

Lesson 1: Programming with Angry Birds

38 minutes

Overview

In this **skill-building** lesson, students will develop sequential algorithms to move a bird from one side of a maze to the pig on the other side. To do this, they will stack code blocks together in a linear sequence, making them move straight, turn left, or turn right.

Purpose

In this lesson, students will develop programming skills on a computer platform. The block-based format of these puzzles help students learn about sequence and concepts, without having to worry about perfecting syntax.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- **AP** - Algorithms & Programming

Agenda

Warm Up (3 minutes)

Introduction

Main Activity (30 minutes)

Programming with Angry Birds

Wrap Up (5 minutes)

Reflection

Extended Learning

Cross-Curricular Opportunity

Objectives

Students will be able to:

- Identify and locate bugs in a program.
- Translate movements into a series of commands.

Preparation

- Play through the puzzles to find any potential problem areas for your class.
- (Optional) Pick a couple of puzzles to do as a group with your class.
- Make sure every student has a reflection journal.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the students

- **Unplugged Maze Blocks** -

Resource [▼ Make a Copy](#)

Vocabulary

- **Algorithm** - A list of steps to finish a task.
- **Bug** - Part of a program that does not work correctly.
- **Debugging** - Finding and fixing problems in an algorithm or program.
- **Sequencing** - Putting commands in correct order so computers can read the commands.

Teaching Guide

Warm Up (3 minutes)

Introduction

Model: Pull up Puzzle 5 to do in front of the class. While working through this puzzle with the class, remind students that making mistakes is okay and remind them that the only way to be successful is to be persistent. Discuss what to do when a program doesn't work (debug it!) or how to get through the frustration that can come with working on a computer.

Next, you'll need to describe how the blocks in the workspace move the bird toward the pig. Show students how to drag blocks from the toolbox and connect them beneath the `when run` block, but don't solve the puzzle.

Discuss: Think about how we would get the bird to the pig using arrows. How do we use these blocks instead?

Have students use their fingers to point the direction that the bird should go next. Once you feel like you have a classroom consensus, try to get students to put into words which block will make that action happen. Roll your mouse over different options and have them shout "Yes" or "No".

Drag blocks into place one at a time, then click "Run" after each one. This will not only let them see how far the bird has gone, but set good habits for when they start working to solve their own puzzles.

Continue this pattern, fixing bugs as they arise, until the bird successfully gets to the pig.

Additional Demonstration

We've included some multiple choice prediction levels. These could be used after finishing the Warm Up.

Prediction Levels:

- [Programming in Maze #1](#)
- [Programming in Maze #2](#)

Transition: Now that students have seen an online puzzle in practice, they should be ready to start solving puzzles of their own. Continue to the lab or bring out their classroom computers.

💡 Teaching Tip

Some students may struggle with turning their bird in the correct direction, particularly when the bird isn't facing up. Remind students that when we say turn left or right, we're giving directions from the bird's point of view.

Main Activity (30 minutes)

Programming with Angry Birds



Video: Maze Intro - Programming with Blocks

Show the students the right way to help classmates by: Don't sit in their chair Don't use their keyboard Don't touch their mouse Make sure the classmate can describe the solution before you walk away



2-7

Skill Building

2

3

4

5

6

7



8

Challenge



9

Practice



10

Prediction



11

Practice



12-13

Challenge

12

13

Circulate: Teachers play a vital role in computer science education and supporting a collaborative and vibrant classroom environment. During online activities, the role of the teacher is primarily one of encouragement and support. Online lessons are meant to be student-centered, so teachers should avoid stepping in when students get stuck. Some ideas on how to do this are:

- Utilize Pair Programming whenever possible during the activity.
- Encourage students with questions/challenges to start by asking their partner.
- Unanswered questions can be escalated to a nearby group, who might already know the solution.
- Have students describe the problem that they're seeing. What is it supposed to do? What does it do? What does that tell you?
- Remind frustrated students that frustration is a step on the path to learning, and that persistence will pay off.
- If a student is still stuck after all of this, ask leading questions to get the student to spot an error on their own.

Discuss: After providing students with end-of-class warnings, grab everyone's attention and get them to reflect on the experiences that they just had.

- Did anyone feel frustrated during any of the puzzles?
- Did anyone notice the need to be persistent?

Transition: Have students grab their ***Thinkspot Journals** and take a moment to reflect on the experience for themselves.

Wrap Up (5 minutes)

Reflection

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today. When completed, this can be used as a review sheet for them to look at in the future.

Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- Draw an activity you like to do that you struggled with the first time. Draw or describe how you got better.

Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

Create Your Own

In small groups, let students design their own mazes and challenge each other to write programs to solve them. For added fun, make life-size mazes with students as the pig and bird.

Cross-Curricular Opportunity

Bugs, Bugs & More Bugs (45-60 minutes)

Computer Science + English Language Arts + Math

Bugs, Bugs & More Bugs is an optional activity aligned to Common Core ELA and Math standards, written by our teacher community. Students will practice debugging as they fix code, correct a math problem and proofread writing samples.

Standards Addressed:

- **CCSS.ELA-LITERACY.L.2.1.B:** Form and use frequently occurring irregular plural nouns (e.g., feet, children, teeth, mice, fish).
- **CCSS.ELA-LITERACY.L.2.2.A:** Capitalize holidays, product names, and geographic names.
- **CCSS.ELA-LITERACY.L.2.2.C:** Use an apostrophe to form contractions and frequently occurring possessives.
- **CCSS.MATH.CONTENT.2.OA.B.2:** Fluently add and subtract within 20 using mental strategies.
- **CCSS.MATH.CONTENT.2.NBT.A.4:** Compare two three-digit numbers based on meanings of the hundreds, tens, and ones digits, using $>$, $=$, and $<$ symbols to record the results of comparisons.



This work is available under a **Creative Commons License (CC BY-NC-SA 4.0)**.

If you are interested in licensing Code.org materials for commercial purposes **contact us**.