FIRIA LABS

LESSON: CodeX Winter Holiday Tag Project

Time: 45 minutes

Overview:

This is an easy coding lesson that involves the entire class and uses radio communication to pass around a pixel image of a present. Students can just pass around the gift, or you can turn it into a game where the game ends by stopping the passing. Whoever has the present can either be the winner or the loser, depending on how you want to organize the game.

Three extensions are provided, which elevate the project to an intermediate level. Each extension shows a different version of the game. Instructions are also provided to the teacher to make this visual game accessible to vision-impaired students.

Coding Objectives:

- I display a pixel image.
- I can use a Boolean variable to determine which image to display.
- I can convert an integer to a string.
- I can use a compound condition.
- I can use an infinite loop to send and receive radio messages.
- I can generate a random number.
- I can define and call a function with a parameter.

Grades 6-8 CS Standards:

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

2-CS-03 Systematically identify and fix problems with computing devices and their components.

2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.

2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.

Grades 9-10 CS Standards:

3A-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.

3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.

3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, or independent but interrelated programs.

3A-AP-18 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

Grades 11-12 CS Standards:

3B-DA-06 Select data collection tools and techniques to generate data sets that support a claim or communicate information.

3B-AP-10 Use and adapt classic algorithms to solve computational problems.

3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.

Preparation:

- Download slides
- Download the starter code
- Be familiar with the final code
- Read through the teaching guide
- Optional: Review instructions for visually-impaired accessibility.

In the folder:

- Holiday Tag project slides
- Holiday Tag final code
- Holiday Tag code with extensions

Agenda:

- Warm-up (5 minutes)
- Complete program using slides (40 minutes) – could be longer with extensions

Teacher Notes:

- This lesson is designed as a class project, with individual students creating their own programs.
- Almost all mistakes made by students are typing mistakes. If students get errors when they run their code, first look over the code for spelling, punctuation and especially indenting.
- This project uses images and pixels. If you have a visually impaired student, instructions are given at the end of the teaching guide on adding a beep as an audio clue with the image.
- The project slides include three extensions. Any or all of them can be done. They do not all have to be done, or in the order given.
- Extension 2 does build off of Extension 1. Extension 3 can be added to either Extension 1 or Extension 2.
- If your students have done Pixel Art, or you want them to do pixel art, you can extend this project into two days. Students can each design their own holiday pixel art. Then do a Save As on the code and use it for the Holiday Tag Project instead of the pixel art gift that comes with the starter code. If students create their own images, they will not pass them to other students, but when they are tagged, their own image will be displayed on the screen.

Project Extensions:

- Extension #1: Use an additional CodeX as a control. The first CodeX still starts the game, but the control CodeX stops the game. It is not a player in the game of tag.
- Extension #2: Add a timer to the code, used by the first CodeX. Generate a random amount of time to play the game. When time is up, see who is “holding” the gift.
- Extension #3: Add a counter to each CodeX. When the game ends, display the number of times each CodeX was “tagged.”

Other Extension Ideas:

- Have students create their own pixel art images to display on their own CodeX devices.
- Learn about displaying JPG images on the display screen. Have students find an image and compress it for the CodeX.
- Use a button press to end the program and clear the screen, or display an ending message.
- Make a list of built-in images. After the game ends, someone has the gift. After a short delay, randomly select an image from the list and display it, like the present is opened.

Cross-curricular Connections:

- **LANGUAGE ARTS:** Have students write a story about gift giving, or playing a childhood game, or a holiday memory.
- **MATH:** Extension 3 adds a counter to each CodeX. Play the game several times and keep track of the count (tag) each time. Create a graph. Calculate the average, mean and mode.
- **MATH:** After extension 3, a counter is added to each CodeX. Keep track of the counts, and calculate probabilities of each CodeX ending up with the present.
- **COMPUTER SCIENCE:** The messages sent in this project are open, without security. Have a discussion on digital security measures and/or encryption. This also meets additional CSTA and computer science standards.
- **COMPUTER SCIENCE:** Computer data is digital, but radio waves are analog. Have a lesson about converting analog and digital data.
- **SCIENCE:** Learn about radio signals.
- **SEL:** Have a discussion about being gracious during gift giving, or winning and losing.

Teaching Guide

Warm-up (5 minutes)

Slide 2 – You can start with a short discussion on gifts during the holiday season. Also, slide 2 mentions the game “hot potato.” If students don’t know what that is, you can give a short explanation or example. Or you can change the wording to a game of tag, or pass the present, or anything that makes sense to your student and age group.

Create/Run the Program (40 minutes)

💻 For this project, it is best if each student has his/her own CodeX. They will create their own program and use it with the CodeX device to play the game.

Students should already have an account in CodeSpace. The programming portion may take longer if your students are doing the extensions.

Students will need access to the starter code. Download the file and then make it accessible to students.

💡 Teaching tip:

This project is not included in CodeSpace. Download and follow the slides. They include step-by-step instructions as well as code segments to guide students through the program code creation.

This program is a class project. Show the slides on a large screen or monitor and have the class work on each step together. Keeping the class together and letting them help each other is the suggested teaching delivery system.

💻 Slides 3-5

These slides give instructions on getting started. Students log in and use the sandbox. They need to create a new file. Then they copy and paste code from the starter code text file (holiday_tag_starter). They should run the code and see the gift on the display screen.

If students are using their own image, they can replace the gift variable with their own image.

💻 Slides 6-7: Step 1

As a class, determine the number of players, and assign a unique number to each student, starting with 1. The last number assigned should be the same as the number of players. Add these constants to the code.

💻 Slides 8-9: Step 2

A Boolean variable is used to keep track of when the CodeX has the present. Every CodeX should start False except the first CodeX.

💻 Slides 10-12: Step 3

Define a function for displaying an image on the screen. If the CodeX has the gift, it is displayed, otherwise a target is displayed. You can use a different image if you want. Students will run the code at least twice to see both images (once for True and once for False). Students can either use two numbers for MY_NUM (1 and a different number) to get a True and False, or change the if statement to if MY_NUM == {their number}. But they need to remember to change the code back after testing.

Code is given for turning on the pixels that students may not be familiar with. It uses a list for the colors. They can use different colors if they want, or even turn on each pixel individually.

💻 Slides 13-14: Step 4

A brief introduction to radio communication. Then students add code to turn on the radio and set the channel. Everyone in the class will be on the same channel.

💻 Slides 15-18: Step 5

An explanation about data types string and integer and the need for conversion. The main program will be modified to use an infinite loop, call the function, and listen for a radio signal.

💻 Slides 19-20: Step 6

Add code to send a radio signal. This should only happen when the CodeX has the gift (have_present = True). The placement of the if statement is important. It should be the first block of code inside the while True loop.

💻 Slide 21: Try it!

This is their first run-through as a game. What will happen is someone will press the button to pass the gift, but it won’t end up anywhere. That is because the random number generated was their own number, but they updated have_present to False. So the present is lost. This problem is fixed in the next step.

💻 Slides 22-23: Step 7

Fix the problem by adding a while loop that gets a new random number if the current is the same as the CodeX number. Now the game should work correctly when it is tried again.

💻 **Slides 24-25: Finished!**

Students play the game. In this basic game, there is no end. It will continue until the CodeX is physically stopped. Some suggestions for playing the game are given, like Tag and Musical Chairs. You can also come up with your own game, or just let students pass the present around.

💻 **Slides 26-42: Extensions**

Three extensions are given. You can choose to extend the project or keep it basic.

- Extension #1 - add an additional CodeX that doesn't play the game but controls when to stop the game.
- Extension #2 - add a timer to stop the game after a random amount of seconds.
- Extension #3 - add a counter for each CodeX and display the number of times it was "tagged" or received the present before the game ended.

Additional suggestions are given in the teacher notes.

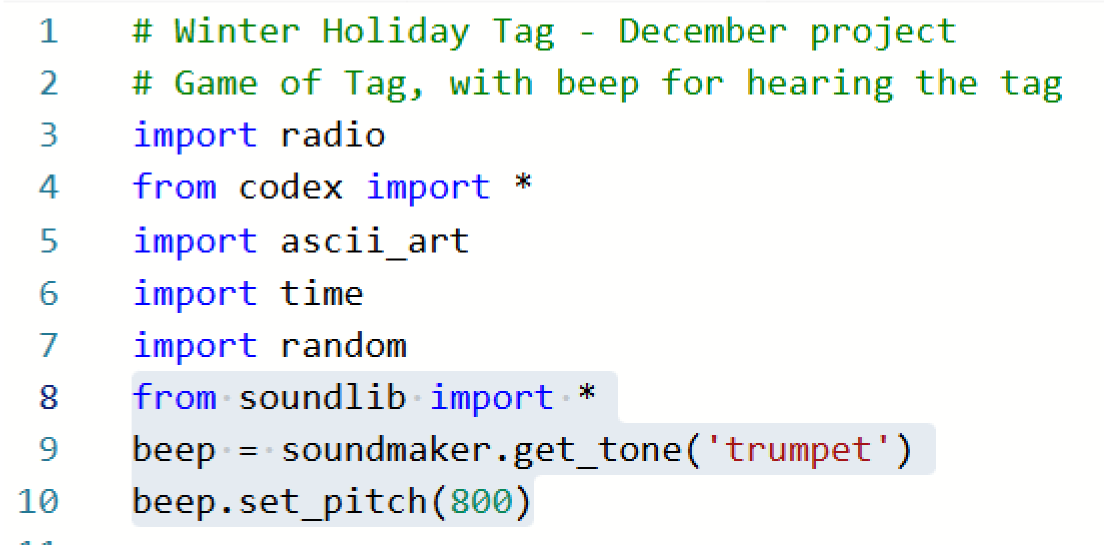
Wrap-up / Optional (5 minutes)

💻 The project doesn't require a wrap-up. You can use an extra-curricular activity from the list above, or have students reflect on the project and something they learned.

Audio Accessibility

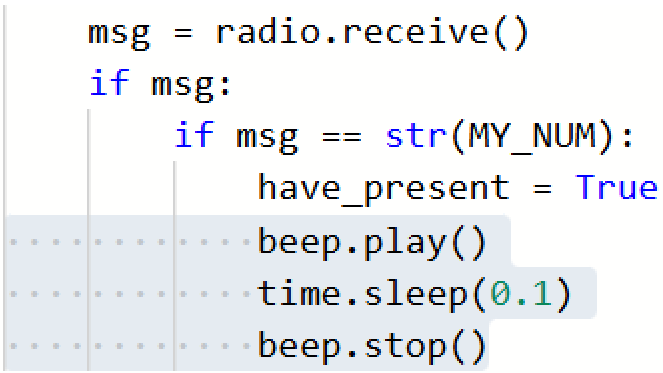
😎 The project uses images and colored pixel lights to indicate if a CodeX has the gift or doesn't have the gift. If you have a student who is visually impaired, they may have difficulty with seeing the images. You can help the student add a few lines of code to their program that will add in a beep when the gift is passed to their CodeX, which can help them know when to press the button and pass the gift.

Add the last three lines of code to the import section of the program:



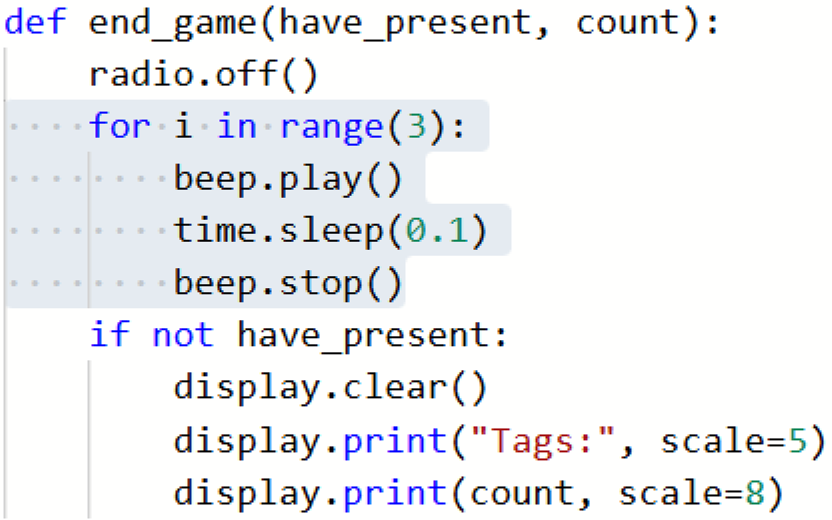
```python
# Winter Holiday Tag - December project
# Game of Tag, with beep for hearing the tag
import radio
from codex import *
import ascii_art
import time
import random
from soundlib import *
beep = soundmaker.get_tone('trumpet')
beep.set_pitch(800)
```

Add the last three lines to the if statement that receives a radio message:



```
    msg = radio.receive()
    if msg:
        if msg == str(MY_NUM):
            have_present = True
            beep.play()
            time.sleep(0.1)
            beep.stop()
```

If you add extension 1 or extension 2, you can add this code to indicate the end of the game. It will be added to the end_game() function:



```
def end_game(have_present, count):
    radio.off()
    for i in range(3):
        beep.play()
        time.sleep(0.1)
        beep.stop()
    if not have_present:
        display.clear()
        display.print("Tags:", scale=5)
        display.print(count, scale=8)
```