

HOPSCOTCH

“Don’t Drop Your Phone” Game

Teacher Notes

Time | 60 minutes

Big Idea | Coding allows you to create something from nothing.

Lesson Goals

- Students will gain exposure to programming fundamentals through creative play, sparking an interest in continued learning.
- Students will build and customize a tapper game.
- (optional) Students will evaluate their own work as well as that of their peers in a positive and helpful manner.

Skill Focus

- Make sense of problems and persevere in solving (CCSS.MATH.PRACTICE.MP1)
- Look for and make use of structure (CCSS.MATH.PRACTICE.MP7)
- Designing solutions (NGSS Practice 6)

Key Vocabulary

- **Coding:** Telling computers what to do
- **Clone:** Generate a copy of something
- **Event:** When something happens
- **Loop:** Code that repeats
- **Random:** When there’s no pattern
- **Range:** The highest and lowest number random can choose between
- **Rule:** Instructions that tell your computer what to do (the command) and when to do it (the event)
- **Sequence:** A list of instructions, in order

Materials

- 1 iPad or iPhone per student, or 1 device per 2 students, for pair programming. Email educators@gethopscotch.com for bulk-created student accounts.
- “Don’t Drop Your Phone” example projects:
 - Tapper example project: <https://community.gethopscotch.com/projects/z2la2yty8>
- “Don’t Drop Your Phone” video tutorial
 - Available on YouTube: <https://youtu.be/QVrmVOs6slg>
 - Available in the app as well

Educator Brief

Anyone, regardless of their experience in programming, can teach this Hour of Code lesson. Just as Hopscotch was built on the principle that anyone can become a great programmer, this lesson is designed on the premise that anyone can teach basic programming, including you!

HOPSCOTCH

Students will code their own custom versions of a tapper game in which they player tries to keep an iPhone from falling in the toilet. Students will build the program from scratch following the instructions from a video tutorial. They will gain exposure to computer science concepts and apply them in novel ways to construct a unique project. You can facilitate this lesson in several ways: 1) You can play the video for the whole class to follow together; or 2) Students can complete the lesson following self-paced tutorials on their devices. During the tutorial, there are opportunities for debugging, sharing, and discussion. Suggested modifications are included below and may prove helpful for students in need of more structure. This lesson assumes that students and teachers have little to no experience coding in Hopscotch.

Lesson

0. Whole Group Discussion (5 minutes)

- Introduce the idea of Hour of Code to students. Get students excited about being able to create technology.
- Introduce the project to students; explain that they will be using Hopscotch to code a tapper game about keeping an iPhone from falling into the toilet.
- (Optional) Share example projects with students. As a class, discuss these projects. What happens? What does the player do to generate fireworks? Can students identify elements of the fireworks that the programmer chose? (e.g. color, size, speed, direction)

1. Getting to Hopscotch (5 min)

1.1 Find the Hopscotch app on your iPad

1.2 Sign in to your account (students may need to create accounts)

1.3 Making a new project: Tap the + button on the bottom of the screen, labelled "Templates." If you are using the in-app tutorial, scroll down and tap "Make a Game in minutes." Then choose the "Don't Drop Your Phone" project. If you are projecting the YouTube video for your class, tap the white circle + button on the bottom right labelled "Create."

2. Getting Familiar with Hopscotch (10min) (optional)

2.1 Exploring the Workspace & Blocks

- Open a blank project in Hopscotch
- Tap the grey "+" button at the bottom of the screen to add an object
- Tap "+Add Code" to explore the editor
- Explore the magenta "When" blocks and discuss the idea of a trigger
- Explore the colorful code blocks to see what they do

2.2 Check In & Share Out

- Ask students to share things that they noticed that might be helpful for projects.
- Answer common questions, if necessary.

HOPSCOTCH

3. Create Your “Don’t Drop Your Phone” game (30min)

3.1 Coding the game (30min)

- Students create projects in Hopscotch in which iPhones fall down the screen and the player taps to keep them from falling in toilets.
- The in-app video tutorial provides step-by-step instructions and explanations.
- The video suggests opportunities for students to pause and add code to their projects. You can use these moments to ask questions, help students debug, or let students help each other.

4. Sharing Apps in the Hopscotch Community (optional) (15min)

4.1 Publishing a Project

- When students finish, they can publish their project in the Hopscotch community. This is a great opportunity for students to share their work with a digital audience. Students can “like” another student’s project and remix it to see how they made it.
- Note: if you are using student accounts created by Hopscotch, your students can find their classmates’ work in their “Following” tab.

4.2 Commenting on a Classmates' Project

- Show students how to search for projects in the Hopscotch community.
- Encourage students to remix their classmates’ projects in order to see how they customized their apps.

5. Reflection (5 min)

5.1 Choose 1-2 student projects to share and discuss as a class

- Show the project(s) to the class and discuss how the student customized their game.
- Students can also share their reflections on coding. How did they imagine it would feel to create a game? How does it feel to have done it? What did they learn? Do they think about technology differently now that they know a little about coding?

Modifications

- Pair programming: this is a great opportunity for students to work together to create strong projects that they are proud of
- Build in time for students to get comfortable in Hopscotch; encourage students to work through all of the introductory tutorials

Curriculum Extensions

- Add an extra challenge - add a score or a timer to the game.
- Self-assessment: Using this rubric as a model¹, encourage students to assess their work, evaluate its effectiveness, and iterate to improve the project.

¹ Rubric: <https://drive.google.com/file/d/0BwKFT8tX5oZVbzkwWHN0OFBOM3c/view>

HOPSCOTCH

Common Core State Standards

- Make sense of problems and persevere in solving (CCSS.MATH.PRACTICE.MP1)
- Look for and make use of structure (CCSS.MATH.PRACTICE.MP7)
- Designing solutions (NGSS Practice 6)