Code Car Simulator

Lesson Overview

Lesson Summary

This lesson is intended to last 45-60 minutes. There are included options to extend beyond the Hour of Code.

Getting Started: (2-5 mins)
- Navigate to the Hour of Code page and Let's Start Coding activity.
- Introduce Code Car Simulator.

Activity: (20-40 mins)
- Facilitate and guide students to complete the Code Car simulation.
- Provide guiding questions to students struggling or needing more challenges.

Wrap-up: (5-10 mins)
- Exit discussion and closing.

Assessment/Extended Learning: (2-5 mins)
- Optional code sharing activity.

Audience:

This lesson is targeted for students in grades 4-8, who are in computer science and/or are interested in learning typed code.

Aligned Standards for Grades 4-8:	Lesson Objectives: As a result of participating in the lesson activities, students will be able to:	Activities:
Common Core Math CCSS.Math.Practice.MP1, CCSS.Math.Practice.MP2, CCSS.Math.Practice.MP5, CCSS.Math.Practice.MP6, CCSS.Math.Practice.MP7, CCSS.Math.Practice.MP8,	- Understand the different parts of the code. - Understand that a code 'loop' is code that repeats over and over and will continue to run until it is told to stop.	- Turn on the Headlight - Tail Light Blink - Flash Both Siren Lights

CCSS.MATH.CONTENT.4.NBT.A.2, CCSS.MATH.CONTENT.5.OA.A.1, CCSS.MATH.CONTENT.5.MD.A.1, CCSS.MATH.CONTENT.6.EE.B.6, CCSS.MATH.CONTENT.8.FA.1 **Common Core ELA** CCSS.ELA-LITERACY.RI.4.1, CCSS.ELA-LITERACY.RI.4-5.7, CCSS.ELA-LITERACY.RI.4-5.10, CCSS.ELA-LITERACY.L.6-8.4, CCSS.ELA-LITERACY.RST.6-8.3, CCSS.ELA-LITERACY.RST.6-8.4, CCSS.ELA-LITERACY.RST.6-8.10 **Next Generation Science Standards** NGSS.3-5-ETS1-2, NGSS.3-5-ETS1-3, NGSS.MS-ETS1-1 **International Society for Technology in Education** ISTE: 4.a., 4.b., 4.c., 5.a., 5.c. 6.b. **Computer Science Teachers Association** CSTA: 1B-AP-10, 1B-AP-11,1B-AP-12, 1B-AP-15, 2-AP-10, 2-AP-11, 2-AP-13	- Adjust and change delay functions. - Change pinMode () numbers. "Comment out" code. - Create and add new code to existing code. - Analyze computer command precision (i.e.Changing delay(1000); to delay(1001); will add one millisecond (1/1000th of a second) of extra delay to the loop). - Combine actions within an 'if' statement. - Write 'if else' statements. - Following patterns in code in order to edit to make a change. - Create and manipulate 'if' statements. - Write digitalWrite commands. - Integrate multiple programs.	- Alternate Siren Lights - Hit the Brakes - Siren Buttons - Siren Toggle Button

Facilitation Guide

Materials, Resources, and Preparation

- Students must use a computer such as Windows, Mac, or Chromebook to manipulate the buttons of the Code Car. Tablets and iPads are not compatible.
- Each activity has a Code Walkthrough video. Instructors may opt to have students watch each activity video as a group, as they move through the activities; or let students use headphones to listen and follow the videos independently, as they move through the activities.
- Some students may choose to manipulate code on one activity more than the other. This is okay, if time is spent unequally between all activities. One code concept may interest a student more than another.

- For extended writing activities and connection to ELA standards, having notebooks or paper for students to record observations is recommended.
- Have participation certificates printed and ready to hand out post-lesson.
- Review the Code Car simulator activities before facilitating the Hour of Code with the Let's Start Coding Code Car Simulator.
- Promote and register your Hour of Code event with Let's Start Coding Code Car Simulator.
- This activity requires the Google Chrome or Mozilla Firefox browser.

Code Vocabulary and Concepts:

pinMode(pin, INPUT/OUTPUT/INPUT_PULLUP);
Pin mode sets up your hardware pin to receive power, allow power to flow through, or allow a resisted amount of power to flow through. Each time you use it, you have to supply the pin number you are addressing and the setting you want for that pin.

digitalWrite(pin, HIGH/LOW);
digitalWrite works with pins set as OUTPUTs in pinMode(). It is a function that sends either 5V or 0V to the pin you supply it. This function only has two settings: on or off. To specify on or off, though, you can use the words HIGH or LOW, true or false, 1 or 0.

delay(milliseconds);
The delay function pauses your program for the number of milliseconds you supply it as an argument.

digitalRead(pin);
The digitalRead() function only takes one argument: the pin number you want to read. Its job is to listen to the pin that the programmer provides and report back what its voltage level is.

if- else if -else
if(condition here is true){ run all of these code statements } else{ run these statements}
An "if" statement creates a branch within your program so that your code 'makes decisions.' It tests a condition and, if that condition is true, some code statements will run. If the condition is false, the 'if' statement is passed over.

== equal to: value == value;
The 'equal to' comparator is two equal signs back-to-back. It is not *setting* two things equal, but is returning 'true' *if two things are equal.* You will often want to

compare two things and do something if they are equal, like checking if the current time is equal to the alarm you have set.

void setup( ){ }
Void setup is technically a function that you create at the top of each program. Inside the curly brackets is the code that you want to run one time as soon as the program starts running. You set things like pinMode in this section.

void loop( ){ }
The loop is another function that Arduino uses as a part of its structure. The code inside the loop function runs over and over as long as the Maker Board is turned on.

HIGH or LOW
When used with pinMode(OUTPUT) and digitalWrite(), HIGH means 5V is sent to the pin you specify. LOW means 0V is sent to the pin.
When used with pinMode(INPUT) or pinMode(INPUT_PULLUP) and digitalRead(), HIGH means that the pin is reporting at least 3 volts. LOW means the pin is reporting less than 3 volts.

Getting Started (2-5 mins)

Introduce the Code Car Activity (2-5 minutes)

Navigate to https://www.letsstartcoding.com/headlight-on-hoc/ to find the Let's Start Coding Hour of Code Activities. The instructor role is to let students progress through the activities independently while the instructor addresses concepts and asks the provided guiding questions.

Explain and discuss how code impacts our everyday lives, about how it is the instructions that tell a computer what to do. Code is written by humans, for humans. It is written by people and it's written to do things for people.

Activity (30-40 minutes)

Read the Example Program and Identify its Parts (2 minutes)

Students should be able to identify what the example program on the page does by reading the comments. The comments are the text in green font and are identified by starting with two forward slashes // or a */. These are notes from another programmer to you.

Let'sStartCoding:Education

Ask some students to share a hypothesis of what will happen when this code runs in the code editor. This exercise will help them build their coding vocabulary and articulate their thoughts in computer terms.

WALKTHROUGH VIDEOS:

Watch the videos for line-by-line explanation of how the example program works.

CHALLENGE:

Experiment with the Code Challenges (10 minutes per activity)

The following questions are good to ask students as they are moving through the activities independently. It is good to share with them that they can go back to previous activities to copy or change code after they have learned something new.

Activity:

Turn Headlight On
- *What other computer items do we see in our home, on the street, in a building, or in a car that may use a light?*
    - *Answer: Students may mention car turn signal or indicators, crosswalk signs, intersection signals, motion detectors, etc.*
- *What happens if you turn a line of code into a comment by placing // in front of the line?*
    - *Answer: If you place // in front of a line of code, it will "comment out" that line and the code will not run for that specific line of code.*

Tail Light Blink
- *What do you see that is similar in this activity compared to the last one?*
    - *Answer: Students may mention the pinMode, digitalWrite, and void setup and loop.*
- *What do you see that is different?*
    - *Answer: Students may mention the delays and how the tail light is blinking and digitalWrite(LOW) or digitalWrite(HIGH).*

Flash both Sirens
- *Can you make the siren lights flash faster?*
    - *Answer: Students will need to make the delays smaller.*
- *Can you make all the lights stay on forever?*
    - *Answer: Students may delete the delay and LOW commands to make the lights stay on indefinitely.*

Alternate siren lights
- *Remember to break down the problem into parts, get key information, and describe what you are trying to do in order to understand the problem.*

- *Answer: Students should look at the code line by line and read the code comments to determine what parts of code go together.*

Brake Pedal
- *When designing new code, what are some ideas that you want your code to do?*
  - *Answer: Student answers will vary on what they are wanting to design.*

Siren Lights Button
- *What other types of variables do you think you can make with Code Car?*
  - *Answer: Student answers and ideas will vary.*

Siren Toggle Buttons
- *Can you read the code and tell what it will do based on what you have read?*
  - *Answer: Students should be able to read the code comments to determine what the code is supposed to do.*
- *Do you know if you are right? Why don't you test it out or see if you can make changes to the code and it will change your result.*
  - *Answer: Student answers and attempts will vary based on what changes they want to make to the code.*

When a student finishes early:

- When students finish an activity, encourage them to go to the next activity or to create something new from the code examples used in all of the activities.
- Students may also go back to the previous lesson and continue to manipulate and create new code combinations.

When a student is struggling:

- If a student is struggling with the activities, encourage him/her to look at patterns in the code, and to try and copy the patterns and see what they do.
- Remind them that there is always the restore button that they can use if they feel that they have messed up the code too much.

ELA and NGSS Connections (optional): Recording Observations (5 minutes)

Coding is an iterative process and students will often take a step, upload it, observe it, and then take another step toward their goal. That is a good thing! Encourage students to have a hypothesis each time they upload a program. Even if they are experimenting, they should have a guess as to what will occur. This process helps students slow down, read their code, and consider it rather than press 'upload' many times until they are lost and frustrated.

Finally, recording solutions can help students think more about their problem-solving skills and identify the flaw in their thinking that led to an error. The solution may be worded in terms of

coding- "I added a delay(1000); to the code"- or in terms of thinking - "I remembered that the loop will run very quickly without code to slow it down."

Wrap-Up (5-10 mins)

- Discuss and share what happened in the activity.
  - How does coding relate to humans?
  - How do you feel about reading code?
  - How do you feel about typing code?
  - Do you think you can create new code?
- Pass out certificates for Hour of Code activity participation.

Assessment/Extended Learning (2-5 mins):

Share Code Car Creations with Classmates (2-5 minutes)

Encourage students to ask the person next to them "What did you make your code do?" And "Can you show me how you did that?" This encourages collaboration and new ways of thinking.

Alternately, you can choose one student volunteer to show their project to the class and describe the process they followed to reach that final project. If space allows or a projector is available, students could gather around and observe the actual lines of code that their classmate wrote. If you have a projector available, showing the code to the rest of the class can reveal different ways of reaching the same solution and spark a Q & A with the student answering questions about their program.

Beyond one Hour

- If you would like to experience more than the Hour of Code provided by Let's Start Coding, then visit https://www.letsstartcoding.com/ .

Let'sStartCoding:Education