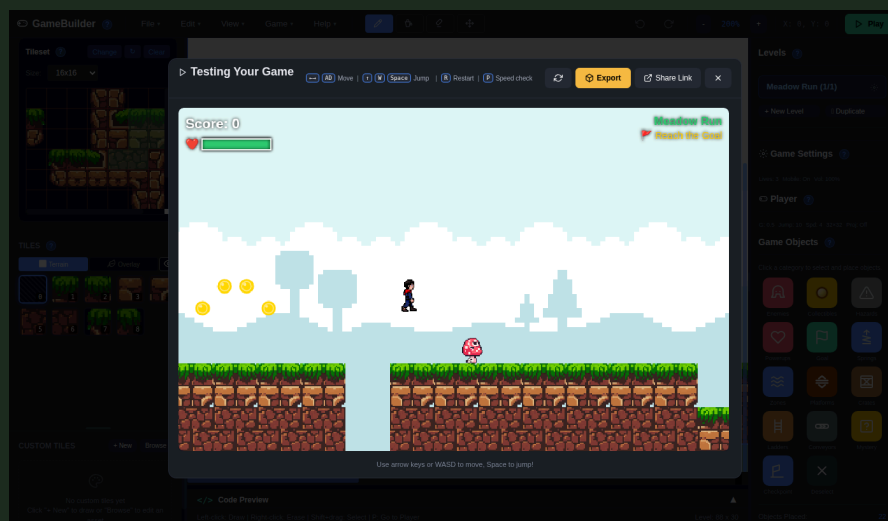


mytekDev

Game Builder: Change the Rules with Variables

Grades 4-8 · 45 minutes

Computer science · Game design



Slides, rubric and answer key

Print and go

No student accounts

Variables you can see

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
- 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 2-3 and 9-12
- Standards: CSTA 2017

Game Builder: Change the Rules with Variables

Grades 4-8 · 45 minutes · Computer science, Game design

Students play Meadow Run, a finished platform game, then change three numbers the game reads: Gravity, Jump Power and Move Speed. One small change traps the player in a valley, and that is the moment a variable makes sense: one number, used everywhere. By the end, each student has a game with rules of their own that still works, and a classmate who has to guess what they changed.

At a glance

ESSENTIAL QUESTION

How can changing one number change a whole game?

OBJECTIVES

Students will be able to:

- explain that a variable is a named number the game reads, and that changing it changes everything that uses it
- predict what a change to Gravity or Jump Power will do, then test the prediction and record what happened
- find and fix a part of a level that a new value has made impossible
- change at least two variables to give a game new rules that a player can still beat

VOCABULARY

variable, value, gravity, jump power, predict, bug, Code Preview

STANDARDS

1B-DA-07, 1B-AP-15, 2-AP-17

TIME AND MATERIALS

45 minutes, grades 4-8.

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

PREP

Make copies of the worksheet and exit ticket. Before class, open mytekdev.com/try/gamemaker?lesson=change-the-rules-with-variables&ref=tpt-gamebuilder-1 once on a student computer to see Meadow Run and to check that your school's filter allows it. No account is needed.

ASSESSMENT

- Formative: the questions under What to look for, asked while students work, and the exit ticket (answer key included).
- Summative: the worksheet's prediction table and the student's game with new rules, scored with the rubric out of 12.

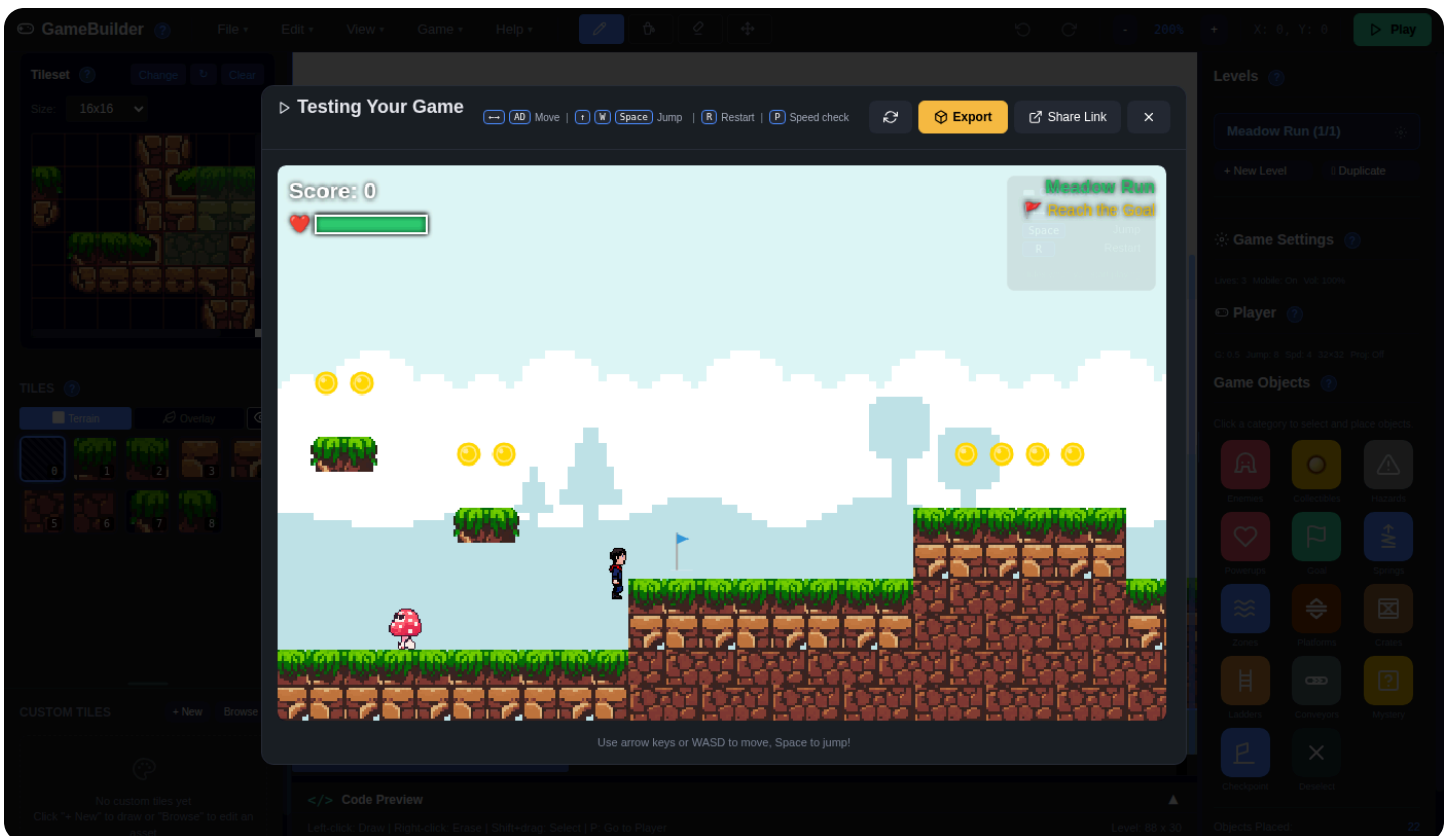
TECHNOLOGY AND PRIVACY

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 make is stored on our server unless they choose to save it to an account. If your school blocks new sites, ask IT to allow mytekdev.com.

Before you teach: do it once yourself

Break and fix Meadow Run yourself once before class, about ten minutes. The arrow keys run, Space jumps, R starts the level over and Esc goes back to the editor.

1. **Open the lesson.** Go to mytekdev.com/try/gamemaker?lesson=change-the-rules-with-variables&ref=tpt-gamebuilder-1. Meadow Run opens in the editor with a short guide along the bottom; press **Play**, run to the star, then press Esc.
2. **Read the variables.** Click **Player** on the right. The **Movement** tab shows **Gravity** 0.5, **Jump Power** 10 and **Move Speed** 4.
3. **Break it.** Set **Jump Power** to **8**, press **Done**, then **Play**, and run right. The player drops into the low valley with the mushroom and cannot jump back out of it.
4. **See it in the code.** Set **Jump Power** back to **10** and press **Done**. Click the **Code Preview** bar under the level, then click **Game Settings** on the right: the code shows `player.velocityY = -10` next to jump power and `+= 0.5` next to gravity; press **Done**.
5. **Gravity.** Set **Gravity** to **0.8** and play: the player is trapped in the valley again. Then try **0.3**, where the player floats and jumps much higher, and set it back to **0.5**.



Jump Power 8: the same level, and the player cannot get out of the valley.

If you see something else: Code Preview shows the gravity and jump code only after `Game Settings` is clicked, not `Player`. Esc switches between the editor and the game, so a second press starts the game again. The line under `Player` on the right shows the current G, Jump and Spd: check it after every test, because a number not set back spoils the next one.

Background in plain words

A variable is a name a program uses for a value it keeps, so the program can read the value wherever it needs it. In Game Builder, Gravity, Jump Power and Move Speed are variables. Each rule of the game is written once and reads those names, so changing one number changes that rule everywhere in the game, on every level, at once. The level itself, the tiles and the objects, is separate: it does not change when a number does.

The rules run 60 times a second, once a frame. Every frame, gravity is added to the player's falling speed. When Space is pressed and the player is on the ground, its upward speed is set to the Jump Power, and while an arrow key is held, its sideways speed is set to the Move Speed. The rules and the level have to fit each other: the valley in Meadow Run has walls 2 tiles tall, a Jump Power of 10 climbs them, and 8 climbs less than 2 tiles. A bigger gravity does the same thing from the other side, by taking the speed away sooner.

STUDENTS ASK	A GOOD ANSWER
Why is the jump power negative in the code?	On a screen, up means a smaller y. A negative speed moves the player up the screen.
Did I break the game?	No, the code still works. The numbers no longer fit the level, so you can fix the number or fix the level.
Why does bigger gravity make the jump lower?	Gravity pulls the player down every frame. Pulling harder stops the rise sooner, so the jump is lower.
Why not just put the number in each place it is used?	Then you would have to find and change every copy. With a variable you change it once, and everything that reads it changes too.

Lesson sequence

MINUTES	STAGE	WHAT YOU DO	WHAT STUDENTS DO
0 to 7	Warm-up Hook	Show the title, essential question and Today I can slides (about 1 minute). Then have students open the link or scan the QR code on the worksheet. A short guide runs along the bottom of the app: tell students to do its first step only, which is to press Play and run to the star, then press Esc and wait. The arrow keys run, Space jumps, R starts the level over and Esc goes back to the editor. Ask: "What one thing would make this game feel different?"	Play Meadow Run to the star. Think of one thing they would change.

MINUTES	STAGE	WHAT YOU DO	WHAT STUDENTS DO
7 to 15	Direct instruction (I do) Model it	Show the vocabulary slide. Then, on the projector, click Player on the right. The Movement tab shows Gravity 0.5, Jump Power 10 and Move Speed 4. Say: each one is a variable, a named number the game reads every time the player moves. Ask the class to predict what Jump Power 8 will do. Set it to 8, press Done and press Play. Run right: the player drops into the low valley with the mushroom and cannot jump back out, because its walls are 2 tiles tall and a jump of 8 now climbs less than 2. Press R to start over, then Esc. Say: the level did not change, only one number did, and that is a bug we made on purpose. Set Jump Power back to 10 and press Done. Then click the Code Preview bar under the level to open it, and click Game Settings on the right: the code shows <code>player.velocityY = -10</code> next to jump power and <code>+= 0.5</code> next to gravity. Those are the same numbers. Press Done.	Watch, predict out loud, and fill in worksheet step 1.
15 to 23	Guided practice (We do) Predict, then test	Students write a prediction for Gravity 0.8 in the table on the worksheet before anyone touches the app. Then together: click Player, set Gravity to 0.8, press Done, press Play, and run to the valley. Gravity 0.8 traps the player too: it pulls harder, so the jump is lower. Students write what happened. Repeat with Gravity 0.3: the player floats and jumps much higher. Before moving on, check that every table has a prediction and a result for both rows, and that everyone has set Gravity back to 0.5.	Write a prediction, test it, and write what happened, for Gravity 0.8 and then 0.3.
23 to 37	Independent practice (You do) Make it yours	Walk the room and ask the questions under What to look for. Repeat the three rules: change at least two numbers, one at a time; press Play after every change; a player must still be able to reach the star. A student whose change breaks a part fixes it either way: change the number back a little, or change the level. To change the level, pick a tile under Tiles on the left and click on the level to draw it (right-click erases), for example a step in the valley wall.	Pick new values, test each one, fix anything that breaks, and write the final numbers on the worksheet.
37 to 45	Closure Swap and exit ticket	Each builder presses Play before the swap, because the Player box on the right shows the numbers. Students trade seats and play a classmate's game for about 3 minutes, staying in the game (no Esc). They write which numbers they think were changed, and whether each went up or down. The builder then says what was really changed. Ask the essential question again, then hand out the exit ticket.	Play a classmate's game and guess what changed. Answer the exit ticket on their own.

What to look for

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

- Which number did you change, and what did you predict?
- Why does Jump Power 8 trap the player in the valley?
- Gravity 0.8 traps the player too. Why does a bigger gravity make the jump lower?
- Show me the jump power in Code Preview (click Game Settings, not Player).
- Your player cannot get past this part. Name two different ways to fix it.

The last one is the real test. A student who can fix it by changing the number or by changing the level understands that the rules and the level work together, and that a variable is one place to change a rule for the whole game.

Cross-curricular extensions

- Science: gravity on the Moon is about one sixth of gravity on Earth. Set Gravity to 0.1, the smallest it goes, and play. Is a Moon level easier or harder? (Answers vary: the player jumps very high and floats, so landing on a small platform takes care.)
- Math: Lesson 2 measures how high each Jump Power reaches, in tiles, and finds out whether doubling the jump doubles the height.
- English: write the rules of your game for a player who has never played, in three sentences or fewer, including how high and how fast the player moves.
- Computer science: open Code Preview and click Game Settings. Find the line for move speed and say what happens to it when the left arrow is pressed. (The code sets the speed to minus the Move Speed, so the player moves left.)

Vocabulary

variable	A name for a number the game keeps and reads. Change the number and everything that uses it changes.
value	The number a variable holds right now, like 10 for Jump Power.
gravity	How fast the player is pulled down. A bigger number pulls harder.
jump power	How fast the player starts moving up when it jumps. A bigger number jumps higher.
predict	Say what you think will happen before you test it.
bug	A part of a program that does not work the way it should.
Code Preview	The panel at the bottom of Game Builder that shows the code behind a setting, with your numbers in it.

Differentiation

ENGLISH LEARNERS

- Post the vocabulary with a picture for each word: the Player box, the valley, the Code Preview panel.
- Model each step on the projector before students try it, and point at the box as you name it.
- Let students fill the prediction table with arrows (up, down, same) instead of sentences.

STUDENTS WITH IEPs OR 504 PLANS

- Where the IEP allows a modified assignment, the goal is one changed variable that still lets a player reach the star; the second variable becomes an extension.
- Pair students: one changes the number, the other plays and reports what happened, then they swap.
- Raise the browser zoom so the Player box is easier to read.

- Read the exit ticket aloud, or let students answer by pointing at their screen.

STUDENTS WHO NEED MORE SUPPORT

- Give a sentence starter for the table: "I think the jump will be ... because ..."
- Suggest a safe range to explore: Gravity from 0.3 to 0.6, Jump Power from 9 to 14.
- Check in at minute 28: can a player still reach the star with the student's numbers?

STUDENTS READY FOR MORE

- Make a level part only your new jump can handle: if you raised Jump Power, draw a wall the old jump could not climb.
- Try Jump Mode, Double Jump, in the same Movement tab, and change the level to need it.
- Change Friction to 1 (ice). What happens when you let go of the arrow? Why does it not matter in the air?

Adapting for other grades

Younger students: For grades 2 and 3, do the guided test as a class on the projector, and have students change only Jump Power in Make it yours. Score only the first two rubric rows (out of 8; 6 or more meets the objectives).

Older students: For grades 9 to 12, ask students to find the three variables in Code Preview, explain in writing why Gravity 0.8 and Jump Power 8 both trap the player, and give their game a short design note: the feel they wanted, and the numbers that made it.

Student worksheet

Name: _____ Date: _____



Open the app: scan the QR code, or go to mytekdev.com/try/gamemaker?lesson=change-the-rules-with-variables

Essential question: How can changing one number change a whole game?

- 1. **Play it.** Press **Play**. Run with the arrow keys, jump with Space, and reach the star. **Esc** goes back.
- 2. **Name the variables.** Click **Player**. Write the three numbers on the Movement tab: Gravity ____ Jump Power ____ Move Speed ____
- 3. **Predict, then test.** Fill in each row of the table: write your prediction first, then change the number, press **Done** and **Play**.
- 4. **Make it yours.** Change at least two numbers. A player must still reach the star. Write your final numbers below.

CHANGE	I PREDICT...	WHAT HAPPENED
Gravity 0.8		
Gravity 0.3		
My change 1:		
My change 2:		

My final numbers: Gravity ____ Jump Power ____ Move Speed ____

- 1. Which prediction surprised you most, or was wrong? What did you learn from it?

- 2. My classmate changed: _____ (higher or lower?)

Exit ticket

Name: _____

1. With Jump Power 8 the player is stuck in the valley. What changed? (A) the level, (B) the value of a variable the game reads, (C) the player's picture.
2. Gravity goes from 0.5 to 0.8. The jump gets: (A) higher, (B) lower, (C) the same.
3. In a classmate's game, the player cannot climb a wall. Name two different ways to fix it.

Exit ticket

Name: _____

1. With Jump Power 8 the player is stuck in the valley. What changed? (A) the level, (B) the value of a variable the game reads, (C) the player's picture.
2. Gravity goes from 0.5 to 0.8. The jump gets: (A) higher, (B) lower, (C) the same.
3. In a classmate's game, the player cannot climb a wall. Name two different ways to fix it.

Answer key: exit ticket

QUESTION	ANSWER	WHY
1	B	The level is the same. Only the number the game reads for the jump changed, and the new jump is too low for the valley wall.
2	B	Gravity pulls the player down. A bigger gravity pulls harder, so the player stops rising sooner.
3	Any two of: raise Jump Power; lower Gravity; make the wall shorter; add a step or a platform in front of it	Full credit for any two fixes that would each work.

Rubric

Score each row from 1 to 4.

Name: _____

CRITERIA	4 EXCEEDS	3 MEETS	2 APPROACHING	1 BEGINNING
Predicts, then tests	Every row of the table has a prediction made before testing and a result, and the student explains the prediction that surprised them most.	Every row of the table has a prediction and a result.	Results are written, but some predictions are missing or were written after testing.	Values were changed, but nothing is written in the table.
New rules that still work	Two or more variables are changed, a player can reach the star, and one part of the level is built for the new rules.	Two or more variables are changed and a player can still reach the star.	Variables are changed, but a part of the level cannot be passed.	No more than one variable is changed, or the game is the same as the starter.
Explains variables	Explains what a variable is, and points to a changed value in Code Preview (opened from Game Settings).	Explains what a variable is, and says what the changed one does to the game.	Names the setting they changed, but not what it does.	Cannot yet say what was changed.

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

Standards

These CSTA 2017 standards describe testing, debugging and predicting from data. Students do not write the game's code in this lesson. They change the values its variables hold, predict what each change will do, and fix what a change breaks. Lesson 2 adds measuring and a math standard.

FRAMEWORK	CODE	GRADES	STANDARD	HOW THE LESSON MEETS IT
CSTA 2017	1B-DA-07	3-5	Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea.	Students predict what a new Gravity or Jump Power will do, test it, and record the result next to the prediction.
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 the part of the level a new value makes impossible, such as the valley at Jump Power 8, and fix it by changing the value or the level.
CSTA 2017	2-AP-17	6-8	Systematically test and refine programs using a range of test cases.	Students test one value at a time across a range, high and low, and keep the values that still let a player reach the star.

Checked October 2026 against: CSTA 2017 progression chart, CSTA 2017 to 2026 crosswalk.

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. Work stays in the browser tab until it is closed.

To keep students' work, and see every student's game 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/game-builder/lessons/change-the-rules-with-variables/start instead of the worksheet link. Each one gets their own copy of Meadow Run, and the same link reopens it next time. A student who started at the worksheet link and then signs up keeps what they made.

Game Builder is a mytekDev app, open source under the GNU General Public License 3.0. The goal star in Meadow Run is from Kenney's UI Pack, released under CC0.

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.

Game Builder is a mytekDev app, open source under the GNU General Public License 3.0. The goal star in Meadow Run is from Kenney's UI Pack, released under CC0.

© 2026 mytekDev. All rights reserved.