

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

5,800

Open access books available

142,000

International authors and editors

180M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Virtual Robotics in Hybrid Teaching and Learning

Sharon Mistretta

Abstract

Traditional robotics instruction in face-to-face classrooms, after-school clubs, and independent competition environments align with expensive, physical robot kits shared by students. Students or parent groups often elect themselves because of previous experience, expertise, or perceived technical ability to dominate the physical robotic platforms' planning, engineering, building, and subsequent programming. This self-elected grabbing of control leaves students who are not regarded as well-positioned to contribute sidelined to observe the self-appointed experts of the group. Virtual robotics platforms provide educators and coaches with the unique opportunity to give every student access to a robot. Each student learns programming, math, and scientific forces that impact robots through simulated physics algorithms. With their customizable virtual environments, virtual robotics platforms such as Vex VR and Robot Virtual Worlds level the playing field. All students can learn, practice, and subsequently contribute to robotics-centered group projects or competitive teams in meaningful ways. This book chapter delineates the strategies to implement virtual robotics in hybrid classroom environments supported by the Technological Pedagogical Content Knowledge (TPACK) framework. Additionally, this chapter reviews how computer-aided design and augmented reality platforms provide students with the opportunity to incorporate 3D objects into virtual worlds.

Keywords: virtual robotics, hybrid teaching environments, TPACK framework, computer-aided design, augmented reality

1. Introduction

Robotics courses, after-school programs, and teams are highly sought-after by school districts and parents who wish to provide their students with science, technology, engineering, and math (STEM) instruction to cultivate a foothold for children in future STEM majors and careers [1–3]. Acquiring physical robotic kits, tools, building and testing space, storage units, computer equipment, and software can be an expensive and time-consuming proposition. Funding for this scope of sustained classroom robotics ranges from well-organized parent-teacher committees [4], business sponsors in return for advertisements on t-shirts [5], and grants from non-profit robot competition entities such as Vex Robotics Education Competition Foundation [6].

The ratio of student-to-robot varies as children enroll in a class, program, or team. A small classroom bundle of robots, such as the Vex IQ, provides five kits, a 12-tile

playfield, 18 generic game objects, storage bins, five “pin tools,” and costs approximately \$2250 [7]. In a classroom, club, or team of 20 students, this kit provides a 4:1 ratio of students to a robot. Classrooms and clubs can implement a generic “build” such as the Vex IQ Clawbot robot to practice fundamental robotics. Competitions, which change every year, require a custom build to enact the unique game established as a challenge by non-profit organizations such as Vex or FIRST Robotics. A team wishing to practice-to-win with their uniquely engineered robot could invest in a competition-size playfield and purchase the new game pieces each year.

With the physical requirements of robotics established, this still leaves teachers and parents with the imperative to provide all students with the opportunity to plan, engineer, build, program, and test a robot. Every physical robotic platform is a synthesis of hardware, software, and firmware. The definition of each of these terms is as follows:

- Hardware is the collection of the physical components of the robot. The robot’s brain, sometimes called a brick, contains ports that connect to motors and sensors using cables of varying lengths and memory boards to house software instructions downloaded from the students’ shared computer.
- Software is a general term that describes computer programs. The programs include the operating system that resides in the robot’s brain to communicate with the programs we write in C, C++, or Python to instruct the robot to move and sense its surroundings.
- Firmware is the collection of instructions in the brain, motors, and sensors that allow the hardware to communicate and enact our programs.

Teachers, parents, and coaches are frequently at a loss of where to begin. This chapter delineates the implementation of virtual robotics as an on-ramp to familiarize educators and their stakeholders with the fundamentals of programming a robot to navigate a virtual world through a simulated physics algorithm. Virtual reality (VR) is a technology that immerses the user into a coded environment that employs visual output to depict different surroundings other than the real world. VR is typically associated with a headset such as Google Cardboard [8] or Merge VR headsets [9] that the user wears to block out the real world and experience new visual input. Vex VR and Robot Virtual Worlds are examples of virtual worlds that depict robots in a computer-generated environment without the use of goggles or headsets. The user observes the robot in an environment on their computer screen.

Virtual robotics provides educators, parents, and coaches with a 1:1 learner to robot ratio. Adults supporting robots in the classrooms, clubs, and competitions should understand the world of robots before making a substantial financial and time investment in physical kits, dedicated building & testing space, and the logistics to field a competition-ready team.

The following sections employ the technological pedagogical and content knowledge (TPACK) framework [10] that delineates the necessary knowledgebase to understand the *intersections* of teaching methods and content knowledge (**Figure 1**) to instruct with technology effectively. Robotics brings science and math instruction to the forefront as students must understand scientific *content* such as force and friction and mathematical concepts such as circle geometry. *Pedagogy* embodies the methods and teaching practices of the component disciplines of science, technology, and math. The ultimate intersection of technology, pedagogy, and content knowledge (TPACK) provides a solid on-ramp to robotics.

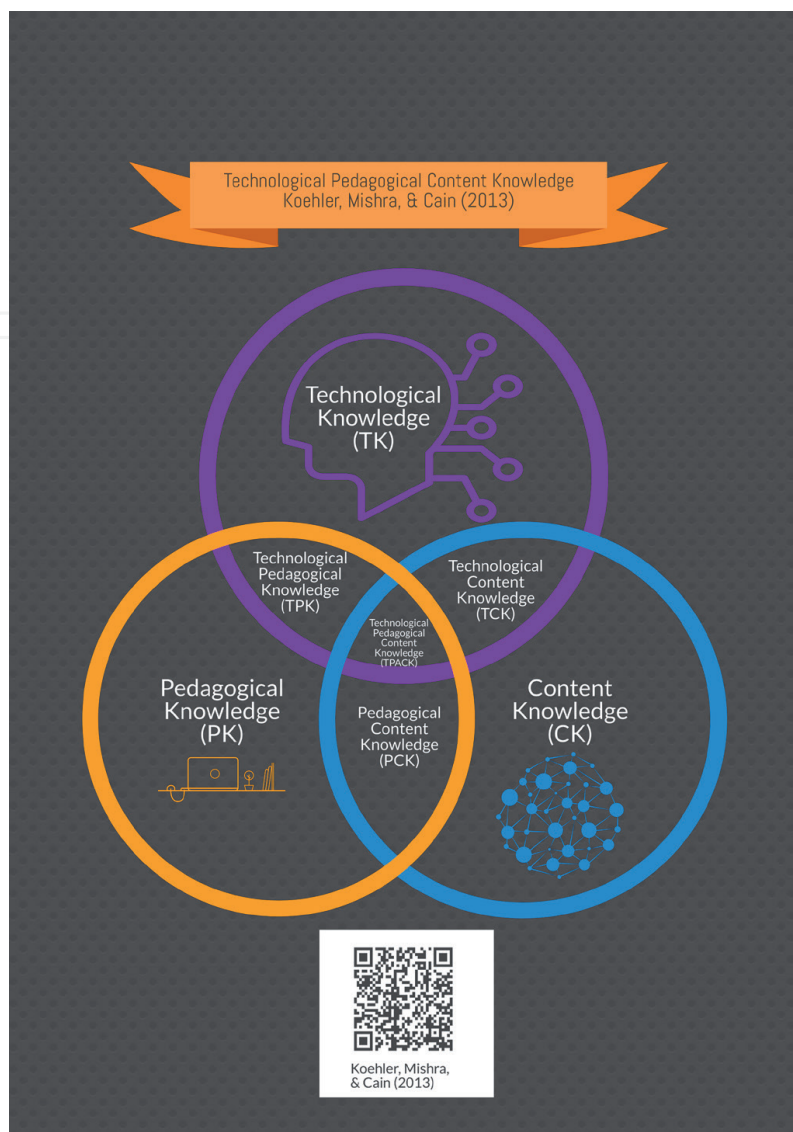


Figure 1.
 Diagram of the intersections of the TPACK framework.

Using TPACK as a framework, the remaining sections of this chapter discuss the following virtual robotics platforms and supporting applications:

- Vex VR [11], a no-cost, browser-based virtual robotic platform.
- Robot Virtual Worlds [12], a licensed software package that includes a customizable virtual world called Level Builder.
- Tinkercad [13] – a computer-aided design (CAD) 3D modeling program offered by Autocad where students create objects to upload “.stl” files to Robot Virtual Worlds’ [12] Level Builder.
- Merge Cube [14] – an augmented reality mobile application program that facilitates viewing student-designed 3D objects to “see” their creation before uploading to Level Builder or printing on a 3D printer.

This chapter concludes with a summary of virtual robotics and suggested transitions using a hybrid approach to virtual and physical robotics.

2. Vex VR

Vex VR is a browser-based virtual robot platform provided by the Vex Robotics Education & Competition (REC) Foundation. This platform is an ideal place to introduce students, teachers, parents, and coaches to the world of robotics. Educators can consider the VR Vex platform as a tool to use with students in face-to-face and synchronous or asynchronous online environments. The goal is to provide *all* students with the *same* learning opportunities. It is essential to consider how educators can use platforms such as Google Classroom [15] and Google Drive [16] to deliver supporting content, such as worksheets, to students. Online materials become accessible to all hybrid learners in the same room, joining via Google Meet [17], or who must enact the lessons when they have access to a shared home computer. Creating multiple entry points for students in a hybrid approach is the ultimate in student-centered learning.

With the hybrid “classroom” organization established, let us define programming. By this author’s definition, programming is writing instructions to cause an object made of plastic, glass, rare metals, and electricity to solve logical and mathematical statements repetitively.

There are several ground rules for every programmer to consider as they journey into this fantastic field of coding and robotics. First and foremost, the program is doing what it is doing because that is what you told it to do. Programmers must think of how the computer interprets our instructions, not how we believe the code should work. Next, concise code is best. Programming is not a competition to write the most lines of code. Concise lines of instructions take up less of the computer’s memory and will run faster. Persistent programmers write great programs. Finish a job. Be proud of your product. Finally, share what you know. Robotics communities have robust collaborative forums and videos on social media. Share techniques to help up-and-coming teams.

2.1 Vex VR technological knowledge (TK)

This section addresses the technical knowledge in the TPACK framework (**Figure 1**) necessary to implement the fundamentals of Vex VR. Since Vex VR is browser-based, there is no need to download software to each student’s computer. A *browser* is software on your computer that communicates with a website’s *server*, an extensive array of disk drives, that responds to your interactions when connected to the internet. Browsers such as Google Chrome [18] create a local, temporary file on each computer called *cache*. However, this temporary file resides on an individual computer and does not necessarily store a student’s Vex VR “vrblocks” file when the computer shuts down. The Vex VR software does allow students to export their developing program to the hard drive of their computer. At this juncture, educators should consider using Google Classroom [15] or the Google Drive [16] of their Gmail account to create folders for each student to upload their exported “vrblocks” work-in-progress to their Google folder *each time* they finish a programming session. Using Google Drive to store files is particularly helpful if two students used the same device on different days or if a student is returning to school after working from home. Creating this workflow organization and reminding students to save, export, and upload their work to their Google folder saves time and frustration. Teach the students to rename each upload of their “vrblocks” file with their name and date. For instance, Mistretta110921.vrblocks. Naming the file avoids confusion with prior iterations of the program and improves students’ organizational skills. Google Drive [16] is capable of housing the “vrblocks”

file. One must download the file to your computer and import it to Vex VR to continue to program.

2.2 Vex VR content knowledge: science and math (CK)

This section addresses the science and math content knowledge in the TPACK framework (**Figure 1**) that underpin the natural forces and numeracy at work. Robotics is the glue that holds STEM together. The science topic of ultrasonic sound-waves correlates to the distance sensors on robots. The Vex VR software simulates the use of ultrasonic sound waves through a physics algorithm to measure the distance from an object in the robot's virtual environment. Educators can connect to the echolocation of bats and dolphins as an activity to explore the properties of ultrasound *before* teaching the blocks that detect the distance from an object to the robot's sensor. It is important to make these connections to students' *schema*, or prior knowledge, to other natural systems that use echolocation.

Kindly refer to **Video 1**, <https://drive.google.com/file/d/1CuYGgckdw5xhYdXFu1jzvtdbHiyeJ8zg/view?usp=sharing> and **Figure 2** as you read the following descriptions of the Vex VR platform. The Vex VR Playground [11], selected in the upper right-hand corner of the blue ribbon on the screen, provides challenges to practice moving the robot around obstacles. This author recommends the Wall Maze (not the Dynamic Wall maze that changes with each run of the program) as a good beginner activity to discover the capabilities of the virtual robot. The programmer has the option to reveal

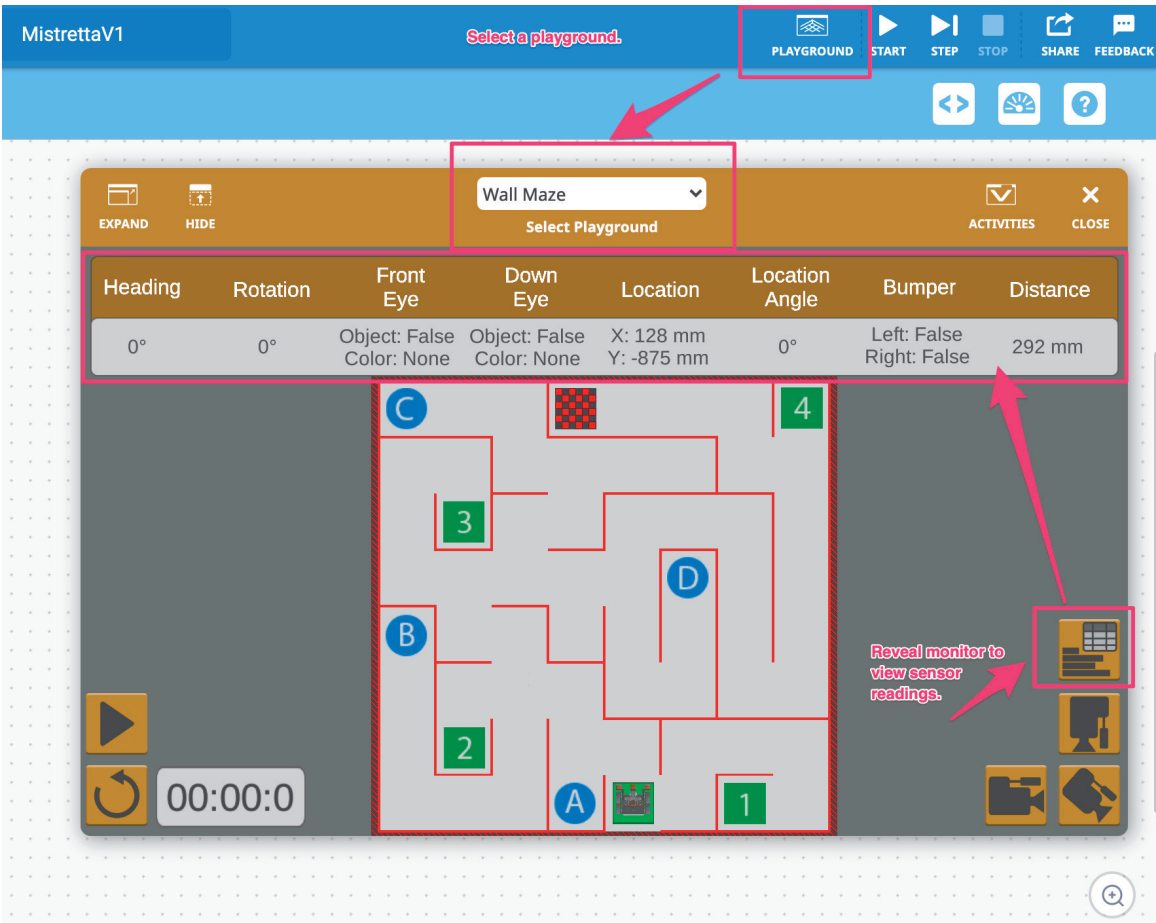


Figure 2. VR vex platform. VR vex is a product of the Robotics Education & Competition (REC) foundation.

a monitor by clicking the button located just above the camera icons in the lower right-hand corner of the Playground pop up window to display the values of the following virtual sensors: front eye, down eye, XY axis location, location angle, bumper value, and distance in millimeters from an object. Based on the readout of the sensors, the programmer can code the robot to stop when the distance threshold is less than a number that they observe on the monitor. Writing code based on the sensor monitor is a tangible application of math to employ comparison operators to calculate when the robot must stop and turn.

2.3 Vex VR technological content knowledge (TCK)

This section elaborates on the intersections of technical and content knowledge (Figure 1) necessary to understand Vex VR. Students who have prior knowledge of programming on the Scratch platform [19], or other websites that use block coding, will recognize the structure of the Vex VR integrated development environment (IDE). Like Scratch programming [19], the code blocks are drag and drop puzzle pieces that join together in the large white work area that dominates the right two-thirds of the screen (Figure 3). Vex VR organizes the blocks into 10 categories: Drivetrain, Magnet, Looks, Events, Control, Sensing, Operators, Variables, My Blocks, and Comments. Vex VR provides you with the “when started” block. The programmer subsequently connects additional blocks based on planning strategies. To learn more about each block, the programmer can click on the question mark in the upper right-hand corner of the screen (Figure 2) and then click on a block in the column that contains the puzzle pieces on the left of the screen. The “Help” column

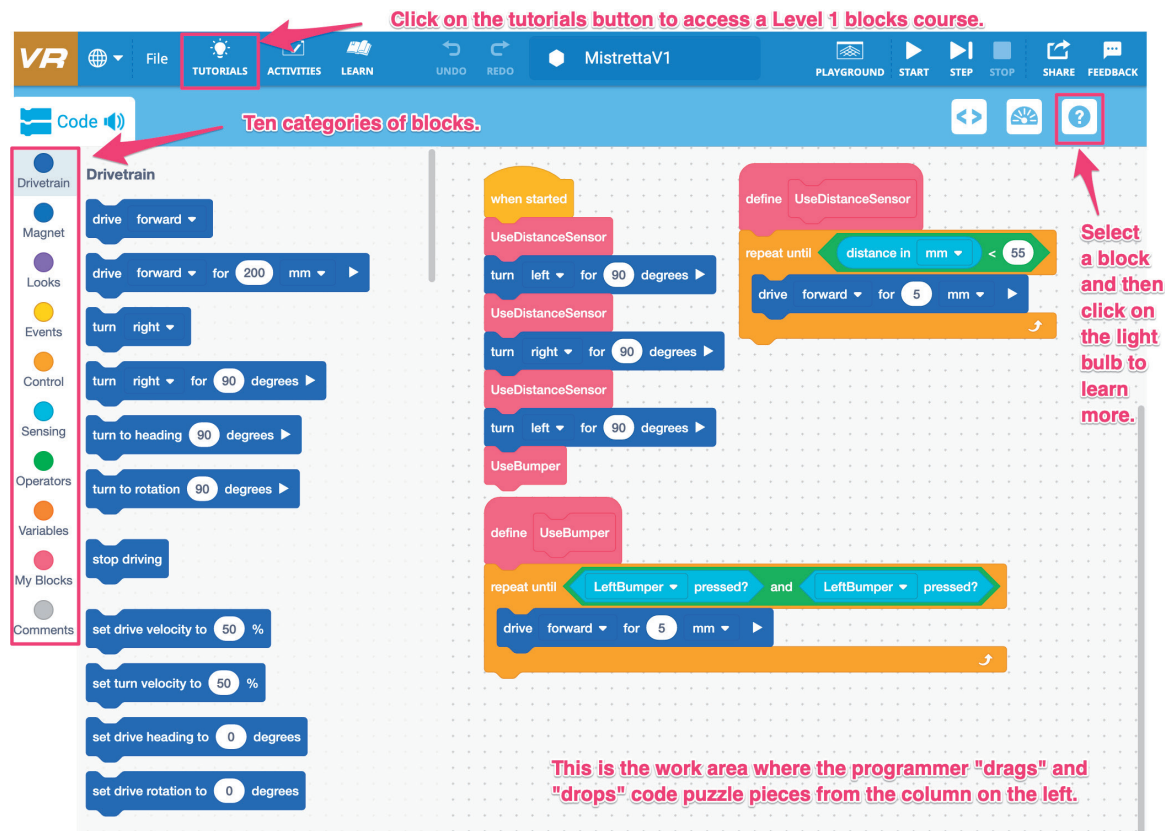


Figure 3. VR vex work area, block categories, and resources. VR vex is a product of the Robotics Education & Competition (REC) foundation.

will populate with information about the selected block. To learn more about topics such as “Driving Forward and Backward” or “Turning,” click on the “Tutorials” button under the lightbulb icon in the top blue “ribbon” of the screen. Vex VR provides a robust Level 1 Blocks Course collection to get the students started.

2.4 Vex VR pedagogical knowledge (PK): computational thinking

This section introduces the pedagogical, teaching methods, knowledge (**Figure 1**) necessary to instruct virtual robotics. Computational thinking is a mindset that is not limited to programming and computer science. It is a set of skills and attitudes that support students’ creative solutions. Educators new to STEM might consider problem-solving as the only component of computational thinking. However, there are two additional skills: abstraction and algorithmic thinking [20]. Educators must emphasize that a computational thinking methodology is an *iterative* approach where mistakes and adjustments are an expected component of the process. **Table 1** summarizes the essential elements of the three computational thinking skills with examples based on the Vex VR Wall Maze challenge [11].

2.5 Vex VR pedagogical content knowledge (PCK)

This section addresses the intersection of pedagogy and content knowledge. Pedagogy is an art, especially when teaching technology. The methods of pedagogy seek multiple entry points to introduce content and provide all students with the opportunity to practice, make mistakes, revise, and reveal understanding. Giving guidance to each student is crucial to advance their knowledge of programming.

Skill	Sub-skill	Example
Problem solving	Decomposition	Student evaluates the Wall Maze to break down the entire maze (large problem) into smaller problems to navigate obstacles from start to finish.
	Redefine Problems	Student examines available code blocks and robot sensors.
	Strategic Decision Making	Student develops several possible solutions and decides which blocks and sensors to employ.
Abstraction	Modeling	Beginner programmer: Student programs a solution using multiple instances of the same blocks to navigate the maze. This can create a very long, concatenated grouping of code.
	Pattern Recognition	Intermediate programmer: Student recognizes that the same blocks are used repetitively.
	Modularity	Advanced programmer: Student identifies generalizable modules using the “My Blocks” feature that consolidates the movements of the robot into recognizing walls with the distance sensor, the bumper switch, left, and right turns.
Algorithmic Thinking	Algorithmic Design	Student develops a step-by-step strategy to call the “My Blocks” modules to create concise code (Figure 3).
	Incremental Design and Evaluation	Student designs, test, and revises code in an iterative approach to solve the maze.

Table 1.
Summary of three computational thinking skills.

Feed Up	“Where am I going?” – Does the student’s developing project exhibit an understanding of the goal of the assignment? Check the student’s understanding before they potentially progress too far in the wrong direction.
Feedback	“How am I going? – As students and educators, we are familiar with the prevailing direction of feedback that answers the questions, “What progress has the student made toward the goal?” With feedback, educators give students input to the overarching question, “How am I going?” that informs students about the trajectory of their work toward the goal. Students frequently ask, “Am I on the right track?”
Feed Forward	“Where to next?” – Offer substantive suggestions to inform students about specific activities that they need to undertake to make progress toward the goal.

Table 2.
Hattie and Temperly’s tri-direction model [21].

Feedback, however, is only one of three components that comprise formative assessments. Hattie and Temperly [21] provide a tri-directional model of helpful feedback: feed up, feedback, and feed forward. **Table 2** elaborates on the three directions of formative assessments.

Hattie and Temperly’s Tri-Direction Model [21].

2.6 Vex VR technological pedagogical knowledge (TPK)

This section discusses the intersections of pedagogy and technology to offer supporting applications to provide feed up, feedback, and feed forward guidance. There are several applications that assist teachers to provide impactful information

Application	Modality	Examples
Mote [22]	Audio Notes	Install the Chrome Mote extension on your computer. Share a worksheet in Google Docs with each student. Use the Mote audio feature in a new comment.
Loom [23]	Screen recording with voiceover and optional on-camera speaker.	Install the Chrome Loom extension on your computer. Download code from your student’s Google folder and upload the “vrblocks” file to the VR Vex in your browser. Review the program to formulate suggested revisions. Click on the Loom extension in your Chrome toolbar and screen record your feedback to your student. Click the Loom checkmark at the bottom of the screen to generate a URL to your feedback. Send the URL to the student via email.
	Small-skill videos	It is helpful to produce small skill videos (Video 1 , https://drive.google.com/file/d/1CuYGckdw5xhYdXFu1jzvtdbHiyeJ8zg/view?usp=sharing), using Loom, to share with students to cover programming techniques. Students can revisit a collection of educator-produced videos to review a skill or work ahead. This is helpful to bring new students up to speed or advanced students to progress ahead. Educators can archive the videos on Google Classroom [15], Google Site [24], or Wakelet [25].
Skitch [26]	Image Annotation	Skitch is part of the Evernote application. Download Skitch to your computer. Take a screenshot of your student’s work with the Skitch application. Use the annotation and text tools to point out the areas of the program or assignment to revise. Figures 2 and 3 are examples of Skitch annotations.

Table 3.
Feed up, feedback, and feed forward tools for educators.

to students. The categories are audio, screen recording with voiceover, and image annotation (**Table 3**). These tools work in face-to-face and online synchronous or asynchronous classwork.

2.7 Vex VR technological pedagogical content knowledge (TPACK)

The successful deployment of the Vex VR platform in a hybrid learning environment with face-to-face, synchronous, and asynchronous entry points compels teachers, coaches, and parents to understand the intersections of technology, pedagogy, and content knowledge. The on-ramp of the browser-based, virtual robotics, Vex VR platform transitions well to the next section that delineates Robomatter's Robot Virtual Worlds platform [12].

3. Robot virtual worlds

The Robot Virtual Worlds (RVW) product, offered by Robomatter, Inc., provides a powerful virtual robot platform [27] that gives programmers an option to run their code on a virtual *or* physical robot. The following information describes the setup of a Windows-based computer lab or laptop cart in *one* building. RVW is licensed software that has an option to obtain a 30-seat perpetual, one-time purchase for Vex or Lego Mindstorm robots [12]. Robomatter wrote RVW to work with the Windows operating system. One must download and install the licenses on *each* computer running the Windows operating system or to each Apple or Chrome computer running a Windows "partition" such as Parallels [28] software on a Mac OS or Chrome OS device. A *partition* is a region of your computer's memory dedicated to simulating the Windows operating system. The cost of a 30-seat, perpetual RVW license is approximately \$600 with an available trial license to test-drive the software [12]. The Robot Virtual Worlds Level Builder, akin to the VR Vex playground, is free. Download a Level Builder package to each computer using Robot Virtual Worlds to "play" challenges or "build" a custom virtual world.

To facilitate students working from home, RVW offers homework pack licenses [12] to install on students' home computers running the Windows operating system. If a student owns a Mac or Chrome computer, the cost to the student for the partition software is approximately \$80. Each student should download the free Level Builder software to facilitate "play" challenges or "build" custom activities. The following sections describe the RVW Vex IQ virtual robot.

3.1 RVW technological knowledge (TK)

This section addresses the technical knowledge in the TPACK framework (**Figure 1**) necessary to implement the fundamentals of the RVW RobotC programming software focusing on a Vex IQ virtual robot. The "C" programming language is an industry standard to program robots [29]. RVW provides a graphical user interface (GUI) with drag and drop blocks similar to VR Vex. Once installed on a computer, RVW provides a desktop icon named ROBOTC for Vex Robotics. Double click on this icon to invoke ROBOTC. Kindly refer to **Video 2**, <https://drive.google.com/file/d/1Aj0sxMaZGRnMeIIa0Ni-jmis9fp7rQjj/view?usp=sharing> and **Table 4** that contain suggested steps for educators to establish RVW on each licensed computer, to create a simple program, and to run this program in the *Turning Challenge* virtual world.

Task	Location in top gray row of the robotC Screen	Notes
1. Establish licenses	Help – Manage Licenses – Add License	One per computer.
2. RVW Package Manager to Install Level Builder	Help – Manage RVW Packages – Checkmark all Packages – Click Install/Update Selected	This is where you install Level Builder. One can also install the Vex IQ Competition Challenges [30] in virtual format here.
3. Review the Help Page	Help – (Choose language)	Review the extensive library of this user manual as needed.
4. Select the Virtual Robot	Robot – Compiler Target – Virtual Worlds	This will download your programs to the virtual robot. When one builds a Vex IQ, select Physical Robot and attach the Vex IQ with the USB cord.
5. Select the Platform Type	Robot – Platform Type – Vex IQ and Robot – Platform Type – Vex Robotics – Vex IQ	This example describes the Vex IQ virtual robot. The Vex Cortex virtual robot is also an option.
6. Select Virtual World to Use	Window – Select Virtual World to Use – RVW Level Builder	This menu option contains a “Download More Packages” option.
7. Select Menu Level	Window – Menu Level – Super User	The Super User options shows all available blocks.
8. Open a sample program	File – Open Sample Program	Select Moving Forward Rotations
9. Compile the Program	Click the Compile Program button at the top of the screen.	One must compile the program each time you make a change.
10. Download to the Virtual Robot	Click the Download to Robot button at the top of the screen	One must download the program each time you make a change. Click Play to reveal the Clawbot IQ page.
11. Select the Level	Select the Turning Challenge as a good beginner activity.	Click “Start Level” to reveal the virtual robot on the playfield. Notice the “play” button to the left of the robot. Click play to enact your program, click the rewind to return to the Start tile, click the “home” icon to return to the Select Level option.

Table 4.
Suggested sequence to establish RVW.

3.2 RVW content knowledge – sensors and geometry (CK)

This section addresses the engineering and geometry content knowledge in the TPACK framework (**Figure 1**) that underpin the robot’s build and associated properties of circle geometry at work. The RVW Vex IQ virtual robot has the same construction and sensors as the physical kit’s “Clawbot” build. A *sensor* is a device attached

Sensor	(P)urpose and (V)alues	Example
Bumper Switch	P: detect an obstacle V: pressed = 1 released = 0	&\$\$\$; [29] This loop of code will move the robot forward one rotation at half power until the bumper switch is pressed indicating that it drove into an object.
Distance Sensor	P: Detects an obstacle with ultrasonic soundwaves V: Measures distance from 50 mm to 1 m	&\$\$\$; [29] This loop of code will move the robot backward one rotation at half power until the distance sensor detects that an object is less than 300 millimeters away. Note: the virtual robot has its distance sensor on the back bumper. Hence, one must turn the robot to face the distance sensor toward the object that you wish to detect
Gyro Sensor	P: Measure the turn rate and calculates the direction of the robot. V: based on 360 degrees of a circle.	&\$\$\$; [29] Always reset the gyro sensor (to 0) before turning the robot
		A gyro turning counterclockwise will increase its values. A gyro turning clockwise will decrease its values.
		This loop of code turns the robot to the left by setting the speed of the robot's left motor to -50 and the right motor to 50 until the gyro value is greater than 90 degrees.
Touch LED	P: In virtual robotics, set the LED to color to denote that a section of code is being enacted V: For example, colorRed, colorGreen.	&\$\$\$; [29] For example, set the touch LED sensor to a different color for each section of code for a visual indication that a section of code is currently running.
Color Sensor	P: Detects the color of obstacles. V: Returns a color name or value of red, green, and blue in 256 levels.	&\$\$\$; [29]
Smart Motors	P: Using an encoder within the motor, it measures speed, direction, time, revolutions, and degrees of turn. V: See example.	&\$\$\$; [29] This block moves the robot forward for three rotations of the wheel at half power.

Table 5.
Robot sensors with examples.

to the robot's brain that detects the environment and sends numbers to the brain to report its findings. The programmer writes the code to respond to the robot's data to navigate the environment. **Table 5** lists the Vex IQ sensors, their purpose, and an example of how to use the sensor in a program.

It is necessary to understand circle geometry to employ the 360° properties of a circle to calculate the turns of the robot using the gyro. Additionally, students can arrive at the circumference of the robot's wheel to determine the distance in millimeters that the device travels in one rotation. Notice the length of the radius of the Clawbot IQ wheels on the start page of the Level Builder virtual robot (**Video 2**, <https://drive.google.com/file/d/1Aj0sxMaZGRnMeIIa0Ni-jmis9fp7rQjj/view?usp=sharing>). The radius of the wheel is 3.2 cm. Therefore, the diameter of this circle is 6.4 cm. Students can use the formula $\Pi * 6.4$ to calculate the circumference of the wheel to arrive at the distance that the robot wheel travels in one rotation. The circumference of the wheel in centimeters is approximately 20 cm or 200 mm. Using

the measuring beam on the virtual robot, the student can calculate how many rotations it will take to travel the Turning Challenge depicted in **Video 2**, <https://drive.google.com/file/d/1Aj0sxMaZGRnMeIIa0Ni-jmis9fp7rQjj/view?usp=sharing>.

3.3 RVW technological content knowledge (TCK)

This section elaborates on the intersections of technical and content knowledge (**Figure 1**) necessary to understand Robot Virtual Worlds. Similar to the drag and drop code blocks described in Section 2.3, RVW is a robust platform that provides RobotC programming in 12 graphical functions categories (**Table 6**).

3.4 RVW pedagogical knowledge (PK) – perseverance rover

This section suggests a pedagogical method (**Figure 1**) to make real world connections to robotics as students program their virtual robot. The Mars Perseverance Rover is a robot launched by NASA in July of 2020 and deployed on Mars on February 18, 2021 [31]. The Perseverance mission team engineered the rover to utilize a

Graphical Function	Use
Program Flow	Contains blocks that the programmer to evaluate the data coming in from the sensors as the robot travels in an environment. The program flow blocks contain three components: the name of the sensor, a comparison symbol such as less than < or greater than >, and a threshold value such as a number, a color, or a Boolean value such as true or false. &\$\$\$; [29] This program flow block evaluates the gyro sensor data and will turn the robot to the right until the value is less than -89.
Variables	The variable blocks allow the programmer to create a named location in the memory of the brain to store a value to use in a programming block. &\$\$\$; [29] The programmer established the variable MyPower, set the value to 100, and then used the variable in the forward block. Using a variable in this manner standardizes the power. The programmer changes the value in one place and recompiles to change the speed of the robot throughout the program.
Simple Behaviors	Contains backward, forward, moveMotor, turnLeft, and turnRight commands. Use the Help – Command Library Vex IQ – Graphical – Simple Behaviors user manual to learn more.
Motor Commands	Contains blocks that address the encoder properties of the motor sensors. Use the Help – Command Library Vex IQ – Graphical – Motor Commands user manual to learn more.
Remote Control	Contains blocks to program the handheld controller that communicates via radio in physical robots.
Timing	Contains blocks to time or delay the program.
Line Tracking	Contains blocks to follow the edge of a line with the color sensor.
Datalog	Creates a graph of data from a selected sensor.
Display	Used to reveal controller, motor, or sensor values on the screen of the brain.
TouchLed	Changes the color on the TouchLED sensor
Distance and Gyro Sensor	Contains blocks to reset the distance and gyro sensors.

Table 6.
Robot virtual worlds 12 graphical function categories.

sophisticated collection of cameras and sensors to navigate the environment of Mars [32]. However, one simple calculation correlates directly to the students’ virtual robot and circle geometry. The mission team calculated the distance that Perseverance travels in one rotation of the wheels, with no slippage on the rocky terrain of Mars, as 1.65 meters [33] using the same formula for the circumference of a circle demonstrated in Section 3.2 of this chapter. Educational philosopher John Dewey asserted that, “We do not learn from experience. We learn from reflecting on experience” [34]. Take the time to have the students pause and reflect on the rover on Mars to appreciate the skills that they are learning as actionable in future STEM careers.

3.5 RVW pedagogical content knowledge (PCK)

To achieve the intersections of pedagogy and content knowledge, teachers, coaches, and parents can reflect on the three types of problems delineated by Kirkley in the Principles for Teaching Problem Solving [35]. The students enacting virtual robotics will solve ill-structured problems without one solution. For every student in a class or on a team, they can develop a unique solution that solves the successful navigation of the virtual robot around the selected challenge. **Table 7** reveals three types of problems and the implications for instructing virtual robotics.

Type	Definition	Example	Pedagogy
Well-structured problem	The same step by step solution with one right answer.	Calculating the circumference of the wheel of a robot using $\Pi \times \text{Diameter}$	Learner memorizes formula.
Moderately structured problem	More than one acceptable solution with one right answer.	Turn the robot 90° to the right to navigate a maze. The learner can use the gyro sensor and setMotor blocks in a repeatUntil loop or a simple behavior of turnRight. Both solutions will turn the robot 90° to the right.	Encourage reflection to real world contexts such as the Perseverance Rover.
			Learner selects a strategy to turn the robot to the right 90°. The solution using the gyro requires more analysis and abstract reasoning.
Ill-structured problem	This is an open-ended problem with many correct solutions.	Challenge the students to move around a maze using all available sensors in their solution.	Encourage students using the simple behavior to develop a solution to employ the gyro and motors.
			Students must plan the direction of the rover to move forward to use the bumper switch, backward to use the distance sensor, motor encoders to calculate distance traveled to navigate the robot to the finish block.

Table 7.
Three types of problems [35].

3.6 RVW technological pedagogical knowledge (TPK)

This section discusses the intersections of pedagogy and technology to offer supporting applications for students to develop their strategies to solve ill-structured problems. Students who complete the challenges provided within the RVW Level Builder software, such as the Turning Challenge, will be ready to create their own virtual environment. As previously demonstrated (**Video 2**, <https://drive.google.com/file/d/1Aj0sxMaZGRnMeIIa0Ni-jmis9fp7rQjj/view?usp=sharing>) there are two options to enact a program in RVW Level Builder. The first is “Play” and the second is “Build.” When the programmer selects “Build”, the system presents a blank playfield with one “Start” Block (**Figure 4**).

3.6.1 Level builder custom virtual environment

Students can customize their virtual Level Builder environment by dragging and dropping objects provided by RVW onto the playfield and saving their unique environment for continued development (**Figure 4**). Most exciting is that the student can create 3D objects using free online platforms such as Tinkercad [13] offered by Autodesk [Autodesk], export the resulting “.stl” file to the download folder of their computer, and import this file to Level Builder (**Video 3**, <https://drive.google.com/file/d/1ddxnW7pp3KdELAbh53vAfdSC5spGUEp0/view?usp=sharing>). Their custom 3D object becomes an obstacle to navigate around or an object to be “picked up” and carried by the arm and claw of the Clawbot Vex IQ virtual robot.

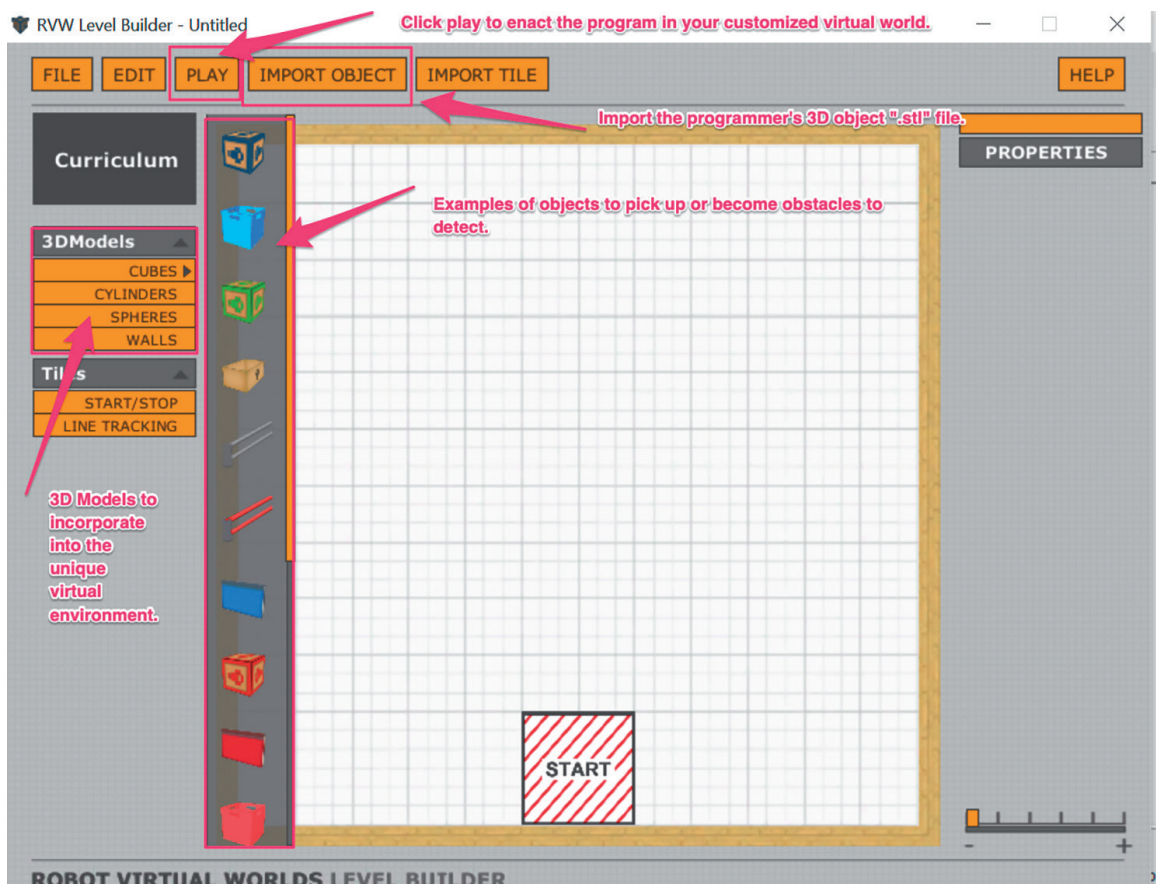


Figure 4.
Robot virtual worlds level builder – build option.

3.6.2 View custom 3D objects in augmented reality

Thus far, this chapter discussed the Vex VR and Robot Virtual Worlds virtual reality applications where the user observes a robot navigating a coded environment on a computer screen. Augmented Reality (AR) is a technology that overlays digital information into a user's real surroundings. Learners use a mobile phone or tablet running an application such as Merge Object Viewer [36] to facilitate the projection of objects onto a Merge Cube [14] into their current environment. This type of AR application is useful to students who designed a 3D object using Tinkercad and wish to view the object in their current environment before uploading it to Level Builder (**Video 4**, https://drive.google.com/file/d/1hTu_54BD9MkXa8bLj4brcMiG-8vw1p4K/view?usp=sharing). Additionally, the student can observe the object for size and expectations before upload to their custom Level Builder environment.

3.6.3 3D printing services

If the teacher, coach, or parent wants to print the student's custom 3D object, companies such as MakeXYZ [37] offer services to upload the.stl file, select inexpensive materials, and ship the resulting 3D object to you. The object can be used later with a Vex IQ robot on a physical playfield. Printing services provide several benefits to teachers, coaches, and parents who do not have access to a 3D printer and raw materials. Printing in 3D requires a large block of time, often overnight for one object. The extruder of the printer that melts the raw material filament can be approximately 280° C or 536° F [38], which requires adult supervision. If the roll of filament becomes jammed during printing, it can pull the extruder off the printer and ruin the printed object. Printing services provide access to 3D objects for students in face-to-face, synchronous, or asynchronous learning environments.

3.7 RVW technological pedagogical content knowledge (TPACK)

The RVW package provides *every* learner with an opportunity to code in RobotC, practice navigation in pre-packaged challenges, and create custom virtual worlds. The integration of student-created 3D objects into RVW with the option to observe and jury their creation using AR applications such as Merge prepares each student to become a producer of custom programs that transitions well to physical robotics.

4. Transition from virtual to physical robotics

Teachers, coaches, and parents who shepherd learners through the world of virtual robotics will be well-positioned to take the leap into physical robotics. The RVW package has a compiler target for a virtual or a physical robot. The students can compile and download the same program that they developed to navigate their custom Level Builder world to a physical Vex robot via a computer-to-robot USB cord. Obstacles crafted from recycled materials and placed on the floor of a classroom, community center, or home provides a workable test environment. It is helpful to create a "game" to have teams of students develop a scenario to have the robot gain "points" as the it navigates a custom, physical playfield. Example of games that this author's students developed included a home base for astronauts on the Moon and a distribution center for clean water in the aftermath of a natural disaster. Each time the

robot accomplished a task using its sensors, the team gained points. It is important to note that teams can share the same physical robots by downloading their program to the robot brain. Sharing the same robots requires teams' agreement on the "build" of the robot to agree upon the location of the sensors.

5. Conclusion


Virtual robotics provides teachers, coaches, parents, and students with a unique opportunity to achieve a one-to-one ratio of robot to student. Vex VR and Robot Virtual Worlds provide students in face-to-face, asynchronous and synchronous settings with the opportunity to build upon their programming expertise while navigating a virtual robot in packaged and customized challenges.

Author details

Sharon Mistretta
Johns Hopkins University School of Education, Baltimore, Maryland, USA

*Address all correspondence to: cottagetechnology@gmail.com

IntechOpen

© 2022 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Burack C, Melchior A, Hoover M. Do after-school robotics programs expand the pipeline into STEM majors in college? (RTP). Proceedings of the ASEE Annual Conference & Exposition. 2018;**9**:1-25
- [2] Costa C. Robotics K-12 and your district: The essence of STEM education and the E-ticket to unlimited possibilities. Leadership. 2017;**46**(4):32-35
- [3] Nugent G, Barker B, Grandgenett N, Welch G. Robotics camps, clubs, and competitions: Results from a US robotics project. Robotics & Autonomous Systems. 2016;**75**:686-691
- [4] We Robot. St. Vrain Valley Schools [Internet]. 2021 [cited 2021 Nov 7]. Available from: <https://www.svvsd.org/2021/10/11/we-robot/>
- [5] Sysgen. SysGen [Internet]. 2014 [cited 2021 Nov 7]. Available from: <https://sysgen.ca/sponsorship-robotics-club/>
- [6] REC Foundation. REC Foundation [Internet]. 2019 [cited 2021 Nov 7]. Available from: <https://www.roboticseducation.org/team-resources/grants/>
- [7] VEX IQ Classroom Bundles (2nd generation) [Internet]. Vexrobotics.com; 2021 [cited 2021 Nov 7]. Available from: <https://www.vexrobotics.com/iq-bundles.html?q=&locale.name=English>
- [8] Google Cardboard. Google VR [Internet]. Google.com; 2021 [cited 2021 Dec 1]. Available from: <https://arvr.google.com/cardboard/>
- [9] Merge Headset. AR/VR Learning & Creation [Internet]. Merge. 2021 [cited 2021 Dec 1]. Available from: <https://mergeedu.com/headset>
- [10] Koehler MJ, Mishra P, Cain W. What is technological pedagogical content knowledge (TPACK)? Journal of Education. 2013;**3**(3):13-19
- [11] VEXcode VR. VEX Robotics [Internet]. Greenville, TX USA: Vexrobotics.com; 2021 [cited 2021 Nov 7]. Available from: <https://www.vexrobotics.com/vexcode-vr?q=&locale.name=English>
- [12] Robot Virtual Worlds [Internet]. Robotvirtualworlds.com. 2021 [cited 2021 Nov 16]. Available from: <https://www.robotvirtualworlds.com/>
- [13] Tinkercad. From mind to design in minutes [Internet]. San Rafael, CA, USA: Tinkercad. 2021 [cited 2021 Nov 7]. Available from: <https://www.tinkercad.com/>
- [14] MergeEdu. AR/VR Learning & Creation [Internet]. San Antonio, TX, USA: Merge. 2021 [cited 2021 Nov 7]. Available from: <https://mergeedu.com/>
- [15] Classroom. Google for Education. Classroom | Google for Education [Internet]. Mountain View, California USA: Google for Education. 2021 [cited 2021 Nov 29]. Available from: https://edu.google.com/intl/ALL_us/products/classroom/
- [16] Google Drive [Internet]. Google.com. 2011 [cited 2021 Nov 29]. Available from: <https://drive.google.com>
- [17] Google Meet [Internet]. Google.com. Google Meet; 2020 [cited 2021 Dec 2]. Available from: <https://meet.google.com/>

- [18] Google Chrome. Download the Fast, Secure Browser from Google [Internet]. Google.com. Mountain View, California USA; 2017. [cited 2021 Dec 3]. Available from: <https://www.google.com/chrome/>
- [19] Scratch. Educators [Internet]. Cambridge, MA, USA: Mit.edu. 2021 [cited 2021 Nov 8]. Available from: <https://scratch.mit.edu/educators>
- [20] Computational Thinking Assessment Guide [Internet]. Available from: <https://chrome.google.com/webstore/detail/mote-voice-notes-feedback/ajphlblkfppdpkgokiejbjfohfhhmk?hl=en-US>
- [21] Hattie J, Timperley H. The Power of Feedback. *Review of Educational Research*. 2007;77(1):81-112
- [22] Mote: Voice notes & feedback [Internet]. Mountain View, California, USA: chrome.google.com. Available from: <https://chrome.google.com/webstore/detail/mote-voice-notes-feedback/ajphlblkfppdpkgokiejbjfohfhhmk>
- [23] Loom. Send a video. Not a thousand words. [Internet]. San Francisco, CA USA: Loom. Loom; 2020 [cited 2021 Nov 8]. Available from: <https://www.loom.com>
- [24] Google Sites [Internet]. Mountain View California, USA: Google.com. 2021 [cited 2021 Nov 29]. Available from: <https://sites.google.com>
- [25] Wakelet. Save, organize and share content [Internet]. Manchester, United Kingdom: Wakelet.com. Wakelet; 2021 [cited 2021 Nov 29]. Available from: <https://wakelet.com/>
- [26] Evernote [Internet]. Redwood City, California, USA: Evernote. 2018 [cited 2021 Nov 8]. Available from: <https://evernote.com/products/skitch>
- [27] Robomatter [Internet]. Pittsburgh, PA USA: Robomatter.com. 2020 [cited 2021 Nov 16]. Available from: <https://www.robomatter.com/>
- [28] Parallels: Mac & Windows Virtualization, Remote Application Server, Mac Management Solutions [Internet]. Seattle, Washington, USA: Parallels.com. 2021 [cited 2021 Nov 16]. Available from: <https://www.parallels.com/>
- [29] ROBOTC for VEX Robotics 4.x (VEX Cortex & VEX IQ) [Internet]. Greenville, TX, USA: Vexrobotics.com. 2014 [cited 2021 Nov 17]. Available from: <https://www.vexrobotics.com/robotc-vexedr-vexiq.html?q=&locale.name=English>
- [30] REC Foundation Competition History [Internet]. Greenville, TX USA: REC Foundation. 2019 [cited 2021 Nov 17]. Available from: <https://www.roboticseducation.org/team-resources/competition-history/>
- [31] Watch in Person [Internet]. Pasadena California, United States: Nasa.gov. 2020 [cited 2021 Nov 30]. Available from: <https://mars.nasa.gov/mars2020/timeline/launch/watch-in-person/>
- [32] Mission Overview [Internet]. Pasadena California, United States: Nasa.gov. 2020 [cited 2021 Nov 30]. Available from: <https://mars.nasa.gov/mars2020/mission/overview/>
- [33] Rover Wheels [Internet]. Pasadena California, United States: Nasa.gov. 2020. [cited 2021 Nov 30]. Available from: <https://mars.nasa.gov/mars2020/spacecraft/rover/wheels/>
- [34] The Collected Works of John Dewey, 1882-1953. Electronic Edition. The Later Works of John Dewey, 1925-1953. vol. 8: 1933, Essays, How We Think, revised

edition. How We Think [Internet].
Charlottesville, Virginia, USA: Jhu.edu.
2021. [cited 2021 Nov 30]. Available
from: <http://pm.nlx.com.proxy1.library.jhu.edu/xtf/view?docId=dewey/dewey.28.xml;chunk.id=div.lw.8.13;toc.depth=1;toc.id=div.lw.8.13;brand=default>

[35] Kirkley J. Principles for Teaching Problem Solving. Bloomington, MN USA: University of Indiana; 2003. Technical Paper #4:1-16

[36] Merge Object Viewer [Internet]. San Antonio, TX USA: Merge Help Center. 2015. [cited 2021]. Available from: <https://support.mergeedu.com/hc/en-us/articles/360052929752-Downloading-Merge-EDU-Apps>

[37] Makexyz [Internet]. Tempe, AZ USA: Makexyz.com. 2018. [cited 2021 Dec 1]. Available from: <https://www.makexyz.com>

[38] Simplify 3D. [Internet]. Simplify3d.com. Cincinnati, OH USA; 2019. [cited 2021 Dec 2]. Available from: <https://www.simplify3d.com/support/print-quality-troubleshooting/overheating/>