KNOW THE CODE

If you have ever experienced a Thunder game at the 'Peake, you know that the arena can get really loud. The sounds of whistles, buzzers, messages and announcements all mix with thousands of fans clapping, cheering and chanting. It is a lot to take in and some of the information can be hard to understand. As loud as the game can be, some messages require no sound at all. To keep up with it all, you have to know the code!

Codes are unique systems of signs or symbols, each with a specific meaning. They are necessary for a variety of tasks like writing music, passing secret messages and providing instructions to computers. Even basketball teams use codes to help them effectively communicate across a noisy arena.

HERE'S WHAT YOU'LL NEED:

- **Charts** *(included)*
- **Pencil**
- **Journal**
- **Stopwatch**
- **Waste basket**
- **Ball or light object** *(such as paper wadded into a ball)*
- **Masking or painter's tape**

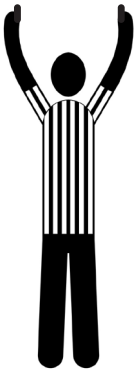
WARMUPS

Picture yourself in the stands, watching your first Thunder game. The players take their places and a referee with long pants and a whistle tosses the ball up signaling the start of the game. As you watch, you notice that there is a lot more going on than just players moving up and down the court. From hand signs and plays yelled by the coach or point guard, to body signals made by the referee, messages are being clearly conveyed to the players in spite of the noise of the arena. At first you might not understand what all of those shouts and signals mean, but over time you will begin to see patterns between the gestures and the action on the court allowing you to understand the information hidden in the code.

Learning to recognize and understand these signals is actually quite similar to when you learned to read. Once you were able to identify the shapes as letters, each with their own sound, you began to understand how they worked together to form words.

Take a moment to look at two different signals the referees may use during the course of a game.

A referee holds one hand up similar to how a police officer may signal to stop approaching traffic. As you look down at the court, you notice that the players have moved to their benches and the countdown clock has stopped. What do you think the signal means?



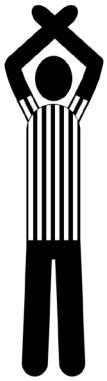
In this example, the referee makes a "thumbs up" with his hands and then moves them above his head, so that his hands look like they are jumping. On the court, two players move to stand on either side of the referee while the others gather close. The referee tosses the ball up between the two players and the game continues. What do you think this signal means?

At the back of this activity there is a chart showing a variety of signals that referees may use in a game.
Let's use the referee signals in our first activity. This warmup will be similar to the game "Simon Says," but you will be using a different code. Here's what you do:

As a large group, select four or more of the signals from the chart.

Choose a response that the group will give for each referee signal you want to use.

For example, if you choose the signal for double foul where the referee crosses their fists and forearms above their head in an "x" like this:



You could decide to have everyone in the group respond by crossing their arms in an "x" across their chest.

You may choose other responses to require more movement and even have the group switch places if space allows. After all of the signals and responses have been selected, select a person to act as the referee. Set a timer for three minutes. The referee will give signals and the group will respond as quickly as possible.

If a member of the group gives the wrong response, the referee can signal a foul (a closed fist in the air) and that member of the group should exit the game.

Let's increase the difficulty of the game:
Referees use these body signals to communicate during a lively and loud situation, so let's add some noise. Select two people from the group and have them stand on each side of the referee making loud noises and silly movements while everyone else watches and responds to the referee's signals.

Repeat the warm up activity with the noise and visual distractions.

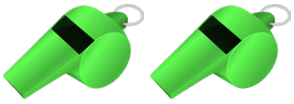
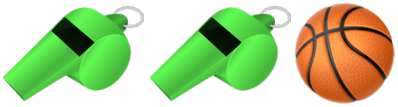
How did the distractions affect your ability to follow the code?

GAME TIME

Coding or giving directions to computers to perform tasks is similar to the game you played in Warmups. A computer programmer codes by creating and uploading a set of step-by-step instructions in a form that a computer can understand. This code tells it exactly what to do in response to a specific input.

Pro Tip: This activity requires groups of at least three. Before starting, create a human-sized coding grid for each group by taping off a 4-foot by 8-foot section and dividing it into 1-foot squares. Mark a starting place and place a wadded up piece of paper or ball in a nearby square and a box or waste basket somewhere on the other side of the grid to act as the basketball goal.

For this activity, we have created the following coding elements using a basketball, whistle, and a sneaker emoji.

EMOJI CODE	INSTRUCTION
	**Go**
	**Stop**
	**Pick up ball**
	**Drop ball**
	**Turn to the left 90 degrees**
	**Turn to right 90 degrees**
	**Move forward one space**

How would you write the following steps in "emoji code"?

- Go
- Two Spaces
- Stop

Assign one of the following roles to each member of your group:

- One person will act as the **computer programmer** and encode or convert a set of instructions into an "emoji code" for a computer.
- Another person will act as the **computer** and decode the "emoji code," converting an encoded message back into instructions that the robot can understand.
- A third person will act as the **robot** and will follow the decoded message in an attempt to perform a task.

The **computer programmer** starts the activity by first observing the relative location of the items in the gridded area and then creates an **algorithm**, or set of steps, that will allow the robot to successfully find a ball, pick it up, walk it to the basket and drop it inside. Using paper and a pencil, the programmer writes the proposed steps using the emoji code. The finished code is then handed to the person acting as the computer.

The **computer** decodes the message and reads it aloud without changing or adding additional instructions.

Without asking questions for clarification, the **robot** performs each step in the directions exactly as the computer reads them.

Was the robot successful? If not, the computer programmer can **debug** or make adjustments to the code and then retest the program.

It may take several attempts before finding success. After each attempt, adjustments can be made.

Take turns playing the role of programmer, moving the location of the basketball goal with each role change.

ANALYZE THE REPLAY
What happened?

How did your observations of the gridded area help you figure out the order of step-by-step instructions?

Did you overlook or miss any important details? How did that affect your code?

Did you have to debug to your initial code? Did you change the order or add additional steps?

Did the process get easier as you went along? Why do you think it did or did not?

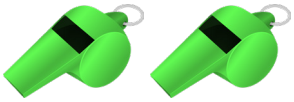
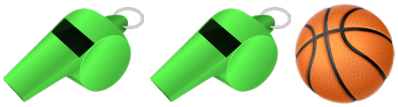
Share and compare your experiences with the other teams. How were they the same? How were they different?

Based on what you've learned so far, why do you think programmers frequently update the coding for their apps?

OVERTIME
Let's take it further

In this activity we are going to see how the introduction of conditional statements affects the coding process. Think of a conditional statement as an "if/then" statement. The code questions whether a certain condition exists, then it commands a specific action based on whether or not it does. Let's start by adding three new symbols to our emoji code to represent the conditional statements. Remember this will allow the person acting as the robot to respond a certain way if the condition is true and another way if it is false.

CONDITIONAL STATEMENTS			
EMOJI CODE	**IF**	**THEN**	**ELSE (if there is not)**
	**IF** the ball lands in the basket	**THEN** add two points to your score	
	**IF** less than two minutes have passed since the program began	**THEN** go back to the start to repeat the instructions	
	**IF** there is a ball in front of you	**THEN** pick it up	**Move** forward 1 space

EMOJI CODE	INSTRUCTION
	**Go**
	**Stop**
	**Pick up ball**
	**Drop ball**
	**Turn to the left 90 degrees**
	**Turn to right 90 degrees**
	**Move forward one space**

Set up the grid area for this activity just like the one for Game Time, but this time place several additional balls, or wadded up papers, throughout the grid. This challenge is timed so you will also need a stopwatch or timer set for two minutes.

Assign a role (programmer, computer, robot) to each member of the team.

The goal of this game is to score as many points as possible in the allotted run time. The computer programmer's job is to write the code that will allow the robot to pick up as many balls as possible and then get them into the basket before the time is out. Remember that the robot must follow the written code exactly.

Once again, the **programmer** starts the activity by observing the grid-covered playing field and then encoding a set of step-by-step instructions for the computer. Once the code is finished, the programmer hands off the instructions and starts the two-minute timer. The **computer** decodes the information and reads the steps for the **robot** to perform.

Pro-tip: If a classmate is available, they can serve as time keeper and score keeper like in a basketball game.

How did using conditional statements make writing a program more difficult?

How did using conditional statements make writing the program easier?

Did using the conditional statements make achieving the task easier?

Can you think of other conditional statements that will make encoding for this game easier?

When do you think conditional statements might be used in computer programming?

Do you think basketball referees, coaches and players use the type of thinking needed to code, encode and decode? If so, how?

COACH'S CORNER

Additional information and explanations for parents and educators

Basketball players have the ability to make quick decisions and respond to the action on the court very quickly. Knowing a common language that allows their teammates and coach to communicate quickly and efficiently is important. Code words, signs and symbols keep messages clearly definable, brief and protected from the opposing team. This makes it easier for the team to change up plays and strategies quickly. Taking the time to work bugs out of plays and adjusting them step-by-step so that they work efficiently during practice can make for successful results during a close game. Even conditional statements can be used to help players know how to react during a play if the opposing team changes defensive schemes.

This activity introduced students to basic coding concepts. A recent study on the effect of unplugged coding activities on the computational thinking skills of middle school students indicated that activities like this one not only improve computational and critical thinking skills but also have a positive effect on student creativity, algorithmic thinking and collaboration.[1]

For additional unplugged coding opportunities that can be used in the classroom, check out these resources:

- CS Unplugged (https://csunplugged.org/en/) is a collection of free teaching material that teaches computer science through engaging games and puzzles that use cards, string, crayons and lots of running around.
- CS Fundamentals Unplugged (https://code.org/curriculum/unplugged) is a similar collection of teacher material from code.org.
- Scratch (https://scratch.mit.edu/about/) is a block-based visual programming language can be used online or offline.

DO YOU WANT TO LEARN MORE?

Research: Algorithm, Cipher, Code, Conditional Statement, Cryptography, Decode, Encode, Encryption, Programming Language, Loops

OKLAHOMA ACADEMIC STANDARDS

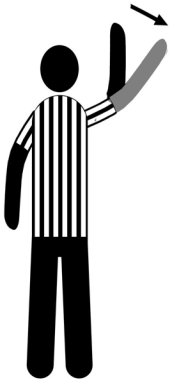
STANDARD	4th Grade	5th Grade	6th Grade
SCIENCE			
PS4-3- Waves and Their Technologies for Information Transfer	●		
COMPUTER SCIENCE			
CS.HS.1- Hardware & Software	●	●	
AP.C.01- Control	●	●	●
AP.M.01 - Modularity	●	●	●
AP.PD.03 – Program Development	●	●	

1. B. Tonbuloğlu, İ. Tonbuloğlu, (2019) T*he Effect of Unplugged Coding Activities on Computational Thinking Skills of Middle School Students Informatics in Education*, 2019, Vol. 18, No. 2, 403–426 Retrieved from: https://www.mii.lt/informatics_in_education/pdf/infedu.2019.19.pdf

A

TIME OUT

B

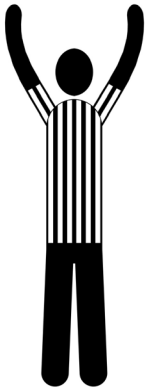


TIME IN

C

3 POINT ATTEMPT

D

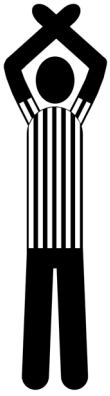


3 POINT SUCCESS

E

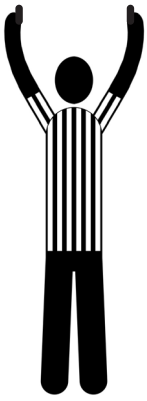
PERSONAL FOUL

F



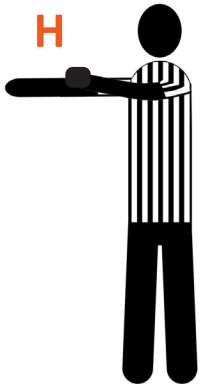
DOUBLE FOUL

G



JUMP BALL
(THUMBS UP)

H



CHARGING
(CLENCHED FIST)

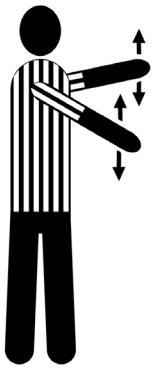
I

HOLDING

J

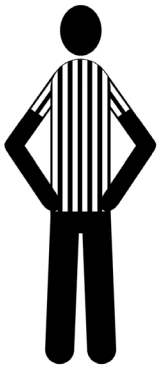
TECHNICAL FOUL

K



ILLEGAL DRIBBLE
(CALLS TEAM COLOR)

L

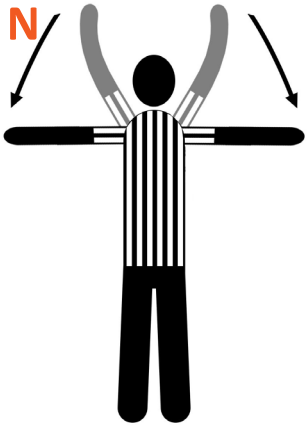


BLOCKING

M

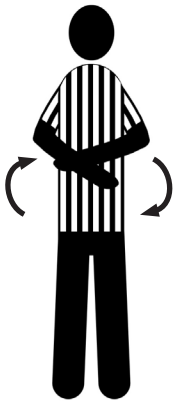
SHOT CLOCK
VIOLATION

N



LOOSE BALL FOUL
(EXTENDS ARMS TO SHOULDERS)

O



TRAVELING
(ROTATING WRISTS)

P

PUSHING