Kids-that-Code

1. Introduction
2. Osmo Coding Kit for iPad
3. Kodable
4. Hour of Code
5. Lego WeDo 2.0
6. Drone – Star Wars BB8
7. Drone – Parrot Jumping Night Drone
8. Robotics 1 – Lego Mindstorms EV3
9. Robotics 2 – Lego Mindstorms EV3
10. Molecules
11. Keeping Children Safe in Car Crashes
12. Books

Note: Click on the arrow in the figures. Some of the video links are still active.

1. Introduction

I spent my entire career as a code developer. I wrote engineering software. To be an expert in writing code, you first need an education in the discipline you are writing the software for plus training in numerical methods. You don't simply go to a 4 week "how to write computer code" class. However, you can start teaching the foundation in logical thinking required for code creation to 5 year olds. For a thought exercise, write a sequence of instructions (i.e., code) on how to brush your teeth and give it to someone else to execute. An adult will start the code off by the statement "put toothpaste on the toothbrush". This will lead to a computational fault unless you started execution in the bathroom with the toothbrush and toothpaste clearly visible. A young child will say, "go to the bathroom" as the first execution statement.

This is a collection of informal Blog posts that I wrote during the summer of 2018 as I was evaluating code training apps and materials for my 6 year old granddaughter, Rheannon. I found nothing perfect. I wish I could merge some of the apps together to create the perfect learning environment. Here is my "best" list to start teaching code at the 5-6 year old level in the order they should be presented. This can of course be used for older children. They will just advance through each stage more quickly, or just jump to Swift Playgrounds and Lego Mindstorms.

(1) Osmo Coding Kit for iPad, which is available through the Apple store. The best part is you write code by assembling tiles in front of the iPad. The child has tactile feedback. Instead of building a Lego structure, the child uses tiles to build code to manipulate an environment. Each tile has an icon (e.g., move, jump, turn) so you don't need reading skills. The iPad camera sees the tiles and moves a figure on the iPad screen to reach an objective after you press the "start" tile. There are tiles for "loop" and "if-then-else" constructs.

(2) iPad app Lightbot Jr for iPad. The Osmo tiles are replaced by icons on the iPad screen. You drag and drop icons to assemble them into a code that moves a robot to an objective. No reading skills are needed. A shortcoming is that there are not enough practice exercises. My main reason for liking this app is that it is a stepping stone from the real world of Osmo physical tiles to the virtual world of writing code only on a computer.

(3) iPad app Rocket Cupcake. This is similar to Lightbot but has many exercises. The icons are dragged and dropped into a code window. The color of each icon changes as the code is executed in sequence. A red color indicates a bug and the code must be changed. A shortcoming is that there are too many popup windows requiring reading skills to navigate through them in order to get back to the programming environment. This is frustrating to a pre-reading child who will need help but should present no problem for older children. Rheannon found a workaround to the popups – she keeps clicking the return arrow until the programming window becomes dominant again.

(4) Tynker for iPad. Reading skills are now necessary. The icons have been replaced by words. Instead of an icon showing a figure jumping up, we now have a tile with the word "jump". The tiles are dragged and dropped into a code window. Again, we are moving a figure to reach an objective. A nice feature is that you can display the Swift code next to the assembled tiles. Swift is a real programming language.

(5) Swift Playgrounds for iPad. This is an Apple training app that uses the real programming language Swift. Swift is used to write real world application software. You write increasingly more complex Swift code as you progress through the training material: (1) fundamentals of swift, (2) beyond the basics, (3) explore the universe.

We moved from the real world using the Osmo tactile tiles to write code that manipulates a figure in the virtual world of a computer screen, then wrote code in the virtual world of the computer (Lightbot, Rocket Cupcake, Tynker, Swift) to move a figure in virtual space, and now we should move back to the real world (i.e., write code on the computer to manipulate an object in the real world. Swift Playgrounds recently expanded its software to allow coding of Bluetooth-enabled robots. This includes Lego WeDo 2.0, which is appropriate for younger children (with parental help), Lego Mindstorms EV3, Sphero SPRK+, and Parrot drones.

Watch out for the BUGS!

Computational machines started off by connecting devices with patch cords. Spiders liked to live in the sockets. Then, the plug did not make a good connection with the socket and the term Bug was coined. A bit later in time computational machines used orange glowing vacuum tubes. You guessed it, these attracted all kinds of flying insects which would cause a short circuit.

2. Coding with Osmo

by: https://www.playosmo.com/en/coding/

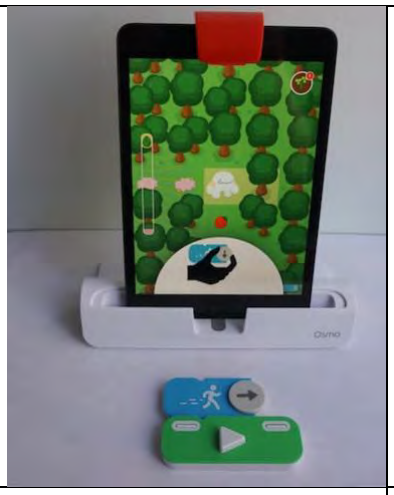
The phrase “concrete to abstract” is heard frequently in Montessori education of young children. The phrase can be rather ambiguous, especially to parents who may have little working knowledge of Montessori practices. I have applied this Montessori philosophy in teaching my granddaughter how to code.

Concrete	Representational	Abstract
A problem is introduced with manipulatives; students explore solving the problem using the manipulatives in writing code.	The manipulatives are replaced with representational pictures; students demonstrate writing code at a pictorial level.	The representational pictures are replaced with a symbolic language; (numbers, operation signs, etc.); students demonstrate writing code using a programming language.

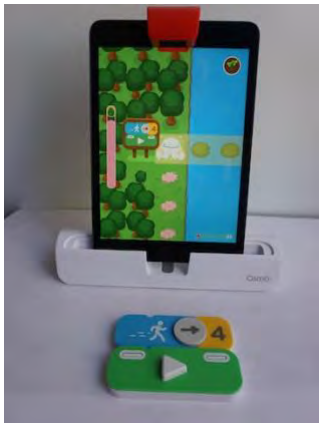
“Osmo Coding Awbie” is for younger children in the Concrete learning stage. My granddaughter, Rheannon, started using this at age 5. She was able to run the application on her own after some initial guidance.

"Osmo Coding Awbie" for iPad, is available through the Playosmo store (https://www.playosmo.com/en/coding/) and the Apple store. Make sure you are purchasing "Coding Awbie" because several other apps are also available making things a bit confusing. You will need an iPad, an Osmo base station ($30), and a Coding Awbie kit ($50).

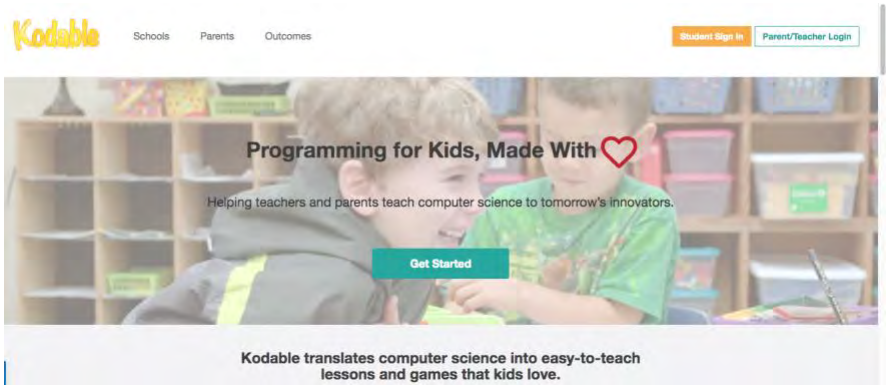
For young children, Maria Montessori believed that "what the hand does, the mind remembers". The OSMO programming tiles can be felt and manipulated so that the hand is always involved in the learning process. You write code by assembling tiles in front of the iPad. The child has tactile feedback. Instead of building a Lego structure, the child uses tiles to build code to manipulate an environment. Each tile has an icon (e.g., walk, jump, loop) so reading skills are not required. The arrow on the tile is rotated to indicate direction. Numbers are attached to the tile to indicate how many steps or jumps to take. The iPad camera sees the tiles and moves a figure on the iPad screen following the code from top to bottom in sequence after you press the "start" tile. Early levels have animated graphics with a hand going though the motions assembling tiles.

		
Walk right 1 square	Rotate the arrow direction and walk down 1 square	Add the number 3 and walk right 3 squares

The on-screen animated help disappears as the child advances through the lessons. They reappear when a new concept begins (e.g., loops). Early lessons include a "Sign Post" offering a suggestion on what tiles to use. Screen help begins to disappear as the child advances through the training exercises.

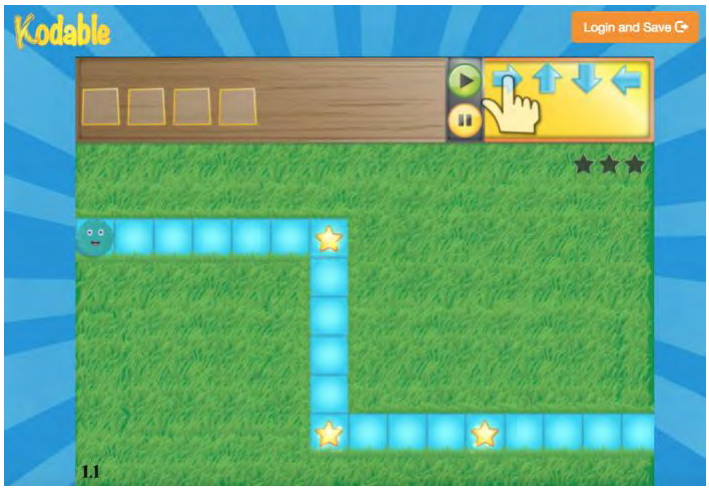


3. Kodable https://www.kodable.com



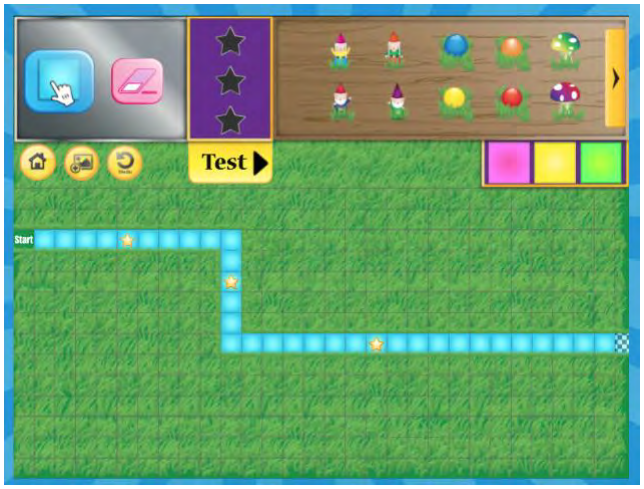
Kodable (https://www.kodable.com) can be run in a web browser or as an iPad app. You can create a free account with limited commands and training exercises (https://www.kodable.com/register) or pay a one time fee of $29 to unlock advanced commands. I suggest trying the free account first to see if it fits your needs. Kodable also offers lesson plans (https://www.kodable.com/hour-of-code#lesson-plans).

A beginner first practices writing sequence code followed by adding conditionals, loops, and then functions. The beginner level is for young pre-reading children. Code is written using icons as shown in the following figure.



The correct sequence of icons will move the fuzz ball through the maze collecting stars. Icons are moved from the symbol table in the upper right to the code blocks in the upper left. Clicking the start button will execute the code. Wrong code (i.e., a bug) will be highlighted in red for fixing before execution will continue. The following figure demonstrates the use of conditionals (i.e., green colored maze square at a path junction point) and loops. There is a bug in the code. Can you find and fix it?

The learning exercises started to become a game for Rheannon. After completing several lessons she was directed to a playground where she could create her own maze and demonstrate her mastery of the commands she just learned. The playground has a “write” tool (the blue button) and an “erase” tool (the pink button) in the upper left.

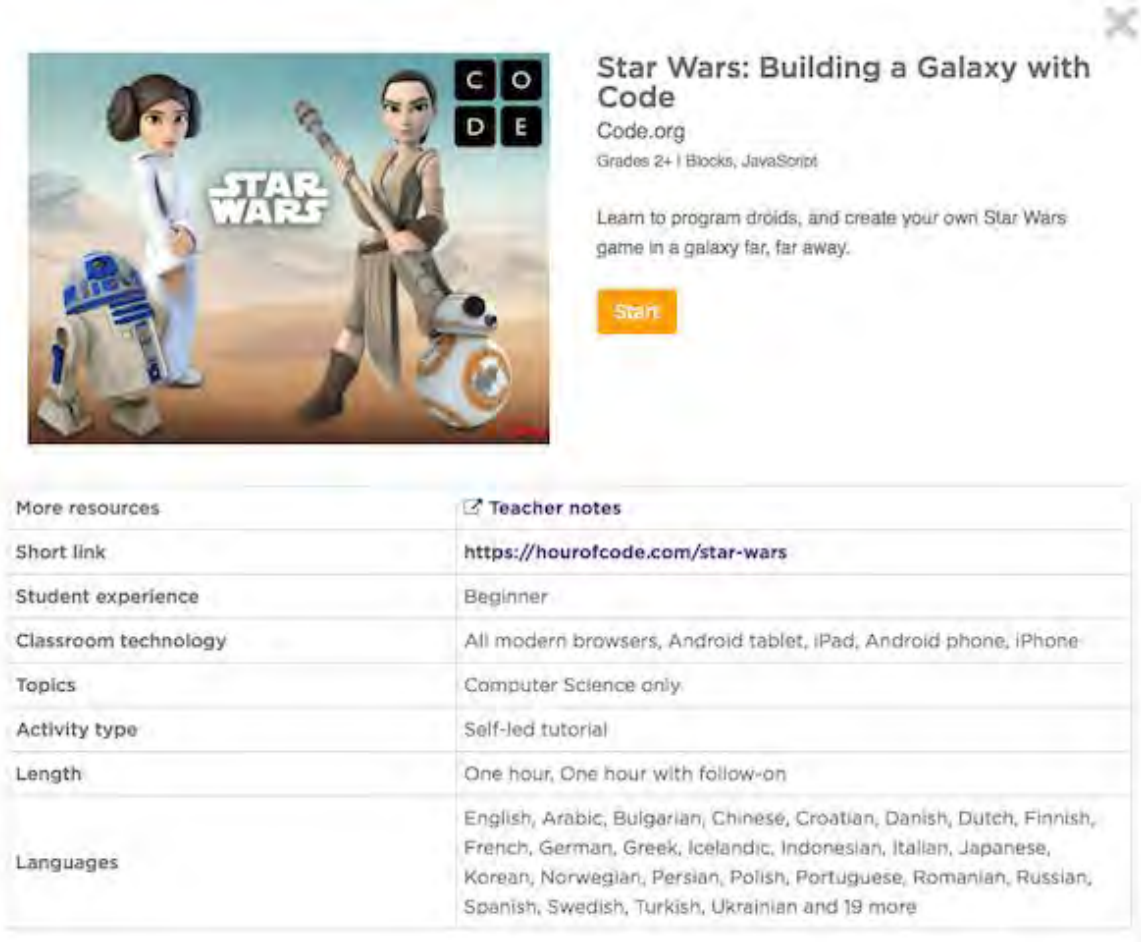


In my opinion, this is the best part of this app and puts it above others that I have tried. Her ability to completely understand the concept of code writing evolved in the playground. She had to use the code syntax she learned in the lessons to create a unique maze. They started out simple as shown in the above figure. We both found this enjoyable as she created increasingly more complex mazes requiring conditionals (i.e., the use of the pink, yellow, and green squares at maze junction points) and loops in an attempt to make it difficult for me to solve. Of course, she had to solve it first to make sure it was a solvable problem and I made purposeful mistakes to see that twinkle in her eyes for one-upping me.

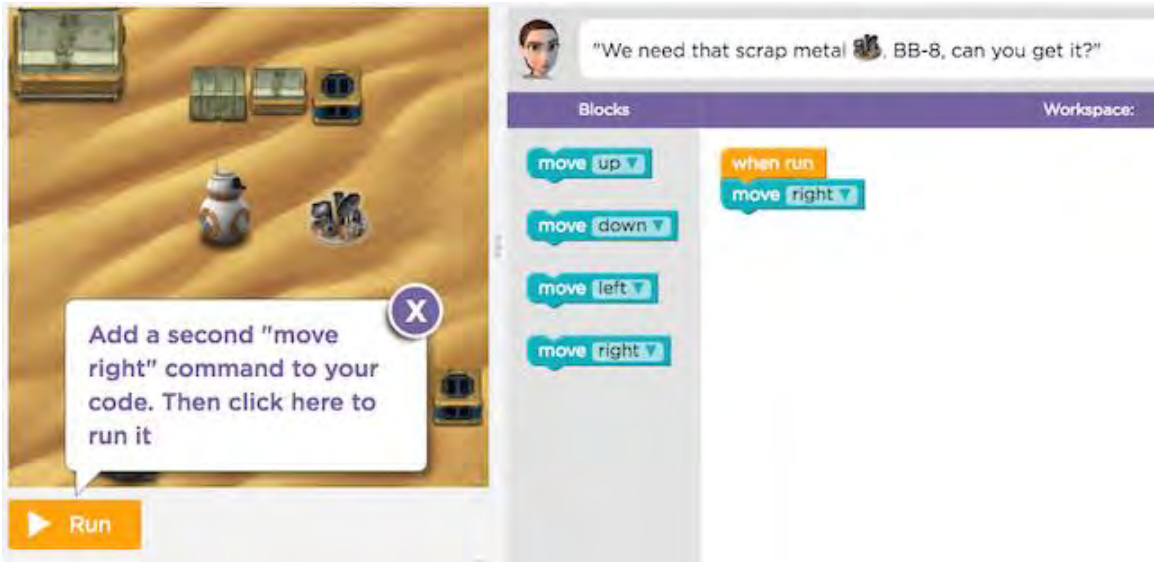
4. Hour of Code

https://hourofcode.com/us/learn

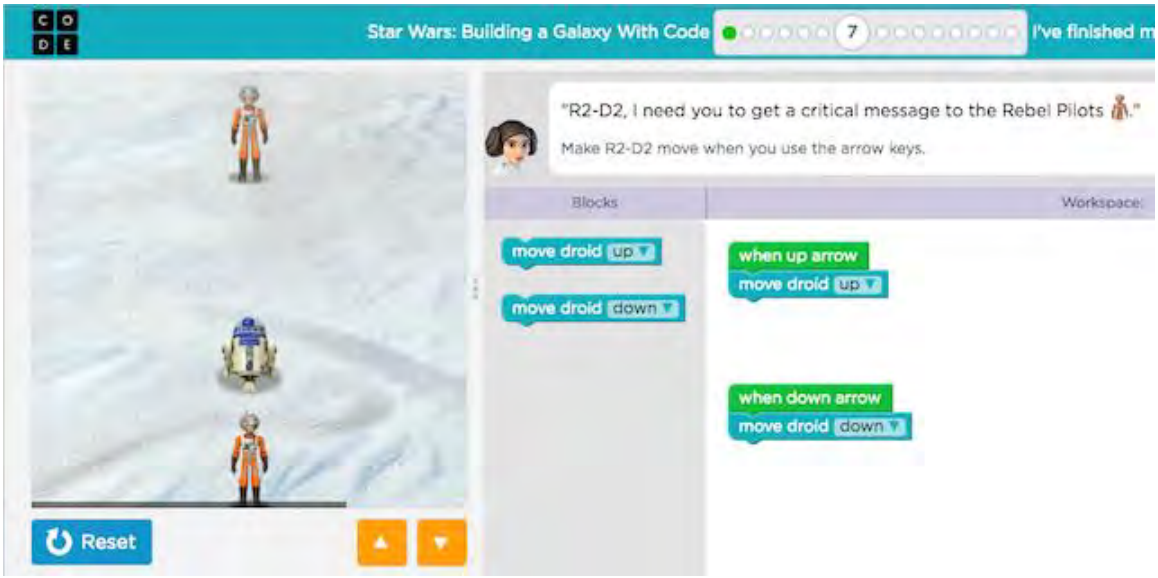
Hour of Code – try a 1 hour coding tutorial designed for all ages in over 45 languages. The 1 hour sessions are designed for classroom teaching. The lessons are divided into age groups (e.g., pre-reader, grades 2-5, grades 6-8, grades 9+) with an additional division between beginners or advanced code writers. Rheannon's Hour of Code class this week at school was "Star Wars: Building a Galaxy with Code". Rheannon is in 1st grade.



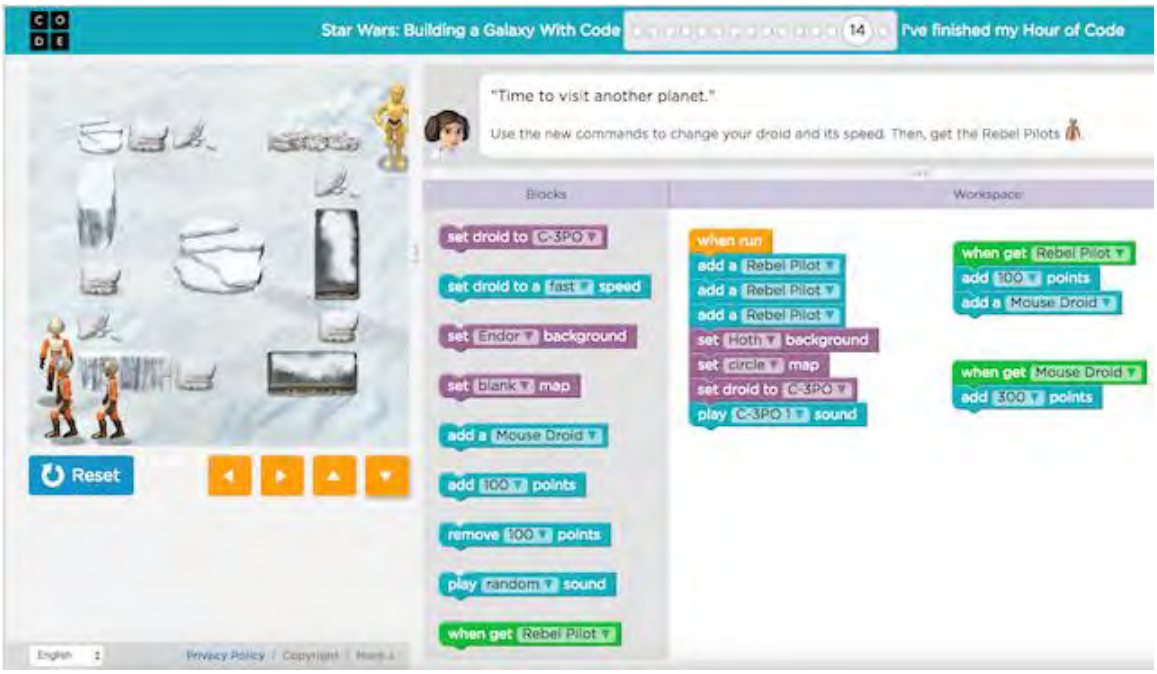
The computer screen is divided into several panes. On the left is the action pane where the computer code runs. The next pane to the right lists the availble code commands. Next is a large pane to write the code by drag-and-droping the commands to form an executable program. Some commands have pull down menus to define attributes.



In addition to writing sequential code, this lesson introduces an "event", such as programming the up and down arrow keys. Once an event is coded, it is saved for future use.



There are 15 exercises or levels to complete. The last few levels involve adding commands to create a game. To win, the player must accumulate points within a defined time limit. There are commands to change the background and sounds with pull down menus.


Star Wars: Building a Galaxy With Code
14
I've finished my Hour of Code
"Time to visit another planet."
Use the new commands to change your droid and its speed. Then, get the Rebel Pilots
Blocks
Workspace
set droid to C-3PO
set droid to a fast speed
set Endor background
set blank map
add a Mouse Droid
add 100 points
remove 100 points
play random sound
when get Rebel Pilot
when run
add a Rebel Pilot
add a Rebel Pilot
add a Rebel Pilot
set Hoth background
set circle map
set droid to C-3PO
play C-3PO 1 sound
when get Rebel Pilot
add 100 points
add a Mouse Droid
when get Mouse Droid
add 300 points
Reset
English
Privacy Policy
Copyright

5. Lego WeDo 2.0

I'm certain that all readers of this Blog have introduced their children to Duplo and LEGO building blocks. What you may not be aware of are the excellent LEGO Educational Kits and teacher resources.

https://education.lego.com/en-us/shop/wedo-2

The coding resources are divided into elementary education age 7+ (WeDo 2.0 kit) and middle school age 10+ (Mindstorms EV3 kit). This review is on WeDo 2.0. You will find the software download button for your favorite device toward the bottom of this link.

https://education.lego.com/en-us/elementary/intro/c/computational-thinking

WeDo 2.0 combines coding with robotics. Rheannon (1st grade) attended a 1 week robotics class last summer. She was teamed with a 3rd grade girl. This class was to be a learning environment but as typical, turned into a race between teams to see who could finish the project first. Both girls brought unique skills that enabled them to compete against the other teams. The construction of the LEGO robot, in this case a race car, required fine motor skills for assembly. The gears and pulleys had to be pushed onto shafts with the right amount of tolerance left between hubs to allow rotation while maintaining drive belt alignment. Reading skills were necessary to read on screen step-by-step instructions. The 3rd grade girl could do the assembly while Rheannon followed the on screen schematics and laid out the required bricks. The next step was writing code to direct the robot to perform a task. They quickly accomplished this. The 3rd grade girl read the instructions and Rheannon, with her previous code writing experience with OSMO and KODABLE, could write the code. Code is written using drag-and-drop icons, just like KODABLE.

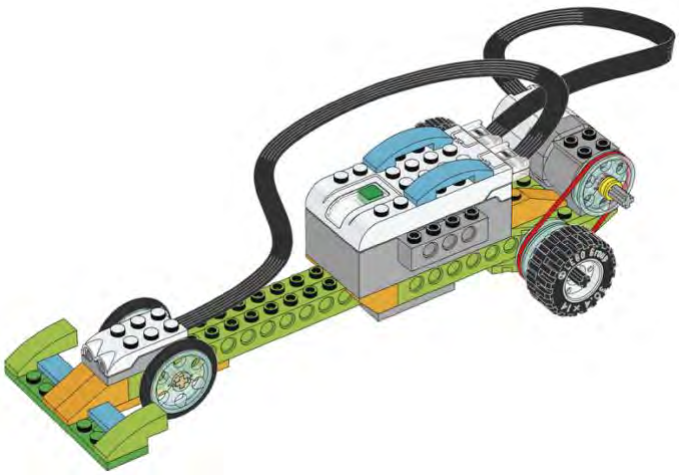
This link provides information on the WeDo 2.0 Core Kit.

https://education.lego.com/en-us/products/lego-education-wedo-2-0-core-set/45300

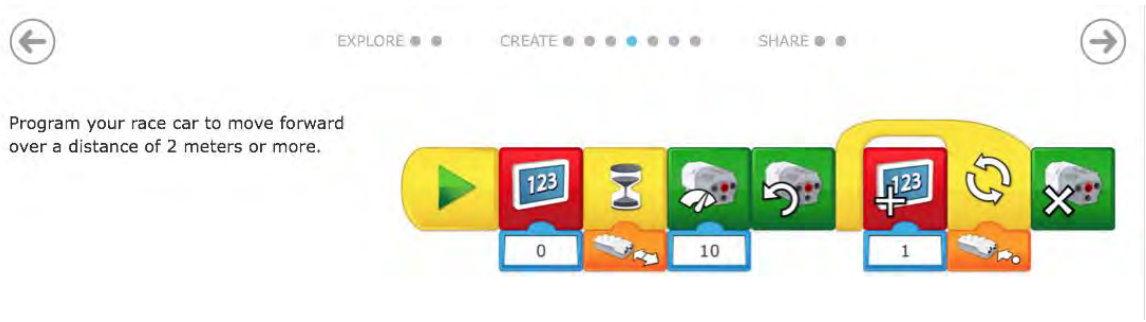
Note that you can download the software and proceed through the lessons and code writing without building a robot. This isn't much fun, but you may want to try this first to see if you like WeDo before buying this $190 kit. In addition to LEGO building elements, the primary robotic elements in this kit are:

- Smarthub → this is a building brick that via Bluetooth connects the sensor elements with the control code on your device (e.g., iPad).
- Motor → a variable speed, variable direction software controlled motor.
- Motion Sensor → detects objects within a range of 15cm.
- Tilt Sensor → reports the direction it is tilted.

Here is a sampling of projects



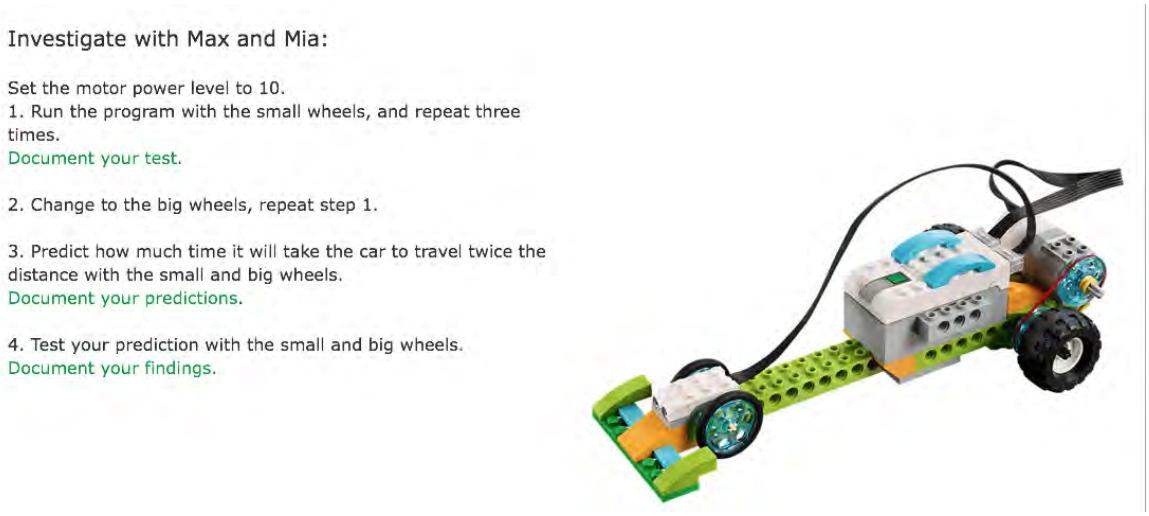
Lesson 2 on Speed involves building a race car using the Smarthub, the speed and directionally controlled motor, and the motion sensor. The dragster control code is:



The control code written on your device (e.g., iPad) consists of these coding blocks from left to right.

1. Start button
2. Initiate timer
3. Wait until motion detector sees movement (e.g., your hand passing in front of the sensor)
4. Turn motor on at power level 10
5. Rotate motor counter clockwise so car moves forward
6. Loop
   - Increment timer
   - Continue loop until sensor detects object (e.g., stop sign)
7. exit loop and turn motor off

There are several class experiments to perform, such as



6. Drone Obstacle Course

Rheannon is very excited. She is old enough to enter a project in the school science fair. Her project is writing code to direct a drone to navigate an obstacle course. She entered the code on the iPad near her hand to direct BB-8 to move around the rocks. Here is a link to a movie showing BB-8's movement.

Rheannon attends a small school with children from toddler thru 8th grade. She is in 1st grade in a multi age primary class, which is for 1st – 3rd grade students. Her project will be demonstrated in her classroom. However, she is excited for the day when she will be in upper elementary (7th - 8th grade) and her science project will be displayed in the auditorium for parents and the whole school to see.

I researched drones and control software appropriate for young children. My first criterion was 2D. Rhea's (her nickname like the heroin in Star Wars) coding experience to date includes forward, back, right and left. I didn't want to confuse things with 3D attributes of up, down, yaw, roll, and pitch. So, flying drones were not considered. Also, flying drones can hit you in the head. Let's stick with something that can only run over your foot if the code has a bug in it. My second criterion was control software that would be understandable to Rhea and build upon her current coding knowledge.

	
R2D2 by LittleBits https://shop.littlebits.cc/pages/starwars	BB-8 by Sphero https://www.sphero.com/starwars/bb8

R2D2 is by LittleBits (https://littlebits.cc/how-it-works). LittleBits is like Snap Circuits (https://elenco.com/brand/snap-circuits/) on steroids. The LittleBits

R2D2 kit comes with the necessary electronic blocks to build a programmable drone. The best part of this was assembling R2D2 following the excellent step-by-step iPad based instructions. The electronic blocks are snapped together and their functionality verified before moving to the next step. The downside, at least for a young child (<12), is the programming language. R2D2 operates using SWIFT code. SWIFT is a programming language developed by Apple for writing apps for devices (e.g., iPhone, iPad, Mac). Apple created the user interface "Swift Playgrounds" to aid in learning SWIFT and writing code to control robots and drones. SWIFT syntax is way beyond Rhea's current code writing capabilities. So, R2D2 is on the shelf collecting dust for a few years.

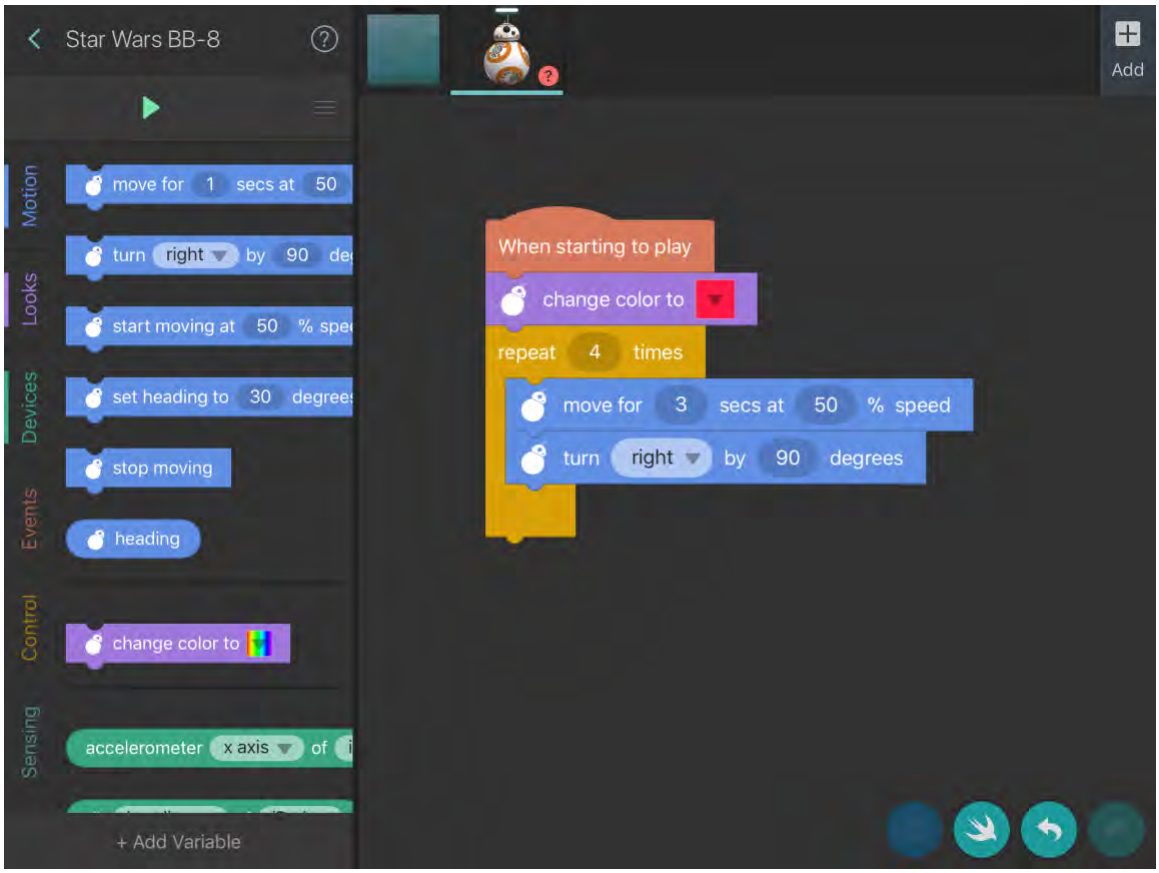
Google (blockly code), MIT (scratch code), and Berkeley (snap code) have been developing open source libraries and visual programming languages to teach young children the logical process behind solving problems. Instead of writing the code from zero, these visual programming languages allow the student to drag and drop boxes that represent a certain action or logical decision. By chaining several of them together they are able to achieve their objective.

The BB-8 droid by Sphero (https://www.sphero.com/starwars/bb8) was perfect. Not only does BB-8 come fully assembled and use code based on the (Google, MIT, Berkeley) design philosophy, it is also cute. You can program BB-8 motion using several iPad applications:

Sphero	https://itunes.apple.com/us/app/sphero-edu/id1017847674?mt=8
Tynker	https://itunes.apple.com/us/app/tynker-coding-for-kids/id805869467?mt=8
Tickle	https://itunes.apple.com/us/app/tickle-app-learn-to-code/id1063639403?mt=8

I found Tickle syntax the best. The Tickle UI can connect to many of the current available drones (e.g., Lego WeDo, Parrot, Sphero, MicroBit). Tickle code to direct BB-8 to travel in a square pattern is shown below. Notice the small bird icon in the lower right of the picture. This converts the block Tickle code into SWIFT code syntax, which is shown in the next picture.

Warning – Communication between the drone and tablet (e.g., iPad) is by Bluetooth Low Energy (BLE) interface. All recent iOS and Android devices use BLE. However, older devices may not. So, before giving your child an old hand-me-down tablet or laptop computer, make sure it's BLE. There are BLE dongles available for old Windows laptops. Also, Parrot drones, which I will discuss in a future article, communicate via WiFi.

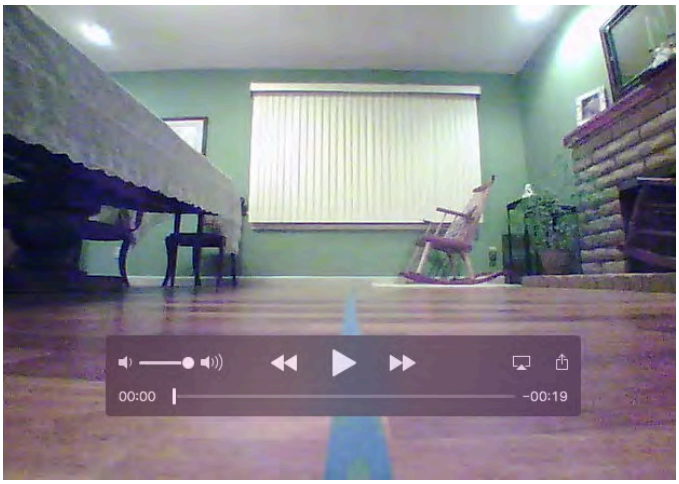


BB8.swift

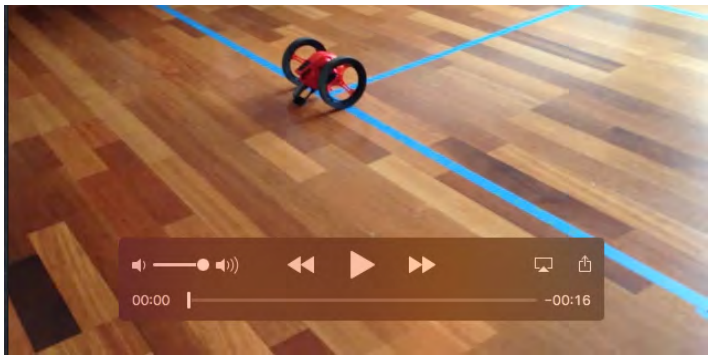
```swift
func onStart() {
    guard let bB8 = getCurrentBB8Device() else {
        return
    }
    bB8.color = 0xff1b47ff
    for index in 0 ..< 4  {
        bB8.move(seconds:3, speed:50)
        bB8.turn(direction:right, degree:90)
    }
}

```

7. Coding the Parrot Jumping Night Drone



Drone Internal Camera

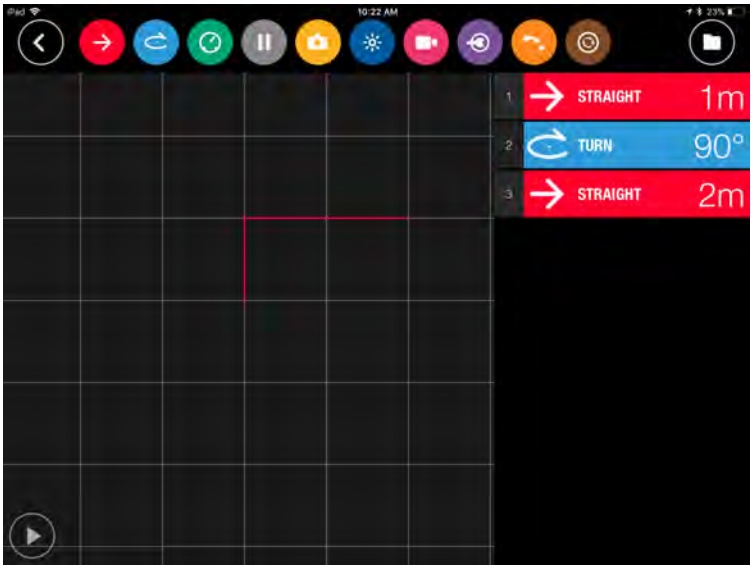


External view of drone

Over the last 2 weeks Rheannon and I have been coding a Parrot Jumping Night Drone. These drones have been discontinued and the remaining stock is being sold at a significant discount on Amazon ($37 - $50). The drone is controlled by code written on an iPad via a WiFi connection. The drone-to-iPad WiFi connection works but with some effort to make the handshake. Sometimes, I would have to turn the drone off, turn iPad WiFi off, and then turn both on to make a connection. Another issue was that the Parrot drone as received would not run until a firmware update was installed. The drone will connect to a computer using a USB cable and mounts as an external drive. This enables installation of the firmware. It took considerable time hunting on the Parrot website to locate the firmware update. It was not obvious where it was. After several Google searches I was finally successful in identifying the link to the firmware.

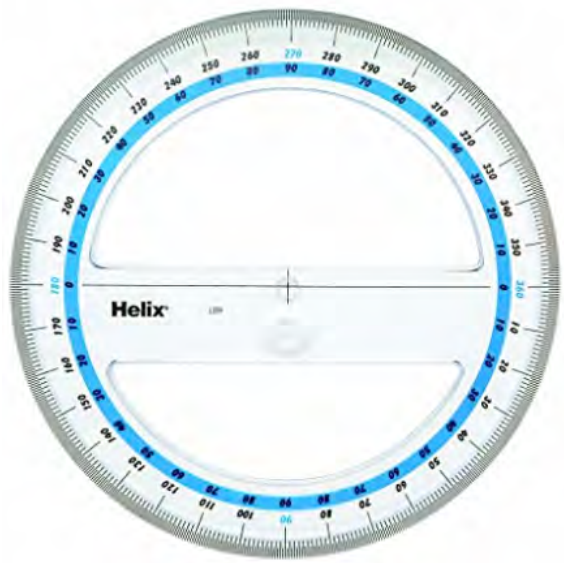
Parrot provides a proprietary app to control and program the drone. It works but is buggy. Many times we would have to delete the code, close the app and start over. So, I'm hesitant in recommending this drone. However, it has turned into a reasonable learning experience for Rheannon.

The Parrot App UI displays 3 windows. The main central window displays a square grid with 1m edge lengths. The top window contains code icons which are dragged-and-dropped into the right hand window to build the code. The



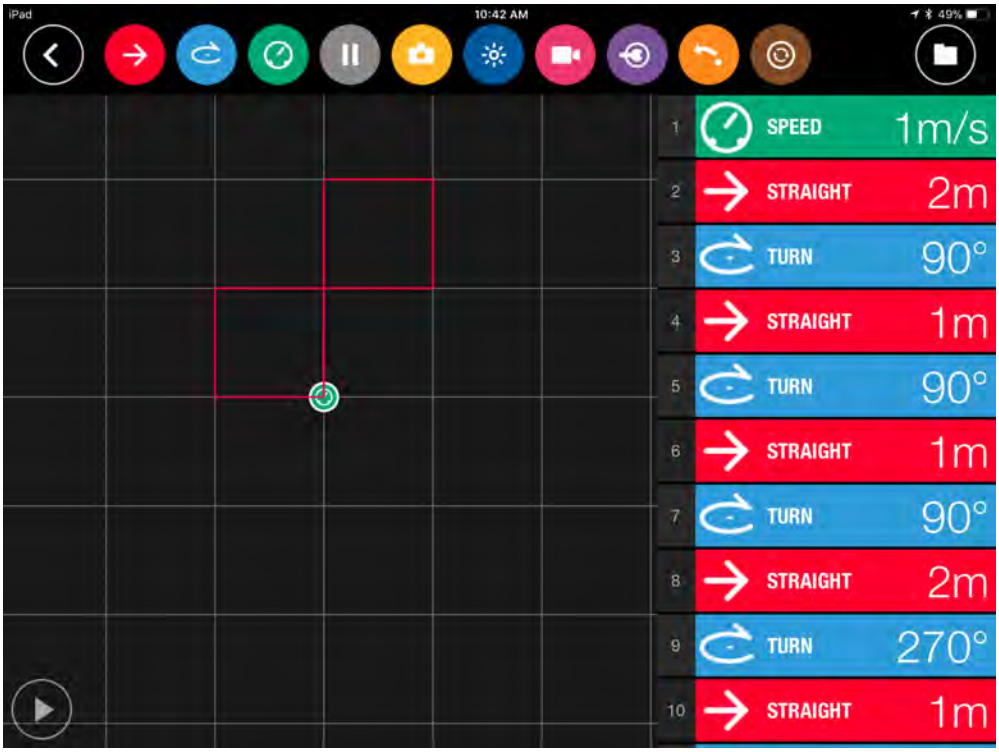
drone motion defined by the code is displayed in red on the grid. This code directs the drone to move forward 1m, turn right, and move forward 2m. The drone motion is very accurate (accepting some slippage due to lack of friction between tires and wood floor) in hitting node points as shown in the above movies.

Because the drone motion was precise, I turned this into a lesson on grids, linear motion and angular motion. The first step was moving the dining room furniture to the wall and laying out a 1m x 1m square grid on the floor with tape. Next was buying a meter stick (remember I'm in the US and we have yard sticks) and a 360 degree protractor. The turn icon only allows clockwise rotation so a 90 degree left turn is defined as a clockwise 270 degree rotation.



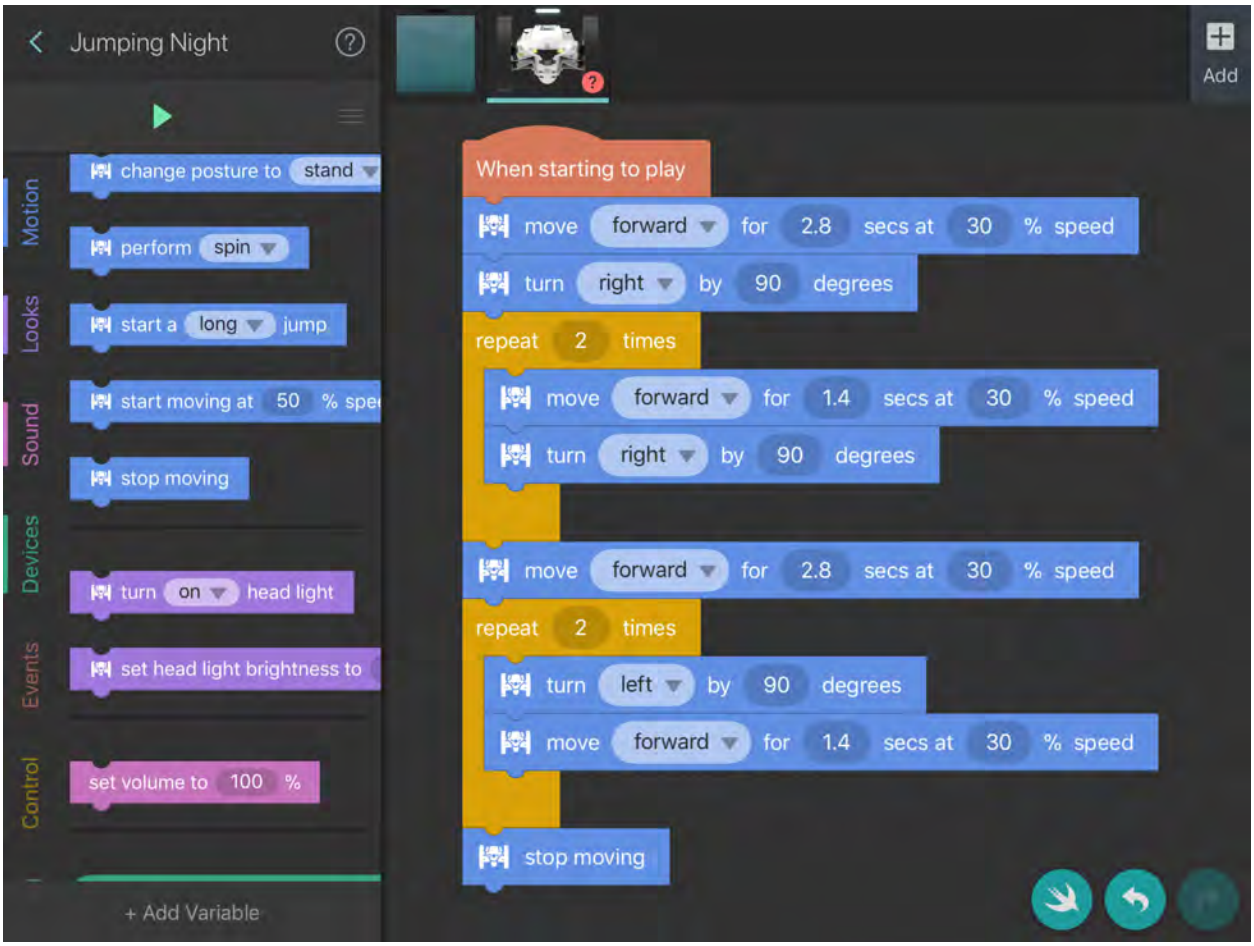
Following is a list of activities we did.

1. The meter stick is graduated in cm and mm. Rhea was impressed that the last number was 1000. This is a big number to her. She also liked that she could count by 10s to get to 100 and then by 100s to get to 1000.
2. We used the meter stick to lay out the grid. I showed her that 500 was half the length. We added marks on the tape to indicate this location.
3. We played human robot. I gave her verbal directions, such as move forward on the grid 1m. Look at the protractor in your hand and turn clockwise 90 degrees. Then it was my turn to be the robot and she gave me voice commands to move on the grid.
4. We wrote code to direct the drone's motion.
5. We experimented with fractional distances (e.g., move forward 0.5m).
6. We experimented with an angular turn of 45 degrees and noted that if the drone traveled 1m it didn't end up at the grid point. I showed her we were traveling on the hypotenuse of a triangle and the distance is more than the grid edge length and can be calculated. Entering a travel distance of 1.4 gets the drone to the grid point.



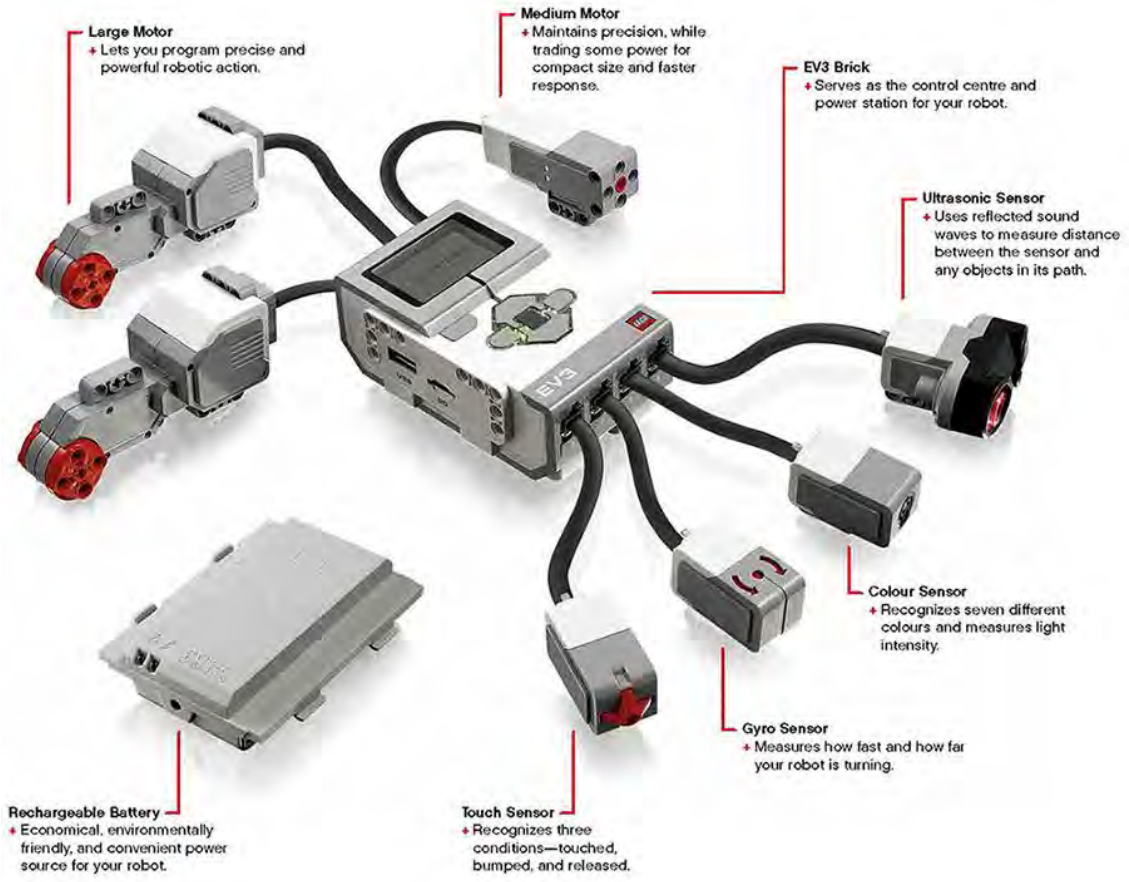
Using the Parrot app, this is the code to direct the drone's motion shown in the movies. Note that commands 11 to 13 to bring the drone back to the starting position are scrolled off the display. Loops are not available. Also, note the command "turn 270" to initiate a left hand turn.

Using the Tickle app, this is the code to direct the drone's motion shown in the movies. Two loops were used to decrease the total number of repetitive code needed by the Parrot app. The "turn" command allowed both a right and left turn as an inline menu option. While the Parrot app allowed specifying a translational distance, the Tickle app required entering a speed and travel time. Several experimental runs were required to zero in on the correct (% speed, time) data pairs to travel 1m. However, once determined, the motion was consistent. The drone traveled 1m at 30% speed for 1.4sec. The drone traveled 2m by doubling the time to 2.8 sec.



8. Lego Mindstorms EV3

Lego Mindstorms EV3 is for you, and if your child finds you playing with it you may have to share. Lego has certainly come a long way from the small multi-colored brick sets. The Lego Mindstorms Education EV3 Core Set costs $412 at the Lego Store (https://www.lego.com/en-us/mindstorms). What makes this exciting is the EV3 Brick and sensors.



The User Guide
https://education.lego.com/en-us/support/mindstorms-ev3/user-guides
available in several languages, lists device specifications showing just how sophisticated this “Toy” is.

EV3 Brick

- LINUX OS
- 300 MHz ARM chip set
- 16MB flash memory, 64 MB RAM memory
- USB port, mini-USB port. SD card port

Large Motor

- 160 rpm max, running torque 20 Ncm, stall torque 40 Ncm
- 1 degree rotational resolution for precise control

Light Sensor

- color sensor – recognizes 7 colors
- reflected light intensity, ambient light intensity

Gyro Sensor

- spin rate 440 rpm
- +/- 3 degree turn resolution

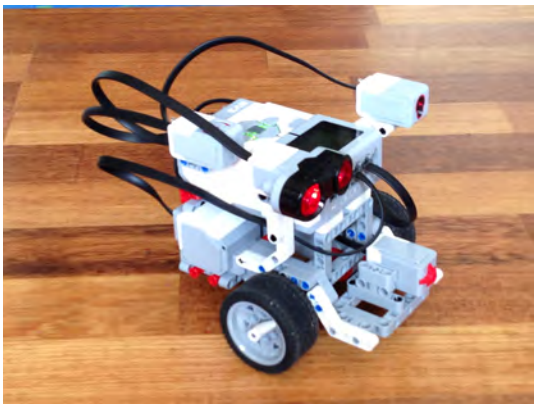
Note that the Brick has LINUX OS installed. You can set up a dual-boot configuration by installing a 2nd OS on a SD card. Several IDEs are available to allow coding the robot in Java, Python, Ruby, Swift, C, C++, SCRATCH, and others. A good starting point if you want to go down this path is
http://www.ev3dev.org/docs/programming-languages/

I spent about 10 hours looking at these alternative options. Unless you have several years of code writing experience in one of these languages, Lego's EV3 Programming IDE is best. My objective for Rheannon (and, kids that code) is to present programming concepts and not get bogged down with a particular language, which will be obsolete when they enter the workforce in 10-20 years. Lego's EV3 IDE runs within LabVIEW on a desktop computer and is also available for tablets. The download link is
https://www.lego.com/en-us/mindstorms/downloads

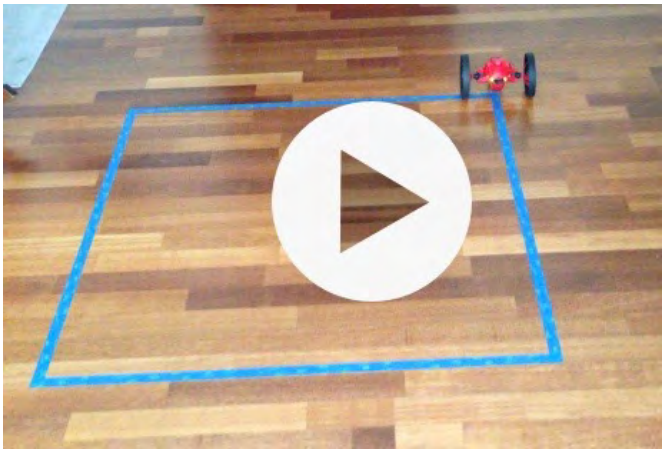
Now I had to make a second decision. What robot should I build? I got way ahead of my capabilities and started with Gyro Girl (well, actually named Gyro Boy, do a Google search). This uses the gyro sensor to build a robot that balances and moves on 2 wheels like the Segway. I then throttled back my enthusiasm, and decided to start off small.

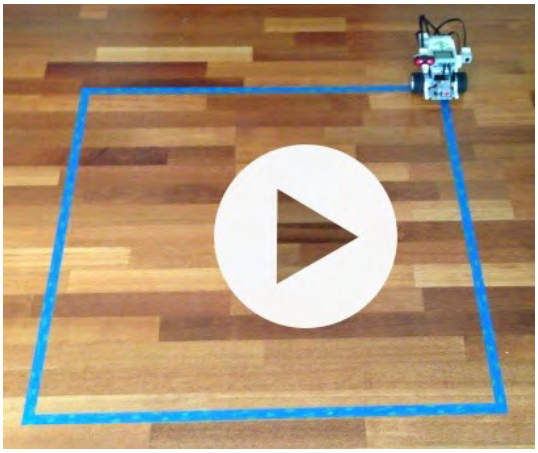
Therefore, I decided to build a "getting started" robot. It has 2 drive wheels using the large motors and a single rear wheel for support. All 4 sensors that come with the core kit are installed. This provides a platform to experiment with as I write code to activate the various sensors to direct the robot to perform various tasks. I will report on my experience next week.

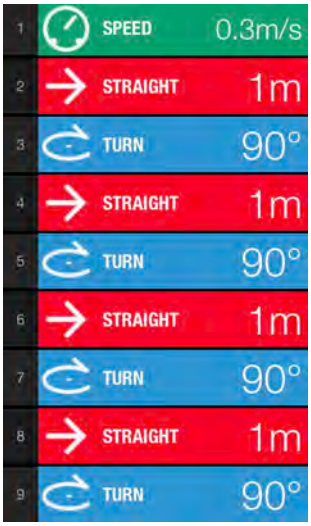
9. EV3 Code to Drive in a Square



I thought a good first step in learning about Lego Mindstorms EV3 was to build a Tribot and have it drive around a square 1m on edge. There is a plethora of information about EV3. Do a Google search on “build EV3 Tribot” or “EV3 drive in a square”. The number of sites identified is overwhelming and it’s difficult choosing where to start. The Art of Lego Mindstorms EV3 Programming, by Terry Griffen, is perfect as a getting started guide. I made some modifications to the build so I could include all 4 sensors that came with the kit. The 2 videos shown below show the Parrot drone and EV3 navigating a 1m square.

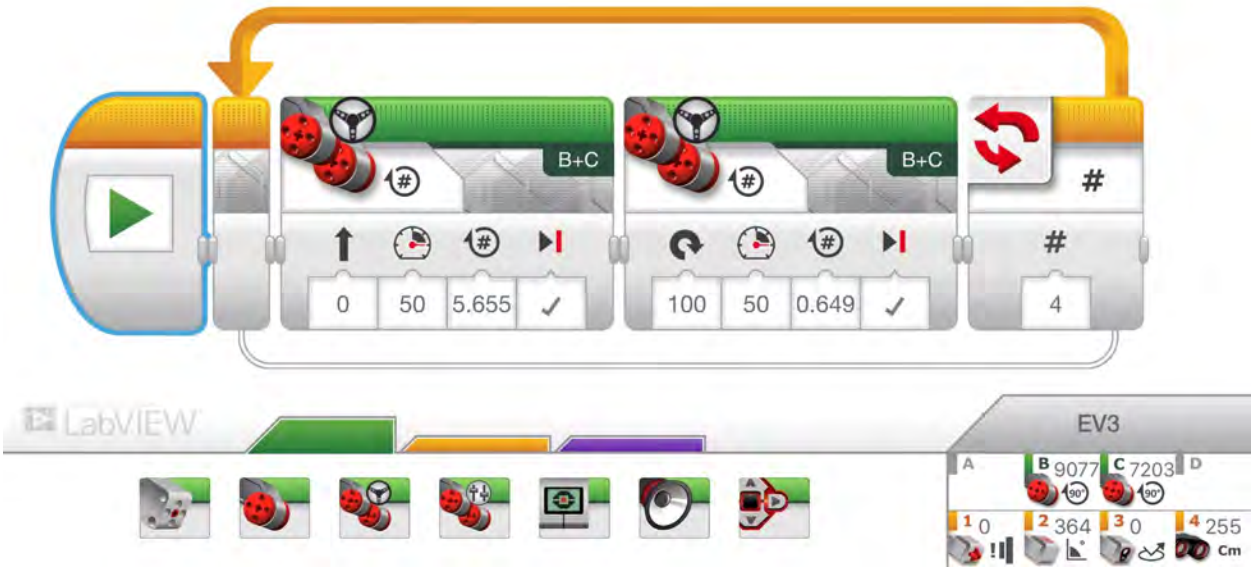






Writing the code to direct the Parrot drone to drive in a square is straightforward. The Parrot iPad app has the macro commands for “straight” and “turn” as shown in the code to the left. The only shortcoming is that “loops” are not available so the commands to drive straight 1m and turn right 90 degrees have to be repeated 4 times.

Code for EV3 is a bit more involved. There are several ways to write the code to direct EV3 to drive straight and turn. One way is shown in the following code. In the lower right corner, LabVIEW shows that the 2 motors are connected to ports B and C. These are then used in the



two “move-steering” command blocks. The sensors, which are not currently being used, are connected to ports 1, 2, 3, 4. The code consists of 2 “move-stearing” blocks contained within a loop to repeat 4 times. Four input parameters are required and their meaning changes with the mode chosen for the command by clicking the clockwise arrow in the upper left corner of the command. The first move-steering block to direct EV3 to move in a straight line for 1m requires 4 input parameters.

1. 0 == both motors synchronized to move EV3 in a straight line
2. 50 == motor power
3. 5.655 == number of motor rotations. The meaning of this parameter changes according to the mode selected (e.g., seconds, degrees, rotations).
4. brake at end

So, how did I know to enter 5.655 for the number of rotations. Several training examples suggest measuring how far EV3 travels for 1 rotation. Then divide this number into 1m to get the number of rotations needed. A more elegant way is to measure the wheel outer diameter and then hand calculate the rotations.

wheel d=0.05629m (using a Vernier caliper)
wheel circumference = πd = 0.1768 m/rotation
number of rotations = 1m / 0.1768 = 5.655 rotations

The second “move-steering” block to direct EV3 to turn right 90 degrees requires 4 input parameters.

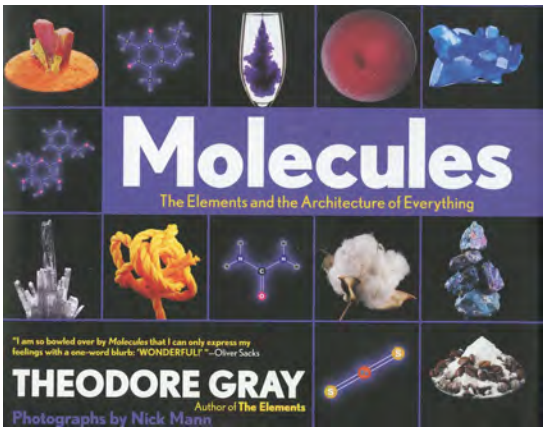
1. 100 == one wheel turns clockwise and the other counterclockwise at the same rate resulting in EV3 rotating around a vertical centerline bisecting the wheel axel.
2. 50 == motor power
3. 0.649 == number of motor rotations
4. brake at end

The number of rotations was calculated as follows:

wheelbase = 0.1460m = rotational diameter
90 degree turn circumference = 0.1460 π / 4 = 0.1147
number of rotations = 0.1147 / 0.1768 = 0.649

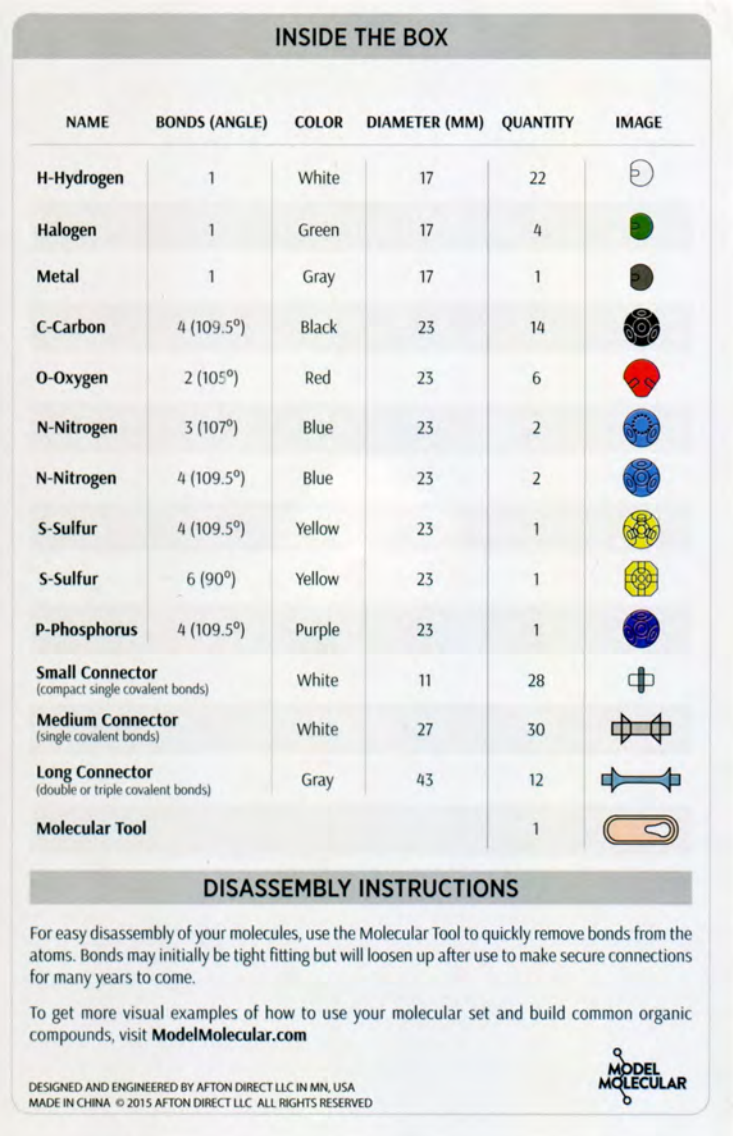
The code block for the gyro sensor can be included in the code which allows direct entry of 90 degrees for the turn. My next step is to activate and play with the various sensors.

10. Molecules



Rheannon is tired of coding and robotics so it's time to move on to other STEM activities. All children love to build structures with wooden or cardboard blocks. I decided to introduce Rhea to building "Molecules". The book I selected to provide guidance is [Molecules: The Elements and the Architecture of Everything,](#) by Theodore Gray. It is also available as an iPad app. The iPad app has informative embedded videos. There is one showing polar and nonpolar fluids (e.g., water and oil) separating. Another video shows nonpolar hexane molecules infiltrating large chain oil molecules thereby dissolving it. In my opinion, the best part of this book is presenting molecular structures for common everyday chemicals that we use.

I purchased the [Duluth labs Organic Chemistry Molecular Model Student Set] from Amazon to build molecules. This set includes a large selection of atoms and bonds. The atoms are color coded and match the molecular pictures in the book.



INSIDE THE BOX

NAME	BONDS (ANGLE)	COLOR	DIAMETER (MM)	QUANTITY	IMAGE
H-Hydrogen	1	White	17	22	
Halogen	1	Green	17	4	
Metal	1	Gray	17	1	
C-Carbon	4 (109.5°)	Black	23	14	
O-Oxygen	2 (105°)	Red	23	6	
N-Nitrogen	3 (107°)	Blue	23	2	
N-Nitrogen	4 (109.5°)	Blue	23	2	
S-Sulfur	4 (109.5°)	Yellow	23	1	
S-Sulfur	6 (90°)	Yellow	23	1	
P-Phosphorus	4 (109.5°)	Purple	23	1	
Small Connector (compact single covalent bonds)		White	11	28	
Medium Connector (single covalent bonds)		White	27	30	
Long Connector (double or triple covalent bonds)		Gray	43	12	
Molecular Tool				1	

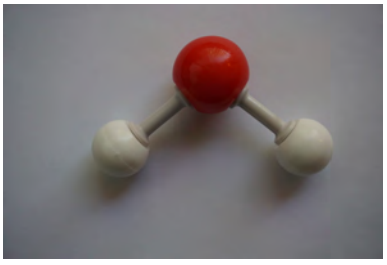
DISASSEMBLY INSTRUCTIONS

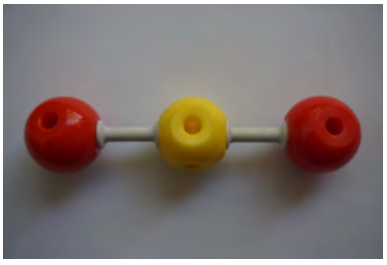
For easy disassembly of your molecules, use the Molecular Tool to quickly remove bonds from the atoms. Bonds may initially be tight fitting but will loosen up after use to make secure connections for many years to come.

To get more visual examples of how to use your molecular set and build common organic compounds, visit **ModelMolecular.com**

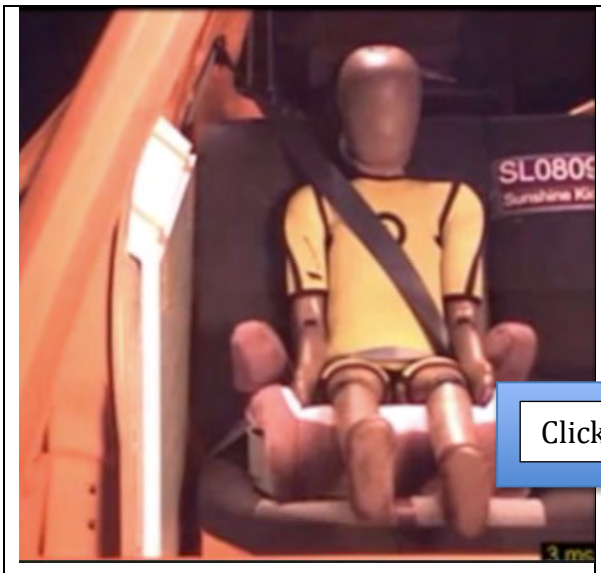
MODEL MOLECULAR

We started off building simple inorganic molecules, such as water H_2O and carbon dioxide CO_2, before moving onto organic molecules (e.g., CH_4). I discussed bond angles with Rhea.





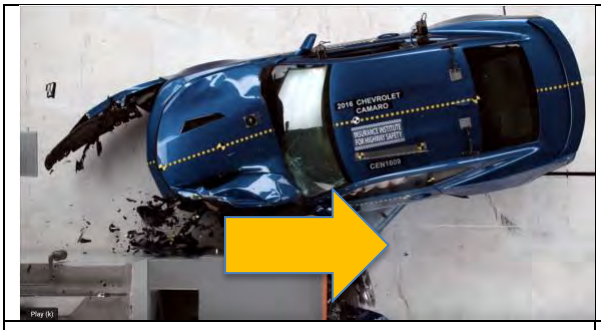
11. Keeping Children Safe in Crashes

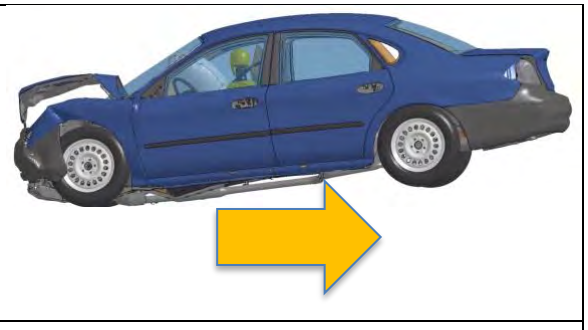


Booster Cushion	High Back Booster Seat

This section is still being thought out. A milestone for a young child is transitioning from a child's car seat to a booster seat. Many parents purchase a booster cushion as shown in the left hand picture above. Children really like this type of seat because of the way it looks. It mimics the seat their parent is sitting in. Parents like it because it is small, light weight, and easy to transfer between vehicles. Please only purchase a high back booster seat shown in the right hand picture. I showed the above video to Rheannon to convince her of the safety issues between the 2 designs.

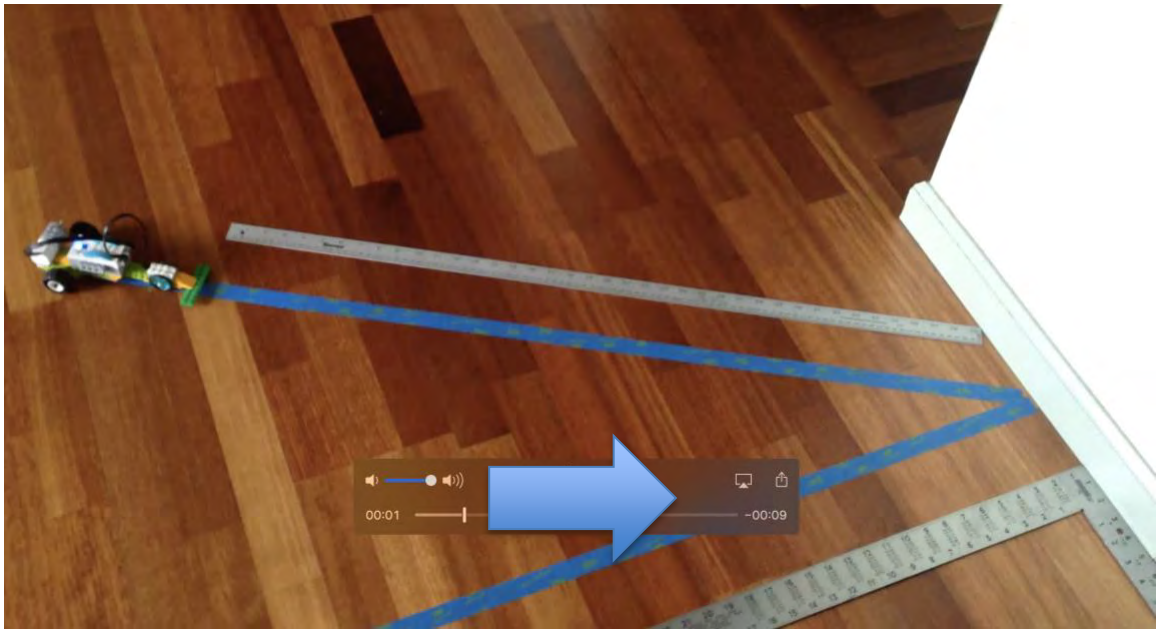
Car crash safety involves both full scale experiments and finite element computer simulation demonstrated in the below videos.



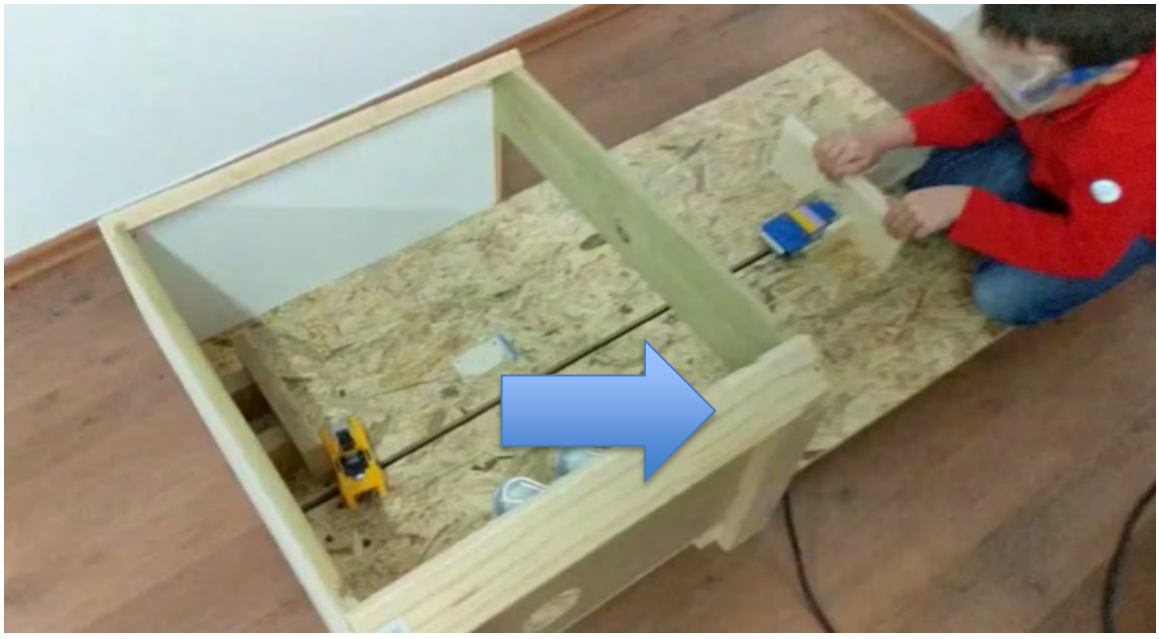


Full scale experiment	Computer simulation

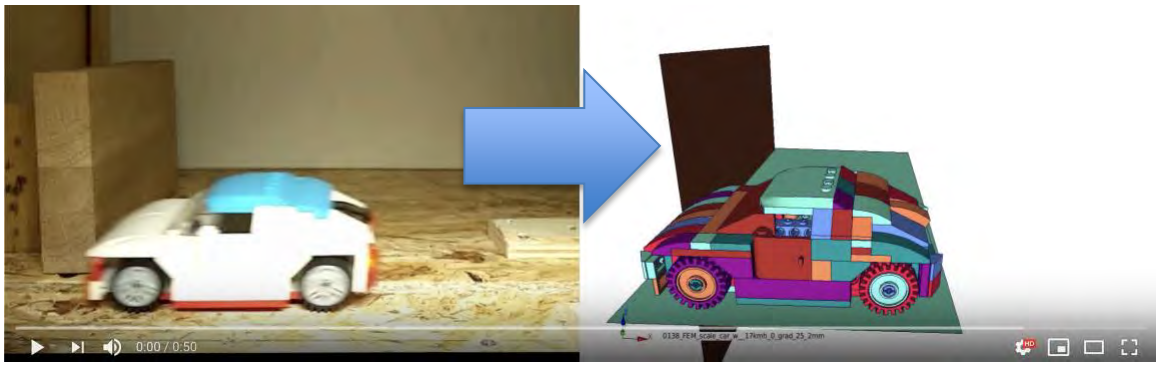
My objective is to create a lesson on Car Crash using Lego vehicles. Then introduce simulation software such as LS-DYNA. I started with the WeDo race car and crashed it into a wall at an angle. The front bumper stays attached at low velocity but breaks free at higher velocities.



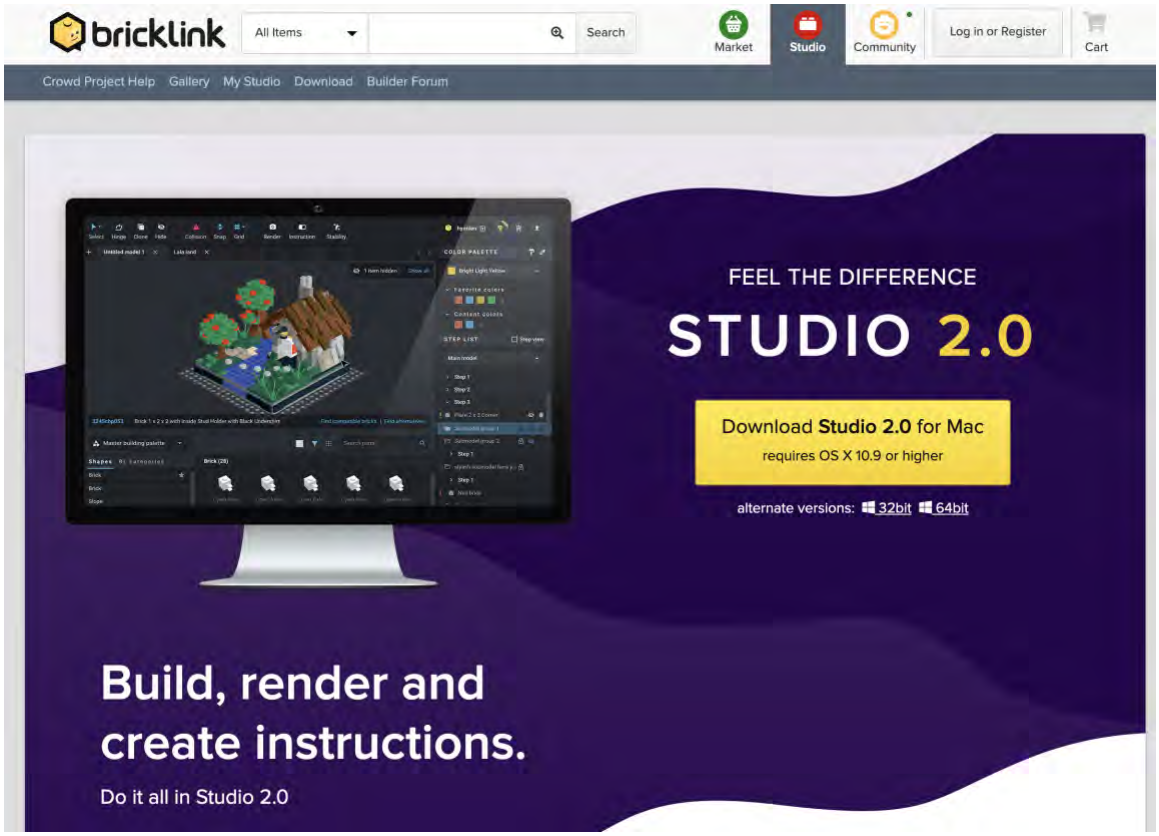
Marko Thiele (SCALE GmbH, Germany) built an acceleration sled to launch a Lego car model into other objects as demonstrated by his son. SAFETY come first -- note that his son is wearing safety glasses.

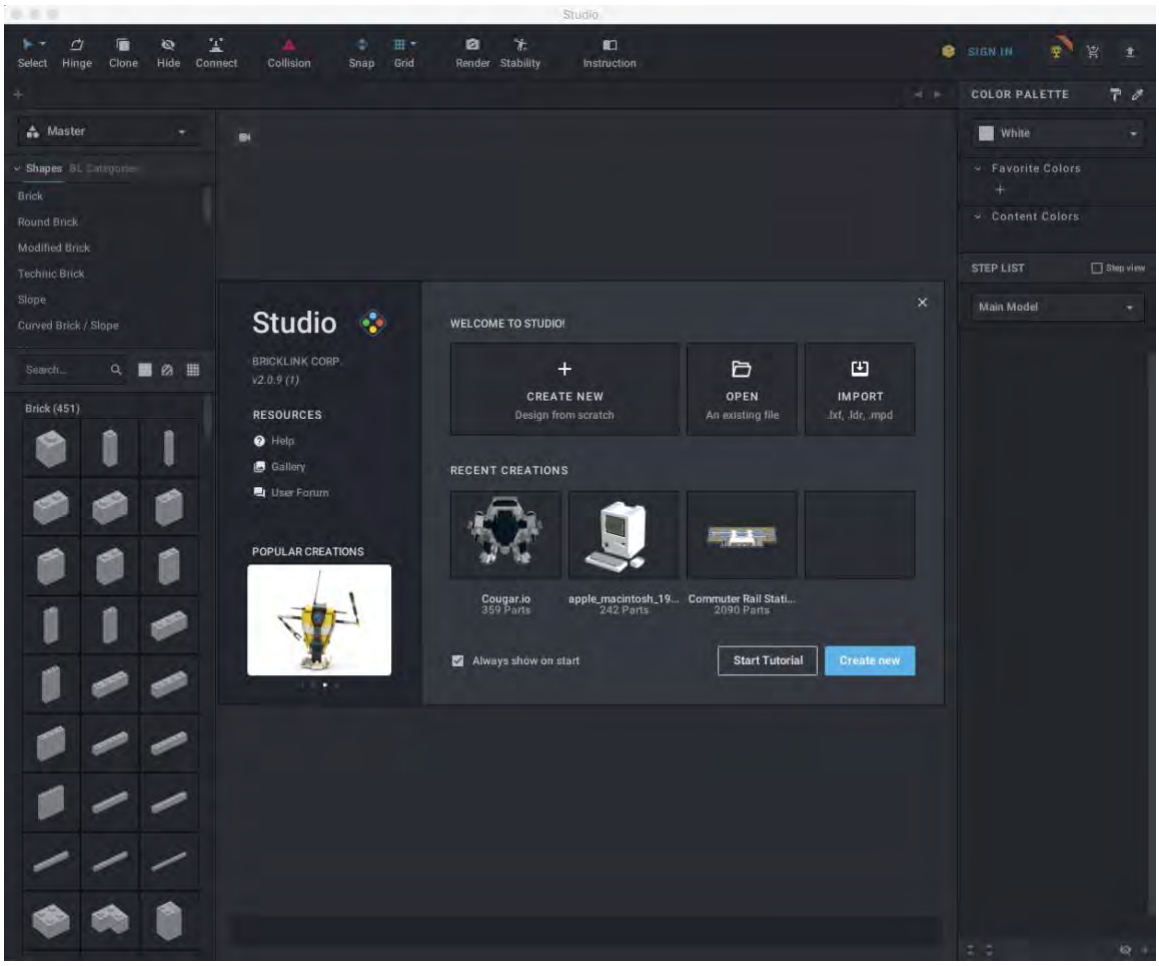


The top can be put on the test cell and high speed cameras mounted to record the experimental car crash. The experiment is shown in the left hand frame and a computer simulation using LS-DYNA is shown in the right hand frame.



BrickLink Studio (https://studio.bricklink.com/v2/build/studio.page) is a CAD system to construct Lego models. The instruction set created is used to help construct a finite element model for the computer simulation using LS-DYNA.




Studio
Select
Hinge
Clone
Hide
Connect
Collision
Snap
Grid
Render
Stability
Instruction
SIGN IN
COLOR PALETTE
Master
White
Shapes
Brick
Round Brick
Modified Brick
Technic Brick
Slope
Curved Brick / Slope
Favorite Colors
Content Colors
STEP LIST
Step view
Main Model
Search...
Brick (451)
Studio
BRICKLINK CORP.
v2.0.9 (1)
RESOURCES
Help
Gallery
User Forum
POPULAR CREATIONS
WELCOME TO STUDIO!
CREATE NEW
Design from scratch
OPEN
An existing file
IMPORT
.txt, .ldr, .mpd
RECENT CREATIONS
Cougar.io
359 Parts
apple_macintosh_19...
242 Parts
Commuter Rail Stati...
2090 Parts
Always show on start
Start Tutorial
Create new

11. Books for girls

Women are under represented in the professional fields of engineering and physics. Here are 3 books that I found which have a girl as the lead character. They are available from Amazon.

Hello Ruby Adventures in Coding
By: Linda Liukas

The Most Magnificent Thing
By: Ashley Spires

Rosie Revere, Engineer
By: Andrea Beaty

And, don't neglect the iPad app "Monument Valley"