IEEE, 2014, pp. 643–648.

[40] J. M. Kory-Westlund and C. Breazeal, "A long-term study of young children's rapport, social emulation, and language learning with a peer-like robot playmate in preschool," *Frontiers in Robotics and AI*, vol. 6, p. 81, 2019.

[41] J. M. Kory Westlund, S. Jeong, H. W. Park, S. Ronfard, A. Adhikari, P. L. Harris, D. DeSteno, and C. L. Breazeal, "Flat vs. expressive storytelling: Young children's learning and retention of a social robot's narrative," *Frontiers in human neuroscience*, vol. 11, p. 295, 2017.

[42] J. K. Westlund and C. Breazeal, "The interplay of robot language level with children's language learning during storytelling," in *Proceedings of the tenth annual ACM/IEEE international conference on human-robot interaction extended abstracts*, 2015, pp. 65–66.

[43] J. B. Janssen, C. C. van der Wal, M. A. Neerincx, and R. Looije, "Motivating children to learn arithmetic with an adaptive robot game," in *International conference on social robotics*, Springer, 2011, pp. 153–162.

[44] I. S. Huang and J. F. Hoorn, "Having an einstein in class. teaching maths with robots is different for boys and girls," in *2018 13th World Congress on Intelligent Control and Automation (WCICA)*, IEEE, 2018, pp. 424–427.

[45] O. Mubin, C. J. Stevens, S. Shahid, A. Al Mahmud, and J.-J. Dong, "A review of the applicability of robots in education," *Journal of Technology in Education and Learning*, vol. 1, no. 209-0015, p. 13, 2013.

[46] A. Gomoll, S. Šabanović, E. Tolar, C. E. Hmelo-Silver, M. Francisco, and O. Lawlor, "Between the social and the technical: Negotiation of human-centered robotics design in a middle school classroom," *International Journal of Social Robotics*, vol. 10, no. 3, pp. 309–324, 2018.

[47] T. Balch, J. Summet, D. Blank, D. Kumar, M. Guzdial, K. O'hara, D. Walker, M. Sweat, G. Gupta, S. Tansley, *et al.*, "Designing personal robots for education: Hardware, software, and curriculum," *IEEE Pervasive Computing*, vol. 7, no. 2, pp. 5–9, 2008.

[48] A. Chiou, "Teaching technology using educational robotics," in *Proceedings of the Australian conference on science and mathematics education (formerly UniServe Science Conference)*, vol. 10, 2012.

[49] *Nao, the humanoid and programmable robot*, https://www.softbankrobotics.com/emea/en/nao, 2020.

[50] *Robothespian*, https://robots.ieee.org/robots/robothespian/, 2020.

[51] *Bioloid, stem standard robot kit*, https://www.trossenrobotics.com/bioloid-stem-standard-robot-kit.aspx, 2020.

[52] *Baxter, a versatile manufacturing robot*, https://robots.ieee.org/robots/baxter/, 2020.

[53] *Darwin op2 robot*, https://www.robotlab.com/store/darwin-op2-robot, 2020.

[54] *Robot tiro*, https://www.roboticstoday.com/robots/tiro, 2020.

[55] *Keepon, a social robot*, https://robots.ieee.org/robots/keepon/, 2020.

[56] A. J. Hirst, J. Johnson, M. Petre, B. A. Price, and M. Richards, "What is the best programming environment/language for teaching robotics using lego mindstorms?" *Artificial Life and Robotics*, vol. 7, no. 3, pp. 124–131, 2003.

[57] M. Alemi, A. Meghdari, and M. Ghazisaedy, "Employing humanoid robots for teaching english language in iranian junior high-schools," *International Journal of Humanoid Robotics*, vol. 11, no. 03, p. 1 450 022, 2014.

[58] I. Leite, M. McCoy, D. Ullman, N. Salomons, and B. Scassellati, "Comparing models of disengagement in individual and group interactions," in *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, IEEE, 2015, pp. 99–105.

[59] Z.-J. You, C.-Y. Shen, C.-W. Chang, B.-J. Liu, and G.-D. Chen, "A robot as a teaching assistant in an english class," in *Sixth IEEE international conference on advanced learning technologies (ICALT'06)*, IEEE, 2006, pp. 87–91.

[60] A. Ramachandran, S. S. Sebo, and B. Scassellati, "Personalized robot tutoring using the assistive tutor pomdp (at-pomdp)," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8050–8057.

[61] B. Smith and J. Mader, "Do i need a robot?" *The Science Teacher*, vol. 83, no. 1, p. 8, 2016.

[62] A. J. Fischer, E. H. Dart, K. C. Radley, D. Richardson, R. Clark, and J. Wimberly, "An evaluation of the effectiveness and acceptability of teleconsultation," *Journal of Educational and Psychological Consultation*, vol. 27, no. 4, pp. 437–458, 2017.

[63] K. Brownlee, J. R. Graham, E. Doucette, N. Hotson, and G. Halverson, "Have communication technologies influenced rural social work practice?" *British Journal of Social Work*, vol. 40, no. 2, pp. 622–637, 2010.

[64] B. J. Bice-Urbach and T. R. Kratochwill, "Teleconsultation: The use of technology to improve evidence-based practices in rural communities," *Journal of School Psychology*, vol. 56, pp. 27–43, 2016.

[65] J. L. Gibson, R. C. Pennington, D. M. Stenhoff, and J. S. Hopper, "Using desktop videoconferencing to deliver interventions to a preschool student with autism," *Topics in Early Childhood Special Education*, vol. 29, no. 4, pp. 214–225, 2010.

[66] *Kubi telepresence robot*, https://www.kubiconnect.com/, 2020.

[67] *Double robotics - telepresence robot for telecommuters*, https://www.doublerobotics.com/, 2020.

[68] *Ohmni robot - ohmnilabs*, https://ohmnilabs.com/, 2020.

[69] *Live the unbelievable*, https://www.oculus.com/, 2020.

[70] *Microsoft hololens 2, for precise, efficient hands-free work*, https://www.microsoft.com/en-us/hololens, 2020.

[71] *Vuforia: Market-leading enterprise ar*, https://www.ptc.com/en/products/vuforia, 2020.

[72] *The leading platform for creating interactive, real-time content*, https://unity.com/, 2020.

[73] K.-T. Huang, C. Ball, J. Francis, R. Ratan, J. Boumis, and J. Fordham, "Augmented versus virtual reality in education: An exploratory study examining science knowledge retention when using augmented reality/virtual reality mobile applications," *Cyberpsychology, Behavior, and Social Networking*, vol. 22, no. 2, pp. 105–110, 2019.

[74] H. Köse and N. Güner-Yildiz, "Augmented reality (ar) as a learning material in special needs education," *Education and Information Technologies*, vol. 26, no. 2, pp. 1921–1936, 2021.

[75] T. Wells and S. Houben, "Collabar–investigating the mediating role of mobile ar interfaces on co-located group collaboration," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–13.

[76] E. G. de Ravé, F. J. Jiménez-Hornero, A. B. Ariza-Villaverde, and J. Taguas-Ruiz, "Diedricar: A mobile augmented reality system designed for the ubiquitous descriptive geometry learning," *Multimedia Tools and Applications*, vol. 75, no. 16, pp. 9641–9663, 2016.

[77] M. J. Maas and J. M. Hughes, "Virtual, augmented and mixed reality in k–12 education: A review of the literature," *Technology, Pedagogy and Education*, vol. 29, no. 2, pp. 231–249, 2020.

[78] M. Dunleavy, "Design principles for augmented reality learning," *TechTrends*, vol. 58, no. 1, pp. 28–34, 2014.

[79] M. Won, M. Mocerino, K.-S. Tang, D. F. Treagust, and R. Tasker, "Interactive immersive virtual reality to enhance students' visualisation of complex molecules," in *Research and Practice in Chemistry Education*, Springer, 2019, pp. 51–64.

[80] T. A. Mikropoulos and A. Natsis, "Educational virtual environments: A ten-year review of empirical research (1999–2009)," *Computers & Education*, vol. 56, no. 3, pp. 769–780, 2011.

[81] J. Radianti, T. A. Majchrzak, J. Fromm, and I. Wohlgenannt, "A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda," *Computers & Education*, vol. 147, p. 103 778, 2020.

[82] G. Makransky, S. Borre-Gude, and R. E. Mayer, "Motivational and cognitive benefits of training in immersive virtual reality based on multiple assessments," *Journal of Computer Assisted Learning*, vol. 35, no. 6, pp. 691–707, 2019.

[83] NASA, "Nasa systems engineering handbook," *NASA/SP-2007-6105*, 2007.

[84] NASA, "Nasa systems engineering handbook," *NASA/SP-610S*, 1995.

[85] G.Pahl, W.Beitz, J.Feldhusen, and K.-H.Grote, "Engineering design: A systematic approach," *Springer 3rd edition*, 2007.

[86] *Yahboom omniduino*, https://category.yahboom.net/collections/home-arduino/products/omniduino, 2020.

[87] *Base64*, https://developer.mozilla.org/en-US/docs/Glossary/Base64, 2020.

[88] *Zeromq an open-source universal messaging library*, https://zeromq.org/, 2020.

[89] S. Masson, P. Potvin, M. Riopel, and L.-M. B. Foisy, "Differences in brain activation between novices and experts in science during a task involving a common misconception in electricity," *Mind, Brain, and Education*, vol. 8, no. 1, pp. 44–55, 2014.

[90] K. Peppler and D. Glosson, "Stitching circuits: Learning about circuitry through e-textile materials," *Journal of Science Education and Technology*, vol. 22, no. 5, pp. 751–763, 2013.

[91] K. Peppler, K. Wohlwend, N. Thompson, V. Tan, and A. Thomas, "Squishing circuits: Circuitry learning with electronics and playdough in early childhood," *Journal of Science Education and Technology*, vol. 28, no. 2, pp. 118–132, 2019.

[92] A. M. Villanueva, Z. Liu, Z. Zhu, X. Du, J. Huang, K. A. Peppler, and K. Ramani, "Robotar: An augmented reality compatible teleconsulting robotics toolkit for augmented makerspace experiences," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–13.