Running the Code Obstacle Course

Kinesthetic Learning for Technology

Learning technology is fun!

It can be fun on the playground without any technology as well! Let's explore how students can perform athletic activities to learn core technological concepts.

Overview of creating an obstacle course teaching technology concepts:

- Code iteration
- If/Else statements
- Conditionals
- Switch/Case statements
- Variables
- For loops

Additional optional concepts include:

- Cartesian plotting to display pixels
- Binary, bits and bytes
- Nested If/Else statements and loops
- Semantics- curly brackets & parentheses
- Code Libraries
- Nested conditionals

Objective:
Establish the above listed concepts in the minds of students through kinesthetic hands-on activities. As an added bonus tire your students out a little before screen time.

Major Common Core Standards: (For more Common Core Standards see last page.)
- A-REI Understand solving equations as a process of reasoning and explain the reasoning (1)
- A-REI Solve equations and inequalities in one variable (3, 4)
- F-IF Understand the concept of a function and use function notation (1, 2, 3)
- F-IF Interpret functions that arise in applications in terms of the context (4, 5, 6)
- F-BF Build a function that models a relationship between two quantities(1, 2)

Obstacle Course Stations Overview

The obstacle course is made up of a number of different stations, each one designed to teach a specific concept. Once you are comfortable with Running the Code you should feel free to redesign the obstacle course yourself to concentrate on whichever concepts you feel are most important. You could also design an obstacle course to focus on one particular concept, creating many different versions of an if statement station, for example. On the next page you will find descriptions of each station type and key aspects of the stations. Stations are covered more in depth on the following pages. The image below is an example course.

Recurring station features:

Variables: As students run the obstacle course they will pick up ping pong balls with numbers (or characters) written on them. These ping pong balls are the students' variables. They will use these variables at most of the stations they encounter in the obstacle course.

Conditionals: Most stations will have a conditional. A conditional is a question (or equation) that is asked in the code which determines the action of the computer or robot. In Running the Code the conditionals are written on a whiteboard or piece of paper that is easily visible at the beginning of the station. When students first approach a station they will have to solve the conditional for either a true or false outcome using the value of their variable.

Loops: Some stations are a subroutine, meaning the student will enter a smaller obstacle course inside of the larger obstacle course. The student will have to complete this subroutine obstacle course a certain number of times. Subroutines include For Loops and While Loops.

a. Each time through students pick up a variable in the form of a ping pong ball

b. If statement splits path of obstacle course

c. Switch/case statement has students throw ping pong ball at specific cup or goal

d. For loop has students loop go around loop multiple times

e. Optional code inside of for loop repeats (in this case a Lamps activity to teach binary and bytes)

f. While loop has students continue an activity until they reach a certain time, goal or other condition

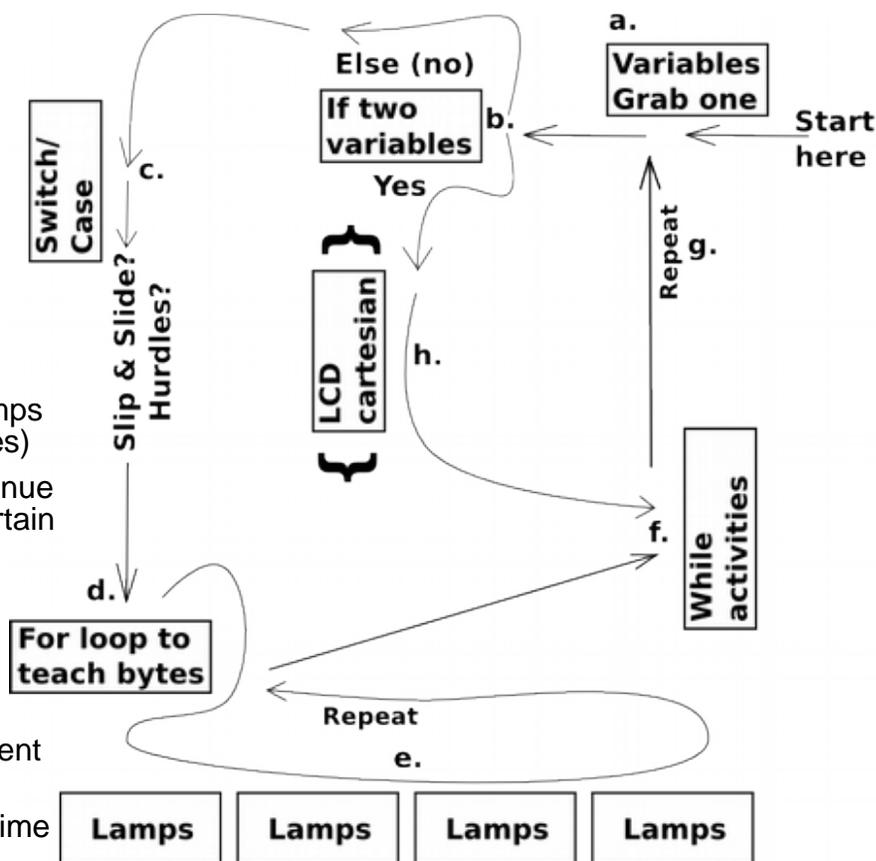**g.** Students reach the end of the obstacle course and repeat

a. Students grab an additional variable

h. Students plot a pixel dependent on their two variables

f. While activities for a second time

g. Repeat as wanted
(real main loops repeat forever or until the computer turns off)

Variables

We'll use ping pong balls with numbers written on them as our integer variables in Running the Code. Variables are pieces of code used to store information. Variables can only store one type of information. An integer variable can only contain numbers. A character variable can only contain characters.

A variable has a name and a value. When a variable is first created, or instantiated, it can be empty or it can have a value. With our ping pong examples we are assigning a value when we instantiate our variables.

Limiting the Variables in your first obstacle course:

We suggest you start off with one integer variable and work your way up to multiple variables or variables of different types.

When you're using just one variable at the beginning you can always refer to the variable as "var" (or change it up to get students used to different variable names). We also suggest that you limit your variables to values of 0 – 5 at the beginning so your obstacle course doesn't get too complex.

Using Variables in the obstacle course:

At the beginning of the obstacle course students will grab a ping pong ball without looking at it. The number written on the ping pong ball will be the variable value that the students use to resolve all the conditionals in the obstacle course unless otherwise instructed. Once students have completed one lap in the obstacle course they will reassign their variable value by picking out a new ping pong ball.

When students reach conditionals they will need to look at their variable and mentally insert the value of the variable into the conditional in order to solve for a true or false. So really, variables are the things that dictate what students do in your obstacle course. You can use this to your advantage and make more interesting obstacles more or less likely.

Reassigning Variables:

When students reach the beginning of the obstacle course for a second time they will reassign the value of their variable. They do this by exchanging their current ping pong ball for another one. If you are designing your own obstacle course you should feel free to reassign variables anywhere in the obstacle course, not just at the beginning. This way you can ensure a certain value for all students at any given point in the obstacle course.

A note about variable names:

Variable naming convention calls for variables to always start with a lower case letter and contain no spaces. Those are the only rules to variable names.

This does mean that if your kids want to name their variables something along the lines of peanutButterIsTastyWithBananasAndChocolate, they can. Get a laugh out of it, then point out that they're probably not going to want to write such a long name over and over again.

Conditionals in the Obstacle Course

The Code and Conditionals in your obstacle course will need to be printed in large text and displayed somewhere that students can easily see them when they approach a station. That way students can find the Conditionals easily so they can solve them and continue having fun through out the obstacle course.

In the obstacle course students will have to use the variables they pick up to solve the conditionals at each code related station. The course is laid out in such a way that if the answer to the conditional is true students go one way, or perform one task and if the answer to the conditional is false students go the other way or perform another task.
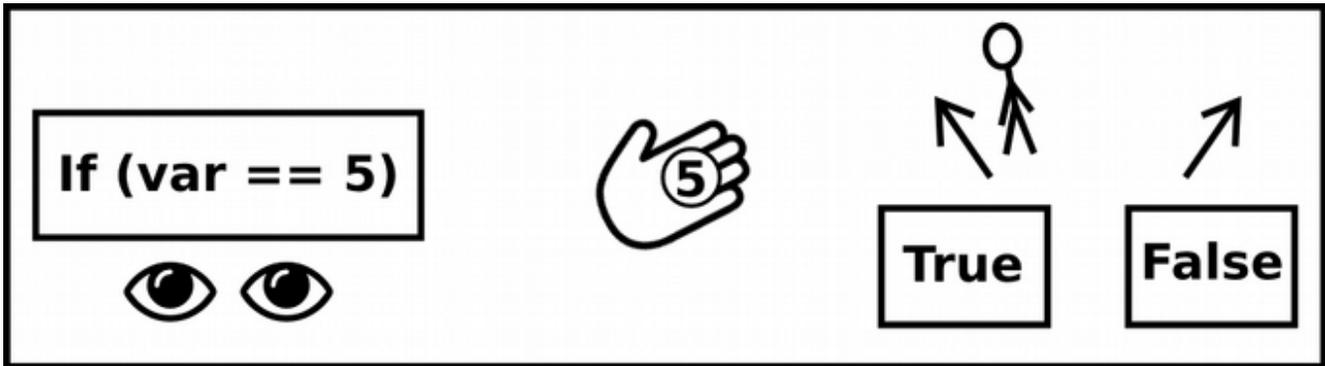
Using More than One Conditional in a Single Station

You can also combine Conditionals so that you are asking more than one question at a time. This can get complicated so we suggest that for your first couple obstacle courses you stay away from combining conditionals.

Symbols for combining Conditionals:

&& -Use the AND symbol to return a TRUE value only if both Conditionals are TRUE.

|| -Use the OR symbol to return a TRUE value if either of the Conditionals are TRUE.

Insert the symbol between the Conditionals you want to combine-

*   if (var <= 5 **&&** var >= 2) { put a block on the tower of blocks }

    Students will put blocks on the tower once if the value of "var" is 2, 3, 4 or 5.

*   while (var == 3 **||** var == 1) { put a block on the tower of blocks }

    As long as "var" is equal to 3 or 1 students will keep putting blocks on the tower.

Logic Table for Combined Conditionals

P	Q	P && Q	P || Q
TRUE	FALSE	FALSE	TRUE
FALSE	TRUE	FALSE	TRUE
TRUE	TRUE	TRUE	TRUE
FALSE	FALSE	FALSE	FALSE

Using Code in the Obstacle Course

You can use If statements in one of two ways in your obstacle course:
- To split the route the students will take into two different paths.

- To make the students either complete an action or not complete an action. These can be non technical actions such as building a block tower or blowing bubbles. The actions can also technical such as reassigning a variable value.

You can use If/Else statements in one of two ways in your obstacle course:
- To split the route the students will take into two different paths.

- To make the students complete one of two action or not complete an action. These can be non technical actions such as building a block tower and blowing bubbles. The actions can also technical such as reassigning a variable value in one of two ways.

You can use While loops in one of two ways in your obstacle course:
- To split the route the students will take into two different paths.

- To make the students do an activity a number of times. These can be non technical actions such as building a block tower and blowing bubbles. The actions can also technical such as reassigning a variable value in one of two ways. Make sure that the While loop eventually changes a variable that is used in the conditional, otherwise you could wind up with students caught inside a While loop... unless that's your intention!

- To make students continue an activity until a threshold or event is reached. For example you might have them put blocks on a tower While the tower hasn't fallen yet.

- To get students to wait- for the station ahead of them to clear, for example.

Using Switch/Case statements in an obstacle course:
- To determine one path among many that a student will take to continue the course
- To control which one item a student picks up out of many different options
- To choose which animal a student pretends to be during a quick game of charades
- To control the X or Y position of a cartesian positioning activity with a checkers board
- To raise or lower the difficulty of task depending on the age of the student

- switch var { case 1: go left, break; case 2: go right, break;
            case 3: go straight, break; default: go back a station, break; }

Using For loops in an obstacle course:
- For loops are a great for laps- whether they are on foot, bicycle, tricycle, piggy backed, in three legged race form or even in a swimming pool!
- They are also great for repetitive actions such as stringing beads on a necklace.
- If you want to make your kids run your obstacle course a certain number of times think about turning the entire obstacle into a For loop. That way whatever number you put into the conditional will be the number of times they run the course.

Preparation for Running the Code

There is a lot of preparation that goes into Running the Code, but it's an experience that will stick with your students for the rest of their lives and is a fun way to celebrate the end of a semester or the achievement of a difficult goal. Let's talk about the various things that go into preparing for a Running the Code obstacle course.

Classroom preparation:

- You need to introduce each of the major concepts that you intend to feature in your obstacle course beforehand in your classroom. These major concepts are listed with each obstacle course plans as well as detailed in this document. You don't have to cover the concepts in depth, a pseudocode discussion is enough to get started.

- You may want to include your students in the planning of the obstacle course, this way they can get a better understanding of how computers make decisions. Including students will also increase their cultural buy-in and build anticipation, as well as fostering teamwork and communication. Students that help plan will also make the running of the obstacle course go a lot smoother.

Material preparation:

- Create a layout complete with scale. Make it complete so people who help you can understand what the obstacle course will eventually look like.

- Write variable values on ping pong balls.

- Write conditionals in large visible text on pieces of paper or other forms of signage.

- If the conditionals or obstacle course stations will not allow you to put directions for actions or paths that students should take after solving the conditional you will need to create additional signage to direct students after they have their conditional solution.

- Collect the various physical parts of your obstacle course. You should bear in mind what you can add or subtract from the obstacle course plan supplied to work with your budget. We also don't have know if you have access to a playground that you can include in your obstacle course planning or when you intend to run your obstacle course. Slip and slides obviously won't work well in the middle of winter, so you may need to substitute a balance beam or a set of hurdles.

- Setting up an obstacle course is a big undertaking, you may want to run the obstacle course for a couple of days. If your obstacle course is outside make sure to bring in any of the delicate obstacle stations at the end of the day.

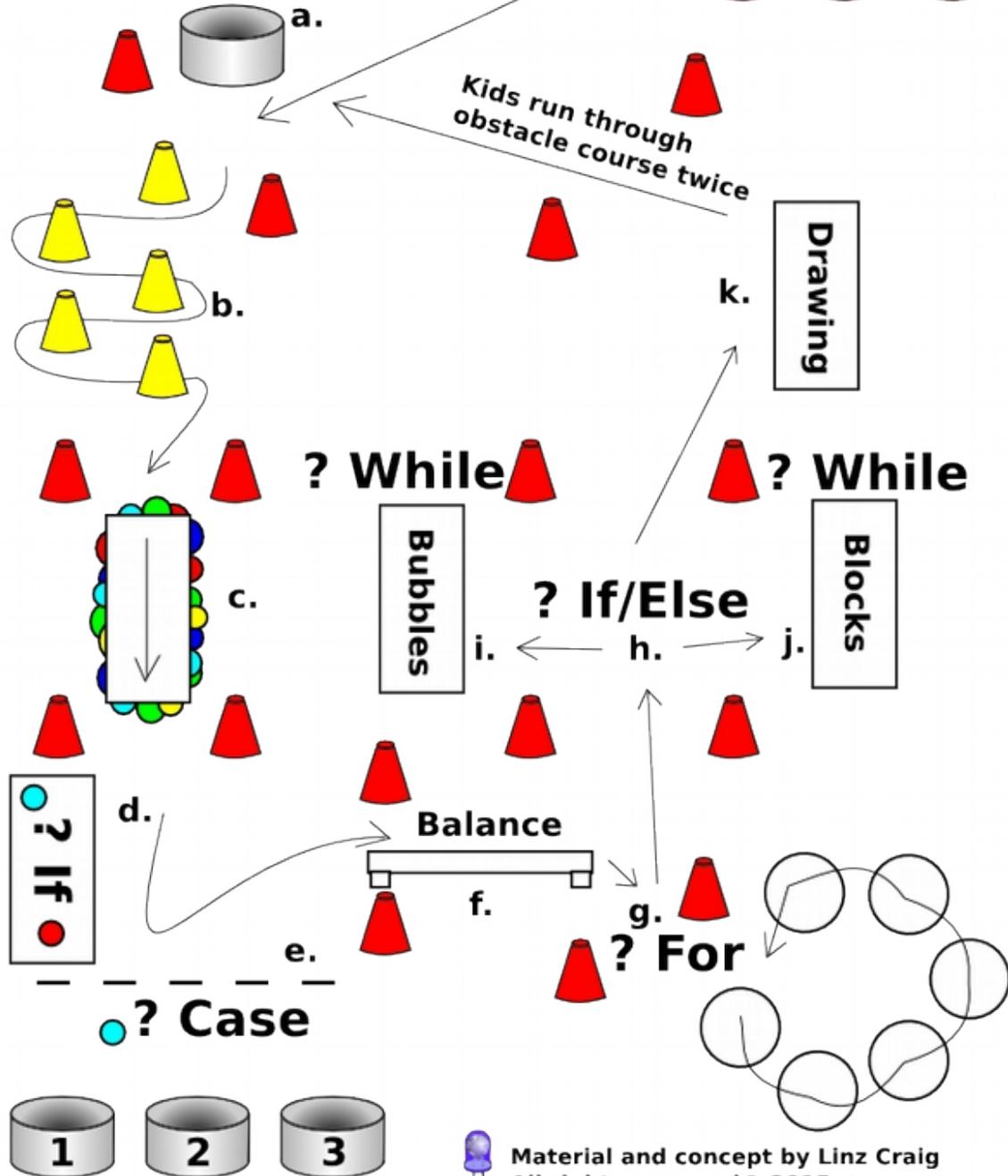## Running the Code (actually using the obstacle course):

- You're going to need to have a helper at each of the major conditional statements to explain the conditionals and obstacle stations to confused students. Luckily there are a couple ways to make this easier on yourself. (1st tip- Include students in planning.)

- Partner with your gym teacher, art teacher, parent volunteers or even your previous class' student leaders who have experienced running the obstacle course. They may not be technologists, but often they're used to explaining physical activities to kids. They are great allies in your efforts at kinesthetic technology education!

- Design your course so that the first student to complete an obstacle station (with your help) stays at the obstacle course to explain it to the next student. Just make sure that the student understands they have to wait until the second student has successfully completed the station before they can move on.

A Simple Plan to Run the Code

a. variable assignment, kids grab a ping pong ball with a number on it
b. kids run around cones   c. kids crawl under table with balloons
d. kids grab a frisbee, baseball or basketball depending on their variable value (if statement)   e. kids throw their ball/frisbee at a bucket depending on their variable value (switch/case)   f. kids walk across a balance beam   g. kids hop from hula hoop to hula hoop, completing the circle 3 times (for statement)   h. kids either build block towers or blow bubble (if/else)   i & j. kids build block tower until it reaches a certain height or blow at least 3 bubbles (while)
k. kids sign their name or draw on the poster

Explanation & Start

Kids run through obstacle course twice

a.

b.

c.

d.

e. **? Case**

1   **2**   **3**

? If

Balance f.

? While (Bubbles) i. ← **? If/Else** h. → j. (Blocks) **? While**

Drawing k.

g. **? For**

Revising A Simple Plan to Run the Code

When we first started doing obstacle courses the first plans were the ones you see in this document. But the actual obstacle courses usually don't wind up looking like these plans. The reasons will be many- You can't get the materials for certain activities, you have specific space constraints, you really need to feature your twenty foot tall purple bunny rabbit statue, you want to add some particular activities to teach one of the concepts your students are currently having trouble understanding or you decided to have your students help you design the course.

The Layout- When Everything Changes

The image on the previous page was the initial plan for one of the first obstacle courses. Then I had to design a plan for the actual space with a scale and real life measurements. (Like space for people to throw frisbees and other athletic gear.) After doing a little thinking and rearranging, as well as tracking down free resources and checking to see what materials we had to work with my plan wound up looking a little different. We also needed to make sure there was plenty of space for throwing frisbees and balls. Here's the final version of the layout-

100 ft
] = 5 ft

Symbol	Description
D	Directions table
	Balloons obstacle table
	Other obstacle table
□	Tub to hold variables
▭	Sign or conditional
O	Hula hoop
▲ ▲	Cones to direct kids
▴	Cones kids run around
●	Target for frisbees etc
0	Soda bottle bowling

a. integer variable
b. cone run
c. soda bottle bowling
c2. color variable
d. balloon table   e. pick ball/frisbee   f. throw at target f2. boolean variable g. cone run
h. if statement
i. blocks/bubbles
j. balance beam
k. hula for loop
l. more ballon table
m. large blocks
n. poster drawing

Additional IF so kids only draw/ use big blocks once

It's important to bear in mind things like space for kids to throw things and how to bunch the more difficult coding concepts so that a teacher or helper can keep an eye on more than one obstacle at a time. You'll also want a clear map key so it's easy for people to help you.

The Parts of the Obstacle Course

Variables:
These are the ping pong balls you will be picking up. There are three types:

- Integers–   These ping pong balls will have numbers on them.
- Characters–   These ping pong balls will have letters on them.
- Booleans–   These ping pong balls will have a T or a F on them.

Each time you come to a container of variables you should grab one if you don't have a variable of that type, or exchange your ping pong ball for another one if you do.

Conditionals:
These are like clues that you have to figure out to get through the obstacle course. They ask a question and then tell you what to do if the answer is "yes." The conditionals are written on signs around the obstacle course. There are four different kinds of signs that will use conditionals or variables:

- If statements
- While statements
- Switch/Case statements
- For loops

Activities:
These are the actions you will take after solving a piece of code with a conditional or variable. They could be as simple as quacking like a duck, running to a new part of the obstacle course or they might be another piece of code with another conditional.

Looping the course:
You may find yourself running around the obstacle course a couple of times. That's ok. That's exactly what computers do when they run code- they loop over and over and over.

Simplifying the Obstacle Course

If you're not ready to jump into an obstacle course that has all of the programming concepts we outlined you can always simplify your obstacle course. Here are some ways you can do that:

Introducing One Concept at a Time

Students who are being introduced to programming for the first time can sometimes be overwhelmed by a more advanced obstacle course. In order to minimize confusion and bolster your students' confidence you can start with a very simple obstacle course that only contains If statements. Only after students feel comfortable with If statements would you move on and introduce an If/Else statement.

You can repeat this type of introduction until you have built an Obstacle Course that contains all of the programming concepts outlined previously, or you can create a new obstacle course for each concept that you want to introduce.

Limiting the Way the Conditionals Effect the Path or Activity of the Student

As previously outlined there are a bunch of different ways you can use conditionals to effect the student's obstacle course experience. One way to simplify their experience is to limit the ways the conditionals effect their path through the obstacle course or how the conditional effects the course activities.

For example, you may want to limit your conditionals so that they only direct the students through the obstacle course. You can find an example of this in the Simply A-Mazing obstacle course layout.

Alternatively you may want to focus on only controlling aspects of the activities instead of the route through the obstacle course. You can do this by making all of your conditionals focus on things such as the numbers of bubbles a participant must blow or the height a block of towers must achieve before he or she can continue on to the next obstacle course activity.

Make it More (or Less) Fun

Another way to alter the difficulty of your obstacle course is to create less stations that use programming and supplement them with activities that are just plain old fun instead of information oriented. While this will cut down on the teaching aspect of your obstacle course, it's important to strike a balance between programming and fun to make sure that the obstacle course is a treat and a draw instead of a chore.