

BlocksVR: Click to Change

Ages 9-13 · about 45 minutes, in one sitting

Your child plays in a small 3D garden where a box spins and changes color when it is clicked, then opens its rules and finds two plain sentences: always spin slowly, and when I get clicked, change to a random color. They give the ball, the cone and the donut rules of their own, built from cards rather than typed code, ending with a donut that counts clicks and says You win! at five. Along the way they find out why one rule only works once.

What you need

- A computer or Chromebook with a current browser and, if you have one, a mouse. It is a 3D world, so it takes a few seconds to load; on an older Chromebook, close other tabs first.
- The lesson link: mytekdev.com/try/blocksvr?lesson=click-to-change&ref=fam-blocksvr-3. No account, no email and nothing to install.
- Optional: the activity sheet at the end of this plan, printed, and a pencil.

Do it together

1. Open mytekdev.com/try/blocksvr?lesson=click-to-change&ref=fam-blocksvr-3 together. A short guide runs along the top; follow it: press **Play**, click the blue box a few times (drag to look around if you need to), press **Stop**, then click **Box** in the **Your world** list on the left. Ask: you clicked the box and it changed color, so how does it know you clicked it?

You'll know it worked when: The box changed color with each click, and its behavior card on the right has an **Edit** button.

2. Press **Edit** on the box's behavior card. The rules open: two cards, each a When menu, an arrow and a second menu. When always, then spin slowly. When I get clicked, then change to a random color. The When is the event, the menu after the arrow is the action, and together they are a rule. The program checks its rules over and over, many times a second, and the click rule acts only in the moment a click happens. If your child is curious, switch to **Code** to see the same two rules written as code.

You'll know it worked when: Your child can point to the event and the action in each card.

3. Click **Ball** in the list and press **+ Add behavior**. The new program starts with the same two rules as the box. Change the click rule's second menu, what to do, to grow bigger. Press **Play** and click the ball twice.

If you get stuck: Press **Stop** before changing a rule, and **Play** to test it.

You'll know it worked when: The ball grows on the first click, and not on the second.

4. Ask why the second click did nothing. Grow bigger means twice its normal size, not twice its size now, so the second click asks for a size it already has. The computer did exactly what the rule said. One fix: press **+ Add a rule** and make a second card, every 2 seconds, go back to normal size, so the next click can grow it again.

You'll know it worked when: Your child says the bug was in the rule, not in the computer.

5. Give the cone a behavior and change its click rule to say Hello!.

You'll know it worked when: In **Play**, clicking the cone makes it say Hello!.

6. Now the donut. Its click rule becomes add 1 point. Press **+ Add a rule** for a second card: When the score reaches 5, then show You win! after the arrow. Press **Play** and click the donut five times.

If you get stuck: A new card does nothing until both of its menus, the When and the what-to-do, have been chosen. Check the second card first.

You'll know it worked when: The score climbs with each click, and the fifth click wins.

7. Swap one When for a different event (the player is near me, every 2 seconds, the space bar is pressed). Before pressing **Play**, your child says what will happen.

You'll know it worked when: Your child's guess matches what the thing does in **Play**, or they can say why it did not.

8. Then the real question, without trying it: if the donut's click rule said add 1 point and the box's did too, how many clicks would it take to win?

You'll know it worked when: Your child says five clicks in all, on either thing, because the score is one number the whole world shares. That means they see that rules on different things can change the same game.

Talk about it

Ask these while you work together. A child who has understood the lesson can answer them without help:

- Point to the event in this rule, and the action.
- How often does the program check its rules?
- Why didn't the ball grow on the second click?
- Your donut never says You win!. Which card would you check first?
- Without trying it: if the donut's click rule said add 1 point and the box's did too, how many clicks would it take to win?

The last one is the real test. A child who says five clicks in all, on either thing, because the score is one number the whole world shares, understands that rules on different things can change the same game state.

For younger and older kids

Younger: For ages 7 and 8, do every step side by side and keep to the box and the ball.

Older: For ages 13 and up, do the whole lesson in the `Code` view: write each rule as an if statement, explain why the click rule needs no else, and fix the ball's bug with a variable that remembers how big it is.

Keep going

- Make the donut also turn green when the score reaches 5, as well as showing You win!.
- Open Code and change spin(45) to spin(360). What changed? What does the number mean?
- Give the box another rule card: when the player is near me, run away from the player. Can you still click it?

Bring it into other subjects

- Language arts: write three rules for a household helper robot as when and do sentences.
- Science: living things respond to events too. A plant turns toward light: what is the event, and what is the action?
- Art: design a garden where every thing responds to a click in a different way, then have someone at home guess each rule.
- Math: a rule that runs every moment can move something at a steady speed, a little at a time.

For your records

Ready to copy into an hour log, a portfolio or a transcript.

LESSON	BlocksVR: Click to Change
SUBJECTS	Computer science
TIME	45 minutes
WHAT WE DID	Read the rules behind a clickable box in a 3D world and named each rule's event and action. Wrote rules for three more objects with rule cards, found and fixed a rule that only worked once, and built a click-five-times game with a shared score.

SKILLS PRACTICED

events and actions; reading and writing rules; predicting before testing; finding a bug in a plan

Activity sheet (optional)

Name: _____ Date: _____



Open the app: scan the QR code, or go to mytekdev.com/try/blocksvr?lesson=click-to-change

Essential question: How does a program know when to do something?

THING	WHEN (EVENT)	DO (ACTION)	IT WORKED? YES OR NO
Box	I get clicked	change to a random color	
Ball			
Cone			
Donut (rule 1)			
Donut (rule 2)			

1. **Give it rules.** Click a thing in the list and press **+ Add behavior**; its rules open. Change the cards. (**Edit** reopens them later.)

2. **Test it.** Press **Play** and click it. Press **Stop** before you change a rule.

1. Why did the ball grow only on the first click?

2. A new event I tried: When _____, Do _____. What happened?

Quick check

Name: _____

1. In the rule "When I get clicked, do turn red", what is the event? (A) turn red, (B) I get clicked, (C) the rule.

2. A ball has the rule "When I get clicked, do grow bigger." You click it three times. How big is it? (A) twice its normal size, (B) eight times, (C) normal size.

3. Write a rule that makes something spin fast when the player is near it.

Answers

From the app's rule cards. A new behavior starts with the box's two rules (always: spin slowly; I get clicked: change to a random color). Grow bigger sets the size to twice normal, so later clicks change nothing. Add 1 point adds one per click; the score is shared by the whole world.

THING	WHEN	DO
Ball	I get clicked	grow bigger
Cone	I get clicked	say "Hello!"
Donut, rule 1	I get clicked	add 1 point
Donut, rule 2	the score reaches 5	show "You win!"

QUICK CHECK	ANSWER	WHY
1	B	The event is what happens; the action is what the program does.
2	A	Grow bigger means twice the normal size, every time.
3	When the player is near me, do spin fast	Any player-near event with a spin action counts, as cards or as code: if (playerDistance() < 4) { spin(360); }.

Want the whole subject taught?

This is one lesson to do together. If you would rather have the whole subject taught, our Information Technology 1 and Information Technology 2 courses teach virtual reality as part of a full course, with an instructor, live and recorded classes, graded work and credit.

If your state or umbrella school asks, this lesson covers CSTA 2017 1B-AP-10, CSTA 2017 1B-AP-15, CSTA 2017 2-AP-17.

BlocksVR is a mytekDev app, open source under the GNU General Public License 3.0, built on A-Frame (MIT License).

Terms of use

This lesson was made by mytekDev. You may print and copy it for the children you teach at home, and change it for them. You may not sell it, share it with other families, or post it publicly.

The app this lesson uses is free and runs in the browser, with nothing to install and no account needed. It is at mytekdev.com.

© 2026 mytekDev. All rights reserved.