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

Ana Villanueva[1], **Zhengzhe Zhu**[2], **Ziyi Liu**[1], **Kylie Peppler**[3], **Thomas Redick**[4] **Karthik Ramani**[1,2]
[1]School of Mechanical Engineering
[2]School of Electrical and Computer Engineering
[4]Department of Psychological Sciences
Purdue University, West Lafayatte, IN, USA
{villana, zhu714, liu1362, tredick, ramani}@purdue.edu
[3]School of Information and Computer Sciences
University of California, Irvine, CA, USA
kpeppler@uci.edu

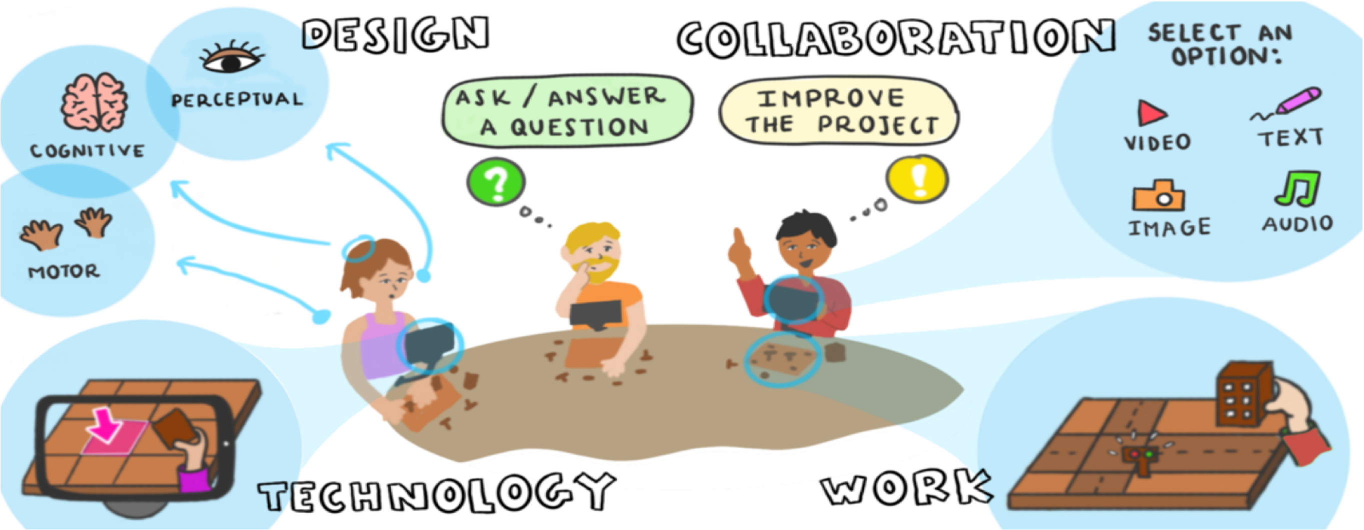


Figure 1. Overview of the four category model from a STEM classroom which implements our Meta-AR-App platform: (a) Design: creating learning content (perceptual, cognitive, motor); (b) Technology: effective learning and problem solving using AR; (c) Collaboration: contributions between instructors and students and improvement of learning content; (d) Work: facilitating and empowering discoveries by manipulating tangibles.

ABSTRACT

Augmented Reality (AR) has become a valuable tool for education and training processes. Meanwhile, cloud-based technologies can foster collaboration and other interaction modalities to enhance learning. We combine the cloud capabilities with AR technologies to present Meta-AR-App, an authoring platform for collaborative AR, which enables authoring between instructors and students. Additionally, we introduce a new application of an established collaboration process, the pull-based development model, to enable sharing and retrieving of AR learning content. We customize this model and create two modalities of interaction for the classroom: local (student to student) and global (instructor to class) pull. Based on observations from our user studies, we organize a four-category classroom model which implements our system: Work, Design, Collaboration, and Technology. Further, our system enables an iterative improvement workflow of the class content and enables synergistic collaboration that empowers students to be active agents in the learning process.

Author Keywords

augmented reality; authoring; classroom; collaboration; Git; pull-based model, version control; STEM; electrical circuitry

INTRODUCTION

Augmented reality (AR), which overlays virtual content into the physical world, offers an entirely new medium for the development and delivery of educational and training content [2]. AR provides students with a unique opportunity of learning-while-making, and enables the acquisition of knowledge through a "hands-on, minds-on" approach [14, 68]. In terms of educational material, AR has been empirically implemented in classrooms for five main applications [70]: (1) collaborative and situated learning by students exploring new interaction modalities in the same environment [10, 12, 15, 34] (e.g., students simultaneously exploring 3D objects in school grounds); (2) selecting and manipulating 3D objects [69] (e.g., look inside the inner-workings of a system); (3) providing students with a social fabric to discuss the learning material

CHI '20, April 25–30, 2020, Honolulu, HI, USA.

ACM ISBN 978-1-4503-6708-0/20/04 ...$15.00.
http://dx.doi.org/10.1145/3313831.3376146

and change attitudes towards real-world issues [9,54,63] (e.g., students exploring an AR environment of melting glaciers); (4) visualizing abstract or invisible concepts [3,35,57] (e.g., pressure, temperature, current flow in a circuit); (5) creating a transition between formal and informal learning [61] (e.g., lecture vs. laboratory experiment). Most of these classroom activities have been programmed/animated using tools such as Unity [65], Unreal Engine [31] or libraries available for programmers such as Google ARCore [40] and Apple ARKit [30]. This process means that considerable coding and animation experience may be necessary to create a customized AR learning experience. Furthermore, these solutions lack the support for ease of creating an educational curriculum, which can usually be a creative and iterative process, and a workflow to foment synergistic collaboration between instructors and students.

Interest-driven classes that merge rigorous concepts from science, technology, engineering, and mathematics (STEM) learning can benefit from a project-based curriculum that emphasizes collaborative inquiry and learning [7,16]. Prior research shows that shyness, fear of appearing unintelligent in front of other students, and lack of academic engagement have been largely at fault for inhibiting in-class participation causing students to drop introductory STEM courses [1,20,56,64]. A collaborative classroom facilitates instructors and students to work together towards solving project-oriented lessons and engaging in different types of interactions. These interactions allow them to answer each other's questions and empower sharing and clarifying the learning content. Thus, it follows that any software targeted to an educational experience has to be tailored towards enabling and moderating these interactions in the classroom. We propose combining AR with the capabilities of cloud technology to introduce the *pull-based collaborative model* [21], a collaboration workflow to upload, share, and download information (i.e., AR content). Students can improve the learning content by adding contributions to the original project created by the instructor. Then, we further customized this workflow to fit the needs of a classroom by creating two types of interaction to moderate the flow of AR content contributions: *local* (i.e., one-to-one student content share) and *global* pull (i.e., instructor approves a student's contribution to the original project given to the class). We envision local pull to be used during class, so that a student's request can be answered by a fellow student, thus relieving some of the burden from the instructor; while global pull can be used after class, when the instructor has had a chance to look through all the contributions made by the students, and determine which ones are the most appropriate to add to the class material. The presence of a moderator (i.e., the instructor) is important to pick the most valuable information to share with the entire class.

We design, develop and assess Meta-AR-App (Figure 1), and present the following contributions from our work:

(1) An **AR-based teaching and learning tool** that supports a STEM educational curriculum by enabling easy-authoring and iterative improvement of class material.

(2)A **collaborative workflow** which leverages cloud technology and supports synergistic interaction modalities between instructors and students inspired by the pull-based development model.

(3) Based on our user studies, we organize a **four category classroom model** for teaching and learning in a classroom with our technology.

RELATED WORK

Our paper is inspired by previous work focused on AR authoring tools and the implementation of the pull-based collaborative development model.

Authoring Tools for AR

Existing platforms such as Unity or Unreal are comprehensive game engines that come with a visual editor and allow assets such as 3D/2D models to be imported and managed [31, 65]. While these platforms are preferred by developers and engineers, educators would require an entire new set of skills, such as coding, modeling, or animation, to author AR content. Thus, creating interactive behavior of the AR assets remains difficult [51]. The vast majority of AR authoring solutions have concentrated on assembly research, ranging from context-aware systems using engineering ontologies [74, 75], automated instructions using computer vision [4–6, 11, 17, 36, 46, 47], linking systems using existing multimedia platforms [24, 58, 66], interaction methods or plugins on top of other platforms [13, 28, 29, 39, 42, 71], and hybrid systems pursuing a combination of the aforementioned [19, 37, 38, 48, 52, 53, 60, 72]. These platforms were constructed to solve specific issues, and to allow different methods of human input in the authoring process. However, given their focalized scope, they allow for limited interaction and scale. Also, they do not support real-time modification of displayed content. Some commercial solutions for general applications, such as Layar, Vuforia Studio or Blippar [25,26,55], have opted for visual interfaces to make the interaction process easy and intuitive, but the capability is limited and isolated, because they are not meant to be deployed for classroom activities. They lack real-time authoring for instructors and students. Furthermore, they do not support an architecture that enables collaboration and interaction among peers. These features are the baseline for an accessible AR-enabled learning environment. For effective learning, an AR platform also needs to subscribe to the rules of multimedia learning, such as supporting the segmentation of information in bite-size pieces, the division of instructions in auditory or visual channels, or the elimination of extraneous material [44]. Meta-AR-App simplifies much of the authoring process by automatically generating animation pathways, segmenting the task information into bite-size pieces, and allowing real-time modification of AR content. Additionally, our platform explores the landscape of project-based STEM classrooms, which means that the AR authoring technology needs to cater to a learning-while-making approach, such as facilitating trial and error authoring efforts and efficient debugging made possible by iterative authoring of learning content.

Collaboration Tools in the Classroom

Collaborative AR technology must be built to integrate the physical environment, while providing the opportunity to share

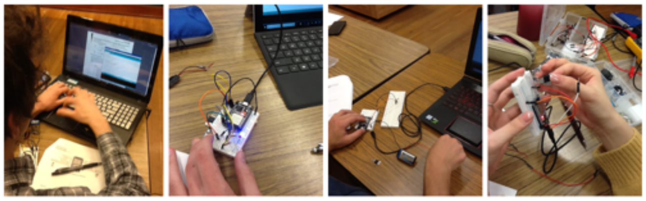


Figure 2. Snapshots taken from circuits and electronics class. We documented students trouble-shooting their circuits.

virtual objects as information, such as annotations, text, videos, and images as supplementary elements between stakeholders (i.e., students, teachers) to investigate their classroom surroundings. However, if we have multiple stakeholders working together collectively and simultaneously on an original project, we need to effectively manage these external contributions to enable conflict resolution by leaving decision-making to a moderator [41].

Other popular non-AR workflows for information sharing in the classrooms have enabled asynchronous [50, 67] or parallel collaboration [62], which has similarly allowed students to read, edit, update content structure between users. However, asynchronous or parallel editing would not work well for an AR classroom [27, 33, 49] because the information would not be time-sensitive and multiple people could simultaneously modify one step of the project, which would create conflict and cause confusion among students (e.g., use of Google Docs).

In open source software (OSS), pull-based development model implements collaboration schemes to streamline the integration of contributions to projects [21]. The pull-based development model became popular within the open source community with platforms such as GitHub, the largest coding repository site for programmers [32]. The typical pull-based model includes: integrators (project creators) who receive contributions from other members (individual software developers) upon pull request and determine whether to merge content based on technical merit or trust [22, 23]. An adaptation of this particular collaboration model would work well in a classroom setup because it is based on version control, which was specifically designed to resolve conflict among multiple changes and multiple stakeholders.

FORMATIVE STUDY AND DESIGN GOALS

To understand how an AR-based platform could support a STEM classroom environment for instructors and students, we shadowed a weekly 3-hour session of a circuits and electronics project-based class in which students built their own robots over a semester period. The observer team was made up of 3 to 4 members, who would take notes and pictures of the class sessions to create a scrapbook of the collected material (Figure 2). The session was led by two graduate teaching assistants with a high level of subject matter expertise, and attended by ten undergraduate sophomore students. Both instructors had prior experiences in creating AR applications, although no augmentation was used in the class. The themes involved voltage, current, basic electronic components, and Internet of Things (IoT) prototyping. The two instructors reported that it took them $\sim$3.5 hours per week to create the student manuals for the class–made up of written and graphical instructions–out of which $\sim$1 hour was spent outlining the content, while $\sim$2.5 hours were spent creating the content. Additionally, we met with the instructors for a 2-hour session prior to class, in which we became familiar with the project manuals. We choose this class because it perfectly illustrates the benefits of augmentation in a spatial task. In the initial classes, we focused on documenting students' behavior, particularly key points at which they were stuck on the material, their realizations and trouble-shooting, and how they interacted with each other and the instructors. Each week we requested the instructors to describe at high level how they would create the learning content using AR and what visual aids (e.g., highlighters, animations, etc.) would improve the experience. Based on our observation of the class, we concluded with a set of design goals to create our system:

Three Main Types of Microtasks in AR Authoring

The instructors identified three types of microtasks to be included inside the steps from the student manuals: (1) visually-oriented microtasks, which can be accomplished by singling out an object (e.g., highlighter) and focusing the student's attention in a particular direction (e.g., arrow), (2) knowledge-oriented microtasks, where the students encode information to understand the instruction, which has to be delivered in a longer format (e.g., textbox, annotation), and (3) spatially-oriented microtasks, which request motor performance and require expert demonstration (e.g., animation, tutorial). These insights were consistent with the human processor model [8], which encodes human processes as perceptual (i.e., visual), cognitive (i.e., knowledge), and motor (i.e., spatial), and inspired the features of our authoring toolbar: each animation step can include a microtask, which can be authored using a suggested set of tools, while the objects can be placed and moved in the scene through our drag-and-drop interface.

An Efficient Collaboration Process

Observations from the class made obvious that students can contribute with new or improved content, especially since not all students work at the same pace, and some students may notice unclear or missing instructions ahead of the rest of the class. We noticed that ahead-of-the-curve students tried to demonstrate their realizations to those around them, but were often impeded due to the information being lost as other students were steps behind and failed to register the aid. A classroom environment is unique in that the students are co-located and that instructors should be capable of moderating the quality of information, so as to filter mistaken additions or changes. We were inspired by the pull-based development process typically used in software development and successfully implemented in software engineering courses [41]. We adapted the process to work in our AR-creation context by providing the instructors with moderation capacity (i.e., decide which content to change or merge), and the students with the option to create and pull content that does not overlap with their already completed work.

High Adaptation for a Variable Environment

Unlike other multimedia tools, AR is heavily dependent on the environment available for exploration. In the class, we

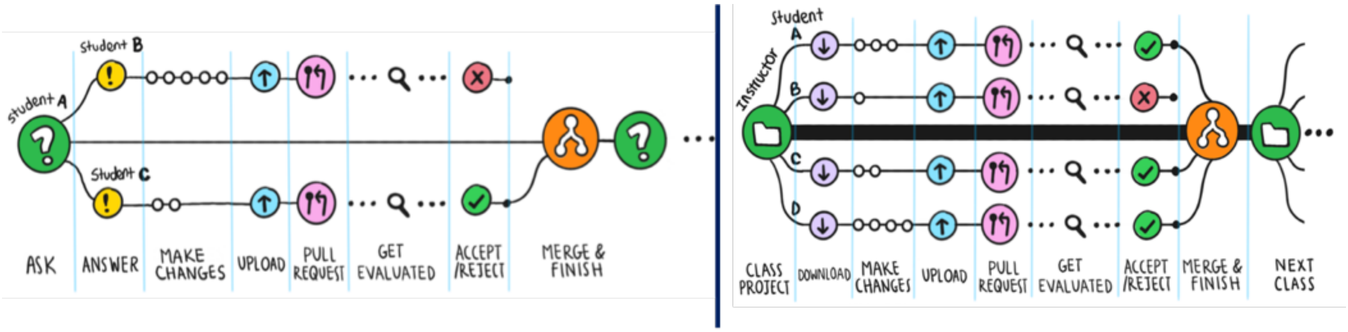


Figure 3. (Left) *Local pull*: Student to student collaboration workflow. (Right) *Global pull*: Workflow for instructors selecting contributions from students to improve their original AR project.

observed how students increased and worked with a wide variety of components, tools, and materials; however, the concepts, knowledge, and rules for working with them, were similar across the board. Thus, in order to save resources and cost, we can recommend instructors working with simple, cheap, and generic components that can be used by students (e.g., a microcontroller), while the augmentation can provide for more complex variations of these components and different phenomena.

THE META-AR-APP FRAMEWORK

Early in the decision process, we realized that the only way to make Meta-AR-App an appealing tool for a classroom environment was to simplify the implementation process, so as to avoid placing an extra burden on the content creator, taking away time from the actual class preparation. Our back-end algorithm allows the user to navigate all the features of the system widgets, which eliminate extraneous steps to create, share, and interact with the learning content. For example, our drag-and-drop interface enables creating an animation by selecting two points from an object to another. This type of animation eliminates the necessity for any lines of code, thus significantly reducing time and workload. Similarly, we borrowed inspiration from existing collaboration processes to achieve a coherent workflow to share and retrieve content.

Collaborative AR Using Pull-Based Development Model

The pull-based model has been widely used for software development and has enabled developers to submit their contributions, typically as source code patches. After contributions have been submitted, they have to be evaluated prior to getting accepted and merged to the existing project. This review process, along with the support for multi-user collaboration, facilitates an attractive model for a classroom environment due to the beneficial presence of a moderator to filter wrong information. The idea is to make the student an active member in the learning process, including becoming a participant in the optimization of the learning content. All students can volunteer–and be rewarded for–their contributions to the class. As such, an active learner becomes a contributor to a network of fellow students and instructors, who are invested in working together towards a similar goal. There are four integral stages in our pull-based development process:

File Management

Unlike common open source projects, which typically include source code files, an AR project includes data files of different formats, such as mp4, jpg, obj, txt and etc. Thus, an effective file management strategy is needed to support the pull-based collaboration process. Drawing inspiration from the file management system in an operating system, we created a structured xml file to store the metadata of every file in the project. These metadata such as file index numbers, file types, creator ID, and many other file attributes, serve as the file handler which can help users keep track of files and perform further operations.

Online Repository Setup

In the context of software development, the online repository contains all the project files created by the moderator (i.e., instructor) and is accessible to the designated group (i.e., members of the class). The instructor pushes/uploads the AR project which was initially created, to the online repository on the cloud and then lets students clone/download it to their local machines for use in the class. The students' local version of the project can be subjected to future changes without affecting the instructor's initial project which is stored online.

Pull Requests

In a typical version control system, pull means downloading and merging the new data to the original project while pull request describes the process where a contributor requests a moderator to pull a contribution. After completing the AR project provided by the instructors, a student can make contributions to it by adding to the parts where he/she thinks more detailed explanations or information are needed. These modifications, which are in the form of text, image, video, or 3D drawing lines, are pushed/uploaded to the cloud for teacher's review. Meanwhile the student sends the pull request which essentially is requesting the teacher to accept his/her contributions.

Contribution Evaluation

Instructors need to verify that the contribution is correct and valuable. Only after the contribution is approved, it can be merged into the original project.

Interaction Modalities

While the traditional pull-based development model lowers the entry barrier for potential contributors (since a pull request can

Figure 4. Main interface of the Meta-AR-App (*Instructor Mode*): (1) Toolbar; (2) Animation Palette; (3) Canvas, (4) Controls, (5) Edit Mode, (6) Collaboration Panel.

be made by anyone), it also increases the burden on integrators, who are responsible for evaluating the proposed changes and integrating them into the main development line [73]. It is particularly true for instructors who serve as the supervisors of the class since they have to both handle pull requests and help out students in need during class time. In order to alleviate the burden of instructors, we introduce an original type of pull-based model, *"local pull"*. Local pull combined with global pull, which is based on the traditional model, are the two types of interaction modalities facilitated by our platform.

Local Pull

Local pull requests are approved by students in need of help and sent by students who volunteer help. Students can help out others by adding explanatory components (e.g., images, video, text and etc.) to the project and sharing them by pull requests. Then, the struggling students can browse the suggestions provided by contributors and choose the most helpful ones to merge while these changes only take effect on their local device. This process, which happens during class without the instructor's involvement, encourages interactions among students while reducing their reliance on instructors.

Global Pull

Global pull requests are approved by the instructor and sent by students. Once the changes are merged, they will take effect globally(i.e, to all the class). Students are only allowed to make a global pull request after they finish the project and these requests are handled by the instructor after class. We implement the global pull to help instructors improve the tutorial which will benefit students from a future class given a new iteration of the learning content.

Figure 3 describes our vision for the customized pull-based collaborative development process. To simplify, we only include three students (*Student A, B, C*) for local pull while one instructor (*Instructor*) and four students (*Student A, B, C, D*) for global pull. However, it is applicable for as many instructors and students as needed.

INTERFACE AND DESIGN RATIONALE

We designed our Meta-AR-App application with specific design goals in consideration: *efficiency*, *accessibility*, and *reusability*. To accomplish these design goals, we needed extensive storyboarding and planning, to ensure that the users could be given all the tools necessary for the creation and access of AR learning content, as well as a coherent process to share and reuse content from contributors. *Efficiency* ensures that every process delivers the user's expectations while investing the least amount of time and effort. For example, if a user wants to create an animation of *object A* moving towards *object B*, rather than make the user trace the path, the system automatically generates the path, upon the selection of each object. *Accessibility* ensures that every feature of the system is cohesively and readily available. If the user requires to post a question or wants to create a specific type of content, then the Panel should easily guide them towards the request. *reusability* ensures that the AR project created by the instructor is reusable for future iterations of the class and to other instructors.

Figure 5. Angle adjustment process: (1) *Object A* and path have the wrong orientation; (2) *Angle Adjustment* tools can be used; (3) *Object A* and path are aligned. Sample objects from our mini smart-city user study.

The main interface of the Meta-AR-App consists of six components (Figure 4). The Toolbar on top presents all the basic tools to manipulate the 3D models. The Animation Palette presents diverse options to provide object behavior, annotations, and tutorials to introduce into the scene. Only one animation option should be active per each step. The Canvas provides space to place the 2D/3D objects, create animations, and add annotations. The Controls allow the users to rewind, or forward the instructions. Edit Mode allows the users to enter the drag-and-drop animation interface to animate the scene. The Collaboration Panel enables the users to participate in the collaboration schemes previously mentioned.

AR Environment Setup

Setup is a prerequisite for interaction with the virtual content within the physical world. This process is necessary to obtain the fiducial markers that are placed in the scene, bind them to a virtual object, or associate data to a position within the environment. The initial setup is enabled by the *Marker Tool* in the Toolbar, which provides access to QR codes. The Toolbar also allows to upload any objects using the *Object Tool* from the local device and to automatically assign each to a marker. Both tools enable distribution across the scene by pressing select + tap on the object or marker. To transform an object, users have to select + drag (*Drag Tool*) the object to a desired location. Then, to rotate, users have to select + rotate (*Rotate Tool*) the object to a desired angle. For precise angle adjustment, users can access the *Angle Adjustment* tools (Figure 5). Objects can be duplicated (select + *Copy Tool* and

Figure 6. Creating Animation Process: (1) select start point from *Object A* and set; (2) select finish point from *Object B* and set; (3) path is generated and instructor can preview animation.

tap + *Paste Tool*), and also deleted (*Delete Tool*). Actions within the virtual environment can be changed by using the *Undo Tool* and *Redo Tool*.

AR Spatial coordinates

Our system enables two different inputs to set the spatial coordinates upon which to overlay the augmentations: (a) QR code tracking, which overlays content directly on tracked object; (b) ground detection, which provides no tracking of objects but sets reference coordinates for AR overlay, and reduces the burden of using multiple QR codes.

Creating Animations

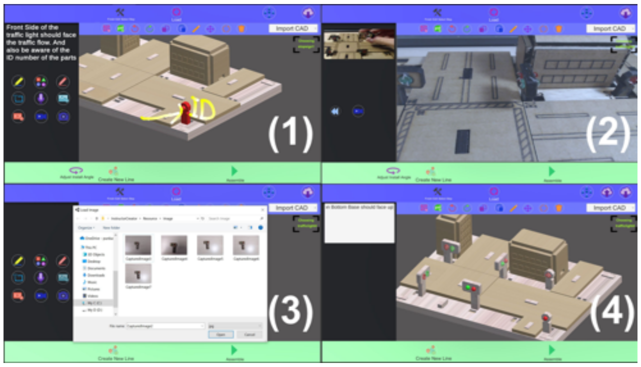
Meta-AR-App allows users to create object animations one at a time in the *Canvas*. Users can select the *Edit Mode* to start the animation (Figure 6). To create a new path from one object moving towards a target object, select + *Set As Moving Object*. Once the target is identified, select + *Set As Target Object*. A path is automatically generated from the object to the target. The animation features include two types of manipulations: (1) transform an object, which allows the users to change the coordinates of the object in the scene, (2) pivot point selection, which allows the path to be generated from a specific point or line from the object towards a specific point or line from the target. The trajectory of the path can be visualized or hidden in the scene, and finally stored.

Authoring Visually-Oriented Microtasks

Visually-oriented microtasks are time sensitive short hints within an animation step that are designed to attract the attention of the user, and deliver visual information. For example, in an animation in which the part of the information is: *If breadboarding situation is to <place a voltage regulator in breadboard>, then <select the voltage regulator LM7805>*. Then, the options available by the *Animation Palette* are: (1) *Highlighter Tool*, which allows users to change the color of the selected object, (2) *Shapes Tool*, which enables users to place a bounding shape surrounding an object, (3) *Draw Tool*, which gives users the capability to draw a sign or figure on the object.

Authoring Knowledge-Oriented Microtasks

Knowledge-oriented microtasks are time sensitive hints to generate, and collect information from users' working memory. This type of information is typically abstract or conceptual, and requires a longer explanation to fit into the overall task.



Front Side of the traffic light should face the traffic flow. And also be aware of the ID number of the parts

Bottom Base should face up

Figure 7. *Animation Palette* examples (from mini-smart city user study): (1) *Highlighter and Draw*; (2) *Take Video* (record); (3) *Diagram* (picture upload); (4) *Annotation* (text). 2D editor as seen from *Instructor Mode* Perspective.

Within the animation, the microtask could be: *If breadboarding situation is to <create a voltage divider>, then <memorize that a voltage divider turns a large voltage into a smaller one by using two series resistors and an input voltage>*. The tools provided by the *Animation Palette* are: (1) *Annotation Tool*, which creates a textbox to deliver a message, (2) *Voice Tool*, which allows to record a voice message, (3) *Diagram Tool*, which allows the user to introduce a diagram or an image into the scene.

Authoring Spatially-Oriented Microtasks

Spatially-oriented microtasks provide just-in-time brief information to properly conduct an operation. For example, in an instruction in which the user needs to perform an action: *If breadboarding situation is to <connect the voltage regulator LM7805 to the microcontroller>, then <connect the output of the LM7805 to an available pin of the microcontroller>*. The suggested tools by the *Animation Palette* are: (1) *Take Picture Tool*, which enables the users to demonstrate an action by an image, (2) *Video Upload Tool*, which allows the users to upload a video with brief instructions, (3) *Take Video Tool*, which gives users the capability to record a mini-tutorial or an example in real time. Since an animation has already been created for each step, these suggested tools may be redundant, and typically recommended for more complex spatial tasks.

Instructor Mode

The functionality of the Animation Palette is similar for our system in *Instructor Mode* (2D editor) and *Student Mode* (3D editor in the physical world) as seen in Figure 7. However, *Instructor Mode* provides instructors with more features to moderate the flow, and quality of the information. Initially, the instructors are in charge of creating the *Original Project*, which is made publicly available to encourage students to make their own contributions. As such, only instructors are given the capability to accept or reject these contributions.

Student Mode

Student Mode allows students to start on the application as the recipients of the content generated by the instructors (Figure 8). Upon cloning the original content, students are given the capability to make modifications, but these contributions can only be accepted upon revision by the instructions. Unlike the

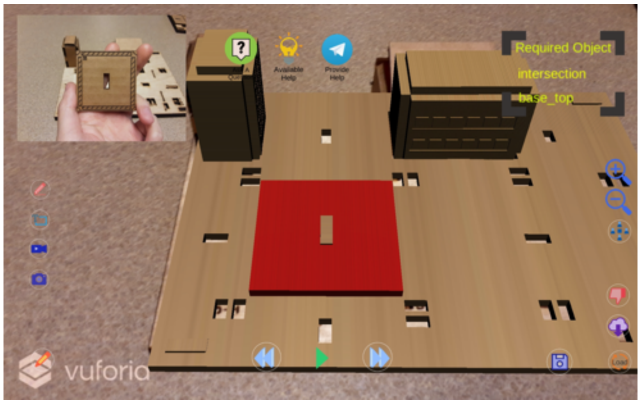


Figure 8. ***Student Mode*** **Main UI.**

Animation Palette from instructor mode, we only kept essential features: *Draw, Annotation, Take Video, and Take Picture*, to avoid too many confusing features (i.e., upload files from local machine) that are unnecessary in real-time collaboration.

IMPLEMENTATION

Meta-AR-App was developed in Unity Game Engine version 2019.3.0a12. We installed our application in Samsung Galaxy Tab A running on Android OS. We built a cloud server to enable file sharing and communication among users. We encoded the metadata of AR project files into an XML file which was uploaded to the sever along with other AR project files in runtime. The server maintained another XML file to keep track of the interactions taking place.

DESIGN USER STUDIES

Controlled User Studies

To create multimedia material for a class (e.g., powerpoint presentations, manuals), instructors initially have to research, review, and outline relevant content before they can proceed to create visualizations or tutorials tailored to the needs of their class. In this controlled user study, we focused on the usability of our system, to evaluate whether the instructors could easily understand and use the platform to compose AR applications. While scripted content may not exactly mimic the complete creation process of an educational AR application, choosing the content of the application enabled us to cover all the concepts and features in our platform. Similarly, we introduced the platform to the students to verify whether they can understand and utilize all the features made available by Meta-AR-App.

Setting and Participants

We recruited 12 participants (8 male, 4 female), and split them into two groups (instructors and students), based on background and experience. The 6 participants selected as instructors (M=24.4 years, SD= 1.63) were current or former teaching assistants with at least one year of teaching experience in STEM classes. We choose instructors from STEM classes for the following reasons: (a) accessibility, (b) electrical circuitry conveniently aligns to the spatial nature of augmented reality, and (c) we wanted to test the application by recreating the curriculum from undergraduate class we shadowed. The 6 participants selected as students (M=21.83 years, SD=2.41) who had completed a minimum of one semester of STEM education courses.

Procedure

We gave each participant a Samsung tablet to complete the tutorial and tasks. The instructors (N=6) were given ’scripts’ (exact procedural paper instructions) of the tasks they were supposed to follow, which were created directly from the first lesson (basic exercises) of the project-based electrical circuitry class we shadowed to inform our design specs. The students (N=6) had to complete the exercises following these instructions, randomly assigned, since all scripts were entirely similar. Each instructor and student received a tutorial lasting approximately 35 minutes on the features of our Meta-AR-App. The participants learned about the main features of our application by following two brief animations between available primitives, which included modifying, customizing, and sharing content. The tutorial included a short walk-through about the capabilities of authoring and collaboration. Then, instructors were in charge of creating an application which included two tasks: (1) Introducing Basic Electronics and Concepts, and (2) Creating a Voltage Divider. The workflow of each task (i.e., an image-based script of the steps), was presented to the instructors, then they were requested to re-create them in AR using our platform. We carefully designed the content to ensure that all the features and functionality of AR authoring were tested. The students received the applications created for them by the instructors, and similarly had to perform a series of predefined modifications, that included all the features of the platform. We provided the context for each task as well as the files (.obj, .png) for the steps that required upload of local sources. Upon completion of the task, participants completed a questionnaire about their experience with Meta-AR-App and sat down with a researcher for a semi-structured interview.

Results

Our 12 participants successfully completed Tasks 1 and 2 with minimal guidance. We presented the instructors and the students with a 5-point Likert scale (1-strongly agree, 5-strongly disagree) to rate their experience using Meta-AR-App. The results we collected are the following: *I consider the tutorial session was sufficient to understand the system*, M=1.58, SD=0.86; *I think the authoring session was enjoyable*, M=2, SD=0.71; *I think the collaboration process was easy*, M=1.9, SD=0.7; *I think my overall experience was enjoyable*, M=2, SD=0.91. The participants were impressed by the ease and the flexibility of our platform: “*[Meta-AR-App] is very easy to use because everything can be created in a matter of drag-and-drop. Even for a non-technical person like me, [Meta-AR-App] allowed me to create timelines for my class. Now, [AR] becomes a new tool I can use in the classroom, especially to keep the students engaged with the material*” (P2). P3 reported a small learning curve, but proceeded to dismiss it: “*[Meta-AR-App] does require basic training, but after becoming familiar with the buttons, gestures, which way things go, I think it was easier. Especially towards the end, when I definitely got the hang of it.*” From our controlled usability studies, our main takeaways were as follows: (a) avoid unnecessary features, which led to the simplified Student Mode we described in the framework section; (b) local-pull contributions should not

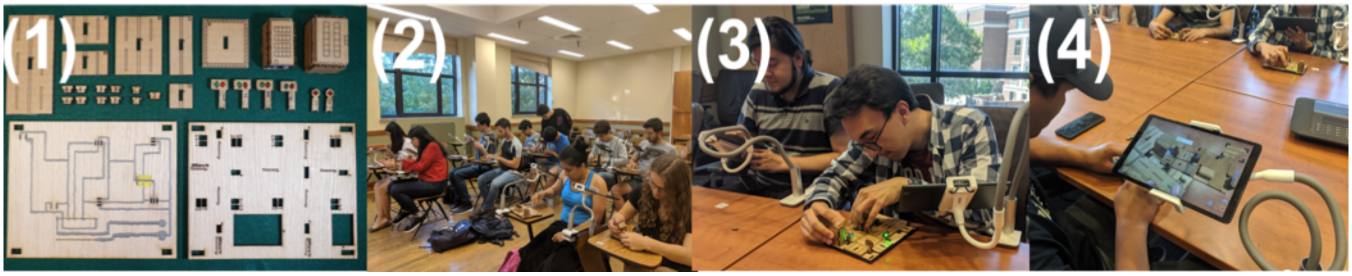


Figure 9. User studies overview: (1) components of the mini-smart city; (2) Meta-AR-App implemented in class session; (3) student debugs the connections in his city; (4) student prepares to contribute using our software.

require instructor's approval but direct student-to-student retrieval, otherwise the burden of selecting correct responses would remain on instructor's shoulders; (c) ground recognition is preferred to multiple QR codes unless individual object tracking is essential, because it provides the AR template with spatial coordinates and saves time for instructors from assigning QR codes to each physical object on the scene. We implemented these takeaways into our system.

Open-Ended User Studies

Setting and Participants

We wanted to investigate open-ended user studies to observe how our system would perform in a real classroom environment (Figure 9). To that end, we recruited 40 undergraduate students (M=23.08 years, SD=2.44), 21 female and 19 male, without a STEM background to participate in a class taught by one instructor from a design-and-tinkering class and an assistant instructor with 2 years of experience on electrical circuitry workshops for underserved youth. After a 30-minute tutorial on the features of the system, the instructors had freedom to design the virtual material for their class using our software. The principal instructor commended the system as *"easy-to-use"* and *"full of potential"*. For this class on IoT development, both choose to teach students how to construct a mini smart-city made out of cardboard material, conductive ink on plywood for the circuit connections, and electrical circuitry components (e.g., LEDs, battery, microcontroller). The objective of the class was to teach students about concepts such as polarity, connections in series and parallel, and current flow. The entire project was comprised of about 25 different steps/actions the users were expected to complete. However, regardless of the condition, they received instructions to 17 of them and were expected to explore and figure out the rest. The instructors designed all the material for the class, including the cardboard pieces for the smart-city, circuitry logic, and the step-by-step AR, explaining the assembly of major components of the task.

Procedure

Each class session lasted approximately two hours and the instructors taught in all sessions. We recorded all classes, and took notes and asked open-ended questions during the sessions to draw our observations. We divided the participants into four classes of ten students per session, each with a different condition: (1) No-AR: a typical class with two instructors; (2) AR-only: a class with AR content created with our software made up of the step-by-step assembly of major components of the smart-city; (3) AR-local: a class with AR content created by the instructors and contributions (e.g., help, hints, answers, suggestions) from student to student (i.e., local pull) using the AR software; (4) AR-local-global: improved AR content based on instructors' selected contributions from students (i.e., global pull) and the option of continuing contributions from student to student (i.e., local pull). For the No-AR condition, the main instructor taught in front of the class, while showing the instructions on assembling the smart-city by using a projector-view of his hands, along with his handling of the components, pausing for the class to catch up, and answering questions. This was the closest way to mimic a classroom and allow students a 3D perspective (e.g., different angles) of the components. After giving the basic instructions, both instructors approached students and helped with debugging.

Results

Table 1 shows the quantitative overall performance of the class per each condition. In order to understand whether an improvement in the AR content warranted less suggestions to the material, we broke down the average contributions of the students in the AR-local (M=2.2,SD=1.66) and the AR-local-global (M=1.7, SD=1.61) and performed a t-test between conditions. The number of contributions for AR-local were statistically significantly higher than for AR-local-global, t(9)=2.24, p=0.02. We also performed a one-way ANOVA to compare the four conditions, in terms of help requests. There was no statistically significant difference between the four conditions as determined by one-way ANOVA (F(3, 36) = 1.12, p = 0.36). As students worked on the smart-city, we evaluated errors by counting how many components or pieces were misplaced or wrongly oriented, each resulting in an "error". Thus, we analyzed the average number of errors for the four conditions. There was a statistically significant difference between groups as determined by one-way ANOVA (F(3,36)=12.37, p=0.00). A Tukey post hoc test revealed that the number of errors was statistically significantly higher for the no-AR condition (5.4±2.46, p=.001) compared to AR-only (1.9±0.94), AR-local (2.3 ± 1.1), and AR-local-global (1.6±0.91). We can observe that introducing AR into the classroom brings a sharp decline in overall error per class during problem-solving. Also, there was a large number of contributions for the AR-local and the AR-local-global conditions from students to their peers were requested and answered or volunteered.

DISCUSSION OF META-AR-APP CLASSROOM MODEL

We consider our platform to be a support tool to teach and learn aspects of STEM subjects–while interacting with 3D virtual and physical objects–,different from the traditional pen-and-pencil methods. Following this new context, we organized

Condition Total #	No-AR	AR-only	AR-local	AR-local-global
Contributions	NA	NA	22	17
Help requests	8	5	10	11
Errors	54	19	23	16

Table 1. Quantitative overall performance of the class per condition.

our class model into *four categories* based on the observations we drew from our user studies, in which each student explored a learning-while-making approach: *Work* (i.e., building the smart-city), *Design* (i.e., creating the content), *Collaboration* (i.e., classroom dynamics), and *Technology* (i.e., using the platform). We evaluated how Meta-AR-App is implemented in a classroom, more specifically how the pull-based model influences all four categories.

Work: Manipulating tangibles

We began the project by clarifying that all students were working towards a goal: every student had to successfully complete the city. Thus, students had to follow some instructions and also, figure out some steps on their own or with help from peers.

Facilitating Discoveries

Along the way, students realized some concepts underlying the task while assembling different pieces. For example, we observed how students would place the battery in the cardboard-based circuit board, then realize that it would not light up the circuit until they flipped the component. Thus, such exploration led to the interpretation of the concept of polarity (i.e., how current flows in one direction for some components). Research has shown that physical manipulatives (tangibles) can support STEM learning [43, 59]. In electrical circuitry, much of the phenomena taking place remains invisible, which can make the learning process difficult. In the no-AR condition this exploration was entirely a trial-and error process, in which these concepts were not always obvious, because other debugging issues in the circuit could be the cause of the circuit not lighting up (e.g., an error in the connections with the conductive ink). Similarly, in the AR-only condition, the instructors did not include AR effects to exemplify current flow; however, the assembly of major components was more straightforward due to the step-by-step instructions. Upon implementation of the pull-based model using the local and local-global conditions, some of the contributions included modifications that emphasized the importance of the direction of current flow (e.g., arrows, drawings) as suggested by the students, which simplified the acquisition of the concept of polarity. While these concepts could also be explained using other media, there is some evidence to suggest that AR provides better results in terms of learning as compared to other media [3, 35].

Design: Creating Learning Content

We observed that our AR system had multiple influences on how instructors created content. We also provided instructors with open mobility to choose how to structure their class and which tasks to choose; although we initially explained that AR technology was particularly salient in tasks that were sequential in nature (e.g., procedures) with phenomena superimposed.

Creating AR in AR

When designing a task for an AR environment, instructors typically need to consider that they are creating the learning content in a 2D environment but that it will be deployed in a 3D environment. The implementation of the pull-based model gave instructors and students a great advantage by enabling the creation of AR content (i.e., 3D models superimposed, 2D images, annotations, shapes, video, and any other technology embedded in a scene or an object) into an AR environment in real-time, pending approval from a moderator. This process of creating AR content in an AR environment enabled new interactions that provided instructors and students with not only spatial information (e.g., navigational cues in the form of annotations, letters to signal the correct orientation/position of an object), but also useful time-based information which related to the amount of time utilized to complete a step/action. For example, the instructors strategically left incomplete steps for the students to figure out. Using the pull-based model, students started contributing AR content on-the-fly and solving inconsistencies or gaps within the original project. This unique interaction made students highly participative as active agents in the learning process.

Collaboration: Between Students and the Class

Students answering questions made by peers

In the No-AR and the AR-only conditions, the burden of debugging each circuit fell almost entirely on the instructors. For example, several students would raise their hands with different concerns, and the instructors would try to assist them one by one, although sometimes this wait period fomented collaboration between the students and their classmates sitting next to them. This type of collaboration was not based on selection, but based on proximity. A common aspect across sessions was that students' first instinct was to refer to the instructors to solve their questions and if the instructor was unavailable, then they asked for help from fellow students, even after the collaborative technology was implemented.

Scalable Help

Once the collaborative model was implemented for the AR-local and AR-local-global conditions, we observed that students providing contributions (e.g., help, hints) were typically the most advanced in the assignment ($\sim$30% of the class). This is different from collaboration conditions based solely on proximity, in that the software allowed for the best students to actively engage in helping the struggling students that were not sitting close to them. For example, contributors often recorded themselves troubleshooting a section of their circuits, took a picture of the orientation of a traffic light, sent an annotation with a recommendation on how to properly connect a component to the circuit board made of plywood. This type of selective collaboration made possible by our collaborative model aided the instructors: by relieving them of the pressure to help students one at a time and by directly providing help to the struggling students that was accurate and timely. The assistant instructor said that she had answered *"more interesting, more challenging questions in session 4 than session 3"*, referring to AR-local-global as the session with more challenging questions. Presumably, this means that as the global pull was implemented for this condition, the learning content was better navigated, thus giving more room for exploration and discovery of the many underlying concepts of the project. Another benefit is that a contribution from one student can be shared with several students as these move forward in the project. This means that help no longer needs to be one-on-one but can be distributed to different students as they access it as needed. One concern with scalability would be how to effectively answer help requests in larger classrooms, in which these requests can be duplicated. We foresee implementing a voting system in which students can upvote the questions they find the most relevant and in need of a prompt response.

Voluntary contributions

An observation from Table 1 is that the number of contributions was higher for both AR local and AR local-global conditions as compared to the number of help requests from students. We found this aspect particularly interesting because students were actively engaged and providing more help than was needed. In both sessions, we observed that students were exploring and stumbling into valuable discoveries, after which they proceeded to share new information with the class. Obviously, not everyone found it relevant at the time, but as students advanced in their projects and caught up, they made use of it.

Technology: Effective Learning and Problem Solving

Efficient debugging by tracing steps

In this category, we include our software and the electrical circuitry components, although the circuitry components are task-dependent and we will emphasize on how the AR technology influences the learning. In the No-AR condition, the only available technologies were the electrical circuits, which empowered students to manipulate components to test ideas or hypotheses (e.g., circuitry concepts). Efficient debugging can lead to learning about working circuits, but it was a slow, painstaking process. Also, exploration beyond this point was limited and dependent on discussion with the class, which was impeded due to instructors being busy helping struggling students. Once the local and local-global conditions were implemented, the debugging process became a collective experience, in which students were contributing with possible ideas on how to solve the circuit.

Iterations can improve the content

The AR-only condition was dependent on the material created by the instructors, which could not be altered since it was in read-only mode, thus exploration was limited to the information that the project provided (e.g., the orientation of small components was not evident, so the concept of current flow was not exemplified by the AR animations). In the AR-local pull, exploration of concepts was facilitated by students helping each other with the debugging process and suggestions to improve the AR content (e.g., added animations or annotations to improve a step, recordings showing how to assemble smaller pieces, arrows to emphasize direction and orientation of a component). The AR-local-global condition was the most student-friendly condition (i.e., the second iteration of the original AR project created by the instructor) mainly because the AR content considerably improved based on contributions made by the students from the AR-local session. Then, as students moved forward with the session, they continued using the technology to follow instructions and also to help each other debugging their circuits.

LIMITATIONS

We included 12 participants for our controlled user studies and 40 students (along with two instructors) for our open-ended user studies, but additional testing is necessary to validate the use of Meta-AR-App across different subjects, classroom dynamics, and accommodations. A semester-long evaluation to analyze the effects and interactions (virtual and in-person) would also provide deeper insights into the role and features of the system, and how it adapts to diverse classrooms. Moreover, it will be interesting to evaluate how multiple iterations of an original project improve the quality of the learning content. We emphasize that Meta-AR-App is a first generation prototype, which means that other features may be added/needed given the large range of classrooms and STEM subjects.

In terms of user scalability, we refer to the potential of a system to handle the growing number of users [18]. Currently, our system is designed to support up to hundreds of concurrent accesses which already exceeds the maximum size of a common class. In the future, if the need for accommodating of a larger class arises, the system can be scaled up by adding more computing resources [45].

CONCLUSION

We presented Meta-AR-App, an authoring platform for collaborative AR. We demonstrated how we can leverage the medium of AR combined with cloud technologies to support selective (i.e., high quality) and timely collaboration, which enables a decrease in error during problem-solving. Apart from these novel interaction modalities, we observed how iterative improvement of the AR learning content (global pull) based on previous contributions made by students (local pull) can improve the original AR project and spark curiosity and creativity among students' learning process.

The next step will be to explore scaling the system to support a community of contributors with reusable templates of project-based AR learning content. The unique aspect of our technology for STEM learning is that it encourages discoveries of complex concepts through a trial-and-error exploration and facilitates effective debugging individually and collectively.

ACKNOWLEDGMENTS

This work was partially supported by the NSF under grants FW-HTF 1839971 and OIA 1937036. We also acknowledge the Feddersen Chair Funds. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agency. We would like to thank Yoshimasa Kitaguchi, Yeliana Torres, Kevin Li, Avinash Ghokale for their help with images, videos, design, and testing. We would also like to thank Iulian Radu for his feedback on this paper.

REFERENCES

[1] Mohd Yusof Abdullah, Noor Rahamah Abu Bakar, and Maizatul Haizan Mahbob. 2012. Student's Participation in Classroom: What Motivates them to Speak up? *Procedia-Social and Behavioral Sciences* 51 (2012), 516–522.

[2] Ronald T Azuma. 1997. A survey of augmented reality. *Presence: Teleoperators & Virtual Environments* 6, 4 (1997), 355–385.

[3] Elham Beheshti, David Kim, Gabrielle Ecanow, and Michael S Horn. 2017. Looking inside the wires: Understanding museum visitor learning with an augmented circuit exhibit. In *Proceedings of the 2017 chi conference on human factors in computing systems*. ACM, 1583–1594.

[4] Bhaskar Bhattacharya. 2016. Automatic generation of augmented reality guided assembly instructions using expert demonstration. (2016).

[5] Bhaskar Bhattacharya and Eliot Winer. 2015. A method for real-time generation of augmented reality work instructions via expert movements. In *The Engineering Reality of Virtual Reality 2015*, Vol. 9392. International Society for Optics and Photonics, 93920G.

[6] Bhaskar Bhattacharya and Eliot H Winer. 2019. Augmented reality via expert demonstration authoring (AREDA). *Computers in Industry* 105 (2019), 61–79.

[7] Josh Brown, Ryan Brown, and Chris Merrill. 2011. Science and technology educators' enacted curriculum: Areas of possible collaboration for an integrative STEM approach in public schools. *Technology and Engineering Teacher* 71, 4 (2011), 30.

[8] Stuart K Card, Allen Newell, and Thomas P Moran. 1983. The Psychology of Human-Computer Interaction. (1983).

[9] Hsin-Yi Chang, Hsin-Kai Wu, and Ying-Shao Hsu. 2013. Integrating a mobile augmented reality activity to contextualize student learning of a socioscientific issue. *British Journal of Educational Technology* 44, 3 (2013), E95–E99.

[10] Tosti HC Chiang, Stephen JH Yang, and Gwo-Jen Hwang. 2014. Students' online interactive patterns in augmented reality-based inquiry activities. *Computers & Education* 78 (2014), 97–108.

[11] Jinhyuk Choi, Youngsun Kim, Myonghee Lee, Gerard J Kim, Yanghee Nam, and Yongmoo Kwon. 2010. k-MART: Authoring tool for mixed reality contents. In *2010 IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 219–220.

[12] Manuel Condado, Isabel Morais, Ryan Quinn, Sahil Patel, Patricia Morreale, Ed Johnston, and Elizabeth Hyde. 2019. Integrating Historical Content with Augmented Reality in an Open Environment. In *International Conference on Human-Computer Interaction*. Springer, 196–205.

[13] S Coquillart, M Göbel, and others. 2004. Authoring of mixed reality applications including multi-marker calibration for mobile devices. In *Eurographics Symposium on Virtual Environments (2004)*. 1–9.

[14] National Research Council and others. 2012. *A framework for K-12 science education: Practices, crosscutting concepts, and core ideas*. National Academies Press.

[15] Matt Dunleavy, Chris Dede, and Rebecca Mitchell. 2009. Affordances and limitations of immersive participatory augmented reality simulations for teaching and learning. *Journal of science Education and Technology* 18, 1 (2009), 7–22.

[16] Mesut Duran, Margret Höft, Dan B Lawson, Brahim Medjahed, and Elsayed A Orady. 2014. Urban high school students' IT/STEM learning: Findings from a collaborative inquiry-and design-based afterschool program. *Journal of Science Education and Technology* 23, 1 (2014), 116–137.

[17] Daniel Eckhoff, Christian Sandor, Christian Lins, Ulrich Eck, Denis Kalkofen, and Andreas Hein. 2018. TutAR: augmented reality tutorials for hands-only procedures. In *Proceedings of the 16th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*. ACM, 8.

[18] Hesham El-Rewini and Mostafa Abd-El-Barr. 2005. *Advanced computer architecture and parallel processing*. Vol. 42. John Wiley & Sons.

[19] Barrett Ens, Fraser Anderson, Tovi Grossman, Michelle Annett, Pourang Irani, and George Fitzmaurice. 2017. Ivy: Exploring spatially situated visual programming for authoring and understanding intelligent environments. In *Proceedings of the 43rd Graphics Interface Conference*. Canadian Human-Computer Communications Society, 156–162.

[20] Josephine A Gasiewski, M Kevin Eagan, Gina A Garcia, Sylvia Hurtado, and Mitchell J Chang. 2012. From gatekeeping to engagement: A multicontextual, mixed method study of student academic engagement in introductory STEM courses. *Research in higher education* 53, 2 (2012), 229–261.

[21] Georgios Gousios, Martin Pinzger, and Arie van Deursen. 2014. An exploratory study of the pull-based software development model. In *Proceedings of the 36th International Conference on Software Engineering*. ACM, 345–355.

[22] Georgios Gousios, Margaret-Anne Storey, and Alberto Bacchelli. 2016. Work practices and challenges in pull-based development: the contributor's perspective. In *Software Engineering (ICSE), 2016 IEEE/ACM 38th International Conference on*. IEEE, 285–296.

[23] Georgios Gousios, Andy Zaidman, Margaret-Anne Storey, and Arie Van Deursen. 2015. Work practices and challenges in pull-based development: the integrator's perspective. In *Proceedings of the 37th International Conference on Software Engineering-Volume 1*. IEEE Press, 358–368.

[24] Paul Grimm, Michael Haller, Volker Paelke, Silvan Reinhold, Christian Reimann, and R Zauner. 2002. AMIRE-authoring mixed reality. In *Augmented Reality Toolkit, The First IEEE International Workshop*. IEEE, 2–pp.

[25] Blippar Group. 2019a. *Blippar*. `https://www.blippar.com/`

[26] Blippar Group. 2019b. *Layar*. `https://www.layar.com/`

[27] Jens Grubert, Tobias Langlotz, and Raphaël Grasset. 2011. Augmented reality browser survey. *Institute for computer graphics and vision, University of Technology Graz, technical report* 1101 (2011).

[28] Sinem Guven, Steven Feiner, and Ohan Oda. 2006. Mobile augmented reality interaction techniques for authoring situated media on-site. In *2006 IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE, 235–236.

[29] Taejin Ha, Mark Billinghurst, and Woontack Woo. 2011. An interactive 3D movement path manipulation method in an augmented reality environment. *Interacting with Computers* 24, 1 (2011), 10–24.

[30] Apple Inc. 2019a. *ARKit2*. `https://developer.apple.com/arkit/`

[31] Epic Games. Inc. 2019b. *Unreal Engine*. `https://www.unrealengine.com/`

[32] GitHub Inc. 2019c. *GitHub*. `https://github.com/`

[33] Andrew Irlitti, Ross T Smith, Stewart Von Itzstein, Mark Billinghurst, and Bruce H Thomas. 2016. Challenges for Asynchronous Collaboration in Augmented Reality. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*. IEEE, 31–35.

[34] Eric Klopfer and Kurt Squire. 2008. Environmental Detectives the development of an augmented reality platform for environmental simulations. *Educational technology research and development* 56, 2 (2008), 203–228.

[35] Pascal Knierim, Francisco Kiss, and Albrecht Schmidt. 2018. Look Inside: Understanding Thermal Flux Through Augmented Reality. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE, 170–171.

[36] Tobias Langlotz, Mathäus Zingerle, Raphael Grasset, Hannes Kaufmann, and Gerhard Reitmayr. 2012. AR record&replay: situated compositing of video content in mobile augmented reality. In *Proceedings of the 24th Australian Computer-Human Interaction Conference*. ACM, 318–326.

[37] Gun A Lee, Gerard J Kim, and Mark Billinghurst. 2005. Immersive authoring: What you experience is what you get (wyxiwyg). *Commun. ACM* 48, 7 (2005), 76–81.

[38] Gun A Lee, Claudia Nelles, Mark Billinghurst, Mark Billinghurst, and Gerard Jounghyun Kim. 2004. Immersive authoring of tangible augmented reality applications. In *Proceedings of the 3rd IEEE/ACM international Symposium on Mixed and Augmented Reality*. IEEE Computer Society, 172–181.

[39] Gun A Lee, Ungyeon Yang, Yongwan Kim, Dongsik Jo, Ki-Hong Kim, Jae Ha Kim, and Jin Sung Choi. 2009. Freeze-Set-Go interaction method for handheld mobile augmented reality environments. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*. ACM, 143–146.

[40] Google LLC. 2019. *ARCore*. `https://developers.google.com/ar/`

[41] Yao Lu, Xinjun Mao, Gang Yin, Tao Wang, and Yu Bai. 2017. Using Pull-Based Collaborative Development Model in Software Engineering Courses: A Case Study. In *International Conference on Database Systems for Advanced Applications*. Springer, 399–410.

[42] Blair MacIntyre, Maribeth Gandy, Steven Dow, and Jay David Bolter. 2004. DART: a toolkit for rapid design exploration of augmented reality experiences. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*. ACM, 197–206.

[43] Taylor Martin and Daniel L Schwartz. 2005. Physically distributed learning: Adapting and reinterpreting physical environments in the development of fraction concepts. *Cognitive science* 29, 4 (2005), 587–625.

[44] Richard E Mayer and Roxana Moreno. 2003. Nine ways to reduce cognitive load in multimedia learning. *Educational psychologist* 38, 1 (2003), 43–52.

[45] Maged Michael, Jose E Moreira, Doron Shiloach, and Robert W Wisniewski. 2007. Scale-up x scale-out: A case study using nutch/lucene. In *2007 IEEE International Parallel and Distributed Processing Symposium*. IEEE, 1–8.

[46] Peter Mohr, Bernhard Kerbl, Michael Donoser, Dieter Schmalstieg, and Denis Kalkofen. 2015. Retargeting technical documentation to augmented reality. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 3337–3346.

[47] Peter Mohr, David Mandl, Markus Tatzgern, Eduardo Veas, Dieter Schmalstieg, and Denis Kalkofen. 2017. Retargeting Video Tutorials Showing Tools With Surface Contact to Augmented Reality. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 6547–6558.

[48] Katharina Mura, Nils Petersen, Markus Huff, and Tandra Ghose. 2013. IBES: a tool for creating instructions based on event segmentation. *Frontiers in psychology* 4 (2013), 994.

[49] Alaeddin Nassani, Huidong Bai, Gun Lee, and Mark Billinghurst. 2015. Tag it!: AR annotation using wearable sensors. In *SIGGRAPH Asia 2015 Mobile Graphics and Interactive Applications*. ACM, 12.

[50] Emmanuel Navarro, Franck Sajous, Bruno Gaume, Laurent Prévot, Hsieh ShuKai, Kuo Tzu-Yi, Pierre Magistry, and Huang Chu-Ren. 2009. Wiktionary and NLP: Improving synonymy networks. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*. Association for Computational Linguistics, 19–27.

[51] Michael Nebeling and Maximilian Speicher. 2018. The Trouble with Augmented Reality/Virtual Reality Authoring Tools. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE, 333–337.

[52] Jong-Seung Park. 2011. AR-Room: a rapid prototyping framework for augmented reality applications. *Multimedia tools and applications* 55, 3 (2011), 725–746.

[53] Nils Petersen and Didier Stricker. 2012. Learning task structure from video examples for workflow tracking and authoring. In *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on*. IEEE, 237–246.

[54] Luke Phipps, Victor Alvarez, Sara de Freitas, Kevin Wong, Michael Baker, and Justin Pettit. 2016. Conserv-AR: A virtual and augmented reality mobile game to enhance students' awareness of wildlife conservation in Western Australia. *Mobile Learning Futures–Sustaining Quality Research and Practice in Mobile Learning* (2016), 214.

[55] PTC. 2019. *Vuforia Studio*. `https://studio.vuforia.com/`

[56] Kelly A Rocca. 2010. Student participation in the college classroom: An extended multidisciplinary literature review. *Communication education* 59, 2 (2010), 185–213.

[57] Hannu Salmi, Helena Thuneberg, and Mari-Pauliina Vainikainen. 2017. Making the invisible observable by Augmented Reality in informal science education context. *International Journal of Science Education, Part B* 7, 3 (2017), 253–268.

[58] Dieter Schmalstieg, Anton Fuhrmann, Gerd Hesina, Zsolt Szalavári, L Miguel Encarnaçao, Michael Gervautz, and Werner Purgathofer. 2002. The studierstube augmented reality project. *Presence: Teleoperators & Virtual Environments* 11, 1 (2002), 33–54.

[59] Bertrand Schneider, Jenelle Wallace, Paulo Blikstein, and Roy Pea. 2013. Preparing for future learning with a tangible user interface: the case of neuroscience. *IEEE Transactions on Learning Technologies* 6, 2 (2013), 117–129.

[60] Hartmut Seichter, Julian Looser, and Mark Billinghurst. 2008. ComposAR: An intuitive tool for authoring AR applications. In *Proceedings of the 7th IEEE/ACM international symposium on mixed and augmented reality*. IEEE Computer Society, 177–178.

[61] Sofoklis Sotiriou and Franz X Bogner. 2008. Visualizing the invisible: augmented reality as an innovative science education scheme. *Advanced Science Letters* 1, 1 (2008), 114–122.

[62] Andrew D Spaeth and Roderick S Black. 2012. Google Docs as a form of collaborative learning. (2012).

[63] Kurt D Squire and Mingfong Jan. 2007. Mad City Mystery: Developing scientific argumentation skills with a place-based augmented reality game on handheld computers. *Journal of science education and technology* 16, 1 (2007), 5–29.

[64] Jeffrey R Stowell, Terrah Oldham, and Dan Bennett. 2010. Using student response systems ("clickers") to combat conformity and shyness. *Teaching of Psychology* 37, 2 (2010), 135–140.

[65] Unity Technologies. 2019. *Unity Game Engine*. `https://unity3d.com/`

[66] Jayzon F Ty, Ma Mercedes T Rodrigo, and Marc Ericson C Santos. 2014. A Mobile Authoring Tool for AR Content Generation Using Images as Annotations. *Philippine Information Technology Journal* 7, 1 (2014), 61–70.

[67] Jakob Voss. 2005. Measuring wikipedia. (2005).

[68] Tory Williams, Jacqueline Krikorian, Jonathan Singer, Christopher Rakes, and Julia Ross. 2019. A High Quality Educative Curriculum in Engineering Fosters Pedagogical Growth. *International Journal of Research in Education and Science* 5, 2 (2019), 657–680.

[69] Hsin-Kai Wu, Joseph S Krajcik, and Elliot Soloway. 2001. 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* 38, 7 (2001), 821–842.

[70] Hsin-Kai Wu, Silvia Wen-Yu Lee, Hsin-Yi Chang, and Jyh-Chong Liang. 2013. Current status, opportunities and challenges of augmented reality in education. *Computers & education* 62 (2013), 41–49.

[71] Han Kyu Yoo and Jong Weon Lee. 2014. Mobile augmented reality system for in-situ 3D modeling and authoring. In *2014 International Conference on Big Data and Smart Computing (BIGCOMP)*. IEEE, 282–285.

[72] Jeongmin Yu, Jin-u Jeon, Gabyong Park, Hyung-il Kim, and Woontack Woo. 2016. A Unified Framework for Remote Collaboration Using Interactive AR Authoring and Hands Tracking. In *International Conference on Distributed, Ambient, and Pervasive Interactions*. Springer, 132–141.

[73] Yue Yu, Huaimin Wang, Vladimir Filkov, Premkumar Devanbu, and Bogdan Vasilescu. 2015. Wait for it: determinants of pull request evaluation latency on GitHub. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*. IEEE, 367–371.

[74] J Zhu, SK Ong, and AYC Nee. 2013. An authorable context-aware augmented reality system to assist the maintenance technicians. *The International Journal of Advanced Manufacturing Technology* 66, 9-12 (2013), 1699–1714.

[75] J Zhu, Soh-Khim Ong, and Andrew YC Nee. 2015. A context-aware augmented reality assisted maintenance system. *International Journal of Computer Integrated Manufacturing* 28, 2 (2015), 213–225.