JTILT

Journal of Technology-Integrated Lessons and Teaching

(2022) Volume 1, Issue 1

"We're Coding A Drone?!" Hands-On Experiences for Online Graduate Students

Irene A. Bal, Loyola University Maryland

Overview

Drones are purchased for PK-12 schools with little thought concerning their use to support learning (Carnahan et al., 2016). The following activity introduces drones and coding to online graduate students, who are PK-12 teachers in an educational technology course, through a social constructivist design. Employed in an online setting, students can consider how drones could be used in their classrooms and be exposed to coding and drones without individual access to the hardware. Students are introduced to block coding and code a drone flight, which is then flown and videotaped by the instructor. Students then view and reflect on their drone flights before discussing integration ideas that directly support their PK-12 classroom and content.

Topics: Coding, Computational thinking, Computer science, Drones

Time: 2.5 hours – 30-minute set up, 60-minute student engagement, 60-minute flights and recording

Materials

For this activity, instructor materials are:

- Tynker free teacher account
- Two mobile devices with Wi-Fi and Bluetooth capability (tablet or phone)
- Tynker App (Apple or Google Play)
- Camera App with video capability
- Parrot Mambo Drones (Fly or Mission – Mission comes with accessories and a controller)
- Large, open, indoor space (with little furniture)
- Flipgrid account
- YouTube Playlist of Drone Article Videos
- Optional: video editing software

The student materials are:

- Computer or mobile device with internet access

Context at a Glance

Setting
Adult students in an educational technology graduate program in an urban, private institute of higher education in the northeast region of the United States.

Modality
Online, Asynchronous

Class Structure
This is one activity in a required eight-week graduate course taught in an online environment (both synchronous and asynchronous). The instructor needs an open space to fly the drones.

Organizational Norms
The students are enrolled in an online program with no expectations of coming to campus.

Learner Characteristics
Students who engage in this activity have a range of prior experience in computer science, programming, and coding with most students having little to no experience.

Instructor Characteristics
One university faculty member developed this activity. They were familiar with computer science, computational thinking, and coding for PK-12 spaces but had less technical knowledge of and experience with drones.

Development Rationale
This activity was created to expose graduate students to coding, computer science, and drones, and make them think about how drones can be used in PK-12 classrooms.

Design Framework
Social Constructivist Design Theory

DOI: 10.13001/jtilt.v1i1.6983

Setup

Prior to setting up this activity, the instructor should acquire access to drones (this can take several weeks depending on the institution) and reserve a large, open, indoor space with no furniture, if possible, for the drone flights. Depending on the institution, this room reservation may need to be weeks or months in advance. Once the drones are acquired and the room is reserved, the instructor needs to set up the activity through the Tynker app before student engagement with the drones (see Instructor Setup Before Activity section). Typically taking around 30 to 60 minutes, this setup includes the digital resources, prompts, and video platform (i.e., Flipgrid) for the activity. Prior to the drone flight date, the instructor should ensure the drones are charged, which can take several hours.

Standards for Educators

The International Society for Technology in Education (ISTE) standards for educators and computational thinking were utilized for this activity. This lesson aligns with the ISTE Standards for Educators 2.4.b and 2.6.c (ISTE, 2017), and with the ISTE Computational Thinking Competencies 5.1.d and 5.3.c (ISTE, 2019).

Context and Setting

This activity is completed in a second-year, eight-week, graduate level educational technology course at an urban, private institution of higher education. The course focuses on digital innovations that support deeper learning in PK-12 classrooms. This activity falls within week three of the eight-week course, which focuses on computational thinking, computer science, and higher-level learning.

Students in this course are PK-12 teachers. Most students are state-level certified teachers who completed a bachelor’s degree in education, work full-time at a PK-12 school, and are enrolled in a graduate program. Regardless of prior education, experience, and selected program, all students have an interest in educational technology in their PK-12 classrooms.

The course was originally piloted in 2018 in a 15-week, face-to-face course. The 2018 face-to-face pilot class provided feedback for future online iterations. Since the 2018 pilot class, this activity has been implemented online with 13 other classes of students consisting of 12 to 24 students who ranged in ages ranging from 21 to 60+ years old. The activity is completed in groups of 3-4 students with over 200 students (56 groups) completing this activity since 2018. The activity is now facilitated asynchronously, and groups are encouraged to meet synchronously to complete the coding lessons and code their drone; however, synchronous collaboration is optional.

This online course includes various collaborative activities, and I allow students to complete this activity in groups. Students can select their own groups, which are the same groups they work with throughout the course. Within computer science, especially in coding, creating collaborative activities for team coding or paired programming supports students in solving problems that arise in the activity as it promotes sharing information and feedback (Sadik et al., 2020).

Although students purchase books and technology for learning within the program, hardware purchases by students are limited. Efforts are made to showcase innovative, digital technology that are affordable and found in PK-12 schools. In PK-12 schools, technology and equipment should be purchased based on the curricular needs (Warschauer & Tate, 2015), but drones are often purchased with little understanding of classroom integration and how they can support learning standards (Carnahan et al., 2016). The integration of drones in this graduate course activity was designed to support students if drones are purchased for their classroom/school, and they are asked to integrate them into their curriculum.

Since this activity was designed for an online course, the selection of technology was crucial to the functionality of the activity and positive experiences of the students. I explored which drones were best for PK-12 students and could be navigated with educational technology apps that were cost-effective. Two Parrot Mambo Mission drones were purchased for this course. The Parrot Mambo Mission drones came with a cannon and miniature balls, a grabber, and a fly pad controller. The cannon and grabber are options for students to include in their codes and drone flights. This activity can be done with different drones and no accessories, though technical applications and directions may need to be updated based on available equipment.

DOI: 10.13001/jtilt.v1i1.6983

The Parrot Mambo drones connect via Bluetooth to the application Tynker, a PK-12 coding platform that supports app-based and web browser coding (see Figure 1). The app-based and web browser functionality were key to the success of this activity as the students in the course have a range of technical ability and access to various devices. The activity needed to support students who primarily use a mobile device for learning as well as students with computers. Tynker also has a free teacher account that includes unlimited student accounts. Due to limited classroom funding, many teachers purchase their own materials and technology (Walker, 2018). Thus, it was important to find an application that waived or required low-cost subscription fees.

Figure 1. Tynker App with Parrot Mambo drone.

Note. From "Media Kit," by Tynker, *About Us*, n.d. (retrieved https://www.tynker.com/about/). 2018 by Tynker. Reprinted with permission.

Since this activity has been refined over various years, the activity presented here is the most up-to-date version, and the critical reflection showcases some of the updates that occurred over the years. This activity is graded but only as a completion grade as the purpose is not to code the perfect drone flight, but instead, to be exposed to computer science and the possibility of using drones in PK-12 classrooms.

Learning Representation

Instructor Setup Before Activity

Before assigning the activity, a few setup items are needed, such as setting up Tynker and creating directions. I do these items prior to the start of the semester, so I can provide the needed information to my students before the week of instruction.

Setting Up a Tynker Class

The first step of setup is creating a teacher account and setting up a class in Tynker. If you use Google Classroom, you can link your Google Classroom to your Tynker account. I do not use Google Classroom as an LMS, so I utilized the 'Create New Classroom' option.

When selecting the 'Create New Classroom' option, complete the required information, such as the classroom name and grade levels. I do not provide students' email addresses when creating students, though you can. Instead, I create usernames and passwords for my students. To make this easy for myself, I created usernames that are the word 'student' and a number (e.g., Student10) and passwords that are the course name (e.g., ETT600). I supply each group with one username and password to complete the activity.

At the time this activity was written, Tynker did not have a way for multiple students to work on one project together but does allow projects to be shared with other students in a view only mode (Tynker, 2020). Since I have the students work in groups for this project and they are not in the same physical space, each group shares one username and password that I supply. To limit the management of who is in each group and which username/password they should receive, I provide my students the usernames and passwords for all the groups that semester so they can select the correct group. Since I work with graduate students, I have not had any issues with students using other group's username/passwords for this activity. You can always use your students' emails or a class code if you think this may be an issue with your students.

Assigning Lessons in Tynker

Once the Tynker course is created and students are added, you can assign Tynker courses and lessons to your students. I only assign the free Tynker courses and lessons to my students, such as the Programming 101 course. Each Tynker course includes multiple lessons (see Figure 2).

DOI: 10.13001/jtilt.v1i1.6983

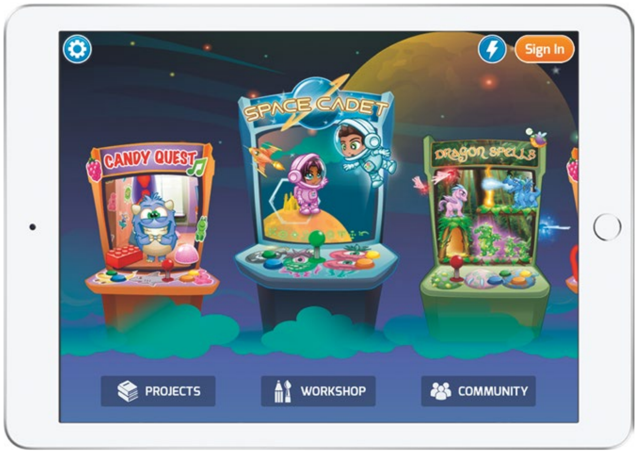


Figure 2. Lessons on Tynker app.

Note. From “Media Kit,” by Tynker, *About Us*, n.d. (retrieved https://www.tynker.com/about/). 2017 by Tynker. Reprinted with permission.

When assigning Tynker courses, you can add all the lessons within the course or select specific lessons. I add all the lessons and indicate in the directions that students can choose the lessons they want. Once a lesson is assigned and students access the lesson, students’ progress can be viewed in the Tynker teacher dashboard. For this activity, my students are expected to complete two lessons, one in the Programming 101 course and one in either the Programming 101 course or Hour of Code course (see Creating Directions for Students section). These two lessons provide enough background to use the Tynker program and to successfully code a drone. Some student groups complete more than this, but most just complete these two lessons.

After setting up Tynker, instructors need to create directions for students. Since I conduct this activity asynchronously, directions are provided in the form of text and video.

Creating Directions for Students

The directions for this activity sit within a HyperDoc that includes all the content and activities for the whole week. HyperDocs are learner-focused lessons, typically presented in Google Docs or Slides, using active, inquiry-based learning (Highfill et al., 2016). HyperDocs can be organized in various manners, and this HyperDoc is created in a modified 5E Lesson Plan (Engage, Explore, Explain, Experience, Share, Reflect, and Extend) with the drone activity located in the Experience section. While the entire HyperDoc is beyond the scope of this article, the needed information for students to code drones is packaged within the Experience section (see Figure 3).

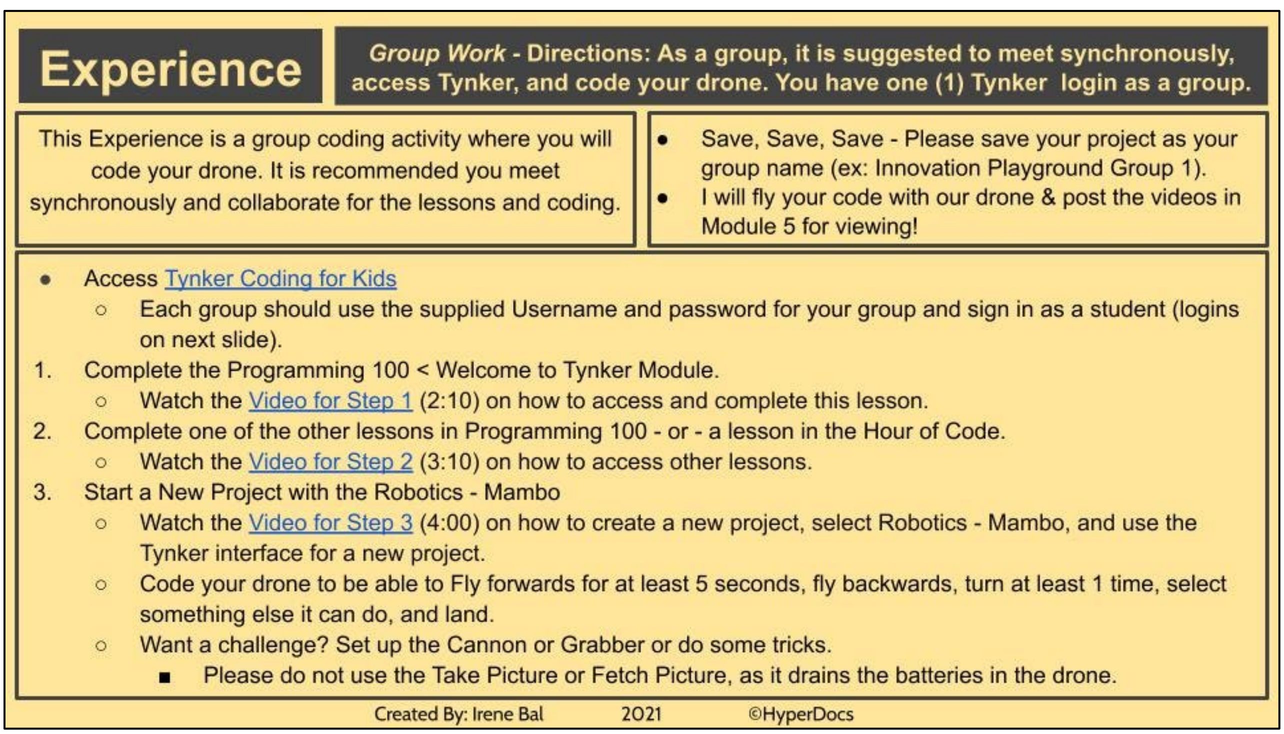


Figure 3. Tynker directions in experience slide.

DOI: 10.13001/jtilt.v1i1.6983

The directions indicate how many lessons students should complete before coding their drone and include some screencast videos on how to access and complete these lessons. The Experience slide (Figure 3) is highly text-based but is one of the only mostly text-based slides in the HyperDoc. First, students are provided a recommendation to work on this drone activity synchronously. Notice, this is not required, as some groups are not able to find the time to meet synchronously every week. If the group decides to work asynchronously, I am unable to determine who in the group completed the required Tynker lessons due to the shared usernames and passwords. I do not concern myself too much over whether everyone in the group has completed the lessons because, although it is expected the lessons are completed, it is their responsibility to complete them, and they are limiting their own success if they do not complete them. Sharing usernames and passwords may need to be revised for learners at other levels such as PK-12 and preservice students if a verification of completed Tynker lessons is needed.

The next directions are to write code and save frequently. The lack of saving projects causes issues for some groups. If they do not save, I cannot access their most recent code. I also detail what will happen after they code the drones, indicating I will fly the drones and record videos of the flights and then add them to Module 5 (week five) for student viewing. After these basic directions, I provide the steps to access Tynker and complete the activity.

The first step is for students to access Tynker and login using the supplied usernames and passwords (see Figure 4; *Note,* usernames and passwords were removed from this image. This example showcases two sections of the course in the same semester (i.e., Monday Night Class and Tuesday Night Class)). A link to Tynker is provided for students to access the platform on their devices.

Since the students are in groups, each group receives one username and password to access Tynker. I list each Group name (1, 2, 3, 4) and the username and password they should use. Again, I found supplying all usernames and passwords within the HyperDoc more efficient as it allows everyone in the group to access the platform and the group flight.

After students access Tynker, they are provided three next steps (see Figure 3). Step one is to complete Programming 100 – Welcome to Tynker Module. A screencast of how to access and complete the lesson is included (see Video 1; select the image to access the video or see the drone video playlist).

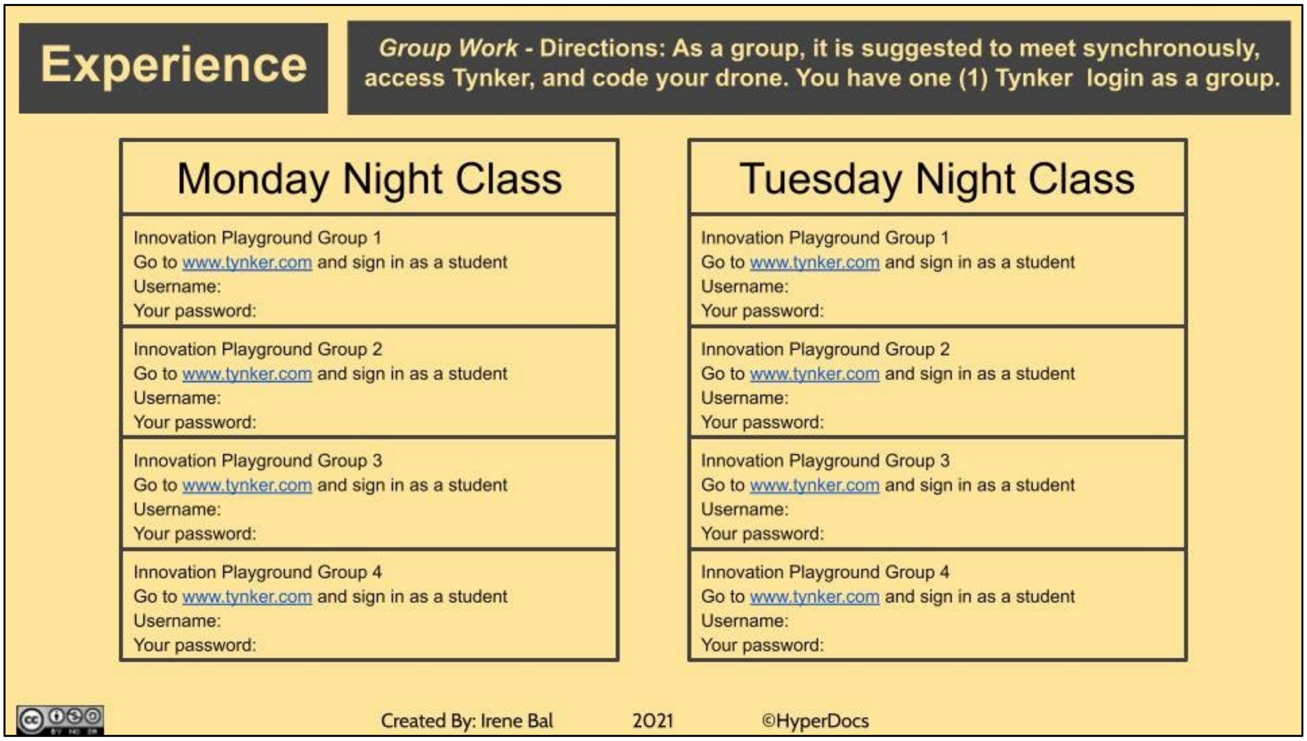
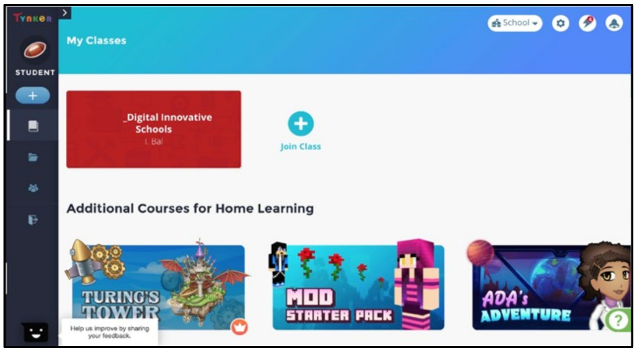


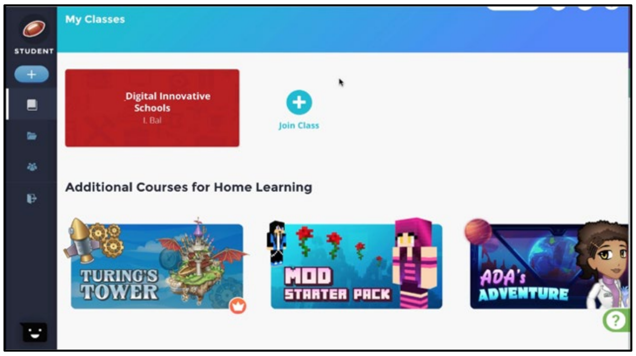
Figure 4. Tynker usernames and passwords in second experience slide.

DOI: 10.13001/jtilt.v1i1.6983



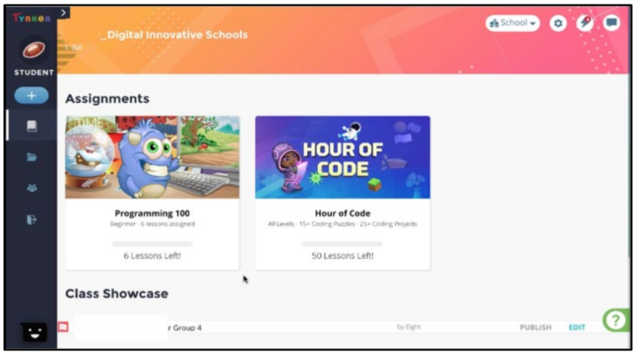
Video 1. Step 1 – accessing Tynker & completing 1st lesson.

After students complete the Welcome to Tynker Module, they are prompted to complete at least one of the other lessons in the Programming 100 or Hour of Code Courses. Instructions provide a screencast of how to access other lessons (see Video 2; select the image or see the [drone video playlist]).



Video 2. Step 2 – accessing other Tynker lessons.

Lastly, students are tasked with creating a new project to code the drone. Instructions include a video on creating new drone projects (see Video 3; select the image or see the [drone video playlist]), the minimum actions for drone flights, and a challenge to code advanced actions into their flights.



Video 3. Step 3 – create a new drone project.

At a minimum, students must code their drones to fly forwards for at least 5 seconds, fly backwards, turn at least 1 time, to select something else it can do (student choice), and land (see Figure 3). The students are also provided the prompt, "Want a challenge? Set up the Cannon or Grabber or do some tricks." These are optional actions that create a more exciting flight. Finally, students are instructed, "Please do not use the Take Picture or Fetch Picture, as it drains the batteries in the drone." Before I fly the drones, I check to see if "Take Picture" or "Fetch Picture" are included, and I remove them from the code if they are to ensure I have enough battery life to fly all the drone flights (see Critical Reflection section for more details).

Once the directions and resources are created, students can access Tynker and begin their learning experience of coding drones. It is recommended to review these directions and resources at the start of each semester to ensure they support any updates to Tynker and coding the drone.

Student Engagement Before Drones

The social constructivist design theory includes three aspects which support learning, "personal interpretation of experience," an "active process," and "multiple perspectives" (Richey et al., 2011, p. 130). As students complete the coding portion of this activity, they navigate through these aspects in a social, collaborative, online learning environment.

Personal Interpretation of Experience

This course supports graduate students who are PK-12 teachers of all subjects in the United States. Since each classroom, school, subject, and grade level differ amongst the students completing this activity, the "personal interpretation of experience" of social constructivism is used (Richey et al., 2011, p. 130). The students in this course are considered experts of their classrooms, grade level, and content area. This expert stance is taken as no instructor of this course can be an expert in every PK-12 subject, grade level, and school/classroom setting, regardless of their experience.

As students navigate this activity, they are required to complete a beginner lesson on block coding (Welcome to Tynker Module) and another lesson found on Tynker (see Creating Directions for

DOI: 10.13001/jtilt.v1i1.6983

Students section). Offering other lessons allows students to select a lesson appropriate for their grade level, subject area, their students' prior knowledge of coding, and their own coding abilities. Although students are encouraged and recommended to complete this in their groups, they can navigate these lessons on their own. They are also provided with other activities outside of Tynker, such as Hour of Code (www.hourofcode.com/us) and the Code Studio at Code.org (https://code.org/). The Hour of Code and Code.org sites allow students to navigate various coding activities on their own without a shared username/password and provides multiple resources for PK-12 classrooms.

Since the students enter this course with various prior experiences in computer science, computational thinking, programming/coding, and drones, additional coding activities, and the option to choose from various activities, provides the students with a more personalized experience, supporting their individual coding and classroom needs. For example, the PreK teacher with limited coding knowledge can explore the pre-reader coding lessons to develop their skills and understanding of what coding looks like in a PreK classroom, while the high school computer science/technology teacher with knowledge of computer programming can explore JavaScript and Python text coding lessons to enhance their coding skills and integrate new coding lessons into their classroom.

The personalized nature of this activity, providing support for the range of prior knowledge and classroom integration experience, only works in a concrete, hands-on, active learning environment where the students are experiencing coding and possible activities that can be brought into their classrooms.

Active Learning Environment

A hands-on approach to computer science and coding is necessary for teachers to develop the skills and experience needed for classroom integration (Carnahan et al., 2016). The challenge that arose for this activity was how to create a hands-on, active learning environment in an online classroom that went beyond the web-based coding lessons to include hardware like drones. In a face-to-face class, this is easy, as the drones can be brought into class for students to interact with. In an online class, students typically learn about the hardware in a more theoretical position with little to no interaction. This activity bridges the gap of the theoretical ideas of computer science, coding, and drones to create concrete, hands-on, active learning in which the flights the students code in Tynker are flown by the instructor. As with the personalization of the learning materials, scaffolding is used to support the students in this activity.

Scaffolding

The options provided in the coding lessons allows students to enter this activity with various levels of prior knowledge and build their experiences through hands-on coding. The required Welcome to Tynker Module may be simplistic for some students but sets a common foundation for using the Tynker program and using block coding. Although most block coding programs are similar, each program functions differently. Through this initial activity, students gain an understanding of how block coding in Tynker works on their devices.

The option of the next lesson allows students to challenge themselves based on their prior coding knowledge and experience. This supports continued growth in coding regardless of prior experience and knowledge. It also allows students to experience coding that is appropriate for their classroom grade level and/or subject area.

The coding resources outside of Tynker, and access to more lessons in Tynker, support students who are interested in the topic, want to learn more, or need more support prior to coding their drones. These supports provide multiple access points for students to grow in their knowledge and skills of coding and scaffold the needed information to successfully code a drone.

Exploration of Multiple Perspectives

Finally, as the students are exploring Tynker and other coding resources and coding their drones, they are encouraged to work within their groups for support and access to multiple experiences and perspectives. Although the coding lessons and resources can be done individually, only one drone flight code is completed per group with the expectation that this be completed collaboratively.

Collaboration

Students are expected to collaborate in the creation of their code for the drone flight. The students are in pre-established groups that are set up in the first week of the course. The students are in these groups throughout the whole course, working on various activities and projects together. Since this drone activity is in week three of the course, the students already know their groupmates and have worked with them in synchronous and asynchronous class assignments prior to this activity.

To support the students in these collaborative activities, time is provided in the first synchronous class for the groups to meet, exchange information, and determine how they will communicate and complete the work. The number of synchronous classes is limited, so students have time to meet with one another and complete the group work each week.

Since most of the students are full-time PK-12 teachers, who also have families, lives, and responsibilities outside of their work and graduate education, their availability and time is limited. To support this, while also supporting the collaborative nature of this course and activity, students are provided options within the activities and directions to complete the work with their group synchronously or asynchronously. Although the directions encourage students to meet synchronously to complete the work (see Figure 3), some groups complete the initial coding lessons and plan their drone project asynchronously.

This asynchronous collaboration does not include the real-time discussions and conversations that a synchronous collaboration includes but does facilitate more planning and communication than meeting in real-time. For example, groups who complete this activity asynchronously have to set deadlines within the activity for the completion of the lessons and coding, create a document or other type of collaboration space to brainstorm the elements that will be included in the code, and assign certain roles within the group — such as the coder (i.e., the person who will actually code in Tynker).

This flexibility in synchronous and asynchronous completion of the activity supports the various lives and schedules of full-time working students and supports last minute schedule conflicts, but it does limit the drone coding activity a bit. Although the asynchronous option is available to students, by recommending the assignment be completed synchronously, most groups complete the activity in a synchronous, real-time manner.

A Limitation of Tynker

When coding for a device, like a drone, on Tynker, a limitation to the program is the students' ability to view a test flight of their code. On the screen, there is a Mambo drone sprite, but, unlike other coding programs (e.g., Scratch) where the sprites move and demonstrate the code, Tynker with the Mambo drone selected, does not preview the flight code. This limits students' ability to preview their flight to ensure it works prior to submitting it.

Drone Flights

Once the students complete their codes for the activity, the drones are flown using their flight code. The students do not have to submit their codes to the LMS or any location; they merely complete the code by the due date listed in the activity. A few days after the submission date, the drones are flown by the instructor accessing the students' codes on Tynker.

Prior to the drone flights, I reserve a room at my institution to fly the drones. The Parrot Mambo drones do not have GPS functionality, so I fly them inside (see Critical Reflection section for more details). Before the first pilot of this activity, I viewed multiple locations around the campus to determine numerous spaces that could be reserved for this portion of the activity. I also considered the ability to move furniture and request furniture changes to ensure the spaces acquired are appropriate for drones. Each semester, every attempt is made to secure a room without furniture, but this is not always possible (see Critical Reflection section).

A few days before the drone flights, the drone batteries are charged to ensure they are ready for flight on the scheduled date. In addition to charged drones, two iPads are utilized for the drone flights. One of the iPads has the Tynker app and connects via Bluetooth to the drone. The other iPad is used to record videos of the drone flights and subsequent code updates.

Using a copy of the groups' Tynker usernames and passwords (or printed; see Figure 4), I login to the first group's account via the Tynker app on the iPad

DOI: 10.13001/jtilt.v1i1.6983

and access their flight code. To access drone flights, fly the drones, and update flight codes (if needed), logging in as student groups is recommended.

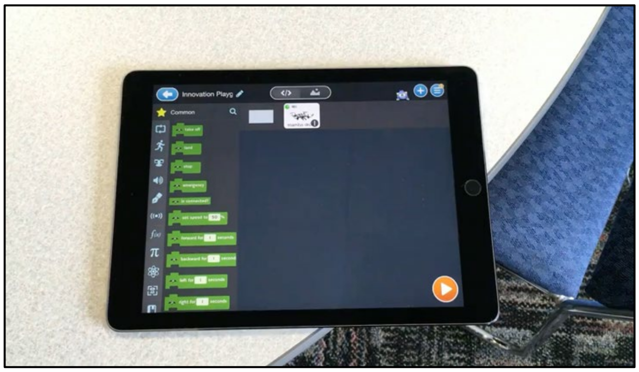
Once the iPad with the Tynker app is connected to the drone and the flight code is accessed, I review it to determine if any hardware, such as the cannon or grabber, needs to be added to the drone and how the drone was coded to start (i.e., automatically or by touching the character on the screen; see Video 4; select image or see the [drone video playlist]). The drone, once outfitted, is then placed on a table (or the floor if no furniture) in the center of the room pointing away from me. It is extremely important that you do not get in the path of the drone for safety. The iPad with the Tynker app can stay on the table while the other iPad is used to video tape the flight. Once the iPad that is taping the flight is cued up with the camera, I start the drone flight and record.

Video 4. Drone flight with grabbers and flip in hallway.

Typically, due to the constraints of the room, the drones hit a ceiling, wall, or piece of furniture if an empty room is not available. Although a predetermined room could be secured with measurements given to students to better inform their flight codes, this goes beyond the purpose of the activity (see Critical Reflection section).

If drone flights did not execute as the students planned, I record myself reviewing and updating their code, explaining why their code did not work and what they may need to consider for future implementation with their students (see Video 4 and Video 5). There are also limitations to the hardware and aeronautic flips and tricks as the drones won't flip or do any aeronautic tricks with the hardware attached. The drones are typically flown multiple times with and without hardware to showcase any aeronautic flips or tricks included in the flight code (see Video 4 and Video 5).

Each of these flights, and any updates to the flight code, are recorded. Then the recordings are edited in a video editor to create one video for each group with the flights, flight code updates, and explanations for their flight code. These videos are posted to Flipgrid for student viewing (see Video 4 and Video 5).



Video 5. Drone flight with cannon and flip in room with furniture.

Student Reflections and Integration

Flipgrid

After drone flights are recorded, I set up an activity in a Flipgrid topic that provides a prompt, an article on how to use drones in PK-12 classrooms, and the drone flight videos. Flipgrid is a video discussion platform that supports audio and video responses. Students and teachers can post pre-recorded video/audio content or record directly within the platform (review the Getting Started: Flipgrid for Educators section on the Flipgrid website for more information on how to set up Flipgrid for your classroom, https://info.flipgrid.com/getting-started.html#educator). The Flipgrid activity for this lesson allows students to watch the flights, reflect on their work, and pose potential integration ideas for their classrooms.

The prompt directs students to watch their drone flight, record their reactions, and reflect on the "fit" drones have in PK-12 classrooms focusing on the question, "If your principal/administrator purchased a Drone for your school & asked you to integrate it into your classroom & work with the other teachers to integrate it in their classrooms, what types of activities would you use/present to the teachers?" (see Figure 5). Each drone flight video is labeled with the specific group, so students know which drone

DOI: 10.13001/jtilt.v1i1.6983

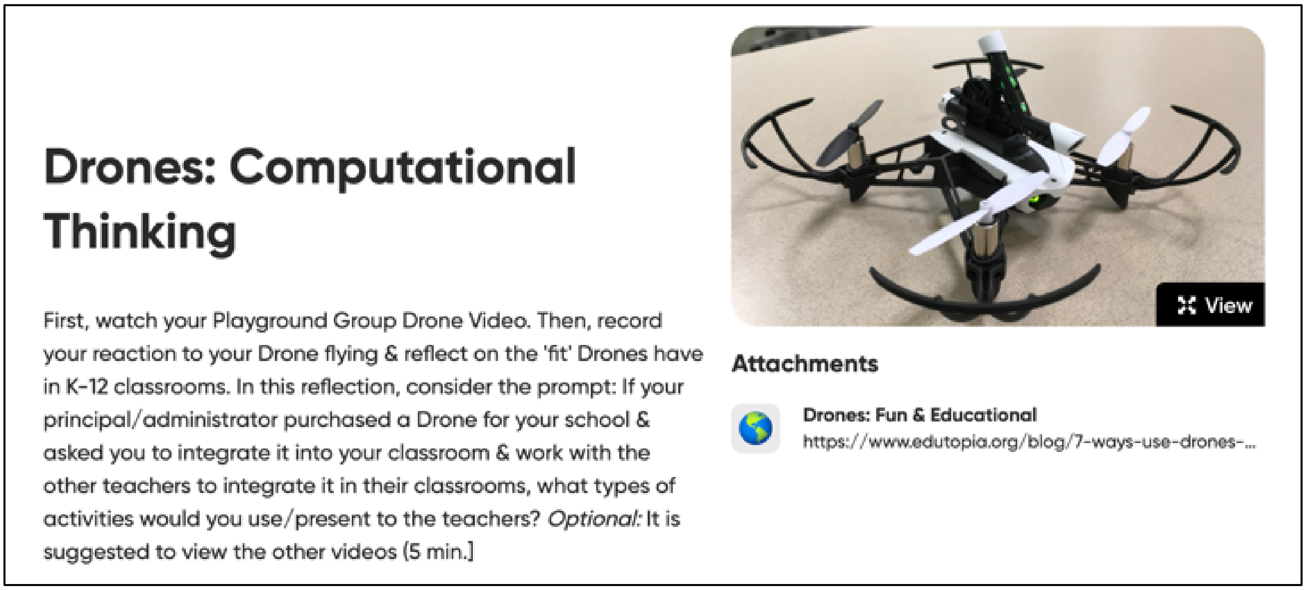


Figure 5. Flipgrid prompt for drone activity.

flight reflects their flight code (see Figure 6). Additionally, an Edutopia article on integrating drones in the PK-12 classroom (Wolpert-Gawron, 2017) is attached to the activity to ignite students' ideas and give them a starting point for their own integration considerations (see Attachments section in Figure 5).

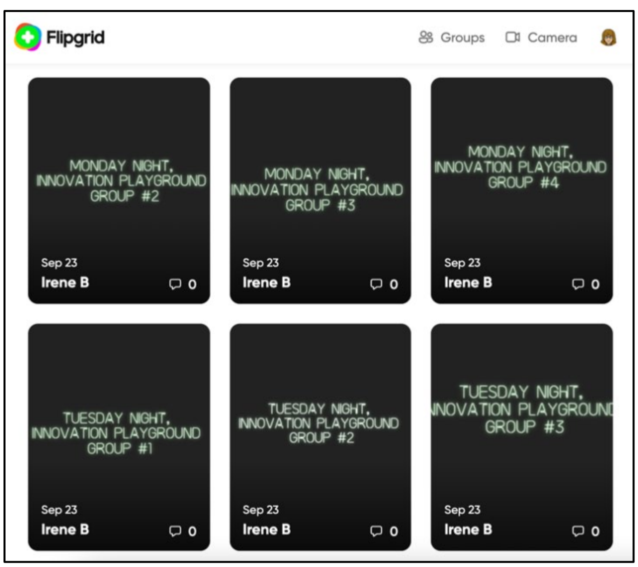
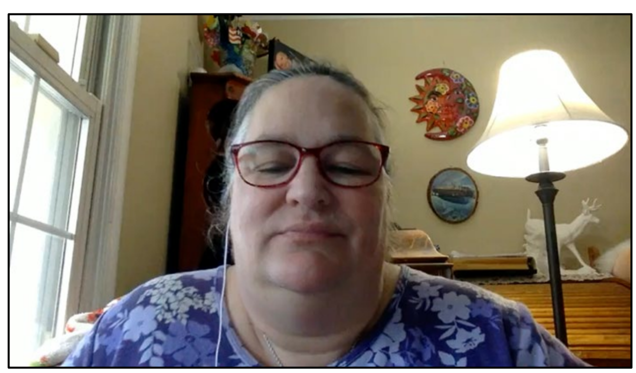


Figure 6. Recorded drone flights in Flipgrid.

Student Responses

Once drone recordings are placed on Flipgrid, students are given access to the Flipgrid topic to watch and respond. Students create video responses discussing their reactions and how drones could fit in their classroom and subject area (see Videos 6 through 10). Responses are limited to five minutes.

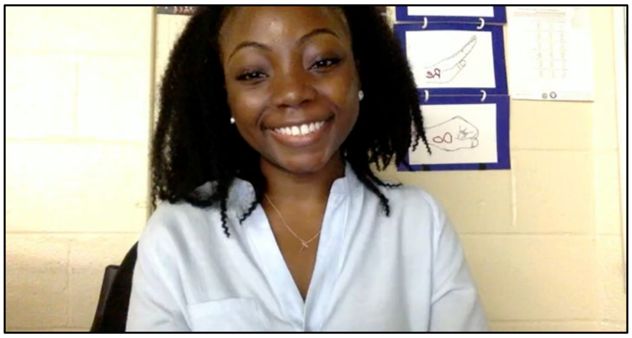
Responses are graded as participation because students are still constructing their knowledge around computer science, coding, and drones in PK-12 education (thus there are no "right answers.") A sample of these responses are provided (see Videos 6 through 10; select images or see the [drone video playlist]) to showcase elementary, middle, and high school teachers' reactions and classroom integration ideas. The students reacting to their drone flight and discussing the "fit" of drones in their classroom is the final aspect of this activity.



Video 6. Elementary school 2nd grade teacher reflection.

DOI: 10.13001/jtilt.v1i1.6983

Video 7. Elementary school 3rd grade teacher reflection.



Video 8. Middle school music teacher reflection.

Video 9. High school engineering teacher reflection.

Video 10. High school English teacher reflection.

Critical Reflection

In this activity, the students have mixed emotions about coding and flying drones. Some students are skeptical and hesitant while other students are excited. Many students state they have never coded or seen a drone before, and some students showcase the drones they have access to in their schools or personally. Over the years, there has not been an issue with buy-in from the students for this activity as most are curious about drones and have little prior exposure. In the previous iterations of this activity, the students' access to drones before or during the activity was not collected. Prior to implementing this lesson, an understanding of student buy-in, current or future access to drones, and student's prior experiences with drones is recommended.

This activity was implemented with 14 classes over the past four years. During this time, there have been various updates and changes to the activity.

The first time this activity was implemented, the class was face-to-face, so the students coded and flew their drones within one class meeting. The allotted time provided for this activity was not enough for students who had no prior knowledge/experience in coding and was too much time for the more experienced coders. This was the only time the activity was conducted with a face-to-face class, with each subsequent activity conducted online, but, regardless of environment, the need for differentiation in learning how to code was evident and became an asynchronous activity with options for individual or group learning.

During the first online implementation of this activity, two classes submitted their codes resulting in nine different drone flights. In these flights, students had the option to use the camera and send an image through Bluetooth. I quickly realized this option killed the drone batteries, which are rechargeable but take time to recharge. With nine flights and the drone batteries dying after each flight due to the images, I updated the directions, so students did not include pictures in their flights. I also began previewing the flight codes to remove the Take Picture or Fetch Picture code if it was included. During this iteration, I also realized I needed extra batteries, so I could record multiple flights at one time.

The same Tynker resources to learn the basics of coding were used in both the first face-to-face class

DOI: 10.13001/jtilt.v1i1.6983

and with the online classes, but during the first online implementation, I realized the shared usernames and passwords for groups did not allow students to redo activities already completed by their classmates. Essentially, the first student to access the coding lessons completed the activities for the rest of the group. This caused confusion for the students, especially those that did not have a strong foundation in coding. I am still using shared usernames and passwords with my groups, but since that first, online implementation, the Tynker app has updated, allowing the retry of lessons. This allows students to complete the courses and lessons in Tynker individually and asynchronously after their groupmates by selecting retry. Since the first iteration, I have also provided more resources in the HyperDoc on block coding, such as the Hour of Code and Code Studio at Code.org (https://code.org/). These resources are optional resources for the students to explore but allow them to gain a better understanding of block coding if needed.

After the first online implementation, it was clear that what I typically communicated in real time in the face-to-face classroom needed to be more explicitly communicated in the online directions, such as the access to Tynker. Another clarification for the online version of the course was the viewing of a test flight of their code in Tynker (see A Limitation of Tynker section). A more explicit statement that the on-screen Mambo drone sprite does not move, and the flight code cannot be previewed has since been included in the directions.

The biggest change and hurdle to this activity is the large, open, indoor space to fly the drones. The Parrot Mambo drones I use do not have GPS functionality, thus if I fly them outside, I run the risk of losing them. By flying them inside, there are limitations of space due to the ceilings, walls, and worse: the furniture. Finding a large, open, indoor space proved difficult before the COVID-19 pandemic. Now, it is even more difficult. For example, the Spring 2020 drone flights were conducted in a colleagues' basement and Summer 2020 drone flights conducted in a hallway (see Video 4). Now, more planning and preparation is needed to ensure access to the space and there is less flexibility of having furniture removed. I frequently have to fly the drones amidst chairs and tables, and in smaller rooms.

As stated earlier, I do not provide room dimensions nor potential obstacles to the students as this is an introductory activity to coding and drones, and I don't want the students to be overwhelmed. I want the students to focus on the code and coding process, not the limitations set by the room.

After the COVID-19 pandemic, I included a lot more explanation in my recordings when I changed the students' code to accommodate the room size. I began to explain why I am changing their code, what prompted me to change the code (e.g., low ceilings, small space, too many chairs), and how that change impacts their flight. This is met with various student reactions with some students wanting the dimensions of the room to code a better flight, and other students thankful that they didn't have to think about room dimensions while also learning how to code. This further differentiation of learning by providing room dimensions with future classes is something I am still exploring as a possible change to the activity, but I always consider the question, "What is the purpose of the activity?" This question usually leads me to omitting room dimensions as this is intended as an introductory activity with reflective integration opportunities and not an activity that tests how well students can apply mathematical principles to create a non-crashing drone flight.

References

Carnahan, C., Zieger, L., & Crowley, K. (2016). *Drones in education: Let your students' imaginations soar*. International Society for Technology in Education.

Highfill, L., Hilton, K., & Landis, S. (2016*). The HyperDoc Handbook: Digital lesson design using Google apps*. EdTech Team Press.

International Society for Technology in Education. (2017). *ISTE standards: Educators*. Retrieved January 9, 2022, https://www.iste.org/standards/iste-standards-for-teachers

International Society for Technology in Education. (2019). *ISTE computational thinking competencies*. Retrieved January 9, 2022, https://www.iste.org/standards/iste-standards-for-computational-thinking

Richey, R. C., Klein, J. D., & Tracey, M. W. (2011). *The instructional design knowledge base: Theory, research, and practice*. Routledge.

DOI: 10.13001/jtilt.v1i1.6983

Sadik, O., Ottenbreit-Leftwich, A. T., & Brush, T. A. (2020). Secondary computer science teachers' pedagogical needs. *International Journal of Computer Science Education in Schools, 4*(1). https://doi.org/10.21585/ijcses.v4i1.79

Takei, J. (2019). How our daily lives affect biodiversity. *Science Scope, 43*(3), 13-19.

Tynker. (n.d.). *About us.* Retrieved January 9, 2022, from https://www.tynker.com/about/

Tynker. (2020, November 5). *New remote collaboration features for educators!* https://www.tynker.com/blog/articles/ideas-and-tips/new-remote-collaboration-features-for-educators/

Walker, T. (2018, September 14). *#OutOfMyPocket: Educators speak out on buying their own school supplies*. NEA. https://www.nea.org/advocating-for-change/new-from-nea/outofmypocket-educators-speak-out-buying-their-own-school

Warschauer, M., & Tate, T. (2015). Going one-to-one, 2.0. *Educational Leadership, 72*(8), 60-65.

Wolpert-Gawron, H. (2017, May 26). Drones can be fun – and educational. *Edutopia.* Retrieved January 19, 2022, from https://www.edutopia.org/blog/7-ways-use-drones-classroom-heather-wolpert-gawron

About the Author

Irene A. Bal is a Lecturer of Educational Technology at Loyola University of Maryland with research interests in microlearning, micro-credentials, teacher professional development, computational thinking and computer science for pre- and inservice teachers, and innovative PK-12 classroom practices and technology. She is currently enrolled in the Instructional Design & Technology Ph.D. Program at Old Dominion University in Virginia. She can be reached at ireneabal1@gmail.com.