Verizon PLTW Coding and Game Design Facilitator Guide User Guide

Contents hide

verizon

Verizon PLTW Coding and Game Design Facilitator Guide

Coding and Game Design Facilitator Guide

Overview

The goal of this experience is to develop a STEM mindset while learning the concepts of video game design. Students will learn the basic constructs of a video game using the Scratch interface. Students use Scratch to learn about algorithms and event-driven programming. Object-oriented concepts are introduced through the use of sprites and the stage. Students use critical thinking and

creativity to build and enhance Hungry Mouse, a game that they develop using Scratch.

Materials
Students will need a computer or tablet with a web browser installed.

Preparation

1. Read through the teacher and student resources.
2. Ensure that your students' computers or tablets have internet connectivity.
3. Decide whether you will have your students use Scratch accounts.

Note: Scratch accounts are optional. However, working without them has limitations.

- If students have a Scratch account, they can log in to their Scratch account and save their work under their account. It will always be available for them to update in the future.
- If they do not have a Scratch account, then:
- If they are working on a computer, they will have to download the project to their computer in order to save their work, and they upload the project from their computer back to Scratch whenever they are ready to work on it again.
- If they are working on a tablet, they may be able to download and save files depending on the file storage of the tablet. If they are unable to download the project onto the tablet, they will need to complete their work in Scratch over the course of one session only. If they want to save their project; they will need to log in with an account.

If you decide to have your students use Scratch accounts, you can choose one of the two methods, based on your school's account creation policies:

- Students can join Scratch independently at **https://scratch.mit.edu/join**, as long as they have an email address.
- You can create student accounts, as long as you sign up as an educator. To do so, request a Scratch Teacher Account at **https://scratch.mit.edu/educators#teacheraccounts**. Once approved (which takes about a day or so), you can use your Teacher Account to create classes, add student accounts, and manage student projects. For more information, please refer to the Scratch FAQ page at **https://scratch.mit.edu/educators/faq**.

Essential Questions

- How do you overcome challenges and persist when solving problems?
- at are ways you can use programming skills to help yourself and others?

Session Length

- 90-120 minutes.

Note

1. Set workable time limits. It is so easy for students to spend too much time working on sprite costumes that they run out of time to completely develop the game!
2. The Extension Challenges section will add time based on how many extensions the students choose to complete.

Facilitation Notes

Begin this experience by watching the Coding and Game Design video with your students to learn more about the day in a life of a game developer.
Communicate with your students how they will work on and save their projects. If you have created a Scratch class or student account, make sure to share that information with your students.
Explore
Go over the contents of Table 1 with your students to make sure they understand the game requirements presented to them. Decide whether you want your students to collaborate using pair programming. In this paradigm, one student will be the driver (the one doing the programming) and the other will be the navigator (the one helping by reviewing code and helping catch errors and making suggestions for improvement). Pair programming's use in industry has shown that it improves collaboration and communication skills. Moreover, it enhances the quality of the software produced. If you use it in your classroom, make sure to have students switch roles regularly. It can either be every time they complete a task or every set number of minutes (like 15 minutes or so.)

Create
Ensure students understand how to access and save their work, whether when logged in or working as guest users. Check-in with students to make sure they understand the pseudocode provided for the Mouse's behavior. Encourage students to test their code repeatedly. This helps catch any bugs in the code early on. Remind
students that solutions rarely work on the first attempt. Solving problems takes patience and perseverance. Testing code often and fixing errors is one aspect of iteration that is common to design and development. The STEM Mindset section focuses further on perseverance. You can view and download the completed code for this game to use as a reference, HungryMouseCompleted, at **https://scratch.mit.edu/projects/365616252**.

STEM Mindset Allow students to follow the directions in the Student Guide to learn how to work in Scratch. Stress that this is a new learning experience. Let students know that evaluation does not solely rely on how the game works, but—more importantly—on how each one participates in the learning process. You will need to model a STEM mindset by stressing the idea that effort builds talent. The following are some phrases to use with students who struggle despite their strong effort:

- Mistakes are normal. This is new material.
- You are not there, yet.
- You might be struggling, but you are making progress.
- Don't give up until you feel proud.
- You can do it. It can be tough or confusing, but you are making progress.
- I admire your persistence.

When students need help with solutions, give them strategies to help themselves (do not always just tell them how to solve the situation):

- Which part is not working as expected? What was the expected behavior and how is it different from what is happening right now? What can be causing the issue?
- What part is difficult for you? Let's look at it.
- Let's think together about ways to improve this.
- Let me add this new bit of information to help you solve this.
- Here is a strategy to try so that you can begin to figure this out.
- Let's ask _____ for advice. S/He may have some ideas.

Answer Key

Explore

- Make Observations
- What do you see? Answer: I see two costumes: Mouse and Mouse-hurt.
- What do you think these are used for? Answer: Mouse is used to show a healthy mouse (before it is caught by the cat), and Mouse-hurt is used to show the mouse has been hurt by the cat.

Make Observations

- Make Observations What do you think are the Events blocks used for?
- Answer: They capture an event that happens, such as when a button is pressed or a sprite (or character) is clicked, and contain code that will run as a response to that event.

Make Observations

- How do you predict how each of the following will behave when the game starts? Student answers may vary. The correct predictions are:
- Mouse: Answer: The mouse will count down "get ready, get set, go!" and then will spin in place following the direction of the mouse-pointer.
- Cat1: Answer: The cat will move side to side on the screen indefinitely.
- Corn Bread: Answer: Nothing will happen to the cornbread, until touched by the mouse. Then it changes its look or disappears.
- Stage: Answer: It sets the score to 0 and the stage to the Woods backdrop.

Create

Make Observations

- What happens when the mouse collides with the cornbread? Answer: The cornbread changes to half-eaten the first time, and to completely gone the second time.
- What happens when the mouse collides with the cats? Answer: Nothing.
- Do these behaviors match the behavior described in the pseudocode above? Answer: The behavior of the cornbread is accurate, but the behavior of the mouse when it collides with the cats is incorrect. The mouse should change to a hurt mouse and the game should stop.

Make Observations

- Do the cats stop moving? Answer: No, they keep moving.
- Do all sprites disappear? Answer: Only the mouse disappears.
- Why or why not? Answer: The code for the mouse has a hidden block. But the other sprites do not have any code that tells them to hide when the game is over.

Extension Challenges

- **A.** Add other pieces of food that the mouse can collect and score more points. Solutions will vary. Students will need to add new sprites that will have event blocks similar to cornbread.
- **B.** Add other predators that can catch the mouse. Solutions will vary. Students will need to add new sprites that will have code that is similar to the cats.
- **C.** Change the behavior of the cats to be random across the screen. Refer to the sample solution, HungryMouseWithExtensions, at **https://scratch.mit.edu/projects/367513306**.
- Add a “You lose!” backdrop that will appear when the mouse is caught by its predators. Refer to the sample solution, HungryMouseWithExtensions, at **https://scratch.mit.edu/projects/367513306**.
- Add another level for the game. Solutions will vary.

Standards

Next Generation Science Standards (NGSS)

MS-ETS1-3 Engineering Design Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem.

ELA Common Core Standards

- CCSS.ELA-LITERACY.RI.6.7 Integrate information presented in different media or formats (e.g., visually, quantitatively) as well as in words to develop a coherent understanding of a topic or issue.
- CCSS.ELA-LITERACY.W.6.1, 7.1, and 8.1 Write arguments to support claims with clear reasons and relevant evidence.
- CCSS.ELA-LITERACY.W.6.2, 7.2 and 8.2 Write informative/explanatory texts to examine a topic and convey ideas, concepts, and information through the selection, organization, and analysis of relevant content.
- CCSS.ELA-LITERACY.SL.6.2 Interpret information presented in diverse media and formats (e.g., visually, quantitatively, orally) and explain how it contributes to a topic, text, or issue under study.
- CCSS.ELA-LITERACY.RST.6-8.1
- Cite specific textual evidence to support analysis of science and technical texts.
- CCSS.ELA-LITERACY.RST.6-8.3 Follow precisely a multistep procedure when carrying out experiments, taking measurements, or performing a technical task
- CCSS.ELA-LITERACY.RST.6-8.4
- Determine the meaning of symbols, key terms, and other domain-specific words and phrases as they are used in a specific scientific or technical context relevant to grades 6-8 texts and topic. CCSS.ELA-LITERACY.RST.6-8.7
- Integrate quantitative or technical information expressed in words in a text with a version of that information expressed visually (e.g., in a flowchart, diagram, model, graph, or table).
- CCSS.ELA-LITERACY.RST.6-8.9
- Compare and contrast the information gained from experiments, simulations, video, or multimedia sources with that gained from reading a text on the same topic.
- CSS.ELA-LITERACY.WHAT.6-8.2 Write informative/explanatory texts, including the narration of historical events, scientific procedures/ experiments, or technical processes.

Computer Science Teachers Association K-12

- 2-AP-10
- Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
- 2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-17
- Systematically test and refine programs using a range of test cases.

Documents / Resources

Verizon PLTW Coding and Game Design Facilitator Guide [pdf] User Guide
PLTW Coding and Game Design Facilitator Guide, PLTW, Coding and Game Design Facilitator Guide, Design Facilitator Guide, Facilitator Guide