Core Apprenticeship Library
Apprenticeship Sector: *Coding & Gaming*

Unit Guide: *Pencil Code*

Lesson #3: *Buttons & Events*

CITIZEN SCHOOLS

Essential Question

- Do you need to be creative to be good at computer programming?
- How can I use computer coding to design new ideas/ products?

Lesson Overview

In this lesson students will continue working independently with Pencil Code Cards and create buttons, key downs, and clicks to trigger various actions. Students will be able to add sound, movement, and even extra turtles to their Pencil Code programs as they build skills in programming! Also, the apprenticeship will dive deeper into the innovation process and further practice generating a list of possible innovations and selecting one or more ideas for innovations.

Lesson Objectives

Standard #1: Citizen Schools students will generate an idea or product that suits a practical or artistic purpose.

Lesson Objective

- SWBAT think of an innovation to one or more of the Pencil Code Cards to fill a specific purpose for the user.

Standard #4: Implement problem solutions using a programming language, including: looping behavior, conditional statements, logic, expressions, variables, and functions.

Lesson Objective

- SWBAT use Pencil Code to create at least one button, key down, or click to trigger actions including sounds, movement, or making extra turtles appear.

Lesson Agenda

5 Minutes	Hook: Feature Project
10 Minutes	Introduction to New Material: Buttons & Events
25 Minutes	Activity 1: Cards
15 Minutes	Activity 2: Game OR Share Projects
30 Minutes	Activity 3: Innovations
5 Minutes	Assessment: Work Product

Lesson Preparation

- Preparation:
    - Practice the Pencil Code Cards for this week. Samples of all of the Pencil Code Cards are available at citizenschools.pencilcode.net. Username: citizenschools, Password: Citizen. Each card is saved in accordance with the file-saving procedure taught in this apprenticeship (week#/cardname). Innovation ideas are included for some cards.
    - Prepare words to add to the Word Wall.
    - Prepare copies of handouts.
    - Review student assessment work from the previous week. Check to make sure that the student mastered the concepts taught the previous week. Write a note to the student appreciating some aspect of their work, making a recommendation for further innovation, and recognizing their work and learning.
    - Pick a project from Week 2 to feature for the Hook. Ideally, feature a different student's work each week. Select a student whose work is exemplary and might push other students' innovations, thinking, and creativity further.
    - Prepare to show students how to create buttons, key downs, and clicks in Pencil Code. Note that this is the first week *without* a tutorial video. Plan to use a teacher computer connected to a projector to demonstrate the programming language.

- Co-teaching plan:
    - You may still have a few new students joining the class this week. Consider having a co-teacher pull them aside and facilitate a modified lesson to the small group combining a review of expectations from Week 1 (and having the students sign the poster) along with an introduction to Pencil Code and the cards from Week 1 and Week 2. Plan to have this same teacher teach a combination of Week 3 and 4 during Week 4 to catch these students up.
    - By Week 3 you will likely have some students who are picking programming up quickly and some who are struggling. Be cautious about the words you use to describe either group of students. Try to focus on praising hard work and effort rather than rewarding only students who accomplish tough coding challenges with ease. Consider assigning co-teachers to groups of students based on their mastery levels so that the teacher could pull two-three students together for a quick reteach if necessary. You might consider rearranging your seating assignments to group students homogeneously by ability, but this practice has drawbacks as well. For example, if all the students who are struggling are sitting together, they might be more inclined to goof off and feel embarrassed and singled out. Instead, if the students are sitting heterogeneously by ability, then a more advanced student might be able to help out a student next to him who is struggling.
    - If you have co-teachers available during Activity 3, consider assigning each teacher a small number of students to check in on. This will ensure that the teacher knows the student's project and programming goals and will be better able to tailor support to the student's needs.

Materials

Every Class:

- Computer with internet access (one per student)
- Computer with projector (one per class)
- Student folders (one per student)
    - Student Guide*
    - Pencil Code Cards completed by each student (from previous weeks)
- Previous weeks' cards organized in a binder or file box
- Roadmap to WOW!

- Expectations poster
- Word Wall
- Stickers (to indicate earning badges)

This Class:

- Week 3 Pencil Code Cards (one per student per card)*
    - (Raining Turtles, Turtle Dodge, Turtle Remote, Jamming with Keys)
- Word Wall words and definitions (user, event)
- Handout - Innovation - Lesson #3 (one per student)

*These materials are located in the lesson folder, not in the Lesson Resources at the end of this lesson.

Hook: Feature Project (5 Minutes)

Teacher's Note: *This Hook repeats each week and is an opportunity for you to highlight each student's work. Ideally, you feature a different student's work each week. Select a student whose work from the previous week is exemplary and might push other students innovations, thinking, and creativity further.*

- **Say:** Welcome back to Pencil Code! Let's begin this week by taking a look at some of the awesome work you completed last week!
    - **Play** the program from the featured student of the week.
- **Point out** why you selected this project to share. Highlight an innovation, a unique piece of code, or a way the student was thinking creatively.
- **Ask** the student who is featured if he or she wants to share anything about the project or ask for any feedback.
    - **For example,** the student could share a future innovation that he or she thought would be cool, but isn't sure how to do or the student could ask for other students ideas of what to try next.
- **Transition:** Today we're going to learn how to include buttons and events in our code. Let's get started!

Introduction to New Material: Buttons & Events (10 Minutes)

Teacher's Note: *This is the first week without a tutorial video. Plan ahead for how you will review the benefits of cueing events in code. Plan to use a teacher computer with a projector to show how buttons and events are created and used.*

- Briefly **review** commands (for) and concepts (loops, nesting loops, indenting code) covered in Week 2.
    - **Note:** You may still have new students joining the apprenticeship this week. Consider having a co-teacher pull them aside and facilitate a modified lesson to the small group combining a review of expectations from Week 1 (and having the students sign the poster) along with an introduction to Pencil Code and the cards from Week 1 and Week 2. Plan to have the same co-teacher combine concepts from Weeks 3 and 4 to teach during Week 4 to catch these students up.
- **Teach** how to add a button, key down, and click event to code.

  - **Connect** back to the word "parameter." **Review** the definition and point out how you must write in the parameters for a button, key down, or click.
  - **Highlight** that this is the first class where there is a difference between the programmer and the user. Review the definition of "programmer" from Week 1--see Word Wall. Add "user" to the Word Wall.
  - **Word Wall Definition: User** = A person who engages with your program and uses it to achieve a goal.
  - **Say** this is where programmers create something that someone else will interact with by clicking on or pressing a button or keying an event. This is the first step towards programming games, creating websites, or really programming anything for a client.
  - **Word Wall Definition: Event =** An action that occurs as a result of the user or another source, such as a mouse being clicked, or a key being pressed.
  - **Say:** Today you are adding events to your programs. You'll add buttons, key downs and clicks that trigger events or trigger something to happen in your program.
- **Review** the basic agenda for the day. Note that this is generally the schedule that we will use in this class each week. The class is very much run workshop-style and students will guide their own work.
  - **Note:** Remember to review, reteach and practice all procedures you taught in Week 1, if necessary. Students will naturally forget from week to week and may need reinforcement. By this week you can expect to briefly remind or review the procedure rather than go through a full reteach. However, if a full reteach is needed, then take the time to do it.
  - Agenda:
    - Work time
    - Pause for sharing projects
    - Think about innovations
    - More work time
    - Save an assessment for the week
- **Transition:** Cue the procedure to get folders, turn on computers, and get to work!

Activity One:
Cards
(25 Minutes)

Teacher's Note: *By Week 3 you will likely have some students who are picking programming up quickly and some who are struggling. Consider assigning co-teachers to groups of students based on their mastery levels so that the teacher could pull two-three students together for a quick reteach, if necessary. You might consider rearranging your seating assignments to group students homogeneously by ability, but this practice has drawbacks as well. For example, if all the students who are struggling are sitting together, they might be more inclined to goof off and feel embarrassed and singled out. Instead, if the students are sitting heterogeneously by ability, then a more advanced student might be able to help out a student next to him who is struggling.*

Teacher's Note: *You now know your students' personalities, likes/ dislikes, strengths, etc. Use this knowledge to plan how to instruct each student in this workshop environment. Plan to encourage students to try specific cards based on what you know about them. Build investment in your apprenticeship by targeting their interests and tapping into their interest in expressing themselves and being creative.*

- **Hand out** Jamming with Keys Card for all students to start with today. This is the easiest card to master in this section.
  - **Give** students 25 minutes of work time.
    - Students should use this time to complete cards and practice creating buttons, key downs, and click cued events. Students can either complete two-three cards and then start

innovating off of them, or students can complete all of the cards. Students should try to complete one of each type of event--a button, a key down, and a click card.

  - Once students complete the Jamming with Keys Card, they should place it in their "Completed Cards" section in their folder and follow the appropriate procedure to select another card.
  - **Remind** students to NAME and SAVE their projects regularly. If needed, set the timer to cue you and the students to remember to save their work. Remember, all projects saved today should start with "week3/" to create a "week3" file folder or directory.
  - During this work time, CTs should be rotating around the room, checking in with students, assisting as needed, and encouraging students to try a new card or a new step in their programming.
    - Encourage students to switch back and forth between text code and block code, as practiced in Week 2. There are benefits to each.
- **Bring** the group back together
- **Say:** We are going to have more work time soon, but first I want to pause and highlight some great work that's being accomplished and think more about the innovations we can make in our projects.

Activity Two:
Game or Share Projects
(15 Minutes)

Teacher's Note: *Use this time to either share projects that students are working on or share a project that pushes their skills further.*

- **Say:** Let's take a look at xx student's project this afternoon…
  - **Share** two-four different students' projects. Highlight how they used buttons, key downs, and clicks. Show how they are using sound or adding movement. Additionally, connect to past lessons and highlight ways that it might have been easier to use text code vs. block code while writing the code or ways that loops have been integrated into the events.
  - **Share** feedback and ideas for further work.

Alternatively, if you'd rather play a game than share student work, consider playing a quick "Guess the Code" game.

- **Say:** I'm going to show you three sets of written code and three programs being run. The game is to guess which code goes with which program. These are a little trickier than last week!
- **Use** the following links to two rounds of "Guess the Code"
  - Day 3, Game 1
  - Day 3, Game 2
  - **Connect** to Lesson 2 by highlighting the use of the variable "x" and the parameters of each loop.
  - Highlight the "replay" button at the bottom. Note the benefits of this button and how it makes the game more user friendly.
- See all games available for this section at http://inventgame.pencilcode.net
- **Connect** to concepts from today and recognize the innovative ways these programs used loops.

Activity Three:
Innovation
(30 Minutes)

Teacher's Note: *If you have co-teachers available, consider assigning each teacher a small number of students to check in on. This will ensure that the teacher knows the student's project and programming goals and will be better able to tailor support to the student's needs. If you notice that some students are not able to make as much progress in the class as other students--or as much as you, the teacher, would like--consider prioritizing cards for these students to complete. What are the top two cards you'd like them to master? Additionally, if they are invested and have access to a computer with internet, make extra copies of the cards and let them take them home to work on during the week.*

Teacher's Note: *This activity is written to allow students to choose to make their idea from the handout a reality or to continue working on their other cards and innovating on those cards. If you would like, require students to create their innovation to create a useful program. It is recommended that you make this decision in the moment. If students are dying to get back to their cards and projects they had been working on, let them. If they need a new goal, then expect them to make their chosen innovative idea a reality.*

- **Say:** Today we're going to dive a little bit deeper into how we pick what ideas to innovate on. Let's start by talking about it. When you've finished a card and you're thinking about what you could try next (what you could change or improve on the program), what do you think about?
    - **Solicit** answers. Sample answers might include: I think about what I want to do, I think about what would be cool for it to do, I think of something that I've seen or heard about elsewhere that I want to try to do here, etc.
- **Say:** If you were designing a video game, what might you think about?
    - Tease out that, as a video game designer or programmer, you'd also think about what would make it easy to understand how to play--as well as cool and fun for the gamer--not just the programmer.
    - Note that when we are computer programmers, we almost always are writing programs for other users and not ourselves. That means, that when we are innovating, we have to think about the user experience, not just what we think will be cool.
    - **Connect to Lesson 1:** In Lesson 1 we talked about the programmer as the artist, engineer, and designer. This is really the role of the designer: to hone a program to make the user experience as good as possible.
- **Highlight**: When we are thinking about improving a computer program, we're thinking about what would make it more interesting, what would make it more useful, and/ or what would make it easier for the user to use.
    - **Connect to future lessons:** We'll talk about this more as we learn more ways to write interactive programs (e.g. games)
- **Say:** Part of innovating is to try and make computers even more useful to people.
- **Share an Example:** the first phones had a receiver that was connected to the base and the base was connected to the wall. Then, someone created a phone that connected to the wall with a cord. Then someone innovated further to create a phone that was wireless--the receiver didn't have a cord to connect to the base! Innovations kept happening, and eventually we ended up with cell phones.
    - Even the cell phone has been through a lot of innovations! Does anyone remember when a cell phone was just a phone? Someone, somewhere, thought to themselves, "Gee, it would be so handy to do more with the phone in my pocket--take photos, keep a calendar, etc." Thus the innovations were thought of to make the phone in our pocket ever more useful to the user!
    - **Note:** Consider showing the pictures included in the Lesson Resources as you walk through innovations to the phone. Many of your students may not remember a time when cellphones were just phones, much less when rotary phones were the norm.
- **Say:** Today, I want to challenge you to think about how you could write a program that would be useful to someone. Often a client comes to a computer programmer with a request to write a program for a particular purpose. Other times, you (the computer programmer), dream up something that will be useful to other people--to market and sell the program as a business. Think of various apps for your cell phone or the computer programmers who created Facebook or online textbooks. Each was created for a specific useful purpose.
- **Say:** Let's use the following scenarios and think of programs we could create for these clients or ideas.

  - **Brainstorm** a few ideas for programs with the students for each of the following scenarios. Ask the students which Pencil Code Card they might start with and what innovations they might make to write a program to suit the purpose.
    - A teacher wants to have the turtle write the homework on the screen.
    - A local business wants you to create a logo for a new sign that will show the turtle drawing the logo as part of the advertisement.
    - You want to create a game to add features to a character.
  - **Say:** See how for each of these problems, we could think of a few different ways to innovate off of something we already have to create something new and better that would be useful to the person in the scenario? This is our task for our innovation this week.
    - **Say:** We brainstormed some great, innovative ideas to each of these scenarios. After we brainstorm, we have to narrow in on one idea and then work to create it. Here's an example of what an innovative idea brought to life could look like.
    - **Show** sample program to represent the innovation solution to the third scenario in citizenschools.pencilcode.net titled week3/snowmangame. Note that this combines cards/ concepts from Week 1 Build a Snowman, Week 2 loops, and Week 3 buttons. Encourage students to combine concepts as they innovate and get creative.
- **Hand out:** Innovation - Lesson #3
  - **Say:** First, for each of the "clients" on the handout, brainstorm a list of possible ideas for a useful innovation reviewing your cards. Then circle the top idea that seems most feasible, has the least limitations, the most advantages, is most interesting to you, and is most useful to the user. Star the one you'd be most interested in and able to create. Then, if you'd like, spend some time working on creating it! If you'd rather keep working on an earlier program or another card, that's OK, but be sure to use this time to think about how you can change the cards to make them better in some way. Be sure to use this time to innovate!
    - **Note:** If students have an alternate idea of how to create a program that is useful for a specific purpose, they can write it at the bottom and work to create that program.
- **Give instructions:** Step 1: complete the Innovation Week 3 Handout. Step 2: Create your innovation idea OR continue working on previous programs focusing on innovative ideas.
- **Give** students 10-15 minutes work time.
- **Remind** students to NAME and SAVE their projects regularly.
- During this work time, CTs should be rotating around the room, checking in with students, assisting as needed, and encouraging students to try a new card or a new step in their programming.
  - Check in on the handout, Innovation - Lesson #3 also.
  - Ensure that students have generated at least one idea for innovation that will make their programs more useful to someone.
  - CTs should also hand out stickers to award badges for the day. Students will likely complete all three challenges for Badge #3 today: button, click, and keydown. They also may add music or sound and complete the music challenge in Badge #4. If they haven't completed all the challenges for Badge #2 yet, that's OK, but encourage them to knock those out too. If needed, explain what the text code is that they need to use. They may be missing "pen" because they haven't used pu() or pd() as these are not represented in the cards. It is more likely that they use these during implementing innovations.
- **Bring** the group back together and **transition** to the Assessment.

Assessment:
Work Product
(5 Minutes)

- **Say:** Today I will collect the Innovation - Lesson #3 handouts along with you submitting a project for your assessment. Be sure to pick the project that you are most proud of from today. Pick that project now and save it as "week3/assessment." In your student folder, fill out the Week 3 Assessment section.
  - **Walk students through** how to fill out the Week 3 Assessment written portion, if necessary. (See Student Guide).
    - I'm proud of this program because ______________________________
    - How does your program help the user? __________________________
    - Which of the following does your program contain?
  - **Say:** Before next week, I will check out your work and next week I'll give you some feedback! Also, next week we will get a chance to share some of our work with each other. We will learn new programming techniques and have new cards to create.
  - **Collect** student folders and the Innovation - Lesson #3 handouts.
  - **Note:** If there is time available, share one or two student projects now.
- **Connect to WOW!** We'll definitely want to include buttons, key downs, and clicks for events in our final WOW! Portfolios, not to mention innovations that create programs that are useful. WOW! here we come!
- **Connect to Week 4**: Next week we will learn how to create functions and more ways to do cool stuff in Pencil Code!
- **Ask** students to follow the procedure for shutting down the computers and cleaning up for the day.
- **Ensure** all students have received stickers for the badges they have earned!

Word Wall - Lesson #3

User
A person who engages with your program and uses it to achieve a goal.

Event
An action that occurs as a result of the user or another source, such as a mouse being clicked, or a key being pressed.

Handout - Innovation - Lesson #3

Name:____ _

Innovation Ideas for the Client: In the table below, **write down as many ideas as you can think of** to create an innovation on one of the Pencil Code Cards to make it useful for the purpose stated.

Client	**Innovation Ideas** (Which card would you start with? What would you do to make it different and better?)
The basketball coach wants a cool screensaver.	
A teacher wants to create a program to play on the projector when he wants to get the students' attention.	
The local library wants to create a cool image for their billboard outside to advertise summer reading.	

Select an Idea: Consider which ideas are most feasible, most interesting to you, and most useful to the client. **Circle the top idea for each client.**

Consider Interest: **Star the one idea you'd be most interested AND able to create!**

(Optional) Do you have another idea of how to innovate on a card to make a program that would be useful to someone or to yourself? If yes, write your idea down here and feel free to work to create that program!

____ _

____

_

____

_

800px-Ericsson_Dialog_in_green.JPG

cl-2200xp-as-300.jpg

t1larg.martin.cooper.jpg

startac.jpg

bignokia.jpg

iphone-1-vs-iphone-6-vs-iphone-6-plus.jpg