

Unit 6 Lesson 1

My Robotic Friends

Resources

Unit 6 Lesson 2

Coding with Comments

Resources

Unit 6 Lesson 3

Building a Foundation

Resources

Unit 6 Lesson 4

Debugging with Scrat

Resources

Unit 6 Lesson 5

Creating Art with Code

Resources

Unit 6 Lesson 6

My Loopy Robotic Friends

Resources

Unit 6 Lesson 7

Drawing Shapes with Loops

Resources

Unit 6 Lesson 8

Nested Loops in Maze

Resources

Unit 6 Lesson 9

Nested Loops with Frozen

Resources

Unit 6 Lesson 10

Conditionals with Cards

Resources

Unit 6 Lesson 11

Conditionals with the Farmer

Resources

Unit 6 Lesson 12

Private and Personal Information

Resources

Unit 6 Lesson 13

Build a Star Wars Game

Resources

Unit 6 Lesson 14

Songwriting

Resources

Unit 6 Lesson 15

Functions in Minecraft

Resources

Unit 6 Lesson 16

Functions with Harvester

Resources

Unit 6 Lesson 17

Digital Sharing

Resources

Unit 6 Lesson 18

Functions with Artist

Resources

Unit 6 Lesson 19

Determine the Concept

Resources

Unit 6 Lesson 20

Learning Sprite Lab

Resources

Sprite Lab Curriculum Pilot

Introduction

What is Sprite Lab?

Sprite Lab is a new tool that we're building to replace Play Lab in Courses E & F. We've heard a *lot* of feedback about the limitations and challenges of creating open-ended projects in Play Lab. Play Lab is a great tool for many of our skillbuilding progressions, but it isn't set-up to enable students to imagine and create expressive, personal, and interesting projects.

Sprite Lab is **designed to be used by 4th and 5th graders to create interactive apps, simulations, and games**. Unlike Play Lab, Sprite Lab is not designed to tell linear stories – we plan to keep Play Lab around for those! – but we hope that Sprite Lab will be better for going beyond what students can do in Play Lab.

What is the Curriculum Pilot?

The Code.org team has been running classroom tests, gathering feedback, and making changes for the last few months. Now we're ready to expand the scope of that testing and feedback to include teachers and students outside of the Code.org community. **We are looking for teachers with 4th and 5th grade students who have completed at least part of Courses E or F** to participate. If you are interested in participating and haven't formally signed up yet, please do so here: <https://bit.ly/csfpilot>. Signing up on that form will help us stay in touch with you about updates and changes.

As part of participating in the pilot, we'll be asking for your help to identify problems and areas of improvement. This means:

- **We will ask you to join at least two pilot calls with the Code.org team.** We will send out more information on when these calls will be each week, but it's important that we gather feedback from you directly on how each lesson went for your class. We know you're busy – we'll provide multiple scheduling options and work with you the best we can.
- **We'll email you no more than twice per-week with information about curriculum updates.** These emails will stop after June 11, and you can opt-out at any time.
- **We'll provide a survey for each lesson.** Please fill these out to let us know how things are going with your students.
- **We hope you'll submit bug reports!** This is a new product, which means we are still ironing out lots of bugs. We ask that you submit bug reports by emailing support@code.org, and mentioning that you are sending feedback on the Sprite Lab Pilot.

Your help is vital in making this new product a success!

Sprite Lab Curriculum Pilot

How Stable is Sprite Lab?

Sprite Lab is brand new, which means we are still ironing out lots of wrinkles. Because of this, **things might break or change at any time.** We will do our best to try to warn you about upcoming changes, but we ask for your patience and flexibility during this pilot.

Curriculum Overview

The curriculum we are testing this year is not meant to teach new computer science concepts. The curriculum we are piloting is primarily about helping students learn to use the tool so that they can create projects of their own design. We will be releasing the curriculum one lesson at a time. You can find the currently released lessons at <https://studio.code.org/s/spritelab>.

Recommended Student Profile:

- Completion of Course D or approximately half of Courses E or F
- 4th or 5th grade

Lesson 1: Introduction to Sprites and Behaviors (Fish Tank)

In this lesson, students will learn about the two concepts at the heart of Sprite Lab: **sprites** and **behaviors**. Sprites are characters or objects on the screen that students can move, change, and manipulate. Behaviors are actions that sprites will take continuously until they are stopped.

[Lesson Plan for this Lesson](#)

[Lesson on Code.org](#)

Lesson 2: Input (Alien Dance Party)

In this lesson, students will build on their knowledge of sprites and behaviors to add **events** to their program. Students should be familiar with events from earlier lessons. Students will use events to create an Alien Dance Crew that will start and stop dancing on command.

[Lesson Plan for this Lesson](#)

[Lesson on Code.org](#)

Sprite Lab Curriculum Pilot

Lesson 3: Sprite Interactions (Virtual Pet)

In this lesson, students will use **when touched** events to make their sprites interact with one another. They'll combine these events with the other things they've learned to create and care for a pet giraffe.

[Lesson Plan for this Lesson](#)

[Lesson on Code.org](#)

Open Project Sandbox

If your students would like to go beyond, we invite you to try our open-ended sandbox at

<https://studio.code.org/p/spritelab>.

Unit 6 Lesson 21

Alien Dance Party

Resources

Sprite Lab Curriculum Pilot

Introduction

What is Sprite Lab?

Sprite Lab is a new tool that we're building to replace Play Lab in Courses E & F. We've heard a *lot* of feedback about the limitations and challenges of creating open-ended projects in Play Lab. Play Lab is a great tool for many of our skillbuilding progressions, but it isn't set-up to enable students to imagine and create expressive, personal, and interesting projects.

Sprite Lab is **designed to be used by 4th and 5th graders to create interactive apps, simulations, and games**. Unlike Play Lab, Sprite Lab is not designed to tell linear stories – we plan to keep Play Lab around for those! – but we hope that Sprite Lab will be better for going beyond what students can do in Play Lab.

What is the Curriculum Pilot?

The Code.org team has been running classroom tests, gathering feedback, and making changes for the last few months. Now we're ready to expand the scope of that testing and feedback to include teachers and students outside of the Code.org community. **We are looking for teachers with 4th and 5th grade students who have completed at least part of Courses E or F** to participate. If you are interested in participating and haven't formally signed up yet, please do so here: <https://bit.ly/csfpilot>. Signing up on that form will help us stay in touch with you about updates and changes.

As part of participating in the pilot, we'll be asking for your help to identify problems and areas of improvement. This means:

- **We will ask you to join at least two pilot calls with the Code.org team.** We will send out more information on when these calls will be each week, but it's important that we gather feedback from you directly on how each lesson went for your class. We know you're busy – we'll provide multiple scheduling options and work with you the best we can.
- **We'll email you no more than twice per-week with information about curriculum updates.** These emails will stop after June 11, and you can opt-out at any time.
- **We'll provide a survey for each lesson.** Please fill these out to let us know how things are going with your students.
- **We hope you'll submit bug reports!** This is a new product, which means we are still ironing out lots of bugs. We ask that you submit bug reports by emailing support@code.org, and mentioning that you are sending feedback on the Sprite Lab Pilot.

Your help is vital in making this new product a success!

Sprite Lab Curriculum Pilot

How Stable is Sprite Lab?

Sprite Lab is brand new, which means we are still ironing out lots of wrinkles. Because of this, **things might break or change at any time.** We will do our best to try to warn you about upcoming changes, but we ask for your patience and flexibility during this pilot.

Curriculum Overview

The curriculum we are testing this year is not meant to teach new computer science concepts. The curriculum we are piloting is primarily about helping students learn to use the tool so that they can create projects of their own design. We will be releasing the curriculum one lesson at a time. You can find the currently released lessons at <https://studio.code.org/s/spritelab>.

Recommended Student Profile:

- Completion of Course D or approximately half of Courses E or F
- 4th or 5th grade

Lesson 1: Introduction to Sprites and Behaviors (Fish Tank)

In this lesson, students will learn about the two concepts at the heart of Sprite Lab: **sprites** and **behaviors**. Sprites are characters or objects on the screen that students can move, change, and manipulate. Behaviors are actions that sprites will take continuously until they are stopped.

[Lesson Plan for this Lesson](#)

[Lesson on Code.org](#)

Lesson 2: Input (Alien Dance Party)

In this lesson, students will build on their knowledge of sprites and behaviors to add **events** to their program. Students should be familiar with events from earlier lessons. Students will use events to create an Alien Dance Crew that will start and stop dancing on command.

[Lesson Plan for this Lesson](#)

[Lesson on Code.org](#)

Sprite Lab Curriculum Pilot

Lesson 3: Sprite Interactions (Virtual Pet)

In this lesson, students will use **when touched** events to make their sprites interact with one another. They'll combine these events with the other things they've learned to create and care for a pet giraffe.

[Lesson Plan for this Lesson](#)

[Lesson on Code.org](#)

Open Project Sandbox

If your students would like to go beyond, we invite you to try our open-ended sandbox at

<https://studio.code.org/p/spritelab>.

Unit 6 Lesson 22

Pet Giraffe

Resources

Unit 6 Lesson 23

Explore Project Ideas

Resources

Use this worksheet to make sure you are following the right steps to get your final project completed correctly and on time. If more space is needed for answers, you can use the back of the paper.

Day 1: Define & Prepare

Time to plan the details of your final project!



Section A: Determine your project style

1) What type of project will you be creating?	Sprite Lab (skip section B) or Artist (skip section C)
2) Do you want this to be a showcase piece or an interactive game?	Showcase Piece (skip section D) or Interactive App (skip section E)
3) Is your project a remix or brand new? (If remix, add URL of original)	
4) Describe in three to five sentences what your project will be.	

Section B: Artist Project

1) Will your drawing be a planned, detailed image which is the same each time, or change each time it's run?	
2) If your drawing stays the same each time it's run, what is the design you are trying to build, and why does it make an interesting project? (More room on back of sheet.)	

Final Project Design Worksheet



3) If your drawing changes each time it is run, describe why it changes and how the program will make that happen.

Section C: Sprite Lab Project / Other

1) How many characters will your game have?

2) Will your app be a story that plays the same way each time it is run, or is it meant to change each time?

3) If your app is a story that runs the same way each time, describe the story it will be telling and the scene where it will take place.

4) If your app is meant to change each time it is run, describe why it changes and how the program will make that happen.

Section D: Interactive App

<p>1) In what way does your app require audience interaction?</p>	
<p>2) For this type of app, you must use at least two of these suggested items.</p> <p>Circle the items you think you will use and describe what you might do with them.</p> <p>(5th grade concepts in red)</p>	<ul style="list-style-type: none">• Random• Conditionals with both `if` and `else`• Nested Loops• Functions• Variables• For Loops

Section E: Showcase Piece

<p>1) For this type of app, you must use at least two of these suggested items.</p> <p>Circle the items you think you will use and describe what you might do with them.</p>	<ul style="list-style-type: none">• Random• Functions• Nested Loops• Variables• For Loops
--	---

Day 2: Try

Coding day! Be sure to refer back to your table from day 1 to make sure your finished product represents the project that you said you were going to make.

Section F: What is going well?

1) What parts of your project are turning out just the way you thought they would?	
2) What are you learning about the concepts or blocks that you're using?	

Section G: What is not going well?

1) Is there anything that you planned to do that you needed to change? Why did you need to change it?	
2) Are there any blocks or concepts that you don't understand how to use? Which ones?	

Day 3: Reflect and Edit

Time to make your project the best it can be. Pass this sheet to a classmate, then have them try your app and fill out the boxes below. Your classmates will have 20 minutes to review your app, then you will have the rest of the class period to fix any issues that they found.

Share your project URL:

--

Section H: The app in general

1) What do you like about this app?	
2) Is there anything about this app you don't understand?	
3) Name two things you might change about this app if you were the one creating it.	

Section I: Assessing the app

1) Does the app match the description from day 1?	
2) Does the app use at least 2 of the concepts from the required category?	
3) Does the app feel like it was well-planned?	

Day 4: Present

Great computer scientists need to be able to show off their accomplishments. It's time to present your work.

Share your project URL:

--

Section J: Describe your work

1) Why did you decide to create this specific project?	
2) What does your project do?	
3) Which of the required ideas did you use, and how did you use them?	

Section K: Reflect on the process

1) What did you learn in creating this project?	
2) What was the most challenging part about this experience? How did you overcome that challenge?	
3) What part of this experience was the most fun?	
4) Describe a time that something did not go as planned and you had to be persistent to get everything to work out in the end.	

Final Project Design Worksheet



5) Is there anything that you would try to change if you had extra time?

6) How well did you and your partner/team work together?

Were you able to make compromises in order to end up with a project that you were both happy with?

7) What else do you want people to know about your project?

The Design Process

Designing software means solving lots of little problems, all the time. The main problem in software design is what to create in the first place.

This process is useful for all kinds of things, but we are going to focus on using it for app design.



- **Define**
 - What kind of app would you like to create?
 - What are your constraints?
 - What does success look like?
- **Prepare**
 - Brainstorm / research possible elements
 - Compare pros and cons
 - Make a plan
- **Try**
 - Put your plan into action
- **Reflect**
 - How do your results compare to the goals you set while defining the app?
 - What can you learn from this or do better next time?
 - What new problems have you discovered?

What it Looks Like

Over the course of the next several weeks, you will have the opportunity to experiment with some existing games and design your own game based off of what you have learned. After creating your game, you will get the chance to present it to others and receive feedback. These steps are all critically important in the software industry, and getting practice with the elements of the design process will help you create better products more efficiently. Here is what the coming weeks will hold as we learn more about the design process.

1. Define & Prepare

- Play existing games to get ideas and understand limitations
- Brainstorm and plan your new or modified app

2. Try

- Follow your plan to build an app

3. Reflect & Edit

- Swap apps with another group to help make your projects better

4. Present

- Show off your final product!

Unit 6 Lesson 24

The Design Process

Resources

Unit 6 Lesson 25

Build Your Project

Resources

Unit 6 Lesson 26

Present Your Project

Resources

Unit 6 Lesson 27

The Internet

Resources

Unit 6 Lesson 28

Crowdsourcing

Resources