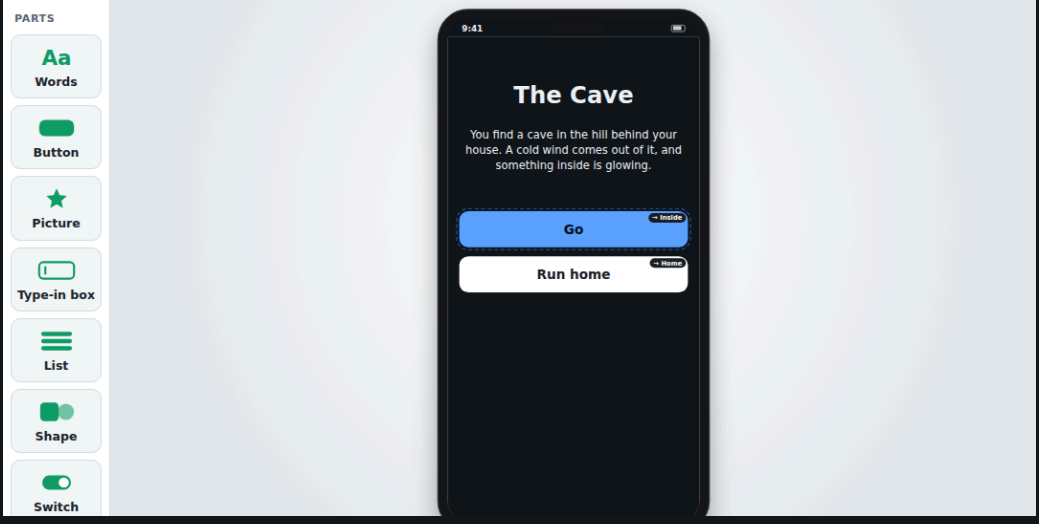
mytekDev

App Maker: Somebody Else Tries It

Grades 6-8 · 45 minutes

Computer science · Technology



Slides, rubric and answer key **Print and go** **No student accounts**

Chromebook friendly

What's included

- A one-page overview: objectives, essential question, vocabulary, standards and assessment
- A ten-minute walkthrough to rehearse before class, and background in plain words for the teacher
- A 45-minute lesson sequence: warm-up, direct instruction, guided practice, independent practice, closure
- Slides for the board: a PowerPoint file that also opens in Google Slides
- A student worksheet with a QR code and a record sheet for a user test
- A 3-question exit ticket, two to a page, with an answer key
- A rubric: 3 criteria, 4 levels, scored out of 12
- Differentiation for English learners, students with IEPs or 504 plans, and students who need more support or more challenge
- Cross-curricular extensions, and notes for teaching it in grades 4 and 5 and in grades 9 and 10
- Standards: CSTA 2017

App Maker: Somebody Else Tries It

Grades 6-8 · 45 minutes · Computer science, Technology

Students get a small story app that its maker never showed to anyone. It has three problems: a screen with no way out, buttons whose words do not say what they do, and a hint that never appears. Students find the first with a guide. Then they trade seats, watch a partner use the app without helping, and write down what the partner taps and expects. They fix the app from what they saw, and the partner tries it again.

At a glance

ESSENTIAL QUESTION

Why does an app work for the person who made it and not for the person who uses it?

OBJECTIVES

Students will be able to:

- explain why the maker of a program is a poor tester of it
- run a user test: watch someone use an app without helping, and record what they do
- turn what a tester did into a change to the app
- test again after a change, with the same person

VOCABULARY

User, User test, Feedback, Dead end, Label

STANDARDS

2-AP-15, 2-AP-17, 1B-AP-15

TIME AND MATERIALS

45 minutes, grades 6-8.

- One computer or Chromebook per student
- A projector
- The worksheet and exit ticket, one copy each per student
- The rubric, one copy per student if you grade the work

PREP

Make copies of the worksheet and exit ticket. Before class, open mytekdev.com/try/appmaker?lesson=somebody-else-tries-it&ref=tpt-appmaker-4 once on a student computer: it opens on a dark story app called The Cave, and checks that your school's filter allows the app. Run it yourself, about ten minutes: press Try it, tap Go, then Take it (you are stuck on a screen with no button); press Keep designing, click Button on the left, type Start again and point it at The Cave. Then click the Inside card and look for the faint line at the bottom: a hint nobody can see. Decide how partners will trade seats.

ASSESSMENT

- Formative: the questions under What to look for, asked while students work, and the exit ticket (answer key included).
- Summative: the worksheet, scored with the rubric.

TECHNOLOGY AND PRIVACY

Runs in any browser, Chromebooks included, with nothing to install. Students need no account and no email: the link works signed out, and nothing they do is stored on our server unless they choose to save. The app they build runs on a phone drawn on the page, so no phones are needed. Making a link for a real phone needs a free account and is not part of this lesson.

Lesson sequence

MINUTES	STAGE	WHAT YOU DO	WHAT STUDENTS DO
0 to 5	**Warm-up** Hook	Show the title, essential question and Today I can slides (about 1 minute). Then the warm-up: stuck in an app with no idea what to tap. What did the maker forget?	Answer the warm-up on the worksheet.
5 to 12	**Direct instruction (I do)** Model it	Project the app with the starter open. Do not press Try it yet. Ask a volunteer who has not seen it to come up and use it while the class watches in silence. Whatever they do, write it on the board: what they tapped, what they said. Then give the three rules of a user test: do not help, do not explain, write what they do. **It's working when:** The hint on the Inside screen starts hidden. In Design it shows faint, with a dashed outline, so the maker can pick it. In Try it nobody can see it.	Watch the volunteer in silence, and fill in the vocabulary on the worksheet.
12 to 20	**Guided practice (We do)** Find one, fix one	Students open the link or scan the QR code. A short guide walks them to the dead end and has them add a way out. When most have fixed it, say there are two more problems, and that they will not find them alone. **Watch for:** Students who start fixing everything they can see. Stop them: the point is to fix what a user trips on, and they have not watched a user yet.	Follow the guide. Add a Start again button to The crystal.
20 to 38	**Independent practice (You do)** Test, fix, test again	Partners trade seats for five minutes. The tester presses Try it and uses the app, saying aloud what they expect each button to do. The maker watches, says nothing, and fills in the record sheet. Then they trade back, fix at least two things, and the same partner tries it again. Walk the room with the What to look for questions. **If they ask:** "What did your tester do that told you the button's words were wrong? What do the words say now?" Evidence	Run the test, fix the app, and run it again.
38 to 45	**Closure** Showcase and exit ticket	Show the Showcase slide. Two students say what their tester did that they did not expect, and what they changed. Hand out the exit ticket.	Answer the exit ticket.

Before you teach: a ten-minute walkthrough

Open mytekdev.com/try/appmaker?lesson=somebody-else-tries-it&ref=tpt-appmaker-4. A short guide opens over the app. The phone shows a dark screen: The Cave, a few lines of story, and two buttons, Go and Run home. Four cards sit under the phone: The Cave, Inside, Home and The crystal.

1. **The dead end.** Press Try it. Tap Go, then Take it. The last screen has a picture and words and no button. There is no way out. Press Keep designing.
2. **Fix it.** You are on The crystal. Click Button on the left. Type Start again in The words on it, and under Then go to pick The Cave.
3. **The labels.** On The Cave, the first button says Go. On Inside, the second says No. Neither says what it does. Better: Go into the cave, and Leave it there.
4. **The hint.** Click the Inside card. At the bottom is a faint line with a dashed outline: Hint: a loose crystal is safe to pick up. It starts hidden and nothing shows it. To make it work, add a button that says Hint, click + Show or hide a part, and choose show and that line.

> The moment that lands is the silence. A maker who is not allowed to help watches their partner hesitate over a button the maker thought was obvious.

Background for the teacher

The person who makes a program is its worst tester. They know which button to tap, so they never tap the wrong one. They know what each label means, so they never read it. They always take the path they built first, so they never reach the screen they forgot to finish. None of this is carelessness. It is what knowing a thing does.

A user test fixes this with a second person and three rules: do not help, do not explain, and write down what they do. What a tester says they think matters less than what they do. A tester who pauses for three seconds over a button has told the maker something, whether or not they say a word.

This app has three problems of three kinds. A dead end is a structure problem: the map of screens has a box with no arrow out. A label that does not say what a button does is a words problem. A hint that never shows is a logic problem: a part that starts hidden, and no button with a step to show it.

Testing again with the same person is what closes the loop. A fix that the maker thinks is clear is a guess until someone else uses it.

App Maker is not a text programming language. Students build each rule as a sentence and the app writes the code, which they can read and cannot edit. It has no loops and no lists, a button holds one But if, and numbers are the only kind of data an app remembers. What it does have is the real thing in small: events, a sequence of steps, variables and a conditional, in an app that runs.

What to look for

Ask these while students work. A student who has understood the lesson can answer them without help:

- Why did the maker of this app never find the dead end?
- What did your tester do that you did not expect?
- What did you want to say while you were watching? Why were you not allowed to?

- Which of your changes came from the test, and which did you just feel like making?
- How do you know your fix worked?

> The last question is the real test: a student who says "my partner tried it again and did not get stuck" has the whole loop.

Cross-curricular extensions

- English: trade a set of written directions with a partner and follow them exactly. Where did the writer know something the reader did not?
- Art: redesign a sign in your school that people misread. Test it on three people before you call it finished.
- Science: a user test is an observation. What makes it a fair one? What would make it unfair?
- At home: watch a family member use an app they have never opened, and write down the first place they pause.

Vocabulary

User	The person a program is made for.
User test	Watching someone use a program, without helping, to find what gets in their way.
Feedback	What a user says or does that tells the maker what to change.
Dead end	A screen with no way to leave it.
Label	The words on a button. A good label says what the button does.

Differentiation

ENGLISH LEARNERS

- Keep the vocabulary slide up, and point at the screen as you say each word: the cards under the phone and the sentences in the panel carry the meaning.
- Let the tester point and the maker write single words: tapped, expected, happened.
- Pair learners who share a language for the test, and have them report in English.

STUDENTS WITH IEPS OR 504 PLANS

- Pair students in threes where needed: one tester, one watcher, one recorder.
- Zoom the browser to 125%; the phone and the panel stay readable.
- Allow one fix instead of two, where the IEP allows a modified assignment.

STUDENTS WHO NEED MORE SUPPORT

- A sentence starter: "My tester tapped ___ and expected ___, but ___."
- Give the record sheet with the first row filled in from the volunteer at the board.

STUDENTS READY FOR MORE

- Make the hint work: add a button that says Hint, and give it a step that shows the hidden line.
- Add a new screen to the story, then test it on a different partner.

- A list of the three problems, to check against after the test and not before.
- Write three things a tester should try in any app, and try them on a classmate's.

Adapting for other grades

Younger students: In grades 4 and 5, do the guide together on the projector, with pairs sharing a computer, one clicking and one recording. Skip the third rubric row; total 8, and 6 meets the objectives. Run the user test as a class, with one volunteer at the projector, before pairs try it.

Older students: In grades 9 and 10, have students write a test plan before the test: five things the tester will be asked to do, and what should happen for each. After the test they mark each pass or fail, and say which failures a plan could have predicted and which only a person could find.

Student worksheet

Name: _____ Date: _____

Open the app: scan the QR code, or go to mytekdev.com/try/appmaker?lesson=somebody-else-tries-it

Essential question: Why does an app work for the person who made it and not for the person who uses it?

Warm-up: Have you ever been stuck in an app or a game with no idea what to tap? What did the maker forget?

_____

1. **Vocabulary.** User, user test, feedback, dead end, label: write each in your own words.

2. **The record.** While your partner uses the app, write for each thing they do: what they tapped, what they expected, and what happened. Do not help.

3. **My fixes.** Two things I changed, and what my tester did that made me change each.

4. **Again.** My tester tried it again. What was better? What was not?

1. Why did the maker of this app never find the dead end?

2. What is one thing you learned from watching that your tester would not have told you?

Exit ticket

Name: _____

1. In a user test, the maker should (A) explain each button before the tester taps it (B) watch, stay quiet, and write down what the tester does (C) use the app while the tester watches

2. The maker of an app is a poor tester of it because (A) makers are careless (B) the maker already knows what to tap, so they never get stuck (C) makers use faster computers

3. A button says OK. Write better words for a button that deletes a saved game.

App Maker: Somebody Else Tries It

Exit ticket

Name: _____

1. In a user test, the maker should (A) explain each button before the tester taps it (B) watch, stay quiet, and write down what the tester does (C) use the app while the tester watches

2. The maker of an app is a poor tester of it because (A) makers are careless (B) the maker already knows what to tap, so they never get stuck (C) makers use faster computers

3. A button says OK. Write better words for a button that deletes a saved game.

Answer key: exit ticket

QUESTION	ANSWER	WHY
1	B	Helping hides the problem the test was meant to find.
2	B	Knowing how it works is what stops the maker from seeing where it does not.
3	Delete this game	Accept any label that says what the button does. "Yes" and "OK" do not.

Rubric

Score each row from 1 to 4.

Name: _____

CRITERIA	4 EXCEEDS	3 MEETS	2 APPROACHING	1 BEGINNING
The record	At least three things the tester did, each with what they expected and what happened.	Two things recorded.	One thing recorded.	No record.
The fixes	At least two changes, each one answering something the tester did.	Two changes, one of them from the test.	One change.	No change.
Testing again	The same tester tried the changed app, and the record says what was better and what was not.	The app was tested again.	The maker tested it alone.	Not tested again.

Total: ____ / 12. A score of 9 or more meets the objectives.

Standards

Program development standards from CSTA 2017. Students seek feedback from a user by watching them, change the program in response, and test it again.

FRAMEWORK	CODE	GRADES	STANDARD	HOW THE LESSON MEETS IT
CSTA 2017	**2-AP-15**	6-8	Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	Students watch a partner use an app, record what the partner does and expects, and change the app to answer it.
CSTA 2017	**2-AP-17**	6-8	Systematically test and refine programs using a range of test cases.	Students test the app again with the same user after each change, and record what improved.
CSTA 2017	**1B-AP-15**	3-5	Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.	Students find and fix three errors of three kinds: a dead end, unclear labels and a hint that never shows.

Checked November 2026 against: CSTA 2017 K-12 standards progression chart.

Optional: save student work in a free class

The whole lesson runs at the link on the worksheet, with no accounts and no sign-up.

To keep students' work, and see it all in one place, make a free class. It takes about ten minutes, once, and covers every lesson in this series: mytekdev.com/tools/class-setup

Students in a class open mytekdev.com/tools/app-maker/lessons/somebody-else-tries-it/start instead of the worksheet link. Each one gets their own copy of the starter, and the same link reopens it next time.

Terms of use

Thank you for downloading this lesson. The plan, the handouts and the art were made by mytekDev.

You may:

- print and copy it for your students
- change it for your own classes
- post it on your class website or learning management system
- share this file with other teachers, whole and unchanged, with this page included

You may not:

- sell it, or put it in a product you sell
- share a changed copy
- claim it as your own

The app this lesson uses is free and runs in the browser. Students need no account. It is at mytekdev.com.

Questions, or found a mistake? Ask on the Teachers Pay Teachers Q&A for this resource, or write to us at mytekdev.com/contact.

© 2026 mytekDev. All rights reserved.