Olin College of Engineering

Teacher's Guide for Scratch Game Development

Designed by Esme Abbot (Olin '24.5), Dre Hilton (Olin '24), Loren Lyttle (Olin '23), Prisha Sadhwani (Olin '23)





Do your students like video games? Do you want an interactive activity that will get your kids moving, while also teaching them about the coordinate grid? An activity that allows them to explore their creativity while learning the fundamental concepts of programming?

This activity will have the students learning how to program a game in Scratch using custom functions and the skills they learn in a preceding interactive activity to understand the basics of programming.

Workshop Requirements

This workshop is designed to be run independently, although the two hours can be split across multiple days

- **Suggested ages/grades:** Grades 6-8
- **Time required:** Two hours (one for initial game development, one for possible game extensions)
- **Required materials:**
  - Laptop with a free Scratch account (one for every group of 2-5 students, depending on class size and number of available computers)
    - the Base Game Code open
    - This workshop can be taught with very little Scratch experience, but you can click here for a brief introduction.
  - A coordinate grid on the floor (can be made of tape or carpet dots or name tag stickers)
  - One laminated copy of these arrow pages, printed on Tabloid (11"x17") Paper, along with tape to secure them to the floor
  - One laminated copy of this extensions handout for each table
  - One pool noodle, for coordinate grid activity

This Teacher's Guide and the accompanying materials were developed by students at Olin College in the course Mathematics and Engineering for Everyone, taught by Sarah Spence Adams, spring 2023. These materials are licensed under the Creative Commons Attribution-NonCommercial_ShareAlike 4.0 International license.

- Access to the Snake Game Extensions in Scratch
  - Extension: Disappearing Food
  - Extension: Continuously Moving Snake
  - Extension: Growing Snake
  - Extension: Obstacles

Optional Materials:

- Crayons and Markers

➢ **Prerequisite Knowledge:**

- Exposure to the coordinate grid

Learning Goals

- ➢ Gain experience with slopes and the coordinate grid
- ➢ Become comfortable going from physically moving on a grid to simulate movement on a computer
- ➢ Practice setting goals and experimenting with how playing around with code affects goals
- ➢ Understand how to work together as a team
- ➢ Get accustomed to the basics of pair-programing
- ➢ Gain confidence in explaining the work they do as individuals and sharing it out in a group

Workshop Flow

This section outlines the agenda that we used to run this workshop and the core activities. Feel free to modify this to meet your classroom needs!

- ➢ **Introduction** [10 minutes]
  - The goal of this is to get students moving and get some energy out! If some of the students don't know each other, or you don't know them, **The Name Game** can be a great way to do this:
    - Select a student to stand in the center of the circle with the pool noodle. This student is "it."
    - To start the game, a student from the circle calls out someone else's name. The student in the center must find that person and gently tap them with the pool noodle. But, there's a catch.
    - To avoid being "it," the person whose name was called can call out someone else's name. If they do so before getting tapped, they can stay in the circle. Then, the student in the center must try to find the newest target before they call out someone else's name.

This Teacher's Guide and the accompanying materials were developed by students at Olin College in the course Mathematics and Engineering for Everyone, taught by Sarah Spence Adams, spring 2023. These materials are licensed under the Creative Commons Attribution-NonCommercial_ShareAlike 4.0 International license.

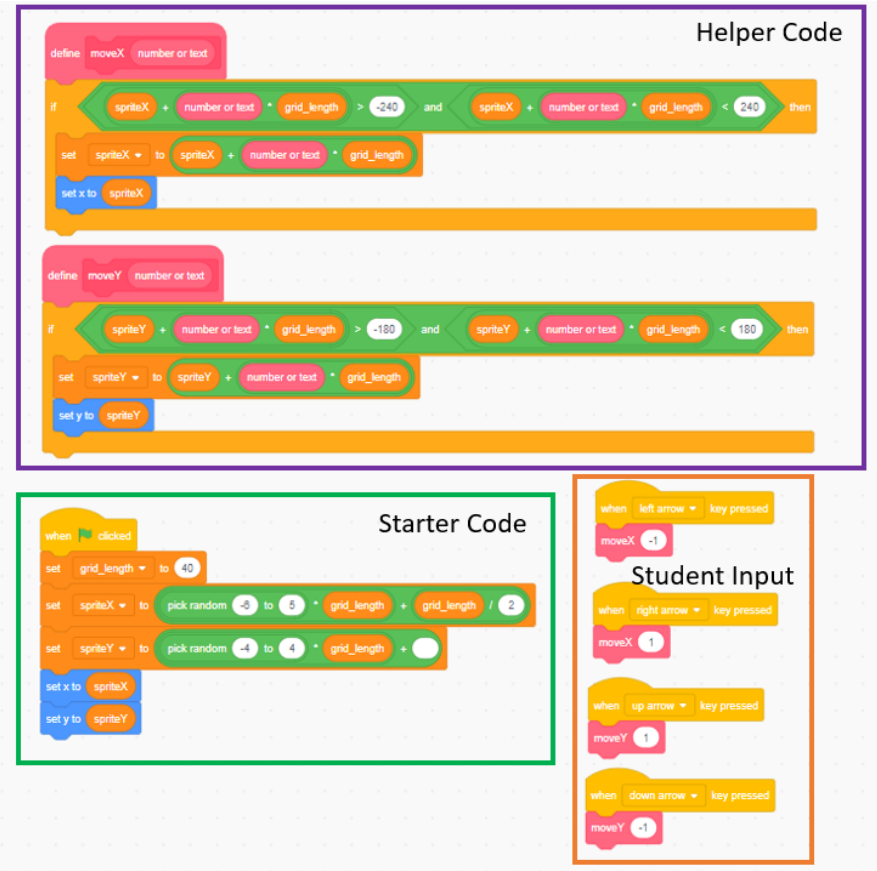
- **Physical Snake Game** [30 minutes]
  - In this phase, students gain an understanding of how a computer and a programmer connect a keyboard to a video game character by playing the game snake in real life
  - Before starting, group students by quadrants of the coordinate grid. Read the Pre Workshop Setup section, and have students label name tags and assemble their quadrant of the grid
  - Each game, each student takes on one of four roles:
    - Snake: This student moves around the grid trying to find food. They can only "understand" coordinate instructions from the computer and ideally can't see the "keyboard student."
    - Keyboard: This student steps on arrow keys to indicate which way the snake should move.
    - Computer: This student watches the Keyboard to see what buttons are pressed, and then tells the snake what coordinate to move to. This person is mimicking what a program would do and learning how to think algorithmically
    - Food: These students choose a position on the board and are Goals for the Snake. When the Snake reaches a Food student, the Food student follows the Snake until all Food students have been reached.
  - Example:
    - Snake is on position (0,0). Keyboard presses right. Computer tells Snake to move to (0, 1). Snake moves to (0,1). These steps repeat one more time to capture the Food at spot (0,2). Food is now attached to Snake, both at spot (0,2). The food joins the end of the snake and now moves according to the last position of the person in front of them in the snake line
  - Tip! To keep excitement up, try timing each game and have the class try to beat its time every game. Make sure each student is engaged in each game, and have students rotate through this role. Ideally, each student should get to do each role at least once.
- **Transition** [5-10 minutes]
  - Sort kids into table groups, ideally of 3-4, maximally 5 students each. This can either be on a self-determined basis, or prescribed depending on what works for your class!

If this is taught over 2 days, this is a great place to pause on Day 1, and then Day 2 picks up with programming in Scratch!

- **Scratch Part 1** [15 minutes]

This Teacher's Guide and the accompanying materials were developed by students at Olin College in the course Mathematics and Engineering for Everyone, taught by Sarah Spence Adams, spring 2023. These materials are licensed under the Creative Commons Attribution-NonCommercial_ShareAlike 4.0 International license.

- In this phase, each team works from the base Scratch game on their laptop to make the snake move around the screen. The finished game can be found here. If the team finishes early, have them try to diagram the helper code and understand what it is doing, or let them play with the game and brainstorm ways to make it more fun or engaging.



➢ **Share Out** [10 minutes]

- Have one student from each team share one thing that went well and one thing that was challenging from their scratch phase. Students can either stay in their groups, or you can have everyone circle up in the center of the room depending on energy level, space, and amount of time you have to transition.

➢ **Scratch Part 2 [40 minutes]**

- In this section, have each team choose an extension from the extensions handout and work through implementing it in scratch. Examples of each completed extension can be found here, and can be used to give teams hints:
  - Extension: Disappearing Food

This Teacher's Guide and the accompanying materials were developed by students at Olin College in the course Mathematics and Engineering for Everyone, taught by Sarah Spence Adams, spring 2023. These materials are licensed under the Creative Commons Attribution-NonCommercial_ShareAlike 4.0 International license.

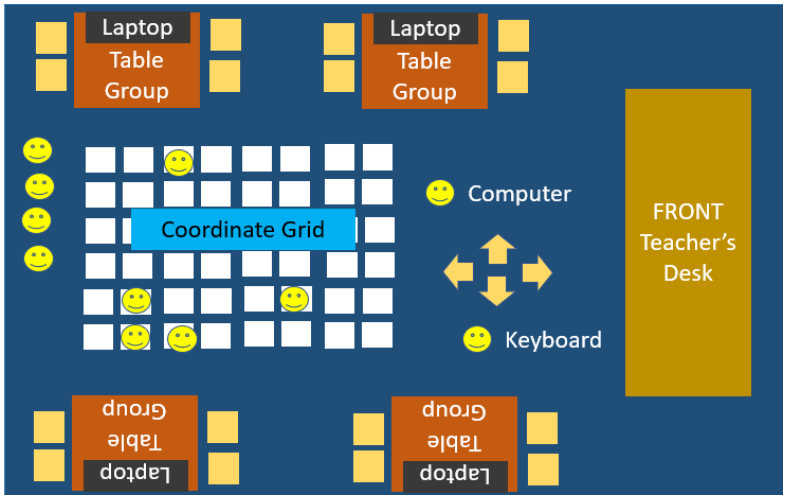
        - Extension: Continuously Moving Snake
        - Extension: Growing Snake
        - Extension: Obstacles
    - Having paper and pencil on hand can be great for students who learn visually, and can encourage the creation of pseudo-code as the programming challenges get more complex.
    - Encourage incremental approaches and lots of testing!
    - Optionally, have students rotate through a “Quality Engineer” role where their goal is to understand the limits of their code and try to find issues with it. Explain that this is a super important job in real life as well!

➢ **Share Out [10 minutes]**

- See Closing Questions and Student Reflections for potential share out mechanisms.

Suggested Tips for Running this Workshop

➢ **Pre-workshop preparation.** Physical room set up:

- The majority of the room should be the coordinate grid on the ground, which we made using name tags stickers with coordinates written on them. Setting this up takes time, but students could be involved in the placement of the coordinate grid to reinforce the concepts of X and Y axes before the start of the workshop.
- We suggest forming groups of tables placed around the edge of the room
- An example classroom setup is shown here:
- 

- We used an 8x8 coordinate grid, with each coordinate placed about 1.5’ away from each other, but this can be expanded or shrunk depending on the size of your classroom.

This Teacher’s Guide and the accompanying materials were developed by students at Olin College in the course Mathematics and Engineering for Everyone, taught by Sarah Spence Adams, spring 2023. These materials are licensed under the Creative Commons Attribution-NonCommercial_ShareAlike 4.0 International license.

- (0,0) can be placed at any corner of the grid, with numbers labeled in each direction, and arrows pointing towards the direction where the numbers are growing.

- **Usage of materials:**
  - We suggest having each laptop open to a remix of the starter code, this scratch project before the workshop starts to minimize transition time.
  - Keyboard handouts should be taped to the floor to prevent slipping, especially if laminated.
- **Tips for using handouts.**
  - Depending on students' comfort with the coordinate grid, it can be helpful to have the "Computer" hold onto the last handout from the arrow keys document so that they have a tool to translate from arrows to the coordinate plane. For increased challenge, this can be omitted.
  - We ran our workshop with each dot on the coordinate grid labeled. Again, for increased challenge, this can be omitted.
  - Some students may want to mark up the extension handout and circle the thing that they choose to work on. Dry or wet erase markers can be a useful tool for this, and it can be good to have crayons and paper on hand.
- **Guiding questions.** These questions can be helpful to ask when students are stuck and/or to help students connect this fun activity to the STEM content lurking beneath the surface.
  - In introducing the snake game activity:
    - "Where have you seen coordinate grids before?".... "Yep, so they show up in science AND math classes, because they're a really useful tool in the real world"
    - "Can someone explain to me how the video game snake works? How do you get the snake to move?"
  - In discussing video game design:
    - "What kind of video games do you like?"
    - "What makes video games fun?"
    - "How does the character move in X video game?"
  - In designing the scratch game:
    - "We have two options, moveX and moveY, can you show me which direction is the "X Axis" on this grid on the screen?"
- **Intentional commentary.** Comments such as the following can help students build and recognize STEM-related mindsets and skills:
  - A huge part of a programmer's job is knowing how to debug their code, but this can also be a really frustrating part of learning to code for younger kids. When this happens, it's helpful to a) have kids work in small chunks so that it's easier to figure out where the mistake may have occurred and b) that this is something that people who work on programming for their job deal with all of the time! It's important to know that while debugging can be frustrating, it doesn't mean that the student is "bad" at programming.
  - Especially during the physical game, it's important to thank and lift up each student for taking on roles. Roles like the computer can be stressful and challenging for students less comfortable with math, so be sure to give them a

This Teacher's Guide and the accompanying materials were developed by students at Olin College in the course Mathematics and Engineering for Everyone, taught by Sarah Spence Adams, spring 2023. These materials are licensed under the Creative Commons Attribution-NonCommercial_ShareAlike 4.0 International license.

strong signal that they are successful. Something like “Thanks so much [name] for taking on that challenging role. You did a great job doing those calculations so fast!” works well.

- **Other Helpful Tips**
  - Students generally have lots of good ideas for video game extensions. If they want to work on something that does not exist, you could encourage them to use their research skills to try and find something similar on Scratch.
  - It can be helpful to have two students on the “Computer” role during the physical game so that they can work together and one student doesn’t feel lost if they get overwhelmed.
  - If you’re new to Scratch or block based programming, check out the introduction to Scratch here!

Closing Questions and Student Reflections

- We found that thumb polls were a great way to gauge student engagement for each activity.
  - We suggest asking kids to rate the following with their thumbs on a scale from “two thumbs up!” to “two thumbs down!”
    - The overall workshop
    - The physical snake game
    - The video game snake
    - Programming in Scratch
- Encourage each team to share out what they did to the other teams. Perhaps allow enough time for everyone to rotate through gallery-walk style to play each of the extensions.
- A great closing question is to focus on how enjoyable being an engineer can be. For example, ask each student to write the answer to “What’s fun about being a videogame designer/programmer/software engineer?” on a sticky note.
  - A quick note on this: we noticed some kids had bad experiences with programming in the past that made it hard for them to be excited about the video game activity. By using “videogame designer” in this question, you can encourage a positive connection to a career in tech without using language that may already have a negative connotation to them.

Extensions and Enrichments

- Math Connections:
  - Technology has lightened a lot of the grunt work in game development over the past 20 years, but novel games cannot be built without a math foundation. Game developers need to use a bunch of math to make the

This Teacher’s Guide and the accompanying materials were developed by students at Olin College in the course Mathematics and Engineering for Everyone, taught by Sarah Spence Adams, spring 2023. These materials are licensed under the Creative Commons Attribution-NonCommercial_ShareAlike 4.0 International license.

games we love come to life. Everything from driving speed to intelligent non-player character (NPC) behavior is defined with arithmetic, geometry, and other mathematical relationships. Have you ever played Angry Birds? Think about how the game may use position differences to check if a bird has made contact with a tower.[1] Old pokemon games? How might they leverage probability to make rare pokemon truly feel like such a find?[2] What about Minecraft? How might you use math to make some blocks harder to break than others?[3]

- Engineering Connections:
  - Game making has visual, functional, and informational aspects that must combine to create a pleasant playable experience. Game development companies often divide this word into separate groups and require active collaboration between teams to create a cohesive product. Critical thinking takes place at all stages of the process to ensure that motivation, direction, and achievements are all conveyed to the player. The process is iterative, and as the games are play tested, teams must be willing to revise, restructure, or pivot from idea to idea.

- Expansions:
  - Expansions for our workshop should aim at getting the kids up and moving. If possible, we would like to develop the engineering iterative process by having them play-test their game rules. This can look like cardboard props, playing pretend, and running around on the floor to feel out how the game would be to play.
  - Depending on how the game is being developed, a final session where the scratch game is presented and played together will be great to boost confidence and inspire future projects.
- Links:
  [1] The physics of Angry Birds: how it works | TechRadar
  [2] https://youtu.be/YexvdFPWXdY?t=650
  [3] Minecraft Math Kit | Minecraft Education

Educational Design Principles Used to Create this Workshop

We are Olin College undergraduate engineering students who are practicing the engineering design cycle through the iterative design, development, testing, and improvement of STEM workshops for elementary (or middle) school students. We began our “user research phase” by visiting our partner students. We engaged our partner students with standard engineering design challenges as a way to get to know what they like, what motivates them, their fine motor building skills, their teaming skills, etc. This informed our creation of custom workshops for our partner

This Teacher’s Guide and the accompanying materials were developed by students at Olin College in the course Mathematics and Engineering for Everyone, taught by Sarah Spence Adams, spring 2023. These materials are licensed under the Creative Commons Attribution-NonCommercial_ShareAlike 4.0 International license.

students, which we then improved and tailored to work with a second class of students.

Our work has also been guided by our learnings about some educational design theories/principles. In this section, we provide information about how these educational design theories/principles have been used in the design of our workshop. We hope this will help teachers understand our motivations and therefore be more able to modify our workshops to work better for their students in their unique educational context.

Throughout our design work, we practiced keeping a tight loop between our goals (i.e., our learning objectives related to STEM content, social-emotional learning, and the development of confidence and belonging in STEM) and the activities we designed to achieve these goals.

- Goal #1: Increase student engagement by emphasizing autonomy.
- Interest in what we are teaching is one of the harder things to gauge when working with middle schoolers. This can happen for a multitude of reasons, from being around friends to the fact that their prefrontal cortex is still developing. To combat this, we wanted to add a layer of autonomy, which gave the students the sense that they were in control even if they weren't. This also uses UDL Strategies for engagement. This encouragement for autonomy can be empowering for students, as well as foster intrinsic motivation and independence.
- In order to achieve autonomy, we let the students drive the design methods (i.e. extensions and themes) with us as resources for guidance as opposed to decision makers.

- Goal #2: Introduce computer science.
- As we all know, the world is moving into a digital age. By taking this into account and teaching the students what the fundamental principles of coding are (even if not completely laid out), we can better prepare them for this future.
- By using Scratch as the basis for our workshop, we were able to introduce computer science through videogame design. This allowed for us to sneak in principles such as Events, basic logic (if-then statements and loops), and debugging strategies while remaining fun and engaging to the kids.

- Goal #3: Help the kids make connections between math and games.
- Based on an article by Math Geek Mama [3], it is clear that kids bridge mathematical gaps better in real life when they are shown the examples of math existing in real life. Because of this, we wanted to somehow integrate this into the lesson.
- By exploring how a snake moves in response to a keyboard and how blocks on the screen behave as a coordinate grid, we were able to show the kids how math and physics generally governed the way the snake moved.

This Teacher's Guide and the accompanying materials were developed by students at Olin College in the course Mathematics and Engineering for Everyone, taught by Sarah Spence Adams, spring 2023. These materials are licensed under the Creative Commons Attribution-NonCommercial_ShareAlike 4.0 International license.

- Goal #4: Provide scaffolding to help every student feel successful in the workshop.
- While working with middle schoolers, it is important to anticipate different digestion times - some middle schoolers will digest material when explained verbally, others visually or through tactile experiences.
- To provide multiple ways to engage with the workshop materials, we took into account principals from the Universal Design for Learning (UDL), which outlines that we should think about the perceptual representation. We created signs that explained directions of our live snake game, handouts paired with explanations of what we were going to do for the day, as well as one-on-one question answering provided by the teaching team.

Please use these materials and tailor them to your students!

Below are the links used for our snake extensions. Feel free to use them as reference for helping your students improve the base snake game.

- Extension: Disappearing Food
- Extension: Continuously Moving Snake
- Extension: Growing Snake
- Extension: Obstacles

References

References used in the motivation and design of our workshop and/or extensions.

[1] https://www.techradar.com/news/software/applications/the-physics-of-angry-birds-how-it-works-1067809

[2] https://youtu.be/YexvdFPWXdY?t=650

[3] https://mathgeekmama.com/math-in-real-life/

This Teacher's Guide and the accompanying materials were developed by students at Olin College in the course Mathematics and Engineering for Everyone, taught by Sarah Spence Adams, spring 2023. These materials are licensed under the Creative Commons Attribution-NonCommercial_ShareAlike 4.0 International license.